

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Laporan ini ditulis dengan menggali informasi dari penelitian-penelitian sebelumnya sebagai bahan perbandingan, baik mengenai kekurangan atau kelebihan yang sudah ada. Selain itu, juga menggali informasi dari buku-buku maupun jurnal dalam rangka mendapatkan suatu informasi yang ada sebelumnya tentang teori yang berkaitan dengan judul yang digunakan untuk memperoleh landasan teori ilmiah.

Penelitian oleh (Hasan, 2019) berjudul "Aplikasi Penyewaan Mobil Berbasis *Website* (Studi Kasus pada Rental Mobil Lotus Purworejo)" mengkaji sistem informasi manajemen yang mencakup pendaftaran akun pelanggan, transaksi *booking*, laporan transaksi, dan pengelolaan data rental mobil. Hasil penelitian ini menunjukkan bahwa sistem berbasis *web* mampu mempermudah proses pendaftaran dan pengelolaan transaksi. Namun, masih ada kelemahan, yaitu proses penyewaan yang tetap manual dan ketiadaan pemantauan ketersediaan mobil secara *real-time*. Kekurangan ini menyoroti kebutuhan akan sistem yang lebih terintegrasi dan otomatis, yang menjadi fokus dalam pengembangan aplikasi penyewaan meja *billiard* pada laporan ini.

Penelitian lainnya oleh (Fadhilurrahman dan Capah, 2020) yang berjudul "Aplikasi Penyewaan Lapangan Futsal Berbasis *Web*" menekankan pada kemudahan penyewaan lapangan dan pengolahan data secara *online*. Aplikasi ini berhasil memudahkan proses penyewaan dan pengelolaan data sewa serta laporan keuangan. penelitian ini menggarisbawahi bahwa aplikasi berbasis *web* dapat meningkatkan efisiensi dan kenyamanan baik bagi pengguna maupun pengelola. Pendekatan serupa akan diterapkan dalam laporan ini untuk memastikan bahwa sistem penyewaan meja *billiard* tidak hanya mengotomatisasi penyewaan tetapi juga menyediakan pemantauan ketersediaan meja secara *real-time*.

Penelitian oleh (Siregar, 2021) yang berjudul "Aplikasi Penyewaan Peralatan *Outdoor* Berbasis *Web*" juga memberikan kontribusi penting dalam memahami aplikasi berbasis *web*. Penelitian ini fokus pada pengelolaan

penyewaan peralatan *outdoor* dengan fitur pendaftaran pelanggan, manajemen stok peralatan, dan pelaporan transaksi. Hasil penelitian menunjukkan bahwa penggunaan aplikasi *web* dapat meningkatkan efisiensi manajemen stok dan transaksi. Namun, kekurangan yang ditemukan adalah kurangnya integrasi dengan sistem pembayaran *online*, yang menyebabkan keterbatasan dalam proses pembayaran.

Aplikasi penyewaan meja *billiard* berbasis website yang dikembangkan dalam penelitian ini menunjukkan keunggulan dalam mengatasi berbagai kekurangan yang ditemukan dalam penelitian-penelitian sebelumnya. Dalam penelitian oleh (Hasan, 2019) sistem informasi manajemen yang dikaji mencakup pendaftaran akun pelanggan, transaksi *booking*, laporan transaksi, dan pengelolaan data rental mobil. Namun, penelitian tersebut mengidentifikasi kelemahan dalam proses penyewaan yang tetap manual dan ketiadaan pemantauan ketersediaan mobil secara *real-time*. Aplikasi penyewaan meja *billiard* ini mengatasi masalah tersebut dengan menyediakan sistem otomatisasi penyewaan dan pemantauan ketersediaan meja secara *real-time*, yang meningkatkan efisiensi dan akurasi dalam pengelolaan penyewaan.

Penelitian oleh (Fadhlurrahman dan Capah, 2020) menekankan kemudahan dalam proses penyewaan lapangan futsal dan pengolahan data secara *online*. Aplikasi yang dikembangkan dalam penelitian tersebut berhasil memudahkan penyewaan dan pengelolaan data sewa serta laporan keuangan. Meskipun penelitian ini menunjukkan bahwa aplikasi berbasis *web* dapat meningkatkan efisiensi dan kenyamanan bagi pengguna dan pengelola, aplikasi penyewaan meja *billiard* yang dikembangkan dalam penelitian ini melangkah lebih jauh dengan menyediakan fitur pemantauan ketersediaan meja secara *real-time* dan sistem pemindahan pemain otomatis, yang memastikan tidak terjadi benturan jadwal atau *double booking*.

Penelitian oleh (Siregar, 2021) mengenai aplikasi penyewaan peralatan *outdoor* berbasis *web* menunjukkan bahwa penggunaan aplikasi *web* dapat meningkatkan efisiensi dalam manajemen stok dan transaksi. Namun, penelitian ini menemukan kekurangan dalam kurangnya integrasi dengan sistem pembayaran

online, yang membatasi proses pembayaran. Aplikasi penyewaan meja *billiard* yang dikembangkan dalam penelitian ini mengatasi kekurangan tersebut dengan mengintegrasikan sistem pembayaran *online* melalui Midtrans, sehingga memudahkan berbagai metode pembayaran dan meningkatkan kepuasan pengguna.

Secara keseluruhan, aplikasi penyewaan meja *billiard* berbasis website ini tidak hanya mengatasi berbagai kekurangan yang ditemukan dalam penelitian-penelitian sebelumnya tetapi juga menawarkan fitur tambahan yang meningkatkan efisiensi, akurasi, dan kenyamanan dalam pengelolaan penyewaan meja *billiard*.

2.2 Aplikasi Website

Aplikasi *website* merupakan sistem informasi yang memungkinkan interaksi antara pengguna dan platform melalui antarmuka berbasis *web*. Proses interaksi ini terbagi menjadi tiga tahap utama yaitu permintaan, pemrosesan, dan jawaban. Dalam konteks ini, sebuah *website* dapat diartikan sebagai kumpulan halaman yang dirancang untuk menyajikan informasi dalam berbagai bentuk, termasuk teks, gambar, animasi, suara, atau kombinasi dari semua elemen tersebut. Informasi yang disajikan pada *website* bisa bersifat statis atau dinamis, dan halaman-halaman tersebut diorganisasikan sedemikian rupa sehingga membentuk struktur yang saling terkait satu sama lain. Hubungan antar halaman ini biasanya diwujudkan melalui jaringan *hyperlink*, memungkinkan pengguna untuk menavigasi dari satu halaman ke halaman lainnya dengan mudah. Menurut (Arribathi, et al., 2019) *website* adalah kumpulan halaman yang berisi informasi dan data yang dapat diakses melalui jaringan internet secara global. Sebuah *website* biasanya terdiri dari berbagai elemen seperti gambar, teks, suara, dan animasi, sehingga menjadikannya media informasi yang menarik bagi pengguna internet untuk dikunjungi.

Aplikasi *website* memiliki banyak keunggulan, seperti aksesibilitas dari berbagai perangkat dengan koneksi internet dan pemeliharaan terpusat yang memastikan versi terbaru tanpa perlu unduhan ulang. Biaya pengembangannya lebih rendah karena satu aplikasi dapat berjalan di berbagai platform. Selain itu,

aplikasi ini mudah diintegrasikan dengan layanan lain melalui API, dapat diskalakan dengan mudah, dan memiliki keamanan tinggi dengan enkripsi SSL dan otentikasi multi-faktor. Teknologi seperti HTML5 dan CSS3 mendukung pengalaman pengguna yang interaktif, dan distribusinya dapat dilakukan langsung tanpa melalui toko aplikasi.

2.3 PHP (Hypertext Preprocessor)

PHP adalah bahasa pemrograman sisi *server* yang digunakan secara luas untuk pengembangan aplikasi *web*. PHP awalnya dikembangkan oleh seorang programmer bernama Rasmus Lerdorf pada tahun 1995, namun semenjak itu selalu dikembangkan oleh kelompok independen yang disebut Group PHP. Menurut (Noviana, 2022) PHP adalah bahasa pemrograman *script server-side* yang dirancang khusus untuk pengembangan *web*. Selain itu, PHP juga dapat digunakan sebagai bahasa pemrograman umum. PHP diciptakan oleh Rasmus Lerdorf pada tahun 1994. Saat ini, PHP merupakan singkatan dari PHP: *Hypertext Preprocessor*, yang merupakan kepanjangan rekursif, yaitu permainan kata di mana singkatan itu sendiri merupakan bagian dari kepanjangannya: PHP: *Hypertext Preprocessor*. PHP tersedia secara gratis (*free*) dan bersifat *Open Source*. PHP dirilis di bawah lisensi PHP License, yang sedikit berbeda dari lisensi GNU *General Public License* (GPL) yang biasanya digunakan untuk proyek *open source*. Sedangkan Menurut (Oetomo dan Mahargiono, 2020:1) PHP adalah bahasa pemrograman yang digunakan secara luas untuk penanganan, pembuatan, dan pengembangan sebuah situs *web* dan biasanya digunakan bersamaan dengan HTML.

PHP memiliki sejumlah kelebihan yang membuatnya populer untuk pengembangan aplikasi *web*. Sebagai bahasa pemrograman *open source*, PHP dapat digunakan dan dimodifikasi secara gratis, yang mengurangi biaya pengembangan. PHP kompatibel dengan berbagai sistem operasi seperti *Windows*, *Linux*, dan *macOS*, memberikan fleksibilitas dalam implementasi. Selain itu, PHP mendukung banyak jenis *database* seperti MySQL, PostgreSQL, dan SQLite, memudahkan integrasi data. Bahasa ini juga dapat diskalakan untuk aplikasi kecil

hingga besar dan mendukung berbagai protokol seperti HTTP, HTTPS, dan FTP. Kemudahan pembelajaran dan sintaks sederhana membuat PHP mudah digunakan, sementara fitur keamanannya, seperti enkripsi data, membantu melindungi aplikasi dari serangan. Berikut adalah tampilan logo PHP.



Gambar 2.1 Logo PHP

2.4 Framework

Menurut (Sallaby dan Kanedi, 2020) *framework* adalah kumpulan berbagai intruksi yang dikelompokkan kedalam suatu *class* dan *function* dengan fungsi masing-masing yang bertujuan untuk memudahkan *developer* dalam pemanggilan *syntax* program yang sama secara berulang-ulang serta dapat menghemat waktu.

Dalam pembuatan *website* ini, menggunakan dua *framework* yang sangat berperan, yaitu:

2.4.1 Bootstrap CSS

Bootstrap merupakan salah satu kerangka kerja (*framework*) untuk membantu pengembangan *web* dengan menggunakan HTML, CSS, dan *Javascript* pada sisi *front-end web*. *Bootstrap* merupakan pilihan banyak pengembang untuk mendesain antarmuka halaman *web*, karena mudah digunakan, dieksplorasi, komponen lengkap, responsif, dapat bekerja pada banyak jenis *browser*, dan juga bersifat *open source*.

Menurut (Oktaviani, et al., 2020) *bootstrap* adalah *framework* yang dikembangkan menggunakan HTML dan CSS, serta dilengkapi dengan efek

JavaScript yang dibangun menggunakan jQuery. *Framework* ini menyediakan berbagai komponen *class* untuk antarmuka dasar yang dirancang untuk menciptakan tampilan yang menarik, bersih, dan ringan. Salah satu fitur unggulannya adalah sistem *grid*, yang memudahkan pengaturan tata letak secara cepat dan mudah. Selain itu, Bootstrap memberikan fleksibilitas kepada pengembang untuk menyesuaikan tampilan *website* dengan menambahkan *class* dan CSS buatan sendiri. Karena kemudahan penggunaan dan kelengkapan fitur yang disediakan, *Bootstrap* menjadi pilihan populer di kalangan pengembang *web* untuk mempercepat proses pengembangan dan memastikan tampilan yang konsisten di berbagai perangkat dan peramban. Berikut adalah tampilan logo *bootstrap* CSS.



Gambar 2.2 Logo Bootstrap CSS

2.4.2 Laravel

Laravel adalah sebuah *framework web* berbasis PHP yang *open-source* dan tidak berbayar, diciptakan oleh Taylor Otwell dan diperuntukkan untuk pengembangan aplikasi *web* yang menggunakan pola MVC. Di Laravel, terdapat komponen *routing* yang berfungsi sebagai jembatan antara permintaan (*request*) dari pengguna dan *controller*. Ini berarti *controller* tidak langsung menerima *request* dari pengguna, melainkan melalui proses *routing* yang mengarahkannya ke *controller* yang sesuai. Hal ini membantu dalam memisahkan logika bisnis (*controller*), tampilan (*view*), dan pengelolaan data (*model*), sehingga membuat aplikasi lebih terstruktur dan mudah untuk dikelola.

Menurut (Hermanto, 2019) Laravel adalah sebuah *framework* PHP yang dirilis di bawah lisensi MIT, dibangun dengan konsep MVC (*model view controller*). Laravel adalah pengembangan *website* berbasis MVP yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu. Berikut adalah tampilan logo Laravel.



Gambar 2.3 Logo Laravel

2.5 Web Server

Web server adalah perangkat lunak yang menjadi tulang punggung dari *World Wide Web* (www), pertama kali muncul pada sekitar tahun 1980-an. *web server* bertugas untuk menangani permintaan dari *client* yang menggunakan *browser* seperti *Netscape Navigator*, *Internet Explorer*, *Mozilla Firefox*, dan lainnya. Ketika *browser* mengirimkan permintaan, *web server* akan memprosesnya dan mengirimkan hasilnya kembali dalam bentuk data yang diminta kepada *browser*.

Data yang dikirimkan oleh *web server* memiliki format standar yang disebut SGML (*Standar General Markup Language*). *Browser* kemudian menampilkan data ini sesuai dengan kemampuannya. Misalnya, *browser* yang hanya mampu menampilkan teks (seperti *Lynx*) tidak akan dapat menampilkan gambar, dan hanya akan menampilkan alternatif teks yang ada. *web server* menggunakan protokol HTTP (*Hypertext Transfer Protocol*) untuk berkomunikasi

dengan *client*-nya, yaitu *web browser*. Protokol ini memungkinkan komunikasi yang saling dimengerti dan lebih mudah antara *web server* dan *browser*.

Menurut (Raharjo, 2020) *web server* adalah perangkat lunak yang menjadi tulang belakang dari *world wide web* (www). *Web server* menunggu permintaan dari *client* yang menggunakan *browser* seperti Netscape Navigator, Internet Explorer, Mozilla, dan program *browser* lainnya. Jika ada permintaan dari *browser*, maka *web server* akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke *browser*. Salah satu *web server* yang digunakan dalam pembuatan *website* ini yaitu:

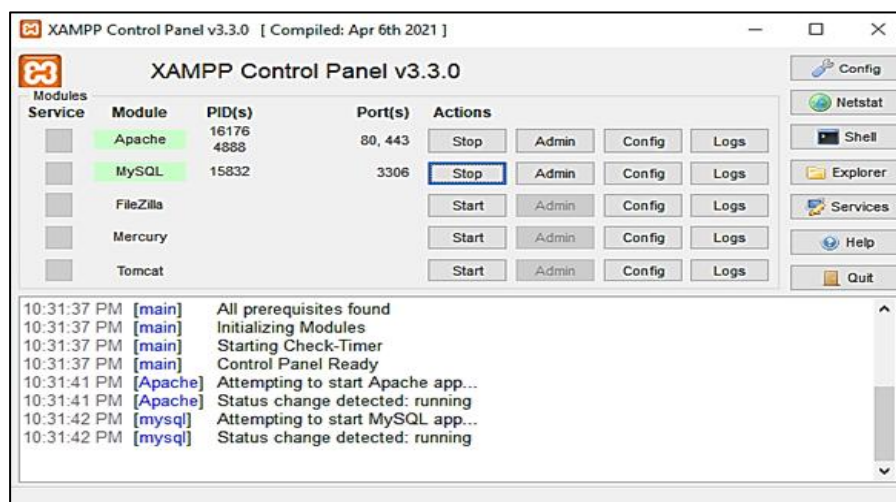
2.5.1 XAMPP

XAMPP adalah perangkat lunak bebas (*free software*), yang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari beberapa program. Fungsi XAMPP sendiri adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri beberapa program antara lain: *Apache HTTP server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl.

XAMPP adalah singkatan dari:

- X (*cross platform*) = Kode penanda dari software *cross platform* yang berarti dapat dijalankan di banyak sistem operasi seperti Windows, Linus, Mac OS dan Solaris
- A (*Apache*) = Aplikasi *web server* gratis dan bisa dikembangkan oleh banyak orang (*open source*) untuk menciptakan halaman *website* yang benar berdasarkan kode program PHP yang ditulis oleh pengembang *web developer*.
- M (MySQL/MariaDB) = Aplikasi *database server* yang menerapkan bahasa pemrograman SQL (*Structured Query Language*) yang berfungsi untuk mengelola dan membuat sistem *database* yang terstruktur dan sistematis seperti mengolah, mengedit, dan menghapus daftar melalui *database*.
- PP (PHP dan Perl) = sebagai bahasa pemrograman yang di dukung. (Siallagan dan Wisnu, 2020).

Berikut adalah tampilan aplikasi XAMPP.



Gambar 2.4 Tampilan XAMPP

Dalam XAMPP, ada dua komponen utama yang sering digunakan oleh pengembang untuk pengembangan aplikasi *web* secara *local* yaitu:

2.5.1.1 MySQL

MySQL merupakan *software* yang tergolong *database server* dan bersifat *open Source*. MySQL adalah sebuah *database management system* (manajemen basis data) menggunakan perintah dasar SQL (*Structured Query Language*) yang cukup terkenal. *Database management system* (DBMS) MySQL multi pengguna dan multi alur ini sudah dipakai lebih dari 6 juta pengguna di seluruh dunia.

MySQL adalah DBMS yang *open source* dengan dua bentuk lisensi, yaitu *free software* (perangkat lunak bebas) dan *shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi MySQL adalah *database server* yang gratis dengan lisensi GNU *General Public License* (GPL) sehingga dapat Anda pakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada. SQL sendiri merupakan suatu bahasa yang dipakai di dalam pengambilan data pada *relational database* atau *database* yang terstruktur. Jadi MySQL adalah *database management system* yang menggunakan bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan *database server*. (Yasin, 2019).



Gambar 2.5 Logo MySQL

2.5.1.2 Apache

Menurut (Kusuma, 2021) *Apache* adalah sebuah nama *web server* yang bertanggung jawab pada *request-response* HTTP dan *logging* informasi secara detail (kegunaan dasarnya). Selain itu, *Apache* juga diartikan sebagai suatu *web server* yang kompak, modular, mengikuti standar protokol HTTP, dan tentu saja sangat digemari. *Apache* dirancang untuk berjalan pada sistem operasi berbasis UNIX, meskipun sekarang juga dapat berjalan pada berbagai *platform* lainnya seperti *Windows* dan *macOS*. Kelebihan utama *Apache* adalah kemampuannya untuk mendukung berbagai fitur yang dapat dikonfigurasi dan disesuaikan sesuai kebutuhan pengguna.

Salah satu fitur yang membuat *Apache* sangat fleksibel adalah modularitasnya. *Apache* dapat diintegrasikan dengan berbagai modul tambahan yang memperluas fungsionalitas dasar *server*. Modul-modul ini termasuk dukungan untuk bahasa pemrograman *server-side* seperti PHP, Perl, dan Python, serta kemampuan untuk menangani berbagai protokol seperti HTTP, HTTPS, FTP, dan masih banyak lagi. Berikut adalah tampilan logo *Apache*.



Gambar 2.6 Logo *Apache*

2.6 Text Editor

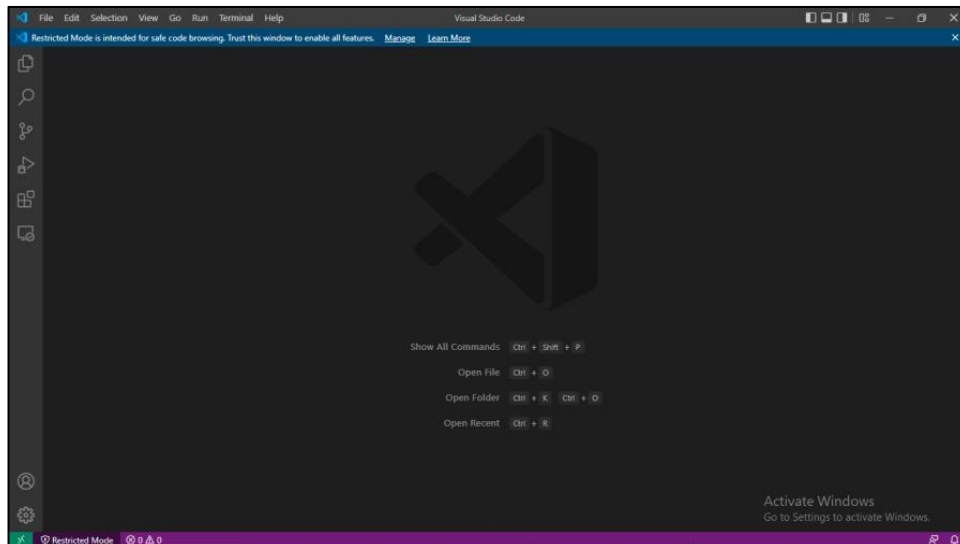
Menurut (Adieb, 2020) *text editor* adalah sebuah perangkat lunak yang memungkinkan pengguna untuk membuka, melihat, dan mengedit *file* teks biasa atau *plain text*. Berbeda dengan *software* pengolah kata seperti Microsoft Word, *text editor* tidak memiliki fitur untuk memformat teks, melainkan fokus pada pengolahan teks murni. Perangkat lunak ini sangat berguna bagi para *programmer* atau *developer* untuk membaca dan menulis kode pemrograman. Salah satu contoh *text editor* yang familiar bagi pengguna Windows adalah Notepad. *Text editor* membantu *programmer* dalam menyusun aplikasi dan membuat *website* secara efektif, berkat berbagai fitur yang disediakan, mulai dari yang sederhana hingga yang dilengkapi dengan plugin tambahan. *Text Editor* adalah sebuah *software* aplikasi atau program komputer yang memungkinkan penggunanya membuat, mengubah, atau mengedit *file* teks. sebuah aplikasi *Text Editor* juga harus dilengkapi dengan fitur pencarian.

Text editor umumnya menawarkan berbagai fitur seperti *syntax highlighting* untuk membedakan elemen-elemen berbeda dalam kode, *auto-completion* untuk memprediksi dan menyelesaikan kode secara otomatis, indentasi untuk mengatur struktur kode, dan navigasi yang memudahkan untuk berpindah-pindah antar bagian teks. Salah satu *text editor* yang digunakan dalam pembuatan *website* ini yaitu:

2.6.1 Visual Studio Code

Visual Studio Code (VS Code) adalah *editor* kode sumber ringan, cepat, dan gratis yang dikembangkan oleh *Microsoft*. *VS Code* dirancang untuk mendukung berbagai bahasa pemrograman dan menyediakan berbagai fitur yang memudahkan pengembangan perangkat lunak. *Visual Studio Code* telah menjadi alat pengembangan *cross-platform* pertama dalam *Microsoft Visual Studio family* yang berjalan di *Windows*, *Linux*, dan *macOS* secara gratis dan *open source* di (<https://github.com/Microsoft/vscode>), aplikasi ini jelas merupakan kode *editor* yang memudahkan untuk mengedit *file* kode dan sistem proyek berbasis *folder* serta menulis *web cross-platform* dan aplikasi *mobile* dengan dukungan

terintegrasi untuk sejumlah besar bahasa dan fitur pengeditan yang kaya seperti *IntelliSense*, *finding symbol references*, *quickly reaching a type definition*, dan banyak lagi. (Sole, 2019:2). Berikut adalah tampilan aplikasi *visual studio code*.



Gambar 2.7 Tampilan *Visual Studio Code*

2.7 UML (Unified Modelling Language)

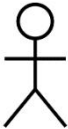
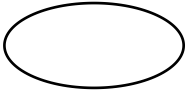


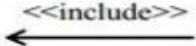
UML (*Unified Modelling Language*) adalah metode pemodelan visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. UML awalnya dibuat oleh *Object Management Group* pada bulan Januari 1997 dengan versi pertama 1.0. UML juga dapat didefinisikan sebagai bahasa standar untuk memvisualisasikan, merancang, dan mendokumentasikan sistem, atau sebagai bahasa standar untuk membuat desain perangkat lunak. UML diharapkan dapat menyederhanakan pengembangan perangkat lunak (RPL) dan memenuhi semua kebutuhan pengguna secara efektif, lengkap dan akurat. Ini termasuk faktor-faktor seperti skalabilitas, ketahanan, dan keamanan.


Unified Modeling Language (UML) adalah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan kebutuhan, melakukan analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Putra dan Andriani, 2019). UML menyediakan standar untuk merancang model suatu sistem. Berikut adalah beberapa jenis diagram UML yang digunakan:

2.7.1 Use Case Diagram

Menurut (Hidayat, et al., 2019) *use case* diagram atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Ini membantu pengembang memahami kebutuhan dan persyaratan pengguna dan memastikan bahwa sistem memenuhi persyaratan ini. Berikut ini adalah simbol – simbol yang digunakan dalam *use case* diagram serta keterangannya seperti yang dijelaskan pada tabel 2.1.

Tabel 2.1 *Use Case Diagram*



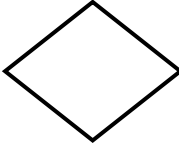

Simbol	Keterangan
	Aktor: Mewakili peran orang, system yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .
	<i>Use case</i> : Abstraksi dan interaksi antar system dan aktor.
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i> .
	Generalisasi: Menunjukkan spesialis actor untuk dapat berpartisipasi dengan <i>use case</i> .
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsional dari <i>use case</i> lainnya.


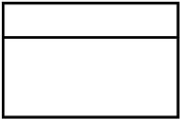
Simbol	Keterangan
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.

2.7.2 Activity Diagram

Menurut (Mare, 2022) *Activity Diagram* merupakan rancangan aliran aktivitas atau aliran kerja dalam sebuah sistem yang akan dijalankan. *Activity Diagram* juga digunakan untuk mendefinisikan atau mengelompokkan aliran tampilan dari sistem tersebut. *Activity Diagram* memiliki komponen dengan bentuk tertentu yang dihubungkan dengan tanda panah. Panah tersebut mengarah ke-urutan aktivitas yang terjadi dari awal hingga akhir. Berikut ini adalah simbol – simbol yang digunakan dalam *activity diagram* serta keterangannya seperti yang dijelaskan pada tabel 2.2.

Tabel 2.2 Activity Diagram

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan system, aktivitas biasanya diawali dengan kata kerja.
	Percabangan/ <i>Decision</i>	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan/ <i>Join</i>	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan menjadi satu.


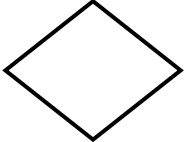
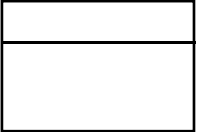




Simbol	Nama	Keterangan
	Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	<i>Swimlane</i>	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

2.7.3 Class Diagram

Menurut (Rumbaugh, et al., 2020) *Class* diagram adalah representasi grafis dari struktur kelas dalam sistem yang menunjukkan kelas, atribut, operasi, dan hubungan antar kelas. *Class* diagram digunakan untuk memodelkan hubungan antar objek dalam sistem dan menggambarkan hierarki serta abstraksi dalam perangkat lunak. *Class* diagram disebut jenis diagram struktur karena menggambarkan apa yang harus ada dalam sistem yang dimodelkan dengan berbagai komponen. Berbagai komponen tersebut dapat mewakili *class* yang akan diprogram, objek utama, atau interaksi antara *class* dan objek.

Menurut (Wulandari dan Nurmiati, 2022) diagram ini mencakup komponen seperti kelas, atribut, metode, dan berbagai jenis hubungan seperti asosiasi, agregasi, komposisi, dan pewarisan. Diagram kelas sangat penting dalam perancangan sistem karena memberikan pandangan terstruktur tentang bagaimana elemen-elemen dalam sistem berinteraksi dan membantu memfasilitasi komunikasi di antara tim pengembang, memastikan desain yang sesuai dengan kebutuhan yang telah diidentifikasi. Berikut ini adalah simbol – simbol yang digunakan dalam *class* diagram serta keterangannya seperti yang dijelaskan pada tabel 2.3.

Tabel 2.3 Class Diagram

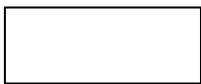
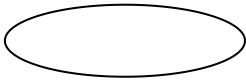
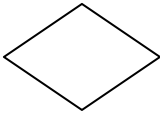

Simbol	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana <i>object</i> anak (<i>descendent</i>) berbagi perilaku dan struktur data dari <i>object</i> yang ada di atasnya <i>object</i> induk (<i>ancestor</i>).
	<i>N-ATX Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 <i>object</i>
	Percabangan/ <i>Decision</i>	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suet hasil yang terukur bagi suet <i>actor</i> .
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suet <i>object</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suet elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Association</i>	Apa yang menghubungkan antara <i>object</i> satu dengan <i>object</i>

Simbol	Nama	Keterangan
		lainnya.

2.8 ERD (Entity Relationship Diagram)

Menurut (Akbar dan Haryanti, 2021) *Entity Relationship Diagram* (ERD) adalah suatu diagram yang digunakan untuk merancang suatu basis data, dipergunakan untuk memperlihatkan hubungan atau relasi antar entitas atau objek yang terlihat beserta atributnya. Dengan kata lain, ERD menjadi suatu model untuk menjelaskan hubungan antardata dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antarrelasi. Berikut ini adalah simbol – simbol yang digunakan dalam ERD serta keterangannya seperti yang dijelaskan pada tabel 2.4.

Tabel 2.4 *Entity Relationship Diagram* (ERD)

No	Simbol	Keterangan
1		Entitas
2		Atribut
3		Hubungan
4		Garis

1) Entitas

Objek dalam dunia nyata yang dapat dibedakan dengan objek lain. Entitas terdiri atas beberapa atribut mengidentifikasi atau membedakan yang satu dengan yang lainnya. Pada setiap entitas baru harus memiliki 1 atribut unik atau yang disebut dengan *primary key*.

2) Atribut Isi dari atribut mempunyai elemen yang dapat mengidentifikasi isi elemen satu dengan yang lain. Ada dua jenis atribut, yaitu:

- a. *Identifier (key)* digunakan untuk menentukan suatu *entity* secara unik (*primary key*).
- b. *Descriptor (nonkey attribute)* digunakan untuk menspesifikasi karakteristik dari suatu *entity* yang tidak unik.

3) Kardinalitas

Menyatakan jumlah himpunan relasi antar entitas. Pemetaan kardinalitas terdiri dari:

- a. *One-to-one*, sebuah entitas pada A berhubungan dengan entitas B paling banyak.
- b. *One-to-many*, sebuah entitas pada A berhubungan dengan entitas B lebih dari satu.
- c. *Many-to-many*, sebuah entitas pada A berhubungan dengan entitas B lebih dari satu dan entitas B berhubungan dengan entitas A lebih dari satu juga.

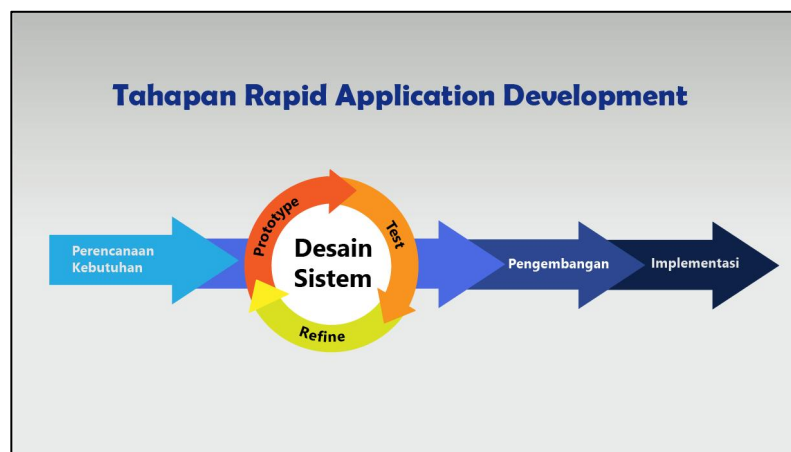
2.8 Framework Pengembangan Perangkat Lunak

2.8.1 RAD (Rapid Application Development)

Menurut (Aini, et al., 2019), metode *Rapid Application Development* (RAD) adalah pendekatan yang digunakan dalam pengembangan perangkat lunak dengan siklus pengembangan yang relatif singkat.

Rapid Application Development (RAD) menonjol dengan pendekatannya yang memungkinkan pengembangan cepat dan fleksibilitas tinggi. Fokus pada siklus pengembangan yang cepat dan *prototyping* memungkinkan tim untuk mengubah desain dan persyaratan selama proses berjalan, mengurangi risiko

dengan umpan balik cepat dari klien. Keterlibatan aktif klien sepanjang proses memastikan bahwa produk yang dihasilkan sesuai dengan kebutuhan. Namun, RAD juga memiliki kekurangan, seperti kebutuhan akan komitmen tinggi dari klien dan tidak cocok untuk proyek skala besar dan sangat kompleks. Keterbatasan desain awal dan ketergantungan pada tim yang sangat terampil juga merupakan tantangan.



Gambar 2.8 Diagram RAD

2.8.2 Metode PIECES

Menurut (Fardela, 2023) PIECES adalah suatu cara untuk mengoreksi atau memperbaiki sistem informasi bagi pengambil keputusan dalam suatu organisasi. Metode PIECES adalah metode analisis sebagai dasar untuk memperoleh pokok-pokok permasalahan yang lebih spesifik. Dalam menganalisis sebuah sistem, biasanya akan dilakukan terhadap beberapa aspek antara lain adalah kinerja, informasi, ekonomi, keamanan aplikasi, efisiensi dan pelayanan pelanggan. Analisis ini disebut dengan PIECES Analysis (Performance, Information, Economy, Control, Efficiency and Service). Metode PIECES mencakup enam variabel yang digunakan untuk menganalisis sistem informasi, yaitu:

1. *Performance* (Keandalan): Variabel ini digunakan untuk mengevaluasi kinerja sebuah sistem, termasuk apakah sistem berjalan dengan baik atau tidak. Kinerja ini dapat diukur dari jumlah data yang ditemukan dan seberapa cepat data tersebut dapat ditemukan.

2. *Information and Data* (Informasi dan Data): Variabel ini digunakan untuk menganalisis jumlah dan kejelasan informasi yang dihasilkan dari suatu pencarian data.
3. *Economics* (Nilai Ekonomis): Variabel ini digunakan untuk menganalisis apakah suatu sistem layak diterapkan dari segi finansial dan biaya yang dikeluarkan, karena biaya yang dikeluarkan mempengaruhi sistem tersebut.
4. *Control and Security* (Pengendalian dan Pengamanan): Variabel ini mengevaluasi sejauh mana pengawasan dan kontrol dilakukan untuk memastikan sistem berjalan dengan baik.
5. *Efficiency* (Efisiensi): Variabel ini digunakan untuk mengevaluasi efisiensi dan efektivitas kinerja sistem, terutama dalam hal otomasi.
6. *Service* (Pelayanan): Variabel ini mengevaluasi kualitas pelayanan yang diberikan oleh sistem, serta mengidentifikasi masalah terkait pelayanan yang ada.

Dengan analisis berdasarkan variabel-variabel ini, metode PIECES membantu dalam memperbaiki dan mengembangkan sistem informasi yang lebih baik.

2.8.3 Metode Blackbox testing

Menurut (Amei, et al., 2021) *Black box testing* adalah metode pengujian perangkat lunak yang terutama memeriksa spesifikasi perangkat lunak yang sedang dikembangkan. Pengujian *black box* dapat menemukan beberapa hal seperti fungsionalitas yang salah atau tidak ada, kesalahan struktur data, kesalahan akses basis data, kesalahan antarmuka, kesalahan kinerja dan inisialisasi, dan akhir. Keuntungan penggunaan metode *blackbox testing* adalah:

1. Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu.
2. Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan.
3. *Programmer* dan *tester* keduanya saling bergantung satu sama lain.