

BAB 2

TINJAUAN PUSTAKA

2.1 Landasan Teori

Bagian ini berisikan kumpulan teori dan konsep penting yang digunakan dalam mengerjakan proyek akhir ini. Penjelasan disini berfungsi untuk memahami cara kerja sistem, teknologi yang digunakan, serta metode yang diterapkan.

2.1.1 Sistem Informasi

Sistem informasi merupakan kombinasi dari komponen teknologi, data, prosedur, dan pengguna yang bekerja secara bersama untuk menghasilkan, mengolah, menyimpan, dan mendistribusikan informasi yang diperlukan untuk mendukung pengambilan keputusan, koordinasi, kontrol, analisis, dan visualisasi dalam organisasi atau perusahaan tertentu dalam mencapai tujuannya.



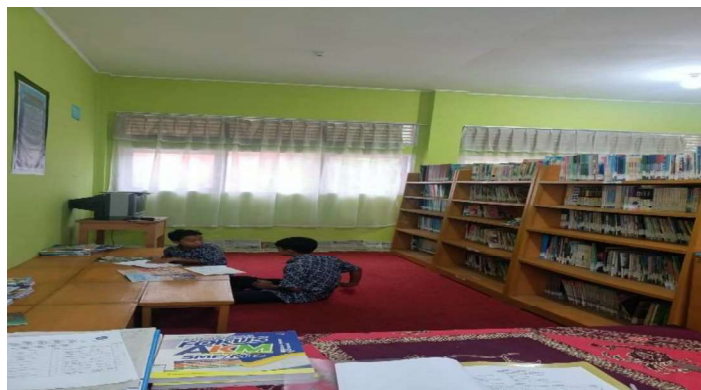
Gambar 2.1 Sistem Informasi

Selain itu sistem informasi menurut (Novi, 2021) juga merupakan kombinasi antara aktivitas manusia dan teknologi seperti pada Gambar 2.1 dimana sistem informasi mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi. . Sistem informasi juga berfungsi untuk menyediakan laporan-laporan yang dibutuhkan oleh pihak-pihak yang berkepentingan pada organisasi tersebut.

2.1.2 Perpustakaan

Perpustakaan adalah institusi yang menyediakan akses terhadap koleksi buku, jurnal, dan bahan informasi lainnya untuk mendukung kebutuhan akademik dan

riset. Dalam era digital, perpustakaan modern telah bertransformasi menjadi pusat informasi yang tidak hanya mengandalkan koleksi fisik tetapi juga menyediakan layanan berbasis teknologi. (Mabruri & Triyanto, 2022) menyatakan bahwa perpustakaan berbasis media digital kini menjadi kebutuhan utama dalam memberikan akses informasi yang lebih luas kepada pemustaka. Salah satu bentuk transformasi ini adalah pemanfaatan media sosial dalam menyebarkan informasi dan meningkatkan interaksi dengan pengguna. Selain itu, (Salsabila dkk., 2024) menekankan bahwa perpustakaan digital memungkinkan akses terhadap sumber daya informasi tanpa batasan geografis dan waktu, sehingga mendukung peningkatan produktivitas akademik. Konsep ini juga diterapkan dalam Perpustakaan UII, di mana sistem digital dikembangkan untuk mengelola koleksi dan memfasilitasi pencarian informasi secara lebih efektif (Kusmayadi, 2022).

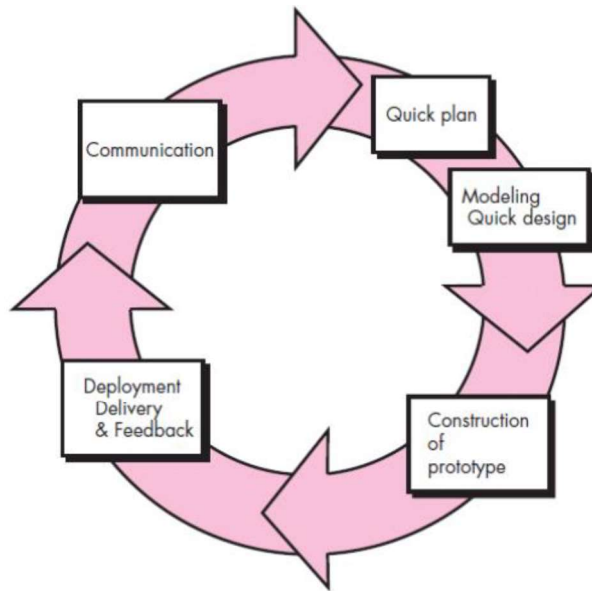


Gambar 2.2 Perpustakaan SMPN 4 Tilatang Kamang

Sejalan dengan konsep perpustakaan digital, Gambar 2.2 menggambarkan kondisi perpustakaan SMPN 4 Tilatang Kamang yang masih menggunakan sistem pengelolaan konvensional, dengan pencatatan manual dalam peminjaman dan pengembalian buku. Hal ini menunjukkan adanya tantangan dalam akses informasi, efisiensi layanan, serta pelacakan status buku. (Mabruri & Triyanto, 2022) serta (Salsabila dkk., 2024) menyarankan bahwa transformasi perpustakaan digital dapat menjadi solusi strategis untuk meningkatkan efisiensi dan fleksibilitas layanan. Implementasi sistem informasi perpustakaan berbasis digital, seperti yang diterapkan di Perpustakaan UII (Kusmayadi, 2022), dapat membantu SMPN 4 Tilatang Kamang dalam mengoptimalkan manajemen koleksi, pencarian buku, serta meningkatkan aksesibilitas informasi bagi siswa dan guru.

2.1.3 Metode Prototype

Menurut (Huda & Kasino, 2022) metode prototype merupakan salah satu metode baru pada pengembangan sebuah sistem yang dimana tahap dan alur yang dipakai meliputi mengembangkan prototype dan mengidentifikasi pengguna yang dapat dilihat pada gambar 2.3.



Gambar 2.3 Tahapan Metode *Prototype*

Metode ini memiliki 5 tahapan dengan peran yang berbeda beda selama proses perancangan aplikasi, berikut adalah tahapan-tahapan pada metode prototype :

1) Communication

Komunikasi merupakan tahap penting dalam seluruh proses pengembangan aplikasi. Selama proses pengujian sistem informasi, beberapa tahap yang dilakukan meliputi perencanaan pengujian, desain kasus uji, eksekusi pengujian, dan analisis hasil pengujian. Setiap tahap ini berperan dalam memastikan bahwa sistem berfungsi dengan baik dan bebas dari masalah. Selain itu, pengujian sistem informasi melibatkan kerja sama antara pengembang dan pengguna dalam mengidentifikasi kebutuhan dan ekspektasi mereka. Penting juga untuk memiliki lingkungan pengujian yang representatif dan mirip dengan lingkungan produksi guna memastikan bahwa hasil pengujian mencerminkan kondisi

yang sebenarnya.

2) Quick Plan

Tahapan ini bertujuan untuk mengidentifikasi kebutuhan sistem yang akan dikembangkan, mulai dari struktur data yang diperlukan, alur proses, hingga input dan outputnya. Pengumpulan kebutuhan pengguna merupakan langkah yang sangat penting dalam pengembangan produk. Proses ini melibatkan identifikasi dan pemahaman mendalam terhadap kebutuhan, preferensi, dan masalah yang dihadapi oleh pengguna. Dengan melakukan pengumpulan kebutuhan yang efektif dan komprehensif, tim pengembang dapat menciptakan solusi yang lebih relevan dan memuaskan. Oleh karena itu, sangat penting bagi tim pengembang untuk melakukan riset yang mendalam dan berinteraksi langsung dengan pengguna guna memastikan bahwa produk yang dikembangkan benar-benar memenuhi kebutuhan client.

3) Modeling & Quick Design

Pada tahap ini, desain awal sistem dibuat dengan menggunakan notasi UML dan pemodelan berbasis objek, seperti Use Case Diagram, Activity Diagram, Sequence Diagram, dan Class Diagram. Proses desain aplikasi sangat penting dalam pengembangan perangkat lunak karena bertujuan untuk memahami kebutuhan pengguna secara mendalam. Selain itu, tim desain juga merancang antarmuka yang intuitif dan menarik agar pengguna dapat dengan mudah berinteraksi dengan sistem. Setelah merancang antarmuka, prototipe aplikasi dibuat untuk diuji oleh pengguna guna memvalidasi desain dan mengumpulkan umpan balik yang berharga. Dengan menggabungkan kebutuhan pengguna dan hasil pengujian, tim desain dapat melakukan perbaikan dan penyempurnaan yang diperlukan guna menciptakan pengalaman pengguna yang baik.

4) Construction of Prototype

Pada tahap ini, proses pengkodean aplikasi dilakukan oleh pengembang dengan menggunakan bahasa pemrograman yang sesuai. Pembuatan kode aplikasi tidak hanya harus berfungsi dengan baik, tetapi juga harus efisien, aman, dan memiliki kualitas yang tinggi. Pengembang harus memastikan

bahwa kode yang mereka tulis mudah dipahami dan dipelihara oleh tim di masa depan. Selain itu, pengujian aplikasi juga merupakan bagian penting dalam proses ini, sehingga kesalahan atau bug dapat diidentifikasi dan diperbaiki sebelum aplikasi diperkenalkan kepada pengguna akhir.

5) Deployment, Delivery & Feedback

Tahapan ini berfokus pada pengujian sistem untuk memastikan kehandalan dan kualitasnya. Berbagai skenario dan kasus uji dijalankan untuk mengidentifikasi kesalahan dan cacat yang mungkin terjadi. Dengan menemukan dan memperbaiki masalah sebelum sistem diimplementasikan, risiko kegagalan sistem dapat dikurangi secara signifikan. Metode pengujian yang tepat sangat penting, seperti pengujian fungsional, pengujian performa, dan pengujian keamanan. Pengujian ini juga memberikan manfaat bagi pengguna, karena memastikan bahwa sistem dapat bekerja dengan baik dan aman. Selain itu, pengujian membantu mengidentifikasi area yang perlu ditingkatkan guna meningkatkan pengalaman pengguna.

2.1.4 *React*

React adalah sebuah pustaka JavaScript yang digunakan untuk membangun antarmuka pengguna, terutama untuk aplikasi satu halaman. React memungkinkan pengembang untuk membuat komponen UI yang dapat digunakan kembali dengan cara yang efisien.

Dalam pengembangan aplikasi, seperti admin perpustakaan, React mempermudah pengelolaan antarmuka dengan membagi aplikasi menjadi komponen-komponen kecil, seperti komponen daftar buku, formulir pencarian, dan pengelolaan pengguna. React juga mendukung integrasi dengan pustaka lain, seperti React Router untuk navigasi antar halaman, dan Axios untuk komunikasi dengan backend. Dengan pendekatan berbasis komponen dan state management, React memungkinkan pengembang untuk membangun aplikasi yang dinamis, interaktif, dan mudah diperluas.

2.1.5 *Node JS*

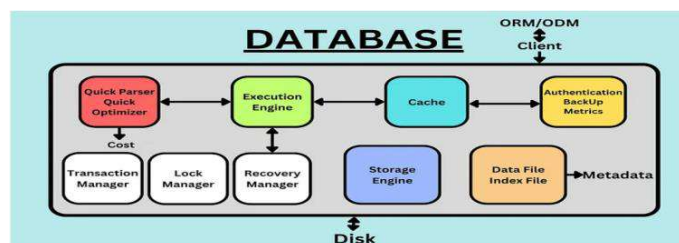
Node.js adalah lingkungan runtime JavaScript yang bersifat open source dan

dapat berjalan di berbagai platform seperti Windows, Linux, Unix, dan macOS. Node.js dibangun di atas mesin JavaScript V8 milik Chrome, memungkinkan pengembang untuk menjalankan kode JavaScript di sisi server.

Dalam konteks pembuatan admin perpustakaan, Node.js dapat digunakan untuk membangun backend yang mengelola fungsi-fungsi seperti pencatatan buku, manajemen pengguna, dan pelacakan peminjaman. Contohnya, Express.js—sebuah framework Node.js—dapat digunakan untuk membangun API RESTful yang memungkinkan komunikasi antara frontend dan database. Database seperti MySQL atau MongoDB dapat diintegrasikan untuk menyimpan data buku dan pengguna. Dengan menggunakan arsitektur ini, admin perpustakaan dapat merespons permintaan pengguna dengan cepat, misalnya, menampilkan daftar buku, menambahkan data buku baru, atau memperbarui status peminjaman.

2.1.6 Database

Database merupakan kumpulan data yang disusun dan disimpan oleh sistem, sehingga memungkinkan data akan dikelola dengan baik. Database berfungsi agar data yang disimpan dapat bertahan lama serta bisa diakses secara akurat dari mana saja. Umumnya tujuan database adalah untuk memperlancar pengelolaan data, menyimpan file secara permanen, serta mempermudah sistem dalam berbagai keperluan, baik di perusahaan ataupun organisasi lainnya (Pulungan dkk., 2023).



Gambar 2.4 Alur Database

Database bekerja seperti yang di gambarkan pada Gambar 2.4 menunjukkan bagaimana database mengelola, memproses, dan menyimpan data secara efisien. Proses dimulai dari ORM/ODM Client, yang mengirimkan permintaan ke Quick Parser dan Quick Optimizer untuk dianalisis dan dioptimalkan sebelum dieksekusi oleh Execution Engine. Data yang sedang diproses dapat disimpan sementara di Cache untuk mempercepat akses, sementara Transaction Manager, Lock Manager, dan Recovery Manager menjaga integritas serta pemulihan data. Hasil eksekusi

kemudian dikelola oleh Storage Engine, yang menyimpan data secara permanen dalam Data File dan Index File. Selain itu, sistem dilengkapi dengan fitur Authentication, Backup, dan Metrics untuk memastikan keamanan serta pemantauan kinerja database, sehingga operasional tetap optimal dan dapat diakses dengan cepat sesuai kebutuhan pengguna.

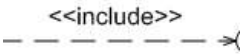
2.1.7 Use Case Diagram

Use case diagram memodelkan perilaku sistem yang akan dikembangkan. Use case menjelaskan interaksi antara satu atau lebih aktor dengan sistem informasi. Diagram ini digunakan untuk mengidentifikasi pengguna yang memiliki akses terhadap fungsi-fungsi tertentu.

Use case diagram menggambarkan fungsionalitas (Santoso dkk., 2024) yang diharapkan dari sebuah sistem. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan system. Use case diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem. Adapun notasi use case yang digunakan dalam penelitian ini dapat dilihat pada tabel 2.1.

Tabel 2.1 Notasi Pada Use Case Diagram

Notasi	Keterangan
Use Case	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .
Aktor  Nama Actor	Actor, orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.
Asosiasi 	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
Extend 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walaupun tanpa use case tambahan itu, mirip dengan prinsip inheritance

Notasi	Keterangan
	pada pemrograman berorientasi objek, biasanya use case tambahan memiliki nama depan yang sama dengan use case yang di tambahkan, arah panah menunjuk pada use case yang dituju.
Include 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini. Ada dua sudut pandang yang cukup besar mengenai include di use case, include berarti use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan.

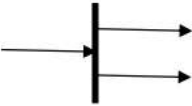
Tabel diatas merupakan notasi pada use case diagram, yang dimana secara keseluruhan menggambarkan penggunaan simbol pada use case diagram.

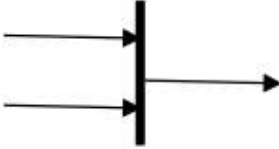
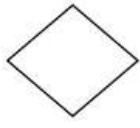
2.1.8 Activity Diagram

Menurut (Widyastuti, 2022), Activity Diagram menunjukkan bagaimana fungsionalitas suatu sistem bekerja, sekaligus menggambarkan interaksi antara aktor dengan sistem tersebut. Aktor di sini merupakan entitas yang dapat berupa manusia atau sistem lain yang memiliki peran dalam menjalankan aktivitas pada sistem.

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Adapun notasi activity diagram yang digunakan dalam penelitian ini dapat dilihat pada tabel 2.2.

Tabel 2.2 Notasi Pada Activity Diagram

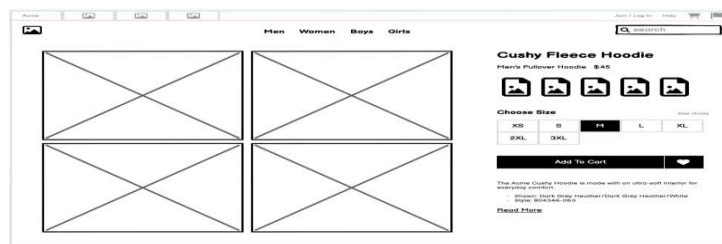
Notasi	Keterangan
	<i>Start point</i> , merupakan titik awal suatu activity
	<i>End point</i> , merupakan titik akhir dari activity
	<i>Activity</i>
	<i>Fork</i> (percabangan), memisahkan beberapa aliran yang melalui activity menjadi beberapa aliran

	Join (penggabungan), melakukan sinkronisasi terhadap beberapa aktivitas kembali menjadi satu aliran
	<i>Decision (branch)</i> , titik control yang memiliki keluaran yang berbeda berdasarkan ekspresi boolean. Setiap titik keputusan memiliki satu masukan dan banyak keluaran
Swirlane	Mengelompokkan activity berdasarkan actor. Diberi label dengan nama object yang bertanggung jawab untuk menjalankan activity atau action

Tabel diatas merupakan notasi pada activity diagram, yang dimana secara keseluruhan menggambarkan penggunaan notasi pada activity diagram.

2.1.9 Wireframe

Wireframe merupakan kerangka desain awal untuk menyusun elemen halaman aplikasi sebelum desain utama seperti pada Gambar 2.5, biasanya dibuat dengan alat seperti Figma. Terdiri dari dua jenis yaitu low fidelity yang berfokus pada struktur dasar tanpa detail seperti warna atau teks dan high fidelity, yang lebih rinci dan mendekati desain akhir (Rayhaan Yusri dkk., 2024), hal ini diperjelas pada gambar 2.5.



Gambar 2.5 Contoh Wireframe

Wireframe terbagi menjadi dua jenis, yaitu low fidelity dan high fidelity. Low fidelity wireframe biasanya berbentuk sketsa sederhana yang hanya menampilkan struktur dasar tanpa elemen visual atau teks yang mendetail. Jenis ini digunakan pada tahap awal untuk mendiskusikan ide atau konsep. Sebaliknya, high fidelity wireframe memiliki detail yang lebih rinci dan mendekati desain akhir, termasuk penggunaan warna, tipografi, dan elemen visual lainnya. High fidelity wireframe sering kali dibuat dengan alat desain digital seperti Figma untuk menghasilkan

representasi yang lebih realistis dari tampilan akhir aplikasi atau website.

2.1.10 ISO 25010

ISO 25010 merupakan bagian dari standar Software Product Quality Requirements and Evaluation (SQuaRE) yang dikembangkan oleh International Organization for Standardization (ISO) untuk mengukur dan mengevaluasi kualitas perangkat lunak dan layanan teknologi informasi. ISO 25010 berfokus pada model kualitas perangkat lunak dengan membagi kualitas menjadi dua aspek utama, yaitu *quality in use* (kualitas dalam penggunaan) dan *product quality* (kualitas produk).

Model ini mencakup delapan karakteristik utama, seperti fungsionalitas, keandalan, efisiensi kinerja, kompatibilitas, kegunaan, keamanan, maintainability, dan portabilitas. Dalam konteks pengembangan perangkat lunak modern, terutama di lingkungan DevOps, ISO 25010 menjadi acuan penting dalam mengukur efektivitas dan efisiensi suatu sistem dengan menilai sejauh mana perangkat lunak memenuhi persyaratan bisnis serta pengalaman pengguna yang optimal (Lapointe-Boisvert dkk., 2024). Berikut adalah beberapa parameter kualitas berdasarkan ISO 25010 yang dapat dilihat pada tabel 2.3.

Tabel 2.3 Parameter Kualitas Berdasarkan ISO 25010

No	<i>Quality Characteristic</i>	<i>Quality Sub-Characteristic</i>
1	<i>Functional Suitability</i>	<i>Functional Completeness</i>
		<i>Functional Correctness</i>
		<i>Functional appropriateness</i>
2	<i>compatibility</i>	<i>Co-Existence</i>
		<i>Interoperability</i>
3	<i>Security</i>	<i>Confidentiality</i>
		<i>Integrity</i>
		<i>Non-Repudiation</i>
		<i>Accountability</i>
		<i>Authenticity</i>

Tabel diatas merupakan aspek penting yang ada pada ISO 25010 untuk memastikan sebuah perangkat lunak benar-benar layak digunakan.

2.1.11 ISO 25023

ISO 25023 merupakan bagian dari standar Software Product Quality Requirements and Evaluation (SQuaRE) yang dikembangkan oleh International Organization for Standardization (ISO) untuk menyediakan metrik kuantitatif dalam mengevaluasi kualitas perangkat lunak dan sistem informasi. ISO 25023

berfungsi sebagai pelengkap ISO 25010, yang mendefinisikan karakteristik kualitas perangkat lunak, dengan menyediakan metode pengukuran yang lebih spesifik dan terukur.

ISO 25023 mencakup berbagai metrik yang diklasifikasikan berdasarkan karakteristik kualitas perangkat lunak, seperti fungsionalitas, keandalan, efisiensi kinerja, kompatibilitas, kegunaan, keamanan, maintainability, dan portabilitas.

Dalam konteks evaluasi aplikasi commerce, ISO 25023 digunakan untuk menganalisis berbagai aspek kualitas, termasuk keandalan sistem, efisiensi sumber daya, dan kegunaan aplikasi. Standar ini menjadi referensi penting bagi pengembang perangkat lunak dalam meningkatkan performa aplikasi agar lebih optimal dan sesuai dengan kebutuhan pengguna. Dengan menerapkan ISO 25023, evaluasi kualitas aplikasi dapat dilakukan secara lebih objektif dan berbasis data, sehingga menghasilkan sistem yang lebih andal dan efisien (Perdana dkk., 2021). Selain itu ada juga beberapa parameter kualitas berdasarkan ISO 25023.

1) *Functional Completeness*

$$X = 1 - \frac{A}{B}$$

Keterangan:

- X = Nilai kelengkapan fungsional
- A = Jumlah fungsi yang hilang
- B = Total fungsi

Pada rumus diatas merupakan parameter rumus, yang dimana kualitas ditentukan oleh selisih antara fungsi yang direncanakan dalam spesifikasi dengan fungsi yang benar-benar terimplementasi; semakin sedikit fungsi yang hilang, semakin mendekati angka satu nilainya.

2) *Functional Correctness*

$$X = 1 - \frac{A}{B}$$

Keterangan:

- X = Nilai kebenaran fungsional (0 - 1).
- A = Jumlah fungsi yang menghasilkan output salah
- B = Total fungsi

Pada rumus diatas merupakan parameter yang menitikberatkan pada

akurasi hasil operasi, yang dihitung dengan meminimalkan rasio kesalahan output terhadap seluruh fungsi yang dieksekusi selama pengujian.

3) *Functional Appropriateness*

$$X = \frac{A}{B}$$

Keterangan:

- X = Nilai kebenaran fungsional.
- A = Jumlah fungsi yang sesuai
- B = Total fungsi

Pada gambar 2.8 merupakan parameter yang menilai efektivitas perangkat lunak dalam memfasilitasi tugas spesifik pengguna, yang diukur melalui proporsi fitur yang benar-benar relevan dan mendukung tujuan operasional pengguna dibandingkan dengan total fitur yang disediakan.

2.1.12 *Figma*

Figma adalah aplikasi berbasis web yang digunakan untuk merancang antarmuka pengguna (UI) dan pengalaman pengguna (UX). Aplikasi ini banyak diminati oleh desainer UI/UX karena mendukung kolaborasi tim secara real time dan menyediakan berbagai fitur yang mempermudah proses desain. Keunggulan lainnya adalah kemudahan akses dan penggunaan, karena Figma dapat digunakan langsung melalui browser tanpa perlu mengunduh aplikasi.

Keunggulan utama Figma adalah kemudahan akses dan penggunaan. Karena Figma berbasis web, pengguna dapat mengaksesnya langsung melalui browser tanpa perlu mengunduh aplikasi tambahan. Hal ini memudahkan pengembang dan desainer untuk bekerja dari berbagai perangkat dan lokasi. Dalam pengembangan aplikasi seperti sistem perpustakaan digital, Figma dapat digunakan untuk membuat desain antarmuka pengguna yang intuitif dan terstruktur. Desain yang dihasilkan dapat diekspor ke format yang kompatibel dengan alat pengembangan, sehingga mempercepat proses transisi dari desain ke pengembangan aplikasi.

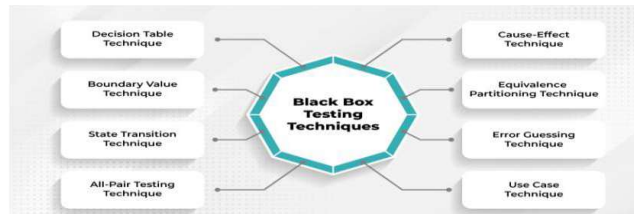
2.1.13 *Blackbox Testing*

Black box testing yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk

mengetahui apakah fungsi fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Black box testing adalah tipe testing yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para tester memandang perangkat lunak seperti layaknya sebuah "kotak hitam" yang tidak penting dilihat isinya, tapi cukup dikenal proses testing dibagian luar, hal ini dapat dilihat pada gambar 2.5. Beberapa teknik testing yang tergolong tipe ini antara lain:

- a) Decision Table Technique
- b) Boundary Value Technique
- c) State Transition Technique
- d) All-Pair Testing Technique
- e) Cause Effect Technique
- f) Equivalence Partitioning Technique
- g) Error Guessing Technique
- h) Use Case Technique



Gambar 2.6 Alur Blackbox Testing

Alur black box testing pada gambar 2.9 yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Black box testing adalah tipe testing yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para tester memandang perangkat lunak seperti layaknya sebuah "kotak hitam" yang tidak penting dilihat isinya, tapi cukup dikenal proses testing dibagian luar.

2.1.14 Usability Testing

Usability testing atau pengujian usability penting untuk dilakukan karena pengguna akan meninggalkan aplikasi dan memilih untuk tidak menggunakannya

kembali apabila aplikasi gagal menunjukkan dengan jelas apa yang pengguna cari atau inginkan dari aplikasi tersebut. Pengukuran usability dapat dilakukan dengan melakukan langkah-langkah yang sama seperti pada penelitian lainnya yaitu pemilihan kuesioner dan partisipan, penentuan ukuran sampel, mengolah dan interpretasi data sesuai dengan karakteristik data penelitian. Secara umum, Pengujian usability ini mengacu pada sejauh mana pengguna mempelajari dan menggunakan produk untuk mencapai tujuannya juga sejauh mana pengguna puas dalam menggunakan produk tersebut.

Menurut Nielsen, 5 komponen yang mendefinisikan usability didefinisikan yaitu:

- a) Learnability: berarti mengukur seberapa mudah pengguna dapat menggunakan produk untuk pertama kali.
- b) Efficiency: berarti mengukur seberapa cepat pengguna dapat melakukan tugasnya
- c) Memorability: berarti seberapa jauh pengguna mengingat proses atau langkah yang dilakukan dalam mencapai tujuannya
- d) Error: berarti sebanyak apa pengguna melakukan error, dan sejauh mana akibatnya dan seberapa mudah untuk mengatasinya
- e) Satisfaction: berarti tanggapan pengguna tentang desain produk dan ketika menggunakan produk

Kemudian disini penulis menggunakan System Usability Scale (SUS) dikenal sebagai kuesioner yang digunakan dalam mengukur kepuasan pengguna yang "quick and dirty", yang berarti penggunaannya sangat cepat dan data yang dihasilkan dapat dipercaya. Kuesioner SUS terdiri dari sepuluh pertanyaan dengan 5 pernyataan positif pada nomor ganjil dan 5 pertanyaan negatif pada nomor genap disertai keterangan pilihan sangat tidak setuju, tidak setuju, netral, setuju, dan sangat setuju.

2.2 Penelitian Terdahulu

Mempelajari penelitian terdahulu dapat memberikan masukan yang berharga dan ide-ide baru terkait penelitian yang akan dilakukan. Berikut adalah beberapa penelitian terdahulu yang akan digunakan sebagai perbandingan dalam pembuatan sistem proyek akhir:

Penelitian pertama dilakukan oleh (Pratiwi dkk., 2018) yaitu sebuah sistem informasi perpustakaan berbasis web pada SMK Yadika 13 Tambun Utara. Penelitian ini bertujuan untuk mengembangkan sistem informasi yang lebih efektif dalam mengelola data pustaka dan transaksi peminjaman. Latar belakang penelitian ini adalah rendahnya efisiensi pengelolaan data secara manual yang sering menyebabkan kesalahan dan penundaan proses pelayanan.

Lalu penelitian kedua yang dilakukan oleh (Chaidir dkk., 2021) pada MTs Al-Husna Depok. Penelitian ini bertujuan untuk merancang sistem informasi perpustakaan berbasis web untuk mempercepat proses peminjaman dan pengembalian buku. Penelitian ini dilatarbelakangi oleh kebutuhan akan sistem yang mampu mengurangi tumpukan pekerjaan pustakawan akibat pengelolaan data secara manual. Metode yang digunakan ialah metode prototype.

Penelitian ketiga dilakukan oleh (Amanda, 2022) untuk membangun sebuah sistem informasi perpustakaan berbasis web yang meningkatkan efektivitas layanan peminjaman dan pengembalian buku. Penelitian ini dilatarbelakangi oleh lamanya waktu yang dibutuhkan dalam proses pencatatan manual, sehingga mengurangi efisiensi operasional. Tujuan utama penelitian adalah: Mempercepat proses peminjaman dan pengembalian buku melalui sistem terkomputerisasi. Menyediakan fitur pengelolaan akun pustakawan untuk kemudahan administrasi. Metode yang digunakan adalah metode prototype.

Penelitian ke empat dilakukan oleh (Amroni & Fatimah, 2024) dengan mengembangkan sistem informasi perpustakaan berbasis web menggunakan metode prototype pada SMK Panca Karya Sentul. Penelitian ini bertujuan untuk memberikan solusi terhadap permasalahan pencatatan data perpustakaan yang masih dilakukan secara manual sehingga sering terjadi kesalahan dalam proses peminjaman, pengembalian, dan pencarian data buku. Metode yang digunakan dalam penelitian ini adalah pendekatan berorientasi objek dengan Unified Modeling Language (UML) serta metode prototype dalam pengembangan perangkat lunak. Teknik pengumpulan data dilakukan melalui observasi dan analisis sistem yang sedang berjalan. Hasil penelitian menunjukkan bahwa sistem informasi perpustakaan berbasis web dapat memudahkan petugas dalam pendataan anggota, pendataan buku, proses peminjaman dan pengembalian, serta pembuatan laporan,

sehingga meminimalkan kehilangan data dan meningkatkan efisiensi layanan perpustakaan di SMK Panca Karya Sentul.

Dan penelitian kelima dilakukan oleh (Nalatissifa dkk., 2023) pada SMK Negeri 1 Bumijawa, yang menghasilkan sebuah sistem informasi perpustakaan berbasis web. Penelitian ini bertujuan untuk menyediakan alat bantu digital yang dapat memudahkan pustakawan dalam mencatat dan memproses data perpustakaan. Tujuan utama penelitian adalah: Mempercepat proses administrasi pustaka melalui digitalisasi. Meningkatkan akurasi data dan mengurangi risiko kehilangan catatan manual. Metode yang digunakan tidak disebutkan oleh peneliti.

Tabel 2.4 Perbandingan variabel penelitian

Nama Peneliti	Judul	Metode	Kelebihan	Kekurangan
(Pratiwi dkk., 2018)	Rancang Bangun Sistem Informasi Perpustakaan Pada Sekolah SMK Yadika 13 Tambun Utara Berbasis Web.	<i>Waterfall</i>	Fokus pada pengelolaan data pustaka yang lebih efektif dan meminimalkan kesalahan manual, Berbasis web sehingga mudah diakses pengguna..	Fitur interaksi pengguna terbatas, seperti pencarian yang kurang fleksibel. Tidak membahas integrasi data antar sekolah.
(Chaidir dkk., 2021)	Rancang Bangun Sistem Informasi Perpustakaan Berbasis Web Pada MTs Al-Husna Depok.	<i>Waterfall</i>	Sistem dirancang untuk mempercepat proses peminjaman dan pengembalian buku. Mempermudah pegawai perpustakaan dalam melacak inventaris buku.	Fokus pada pengelolaan internal, kurang fitur untuk meningkatkan pengalaman pengguna akhir.
(Amanda, 2022)	Rancang Bangun Sistem Informasi	<i>Prototype</i>	Penggunaan web meningkatkan aksesibilitas dan efisiensi layanan	Sistem cenderung hanya untuk internal pustakawan, minim pelibatan anggota

Nama Peneliti	Judul	Metode	Kelebihan	Kekurangan
	Perpustakaan Berbasis Web Untuk Efektivitas Layanan Peminjaman dan Pengembalian Buku.		perpustakaan. Menyediakan pengelolaan berbasis akun untuk pustakawan.	perpustakaan dalam pengelolaan.
(Amroni & Fatimah, 2024)	Rancang Bangun Sistem Informasi Perpustakaan Berbasis Web dengan Metode Prototype(Studi Kasus : SMK Panca Karya Sentul).	<i>Prototype</i>	mempermudah pendataan anggota, pendataan buku, serta proses peminjaman dan pengembalian buku, sehingga meningkatkan efisiensi kerja petugas dan mengurangi kesalahan pencatatan.	evaluasi sistem masih sebatas pengujian fungsional tanpa pengukuran kepuasan pengguna, aspek keamanan sistem belum dibahas secara mendalam, dan fitur yang disediakan masih terbatas pada pengelolaan buku fisik tanpa dukungan fitur lanjutan seperti e-book atau integrasi dengan sistem akademik sekolah.
(Nalatissiffa dkk., 2023)	Rancang Bangun Sistem Informasi Perpustakaan Berbasis Web Pada SMK Negeri 1 Bumijawa.	Tidak menyebutkan metode	Fitur antarmuka sederhana dan mudah digunakan oleh pustakawan. Mendukung pengelolaan data anggota perpustakaan.	Kurang mendukung penggunaan multi-platform (seperti, perangkat seluler).

Berdasarkan penelitian terdahulu, pengembangan sistem informasi perpustakaan dilakukan dengan beberapa metode yang berbeda. Penelitian oleh (Pratiwi dkk., 2018) dan (Chaidir dkk., 2021) menggunakan metode Waterfall, sedangkan penelitian oleh (Amanda, 2022) serta (Amroni & Fatimah, 2024)

menggunakan metode Prototype. Metode Waterfall memiliki tahapan yang terstruktur, namun kurang fleksibel ketika terdapat perubahan kebutuhan dari pengguna. Jika terjadi kesalahan atau kebutuhan baru, sistem sulit diperbaiki karena harus menunggu tahap berikutnya selesai.

Berbeda dengan metode tersebut, metode *Prototype* memungkinkan pengembang membuat contoh sistem lebih awal dan memperbaikinya secara bertahap berdasarkan masukan dari pengguna. Dengan cara ini, sistem dapat disesuaikan langsung dengan kebutuhan pustakawan dan siswa. Meskipun beberapa penelitian telah menggunakan metode Prototype, sistem yang dikembangkan masih berfokus pada pengelolaan data perpustakaan saja. Penelitian ini mengembangkan sistem dengan metode *Prototype* yang dipadukan dengan fitur gamifikasi berupa poin, badge, dan papan peringkat (leaderboard). Pendekatan ini menjadi pembeda dari penelitian sebelumnya karena sistem tidak hanya membantu administrasi perpustakaan, tetapi juga dirancang untuk meningkatkan minat baca dan partisipasi siswa.