

## **BAB II**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian Beberapa hasil penelitian yang pernah dilakukan oleh peneliti sebelumnya yang memiliki bidang dan tema yang sama dengan penelitian yang akan dilakukan. Maka, penulis melakukan kajian terhadap beberapa hasil penelitian yang telah dilakukan oleh peneliti sebelumnya.

1. Sistem informasi perpustakaan berbasis web telah banyak diterapkan di berbagai institusi pendidikan untuk mempermudah akses dan manajemen perpustakaan. Menurut penelitian oleh Rahman (2020), sistem ini memungkinkan pengguna untuk melakukan peminjaman, pengembalian, dan pencarian buku secara online, yang meningkatkan akurasi dan aksesibilitas layanan perpustakaan.
2. Studi oleh Ananda dan Hidayat (2019) menunjukkan bahwa penerapan sistem informasi perpustakaan berbasis web di sekolah-sekolah menengah mampu meningkatkan kualitas pelayanan perpustakaan, serta mengurangi waktu yang dibutuhkan untuk administrasi perpustakaan .
3. Framework CodeIgniter dikenal karena kemudahannya dalam pengembangan aplikasi web, terutama dalam hal kecepatan dan efisiensi kinerja. Penelitian oleh Setiawan (2021) menunjukkan bahwa CodeIgniter mampu menghasilkan aplikasi yang responsif dengan pengelolaan sumber daya yang optimal .

4. Menurut studi oleh Prasetyo dan Nugroho (2022), penerapan CodeIgniter pada sistem informasi berbasis web terbukti mempercepat proses pengembangan dan memudahkan integrasi dengan database MySQL. Hal ini membuat CodeIgniter menjadi pilihan yang populer di kalangan pengembang web dalam lima tahun terakhir .
5. Studi Dampak pada Pengguna: Hasil penelitian dari Putri dan Wicaksono (2019) menunjukkan bahwa implementasi sistem informasi perpustakaan berbasis web memiliki dampak positif terhadap kepuasan pengguna. Pengguna merasa lebih mudah dalam mengakses informasi terkait ketersediaan buku dan status peminjaman .
6. Pengembangan Sistem Informasi Berkelanjutan: Menurut penelitian oleh Sari dan Yulianto (2020), sistem informasi perpustakaan yang dirancang dengan baik mampu beradaptasi dengan perkembangan teknologi dan kebutuhan pengguna, sehingga dapat terus memberikan manfaat jangka panjang .
7. Arsitektur MVC memisahkan logika aplikasi, tampilan, dan data, sehingga memudahkan pengembangan, pemeliharaan, dan pengembangan lebih lanjut. Penelitian yang dilakukan oleh Kurniawan et al. (2021) menekankan bahwa MVC memberikan struktur yang lebih baik untuk aplikasi besar dan kecil, serta meningkatkan kualitas kode yang dihasilkan.

Dari keseluruhan tinjauan pustaka, dapat disimpulkan bahwa penelitian-penelitian sebelumnya telah menggaris bawahi pentingnya pemanfaatan teknologi berbasis web dalam pengelolaan dan pengarsipan data buku di perpustakaan. Adanya tren menuju aplikasi berbasis web dan perpustakaan digital juga mencerminkan upaya untuk meningkatkan efektivitas, aksesibilitas, dan mempermudah pencarian informasi dalam pengelolaan perpustakaan di berbagai lembaga pendidikan dan umum. Oleh karena itu, penelitian ini akan memberikan kontribusi lebih lanjut dalam pengembangan sistem informasi perpustakaan yang dapat memenuhi kebutuhan dan tuntutan era digital saat ini.

**Table 2.1** Tinjauan Pustaka

No	Judul	Penulis	Hasil/Kesimpulan
1	<i>PHP and MySQL Web Development</i> . Addison-Wesley.	Beck, K. (2019)	Buku ini menjelaskan bagaimana mengintegrasikan PHP dengan MySQL untuk menciptakan aplikasi web
2	<i>Object-Oriented Analysis and Design with Applications</i> . Addison-Wesley.	Booch, G. (2020)	penerapan prinsip-prinsip analisis dan desain berorientasi objek membantu dalam menciptakan sistem yang terstruktur dan mudah dipelihara.
3	<i>UML for the IT Architect: A Comprehensive Guide to Object-Oriented Modeling</i> . Wiley.	Fowler, M., & Scott, K. (2020)	penggunaan UML dalam konteks arsitektur TI mempermudah proses desain dan pengembangan sistem perangkat lunak yang kompleks.
4	Penerapan Arsitektur MVC dalam Pengembangan Sistem Informasi. Jurnal Rekayasa Perangkat Lunak	Kurniawan, A., et al. (2021)	arsitektur MVC meningkatkan struktur dan organisasi sistem informasi, memudahkan pengembangan dan pemeliharaan.
5	<i>Penerapan CodeIgniter pada Sistem Informasi Berbasis Web</i> . Jurnal Sistem Informasi	Prasetyo, D., & Nugroho, T. (2022)	CodeIgniter merupakan framework yang efektif untuk pengembangan aplikasi web dengan struktur yang jelas dan dokumentasi lengkap.
6	Pengembangan Sistem Informasi Perpustakaan Berbasis Web. Jurnal Teknologi Informasi	Rahman, A. (2020)	pengembangan sistem berbasis web harus mempertimbangkan berbagai aspek teknis dan fungsional untuk memastikan keberhasilan implementasi.
7	Keunggulan Framework CodeIgniter dalam Pengembangan Web. Jurnal Komputasi	Setiawan, B. (2021)	CodeIgniter adalah alat yang sangat berguna untuk pengembangan aplikasi web yang cepat dan efisien.

Seperti terlihat pada **Tabel 2.1**, perbedaan dari ketiga referensi dengan judul yang diangkat oleh penulis terletak pada penggunaan sistem informasi dan pengolahan data perpustakaan seperti data buku, data anggota, data peminjaman, data pengembalian dan data laporan-laporan yang dilakukan

secara tepat dan cepat. Sehingga siswa merasa dilayani secara baik dengan adanya sistem tersebut.

## **2.2 Dasar Teori**

### **2.2.1 Sistem Informasi Perpustakaan**

Sistem informasi perpustakaan adalah sebuah sistem yang dirancang untuk mengelola berbagai aspek dari operasi perpustakaan, termasuk katalog buku, peminjaman, pengembalian, dan denda. Menurut O'Connor dan Nilsen (2020), sistem ini bertujuan untuk meningkatkan efisiensi operasional dan mempermudah akses informasi bagi pengguna.

Fungsionalitas dasar dari sistem informasi perpustakaan meliputi manajemen buku, anggota, peminjaman, pengembalian, dan laporan. Sistem ini juga harus mampu mengelola denda dan menyediakan informasi tentang ketersediaan buku (Smith, 2021).

### **2.2.2 PHP**

PHP (Hypertext Preprocessor) adalah bahasa pemrograman server-side yang dirancang untuk pengembangan web dinamis. PHP memungkinkan pengembang untuk membuat konten web yang dapat berinteraksi dengan basis data dan menghasilkan halaman web yang dinamis dan interaktif. PHP pertama kali dikembangkan oleh Rasmus Lerdorf pada tahun 1994 dan telah berkembang menjadi salah satu

bahasa pemrograman yang paling populer untuk pengembangan web (Welling & Thomson, 2021).

PHP sering digunakan untuk berinteraksi dengan sistem manajemen basis data, seperti MySQL, PostgreSQL, dan SQLite. Ini memungkinkan pengembangan aplikasi web yang dapat menyimpan, mengambil, dan memanipulasi data dengan mudah (Beck, 2019).

### **2.2.3 Mysql**

MySQL adalah sistem manajemen basis data relasional (RDBMS) open-source yang populer dan digunakan untuk mengelola data dalam berbagai aplikasi, dari situs web kecil hingga aplikasi perusahaan besar. MySQL menggunakan bahasa Structured Query Language (SQL) untuk pengelolaan data (Korth & Silberschatz, 2021).

MySQL memungkinkan pengguna untuk menulis dan mengeksekusi query SQL untuk mendapatkan informasi dari database, memperbarui data, dan melakukan analisis data (Elmasri & Navathe, 2021).

### **2.2.4 Codeigniter**

CodeIgniter adalah framework PHP yang dirancang untuk mempermudah pengembangan aplikasi web dengan struktur yang ringan dan cepat. Menurut Setiawan (2021), CodeIgniter menyediakan arsitektur Model-View-Controller (MVC) yang memisahkan logika

aplikasi, tampilan, dan data, sehingga mempermudah pengembangan dan pemeliharaan aplikasi.

CodeIgniter dikenal karena kemudahannya dalam penggunaan dan konfigurasi, serta kemampuannya untuk meningkatkan kecepatan pengembangan aplikasi web. Framework ini juga menawarkan dokumentasi yang lengkap dan dukungan komunitas yang aktif (Prasetyo & Nugroho, 2022).

### **2.2.5 Arsitektur Model-View-Controller (MVC)**

Arsitektur MVC adalah pola desain perangkat lunak yang memisahkan aplikasi menjadi tiga komponen utama: Model (data dan logika bisnis), View (antarmuka pengguna), dan Controller (pengendali alur aplikasi). Menurut Kurniawan et al. (2021), pemisahan ini memungkinkan pengembangan dan pemeliharaan aplikasi yang lebih terstruktur dan terorganisir.

MVC membantu dalam mengelola kompleksitas aplikasi dengan membagi tanggung jawab. Ini memudahkan pengembangan, pengujian, dan pemeliharaan aplikasi karena setiap komponen memiliki fungsi dan tanggung jawab yang jelas (Sari & Yulianto, 2020).

### **2.2.6 Unified Modeling Language (UML)**

Unified Modeling Language (UML) adalah bahasa pemodelan standar yang digunakan dalam rekayasa perangkat lunak untuk

memvisualisasikan, spesifikasikan, membangun, dan mendokumentasikan artefak sistem perangkat lunak. UML membantu pengembang untuk memodelkan berbagai aspek dari sistem perangkat lunak secara terstruktur dan sistematis. UML dikembangkan oleh Grady Booch, Ivar Jacobson, dan James Rumbaugh pada tahun 1997 dan menjadi standar industri yang diakui (Rumbaugh, Jacobson, & Booch, 2019).

Diagram ini menggambarkan bagaimana sistem berperilaku dan berinteraksi. Jenis-jenis diagram perilaku termasuk Diagram Kasus Penggunaan (Use Case Diagram), Diagram Aktivitas (Activity Diagram), Diagram Urutan (Sequence Diagram), dan Diagram Kolaborasi (Collaboration Diagram) (Booch, 2020).

### **2.2.7 Use Case Diagram**

Use Case Diagram adalah salah satu jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan fungsionalitas sistem dari perspektif pengguna. Diagram ini menunjukkan interaksi antara aktor (pengguna atau sistem eksternal) dan sistem, serta bagaimana aktor menggunakan fitur atau kasus penggunaan tertentu dari sistem (Jacobson, 2021).

Kasus penggunaan adalah fungsi atau layanan yang ditawarkan oleh sistem kepada aktor. Setiap kasus penggunaan menggambarkan sebuah interaksi yang diharapkan antara aktor dan sistem untuk mencapai tujuan tertentu (Booch, 2020).



### **2.2.8 Activity Diagram**

Activity Diagram adalah jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan alur kerja atau proses yang melibatkan aktivitas dan keputusan dalam sistem. Diagram ini menunjukkan bagaimana aktivitas-aktivitas tertentu berhubungan satu sama lain dan bagaimana alur proses dilakukan dari awal hingga akhir (Fowler, 2019).

Diagram ini digunakan untuk mendokumentasikan proses bisnis atau alur kerja sistem, sehingga memudahkan pemahaman dan pemeliharaan proses di masa depan (Sommerville, 2020).

### **2.2.9 Class Diagram**

Class Diagram adalah salah satu jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan struktur statis dari sistem perangkat lunak. Diagram ini menunjukkan kelas-kelas yang ada dalam sistem, atribut, metode, serta hubungan antara kelas-kelas tersebut (Booch, 2020).

Diagram ini digunakan untuk merancang sistem perangkat lunak dengan mendefinisikan struktur kelas dan hubungan antara kelas, sehingga memberikan gambaran tentang arsitektur sistem (Fowler, 2019).

### **2.2.10 Sequence Diagram**

Sequence Diagram adalah jenis diagram dalam Unified Modeling Language (UML) yang menggambarkan interaksi dinamis antara objek dalam sistem melalui urutan pesan yang dikirimkan antar objek. Diagram ini fokus pada waktu dan urutan pesan untuk memodelkan bagaimana sistem merespons berbagai skenario (Fowler & Scott, 2020).

Diagram ini digunakan untuk mendokumentasikan proses interaksi sistem yang kompleks, membantu dalam pemahaman dan pengelolaan alur kerja (Fowler, 2019).

### **2.2.11 Peminjaman Buku**

Proses peminjaman dan pengembalian buku harus diatur dengan baik untuk memastikan bahwa buku dikelola dengan efisien. Hal ini termasuk pencatatan yang akurat dan penanganan masalah seperti buku hilang atau rusak (Rahman, 2020).

Prasetyo dan Nugroho (2022) menyatakan bahwa peminjaman buku dapat dilakukan melalui antarmuka pengguna yang intuitif, di mana anggota perpustakaan dapat memilih buku yang tersedia, memeriksa status ketersediaan, dan mengajukan permohonan peminjaman. Setelah buku dipinjam, sistem akan mencatat waktu peminjaman dan mengingatkan anggota mengenai tanggal pengembalian.

### **2.2.12 Pengelolaan Denda**

Pengelolaan denda dalam sistem informasi perpustakaan penting untuk mengatasi keterlambatan pengembalian buku. Menurut penelitian oleh Putri dan Wicaksono (2019), sistem yang efektif dalam manajemen denda dapat meningkatkan kepatuhan anggota terhadap aturan peminjaman dan mengurangi kerugian finansial bagi perpustakaan.

Menurut Putri dan Wicaksono (2019), sistem denda yang efektif dalam perpustakaan dapat membantu memastikan bahwa anggota lebih bertanggung jawab dalam mengelola buku yang mereka pinjam. Sistem ini biasanya mencakup mekanisme otomatis untuk menghitung denda berdasarkan lamanya keterlambatan atau tingkat kerusakan buku. Hal ini memberikan transparansi dan konsistensi dalam pemberlakuan denda, sehingga setiap anggota diperlakukan adil.