

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Temuan dari penelitian terdahulu disajikan dalam bentuk tabel agar Anda dapat dengan mudah membandingkannya dengan penelitian yang diusulkan penulis. Berdasarkan penelitian-penelitian tersebut, dapat disimpulkan bahwa penggunaan teknologi informasi khususnya sistem yang memanfaatkan kode QR menjadikan pengelolaan data inventaris lebih baik dan lebih cepat. Sistem semacam ini umumnya bertujuan menggantikan tugas manual konvensional dengan metode berbasis komputer, sehingga mempermudah pencatatan data, pencarian barang, dan pemantauan inventaris. Namun, beberapa penelitian terdahulu masih memiliki keterbatasan; sistem yang dikembangkan biasanya hanya menangani fungsi dasar seperti pemantauan inventaris dan belum mencakup fitur yang lebih lengkap, seperti pelacakan aset secara rinci, pengelolaan peminjaman, serta pelaporan barang rusak. Oleh karena itu, penelitian ini mengusulkan pengembangan sistem pengelolaan aset berbasis kode QR. Sistem ini akan dilengkapi dengan fitur untuk pelacakan aset, pengelolaan peminjaman, dan pelaporan kerusakan. Pengembangan sistem akan menggunakan metode *Agile Development* untuk memastikan sistem bersifat fleksibel, dapat disesuaikan dengan kebutuhan, serta memenuhi persyaratan pengguna..

Tabel 2.1 Penelitian Terdahulu

No	Peneliti	Judul Penelitian	Metode/ Teknologi	Kelebihan / FituUtama	Perbedaan dengan Penelitian Ini
1.	Pranoto andSediyono.[1]	Perancangan Sistem Informasi Inventaris Barang Berbasis Web.	<i>Waterfall</i>	<i>Input</i> barang masuk dan barang keluar, Mempermudah untuk mengelola data barang secara online tanpa harus menggunakan buku secara manual	Penelitian sebelumnya hanya berfokus pada pengelolaan barang dan masih <i>Input</i> manual di dalam web, sedangkan penelitian ini sudah berbasis <i>QRCode</i> yang otoatis menampilkan data barang secara realtime
2.	Kharismatus Silpa.[2]	Siventasa(Sistem Informasi Inventaris Aset) Dengan <i>QR Code</i> Pada STIKes Muhammadiyah Tegal.	<i>Rapid Application Development(RAD).</i>	<i>QRCode</i> , laporan kerusakan barang, Memudahkan bagian administrasi dalam mengelola data inventaris asset. Mengurangi kehilangan dan kerusakan data inventaris asset	Sistem ini memiliki dua fitur utama yaitu :pengelolaan barang dan laporan kerusakan perbedaan dengan system penulis Adalah system yang penulis buat memiliki 3 fitur pelacakan barang, laporan kerusakan dan peminjaman barang
3.	Risky Cakra Mandala.[3]	Pengembangan Sistem Inventaris Barang Berbasis <i>QR Code</i> pada Badan	Black-box testing	Pemantauan barang dengan <i>QR Code</i> , Sistem ini mendukung kegiatan pengolahan dan	Sistem yang di buat di penelitian ini spesifik untuk pengelolaan barang berbasis <i>QR Code</i> berbeda dengan penelitian ini yang didalam satu

		Kepegawaian Daerah Provinsi Bengkulu		pemantauan data barang di BKD Provinsi Bengkulu secara cepat, baik, dan akurat. Kata- kunci.	sistem sudah mencakup pengelolaan, pelaporan barang dan peminjaman barang .
--	--	--	--	--	---

Berdasarkan penelitian terdahulu, penggunaan teknologi informasi khususnya sistem yang memanfaatkan kode QR terbukti membuat pengelolaan data inventaris menjadi lebih efektif dan efisien. Sistem semacam ini umumnya menggantikan metode manual konvensional dengan sistem berbasis komputer, sehingga mempermudah pencatatan, pencarian, dan pelacakan barang. Namun, sejumlah penelitian yang ada masih memiliki keterbatasan karena sistem yang dikembangkan biasanya hanya menangani tugas inventaris sederhana dan belum mencakup fitur yang lebih komprehensif, seperti pelacakan aset secara rinci, pengelolaan peminjaman, dan pelaporan kerusakan secara terintegrasi. Oleh karena itu, penelitian ini mengusulkan pengembangan sistem pengelolaan aset berbasis kode QR. Sistem ini dirancang untuk membantu pelacakan aset, pengelolaan peminjaman, serta pelaporan kerusakan. Untuk memastikan sistem bersifat fleksibel dan memenuhi kebutuhan pengguna, pengembangan sistem ini akan menggunakan metode *Agile Development*.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Sistem merupakan istilah yang berasal dari bahasa Latin *systema* dan bahasa Yunani *systema*, yang mengacu pada suatu kesatuan yang tersusun atas berbagai komponen atau elemen yang saling berhubungan dan bekerja sama untuk mencapai tujuan tertentu. Hubungan antarkomponen tersebut memungkinkan terjadinya aliran informasi, materi, maupun energi secara teratur sehingga proses dalam sistem dapat berjalan secara efektif. Dalam berbagai bidang ilmu, sistem sering dipahami sebagai sekumpulan entitas yang saling berinteraksi, bahkan dalam kondisi tertentu dapat direpresentasikan menggunakan model matematis untuk mempermudah analisis. Secara umum, sistem juga dapat diartikan sebagai kumpulan bagian yang saling berkaitan dan saling memengaruhi dalam suatu lingkungan tertentu. Setiap bagian memiliki fungsi masing-masing, namun seluruh bagian tersebut bekerja secara terpadu untuk menghasilkan tujuan yang sama. Contoh sederhana dari sebuah sistem dapat ditemukan pada organisasi, perusahaan, maupun negara, di mana setiap unsur memiliki peran yang saling mendukung sehingga keseluruhan sistem dapat berfungsi dengan baik. Sementara itu, informasi merupakan sekumpulan pesan atau data yang telah diproses sehingga memiliki makna dan dapat dipahami oleh penerimanya. Informasi dapat disampaikan dalam berbagai bentuk, seperti tulisan, simbol, tanda, maupun sinyal yang ditransmisikan

melalui media tertentu. Dalam konteks sistem informasi, informasi berperan sebagai hasil pengolahan data yang memberikan nilai tambah sehingga dapat digunakan sebagai dasar dalam pengambilan keputusan maupun penyelesaian suatu permasalahan. Informasi juga dapat dipahami sebagai pengetahuan yang diperoleh seseorang melalui proses pembelajaran, pengalaman, observasi, maupun instruksi. Dengan demikian, informasi bukan sekadar kumpulan data mentah, melainkan data yang telah diberi konteks sehingga memiliki arti dan manfaat bagi penggunaannya. Sebagai contoh, angka-angka yang terdapat dalam sebuah spreadsheet belum dapat disebut informasi apabila belum diolah. Setelah data tersebut disusun menjadi laporan, seperti laporan laba rugi atau neraca keuangan, data tersebut berubah menjadi informasi yang mampu memberikan gambaran mengenai kondisi keuangan suatu organisasi dan dapat dijadikan dasar dalam proses pengambilan keputusan.[4]

2.2.2 Inventaris Barang Aset

Inventaris barang merupakan proses pendataan dan pengelolaan seluruh barang yang dimiliki oleh individu, perusahaan, maupun instansi pemerintah. Kegiatan ini bertujuan untuk mengetahui jumlah, kondisi, lokasi, serta nilai aset secara akurat sehingga memudahkan proses pengawasan dan pengendalian. Aset sendiri adalah sumber daya yang memiliki nilai ekonomi serta masa manfaat yang relatif panjang, sehingga dapat digunakan untuk mendukung operasional dan

memberikan manfaat sesuai dengan kebutuhan organisasi. Contoh aset meliputi tanah, bangunan, gedung, jalan, mesin, peralatan, dan berbagai fasilitas lainnya. Seiring dengan meningkatnya kebutuhan akan sarana dan prasarana dalam menunjang aktivitas kerja, setiap organisasi perlu melaksanakan inventarisasi aset secara berkala, umumnya minimal satu kali dalam satu tahun, guna memastikan data aset tetap akurat, memantau kondisi aset, serta mendukung pengambilan keputusan dalam pengelolaannya.[5]

2.2.3 Quick Response Code (QR Code)

QR Code (*Quick Response Code*) merupakan jenis kode dua dimensi yang dirancang untuk menyimpan berbagai bentuk informasi, seperti teks, angka, tautan (*URL*), gambar, maupun data lainnya. Teknologi ini pertama kali dikembangkan oleh Denso Corporation di Jepang pada tahun 1994 untuk memenuhi kebutuhan industri otomotif dalam proses identifikasi komponen secara lebih cepat dan akurat. Seiring perkembangan teknologi, penggunaan QR Code tidak lagi terbatas pada sektor industri, tetapi telah diterapkan secara luas pada berbagai bidang karena kemampuannya dalam menyimpan informasi dengan kapasitas yang lebih besar dibandingkan kode batang (*barCode*) konvensional. Secara fungsional, QR Code memiliki dua mekanisme utama, yaitu proses pembuatan (*generate*) dan proses pembacaan (*scan*). Pada tahap *generate*, pengguna dapat memasukkan berbagai jenis informasi, seperti teks, alamat situs web, atau data lainnya yang

kemudian dikonversi menjadi sebuah gambar *QR Code*. Gambar tersebut dapat dibagikan atau dicetak sehingga mudah diakses oleh pengguna lain. Selanjutnya, pada tahap *scan*, perangkat yang dilengkapi kamera atau pemindai khusus akan membaca pola *QR Code* dan menerjemahkan informasi yang tersimpan di dalamnya secara otomatis. Kombinasi kedua proses tersebut menjadikan *QR Code* sebagai media pertukaran informasi yang cepat, praktis, dan efisien. Dibandingkan dengan *barCode* satu dimensi, *QR Code* memiliki kapasitas penyimpanan data yang jauh lebih besar. Selain mampu menyimpan angka dan huruf, *QR Code* juga dapat menampung data biner (*binary*) serta karakter Kanji, sehingga lebih fleksibel untuk berbagai kebutuhan. Keunggulan tersebut menyebabkan *QR Code* banyak dimanfaatkan dalam berbagai sektor, seperti sistem inventaris, pembayaran digital, autentikasi, pelacakan barang, hingga penyebaran informasi melalui internet. Salah satu penerapan *QR Code* yang banyak digunakan adalah sebagai media promosi dan pemasaran. Dengan memindai *QR Code* menggunakan perangkat seluler yang terhubung ke internet, pengguna dapat langsung mengakses informasi produk, katalog, ataupun halaman *Website* tanpa perlu mengetik alamat secara manual. Cara ini tidak hanya mempermudah penyampaian informasi kepada pelanggan, tetapi juga menjadi strategi promosi yang lebih hemat biaya serta mampu menjangkau masyarakat dalam cakupan yang lebih luas.[6]

2.2.4 Sistem Inventaris Berbasis QR Code

Sistem inventaris berbasis **QR Code** merupakan sistem pengelolaan aset yang menerapkan teknologi *QR Code* sebagai identitas unik pada setiap barang. Setiap aset diberikan kode QR yang berisi informasi tertentu sehingga dapat dikenali secara cepat dan akurat. Melalui proses pemindaian (*scan*) menggunakan perangkat yang mendukung, pengguna dapat memperoleh informasi lengkap mengenai aset, seperti identitas barang, kondisi, lokasi, serta status peminjaman. Selain itu, sistem juga menyediakan fasilitas untuk melaporkan kerusakan aset secara langsung, sehingga proses pencatatan, pemantauan, dan pengelolaan inventaris menjadi lebih efektif, efisien, dan terintegrasi.

2.2.5 Website dan Aplikasi Berbasis web

Website merupakan media digital yang menyajikan berbagai informasi dan dapat diakses melalui jaringan internet menggunakan perangkat yang terhubung dengan koneksi internet. Keberadaan *Website* memungkinkan pengguna memperoleh informasi maupun layanan secara cepat tanpa dibatasi oleh lokasi. Oleh karena itu, *Website* menjadi salah satu *platForm* yang banyak dimanfaatkan dalam pengembangan sistem informasi di berbagai bidang. Pengembangan sistem informasi merupakan serangkaian kegiatan yang dilakukan untuk merancang, membangun, dan mengimplementasikan sistem berbasis komputer guna membantu menyelesaikan permasalahan yang

dihadapi organisasi maupun mendukung pemanfaatan peluang yang tersedia. Sistem informasi berbasis *Website* memberikan berbagai keuntungan, seperti meningkatkan efisiensi pekerjaan, mempercepat proses pengolahan data, mempermudah penyampaian informasi, serta mendukung pengambilan keputusan yang lebih efektif. Agar sistem yang dihasilkan sesuai dengan kebutuhan pengguna dan tujuan organisasi, proses pengembangannya perlu mengikuti model atau metode pengembangan perangkat lunak yang dipilih berdasarkan karakteristik dan kebutuhan proyek.[7]

2.2.6 Basis Data

Basis data (*database*) merupakan sekumpulan data yang disusun secara sistematis dan terstruktur sehingga dapat disimpan, dikelola, serta diakses melalui sistem komputer. Data yang terdapat di dalam basis data umumnya dikelompokkan ke dalam beberapa tabel yang saling berhubungan melalui relasi tertentu. Dengan adanya struktur tersebut, proses penyimpanan, pencarian, pembaruan, dan pengolahan data dapat dilakukan secara lebih efektif dan efisien. Dalam suatu organisasi, basis data berperan sebagai pusat penyimpanan informasi sehingga seluruh data dapat dikelola dalam satu sistem yang terintegrasi. Keberadaan basis data memudahkan pengguna dalam memperoleh informasi secara cepat, mengurangi duplikasi data, serta meningkatkan efisiensi dalam proses pengelolaan informasi. Selain itu, basis data juga mendukung kegiatan analisis data yang dapat digunakan

sebagai dasar dalam proses pengambilan keputusan yang lebih tepat dan akurat. Penerapan basis data memberikan berbagai manfaat bagi organisasi maupun pengguna. Salah satu manfaat utamanya adalah meningkatkan efisiensi waktu dan biaya, karena proses penyimpanan, pencarian, serta pengolahan data dapat dilakukan secara otomatis dan lebih cepat dibandingkan metode manual. Kondisi ini juga membantu mempercepat penyajian informasi sehingga mendukung proses pengambilan keputusan secara lebih efektif. Manfaat lainnya adalah menjaga konsistensi dan keakuratan data. Melalui mekanisme pengelolaan yang terstruktur, basis data dapat meminimalkan terjadinya duplikasi maupun ketidaksesuaian informasi. Dengan demikian, data yang tersimpan memiliki tingkat keandalan yang lebih tinggi sehingga dapat digunakan sebagai sumber informasi yang akurat dalam mendukung aktivitas operasional maupun perencanaan organisasi.[8]

2.2.7 Agile Developmet

Metode *Agile Development* merupakan pendekatan dalam pengembangan perangkat lunak yang menekankan proses pengembangan secara bertahap dan iteratif. Pendekatan ini dirancang agar tim pengembang mampu merespons perubahan kebutuhan secara cepat selama proses pengembangan berlangsung. Dengan menerapkan *Agile Development*, setiap tahapan pengembangan dilakukan dalam siklus yang relatif singkat sehingga evaluasi dan penyempurnaan sistem dapat dilakukan secara berkelanjutan sesuai dengan masukan dari

pengguna maupun perubahan kebutuhan. *Agile Development* berfokus pada fleksibilitas, kolaborasi, serta kemampuan beradaptasi terhadap berbagai perubahan yang mungkin terjadi selama proyek berlangsung. Melalui pendekatan ini, pengembang tidak hanya berorientasi pada penyelesaian produk akhir, tetapi juga memastikan bahwa setiap peningkatan yang dilakukan mampu memberikan nilai tambah bagi pengguna. Oleh karena itu, metode *Agile* banyak digunakan dalam pengembangan sistem informasi modern karena mampu menghasilkan perangkat lunak yang lebih responsif, berkualitas, dan sesuai dengan kebutuhan pengguna.[9]

2.2.8 Framework ExpressJS

ExpressJS merupakan *Framework* pengembangan aplikasi web yang berjalan di atas platForm *Node.js* dan dirancang untuk mempermudah proses pembangunan aplikasi web maupun layanan berbasis *Application Programming Interface* (API). Dengan menyediakan berbagai fitur dasar yang diperlukan dalam pengembangan sisi server, *ExpressJS* memungkinkan pengembang membangun aplikasi yang cepat, andal, dan mudah dikembangkan sesuai kebutuhan. *Node.js* sendiri merupakan *runtime environment JavaScript* yang menggunakan arsitektur *event-driven* dan *non-blocking I/O*, sehingga mampu menangani banyak permintaan (*Request*) secara bersamaan dengan penggunaan sumber daya yang efisien. Karakteristik tersebut menjadikan *Node.js* sangat sesuai untuk

pengembangan aplikasi web yang membutuhkan performa tinggi serta mampu melayani banyak pengguna secara bersamaan. Sebagai *Framework backend*, *ExpressJS* dikenal memiliki desain yang ringan (*lightweight*) dan minimalis karena hanya menyediakan komponen inti yang dibutuhkan dalam pengembangan aplikasi. Pendekatan ini membantu mengurangi *overhead* sehingga aplikasi dapat berjalan lebih cepat dan efisien. Selain itu, *ExpressJS* bersifat *unopinionated*, yaitu tidak menentukan struktur proyek secara kaku, sehingga pengembang memiliki kebebasan dalam mengatur arsitektur aplikasi sesuai dengan kebutuhan dan standar yang digunakan. *ExpressJS* juga menyediakan mekanisme *middleware* yang memudahkan pengelolaan proses *Request* dan *response* secara terstruktur. Melalui *middleware*, berbagai fitur tambahan seperti autentikasi pengguna, pencatatan aktivitas (*logging*), validasi data, hingga pemrosesan data (*parsing*) dapat diimplementasikan dengan lebih mudah. Dalam pengembangan layanan berbasis REST API, *ExpressJS* memfasilitasi pengelolaan endpoint secara sederhana dan terorganisasi, sehingga proses komunikasi antara aplikasi *frontend* dan *backend* dapat berlangsung secara efektif dan efisien.[10]

2.2.9 *MYSQL*

MySQL merupakan salah satu sistem manajemen basis data relasional (*Relational Database Management System* atau RDBMS) yang banyak digunakan dalam pengembangan aplikasi, khususnya

aplikasi berbasis web. Popularitas *MySQL* didukung oleh kemampuannya dalam mengelola data secara stabil, cepat, dan andal sehingga sering dipilih sebagai media penyimpanan data pada berbagai jenis sistem informasi. Selain berfungsi sebagai server basis data, *MySQL* juga mendukung penggunaan bahasa SQL (*Structured Query Language*) yang digunakan untuk mengelola, memanipulasi, serta mengambil data dari basis data. Sebagai perangkat lunak *open source*, *MySQL* didistribusikan di bawah lisensi GNU *General Public License* (GPL), sehingga dapat digunakan dan dikembangkan secara bebas oleh pengguna. *MySQL* juga bersifat *multiplatform*, yang berarti dapat dijalankan pada berbagai sistem operasi, seperti Windows, Linux, maupun macOS. Kemudahan instalasi dan konfigurasi menjadikan *MySQL* sebagai salah satu basis data yang banyak digunakan baik untuk kebutuhan pembelajaran maupun pengembangan aplikasi berskala kecil hingga besar. *MySQL* memiliki berbagai keunggulan yang mendukung proses pengelolaan data. Beberapa di antaranya adalah kecepatan dalam memproses query, antarmuka yang mudah dipahami (*user friendly*), kemampuan menangani banyak pengguna secara bersamaan (*multi-user*), serta dukungan penuh terhadap standar SQL. Selain itu, *MySQL* menyediakan mekanisme keamanan yang baik melalui pengaturan hak akses pengguna sehingga data dapat dikelola secara lebih aman dan terkontrol. Dalam proses pengolahan data, *MySQL* menyediakan beragam operator dan fungsi yang dapat

dimanfaatkan untuk menjalankan perintah SQL, terutama pada operasi seperti *SELECT*, *INSERT*, *UPDATE*, dan *DELETE*, termasuk penggunaan klausa *WHERE* untuk melakukan penyaringan data sesuai kebutuhan. Dengan performa yang tinggi, kemampuan untuk dikembangkan sesuai skala aplikasi, serta tingkat keamanan yang baik, *MySQL* menjadi salah satu solusi basis data yang banyak diterapkan pada pengembangan *Website*, aplikasi mobile, maupun sistem informasi lainnya.[11].

2.2.10 NodeJS

Node.js merupakan lingkungan eksekusi (*runtime environment*) yang memungkinkan bahasa pemrograman *JavaScript* dijalankan di sisi server (*server-side*). Teknologi ini dikembangkan untuk mendukung pembuatan aplikasi berbasis web dengan memanfaatkan *JavaScript* sebagai bahasa pemrograman utama, sehingga pengembang dapat menggunakan satu bahasa yang sama baik pada sisi *frontend* maupun *backend*. Dengan demikian, *Node.js* menjadi alternatif yang efisien dalam pengembangan aplikasi modern selain bahasa pemrograman server seperti PHP, Ruby, maupun Perl. Salah satu keunggulan *Node.js* adalah sifatnya yang *cross-platForm*, sehingga dapat dijalankan pada berbagai sistem operasi, seperti Windows, Linux, dan macOS, tanpa memerlukan perubahan pada kode program. Kemampuan ini memudahkan proses pengembangan, pengujian, maupun implementasi aplikasi pada berbagai lingkungan

kerja. *Node.js* juga dilengkapi dengan modul HTTP bawaan yang memungkinkan pengembang membangun dan menjalankan server web secara langsung tanpa bergantung pada perangkat lunak server web seperti Apache atau Nginx. Selain itu, *Node.js* menerapkan arsitektur *event-driven* dan mekanisme *non-blocking I/O*, sehingga mampu menangani banyak permintaan (*Request*) secara bersamaan dengan penggunaan sumber daya yang lebih efisien. Karakteristik tersebut menjadikan *Node.js* sebagai salah satu platform yang banyak digunakan dalam pengembangan aplikasi web, layanan REST API, aplikasi *real-time*, maupun sistem yang membutuhkan performa tinggi.[12]

2.2.11 Framework *React*

React JS merupakan pustaka (*library*) *JavaScript* bersifat *open source* yang dikembangkan untuk membangun antarmuka pengguna (*User Interface* atau UI), khususnya pada aplikasi berbasis *Single Page Application* (SPA). *React* pertama kali dikembangkan oleh Jordan Walke dan kemudian dipelihara oleh Meta (sebelumnya Facebook) beserta komunitas pengembang di seluruh dunia. Teknologi ini banyak digunakan dalam pengembangan sisi *frontend* karena mampu menghasilkan antarmuka yang interaktif, responsif, dan mudah dikelola. Salah satu konsep utama yang diterapkan oleh *React JS* adalah penggunaan *reusable component*, yaitu komponen antarmuka yang dapat digunakan kembali pada berbagai bagian

aplikasi tanpa harus dibuat dari awal. Pendekatan berbasis komponen tersebut membuat proses pengembangan menjadi lebih efisien, mempermudah pemeliharaan kode, serta meningkatkan konsistensi tampilan aplikasi. Selain itu, *React* memungkinkan pengembang menyusun antarmuka secara modular sehingga setiap komponen dapat dikembangkan dan dikelola secara terpisah. *React JS* juga mendukung pembuatan tampilan yang responsif pada berbagai perangkat, baik desktop maupun perangkat seluler. Dengan memanfaatkan mekanisme *Virtual DOM*, *React* mampu memperbarui bagian antarmuka yang mengalami perubahan tanpa harus memuat ulang seluruh halaman, sehingga performa aplikasi menjadi lebih cepat dan pengalaman pengguna menjadi lebih baik. Keunggulan tersebut menjadikan *React* sebagai salah satu teknologi yang banyak digunakan dalam pengembangan aplikasi web modern. Popularitas *React JS* dapat dilihat dari banyaknya perusahaan teknologi yang memanfaatkannya dalam pengembangan produk digital, seperti Facebook, Instagram, WhatsApp, Netflix, Airbnb, Dropbox, dan berbagai aplikasi berbasis web lainnya. Dukungan komunitas yang besar, dokumentasi yang lengkap, serta ekosistem yang terus berkembang menjadikan *React JS* sebagai salah satu pilihan utama dalam membangun antarmuka aplikasi web yang modern, dinamis, dan mudah dikembangkan.[13]

2.2.12 UI/UX

User Interface Design (UI) merupakan salah satu aspek penting dalam pengembangan perangkat lunak karena berfungsi sebagai media interaksi antara pengguna dengan sistem. UI mencakup seluruh elemen visual yang digunakan pengguna untuk mengakses, memahami, dan menjalankan berbagai fungsi yang tersedia pada suatu aplikasi. Antarmuka yang dirancang dengan baik akan membantu pengguna berinteraksi dengan sistem secara lebih mudah, cepat, dan efisien sehingga tujuan penggunaan aplikasi dapat tercapai dengan optimal. Kualitas desain UI memiliki pengaruh yang besar terhadap tingkat kemudahan penggunaan (*usability*) suatu perangkat lunak. Antarmuka yang sederhana, konsisten, dan mudah dipahami akan meningkatkan kenyamanan pengguna dalam menjalankan berbagai fitur yang tersedia. Sebaliknya, desain yang kurang baik dapat menimbulkan kebingungan, meningkatkan kemungkinan terjadinya kesalahan, serta mengurangi minat pengguna untuk terus menggunakan aplikasi. Oleh karena itu, proses perancangan UI perlu memperhatikan aspek kemudahan penggunaan, efisiensi interaksi, konsistensi tampilan, dan aksesibilitas agar aplikasi mampu memberikan pengalaman penggunaan yang optimal. Selain UI, *User Experience* (UX) juga menjadi komponen yang tidak kalah penting dalam proses pengembangan perangkat lunak. UX mengacu pada keseluruhan pengalaman yang dirasakan pengguna selama

berinteraksi dengan suatu sistem, mulai dari kemudahan dalam memahami fitur, kenyamanan saat menggunakan aplikasi, hingga kepuasan yang diperoleh setelah menyelesaikan suatu tugas. Pengalaman tersebut dipengaruhi oleh berbagai faktor, seperti kemudahan navigasi, kecepatan sistem, desain visual, serta kemampuan aplikasi dalam memenuhi kebutuhan pengguna. UI dan UX merupakan dua konsep yang saling melengkapi dan tidak dapat dipisahkan dalam pengembangan aplikasi. Desain antarmuka yang menarik dan mudah digunakan akan memberikan pengalaman yang lebih positif kepada pengguna. Sebaliknya, pengalaman pengguna yang baik tidak hanya bergantung pada tampilan visual, tetapi juga pada bagaimana sistem mampu bekerja secara efektif, efisien, dan sesuai dengan harapan pengguna. Oleh karena itu, penerapan prinsip UI dan UX secara seimbang menjadi faktor penting dalam menghasilkan perangkat lunak yang tidak hanya memiliki tampilan yang menarik, tetapi juga mudah digunakan, memberikan kenyamanan, serta meningkatkan kepuasan pengguna.[14]

2.2.13 Arsitecture Enterprise

Arsitektur Enterprise (Enterprise Architecture) merupakan suatu kerangka yang menggambarkan struktur organisasi secara menyeluruh, meliputi proses bisnis, tata kelola, prinsip organisasi, serta teknologi yang digunakan untuk mendukung pencapaian tujuan organisasi. Konsep ini berfungsi sebagai pedoman dalam mengelola

perubahan, menyelaraskan strategi bisnis dengan teknologi informasi, serta menciptakan standar operasional yang mendukung efektivitas proses bisnis. Dalam penerapannya, *Arsitektur Enterprise* menjelaskan hubungan antara berbagai komponen yang terdapat di dalam organisasi, termasuk keterkaitannya dengan lingkungan eksternal. Melalui pendekatan tersebut, organisasi dapat memahami bagaimana setiap komponen saling berinteraksi dan berkontribusi terhadap pencapaian tujuan yang telah ditetapkan. Selain itu, *Arsitektur Enterprise* juga menjadi dasar dalam proses analisis, perancangan, pengembangan, serta evaluasi sistem sehingga organisasi mampu beradaptasi dengan perubahan kebutuhan bisnis maupun perkembangan teknologi. Secara umum, *Arsitektur Enterprise* memberikan gambaran yang komprehensif mengenai berbagai aspek organisasi, seperti visi dan tujuan, struktur organisasi, peran setiap unit kerja, proses bisnis, pengelolaan informasi, serta pemanfaatan teknologi informasi. Pendekatan ini membantu organisasi memastikan bahwa seluruh sumber daya yang dimiliki dapat dikelola secara terintegrasi dan selaras dengan strategi bisnis. Melalui *Arsitektur Enterprise*, organisasi dapat mengidentifikasi komponen-komponen utama yang mendukung operasional, seperti sumber daya manusia, proses bisnis, infrastruktur teknologi informasi, data, serta sumber daya lainnya. Seluruh komponen tersebut dirancang agar dapat bekerja secara terpadu sehingga mampu

meningkatkan efisiensi operasional, mendukung pengambilan keputusan, serta membantu organisasi mencapai tujuan bisnis secara lebih efektif.[15]

2.2.14 Balsamiq Wireframe

Balsamiq Mockups merupakan perangkat lunak yang digunakan untuk merancang antarmuka pengguna (*User Interface*) pada tahap awal pengembangan sistem. Aplikasi ini dirancang untuk membantu pengembang menyusun gambaran awal tampilan aplikasi secara cepat melalui pendekatan visual, sehingga rancangan sistem dapat dipahami dengan lebih mudah sebelum memasuki proses implementasi. Dengan menggunakan *Balsamiq Mockups*, pengembang dapat menyusun konsep antarmuka secara sistematis serta melakukan validasi terhadap kebutuhan pengguna sejak tahap perancangan. *Balsamiq Mockups* menerapkan konsep *low-fidelity wireframe*, yaitu rancangan antarmuka yang berfokus pada penyusunan struktur, tata letak, serta alur navigasi aplikasi tanpa menampilkan detail visual secara lengkap. Pendekatan ini memudahkan proses diskusi, evaluasi, dan penyempurnaan desain karena perhatian lebih difokuskan pada fungsi dan pengalaman penggunaan dibandingkan aspek estetika. Dengan demikian, perubahan rancangan dapat dilakukan lebih cepat sebelum tahap pengkodean (*coding*) dimulai. Dalam pengembangan aplikasi berbasis web maupun perangkat bergerak (*mobile*), *Balsamiq Mockups* banyak dimanfaatkan karena menyediakan berbagai

komponen antarmuka yang siap digunakan, seperti tombol (*button*), Formulir (*Form*), menu navigasi, tabel, ikon, hingga berbagai jenis tata letak halaman (*layout*). Ketersediaan komponen tersebut membantu mempercepat proses pembuatan prototipe serta mempermudah komunikasi antara pengembang, analis sistem, dan pengguna dalam memvisualisasikan rancangan aplikasi yang akan dibangun.[16]

2.2.15 Black-Box Testing

Black Box Testing merupakan salah satu metode pengujian perangkat lunak yang berorientasi pada pengujian fungsional sistem tanpa memperhatikan struktur maupun kode program yang digunakan. Pengujian dilakukan berdasarkan kebutuhan dan spesifikasi sistem untuk memastikan bahwa setiap fitur mampu memberikan keluaran yang sesuai dengan hasil yang diharapkan. Dengan pendekatan ini, penguji hanya berinteraksi melalui masukan (*Input*) dan keluaran (*Output*) sistem tanpa mengetahui mekanisme pemrosesan yang terjadi di dalam aplikasi. Sebagai metode pengujian fungsional, *Black Box Testing* bertujuan untuk memverifikasi bahwa seluruh fungsi pada perangkat lunak telah berjalan sesuai dengan kebutuhan pengguna. Setiap skenario pengujian disusun berdasarkan spesifikasi yang telah ditetapkan, kemudian hasil yang diperoleh dibandingkan dengan hasil yang diharapkan. Apabila keluaran yang dihasilkan sesuai dengan spesifikasi, maka fungsi tersebut dinyatakan berhasil

memenuhi kebutuhan sistem. Berbagai penelitian menunjukkan bahwa *Black Box Testing* merupakan metode yang efektif untuk mengidentifikasi kesalahan fungsional serta memastikan kualitas perangkat lunak sebelum digunakan oleh pengguna. Melalui penyusunan *test case* yang sistematis, pengembang dapat mengevaluasi perilaku sistem pada berbagai kondisi sehingga dapat diketahui apakah seluruh fitur telah bekerja sesuai dengan rancangan yang telah dibuat. Selain itu, metode ini juga membantu memverifikasi kesesuaian implementasi sistem terhadap kebutuhan fungsional yang telah ditentukan pada tahap analisis. Karena dilakukan dari sudut pandang pengguna, *Black Box Testing* mampu memberikan gambaran mengenai kualitas sistem berdasarkan pengalaman penggunaan (*user perspective*). Dalam penerapannya, metode ini dapat dipadukan dengan berbagai teknik pengujian, salah satunya *Equivalence Partitioning*, yaitu teknik yang mengelompokkan data masukan ke dalam beberapa kelas yang dianggap memiliki karakteristik serupa. Pendekatan tersebut membantu mengurangi jumlah skenario pengujian tanpa mengurangi efektivitas proses validasi. Dengan demikian, *Black Box Testing* menjadi salah satu metode yang penting dalam proses pengembangan perangkat lunak karena mampu memastikan bahwa sistem memiliki fungsi yang benar, andal, serta siap digunakan sesuai dengan kebutuhan pengguna.[17]

2.2.16 UML(*Unified Modeling Language*)

Unified Modeling Language (UML) merupakan bahasa pemodelan standar yang digunakan untuk memvisualisasikan, menganalisis, dan merancang sistem perangkat lunak, terutama yang menerapkan konsep *object-oriented programming* (OOP). UML membantu pengembang menggambarkan struktur, perilaku, serta interaksi antar komponen sistem dalam bentuk diagram sehingga rancangan sistem dapat dipahami dengan lebih jelas sebelum memasuki tahap implementasi. Dalam proses pengembangan perangkat lunak, UML berperan sebagai sarana untuk mendokumentasikan kebutuhan sistem, merancang arsitektur, serta menjelaskan hubungan antar komponen yang terdapat di dalam aplikasi. Melalui pemodelan visual, pengembang dapat memahami proses bisnis, aliran data, serta interaksi antara pengguna dan sistem secara lebih sistematis. Selain itu, UML juga mempermudah komunikasi antara analis sistem, pengembang, dan pemangku kepentingan (*stakeholder*) karena rancangan sistem dapat disajikan dalam bentuk diagram yang mudah dipahami. UML menyediakan berbagai jenis diagram yang digunakan sesuai dengan kebutuhan analisis dan perancangan sistem. Beberapa diagram yang paling umum digunakan meliputi *Use Case Diagram*, yang berfungsi menggambarkan interaksi antara aktor dan sistem; *Activity Diagram*, yang digunakan untuk menunjukkan alur aktivitas atau proses bisnis;

Class Diagram, yang menjelaskan struktur kelas beserta hubungan antar kelas dalam sistem; serta *Sequence Diagram*, yang menggambarkan urutan komunikasi atau pertukaran pesan antar objek ketika suatu proses dijalankan. Penggunaan diagram-diagram tersebut membantu pengembang memperoleh gambaran yang lebih jelas mengenai cara kerja sistem sebelum proses pengkodean dilakukan. Dengan demikian, UML menjadi salah satu alat bantu yang penting dalam pengembangan perangkat lunak karena mampu menyajikan rancangan sistem secara terstruktur, meningkatkan kualitas dokumentasi, serta mengurangi kemungkinan terjadinya kesalahan pada tahap implementasi. Selain itu, pemodelan menggunakan UML juga mendukung proses pengembangan yang lebih terorganisasi sehingga sistem yang dihasilkan lebih mudah dipelihara dan dikembangkan di masa mendatang.[18]

2.2.17 *Application Programming Interface API*

Application Programming Interface (API) merupakan sekumpulan aturan, protokol, dan antarmuka yang memungkinkan suatu aplikasi berkomunikasi serta bertukar data dengan aplikasi atau layanan lainnya. API memberikan akses kepada pengembang untuk memanfaatkan fungsi atau layanan yang telah tersedia tanpa harus membangun kembali fitur tersebut dari awal, sehingga proses pengembangan aplikasi menjadi lebih efisien dan terstruktur. Dalam pengembangan aplikasi berbasis web, API umumnya bekerja

menggunakan protokol *Hypertext Transfer Protocol* (HTTP) sebagai media komunikasi antara *client* dan *server*. Ketika aplikasi mengirimkan permintaan (*Request*) ke suatu API, server akan memproses permintaan tersebut dan mengembalikan hasilnya dalam *Format* pertukaran data, seperti *JavaScript Object Notation* (JSON) atau *Extensible Markup Language* (XML). Mekanisme ini memungkinkan berbagai aplikasi yang dibangun menggunakan teknologi berbeda tetap dapat saling berkomunikasi dan bertukar informasi dengan baik. Salah satu tujuan utama penggunaan API adalah mempermudah integrasi antar sistem sehingga data dapat dibagikan secara otomatis tanpa memerlukan proses yang rumit. Selain itu, API juga membantu mempercepat proses pengembangan perangkat lunak karena pengembang dapat memanfaatkan layanan yang telah tersedia, seperti autentikasi, pembayaran digital, maupun layanan pihak ketiga lainnya. Dengan pendekatan tersebut, waktu pengembangan menjadi lebih singkat serta kualitas aplikasi dapat ditingkatkan melalui penggunaan layanan yang telah teruji. Pada tingkat sistem operasi maupun aplikasi, API berperan sebagai penghubung yang memungkinkan berbagai komponen perangkat lunak saling berinteraksi sesuai dengan aturan dan protokol yang telah ditetapkan. Melalui mekanisme tersebut, setiap aplikasi dapat mengakses fungsi tertentu secara aman, terstandarisasi, dan efisien

tanpa harus mengetahui secara langsung implementasi internal dari sistem yang digunakan.[19]

2.2.18 SPA(*Single Page Application*)

Single Page Application (SPA) merupakan arsitektur aplikasi web yang bekerja dengan memuat satu halaman utama, kemudian memperbarui isi halaman secara dinamis tanpa melakukan proses *reload* atau *refresh* keseluruhan halaman. Pada konsep ini, sebagian besar interaksi pengguna dikelola di sisi *client*, sehingga perpindahan antar tampilan dapat berlangsung dengan lebih cepat dan memberikan pengalaman penggunaan yang lebih responsif dibandingkan aplikasi web konvensional. Dalam implementasinya, SPA memanfaatkan teknologi seperti AJAX, Fetch API, atau Web API lainnya untuk melakukan pertukaran data dengan server secara *asynchronous*. Dengan mekanisme tersebut, aplikasi hanya mengambil data yang diperlukan tanpa harus memuat ulang seluruh halaman. Pendekatan ini mampu meningkatkan performa aplikasi, mempercepat waktu respons, serta mengurangi jeda yang dirasakan pengguna ketika berpindah antar fitur atau halaman. Dari sisi pengguna (*end user*), aplikasi berbasis SPA menawarkan pengalaman yang lebih nyaman karena navigasi berlangsung secara cepat dan mulus. Perubahan tampilan maupun penyaringan (*filtering*) data dapat dilakukan secara langsung tanpa mengganggu aktivitas pengguna. Hal tersebut menjadikan SPA banyak digunakan pada aplikasi web modern yang

membutuhkan interaksi tinggi dan pengalaman pengguna yang lebih baik. Selain meningkatkan kenyamanan pengguna, penerapan konsep SPA juga membantu mengurangi beban kerja server. Setelah halaman utama berhasil dimuat, komunikasi antara *client* dan *server* hanya dilakukan untuk mengambil atau mengirimkan data yang dibutuhkan, bukan keseluruhan halaman web. Dengan demikian, penggunaan bandwidth menjadi lebih efisien, konsumsi sumber daya server dapat diminimalkan, dan performa aplikasi secara keseluruhan menjadi lebih optimal dibandingkan aplikasi web tradisional (*multi-page application* atau *non-SPA*) yang harus memuat ulang halaman setiap kali pengguna berpindah menu.[20]

2.2.19 Postman

Postman merupakan perangkat lunak yang digunakan untuk melakukan pengujian dan pengembangan *Application Programming Interface* (API), khususnya API berbasis REST. Aplikasi ini menyediakan antarmuka grafis (*Graphical User Interface/GUI*) yang memudahkan pengembang dalam mengirim berbagai jenis permintaan (*Request*) HTTP, seperti GET, POST, PUT, PATCH, dan *DELETE*, tanpa harus menulis kode secara langsung. Dengan demikian, proses pengujian fungsi API dapat dilakukan dengan lebih cepat dan efisien. Pada awal pengembangannya, *Postman* hadir sebagai ekstensi (*plugin*) pada peramban Google Chrome. Seiring perkembangannya, *Postman* kini tersedia sebagai aplikasi mandiri

(*standalone application*) yang dapat dijalankan pada berbagai sistem operasi, seperti Windows, macOS, dan Linux. Selain digunakan untuk menguji endpoint API, *Postman* juga menyediakan berbagai fitur pendukung, seperti pembuatan dokumentasi API, pengelolaan *collection*, pengujian otomatis (*API testing*), pemantauan performa (*API monitoring*), serta kolaborasi tim secara *real-time*. Dalam proses pengembangan perangkat lunak, *Postman* banyak dimanfaatkan untuk memverifikasi respons API dengan mengirimkan data masukan dan menganalisis hasil yang dikembalikan oleh server. Melalui aplikasi ini, pengembang dapat memastikan bahwa setiap endpoint bekerja sesuai dengan spesifikasi yang telah dirancang, baik dari sisi status respons, struktur data, maupun fungsionalitas sistem. Oleh karena itu, *Postman* menjadi salah satu alat yang banyak digunakan dalam pengembangan dan pengujian layanan berbasis REST API karena mampu meningkatkan efisiensi proses pengujian sekaligus mempermudah proses *debugging*. [21]