

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Bab ini menguraikan pandangan singkat mengenai penelitian sebelumnya yang terkait terhadap penelitian yang akan dibuat. Tujuan acuan ini adalah memperluas teori dalam penelitian yang akan dilakukan.

Penelitian terdahulu yang berkaitan dengan studi ini yaitu membahas tentang sistem informasi. Penelitian ini bertujuan membuat sistem informasi akademik untuk pengelolaan nilai dan jadwal siswa pada SMA Muhammadiyah 4, dengan menggunakan metode *Agile Software Development* dan model *Scrum*. Hasilnya, akan memudahkan guru dan siswa mengakses nilai ujian dan rapot secara online [3].

Ninda Lutfiani, Eka Purnama Harahap, Qurotul Aini, Alfian Dimas Ahsanul Rizki Ahmad, dan Untung Rahardja meneliti "Inovasi Manajemen Proyek I-Learning Menggunakan Metode *Agile Scrum*" yang bertujuan membuat manajemen proyek iLearning dengan metode yang efektif. Hasil penelitian menunjukkan bahwa metode tersebut memang efektif untuk manajemen proyek iLearning [4].

Penelitian selanjutnya melakukan penelitian dalam rangka meningkatkan efisiensi pelayanan kawasan dan pusat, penelitian terbaru telah merancang sebuah sistem informasi layanan berbasis *Scrum*. Tujuannya adalah untuk membantu pengembangan SDM di kawasan dan pusat dengan menyediakan perangkat lunak yang mudah digunakan oleh pegawai. Pengelola kawasan dan pusat dapat memperoleh informasi yang akurat dan terintegrasi, serta memantau kinerja SDM secara efektif. Perangkat lunak yang dibuat dalam penelitian ini juga dapat mempermudah koordinasi dan komunikasi antara pegawai di kawasan dan pusat [5].

Mohammad Fuad Farisi, Budi Praptono, dan Maria Dellarosawati Idawicaksati mempublikasikan penelitian "Perancangan Aplikasi Pencatatan Keuangan Studi Kasus pada Amanah Laundry Bogor Menggunakan *Agile Development Methods*" yang bertujuan membantu bagian keuangan dalam membuat laporan keuangan dengan aplikasi web yang dikembangkan dengan metode *Agile development*. Hasil dari penelitian ini adalah aplikasi pencatatan pemasukan dan pengeluaran serta rekap data keuangan [6].

Dedi Setiadi mempublikasikan penelitian "Sistem Informasi Keuangan Investasi Reksadana Berbasis Aplikasi Web" yang bertujuan memberikan literasi keuangan melalui aplikasi perencanaan keuangan berbasis *progressive web apps* dengan metode *Agile scrum*. Hasilnya adalah aplikasi yang membantu investor melakukan transaksi reksadana secara online, mudah digunakan, dan membantu literasi keuangan [7].

I Gusti Ngurah Suryantara, Ricky Raymond, dan Dewi Candani Sulaiman mempublikasikan penelitian "Pengembangan Aplikasi Penjualan Mobil dengan Framework *Scrum* pada PT XYZ" yang menggunakan kajian kualitatif dan metode *Agile scrum* untuk memahami implementasi arsitektur bisnis dalam pengembangan aplikasi penjualan mobil. Hasilnya adalah wawasan mengenai proses bisnis penjualan mobil [8].

Karya Hasya Sabila, Budi Praptono, dan Isnaeni Yuli Arini, berjudul "Perancangan Aplikasi Pencatatan Laporan Keuangan Menggunakan Metode *Agile*" menunjukkan solusi bagi UMKM baru, Kedai Intikopi, yang mengalami masalah dalam pengelolaan keuangan. Sistem manual yang digunakan sebelumnya menghasilkan laporan yang tidak akurat. Studi ini menggunakan metodologi *Agile* dan hasilnya menunjukkan bahwa aplikasi yang dibuat memenuhi kebutuhan pengguna dan memiliki fitur untuk menambahkan catatan pemasukan dan pengeluaran yang saling berhubungan sehingga laporan keuangan dapat dicetak secara otomatis [9].

Berdasarkan tinjauan pustaka ini menyatakan bahwa peneliti melakukan tinjauan pustaka sebagai referensi dalam menentukan metode

dan memperoleh pengetahuan mengenai metode yang digunakan. Peneliti menggunakan jurnal tabel sebagai bahan acuan dan menggunakan beberapa jurnal lain untuk membantu dalam desain sistem yang akan dibangun. Peneliti berupaya memperbaiki hasil melalui perubahan pada kinerja yang dirasa perlu.

Tabel 2. 1 Penelitian Sebelumnya

No	Judul	Masalah	Metode	Hasil	Perbedaan
1	Inovasi Manajemen Proyek ILearning Menggunakan Metode <i>Agile Scrum</i> (2020)	Tentang diskusi mengenai manajemen proyek iLearning yang berlangsung dengan baik belum banyak dilakukan.	<i>Agile Scrum</i> (Metode manajemen yang menggabungkan <i>scrum</i> dan Kanban)	Hasil penelitian menunjukkan bahwa penggunaan metode <i>Agile</i> pada manajemen proyek iLearning menjadi lebih efektif.	Penelitian terdahulu menggunakan metode <i>Agile scrum</i> pada manajemen proyek i-learning sedangkan penelitian oleh peneliti akan menggunakan metode <i>scrum</i> pada laporan keuangan Kedai Kontekstual Kopi.
2	Penerapan Metode <i>Scrum</i> dalam Pengembangan Sistem Informasi Layanan Kawasan (2020)	Keterbatasan sumber daya manusia (SDM) di LIPI membutuhkan redistribusi pegawai untuk memperbaiki hal ini.	<i>Agile Scrum</i>	Dengan Sistem Informasi Layanan Pegawai, masalah akibat redistribusi pegawai bisa terpecahkan.	Penelitian terdahulu menggunakan metode <i>scrum</i> dalam pengembangan sistem informasi layanan karyawan sedangkan penelitian oleh peneliti akan menggunakan metode <i>scrum</i> pada laporan keuangan Kedai Kontekstual Kopi.

No	Judul	Masalah	Metode	Hasil	Perbedaan
3	Perancangan Aplikasi Pencatatan Keuangan Studi Kasus Pada Amanah <i>Laundry</i> Bogor Menggunakan <i>Agile Development Methods</i> (2020)	Pencatatan keuangan manual membuat proses tidak efisien dan menghabiskan banyak waktu.	<i>Agile Development Methods</i>	Aplikasi keuangan memiliki fitur untuk mencatat pemasukan, pengeluaran dan merekap data keuangan.	Penelitian terdahulu menggunakan metode <i>Agile development methods</i> pada perancangan pencatatan keuangan amanah <i>laundry</i> bogor sedangkan penelitian oleh peneliti akan menggunakan metode <i>scrum</i> pada laporan keuangan Kedai Kontekstual Kopi.
4	Sistem Informasi Keuangan Investasi Reksadana Berbasis Aplikasi <i>Web</i> (2021)	Masyarakat Indonesia belum begitu paham tentang industri investasi seperti pasar modal.	<i>Agile Scrum</i>	Aplikasi perencanaan keuangan membantu dengan memasukkan instrumen investasi seperti reksadana.	Penelitian terdahulu menggunakan metode <i>Agile scrum</i> pada sistem informasi keuangan investasi reksadana sedangkan penelitian oleh peneliti akan menggunakan metode <i>scrum</i> pada laporan keuangan Kedai Kontekstual Kopi.

No	Judul	Masalah	Metode	Hasil	Perbedaan
5	Pengembangan Aplikasi Penjualan Mobil dengan Framework <i>Scrum</i> pada PT XYZ (2021)	Perusahaan harus membuat rencana yang matang untuk membangun aplikasi terintegrasi agar bisa berinvestasi dengan benar dalam teknologi informasi.	<i>Agile Scrum</i>	Hasil dari penelitian ini akan memberikan pandangan tentang proses bisnis penjualan mobil.	Penelitian terdahulu menggunakan metode <i>Agile scrum</i> pada pengembangan aplikasi penjualan mobil pada PT XYZ sedangkan penelitian oleh peneliti akan menggunakan metode <i>scrum</i> pada laporan keuangan Kedai Kontekstual Kopi
6	Perancangan Aplikasi Pencatatan Laporan Keuangan Dengan Menggunakan Metode <i>Agile</i> (2021)	Bagian keuangan masih memiliki kerugian karena menggunakan metode manual, sehingga laporan yang dihasilkan tidak selalu akurat	<i>Agile</i>	Hasil pengujian menunjukkan bahwa aplikasi sudah cocok dengan kebutuhan manajemen keuangan dan memiliki fitur tambahpemasukan/pengeluaran yang terintegrasi untuk laporan otomatis.	Penelitian terdahulu menggunakan metode <i>Agile</i> pada laporan keuangan Kedai Intikopi sedangkan penelitian oleh peneliti akan menggunakan metode <i>scrum</i> pada laporan keuangan Kedai Kontekstual Kopi.

No	Judul	Masalah	Metode	Hasil	Perbedaan
7	Penerapan Metode <i>Agile Scrum</i> Pada Rancangan SisIAM4 Penerapan Metode <i>Agile Scrum</i> Pada Rancangan SisIAM4 (2022)	Penelitian ini membuka solusi baru bagi guru dan siswa SMA Muhammadiyah 4, yaitu aplikasi pengelolaan nilai ujian dan rapot secara online.	<i>Agile Scrum</i>	Hasil dari penelitian ini dapat mempermudah pengelolaan dan mempermudah siswa dalam mengetahui hasil nilai.	Penelitian terdahulu menggunakan metode <i>Agile scrum</i> pada rancangan SisIAM4 sedangkan penelitian oleh peneliti akan menggunakan metode <i>scrum</i> pada laporan keuangan Kedai Kontekstual Kopi.

2.2 Dasar Teori

2.2.1 Laporan Keuangan

Laporan Keuangan merangkum semua transaksi keuangan yang terjadi selama satu tahun buku, yang dicatat dalam proses pencatatan. Interpretasi Laporan Keuangan merupakan bagian penting dari proses pelaporan keuangan, dengan mengacu pada Standar Akuntansi Keuangan. Laporan Keuangan yang lengkap terdiri dari Neraca, Laporan Perubahan Posisi Keuangan (dalam bentuk Laporan Aliran Kas atau Laporan Aliran Dana), Schedul, dan informasi tambahan lainnya. Melalui laporan keuangan, informasi penting disajikan bagi pihak internal dan eksternal, sebagai tujuan utama dalam proses pelaporan keuangan. Beberapa tujuan dari laporan keuangan adalah: [10]

1. Untuk memberikan gambaran detail mengenai jumlah dan jenis aset yang dimiliki oleh perusahaan pada saat ini.
2. Untuk menunjukkan jenis dan jumlah utang serta modal perusahaan pada saat ini.
3. Memberikan informasi rinci tentang jenis dan jumlah pendapatan yang diperoleh oleh perusahaan dalam periode tertentu.
4. Menyajikan informasi tentang jumlah dan jenis pengeluaran perusahaan pada periode tersebut.
5. Untuk menampilkan hasil kinerja manajemen perusahaan dalam periode tertentu.
6. Laporan keuangan juga menyediakan informasi yang terdapat dalam catatan-catatan di dalamnya.

2.2.2 Website

Perkembangan teknologi dan data yang sangat cepat mempengaruhi pandangan masyarakat tentang teknologi informasi. Banyak orang menggunakan *website* sebagai sarana untuk memperkenalkan bisnis atau sebagai sumber informasi. *Website* memiliki beberapa elemen seperti teks, foto, suara, dan animasi, sehingga memberikan informasi yang menarik bagi

pengguna. Proses pengumpulan, pengelolaan, dan penyediaan informasi dilakukan oleh teknologi *website* untuk mempermudah akses. Keunggulan *website* sebagai salah satu layanan internet adalah karena kemampuannya menyajikan informasi secara interaktif melalui teks, gambar, suara, dan video. *Website* adalah layanan internet paling banyak digunakan [11].

2.2.3 Metode Pengembangan Perangkat Lunak

Peneliti akan menggunakan metode *agile* sebagai metodenya dalam pengembangan perangkat lunak.

2.2.3.1 Metode *Agile software development*

Agile software development memiliki filosofi yang menekankan pada kecepatan dan fleksibilitas. Ia merupakan metodologi pengembangan perangkat lunak yang memprioritaskan adaptasi terhadap perubahan dan memberikan lebih banyak kontrol kepada tim pengembangan. Pengembangan perangkat lunak berbasis *Agile* memiliki berbagai model, antara lain *Extreme Programming*, *Adaptive Software Development*, *Dynamic Systems Development Method*, dan *Model Scrum*, masing-masing memiliki pendekatan dan cara unik dalam menjalankan proses pengembangan [12].

2.2.3.2 *Extreme Programming*

Extreme programming memiliki konsep pengembangan yang bersifat *Agile*, dengan fokus pada peningkatan efisiensi dan kualitas pengembangan perangkat lunak. Metode ini memiliki tahapan terstruktur dalam proses pengembangan, mulai dari eksplorasi, perencanaan, literasi pengembangan, produksi, hingga pemeliharaan. Tujuan dari konsep ini adalah untuk mengurangi kesalahan dan memastikan kebutuhan perangkat lunak terpenuhi melalui partisipasi pengguna [13].

2.2.3.3 *Adaptive Software Development*

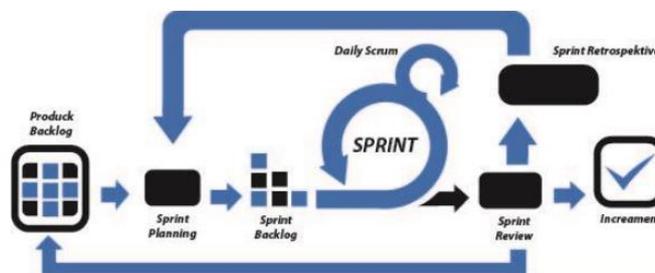
Adaptive software development adalah salah satu metodologi dari sekian banyak metodologi *Agile development*. *Adaptive software development* memfokuskan pada pengembangan perangkat lunak jangka pendek yang membutuhkan kemampuan untuk beradaptasi dengan cepat.

Proses *Adaptive software development* terdiri dari tahapan *speculation*, *collaboration*, dan *learning* [14].

2.2.3.4 *Dynamic Systems Development Method*

Pengembangan perangkat lunak berbasis *Dynamic Systems Development Method* (DSDM) memprioritaskan partisipasi pengguna dan respon terhadap perubahan, dengan tujuan untuk membangun perangkat lunak yang sesuai dengan kebutuhan bisnis secara efisien dari segi waktu dan anggaran. Model DSDM terdiri dari tiga tahap, yaitu tahap sebelum proyek, siklus hidup proyek, dan tahap setelah proyek. Salah satu kerangka kerja DSDM yang penting adalah *Rapid Application Development* (RAD), yang memperkenalkan pendekatan iteratif dan kolaboratif antara pengguna dan tim pengembang. Dengan menerapkan prinsip-prinsip DSDM, tim pengembang dapat menghasilkan perangkat lunak yang lebih terfokus pada kebutuhan pengguna, fleksibel, dan sesuai dengan batasan waktu dan anggaran yang telah ditetapkan [15].

2.2.3.5 *Model Scrum*



Gambar 2. 1 Tahapan *Scrum* Model

Scrum merupakan metode pengembangan perangkat lunak yang berkonsentrasi pada kecepatan, fleksibilitas, dan strategi. Proses pengembangan dibagi menjadi unit-unit yang saling mendukung. Metode *scrum* memiliki beberapa tahapan, seperti pembuatan *Product Backlog*, *sprint planning*, *sprint backlog*, *daily scrum*, *sprint review*, dan *sprint retrospective*. Tahap-tahapnya meliputi: mengumpulkan kebutuhan aplikasi pada *Product Backlog*, merencanakan dan menetapkan waktu serta menyusun backlog pada *sprint planning*, melaksanakan eksekusi backlog

pada *sprint* backlog, melakukan sinkronisasi dan monitoring kinerja pada *daily scrum*, meninjau kembali aktivitas pada *sprint review*, dan memberikan kritik dan saran pada *sprint retrospective* [16].

2.2.4 Bahasa Pemrograman

Bahasa pemrograman merupakan intruksi untuk memerintah komputer. Bahasa pemrograman memiliki beberapa macam yaitu :

2.2.4.1 HTML

Singkatan *Hyper Text Markup Language* (HTML) menunjukkan bahasa yang memberikan arahan bagi peramban web dalam penyusunan dan tampilan konten web. HTML adalah bahasa pemrograman visual yang memformat halaman web dengan tag, yang berisi info yang akan ditampilkan dan memformat informasi itu [17].

2.2.4.2 CSS

CSS adalah bahasa pemrograman yang memungkinkan pengaturan visual yang lebih baik daripada tag standar HTML. Dengan menggunakan CSS, tampilan visual beberapa dokumen dapat dikendalikan bersamaan melalui sekumpulan atribut pemformatan. Keuntungan menggunakan CSS adalah dalam melakukan perubahan format, semua dokumen dapat diperbaiki sekaligus, bukan per dokumen [18].

2.2.4.3 Javascript

Javascript, sebagai bahasa skrip yang digunakan di sisi klien atau browser, memberikan kemampuan untuk memanipulasi HTML dan CSS. Hal ini memungkinkan pengembang web untuk meningkatkan fitur situs web dengan cara yang lebih interaktif, animatif, dan dinamis. Misalnya, *Javascript* dapat digunakan untuk validasi formulir, mengirim dan menerima data dari *server*, dan memberikan *feedback real-time* kepada pengguna. Namun, keberhasilan penggunaan *Javascript* dalam meningkatkan fitur situs web tergantung pada kompatibilitasnya dengan berbagai jenis *browser* yang ada. Oleh karena itu, pengujian dan

memastikan kompatibilitas *Javascript* pada berbagai *browser* sangat penting untuk memastikan kinerja yang optimal pada situs web [19].

2.2.4.4 PHP

PHP adalah bahasa *scripting server-side* yang populer untuk membuat situs web dinamis. Dengan fleksibilitasnya, PHP dapat berjalan pada berbagai sistem operasi seperti Windows, Linux, dan Mac OS, serta pada berbagai jenis web server selain Apache. PHP memudahkan pengembangan situs web dengan menggunakan *database*, dengan dukungan sistem manajemen *database* seperti MySQL, Oracle, Microsoft Access, dan PostgreSQL. Dengan PHP, pengguna dapat membuat situs web yang menarik dan interaktif, serta memungkinkan fitur seperti validasi form, otentikasi pengguna, dan interaksi dengan *server* [20].

2.2.5 Perancangan Basis Data

Perancangan basis data merupakan proses penting yang dilakukan sebelum membangun suatu sistem informasi. Hal yang penting dalam pembuatan sistem adalah memastikan konsistensi, keandalan, dan efisiensi data. Perancangan basis data yang digunakan yaitu :

2.2.5.1 MySQL

MySQL adalah sistem manajemen *database* yang didasarkan pada bahasa SQL untuk mengelola data dan informasi. SQL, yang merupakan singkatan dari *Structured Query Language*, pertama kali didefinisikan oleh *American National Standards Institute* (ANSI) pada tahun 1986 dan memiliki konsep khusus untuk pengolahan *database*, SQL menjadi dasar dari sistem manajemen *database* MySQL. Sumber terbuka, MySQL bersifat relational sehingga data dikelola dalam beberapa tabel terpisah untuk mempermudah manipulasi. Baik untuk *database* kecil maupun besar, MySQL merupakan pilihan yang populer dan efisien [21].

2.2.5.2 Unified Modeling Language

UML atau *Unified Modeling Language* merupakan bahasa visual standar yang digunakan dalam pengembangan perangkat lunak berbasis objek. UML dapat membantu para pengembang perangkat lunak dalam

memodelkan sistem perangkat lunak yang kompleks dengan cara yang lebih mudah dipahami. Selain itu, UML juga menyediakan berbagai jenis diagram yang dapat digunakan dalam proses perancangan perangkat lunak, seperti diagram *use case*, *class diagram*, *activity diagram*, *sequence diagram*, dan masih banyak lagi [22]. Dalam penggunaannya, pengembang perangkat lunak memilih diagram yang tepat dan sesuai dengan kebutuhan perangkat lunak yang akan dikembangkan, yaitu:

1. *Use case diagram*

Use case Diagram menggambarkan interaksi antara aktor dan sistem dalam sistem informasi serta memberikan pandangan eksternal terhadap sistem dan interaksinya dengan dunia luar [23]. Berikut simbol pada *use case diagram* yang ditunjukkan pada Tabel 2.2 [24].

Tabel 2. 2 Simbol *use case diagram*

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Subjek di luar sistem yang berinteraksi dengan sistem yang dibangun
2		<i>Dependency</i>	Hubungan antar elemen dimana elemen independen mempengaruhi elemen lain yang tidak mandiri
3		<i>Use case</i>	Unit yang saling bertukar pesan dan dinyatakan dengan frase yang mengandung kata kerja

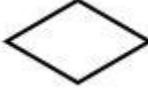
4		<i>System</i>	Spesifikasi yang menunjukkan sistem dengan batasan tertentu
5		<i>Assosiasi</i>	Komunikasi antara aktor dan <i>use case</i> dalam suatu <i>use case</i>
6		<i>Include</i>	Penyertaan <i>use case</i> dalam <i>use case</i> lain
7		<i>Generalization</i>	Hubungan umum khusus antara dua <i>use case</i>
8		<i>Extend</i>	Perluasan perilaku <i>use case</i> target dari sumber <i>use case</i>
9		<i>Collaboration</i>	Kerja sama elemen dan aturan untuk mencapai tujuan yang lebih besar
10		<i>Note</i>	Sumber daya komputasi yang mencerminkan elemen aplikasi saat dijalankan

2. Activity Diagram

Activity Diagram adalah jenis diagram UML yang digunakan untuk menggambarkan alur kerja atau proses dalam objek atau sistem. Diagram ini menampilkan urutan aktivitas atau tindakan yang harus dilakukan untuk mencapai tujuan tertentu, dengan alur proses yang terstruktur dari titik awal hingga akhir. Aktivitas tersebut dapat berupa tugas, pengambilan

keputusan, atau komunikasi dengan sistem lain. *Activity Diagram* membantu pengembang perangkat lunak untuk memahami dan mengorganisir proses bisnis secara visual, sehingga dapat meningkatkan efisiensi dan kualitas pengembangan perangkat lunak [25]. Berikut simbol pada *Activity Diagram* yang ditunjukkan pada Tabel 2.3 [24].

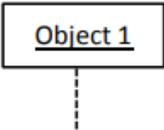
Tabel 2. 3 Simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1		Status Awal	Awal dari aktivitas sistem ditunjukkan dalam diagram aktivitas
2		Aktivitas	Aktivitas dilakukan oleh sistem dan diawali dengan kata kerja
3		Percabangan/ <i>Decision</i>	Kondisi dimana ada pilihan aktivitas lebih dari satu
4		Penggabungan/ <i>Join</i>	Proses penggabungan lebih dari satu aktivitas menjadi satu
5		Status akhir	Akhir dari aktivitas sistem ditunjukkan dalam diagram aktivitas
6		<i>Swimlane</i>	Organisasi bisnis yang memisahkan tanggung jawab atas aktivitas yang terjadi

3. *Sequence Diagram*

Diagram Urutan atau *Sequence Diagram* menampilkan urutan interaksi objek dalam sistem dalam rentang waktu. Termasuk pesan yang diterima antara pengguna, tampilan, dan lainnya. Berkaitan dengan skenario tindakan yang terjadi sebagai respons terhadap suatu peristiwa dan memunculkan *output* yang sesuai. Menjelaskan apa yang memicu aktivitas, proses internal dan *output* yang dihasilkan [26]. Berikut simbol pada *Sequence Diagram* yang ditunjukkan pada Tabel 2.4 [27].

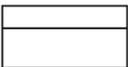
Tabel 2. 4 Simbol *Sequence Diagram*

No	Gambar	Nama	Keterangan
1		Objek/Aktor	Objek Berasal dari Kelas. Sebuah entitas yang dapat diberi nama dengan kelas atau objek. Aktor termasuk objek. Garis putus-putus menggambarkan jangka hidup suatu objek.
2		Aktivasi	Menandakan Jangka Hidup Objek.
3		Pesan	Komunikasi antar objek. Suatu objek dapat mengirimkan pesan ke objek lain. Interaksi antar objek ditampilkan melalui operasi pada diagram kelas
4		<i>Return</i>	Balasan dari Komunikasi Objek.

4. *Class diagram*

Class diagram dalam UML memiliki peran kunci dalam pemodelan. Tujuannya adalah untuk membuat model logika dari suatu sistem. Class diagram menunjukkan skema arsitektur dari sistem yang sedang diteliti. Terdiri dari class, atribut, dan method, setiap class dihubungkan oleh garis bernama asosiasi [25]. Berikut simbol pada *class diagram* yang ditunjukkan pada Tabel 2.5 [27].

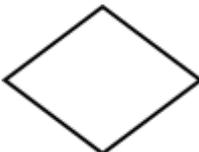
Tabel 2. 5 Simbol *Class diagram*

No	Gambar	Nama	Keterangan
1		<i>Class</i>	Struktur Sistem yang Terdiri dari Kelas.
2		<i>Interface</i>	Konsep yang Sama dengan Interface pada Pemrograman Berorientasi Objek.
3		Asosiasi	Hubungan Umum Antara Kelas dengan Multiplisitas.
4		Asosiasi berarah	Kelas Satu yang Digunakan Oleh Kelas Lain dengan Multiplisitas.
5		<i>Generalization</i>	Hubungan Generalisasi dan Spesialisasi (Umum Khusus) Antara Kelas.
6		<i>Dependency</i>	Ketergantungan Antar Kelas.
7		Agresiasi	Hubungan Antar Kelas sebagai Bagian Utuh.

2.2.5.3 Flowchart

Flowchart adalah sebuah diagram yang menggunakan simbol-simbol untuk merepresentasikan langkah-langkah dan hubungan antar langkah dalam suatu proses atau program [28]. Berikut simbol pada *flowchart* yang ditunjukkan pada Tabel 2.6 [29].

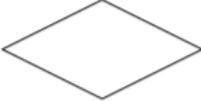
Tabel 2. 6 Simbol *flowchart*

No	Simbol	Nama	Keterangan
1		<i>Terminator</i>	Awal atau akhir dari program.
2		Garis alir	Aliran program.
3		Preparation	Penentuan nilai awal.
4		Proses	Pengolahan data.
5		<i>Input/output</i> data	Masukan/keluaran data, parameter, informasi.
6		<i>Decision</i>	Perbandingan pernyataan dan penentuan pilihan untuk tindakan selanjutnya.
7		<i>On Page connector</i>	Sambung antar bagian flowchart pada halaman yang sama.
8		<i>Off Page connector</i>	Sambung antar bagian flowchart pada halaman yang berbeda.

2.2.5.4 Entity Relationship Diagram

Entity Relationship Diagram (ERD) memiliki tata letak dari atas ke bawah dalam merancang basis data. ERD digunakan sebagai alat untuk menentukan entitas penting dan menjelaskan hubungan antara entitas [30]. Simbol pada ERD ditunjukkan dalam Tabel 2.7 [31].

Tabel 2. 7 Simbol *entity relationship diagram*

No	Simbol	Nama	Keterangan
1		<i>Entity</i>	Koleksi objek yang dapat dikenali dan didefinisikan secara unik.
2		Atribut	<i>Properties</i> dari <i>entity</i> atau relasi yang memberikan deskripsi lebih rinci.
3		<i>Relationship</i>	Hubungan antar satu atau lebih <i>entity</i> .
4		<i>Link</i>	Penghubung antar himpunan, relasi dan atribut dari <i>entity</i> .

2.2.6 Web Server

Peneliti menggunakan software web server XAMPP yang memberikan layanan data.

2.2.6.1 XAMPP

XAMPP sebagai solusi *open source* berbasis PHP menggabungkan beberapa *software* dalam satu paket. XAMPP terdiri dari Apache, MySQL, PHP, *FileZilla FTP Server*, *phpMyAdmin*, dll. Fungsinya sebagai *server* lokal yang terdiri dari Apache HTTP *Server*, *database* MySQL, dan PHP *interpreter* [32].

2.2.7 Pengujian Sistem

Pengujian *black box* dipakai oleh peneliti sebagai pengujian sistem untuk memverifikasi kebutuhan perangkat lunak, melalui proses pengujian secara manual.

2.2.8.1 Black Box

Black box testing merupakan sebuah metode pengujian perangkat lunak yang berfokus pada fungsi dan fitur yang diinginkan oleh pengguna. Pada metode pengujian ini, pengujian dilakukan dari luar sistem tanpa memperhatikan bagaimana program tersebut dirancang atau dikodekan. Tujuan dari *black box testing* adalah untuk mengetahui apakah sistem mampu melakukan tugas yang diminta oleh pengguna secara efektif dan efisien, serta memastikan bahwa kesalahan dan kegagalan pada sistem dapat diidentifikasi dan diperbaiki. Metode pengujian ini sangat penting dalam proses pengembangan perangkat lunak untuk memastikan bahwa produk akhir yang dihasilkan dapat berfungsi sesuai dengan spesifikasi yang ditetapkan [33].