

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Dalam melakukan penelitian ini, penulis melakukan tinjauan Pustaka terhadap penelitian-penelitian terdahulu sebagai penunjang penelitian yang akan dilakukan oleh penulis. Di bawah ini adalah tinjauan literatur yang telah diteliti sebelumnya :

**Tabel 2.1 Daftar Literatur**

No	Detail Jurnal	
1	Judul	Pengembangan Sistem Informasi Penerimaan Siswa Baru Berbasis Web (Studi Kasus : MI Bani Hasyim Desa Lengkong Kecamatan Cerme)
	Penerbit	Jurnal Sistem Informasi dan Telematika (Telekomunikasi, Multimedia dan Informatika)
	Penulis	(Firmansyah & Chotijah)
	Tahun	2022
	Permasalahan	Sistem penerimaan siswa baru di sekolah dasar MI Bani Hasyim masih dilakukan secara manual, di mana calon siswa dan wali siswa harus datang langsung ke MI Bani Hasyim untuk mengisi formulir pendaftaran. Melalui sistem ini, proses penerimaan siswa belum efektif dan efisien karena memerlukan waktu dan biaya yang tidak sedikit
	Tujuan Penelitian	Membuat sistem informasi pendaftaran siswa baru berbasis web
	Metode Penelitian	Metode <i>Waterfall</i>
	Hasil Penelitian	Sistem informasi pendaftaran baru berbasis web yang diharapkan akan memberi kemudahan bagi <i>stakeholder</i> yang terlibat dalam penerimaan calon siswa, panitia penerimaan siswa baru dan yang lainnya

No	Detail Jurnal	
2	Judul	Perancangan Sistem Informasi Penerimaan Siswa Baru Berbasis Web Pada SMK Bandara
	Penerbit	Jurnal PROSISKO
	Penulis	(Septiarina et al., 2021)
	Tahun	2021
	Permasalahan	Sistem informasi penerimaan siswa baru pada SMK Bandara selama ini masih dilakukan secara manual dan belum terkomputerisasi
	Tujuan Penelitian	Membuat sistem informasi untuk pembuatan web tentang penerimaan siswa baru pada SMK Bandara
	Metode Penelitian	Metode <i>Prototype</i>
	Hasil Penelitian	Sistem informasi penerimaan calon siswa berbasis web yang dapat membantu pihak sekolah dan calon siswa untuk mendaftarkan diri di mana pun, kapan pun dan tidak perlu datang ke sekolah
3	Judul	Sistem Informasi Penerimaan Siswa Baru Pada SMK Bekasi Berbasis <i>Website</i>
	Penerbit	Bina Insani ICT <i>Journal</i>
	Penulis	(Putra & Shadiq)
	Tahun	2020
	Permasalahan	Penerimaan siswa baru pihak sekolah masih menggunakan sistem manual yakni calon siswa mengisi kertas formulir yang diberikan panitia pendaftaran
	Tujuan Penelitian	Membantu pihak sekolah mengelola data siswa dalam proses penerimaan siswa baru menggunakan <i>website</i> yang mana dapat diakses melalui <i>smartphone</i> sehingga kegiatan penerimaan dapat berjalan dengan efektif
	Metode Penelitian	Metode <i>Waterfall</i>
	Hasil Penelitian	Sistem informasi penerimaan siswa baru berbasis web menggunakan <i>framework codeigniter</i> sehingga dapat

No	Detail Jurnal	
		membantu mempermudah siswa dalam mendaftar dan pihak sekolah dalam melakukan kegiatan penerimaan siswa baru secara <i>online</i>
4	Judul	Perancangan Sistem Informasi Penerimaan Siswa Baru Berbasis Web Pada SD Negeri 43 OKU
	Penerbit	Informatika dan Teknologi (INTECH)
	Penulis	(Ocka Dharma Putra et al.)
	Tahun	2020
	Permasalahan	Sistem penerimaan siswa baru yang berjalan pada SD Negeri 43 OKU saat ini masih menggunakan sistem manual. Calon siswa harus datang langsung ke sekolah untuk mengetahui informasi pendaftaran, melakukan proses pendaftaran hingga melihat pengumuman hasil seleksi
	Tujuan Penelitian	Sistem informasi penerimaan siswa baru yang dibangun menggunakan Bahasa pemrograman berbasis web sehingga dapat diakses oleh calon siswa melalui Jaringan internet
	Metode Penelitian	Metode <i>Waterfall</i>
Hasil Penelitian	Dengan adanya sistem penerimaan berbasis web proses penerimaan akan menjadi lebih mudah dan cepat. Selain itu laporan dapat dibuat secara otomatis oleh sistem	
5	Judul	Rancang Bangun Aplikasi Penerimaan Siswa Baru Berbasis Web Menggunakan <i>Framework Codeigniter</i>
	Penerbit	Jurnal RESPONSIF
	Penulis	(Anissa & Prasetio)
	Tahun	2021
	Permasalahan	SMK Ma'arif Cicalengka Kabupaten Bandung saat ini belum memanfaatkan perkembangan teknologi secara maksimal, segala aktifitas yang menyangkut keperluan

No	Detail Jurnal	
		sekolah dan siswa dilakukan secara konvensional, salah satunya pendaftaran siswa baru di mana para calon siswa dan walinya diharuskan datang langsung ke sekolah
	Tujuan Penelitian	Mempermudah mekanisme penerimaan siswa baru menjadi sistem yang terkomputerisasi berbasis web menggunakan <i>framework codeigniter</i>
	Metode Penelitian	Metode <i>Waterfall</i>
	Hasil Penelitian	Dengan adanya aplikasi ini pelaksanaan penerimaan siswa baru menjadi lebih transparan dan akomodatif

Kesimpulan yang dapat penulis ambil tinjauan dari literatur di atas yaitu sebagai berikut :

1. Menggunakan metode pengembangan *Extreme Programming* (XP).
2. Tahap pengembangan aplikasi menggunakan *Framework Laravel*.
3. Pengujian sistem menggunakan pengujian ISO 25010.

## 2.2 Konsep Dasar Sistem Informasi

### 2.2.1 Pengertian Sistem

Sistem merupakan sekumpulan individu yang bekerja sama mengikuti aturan yang terstruktur dan sistematis untuk membentuk satu entitas yang menjalankan fungsi tertentu demi mencapai tujuan. Sistem memiliki berbagai karakteristik atau atribut seperti komponen, batasan, lingkungan eksternal, penghubung *input, output*, proses, tujuan (Mufti Prasetyo et al., 2022).

Istilah “sistem” berasal dari bahasa Latin (*systema*) dan Yunani (*sustema*) yang merujuk pada kesatuan yang terdiri dari komponen atau elemen yang saling terhubung untuk memudahkan aliran informasi, materi, atau energi guna mencapai suatu tujuan. Konsep ini sering digunakan untuk menggambarkan sekumpulan entitas yang saling berinteraksi, di mana model matematika sering kali dapat dikembangkan (Wulandari et al., 2022). Selain itu, sistem dapat dipahami sebagai jaringan prosedur yang saling terkait, bekerja bersama untuk melaksanakan kegiatan atau mencapai tujuan tertentu.

Berdasarkan kajian literatur yang telah disampaikan, sistem dapat didefinisikan sebagai kumpulan komponen, jaringan, dan elemen yang bekerja sama untuk membentuk kesatuan guna mencapai tujuan bersama.

### **2.2.2 Pengertian Informasi**

Informasi adalah hasil olahan data yang disajikan dalam bentuk yang lebih bermanfaat dan relevan bagi penerimanya, dengan tujuan mengurangi ketidakpastian dalam proses pengambilan keputusan terkait suatu situasi (Mufti Prasetyo et al., 2022)

Dalam konteks lain, informasi dapat diartikan sebagai data yang telah diproses sehingga menjadi lebih berguna dan aplikatif untuk membantu pengambilan keputusan saat ini atau di masa depan (Wulandari et al., 2022).

Selain itu, informasi juga bisa diartikan sebagai data yang telah diolah sehingga menjadi lebih berarti dan memiliki nilai guna yang tinggi bagi penerimanya.

Berdasarkan pemahaman dari literatur yang telah dijelaskan, informasi dapat disimpulkan sebagai data yang telah diproses dan dikelola sedemikian rupa sehingga menjadi berguna dan relevan bagi pengguna dalam berbagai konteks keputusan.

### **2.2.3 Pengertian Sistem Informasi**

Sistem informasi adalah gabungan teratur dari elemen-elemen seperti manusia, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber data yang bekerja sama untuk mengumpulkan, memproses, dan menyebarkan informasi di dalam sebuah organisasi (Mufti Prasetyo et al., 2022).

Sistem ini dirancang untuk memenuhi kebutuhan pengolahan transaksi sehari-hari, mendukung kegiatan operasional, manajerial, serta strategis organisasi, dan juga menyediakan laporan-laporan penting bagi pihak eksternal yang memerlukan (Wulandari et al., 2022).

Selain itu, sistem informasi mengacu pada metode terstruktur untuk mengumpulkan, memasukkan, memproses, menyimpan, serta mengelola data, serta menyajikan dalam bentuk laporan yang dapat mendukung perusahaan atau organisasi dalam mencapai tujuan mereka (Gulo et al., 2021).

Berdasarkan penjelasan dari berbagai literatur yang telah disampaikan, sistem informasi dapat dipahami sebagai integritas perangkat keras, perangkat lunak, dan sumber daya manusia yang berkolaborasi secara harmonis untuk menghasilkan informasi yang bermanfaat dan mendukung pengguna dalam mencapai berbagai tujuan organisasi.

### **2.3 Penerimaan Santri Baru**

Penerimaan Santri Baru (PSB) adalah merupakan kegiatan administrasi yang vital pada sebuah pesantren. Administrasi pendidikan merupakan suatu proses yang dilakukan oleh dua orang atau lebih dalam suatu organisasi yang berguna untuk mencapai tujuan dari organisasi tersebut (Rosanti Sarumaha & Afriansyah, 2019). Melalui kegiatan ini pesantren akan mendapatkan calon santri baru yang akan meneruskan kegiatan pendidikan di pesantren tersebut. Oleh karena itu, kegiatan ini harus dilakukan dengan efektif dan efisien untuk meningkatkan pelayanan pesantren.

Penerimaan Santri Baru yang dilaksanakan menggunakan prosedur-prosedur manual, seperti penggunaan formulir manual, perekaman data manual, pengarsipan manual dan sebagainya menimbulkan banyak kendala. Oleh karena itu, diperlukan sistem informasi berbasis *website* untuk mengatasi kendala-kendala pada administrasi berbasis prosedur-prosedur manual.

### **2.4 Alat Pengembangan Sistem**

#### **2.4.1 HTML (*Hyper Text Markup Language*)**

HTML (*Hyper Text Markup Language*) adalah bahasa yang menggunakan serangkaian tanda khusus, atau yang dikenal sebagai *tag*, untuk menginstruksikan browser tentang cara menampilkan halaman *website* dengan benar (Mufti Prasetyo et al., 2022). Ketika kita membuka sebuah situs *website*, sebenarnya kita sedang mengakses dokumen yang ditulis menggunakan format HTML. HTML berfungsi sebagai protokol yang mengatur transfer data atau dokumen dari server *website* ke browser kita. Dengan HTML, kita bisa menjelajahi internet dan menikmati tampilan halaman *website* yang beragam dan menarik. HTML menawarkan berbagai kemampuan, antara lain (Gulo et al., 2021) :

- a. Menyusun dan mengatur tampilan halaman web serta kontennya.
- b. Membuat tabel untuk menampilkan data dengan cara yang terstruktur.

- c. Menerbitkan halaman web ke internet agar dapat diakses orang lain.
- d. Membuat formulir yang dapat digunakan untuk pendaftaran dan transaksi *online*.

Dengan HTML, Anda memiliki alat yang kuat untuk menciptakan dan mengelola berbagai elemen web yang membuat pengalaman *browsing* menjadi lebih interaktif dan informatif.

#### **2.4.2 PHP (*Personal Home Page/Hypertext Preprocessor*)**

*Hypertext Preprocessor* (PHP), awalnya dikenal sebagai *Personal Home Page Hypertext Processor*, adalah bahasa pemrograman yang digunakan untuk *scripting* di sisi server. Ini berarti bahwa PHP menjalankan skrip di server, kemudian hasilnya dikirimkan ke klien, yaitu pengguna yang mengakses halaman web melalui browser mereka (Wulandari et al., 2022). PHP memiliki kemiripan dengan beberapa bahasa pemrograman lain seperti C, *Java*, dan *Perl*, dan dikenal karena kemudahannya untuk dipelajari.

Dalam konteks web, PHP berfungsi sebagai alat pemrosesan data di server, mengubah permintaan dari klien menjadi informasi yang dapat ditampilkan di halaman web. PHP juga berperan sebagai penghubung antara aplikasi web dan *database*, memungkinkan pengambilan dan manipulasi data secara dinamis (Gulo et al., 2021). Sebagai bahasa *scripting* sisi server, PHP bekerja bersama HTML untuk menciptakan halaman web yang interaktif dan dinamis. Ini berarti bahwa semua perintah PHP diproses di server, sementara hasil akhirnya disisipkan dalam dokumen HTML yang dikirim ke browser.

Ketika pengguna mengakses situs yang menggunakan PHP, server terlebih dahulu memproses skrip PHP sebelum mengirimkan halaman HTML yang sudah diproses ke browser. Hal ini menjaga kode PHP tetap tersembunyi dari pengguna, sehingga meningkatkan keamanan halaman web dibandingkan dengan beberapa teknologi lainnya, seperti ASP (Edy Subowo & Meidika Saputra, 2019). PHP juga kompatibel dengan berbagai *database*, termasuk *MySQL*. Ini memungkinkan PHP untuk mengakses dan memanipulasi data dalam database *MySQL* melalui halaman web, memberikan fleksibilitas tambahan dalam pengelolaan data (Pratiwi et al., 2021).

### 2.4.3 CSS (*Cascading Style Sheets*)

CSS (*Cascade Style Sheets*) adalah bahasa pemrograman yang digunakan untuk mendefinisikan tampilan halaman web, termasuk elemen seperti warna, tata letak, dan jenis *font*. Dengan CSS, seorang pengembang web dapat merancang halaman yang responsif dan dapat menyesuaikan tampilannya dengan berbagai ukuran layar, dari *smartphone* hingga *dekstop*. Biasanya, CSS dipisahkan dari kode HTML, meskipun dapat juga disematkan langsung di dalam halaman HTML jika diperlukan.

CSS memainkan peran penting dalam menciptakan desain web yang menarik dan fungsional dengan mengatur estetika dan tata letak halaman. Ini memungkinkan pengembang untuk membuat halaman web yang tidak hanya informatif tetapi juga elegan dan visual menarik (Gulo et al., 2021).

Dengan CSS, Anda dapat mengubah tampilan halaman secara menyeluruh tanpa memengaruhi struktur konten, sehingga memudahkan perancangan dan pemeliharaan situs web yang menarik dan profesional.

### 2.4.4 JavaScript

*JavaScript* adalah bahasa pemrograman web yang tergolong dalam *Client Side Programming Language*. Ini berarti bahwa *JavaScript* dijalankan di sisi klien, yaitu di dalam web browser seperti *Google Chrome*, *Mozilla Firefox*, *Opera Mini*, dan lainnya. Diciptakan pada pertengahan tahun 1990-an, *JavaScript* memainkan peran penting dalam mengatur interaksi halaman web dengan pengguna. Meskipun namanya mirip dengan Java, kedua bahasa ini sebenarnya sangat berbeda. *JavaScript* dapat disisipkan langsung ke dalam dokumen HTML atau digunakan sebagai *file* terpisah yang dikaitkan dengan dokumen HTML yang relevan. Fitur-fitur dalam *JavaScript* dirancang khusus untuk mengelola dan meningkatkan pengalaman pengguna di halaman web, menjadikannya alat yang sangat berharga dalam mengembangkan web modern.

### 2.4.5 Laravel

*Laravel* adalah sebuah *framework* PHP bersifat *open source* yang dirilis dibawah lisensi MIT dan dibangun dengan konsep MVC (*Model View Controller*). *Laravel* adalah pengembangan *website* berbasis MVC yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya

pemeliharaan. *Laravel* merupakan *framework* PHP terbaik yang dikembangkan oleh Taylor Otwell.

Menurut (Endra et al., 2021), *Laravel* merupakan kerangka kerja pengembangan web MVC yang meningkatkan kualitas perangkat lunak dengan meningkatkan produktivitas tenaga dengan sintak pengembangan dan perbaikan serta secara signifikan mengurangi waktu implementasi.

#### **2.4.6 XAMPP (*Cross-Platform (X), Apache, MySQL, PHP, Perl*)**

XAMPP adalah sebuah aplikasi yang berfungsi sebagai server lokal atau *localhost* yang terintegrasi dengan berbagai komponen penting. Paket ini mencakup beberapa program kunci, seperti *Apache* HTTP Server untuk layanan web, *MySQL* sebagai sistem manajemen *database*, serta interpreter untuk bahasa pemrograman PHP dan *Perl*. Nama XAMPP sendiri merupakan akronim yang menggabungkan elemen-elemen utama dari sistem ini : “X” untuk representasi sistem operasi yang beragam, dan “AMP” yang mewakili *Apache, MySQL, PHP, dan Perl*.

Sebagai perangkat lunak sumber terbuka, XAMPP dikembangkan oleh komunitas *Apache* dan menyediakan solusi lengkap untuk pengembangan web. Paket ini mencakup distribusi *Apache* untuk server web, *MariaDB* sebagai pengganti *MySQL*, serta PHP dan *Perl* untuk pemrograman *server-side*, memungkinkan pengembangan untuk dengan mudah membangun dan menguji aplikasi web secara lokal. Dengan XAMPP, proses pengembangan web menjadi lebih mudah dan efisien, menjadikan pilihan populer di kalangan pengembang perangkat lunak.

#### **2.4.7 Visual Studio Code**

*Visual Studio Code* merupakan *software* yang dapat digunakan dalam pembuatan kode-kode program. Dalam hal ini *Visual Studio Code* digunakan dalam proses pembuatan *website* untuk Pondok Pesantren Minhajul Huda. *Visual Studio Code* adalah *software* yang sangat ringan, namun kuat digunakan dalam proses pembuatan kode-kode program dengan *Built-in* dukungan untuk *JavaScript*, naskah dan *Node.js* dan memiliki *array* dan tentunya memiliki beragam ekstensi yang tersedia untuk bahasa lain, seperti *C++*, *C#*, *Python*, dan *PHP*.

## 2.5 *Database*

*Database* dapat didefinisikan sebagai kumpulan informasi atau data yang saling terhubung dan disimpan di luar komputer. Untuk mengelola dan memanipulasi data tersebut, diperlukan perangkat lunak khusus. Saat ini, *database* modern dikembangkan dengan bantuan perangkat lunak di komputer agar pengguna dapat mengakses dan mengelola data dengan mudah dan efisien. Perangkat lunak yang digunakan untuk tujuan ini adalah Sistem Manajemen basis data (DBMS). DBMS adalah sekumpulan program perangkat lunak canggih yang mengatur struktur, manajemen, dan penyimpanan data dalam *database*, serta memastikan integritas dan keamanan data (Habibullah Yusril et al., 2023).

### 2.5.1 *MySQL (My Structured Query Language)*

*MySQL* adalah salah satu sistem *database* yang paling populer dan digunakan secara global. Pada dasarnya, *MySQL* beroperasi menggunakan SQL (*Structured Query Language*), yang menjadikan salah satu standar utama dalam pengelolaan data di seluruh dunia (Gulo et al., 2021). *MySQL* menawarkan dua jenis lisensi : *Free Software* dan *Shareware*. Versi yang paling umum digunakan adalah *MySQL Free Software* yang dirilis di bawah Lisensi GNU/GPL (*General Public License*).

*MySQL* pertama kali dikembangkan oleh Michael Widenius, seorang *programmer database* yang visioner. Selain berfungsi sebagai *database server*, *MySQL* juga dapat berperan sebagai *client* yang mengakses *database MySQL*, membuatnya fleksibel dan serbaguna dalam penggunaannya (Edy Subowo & Meidika Saputra, 2019). Meskipun *MySQL* tersedia secara gratis di bawah lisensi GNU GPL, terdapat juga opsi lisensi *komersil* untuk kasus-kasus di mana penggunaan GPL tidak sesuai.

Dengan kemampuannya untuk berfungsi baik sebagai server maupun *client*, serta ketersediaan dalam berbagai model lisensi, *MySQL* menjadi pilihan yang sangat fleksibel dan terjangkau untuk berbagai kebutuhan pengelolaan *database*.

## 2.6 *Metode Extreme Programming (XP)*

*Extreme Programming* adalah salah satu metode rekayasa perangkat lunak yang umum digunakan bagi pengembang untuk mengembangkan suatu aplikasi.

*Extreme programming* banyak digunakan pengembang untuk mengembangkan perangkat lunak dengan cepat (Suryanta, 2017). Ada beberapa tahap pengembangan perangkat lunak dengan *Extreme Programming* yaitu sebagai berikut :

1. Perencanaan

Pada tahap ini dimulai dengan konteks bisnis yang dari aplikasi yang akan dibuat. Penentuan fitur yang akan dibuat, kemudian fungsi aplikasi, waktu, biaya, *output* aplikasi dan alur pengembangan tersebut.

2. Perancangan

Tahapan ini membuat desain aplikasi yang akan dibuat secara sederhana. Alat yang sering digunakan pada tahap ini adalah *Class Responsibility Collaborator* (CRC). CRC digunakan dalam proses membangun kelas yang akan digunakan pada *use case* diagram, *class* diagram dan objek.

3. Pengkodean

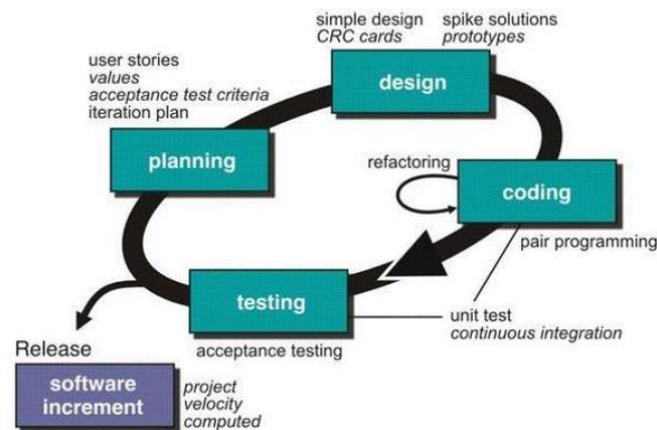
Tahapan ini merupakan tahapan utama dalam pengembangan aplikasi dengan *extreme programming* yaitu *pair programming*.

4. Pengujian

Tahapan terakhir berfokus terhadap pengujian fitur, apakah dapat berfungsi dengan benar dan memastikan tidak terdapat eror. Sesuai dengan permintaan klien.

### 2.6.1 Tahapan Metode *Extreme Programming* (XP)

Pada metode ini, pengembang harus mengikuti empat Langkah berikut untuk menyelesaikan pekerjaan pengembangan :



**Gambar 2.1 Tahapan Metode *Extreme Programming***

Sumber (astizardian.wordpress.com)

## 1. Perencanaan (*Planning*)

Tahapan ini merupakan tahapan yang diperlukan sebelum pengembang membuat sistem. Tahapan ini sangat penting karena diperlukan perencanaan atau analisa kebutuhan pengguna pada saat pembuatan sistem. Pengembang memutuskan bagaimana membangun hasil cerita pengguna sesuai dengan janji yang disepakati, Adapun *story-story* yang dilakukan dengan cara :

- a. Pengguna menceritakan apa permasalahan pada sistem yang digunakan dan sistem seperti apa yang akan dibangun.
- b. Berdasarkan hasil cerita pengguna maka peneliti menentukan poin pada bagian *value* untuk memutuskan apa saja yang akan dibangun.
- c. Dari hasil kesepakatan tersebut maka peneliti menentukan *acceptance test criteria* yaitu menentukan kriteria apa saja yang nantinya sebagai acuan terhadap sistem yang akan diuji.
- d. Sehingga hasil peneliti menyimpulkan berapa kali akan dilakukan *release* dan perbaikan pada tahap *iteration plan* merencanakan beberapa kali akan dilakukan uji terhadap sistem yang dibangun.

## 2. Perancangan (*Design*)

Setelah tahap perencanaan selesai, tahap selanjutnya adalah desain. Pada tahap ini pengembang merancang dengan membuat model yang diawali dengan pemodelan sistem, dilanjutkan dengan pemodelan arsitektur dan pemodelan basis data untuk memberikan gambaran tentang sistem yang akan dibangun.

- a. *Simple Design* adalah pengembang membangun perangkat lunak dengan desain sederhana. Dimulai dengan desain yang sederhana dilakukan menggunakan UML seperti *Use Case Diagram* dan *Class Diagram*.
- b. CRC Card digunakan untuk mengenali dan mengatur *object oriented class* yang sesuai dengan pengembangan. Jika pada saat perancangan terdapat ketidaksesuaian maupun perbaikan maka akan dilakukan.
- c. *Prototype* adalah bagian perancangan berupa *user interface* dalam bentuk *wireframing* untuk mempermudah pengguna melihat desain sistem.

### 3. Pengkodean (*Coding*)

Pada proses pengkodean, peneliti menyesuaikan terhadap *story* pengguna sehingga sistem yang dibangun sesuai. Proses pengkodean yang dilakukan yaitu :

- a. *Pair Programming* merupakan tahap sistem dibangun dengan Bahasa pemrograman dan media Penyimpanan yang telah disepakati.
- b. *Refactory* merupakan tahap yang dilakukan Ketika terjadi ke tidak sesuaian kode program sehingga dilakukan perbaikan guna mendapatkan hasil yang sesuai.

### 4. Pengujian (*Testing*)

Tahap pengujian dilakukan oleh pengguna sebagai *user* dengan melakukan uji sesuai dengan *acceptance test* yang telah ditentukan dan disetujui. Unit test yang telah dibuat fokus pada keseluruhan fitur dan fungsional sistem. Sehingga sistem dapat disimpulkan telah sesuai dan dapat dirilis.

## 2.7 *Unified Modeling Language (UML)*

Menurut (Salahuddin & Sukamto, 2018) *Unified Modeling Language (UML)* adalah bahasa standar yang sangat populer di industri perangkat lunak. UML digunakan secara luas untuk berbagai keperluan, termasuk mendefinisikan kebutuhan sistem, melakukan analisis dan perancangan, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Dengan kemampuan untuk menyediakan representasi visual yang jelas dan konsisten, UML mempermudah komunikasi antara tim pengembang dan pemangku kepentingan, serta meningkatkan pemahaman terhadap struktur dan dinamika sistem.

UML meningkatkan pengembangan untuk merancang dan memodelkan sistem dengan berbagai jenis diagram. Masing-masing diagram menawarkan pandangan yang berbeda dari sistem yang sedang dibangun. Melalui penggunaan UML, proses pengembangan perangkat lunak menjadi lebih terstruktur dan terorganisir, memungkinkan tim untuk menangani kompleksitas proyek dengan lebih efisien dan efektif. Dalam industri yang terus berkembang ini, UML menjadi alat penting untuk memastikan bahwa sistem perangkat lunak memenuhi kebutuhan dan harapan pengguna secara akurat dan konsisten.

### 2.7.1 Use Case Diagram

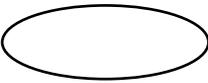
Menurut (Salahuddin & Sukamto, 2018) *use case* atau diagram *use case* adalah metode pemodelan yang digunakan untuk menggambarkan perilaku sistem informasi yang akan dikembangkan. Diagram ini menunjukkan interaksi antara satu atau lebih aktor dengan sistem, menyediakan panduan visual tentang bagaimana sistem akan berfungsi dalam konteks penggunaan nyata.

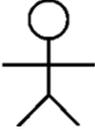
Penamaan *use case* haruslah sederhana dan mudah dipahami, agar semua pihak yang terlibat dapat mengerti dengan jelas fungsionalitas yang dimaksud. Ada dua elemen utama dalam *use case* yang perlu diperhatikan : aktor dan *use case* itu sendiri.

1. Aktor adalah individu, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang sedang dikembangkan, berada di luar batasan sistem itu sendiri. Meskipun simbol yang digunakan untuk aktor sering berupa gambar orang, aktor tidak selalu harus berupa manusia, bisa juga berupa sistem lain atau proses.
2. *Use Case* menggambarkan fungsionalitas yang disediakan oleh sistem sebagai unit-unit terpisah yang berkomunikasi satu sama lain atau dengan aktor. Ini merinci bagaimana sistem menangani berbagai operasi dan skenario berdasarkan interaksi tersebut.

Salahuddin & Sukamto, 2018 juga menjelaskan simbol-simbol yang digunakan dalam diagram *use case*, yang memperjelas representasi visual dari elemen-elemen ini dan bagaimana mereka saling berhubungan. Diagram *use case* ini dan bagaimana mereka saling berhubungan. Diagram *use case* ini berfungsi sebagai alat yang sangat berharga untuk merancang sistem dengan memahami kebutuhan pengguna dan interaksi sistem diharapkan, memastikan bahwa pengembangan sistem dapat dilakukan dengan orientasi yang jelas dan terarah.

**Tabel 2.2 Simbol Use Case Diagram**

No	Simbol - simbol	Keterangan
1	<p style="text-align: center;"><i>Use Case</i></p> 	Fungsionalitas sistem disajikan melalui unit-unit yang saling berkomunikasi satu sama lain atau dengan aktor, dan biasanya diwakili dengan kata kerja di awal nama <i>use case</i>

No	Simbol - simbol	Keterangan
2	<p data-bbox="491 383 571 416">Aktor</p> 	<p data-bbox="687 288 1353 651">Aktor bisa berupa individu, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang sedang dikembangkan. Meskipun simbol yang digunakan untuk aktor sering kali berupa gambar orang, aktor tidak selalu merujuk pada manusia, biasanya aktor dinyatakan dengan kata benda di awal nama aktor</p>
3	<p data-bbox="427 898 635 931">Ektensi / <i>extend</i></p> <p data-bbox="464 954 600 981">&lt;&lt;extend&gt;&gt;</p> 	<p data-bbox="687 674 1353 1256">Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> menunjukkan bahwa <i>use case</i> tambahan tersebut dapat berfungsi secara independen, meskipun adanya <i>use case</i> tambahan tersebut. Konsep ini mirip dengan prinsip pewaris dalam pemrograman berorientasi objek. Biasanya, <i>use case</i> tambahan memiliki nama yang diawali dengan nama <i>use case</i> utama yang ditambahkan, dan arah panah menunjukkan ke arah <i>use case</i> tambahan. Biasanya, <i>use case</i> tambahan memiliki tipe yang sama dengan <i>use case</i> induknya</p>
4	<p data-bbox="464 1283 600 1368">Asosiasi / <i>association</i></p> 	<p data-bbox="687 1279 1353 1424">Interaksi antara aktor dan <i>use case</i> terjadi ketika aktor terlibat dalam <i>use case</i> atau ketika <i>use case</i> berkomunikasi dengan aktor.</p>
5	<p data-bbox="440 1559 624 1644">Generalisasi / <i>generalization</i></p> 	<p data-bbox="687 1447 1353 1816">Hubungan generalisasi dengan spesialisasi (umum-khusus) antara dua <i>use case</i> terjadi ketika satu <i>use case</i> mencakup fungsi yang lebih umum dibandingkan dengan yang lainnya. Sebagai contoh, panah yang mengarah pada <i>use case</i> menunjukkan bahwa <i>use case</i> tersebut adalah generalisasi umum) dari <i>use case</i> yang dituju.</p>
6	<p data-bbox="427 1839 635 1924">Menggunakan / <i>include / uses</i></p> <p data-bbox="459 1946 603 1973">&lt;&lt;include&gt;&gt;</p>	<p data-bbox="687 1839 1353 1924">Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> terjadi ketika <i>use case</i> yang ditambahkan memerlukan <i>use</i></p>

No	Simbol - simbol	Keterangan
		<i>case</i> tersebut untuk menjalankan fungsinya atau sebagai syarat agar <i>use case</i> dapat dijalankan.

Sumber: (Salahuddin & Sukamto, 2018)

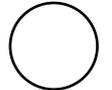
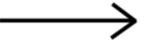
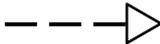
### 2.7.2 Class Diagram

Menurut (Salahuddin & Sukamto, 2018) *class* diagram menggambarkan struktur sistem dengan mendefinisikan kelas-kelas yang diperlukan untuk membangun sistem. Kelas memiliki apa yang disebut atribut, *method* atau operasi. Berikut adalah penjelasan dari atribut dan *method*.

1. Atribut merupakan *variable* yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi yang dimiliki suatu kelas.

Berikut merupakan simbol-simbol yang terdapat pada class diagram dapat dilihat pada tabel 2.3 di bawah ini.

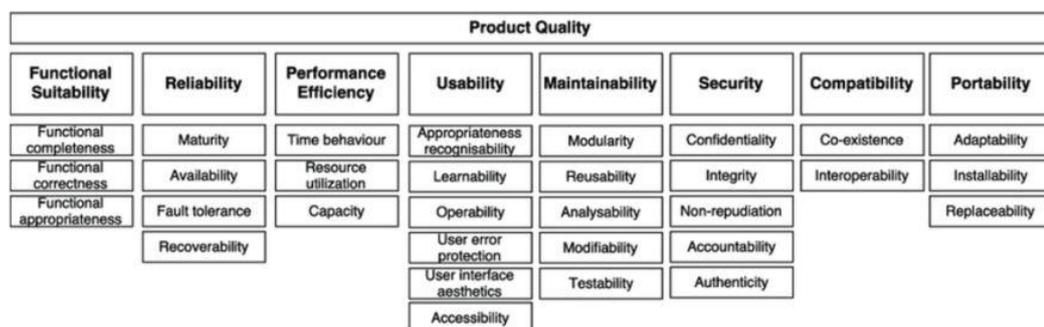
**Tabel 2.3 Simbol Class Diagram**

No	Simbol	Nama	Keterangan
1		<i>Class</i>	Kumpulan objek dengan atribut dan operasi yang sama
2		<i>Interface</i>	<i>Interface</i> sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3		<i>Asosiasi Berarah</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan simbol
4		<i>Generalisasi</i>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
5		<i>Dependency</i>	Relasi antar kelas dengan makna ketergantungan antar kelas
6		<i>Agregasi</i>	Relasi antar kelas dengan makna semua bagian

No	Simbol	Nama	Keterangan
7	_____	<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

## 2.8 Pengujian ISO 25010

Model ISO 25010 merupakan salah satu kerangka untuk menguji dan mengevaluasi kualitas perangkat lunak dan merupakan bagian dari *Software Product Quality Requirements and Evaluation (SQuaRE)*. (Lamada et al., 2020), mengungkapkan bahwa di antara berbagai standar pengujian, ISO 9126 dan ISO 25010 adalah standar Internasional untuk pengujian perangkat lunak. Standar ISO 25010 dikembangkan sebagai alternatif standar ISO 9126 berdasarkan perkembangan ICT (*Information and Communication Technology*). Model ini sebelumnya, yaitu ISO 9126. Dalam ISO 25010, terdapat delapan domain (kriteria) utama, yang mencakup beberapa sub-domain tambahan serta beberapa sub-domain yang dipindahkan dari domain lain. Struktur lengkap dari ISO 25010 dapat dilihat pada gambar 2.2 di bawah ini.



**Gambar 2.2 Karakteristik ISO 25010**

Berikut ini adalah penjelasan dari gambar di atas yang merupakan delapan karakteristik ISO 25010. Adapun penjelasannya yaitu :

1. *Functionality Suitability*, menggunakan alat penelitian berupa *test case* dengan *Guttman scale*. *Guttman scale*/skala *guttman* digunakan untuk memberikan jawaban unik untuk pertanyaan yang Anda coba ujikan.
2. *Performance Efficiency*, dilakukan untuk menguji kinerja aplikasi yang dikembangkan.

3. *Compatibility*, merupakan kemampuan suatu komponen atau sistem untuk bertukar informasi.
4. *Usability*, untuk menganalisis umpan balik pengguna menggunakan skala 5 pilihan.
5. *Reliability*, dilakukan untuk pengujian keandalan atau keterpercayaan sistem.
6. *Security*, untuk menguji sejauh mana sistem atau produk yang menyediakan layanan melindungi dari akses, penggunaan, modifikasi, gangguan, atau pengungkapan bahaya.
7. *Maintainability*, menggunakan alat ukur yang telah diuji oleh peneliti langsung di lapangan kegiatan, sesuai dengan alat uji yang disebutkan, pengujian ini terdiri dari 3 aspek, yaitu alat ukur, perhitungan konsistensi kesederhanaan.
8. *Portability*, pengujian sejauh mana suatu sistem atau produk dapat dipindahkan dari suatu ruangan ke ruangan lainnya.

Dari kedelapan domain di atas penulis menggunakan dua domain antara lain *functional suitability* dan *usability*. Penjelasan domain yang digunakan dalam penelitian ini dapat dilihat pada tabel 2.4 di bawah ini.

**Tabel 2.4 Domain ISO 25010**

No	Domain	Sub-Domain	Indikator <i>Testing</i>
1	<i>Functional Suitability</i>	<i>Functional completeness</i>	Sejauh mana fungsi yang disediakan mencakup semua tugas dan tujuan pengguna secara spesifik.
		<i>Functional correctness</i>	Sejauh mana sistem menyediakan hasil benar sesuai kebutuhan.
		<i>Functional Appropriateness</i>	Sejauh mana fungsi yang disediakan mampu memfasilitasi penyelesaian tugas dan tujuan tertentu.
2	<i>Usability</i>	<i>Appropriateness recognizability</i>	Sejauh mana pengguna dapat mengetahui apakah sistem atau produk sesuai kebutuhan mereka.

No	Domain	Sub-Domain	Indikator <i>Testing</i>
		<i>Learnability</i>	Sejauh mana sistem dapat digunakan oleh pengguna untuk mencapai tujuan tertentu yang belajar menggunakan sistem atau produk dengan efisien, efektif, kebebasan dari risiko dan kepuasan dalam konteks tertentu.
		<i>Operability</i>	Sejauh mana produk atau sistem mudah dioperasikan dan dikontrol.
		<i>User error protection</i>	Sejauh mana produk atau sistem melindungi pengguna terhadap membuat kesalahan.
		<i>User interface aesthetics</i>	Sejauh mana antarmuka pengguna dari produk atau sistem memungkinkan interaksi yang menyenangkan dan memuaskan pengguna.
		<i>Accessibility</i>	Sejauh mana produk atau sistem dapat digunakan oleh semua kalangan untuk mencapai tujuan tertentu sesuai konteks penggunaan.

Sumber : (Gunawan & Triantoro, 2017)

## 2.9 Skala *Likert*

Skala *Likert* adalah alat yang digunakan untuk menilai sikap, pendapat, dan persepsi individu atau kelompok terhadap fenomena sosial tertentu. Dengan menggunakan skala *Likert*, variabel yang ingin diukur diuraikan menjadi beberapa indikator. Indikator-indikator ini kemudian menjadi dasar dalam menyusun item-item instrumen, baik berupa pernyataan maupun pertanyaan. Setiap item dalam instrumen ini menggunakan skala *Likert* yang menawarkan berbagai gradasi

jawaban, mulai dari sangat positif hingga sangat negatif, sebagaimana dijelaskan dalam tabel 2.5 berikut.

**Tabel 2.5 Score Skala Likert**

Keterangan	Score
Setuju/ Selalu/Sangat Positif Diberi Skor	5
Setuju/Sering/Positif Diberi Skor	4
Ragu-Ragu/Kadang-Kadang/Netral Diberi Skor	3
Tidak Setuju/Hampir Tidak Pernah/Negatif Diberi Skor	2
Sangat Tidak Setuju/Tidak Pernah Diberi Skor	1

Instrumen untuk menilai kelayakan sistem yang memanfaatkan skala *Likert* dapat disusun dalam format tabel seperti yang tertera di bawah ini.

**Tabel 2.6 Kualifikasi Skala Likert**

Tingkat Pencapaian	Kualifikasi	Keterangan
90% - 100%	Sangat Baik	Tidak Perlu revisi
75% - 89%	Baik	Sedikit Revisi
65% - 74%	Cukup	Di Revisi Secukupnya
55% - 64%	Kurang	Banyak hal yang direvisi
0% - 54%	Sangat Kurang	Diulang Membuat Produk

Berdasarkan pengujian yang telah dilakukan maka akan dilakukan perhitungan dengan rumus :

$$\text{Score Aktual} = (5 \times \text{Nilai Jawaban}) + (4 \times \text{Nilai Jawaban}) + (3 \times \text{Nilai Jawaban}) + (2 \times \text{Nilai jawaban}) + (1 \times \text{Nilai jawaban})$$

$$\text{Score Ideal} = \text{Bobot Terbesar} \times \text{Total responden}$$

$$\text{Result} = (\text{Score Aktual} : \text{Score Ideal}) \times 100.$$

## 2.10 Skala Guttman

Skala *Guttman* adalah metode untuk mengukur sikap atau pendapat dengan menyusun item-item dalam urutan hierarkis. Item-item ini disusun dari yang paling sederhana hingga yang paling kompleks. Tujuannya adalah untuk mengukur

seberapa konsisten seseorang dalam menyetujui pernyataan yang disusun dalam urutan ini.

Cara Kerja :

- Hierarki Item : Dalam Skala *Guttman*, setiap item diletakkan dalam urutan tertentu. Jika seseorang setuju dengan item yang lebih kompleks, mereka juga harus setuju dengan item-item yang lebih sederhana yang terletak di bawahnya.

Pengukuran :

- Skor : Skor dihitung berdasarkan seberapa banyak item yang disetujui oleh responden.

$$Skor = \sum_{i=1}^n x^i \quad \text{Di mana :}$$

$x_i = 1$  jika setuju dengan item ke -  $i$ , atau 0 jika tidak setuju.

$\pi =$  jumlah item.

- Konsisten Pola : Menilai apakah respons mengikuti urutan hierarki yang diharapkan. Jika seseorang setuju dengan item yang lebih tinggi, mereka seharusnya setuju dengan semua item yang lebih rendah.

Koefisien *Scalogram* :

- Koefisien *Scalogram* mengukur seberapa baik item mengikuti urutan ya diharapkan.

$$S = \frac{\text{Jumlah item yang sesuai}}{\text{Jumlah item yang diharapkan}}$$