



## **BAB II**

# **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Sebagai acuan dalam pengembangan Sistem Informasi Presensi dan Pembelajaran Les Berbasis *Website*, maka penulis mencari referensi terhadap beberapa penelitian terdahulu yang keterkaitan langsung maupun tidak langsung dengan penelitian penulis.

Penelitian pertama dilakukan oleh Alif Gilang Mulia tahun 2020 yang berjudul “Sistem Informasi Absensi Berbasis *Web* di Politeknik Negeri Padang”. Pada penelitian sistem ini bertujuan untuk memudahkan mahasiswa dalam melakukan presensi dan membantu proses pelaporan absen setiap mahasiswa. Penelitian ini menggunakan metode pengembangan sistem *Prototype* dan bahasa pemrograman PHP serta basis data *MySQL* serta *framework Codeigniter*. Sistem yang dibangun yaitu sistem absensi berbasis website yang dapat membantu proses pelaporan absen setiap mahasiswa ke kantor registrar yang ada di Politeknik sehingga data yang didapatkan akurat.[4].

Penelitian kedua dilakukan oleh Triyono, Rosiana Safitri dan Taufik Gunawan pada tahun 2018 dengan judul “Perancangan Sistem Informasi Absensi Guru Dan Staff Pada SMK Pancakarya Tangerang Berbasis *Web*”. Tujuan dari penelitian ini untuk mengatasi permasalahan absensi yang dihadapi sehingga mempermudah bagi guru dan staff yang ingin melakukan absensi. Membangun aplikasi berbasis *web* yang memudahkan guru dan *staff* dalam melakukan absensi dan memudahkan dalam mendapatkan laporan yang tepat dan akurat. Metode pengembangan sistem yang akan digunakan adalah model *waterfall* [5].

Penelitian ketiga dilakukan oleh Sarah Nur, dan Iyan Gustiana pada tahun 2018 dengan judul “Sistem Informasi Akademik Bimbingan Belajar Berbasis *Web* Di ILC Garut”. Tujuan dari penelitian ini adalah untuk merancang sistem bimbingan belajar yang memudahkan kegiatan operasional sehingga dibangunlah sistem informasi bimbingan belajar berbasis *website* yang memudahkan kegiatan operasional seperti proses pengolahan data dan penginputan secara terkomputerisasi dan efisien

sehingga tidak lagi dicatat dalam buku. Metode pengembangan sistem yang akan digunakan adalah model *waterfall* [6].

Penelitian keempat dilakukan oleh Alex Holyfiel, Mendarissan Aritonang dan Murfia J. Purba pada tahun 2021 dengan judul “Sistem Informasi Bimbingan Belajar Number One Medan Berbasis *Web*”. Pendaftaran dan pembayaran les masih menggunakan buku dalam pengarsipannya sehingga belum optimal dalam pendataannya. Maka dibangunlah sistem informasi bimbingan belajar untuk membantu orang tua siswa dan calon siswa bimbingan belajar memberikan informasi tentang Bimbingan Belajar Number One Medan. Pengembangan sistem menggunakan metode penelitian deskriptif dengan pendekatan desain berorientasi objek dan model *prototyping* [7].

Penelitian kelima yang telah dilakukan oleh Muhammad Ramadhan, Nur Alam Arifin tahun 2021 dengan judul “Sistem Informasi Bimbingan Belajar Indo Prestasi Dan *English For Communication*”. Pengolahan data pada bimbingan belajar ini masih menggunakan sistem pembukuan sehingga kurang efektif sehingga dibangunlah sistem informasi berbasis *website* ini untuk membuat penyimpanan data transaksi yang lebih aman dan mudah dibanding menggunakan kertas, membuat proses pencarian data lebih cepat, dan membuat laporan dengan lebih efektif dibanding menggunakan sistem pembukuan. Metode yang digunakan pada penelitian ini adalah metode *grounded research* dengan pengembangan sistem menggunakan metode *waterfall* [2].

Setelah melihat penelitian lain yang sudah ada maka penulis akan mengembangkan sistem dengan judul “Sistem Informasi Presensi dan Pembelajaran Les Berbasis *Website*” yang bertujuan untuk membangun sistem informasi yang dapat mempermudah siswa dan guru les dalam proses presensi, membantu admin dalam proses pengolahan data, mempercepat proses absensi, menghindari kecurangan dalam proses absensi, membantu siswa dalam mendapatkan modul pembelajaran, mengerjakan latihan kuis serta melihat nilai. Perbedaan sistem yang dibangun penulis dengan penelitian sebelumnya yaitu menggunakan *Framework Codeigniter 3* dengan metode pengembangan sistem menggunakan *waterfall*. Pada proses presensi terdapat fitur validasi

presensi siswa yang telah hadir, fitur cetak pada rekapan data presensi dan rekapan data nilai sesuai nama siswanya. Siswa juga dapat mengakses *link* kuis yang terhubung pada *google form*.

**Tabel 2. 1** Tabel Perbandingan Penelitian Sebelumnya

Penulis, Tahun dan Judul	Tujuan	Metode	Hasil yang Dicapai
Alif Gilang Mulia (2020), “Sistem Informasi Absensi Berbasis Web di Politeknik Negeri Padang”	Bertujuan untuk memudahkan mahasiswa dalam melakukan presensi dan membantu proses pelaporan absen setiap mahasiswa.	Metode penelitian yang digunakan yaitu metode <i>Prototype</i> dan menggunakan <i>framework Codeigniter</i>	Membangun sistem absensi berbasis <i>website</i> yang dapat membantu proses pelaporan absen setiap mahasiswa ke kantor registrar yang ada di Politeknik sehingga data yang didapatkan akurat.
Triyono, Rosiana Safitri dan Taufik Gunawan (2018) “Perancangan Sistem Informasi Absensi Guru Dan Staff Pada Smk Pancakarya	Bertujuan untuk mengatasi permasalahan absensi yang dihadapi sehingga mempermudah bagi guru dan staff yang ingin melakukan absensi.	Metode pengembangan sistem yang akan digunakan adalah model <i>waterfall</i> .	Membangun aplikasi berbasis <i>web</i> yang memudahkan guru dan <i>staff</i> dalam melakukan absensi dan memudahkan dalam mendapatkan

Tangerang Berbasis Web”			laporan yang tepat dan akurat.
Sarah Nur, dan Iyan Gustiana (2018) “Sistem Informasi Akademik Bimbingan Belajar Berbasis Web Di ILC Garut”	Untuk merancang sistem bimbingan belajar yang memudahkan kegiatan operasional.	Metode pengembangan sistem yang akan digunakan adalah model <i>waterfall</i>	Merancang sistem informasi bimbingan belajar berbasis <i>website</i> yang memudahkan kegiatan operasional seperti proses pengolahan data dan penginputan secara terkomputerisasi dan efisien sehingga tidak lagi dicatat dalam buku
Alex Holyfiel, Mendarissan Aritonang dan Murfia J. Purba (2021)	Sistem ini dibangun untuk membantu orang tua siswa dan calon siswa bimbingan belajar memberikan informasi	Pengembangan sistem menggunakan metode penelitian deskriptif dengan pendekatan desain berorientasi	Membangun sistem informasi bimbingan belajar yang berbasis <i>website</i> yang membantu orang tua

<p>“Sistem Informasi Bimbingan Belajar Number One Medan Berbasis <i>Web</i>”</p>	<p>tentang Bimbingan Belajar <i>Number One</i> Medan</p>	<p>objek dan model <i>prototyping</i>.</p>	<p>siswa dan calon siswa bimbingan belajar dengan memberikan informasi tentang Bimbingan Belajar <i>Number One</i> Medan</p>
<p>Muhammad Ramadhan, Nur Alam Arifin (2021) “Sistem Informasi Bimbingan Belajar Indo Prestasi Dan <i>English For Communication</i>”</p>	<p>Sistem ini dibangun untuk membuat penyimpanan data transaksi yang lebih aman dan mudah dibanding menggunakan kertas, membuat proses pencarian data lebih cepat, dan membuat laporan dengan lebih efektif dibanding menggunakan sistem pembukuan.</p>	<p>etode yang digunakan pada penelitian ini adalah metode <i>grounded research</i> dengan pengembangan sistem menggunakan metode <i>waterfall</i>.</p>	<p>Membangun sistem informasi bimbingan belajar untuk pengelolaan data yang lebih efektif dibandingkan dengan sistem pembukuan secara manual.</p>
<p>Sekar Arini Damayanti (2022) "Sistem</p>	<p>Bertujuan untuk membangun sistem</p>	<p>Metode pengembangan sistem</p>	<p>Membangun sistem informasi presensi</p>

<p>Informasi Presensi dan Pembelajaran Les Berbasis <i>Website</i>”</p>	<p>informasi yang dapat mempermudah siswa dan guru les dalam proses presensi, membantu admin dalam proses pengolahan data, mempercepat proses absensi, menghindari kecurangan dalam proses absensi, membantu siswa dalam mendapatkan modul pembelajaran, mengerjakan latihan kuis serta melihat nilai</p>	<p>yaitu menggunakan <i>waterfall</i>.</p>	<p>dan pembelajaran les berbasis <i>website</i> yang memiliki kelebihan terdapat fitur validasi presensi siswa yang telah hadir, fitur cetak pada rekapan data presensi dan rekapan data nilai sesuai nama siswanya. Siswa juga dapat mengakses <i>link</i> kuis yang terhubung pada <i>google form</i></p>
---	---	--	---

## **2.2 Landasan Teori**

### **2.2.1 Sistem Informasi**

Sistem informasi merupakan suatu sistem dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur dan pengendalian untuk mendapatkan jalur komunikasi penting, serta menyediakan suatu dasar informasi untuk pengambilan keputusan. Komponen dalam sistem informasi yaitu [8] :

1. *Hardware* (perangkat keras), terdiri dari komputer, printer dan jaringan serta komponen lainnya.
2. *Software* (perangkat lunak), merupakan kumpulan perintah yang ditulis dengan aturan tertentu untuk memerintahkan komputer untuk melaksanakan tugas tertentu.
3. *Data*, merupakan komponen data yang akan diproses lebih lanjut untuk menghasilkan informasi dan tersimpan dalam jangka waktu tertentu.
4. *Prosedur, instruksi dan kebijaksanaan* untuk mengoperasikan suatu sistem.

### **2.2.2 Presensi**

Presensi merupakan suatu kegiatan pencatatan terhadap setiap kehadiran siswa dan guru dengan tujuan untuk mengetahui data yang berkaitan dengan kehadiran siswa dan guru secara periodik baik harian maupun bulanan. Presensi menjadi bukti otentik kehadiran siswa dalam ikut melaksanakan pembelajaran yang dilaksanakan seorang guru. Presensi dapat menjadi informasi kedisiplinan dan memberi pengaruh terhadap prestasi yang dilakukan sebagai evaluasi dan prestasi belajar yang diraih siswa dan guru [9].

### **2.2.3 Bimbingan Belajar**

Bimbingan belajar merupakan pemberi jasa bimbingan tambahan yang diarahkan kepada siswa untuk memperoleh pendidikan yang sesuai dengan keinginan, minat, bakat, keahliannya dan membantu siswa menentukan solusi belajar yang efektif dan efisien sehingga mendapatkan hasil yang maksimal dalam prestasi belajarnya. Bimbingan belajar

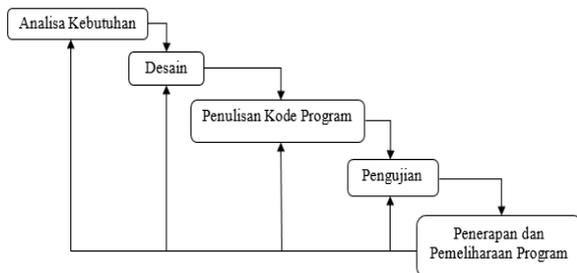
membantu siswa menyelesaikan masalah kesulitan pelajaran disekolahnya dan untuk mengembangkan kemampuan bersosialisasi dengan siswa dari sekolah yang berbeda [6].

#### 2.2.4 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah disiplin teknologi yang menggabungkan konsep ilmu komputer, ekonomi, kemampuan komunikasi serta ilmu manajemen dengan pendekatan pemecahan masalah dari teknik. Salah satu tujuan dasar rekayasa perangkat lunak adalah untuk mengembangkan metode dan prosedur pengembangan perangkat lunak yang dapat meningkatkan sistem yang besar dan dapat digunakan secara konsisten untuk menghasilkan perangkat lunak berkualitas tinggi dengan biaya yang rendah dan menghemat waktu. Sedangkan tujuan utama rekayasa perangkat lunak adalah membantu pengembang mendapatkan perangkat lunak yang berkualitas tinggi [10]. Rekayasa perangkat lunak mempunyai aspek yang mana akan terlibat dalam penelitian ini, diantara yaitu :

#### 2.2.5 Metode Pengembangan Sistem

Sistem ini menggunakan metode *waterfall* yang digunakan untuk pengembangan perangkat lunak. Metode ini dilakukan secara terstruktur dan berurutan, jika tahap pertama belum selesai, maka tahap kedua tidak bisa berjalan dan tidak dapat mengulang ke tahap sebelumnya jika tahap pertama sudah selesai karena saling berkaitan untuk mengurangi terjadinya kesalahan [11].



**Gambar 2. 1** Metode *Waterfall*  
(Sumber : Ian Sommerville, 2011 [12] )

Tahapan-tahapan pengembangan sistem yaitu :

1. Analisa Kebutuhan

Tahap ini merupakan tahapan perencanaan dengan mengumpulkan data-data yang berkaitan dengan sistem yang akan dibangun bisa dengan wawancara atau observasi. Peneliti menganalisa dengan mencari informasi dari klien tentang *software* apa yang sedang dibutuhkan pada tempat penelitian.

2. Desain

Desain perangkat lunak berfokus pada desain pembuatan program perangkat lunak seperti struktur data, arsitektur perangkat lunak, perancangan antarmuka, dan prosedur pengodean. Permasalahan yang ada pada analisa kebutuhan dapat diselesaikan dengan menentukan algoritma dan alur perangkat lunak dari sistem secara keseluruhan yaitu menggunakan perangkat pemodelan sistem seperti diagram ERD, data *flow diagram*.

3. Penulisan Kode Program

Tahap ini merupakan tahapan pembuatan sistem dengan menggunakan kode-kode bahasa pemrograman tertentu yang dapat dikenali komputer. Desain sistem yang telah dibuat lalu diimplementasikan pada kode program. Pengkodean program pada sistem yang dibuat menggunakan bahasa *Framework Codeigniter 3* dan *database MySQL*.

4. Pengujian Program

Pada tahap ini semua kode yang telah dibuat dilakukan evaluasi dengan pengujian program untuk mengetahui apakah sistem yang dibuat sudah layak dan sesuai dengan yang diinginkan klien serta untuk mengidentifikasi adanya kegagalan atau kesalahan sebelum dilakukan evaluasi atau perbaikan sistem. Metode yang digunakan untuk pengujian sistem ini yaitu *Black Box Texting*.

5. Penerapan dan Pemeliharaan Program

Tahap ini sistem sudah selesai dibangun dan dapat dijalankan oleh pengguna. *Software* yang dihasilkan harus sering dipantau agar tetap berjalan sesuai permintaan klien. Pemeliharaan sistem dilakukan apabila klien ingin mengembangkan sistem secara fungsional atau terdapat perubahan apabila sistem terjadi kesalahan [12].

### 2.2.6 Metode *Black Box Testing*

Pengujian *black box* merupakan sebuah metode pengujian software dimana penguji tidak mengetahui internal struktur, desain, dan implementasian dari suatu bagian yang sedang diuji. Pengujian sistem dilakukan untuk melihat apakah sistem yang dibangun sudah sesuai dengan keinginan dari pelanggan dan layak atau digunakan. Tujuan dari metode *blackbox* untuk mengetahui apakah sistem telah benar menampilkan kesalahan yang ada jika terjadi *human error*. Pada pengujian *blackbox* hanya mengambil hasil *output* melalui data uji dan mengecek fungsionalitas dari *software*. Penguji tidak perlu memiliki pengetahuan tentang Bahasa pemrograman hanya saja menguji pada tampilan luar (*interface*) [3].

### 2.2.7 PBO (Pemrograman Berbasis Objek)

Pemrograman Berbasis Objek merupakan sebuah konsep pemrograman yang menggambarkan proses penyelesaian masalah pada program diibaratkan sebagai objek yang saling berinteraksi satu sama lain. Objek pada PBO berupa sekumpulan *field* atau atribut dan metode yang terkait dengan objek tersebut. Bentuk dari objek atau variabel tersebut bersifat acak tergantung program yang akan dibuat. Kelebihan dari PBO yaitu :

1. *Reusable*, kode program lebih efisien seperti kelas dan objek dapat digunakan berulang kali pada skenario yang sama
2. *Extensibility*, pemrogram bisa membuat method baru atau mengubah method yang sudah ada kemudian diubah sesuai keinginannya tanpa harus membuat kode dari awal.
3. *Maintainability*, program yang sudah dibuat akan lebih mudah dikelola jika program yang dibuat dalam skala besar yang memungkinkan terjadinya *error* dalam proses pengembangannya dapat diatasi dengan OOP karena OOP sudah menggunakan konsep modularitas [13].

### 2.2.8 Basis Data

Basis data merupakan suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, Basis data dapat diartikan sebagai sekumpulan data yang disusun dalam bentuk beberapa tabel yang saling memiliki relasi maupun dapat berdiri sendiri. *Database* dapat menentukan kualitas informasi yang akurat, tepat waktu dan keterkaitan serta mengurangi duplikasi data [8].

*Software* yang digunakan untuk mengelola basis data yaitu *Database Management System* (DBMS) merupakan suatu sistem *software* yang memungkinkan seorang *user* dapat mendefinisikan, membuat, dan memelihara serta menyediakan akses terkontrol terhadap data. Bahasa yang digunakan dibagi menjadi dua diantaranya :

1. *Data Definision Language* (DDL)

DDL adalah hasil kompilasi dari perintah DDL adalah satu set dari tabel yang disimpan pada file khusus disebut *data dictionary/directory*.

2. *Data Manipulation Language* (DML)

DML adalah bahasa yang memperbolehkan pengguna untuk mengakses atau memanipulasi data sebagaimana yang telah diorganisasikan sebelumnya dalam model data yang tepat [8].

### 2.2.9 UML

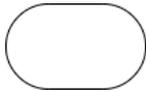
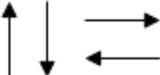
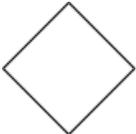
UML (*Unified Modelling Language*) merupakan suatu metode permodelan yang disajikan secara visual sebagai sarana untuk perancangan sistem berorientasi objek. UML sebagai bahasa yang secara umum digunakan pada visualisasi, perancangan, dan juga pendokumentasian sistem *software* [14].

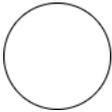
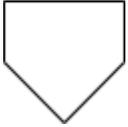
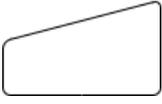
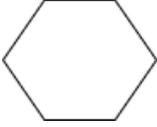
1. *Flowchart*

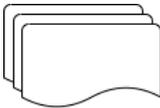
*Flowchart* merupakan alat pemetaan sederhana yang menunjukkan urutan tindakan dalam proses dan bentuk yang mudah dibaca dan dikomunikasikan. *Flowchart* digunakan untuk menjelaskan alur kerja atau urutan prosedur dari suatu program kepada orang lain. *Flowchart* membantu analis dalam memecahkan masalah segmen yang

lebih kecil khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. Simbol-simbol dalam *flowchart* dapat dilihat pada tabel 2.2 dibawah ini [15] :

**Tabel 2. 2** Simbol *Flowchart*

No	Gambar Simbol	Nama	Keterangan
1.		<i>Terminal Point</i>	Simbol yang menyatakan awal atau akhir suatu proses.
2.		<i>Flow direction</i>	Simbol yang menyatakan alur proses kerja dan menghubungkan antar simbol.
3.		<i>Proses</i>	Simbol yang menyatakan suatu proses yang dilakukan oleh komputer.
4.		<i>Decision</i>	Digunakan untuk memilih keputusan dalam program sesuai dengan kondisi tertentu.
5.		<i>Input output</i>	Simbol yang menyatakan <i>input</i> atau <i>output</i> yang tidak tergantung jenis peralatannya.

6.		<i>Predefined process</i>	Menunjukkan pelaksanaan suatu bagian prosedur, bagian prosedur yang terinformasi belum detail dan akan diperinci ditempat lain.
7.		<i>Connector (On-page)</i>	Simbol yang menghubungkan dengan proses lainnya dalam satu halaman yang sama.
8.		<i>Connector (Off-page)</i>	Simbol yang menghubungkan dengan proses lainnya dalam halaman yang berbeda.
9.		<i>Manual Input</i>	Simbol memasukkan data secara manual menggunakan <i>online keyboard</i> .
10.		<i>Preparation</i>	Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.

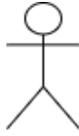
11.		<i>Manual Operation</i>	Menunjukkan proses/kegiatan yang dilakukan tanpa menggunakan komputer.
12.		<i>Document</i>	Mencetak keluaran dalam bentuk dokumen (melalui <i>printer</i> ).
13.		<i>Multiple document</i>	Sama seperti simbol dokument hanya saja dokumen yang digunakan lebih dari satu.
14.		<i>Display</i>	Mencetak keluaran dalam layar monitor
15.		<i>Delay</i>	Menunjukkan bahwa adanya proses <i>delay</i> .

## 2. Use Case Diagram

Salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor. *Use case* diagram dapat mendeskripsikan tipe interaksi antara pemakai sistem dengan sistemnya. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. interaksi antara sistem dan aktor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah

sistem dipakai. Simbol-simbol dalam *use case diagram* dapat dilihat pada tabel 2.3 dibawah ini [14] :

**Tabel 2. 3** Simbol *Use Case Diagram*

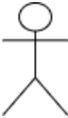
No	Gambar Simbol	Nama	Keterangan
1.		<i>Use Case</i>	Deskripsi urutan dari aksi-aksi yang ditampilkan sistem bagi suatu aktor.
2.		Aktor	Aktor merupakan orang, sistem atau perangkat keras yang akan berhubungan dengan <i>use case</i> .
3.		<i>Extend</i>	Menunjukkan bahwa <i>use case</i> target merupakan tambahan fungsionalitas dari <i>use case</i> lainnya.
4.		<i>Association</i>	Penghubung antara aktor dan <i>use case</i> .
5.		<i>Include</i>	Menunjukkan sebuah <i>use case</i> memerlukan <i>use case</i> yang lain untuk menjalankan fungsinya.

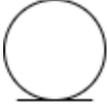
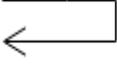
### 3. *Sequence Diagram*

Salah satu jenis diagram pada UML yang menjelaskan interaksi berdasarkan waktu. dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk menghasilkan sesuatu seperti pada *use case*

*diagram*. biasa digunakan untuk menggambarkan rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah kejadian untuk menghasilkan *output* tertentu. Simbol-simbol dalam dapat dilihat pada tabel 2.4 dibawah ini [14] :

**Tabel 2. 4** Simbol *Sequence Diagram*

No	Gambar Simbol	Nama	Keterangan
1.		<i>Object</i>	Menggambarkan sebuah <i>class</i> atau <i>object</i> .
2.		Aktor	Menggambarkan pengguna yang berinteraksi dengan sistem.
3.		<i>Lifeline</i>	Menggambarkan garis titik-titik yang terhubung dengan objek.
4.		<i>Activation boxes</i>	Menggambarkan panjang waktu yang dibutuhkan sebuah objek dalam mengerjakan tugasnya.

5.		<i>Boundary</i>	Digunakan untuk menggambarkan sebuah <i>form</i> .
6.		<i>Control</i>	Digunakan untuk menghubungkan <i>boundary</i> dengan tabel.
7.		<i>Entity Class</i>	Menggambarkan sistem awal yang menjadi landasan untuk menyusun basis data.
8.		<i>Message</i>	Menggambarkan pesan atau interaksi antar objek.
9.		<i>Message to self</i>	Menggambarkan pesan balikan atau reaksi dari objek sebelumnya.
10.		<i>Message return</i>	Menggambarkan hasil dari pengiriman <i>message</i> .

#### 4. *Activity Diagram*

*Activity diagram* menggambarkan sistem yang akan dirancang agar *workflow* (aliran kerja) pada sistem perangkat lunak dapat mudah dipahami. Yang perlu diperhatikan adalah bahwa *activity diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor. Simbol-simbol dalam *activity diagram* dapat dilihat pada tabel 2.5 dibawah ini.

**Tabel 2. 5** Simbol *Activity Diagram*

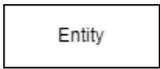
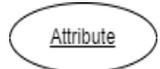
Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
<i>Action</i> 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabunga dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
<i>Swinline</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

### 2.2.10 Entity Relationship Diagram (ERD)

*ERD* merupakan sebuah teknik pendekatan yang menggambarkan hubungan antara model. Dalam hubungan tersebut *ERD* menunjukkan objek yang berupa data (*Entity*) dan hubungan (*Relationship*) yang ada pada *Entity* berikutnya. Menurut Simarmata, *ERD* adalah suatu pemodelan data yang membantu untuk mengatur data dalam suatu proyek

ke dalam entitas - entitas dan menentukan hubungan entitas tersebut. Simbol-simbol dalam ERD dapat dilihat pada tabel 2.6 dibawah ini [14]:

**Tabel 2.6** Simbol *ERD*

No	Gambar Simbol	Nama	Keterangan
1.		<i>Entity</i>	Menyatakan himpunan entitas yang perlu disimpan dalam <i>database</i> .
2.		<i>Relationship</i>	Digunakan untuk menghubungkan entitas yang ada.
3.		<i>Atribut</i>	Entitas atau relasi yang menyediakan penjelasan detail tentang entitas atau relasi.
4.		<i>Atribut Multivalue</i>	Atribut yang memiliki nilai lebih dari satu setiap baris data.
5.		<i>Atribut Primary Key</i>	Atribut yang menjadi kunci utama dan tidak boleh sama dengan atribut yang lain.
6.		<i>Association</i>	Menghubungkan atribut dengan entitas dan entitas dengan relasi.

### 2.2.11 Framework

*Framework* adalah kumpulan intruksi-intruksi program yang disatukan dalam class dan fungsi masing-masing untuk memudahkan *developer* dalam pemanggilannya tanpa harus menuliskan *syntax program* yang sama dan berulang-ulang. Kegunaan *framework* untuk menghemat waktu dan mencegah penulisan *syntax* secara berulang-ulang sehingga terbentuk *source code* yang bersih dan terstruktur [16].

#### 1. CodeIgniter

*CodeIgniter* adalah sebuah *Framework PHP* yang bersifat open source dan menggunakan metode pemisahan komponen utama MVC (*Model, View, Controller*) untuk memudahkan *Developer* atau *programmer* dalam membangun sebuah aplikasi berbasis *website* tanpa harus membuatnya dari awal. CI memiliki *library* yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh aplikasi berbasis web seperti mengakses *database*, memvalidasi form sehingga sistem yang dikembangkan dapat dibangun dengan mudah. CI juga merupakan *Framework* dengan dokumentasi yang lengkap dan jelas. *Source code CI* dilengkapi dengan *comment* sehingga dapat memperjelas fungsi sebuah kode program dan CI yang dihasilkan sangat bersih (*clean*) dan *search Engine Friendly* (SEF). Selain itu, struktur dan susunan yang logis dari *codeigniter* membuat aplikasi menjadi semakin teratur dan dapat terfokuskan pada fitur-fitur apa yang akan dibutuhkan dalam pembuatan aplikasi tersebut [17].

### 2.2.12 Pengujian Usability

*Usability* adalah ukuran kualitas untuk menilai bagaimana kemudahan antarmuka pengguna untuk digunakan. *Usability* terdiri dari lima komponen kualitas yang terdapat dalam pertanyaan kuisisioner adalah sebagai berikut:

1. *Learnability*, adalah tingkat kemudahan bagi seorang pengguna sistem yang dapat dilihat berdasarkan pemakaian fitur dan fungsi yang terdapat dalam sistem
2. *Efficiency*, adalah tingkat kecepatan seorang pengguna dalam melakukan fungsi-fungsi yang terdapat dalam sistem

3. *Memorability*, adalah tingkat kemampuan seorang pengguna dalam mengingat dan mempertahankan pengetahuannya setelah beberapa waktu tidak mengoperasikan sistem
4. *Errors*, adalah tingkat atau jumlah error atau kesalahan yang dilakukan oleh seorang pengguna sistem, tingkat kebosanan dalam menghadapi error, dan cara memperbaikinya
5. *Satisfaction*, adalah tingkat kepuasan pengguna setelah berinteraksi dengan sistem

Setiap pertanyaan pada pengujian *usability* menggunakan skala *Likert*, dan memiliki jawaban 1-4. Skala *likert* digunakan untuk mengukur pendapat, persepsi, atau sikap dari perorangan atau sekelompok orang terhadap fenomena sosial tertentu. Adapun jawaban setiap pertanyaan yang digunakan dapat dinilai dengan skor seperti tabel 2.7.

**Tabel 2. 7** Skala *Likert*

Jawaban	Skor
Sangat tidak setuju	1
Tidak setuju	2
Setuju	3
Sangat setuju	4

### Rumus Untuk Menghitung Pengujian Kuisioner *Usability*

$$U = \frac{np}{nm} \times 100\%$$

Keterangan:

U = Nilai hasil uji *usability*

Np = Nilai perolehan seluruh hasil kuisioner atau jawaban reponden

Nm = Nilai maksimal yang diperoleh (jumlah responden x nilai maksimal skala *Likert* yang dipakai x jumlah pertanyaan) [18]

**Tabel 2. 8** Predikat Hasil *Usability*

Presensi hasil uji <i>usability</i>	Predikat
81 - 100 %	Sangat baik
61 - 80 %	Baik
41 - 60 %	Cukup baik
21 - 40 %	Kurang baik
0 - 20 %	Tidak baik