

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

2.1.1 Pengertian Sistem

Dalam kehidupan sehari-hari, kita sering mendengar istilah “sistem”. Secara umum, sistem dapat diartikan sebagai suatu kesatuan yang terdiri dari elemen-elemen yang saling berhubungan dan bekerja sama untuk mencapai suatu tujuan tertentu. Istilah ini tidak hanya berlaku di bidang teknologi, tetapi juga dalam bidang organisasi, pendidikan, ekonomi, bahkan dalam kehidupan sosial. Konsep sistem sangat luas dan bisa diaplikasikan dalam berbagai konteks.

Menurut pendapat ahli Jogiyanto (2024), Sistem adalah kumpulan dari elemen – elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem ini menggambarkan suatu kejadian – kejadian dan kesatuan yang nyata adalah suatu objek nyata, seperti tempat, benda dan orang-orang yang betul-betul ada dan terjadi. Dalam definisi ini, sistem tidak hanya dipandang sebagai sekumpulan komponen, tetapi juga sebagai suatu proses yang melibatkan interaksi dan keterkaitan antara bagian-bagian di dalamnya. Artinya, setiap komponen dalam sistem memiliki peran masing-masing yang saling menunjang satu sama lain. Pandangan Jogiyanto ini didukung oleh penelitian Sari & Nugroho (2021) yang menggarisbawahi bahwa keterpaduan prosedur dan alur kerja menjadi kunci utama dalam keberhasilan implementasi sistem informasi, khususnya dalam konteks organisasi bisnis.

Sementara itu, Ludwig von Bertalanffy sebagai pelopor teori sistem menyatakan bahwa sistem adalah kesatuan dari elemen-elemen yang saling berinteraksi dan saling memengaruhi dalam satu kesatuan yang terorganisir. Pandangan ini menekankan bahwa sistem bukanlah sesuatu yang berdiri sendiri, tetapi terbentuk dari elemen-elemen yang saling berkaitan erat. Interaksi antar elemen inilah yang membuat sistem

dapat berfungsi secara optimal. Penelitian Prabowo (2020) memperkuat pandangan ini dengan menunjukkan bahwa sistem yang tidak memperhatikan hubungan antar komponennya akan mudah mengalami disfungsi, terutama dalam penerapan sistem berbasis teknologi informasi.

Sebagai contoh sederhana, sebuah sistem transportasi terdiri dari berbagai elemen seperti kendaraan, rambu lalu lintas, jalan, pengemudi, dan peraturan. Jika salah satu komponen tidak berjalan sebagaimana mestinya, maka sistem secara keseluruhan akan terganggu. Hal yang sama berlaku pada sistem informasi di sebuah perusahaan atau usaha rumahan seperti konveksi, di mana elemen-elemen seperti data, perangkat lunak, perangkat keras, prosedur kerja, dan sumber daya manusia harus saling terintegrasi agar dapat mendukung pencapaian tujuan perusahaan secara efektif dan efisien. Studi dari Wulandari et al. (2022) yang meneliti pengembangan sistem informasi berbasis web pada industri kecil menunjukkan bahwa sistem yang dirancang dengan mempertimbangkan integrasi antar elemen dapat meningkatkan efisiensi, mempercepat aliran informasi, dan mengurangi kesalahan data.

Lebih lanjut, pendekatan sistemik juga diterapkan dalam manajemen usaha kecil dan menengah (UKM). Penelitian oleh Ramadhan & Fitriani (2022) menekankan bahwa dengan memandang usaha sebagai suatu sistem, pemilik usaha dapat lebih mudah dalam mengelola proses operasional, mulai dari pencatatan keuangan hingga pengelolaan stok barang. Penelitian ini juga menyebutkan bahwa sistem yang dirancang dengan baik mampu meningkatkan akurasi pengambilan keputusan. Senada dengan itu, Nurfadillah (2023) menguraikan bahwa penerapan sistem informasi dalam usaha konveksi membantu pemilik usaha untuk meminimalkan risiko kesalahan pencatatan serta mempercepat proses monitoring keuangan dan inventori.

Dari berbagai pendapat dan penelitian tersebut, dapat disimpulkan bahwa sistem adalah suatu kesatuan yang terdiri dari berbagai elemen yang saling berhubungan dan saling mendukung satu sama lain untuk mencapai tujuan tertentu. Dalam konteks penelitian ini, pemahaman

terhadap konsep sistem menjadi sangat penting karena menjadi dasar dalam perancangan dan pengembangan sistem informasi yang dapat membantu perusahaan, khususnya konveksi SCOTDN, dalam mengelola keuangan dan stok barang secara lebih terstruktur, efisien, dan minim risiko kesalahan.

2.1.2 Rancang Bangun

Rancang bangun merupakan istilah yang sering digunakan dalam dunia rekayasa, teknologi informasi, dan arsitektur sistem. Secara umum, rancang bangun dapat diartikan sebagai suatu proses perencanaan dan pengembangan terhadap suatu sistem, perangkat, atau bangunan secara menyeluruh — dimulai dari tahap perancangan (desain), hingga ke tahap pembangunan (implementasi) dari suatu konsep menjadi sesuatu yang fungsional dan dapat digunakan.

Dalam konteks sistem informasi, rancang bangun mengacu pada proses merancang struktur dan komponen sistem berdasarkan kebutuhan pengguna, kemudian merealisasikannya dalam bentuk aplikasi atau program yang siap digunakan. Proses ini tidak hanya menitikberatkan pada bagaimana sistem tersebut terlihat dari sisi antarmuka (user interface), tetapi juga memperhatikan bagaimana sistem bekerja di balik layar (logika sistem, alur data, basis data, dan sebagainya).

Menurut Rosa dan Salahudin (2020), rancang bangun perangkat lunak merupakan fase krusial dalam siklus pengembangan sistem yang menjembatani kebutuhan fungsional dan non-fungsional pengguna dengan solusi teknis yang akan diimplementasikan. Tahap ini mencakup perumusan model sistem, organisasi struktur data, perancangan algoritma dan proses, serta desain antarmuka pengguna yang intuitif. Secara praktis, rancang bangun menjadi landasan utama sebelum sistem diimplementasikan dan dioperasikan.

Sebagai contoh, dalam membangun sistem informasi pembukuan keuangan dan manajemen stok untuk konveksi SCOTDN, tahap rancang bangun mencakup kegiatan seperti menggali kebutuhan pemilik usaha, menyusun diagram alur kerja, mendesain struktur database, hingga

menyusun desain tampilan halaman web yang akan digunakan. Tanpa proses perancangan yang matang, sistem yang dibangun kemungkinan besar tidak akan sesuai dengan kebutuhan pengguna, bahkan bisa berujung pada ketidakefisienan dan tidak dapat digunakan secara optimal.

Dengan demikian, rancang bangun dapat disimpulkan sebagai suatu proses menyeluruh yang mencakup analisis kebutuhan, perancangan sistem, serta pengembangan hingga sistem tersebut siap digunakan. Proses ini memegang peranan penting dalam menciptakan sistem informasi yang efektif, efisien, dan mampu menyelesaikan permasalahan nyata yang dihadapi oleh suatu organisasi atau perusahaan.

2.1.3 Sistem Informasi

Menurut O'Brien dan Marakas (2020), sistem informasi dapat didefinisikan sebagai kombinasi terorganisir dari orang-orang, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi untuk mendukung operasi dan manajemen dalam suatu organisasi. Definisi ini menyoroti bahwa sistem informasi bukanlah entitas tunggal, melainkan sebuah ekosistem yang bekerja secara sinergis. Manusia di dalamnya berperan sebagai pengguna, pengembang, dan pengambil keputusan; fasilitas teknologi seperti komputer, server, dan perangkat jaringan menjadi tulang punggung operasional; media penyimpanan dan transmisi data memastikan informasi tersedia; prosedur dan kebijakan mengatur alur kerja; dan pengendalian diterapkan untuk menjaga integritas dan keamanan data. Intinya, sistem informasi dirancang untuk merampingkan proses bisnis, mendukung transaksi rutin, serta menyediakan data yang esensial bagi manajemen dan para pemangku kepentingan, baik internal maupun eksternal, untuk membuat keputusan yang tepat.

Sejalan dengan pandangan tersebut, Laudon dan Laudon (2022) menekankan bahwa sistem informasi adalah sistem di dalam suatu organisasi yang dirancang untuk mendukung operasional harian, mulai dari pengolahan transaksi hingga penyediaan laporan yang dibutuhkan

untuk perencanaan dan pengendalian manajerial. Mereka menyoroti bahwa sistem informasi memiliki peran ganda: pertama, sebagai alat untuk mengotomatisasi proses bisnis yang berulang dan volume tinggi, seperti entri pesanan atau pembayaran gaji; kedua, sebagai sarana untuk menghasilkan wawasan strategis melalui analisis data, yang pada gilirannya membantu organisasi beradaptasi dengan perubahan pasar dan mencapai keunggulan kompetitif. Oleh karena itu, sebuah sistem informasi yang efektif tidak hanya mempercepat alur kerja, tetapi juga meningkatkan kemampuan organisasi untuk merespons secara dinamis terhadap tantangan dan peluang.

2.1.4 Pembukuan Keuangan

Pembukuan keuangan adalah proses pencatatan secara sistematis terhadap seluruh transaksi keuangan yang terjadi dalam suatu entitas, baik itu perusahaan, organisasi, maupun individu. Kegiatan ini mencakup pencatatan pemasukan, pengeluaran, aset, kewajiban, hingga modal yang digunakan atau dihasilkan selama periode tertentu. Tujuan utama dari pembukuan adalah untuk memberikan gambaran yang jelas dan akurat mengenai kondisi keuangan, sehingga memudahkan dalam pengambilan keputusan, pelaporan pajak, maupun evaluasi kinerja usaha.

Dalam praktiknya, pembukuan keuangan tidak hanya sebatas mencatat angka, tetapi juga melibatkan pengelompokan data sesuai kategori, memastikan ketepatan waktu pencatatan, dan menjaga konsistensi serta keakuratan informasi. Dengan pembukuan yang tertib dan teratur, pelaku usaha bisa mengetahui sejauh mana aktivitas bisnis mereka berjalan dengan sehat secara finansial.

2.1.5 Manajemen Stok

Manajemen stok adalah serangkaian kegiatan yang berkaitan dengan pengelolaan persediaan barang dalam suatu usaha atau organisasi, mulai dari proses perencanaan, pengadaan, penyimpanan, hingga pendistribusian barang tersebut. Tujuan dari manajemen stok adalah untuk memastikan bahwa ketersediaan barang selalu dalam jumlah yang tepat—

tidak kekurangan yang bisa menghambat operasional, dan tidak berlebihan yang bisa menimbulkan pemborosan.

Dalam praktiknya, manajemen stok bukan hanya soal menghitung jumlah barang yang ada di gudang. Lebih dari itu, manajemen stok mencakup strategi pengaturan alur masuk dan keluar barang, pemantauan masa simpan (khususnya untuk barang yang mudah rusak), serta penentuan titik pemesanan ulang agar pasokan tetap stabil. Pengelolaan stok yang baik akan membantu usaha berjalan lebih efisien, menghindari risiko kerugian, dan menjaga kepuasan pelanggan karena kebutuhan mereka bisa dipenuhi tepat waktu.

2.1.6 *Rapid Application Development*

Rapid Application Development (RAD) adalah pendekatan dalam pengembangan perangkat lunak yang menekankan pada kecepatan proses pembuatan aplikasi dengan tetap menjaga kualitas hasil akhirnya. Metode ini muncul sebagai respons terhadap kebutuhan akan sistem yang bisa dikembangkan dalam waktu relatif singkat, terutama ketika kebutuhan pengguna sering berubah atau berkembang seiring waktu.



Gambar.2. 1 *Rapid Application Development*

Dalam metode RAD, proses pengembangan dilakukan secara iteratif dan fleksibel, artinya tim pengembang akan membangun prototipe (contoh awal aplikasi) terlebih dahulu, kemudian memperbaikinya secara bertahap berdasarkan masukan dari pengguna. Pendekatan ini sangat mengandalkan kolaborasi antara tim teknis dan pengguna akhir, agar aplikasi yang dihasilkan benar-benar sesuai dengan kebutuhan di lapangan.

Keunggulan dari RAD terletak pada kemampuannya menyesuaikan diri dengan perubahan, mengurangi waktu pengembangan, serta mendorong keterlibatan aktif dari semua pihak yang terlibat. Namun, agar RAD berjalan dengan efektif, dibutuhkan tim yang solid, komunikasi yang baik, serta pengguna yang responsif dalam memberikan umpan balik.

2.1.7 Angular

Angular adalah sebuah *framework open-source* berbasis JavaScript yang dikembangkan oleh Google. Framework ini digunakan untuk membangun aplikasi web yang dinamis dan interaktif, terutama aplikasi berbasis *single-page (SPA)*, di mana pengguna bisa berinteraksi dengan halaman tanpa harus memuat ulang seluruh isi website setiap kali berpindah tampilan.

Angular memanfaatkan bahasa pemrograman *TypeScript*, yang merupakan pengembangan dari JavaScript dengan fitur tambahan seperti tipe data statis dan alat bantu pengembangan yang lebih kuat. Salah satu kekuatan utama Angular adalah strukturnya yang terorganisir dan komponen-komponennya yang bisa digunakan kembali, sehingga memudahkan tim pengembang dalam membuat aplikasi yang berskala besar dan kompleks.

Selain itu, Angular juga dilengkapi dengan fitur-fitur canggih seperti *two-way data binding* (sinkronisasi otomatis antara data dan tampilan), *dependency injection* (untuk memudahkan pengelolaan dependensi), serta sistem routing yang memungkinkan navigasi antar halaman secara efisien. Semua ini membuat Angular menjadi pilihan yang cukup populer di kalangan developer untuk membangun aplikasi web modern.

2.1.8 NODE.JS

Node.js adalah sebuah platform runtime yang memungkinkan kita menjalankan kode JavaScript di luar lingkungan browser. Jadi, kalau biasanya JavaScript hanya digunakan di sisi client (di dalam browser), dengan Node.js kita bisa menjalankan JavaScript di sisi server. Ini

menjadikan Node.js sangat berguna untuk membangun aplikasi web yang cepat, ringan, dan efisien.

Node.js dibangun di atas mesin JavaScript V8 milik Google Chrome, yang terkenal karena performanya yang tinggi. Salah satu ciri khas Node.js adalah arsitekturnya yang non-blocking dan berbasis event-driven. Artinya, Node.js mampu menangani banyak permintaan sekaligus tanpa harus menunggu satu proses selesai dulu baru lanjut ke proses berikutnya. Hal ini membuat Node.js sangat cocok digunakan untuk aplikasi real-time seperti chat, notifikasi, atau sistem yang melibatkan banyak koneksi secara bersamaan.

Selain itu, ekosistem Node.js juga sangat luas, dengan adanya npm (*Node Package Manager*) yang menyediakan ribuan paket atau modul siap pakai, sehingga pengembangan aplikasi menjadi jauh lebih cepat dan praktis. Karena fleksibilitas dan performanya, Node.js banyak digunakan oleh perusahaan teknologi besar maupun developer independen untuk membangun berbagai jenis aplikasi, mulai dari API backend, server streaming, hingga aplikasi IoT.

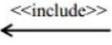
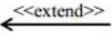
2.1.9 Unifield Modeling Language

Unified Modeling Language (UML) adalah sebuah bahasa visual standar yang digunakan untuk memodelkan, merancang, dan mendokumentasikan sistem perangkat lunak. Dengan kata lain, UML membantu para pengembang, analis sistem, maupun pemangku kepentingan lainnya untuk memahami bagaimana suatu sistem bekerja melalui representasi grafis yang mudah dipahami.

UML tidak hanya terbatas untuk menggambarkan struktur sistem, tetapi juga mencakup perilaku, alur kerja, interaksi antar komponen, hingga proses bisnis yang mendukung sistem tersebut. Terdapat berbagai jenis diagram dalam UML, seperti *use case diagram*, *class diagram*, *activity diagram*, *sequence diagram*, dan lain sebagainya. Masing-masing diagram memiliki fungsi tersendiri, tergantung aspek sistem apa yang ingin dijelaskan.

Salah satu alasan mengapa UML banyak digunakan adalah karena sifatnya yang seragam dan dapat dipahami lintas tim, baik oleh programmer, analis, manajer proyek, hingga klien non-teknis. Dengan adanya model UML, proses komunikasi antar pihak menjadi lebih mudah, dan pengembangan sistem bisa dilakukan secara lebih terstruktur, terencana, serta minim kesalahan.

2.1.10 Use Case Diagram

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

Gambar 2. 2 Notasi pada Use Case Diagram

Sumber: dicoding.com

Use Case Diagram adalah salah satu jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan interaksi antara pengguna (yang disebut aktor) dan sistem yang sedang dikembangkan. Diagram ini berfungsi untuk menunjukkan apa saja yang bisa dilakukan oleh pengguna terhadap sistem, bukan bagaimana sistem itu bekerja secara teknis di dalamnya.

Dengan kata lain, *use case diagram* memberikan gambaran umum tentang fungsi-fungsi utama yang tersedia dalam sebuah sistem dari sudut

pandang pengguna. Setiap use case mewakili satu aktivitas atau layanan yang disediakan sistem, misalnya "login", "mengisi formulir", atau "melihat laporan". Sementara aktor bisa berupa manusia, sistem lain, atau perangkat yang berinteraksi dengan sistem utama.

Use case diagram sering digunakan di tahap awal pengembangan sistem, karena dapat membantu tim pengembang dan pemangku kepentingan untuk menyepakati kebutuhan sistem dengan cara yang visual dan mudah dimengerti. Diagram ini juga bermanfaat dalam mendokumentasikan ruang lingkup sistem secara jelas dan terstruktur.

2.1.11 Activity Diagram

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Gambar 2. 3 Notasi pada Use Case Activity Diagram

Sumber: dicoding.com

Activity Diagram adalah salah satu jenis diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan alur kerja atau urutan aktivitas dalam sebuah proses bisnis atau sistem. Diagram ini menunjukkan bagaimana sebuah proses dimulai, bagaimana langkah-langkahnya dijalankan,

bagaimana keputusan dibuat di tengah jalan, hingga bagaimana proses tersebut berakhir.

Secara visual, activity diagram menyerupai diagram alir (*flowchart*), namun lebih fokus pada aspek aktivitas dan alur proses dari satu langkah ke langkah lainnya dalam sistem. Setiap aktivitas dalam diagram biasanya digambarkan sebagai kotak atau oval, sedangkan panah menunjukkan urutan atau transisi antar aktivitas. Ada juga simbol keputusan yang menunjukkan percabangan proses berdasarkan kondisi tertentu.

Activity diagram sangat berguna dalam menjelaskan proses yang kompleks, baik dalam sistem informasi maupun dalam kegiatan bisnis sehari-hari. Diagram ini membantu semua pihak yang terlibat, baik tim teknis maupun non-teknis untuk memahami bagaimana sebuah proses berjalan dari awal hingga akhir, dan di mana saja potensi pengambilan keputusan atau kondisi khusus terjadi.

Dengan menggunakan activity diagram, proses analisis sistem menjadi lebih jelas dan terstruktur, sehingga memudahkan pengembang dalam merancang sistem yang sesuai kebutuhan.

2.1.12 Sequence Diagram

Sequence Diagram adalah salah satu jenis diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan urutan interaksi antar objek dalam suatu sistem dari waktu ke waktu. Diagram ini memvisualisasikan bagaimana pesan atau perintah dikirim antar objek secara berurutan, dalam rangka menjalankan suatu proses atau fungsi tertentu di dalam sistem.

Setiap objek dalam sequence diagram diwakili oleh sebuah garis vertikal yang disebut *lifeline*, dan interaksi antar objek digambarkan menggunakan panah horizontal yang menunjukkan aliran pesan. Panah ini menjelaskan siapa yang mengirim pesan, siapa yang menerima, serta urutan terjadinya komunikasi tersebut. Dengan begitu, *sequence diagram* bisa membantu memperlihatkan dinamika proses secara detail mulai dari siapa yang memulai interaksi, bagaimana proses itu mengalir, hingga kapan interaksi tersebut berakhir.

Sequence diagram sangat berguna terutama pada tahap perancangan sistem, karena diagram ini memberikan gambaran yang jelas tentang bagaimana

berbagai komponen dalam sistem saling berinteraksi. Ini tidak hanya membantu tim pengembang dalam memahami logika proses, tetapi juga memudahkan identifikasi kesalahan atau ketidaksesuaian sejak awal.

Dengan kata lain, sequence diagram membantu menjembatani pemahaman antara konsep logika sistem dan implementasi teknisnya melalui visualisasi interaksi yang runtut dan sistematis.

2.1.13 Class Diagram

Class Diagram adalah salah satu jenis diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk memodelkan struktur statis sebuah sistem perangkat lunak. Diagram ini menggambarkan kelas-kelas yang ada dalam sistem, atribut (data) yang dimiliki oleh setiap kelas, serta hubungan antar kelas tersebut.

Setiap kelas dalam diagram ini biasanya direpresentasikan dengan sebuah kotak yang dibagi menjadi tiga bagian: nama kelas, atribut (informasi atau data yang disimpan), dan operasi atau metode (fungsi yang bisa dijalankan oleh kelas tersebut). Selain itu, class diagram juga menunjukkan bagaimana kelas-kelas itu saling berhubungan, misalnya melalui asosiasi, pewarisan (inheritance), atau agregasi.

Class diagram penting karena memberikan gambaran jelas tentang bagaimana data dan perilaku sistem diorganisasi dan saling terhubung. Dengan diagram ini, pengembang bisa memahami struktur inti sistem secara menyeluruh, yang sangat berguna saat merancang database, menulis kode, atau mendokumentasikan sistem.

Selain itu, class diagram juga membantu memastikan bahwa setiap bagian sistem sudah terdefinisi dengan baik dan sesuai dengan kebutuhan bisnis atau fungsional yang diinginkan. Karena sifatnya yang mendetail dan terstruktur, class diagram sering menjadi dasar bagi proses pengembangan perangkat lunak yang terorganisir dan efisien.

2.1.14 Black Box Testing

Black Box Testing adalah salah satu metode pengujian perangkat lunak yang fokus pada pengujian fungsi dan perilaku sistem tanpa melihat struktur internal atau kode programnya. Dengan kata lain, pada tahap ini, penguji hanya

memperhatikan input yang dimasukkan ke dalam aplikasi dan output yang dihasilkan, tanpa mengetahui bagaimana proses di dalam sistem berjalan.

Tujuan utama dari *Black Box Testing* adalah memastikan bahwa sistem bekerja sesuai dengan spesifikasi dan kebutuhan pengguna. Pengujian ini biasanya dilakukan dengan mencoba berbagai skenario penggunaan, termasuk kondisi normal, batasan, dan kemungkinan kesalahan, untuk melihat apakah sistem memberikan respons yang benar dan sesuai harapan.

Metode ini sangat berguna karena bisa dilakukan oleh orang yang bukan pengembang, seperti tester independen atau bahkan pengguna akhir, sehingga memberikan sudut pandang yang objektif terhadap kualitas aplikasi. Namun, karena penguji tidak melihat kode, *Black Box Testing* tidak dapat menemukan kesalahan yang terkait dengan logika internal atau struktur program.

Singkatnya, *Black Box Testing* membantu memastikan bahwa aplikasi “berfungsi dengan benar” dari sisi pengguna, tanpa harus memahami detail teknis di balik layar.

2.1.15 Website

Mengacu pada definisi modern, Rosa dan Salahudin (2020) menjelaskan bahwa website merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen dalam website dikenal sebagai laman web, dan tautan dalam website memungkinkan pengguna untuk berpindah dari satu halaman ke halaman lain, baik antar halaman yang disimpan di server yang sama maupun antar server yang tersebar di seluruh dunia. Laman web ini dapat diakses atau dibaca melalui peramban web populer seperti Google Chrome, Mozilla Firefox, Microsoft Edge, dan lainnya. Lebih dari sekadar kumpulan dokumen, sebuah website modern seringkali berfungsi sebagai platform interaktif yang menyediakan berbagai layanan, mulai dari e-commerce, media sosial, hingga aplikasi berbasis web yang kompleks.

Menurut Laudon dan Laudon (2022) menekankan bahwa website adalah salah satu bentuk sistem informasi berbasis web yang menyediakan antarmuka bagi pengguna untuk berinteraksi dengan data dan layanan yang disimpan di server. Website tidak hanya menyajikan informasi statis, tetapi juga memungkinkan interaksi dinamis seperti pengisian formulir, pencarian data,

hingga transaksi daring. Desain dan fungsionalitas sebuah website menjadi krusial dalam menentukan pengalaman pengguna dan efektivitas tujuan yang ingin dicapai, baik itu tujuan komersial, informasional, maupun fungsional.

2.2 Tinjauan Studi

Dalam penulisan penelitian ini dilakukan pengamatan dengan beberapa penelitian sebelumnya antara lain :

1. Tinjauan studi pertama diambil dari penelitian berjudul “Perancangan Sistem Informasi Keuangan Berbasis Web pada GKS Mauliru Menggunakan Metode *Rapid Application Development*” yang ditulis oleh Eben Panja dan Danny Manongga, diterbitkan dalam JATI: Jurnal Mahasiswa Teknik Informatika, Vol. 7 No. 1, Februari 2023. Penelitian ini mengkaji permasalahan pengelolaan keuangan di lingkungan Gereja Kristen Sumba (GKS) Mauliru yang masih dilakukan secara manual melalui Excel dan pembukuan fisik, yang menyebabkan ketidakefisienan serta risiko keterlambatan dan kesalahan dalam pelaporan keuangan antar cabang gereja. Sistem informasi yang dikembangkan bertujuan memfasilitasi pencatatan dan pelaporan keuangan secara cepat dan terpusat melalui platform web, yang memungkinkan tiap cabang gereja untuk melakukan input data secara mandiri. Pengembangan sistem menggunakan metode *Rapid Application Development (RAD)* karena dianggap mampu mempercepat siklus pembangunan aplikasi dan mempermudah proses revisi melalui prototipe interaktif. Penelitian ini melibatkan tahapan perencanaan kebutuhan, desain sistem menggunakan *UML (use case, activity diagram, sequence diagram)*, implementasi sistem berbasis web, dan pengujian dengan metode *black box*. Hasil uji fungsional menunjukkan seluruh fitur utama berjalan dengan baik, seperti pencatatan transaksi, pengelolaan kategori, serta pembuatan laporan keuangan dalam format PDF. Studi ini relevan sebagai acuan dalam pengembangan sistem serupa untuk organisasi atau komunitas berbasis cabang, termasuk usaha kecil atau UMKM, karena menyoroti pentingnya digitalisasi proses keuangan guna meningkatkan efisiensi, transparansi,

dan akurasi laporan keuangan secara real-time. Pendekatan RAD juga menjadi referensi metodologis yang tepat untuk pengembangan sistem dengan kebutuhan adaptasi yang cepat dan waktu implementasi yang terbatas.

2. Tinjauan studi ini diambil dari penelitian berjudul "Sistem Informasi Penggajian Pegawai Pada SMK Berbasis Website dengan Menggunakan Metode Rapid Application Development" yang ditulis oleh Mila Rosanah, Audy Nur Amaliyah, dan Angga Ardiansyah, dan diterbitkan dalam JASIKA (Jurnal Sistem Informasi Akuntansi) Vol. 03, No. 01, Mei 2023. Penelitian tersebut mengkaji pentingnya sistem informasi yang efektif dan efisien untuk pengelolaan penggajian pegawai di Sekolah Menengah Kejuruan (SMK). Permasalahan yang diidentifikasi adalah kebutuhan akan sistematisasi dalam pengelolaan data dan proses penggajian agar dapat menghasilkan laporan yang akurat dan tepat waktu. Sistem informasi yang dikembangkan dalam penelitian ini bertujuan untuk mengelola data dan proses penggajian pegawai, serta menyediakan laporan gaji yang akurat dan tepat waktu. Untuk mencapai tujuan tersebut, metode *Rapid Application Development (RAD)* dipilih karena fokusnya pada kecepatan dan iterasi dalam pengembangan sistem. Penerapan metode RAD diyakini dapat mempercepat waktu pengembangan sistem, meningkatkan fleksibilitas dalam menghadapi perubahan kebutuhan, dan memastikan keakuratan data yang diolah. Tahapan RAD yang digunakan meliputi *Requirements Planning, Design Workshop, dan Implementation*. Hasil dari penelitian ini menunjukkan bahwa sistem yang dibangun mampu mengelola berbagai data, termasuk data staf keuangan, pegawai, absensi, jabatan, serta komponen-komponen gaji seperti honor, insentif, potongan, gaji pokok, dan tunjangan. Fungsionalitas sistem dirancang untuk mengakomodasi berbagai peran: administrator memiliki akses penuh untuk mengelola seluruh sistem, staf keuangan dapat mengelola data terkait keuangan dan pegawai (kecuali data administrator dan staf keuangan lainnya), kepala sekolah dapat melihat data pegawai dan laporan penggajian, dan pegawai dapat melakukan absensi, mengelola data

pribadi, serta mencetak slip gaji. Studi ini sangat relevan sebagai referensi dalam pengembangan sistem informasi yang membutuhkan kecepatan dan adaptasi terhadap perubahan kebutuhan, seperti pada pengelolaan pembukuan keuangan dan manajemen stok, karena menunjukkan bagaimana pendekatan RAD dapat menghasilkan sistem yang terstruktur, terkoordinasi, efektif, efisien, dan sesuai dengan kebutuhan pengguna.

3. Tinjauan studi ketiga berasal dari penelitian berjudul “Rancang Bangun Sistem Informasi Akuntansi Berbasis Web Menggunakan *Rapid Application Development* Model di PT Adhitama Mitra Nusantara” yang ditulis oleh March Angga Vehryza Wardhana dkk., dan dipublikasikan dalam prosiding SNESTIK III tahun 2023. Penelitian ini membahas bagaimana pengelolaan akuntansi secara manual di perusahaan menyebabkan proses pencatatan transaksi dan pembuatan laporan keuangan menjadi lambat dan rawan kesalahan. Untuk mengatasi masalah tersebut, peneliti membangun sistem informasi akuntansi berbasis web dengan metode *Rapid Application Development (RAD)*, yang memungkinkan pengembangan sistem dalam waktu singkat dengan proses iteratif. Sistem yang dikembangkan memiliki fitur utama seperti input transaksi, pembuatan jurnal otomatis, hingga penyusunan laporan keuangan seperti laporan laba rugi dan neraca yang dapat diunduh dalam format PDF maupun Excel. Penelitian ini juga menunjukkan bahwa sistem yang dibuat mudah digunakan dan sesuai dengan kebutuhan pengguna, berdasarkan hasil uji kelayakan menggunakan standar ISO 9126. Studi ini sangat relevan bagi mahasiswa Sistem Informasi karena menunjukkan penerapan nyata metode RAD dalam merancang sistem akuntansi yang efisien dan terintegrasi, serta memberikan gambaran lengkap tentang bagaimana teknologi informasi dapat menyederhanakan proses bisnis dalam dunia kerja.

4. Tinjauan studi keempat berasal dari penelitian berjudul “*Perancangan Sistem Informasi Penjualan pada Toko Royal Ice Cream dengan Metode Rapid Application Development*” yang ditulis oleh Vivi Maria dkk., dan dipublikasikan dalam *Jurnal of Informatics (JIKA)* Vol. 7 No. 4, Oktober

2023. Penelitian ini membahas permasalahan yang umum terjadi di toko skala kecil, yaitu sistem pencatatan transaksi yang masih dilakukan secara manual, seperti penggunaan kalkulator untuk menghitung total penjualan, pencatatan stok yang tidak terintegrasi, dan kesulitan dalam menyusun laporan keuangan harian maupun bulanan. Untuk menjawab permasalahan tersebut, peneliti membangun sistem informasi penjualan berbasis web menggunakan framework Laravel dengan pendekatan metode *Rapid Application Development (RAD)*, yang melibatkan empat tahapan utama: analisis kebutuhan, pemodelan sistem, desain, dan konstruksi. Sistem ini dirancang menggunakan alat bantu seperti *Use Case Diagram*, *Activity Diagram*, *ERD (Entity Relationship Diagram)*, dan *LRS (Logical Record Structure)* untuk menggambarkan proses bisnis secara detail. Sistem yang dikembangkan memfasilitasi admin, kasir, dan bagian gudang dalam melakukan pengelolaan data penjualan, pembelian, stok barang, hingga laporan keuangan secara terpusat dan otomatis. Bagi mahasiswa Sistem Informasi, studi ini memberikan gambaran nyata tentang bagaimana metode RAD dan konsep pemodelan UML dapat diterapkan dalam pengembangan sistem yang mendukung kegiatan operasional bisnis kecil secara efisien dan berbasis data.

5. Tinjauan studi kelima diambil dari penelitian berjudul “Penerapan Model *Rapid Application Development* dalam Pembuatan Sistem Informasi Akuntansi Penjualan dan Pembelian Berbasis Web” yang ditulis oleh Weiskhy Steven Dharmawan dan dipublikasikan dalam JUSTIAN: Jurnal Sistem Informasi Akuntansi, Vol. 6 No. 1, Maret 2025. Penelitian ini mengangkat permasalahan pencatatan transaksi penjualan dan pembelian pada PT. Maissa Muda Nusantara yang masih dilakukan secara manual menggunakan kertas dan Excel, sehingga proses penyusunan laporan tidak hanya lambat tetapi juga kurang sesuai dengan Standar Akuntansi Keuangan (SAK). Untuk menjawab permasalahan tersebut, peneliti membangun sistem informasi akuntansi berbasis web dengan menggunakan metode *Rapid Application Development (RAD)*, yang melibatkan tahapan pemodelan bisnis, pemodelan data, pemodelan proses,

pembuatan aplikasi, dan pengujian. Sistem ini dikembangkan dengan framework Laravel dan mendukung dua jenis pengguna: admin dan direktur. Fitur utama sistem meliputi pencatatan akun, pengelolaan data tanah dan rumah, transaksi pembelian dan penjualan, serta pelaporan keuangan seperti laba rugi dan perubahan modal. Penelitian ini sangat relevan bagi mahasiswa Sistem Informasi karena menunjukkan bagaimana pendekatan RAD dan pemodelan berbasis UML (seperti use case dan activity diagram) dapat diterapkan dalam pengembangan aplikasi akuntansi untuk perusahaan properti, serta pentingnya digitalisasi untuk meningkatkan efisiensi dan akurasi dalam pengelolaan data transaksi keuangan.

