

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Pada penelitian ini, penulis melakukan tinjauan pustaka pada penelitian sebelumnya yang berkaitan dengan judul dan pokok bahasan pada penelitian.

Adapun tinjauan pustaka yang digunakan dapat dilihat dibawah ini :

**Tabel 2. 1** Tinjauan Pustaka

<b>Literatur</b>	Literatur 1
<b>Judul</b>	Pembuatan Sistem Informasi Pengarsipan Surat Pada Kantor Desa Sandik Berbasis Website, (2021)
<b>Penulis</b>	Ande Rizky Riefnaldi, Arik Aranta, Muhammad Muaidi
<b>Metode</b>	Waterfall, User Acceptance Testing
<b>Masalah</b>	Pengarsipan surat masih dilakukan secara manual pada kantor pemerintah daerah yang terletak di Desa Sandik, Lombok Barat, Nusa Tenggara Barat
<b>Literatur</b>	Literatur 2
<b>Judul</b>	Sistem Informasi Kearsipan Menggunakan Framework Laravel (Studi Kasus: Prodi Sistem Informasi Universitas Peradaban), (2022)
<b>Penulis</b>	Maisila Cendani, Danar Ardian Pramana, Eko Sudrajat
<b>Metode</b>	Prototype, Observasi, Wawancara
<b>Masalah</b>	Kearsipan di Universitas Peradaban masih menggunakan cara konvensional dan belum terkomputerisasi, sehingga membutuhkan waktu yang cukup lama dalam melakukan proses pengelolaan Arsip dan datapun rentan hilang
<b>Literatur</b>	Literatur 3
<b>Judul</b>	Perancangan Program E-Arsip Berbasis Website Di Kelurahan Rawa Bunga Jakarta Timur, (2023)
<b>Penulis</b>	Geri Darmawan, Ahmad Al Kaafi
<b>Metode</b>	Waterfall, Observasi, Wawancara
<b>Masalah</b>	Pengelolaan dokumen di Kelurahan Rawa Bunga Jakarta Timur masih mengalami masalah terkait efisiensi, keandalan, dan keamanan.
<b>Literatur</b>	Literatur 4
<b>Judul</b>	Perancangan Sistem Informasi E-Arsip Dokumen di Bappelitbang Kota Bandung Berbasis PHPRAD, (2023)
<b>Penulis</b>	Firman Ibrahim, Philip Broos, Candra Mecca Susyana, Muthmainnah

<b>Metode</b>	SDLC
<b>Masalah</b>	Bappelitbang Kota Bandung khususnya di Bidang Perencanaan Pembangunan Ekonomi Dan Pembiayaan Pembangunan pada proses pengarsipan dokumen yang masih terbilang manual yang disimpan pada tempat pengarsipan
<b>Literatur</b>	Literatur 5
<b>Judul</b>	E-Arsip Persuratan Sekretariat Daerah Kabupaten Luwu, (2023)
<b>Penulis</b>	Dianradika Prasti, Muhammad Idham Rusdi, Ratnasari Kamaruddin, dan Siti Jamilah Br Tarigan
<b>Metode</b>	Waterfall, R&D
<b>Masalah</b>	Meningkatnya jumlah surat masuk dan surat keluar juga meningkatkan pengelolaan surat masuk dan surat keluar, namun pengelolaan surat masuk dan surat keluar selama ini dilakukan secara manual. Ada beberapa masalah dengan pengelolaan surat masuk dan surat keluar.

### 2.1.1 Literatur 1

Pada literatur yang berjudul “Pembuatan Sistem Informasi Pengarsipan Surat Pada Kantor Desa Sandik Berbasis Website” pada tahun 2021 dengan penulis Ande Rizky Riefnaldi, Arik Aranta dan Muhammad Muaidi mengangkat sebuah permasalahan yaitu Kantor Desa Sandik merupakan kantor milik pemerintah daerah yang terletak di Desa Sandik, Lombok Barat, Nusa Tenggara Barat yang melayani masyarakat dalam melakukan semua hal yang berkaitan dengan urusan pemerintahan. Pada Kantor Desa Sandik ini memiliki cukup banyak data dalam hal pengarsipan surat, baik surat masuk maupun surat keluar. Namun pengarsipan surat masih dilakukan secara manual sehingga dalam jangka waktu panjang akan menyebabkan terjadinya penumpukan data arsip yang ada, baik itu surat masuk dan surat keluar (Riefnaldi et al., 2021).

Sehingga penulis membuat sebuah sistem informasi pengarsipan surat pada Kantor Desa Sandik dengan menggunakan metode pengembangan *waterfall*,

*framework bootstrap* untuk *web design*. Pengujian yang dilakukan penulis mendapatkan persentase 100% dari 5 responden.

### **2.1.2 Literatur 2**

Pada literatur yang berjudul “Sistem Informasi Kearsipan Menggunakan Framework Laravel (Studi Kasus: Prodi Sistem Informasi Universitas Peradaban)” pada tahun 2022 dengan penulis Maisila Cendani, Danar Ardian Pramana dan Eko Sudrajat mengangkat sebuah permasalahan yaitu Kearsipan di Universitas Peradaban masih menggunakan cara konvensional dan belum terkomputerisasi, sehingga membutuhkan waktu yang cukup lama dalam melakukan proses pengelolaan Arsip dan datapun rentan hilang (Cendani et al., 2023).

Sehingga penulis membuat sebuah sistem informasi kearsipan dengan menggunakan metode pengembangan *prototype*, *framework laravel* untuk pembuatan *website*. Pengujian aplikasi yang dilakukan penulis menggunakan *Blackbox Testing* dan *User Acceptance Test* menyatakan bahwa aplikasi sudah berjalan dengan sesuai fungsi.

### **2.1.3 Literatur 3**

Pada literatur yang berjudul “Perancangan Program E-Arsip Berbasis Website Di Kelurahan Rawa Bunga Jakarta Timur” pada tahun 2023 dengan penulis Geri Darmawan dan Ahmad Al Kaafi mengangkat sebuah permasalahan yaitu pengelolaan dokumen di Kelurahan Rawa Bunga Jakarta Timur masih mengalami masalah terkait efisiensi, keandalan, dan keamanan. Tanggung jawab administrasi yang semakin kompleks membuat proses pengarsipan dan pencarian dokumen konvensional yang dilakukan secara manual tidak efektif lagi. Kelurahan Rawa Bunga mengalami peningkatan jumlah dokumen yang perlu dikelola, termasuk

surat permohonan, keputusan, pemberitahuan, dan catatan lainnya (Darmawan & Al Kaafi, 2023).

Sehingga penulis membuat sebuah sistem e-arsip surat pada kantor Kelurahan Rawa Bunga, Jakarta Timur dengan menggunakan metode pengembangan *waterfall*.

#### **2.1.4 Literatur 4**

Pada literatur yang berjudul “Perancangan Sistem Informasi E-Arsip Dokumen di Bappelitbang Kota Bandung Berbasis PHPRAD” pada tahun 2023 dengan penulis Firman Ibrahim, Philip Broos, Candra Mecca Susyana dan Muthmainnah mengangkat sebuah permasalahan yaitu Bappelitbang Kota Bandung khususnya di Bidang Perencanaan Pembangunan Ekonomi Dan Pembiayaan Pembangunan yaitu pada proses pengarsipan dokumen yang masih terbilang manual yang disimpan pada tempat pengarsipan. Menurut penulis Hal ini tentu saja tidak lagi efektif dan efisien dari sisi waktu dan tempat dikarenakan dokumen arsip akan mengalami penambahan dari waktu ke waktu (Ibrahim et al., 2023).

Sehingga penulis membuat sebuah aplikasi e-arsip pada Bappelitbang Kota Bandung dengan menggunakan metode pengembangan *software development life cycle*. Pengujian yang dilakukan penulis untuk menguji aplikasi menggunakan *software PHPRAD* dan *Microsoft Exel 2013*.

#### **2.1.5 Literatur 5**

Pada literatur yang berjudul “E-Arsip Persuratan Sekretariat Daerah Kabupaten Luwu” pada tahun 2023 dengan penulis Dianradika Prasti, Muhammad Idham Rusdi, Ratnasari Kamaruddin dan Siti Jamilah Br Tarigan mengangkat sebuah permasalahan yaitu Meningkatnya jumlah surat masuk dan surat keluar juga

meningkatkan pengelolaan surat masuk dan surat keluar, namun pengelolaan surat masuk dan surat keluar selama ini dilakukan secara manual. Ada beberapa masalah dengan pengelolaan surat masuk dan surat keluar seperti ini. Misalnya, dengan bertambahnya jumlah surat masuk dan surat keluar, arsip dibuat, sehingga sulit untuk menemukan data arsip lama (Prasti et al., 2023).

Dikarenakan Arsip memegang peranan penting di Kantor Bupati Luwu yang merupakan bagian dari Organisasi Sekretariat Daerah, sehingga penulis membuat sebuah sistem informasi administrasi pada Kantor Bupati Kabupate Luwu dengan menggunakan metode pengembangan *waterfall*.

Setelah melakukan tinjauan literatur terhadap penelitian yang telah ada, penulis membangun Sistem Informasi E-Filling dengan fitur *Advance Search* dan *Responsive Design*. Penelitian yang dilakukan oleh penulis ini memiliki perbedaan signifikan dibandingkan dengan penelitian sebelumnya, yaitu:

1. Sistem informasi yang akan dibuat terdapat fitur *Advance Search* dan *Responsive Design*.
2. Metode pengujian aplikasi yang dipakai menggunakan ISO 25010.
3. Skala pengukuran yang dipakai menggunakan Skala Likert.

## **2.2 Rancang Bangun**

Menurut (Surahmat, 2023) “rancang bangun merupakan proses mengembangkan dan memperbaiki sistem atau aplikasi yang sudah ada ataupun belum ada dengan beberapa komponen yang digunakan dari hasil proses analisa sistem dimana mengharapkan dapat menghasilkan aplikasi yang berkualitas sesuai dengan kebutuhan yang diinginkan”. (Putra et al., 2020) menyatakan bahwa “rancang bangun adalah proses pembangunan sistem untuk menciptakan sistem

baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun hanya sebagian”. Dengan demikian Rancang bangun merupakan proses mengembangkan dan memperbaiki sistem atau aplikasi, baik yang sudah ada maupun yang belum ada, dengan menggunakan beberapa komponen hasil proses analisa sistem. Tujuan dari rancang bangun ini adalah untuk menghasilkan aplikasi yang berkualitas dan sesuai dengan kebutuhan yang diinginkan. Dalam prosesnya, rancang bangun melibatkan berbagai tahap, mulai dari perencanaan, perancangan, implementasi, hingga pengujian dan evaluasi untuk memastikan bahwa aplikasi yang dihasilkan dapat berfungsi dengan baik dan efisien.

Proses rancang bangun ini sangat penting dalam pengembangan teknologi informasi, karena memastikan bahwa setiap komponen sistem terintegrasi dengan baik dan dapat bekerja sesuai dengan tujuan yang telah ditetapkan. Dengan melalui proses analisa yang mendalam, kebutuhan pengguna dapat diidentifikasi dan diakomodasi secara tepat dalam desain aplikasi. Hasil akhirnya adalah sebuah aplikasi yang tidak hanya memenuhi spesifikasi teknis, tetapi juga memberikan nilai tambah bagi pengguna, meningkatkan produktivitas, dan mempermudah berbagai tugas.

### **2.3 Sistem *E-Filing***

*Electronic Filing System (EFS)* adalah solusi manajemen dokumen elektronik yang dirancang untuk mengoptimalkan penyimpanan dan pengelolaan dokumen dalam bentuk digital. Sistem ini mencakup semua perangkat dan aplikasi yang diperlukan untuk mendukung pengelolaan dokumen secara efektif, yang dijabarkan dalam konsep solusi. Implementasi EFS dimulai dengan proses konversi dokumen

kertas menjadi dokumen elektronik, atau yang biasa disebut sebagai document imaging. Proses ini menggunakan berbagai alat untuk menyimpan data dalam bentuk data terstruktur (data teks) dan format data tidak terstruktur (gambar dan file elektronik lainnya). Dengan menggunakan EFS, data dapat disimpan dengan cara yang lebih terorganisir dan efisien. Sistem integrasi ini memungkinkan pengguna untuk mencari dan mengakses dokumen dengan lebih mudah dan cepat, meningkatkan produktivitas dan mengurangi waktu yang diperlukan untuk menemukan informasi yang dibutuhkan. Selain itu, EFS membantu mengurangi ketergantungan pada dokumen fisik, mengurangi risiko kehilangan atau kerusakan dokumen, serta menghemat ruang penyimpanan. Dengan demikian, EFS menjadi alat penting dalam modernisasi manajemen dokumen di berbagai organisasi, memastikan bahwa semua dokumen dapat diakses dengan cepat dan aman kapan saja dibutuhkan (Priambodo & Juraini, 2023).

#### **2.4 Website**

Menurut yang dikutip dari jurnal (Dhaniswara et al., 2023) “*Website* merupakan platform yang berisikan informasi yang dapat dibaca dan mudah untuk diakses semua orang”. Menurut yang dikutip dari jurnal (Putra et al., 2020) “*Website* diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa text, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet”. Dapat disimpulkan bahwa *Website* sebagai sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital, baik berupa teks, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet. *Website* merupakan *platform* yang berisikan informasi yang dapat dibaca dan mudah untuk

diakses oleh semua orang. Sebagai medium digital, *website* memungkinkan pengguna untuk mengakses berbagai jenis konten dan informasi dari mana saja dan kapan saja selama mereka memiliki koneksi internet. Hal ini membuat *website* menjadi alat yang sangat efektif untuk menyebarkan informasi secara luas dan cepat. Informasi yang disajikan di *website* bisa sangat bervariasi, mulai dari berita, artikel, produk, layanan, hingga hiburan. Keberadaan *website* juga memungkinkan interaksi antara penyedia informasi dan pengguna melalui fitur-fitur seperti komentar, formulir kontak, dan media sosial. Selain itu, *website* dapat diakses melalui berbagai perangkat, termasuk komputer, tablet, dan *smartphone*, sehingga memudahkan pengguna dalam mengakses informasi sesuai kebutuhan mereka. Dengan desain yang responsif dan navigasi yang intuitif, *website* dapat memberikan pengalaman pengguna yang optimal, membantu mereka menemukan informasi yang dicari dengan cepat dan efisien. Semua aspek ini menjadikan *website* sebagai salah satu alat komunikasi dan informasi paling penting di era digital ini.

## **2.5 Advance Search**

Menurut (Nafisah, 2022) “*Advanced Search* adalah alat yang dirancang untuk membantu pengguna dalam mencari informasi sesuai dengan kriteria kata kunci yang ditentukan”. Menurut (Nasrudiansyah & Sukmono, 2023) “*Advanced Search* ialah alat untuk membantu mencari sesuai kriteria kata kunci konten”. Menurut (Lityaningrum, 2021) “*Advance Search* adalah fitur untuk mendapatkan data dengan kategori yang ditentukan oleh pengguna”. Dapat disimpulkan dari literatur yang ada bahwa *Advanced Search* adalah sebuah alat yang dirancang khusus untuk membantu pengguna dalam mencari informasi dengan menggunakan kriteria kata kunci yang telah ditentukan. Dengan fitur ini, pengguna dapat dengan mudah






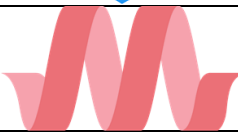



mencari konten tertentu yang relevan dengan kata kunci yang mereka masukkan, mempermudah proses penemuan informasi yang spesifik dan tepat sasaran. Selain itu, *Advanced Search* juga berfungsi sebagai fitur untuk mengumpulkan data dengan kategori yang dipilih langsung oleh pengguna. Dengan demikian, pengguna memiliki kontrol penuh untuk menyesuaikan pencariannya sesuai dengan kebutuhan mereka, baik itu untuk tujuan penelitian, penemuan informasi terperinci, atau untuk mendapatkan data dengan spesifikasi tertentu. Fitur ini membantu meningkatkan efisiensi pengguna dalam mengeksplorasi dan mengakses informasi yang relevan secara cepat dan efektif, tanpa harus melalui proses pencarian manual yang rumit atau tidak efisien.

## **2.6 Responsive Web Design**

Menurut (Regal & Widiyari, 2020) “*Responsive web design* merupakan pendekatan yang menyarankan bahwa desain dan pengembangan *website* harus memperhatikan tingkah laku dan lingkungan dari pengakses berdasarkan ukuran layar, *platform*, dan orientasi dari *device* yang digunakan”. Menurut (Setiawan & Widodo, 2021) “*Web Responsive*, yaitu merupakan salah satu teknik yang dapat membuat proses perancangan aplikasi dan situs *web* untuk berbagai jenis perangkat menjadi lebih mudah”. Hal ini dikarenakan bahwa dengan menggunakan *responsive design*, perancang dimungkinkan untuk dapat menerapkan solusi bagi berbagai resolusi layar, *density*, dan rasio aspek pada banyak jenis perangkat. Dengan demikian dapat disimpulkan bahwa *Responsive web design* adalah pendekatan yang menyarankan bahwa desain dan pengembangan *website* harus mempertimbangkan perilaku dan lingkungan pengguna berdasarkan ukuran layar, *platform*, dan

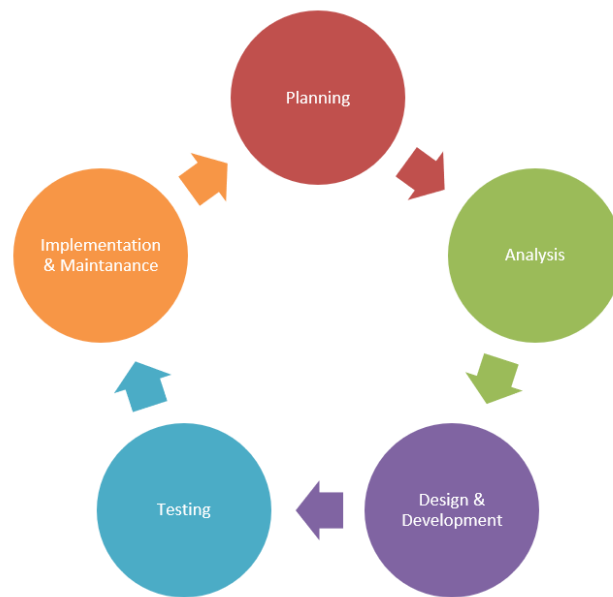
orientasi perangkat yang digunakan. Terdapat beberapa *framework* untuk *Web Design* yang dapat dilihat sebagai berikut:

**Tabel 2. 2** Tabel *Framework Web Design*

No	Logo	Nama <i>Framework</i>
1		<i>Bootstrap</i>
2		<i>Semantic UI</i>
3		<i>UI Kit</i>
4		<i>Materialize CSS</i>
5		<i>Tailwind CSS</i>
6		<i>Pure CSS</i>
7		<i>Miligram CSS</i>

## 2.7 Metode *Web Development Life Cycle* (WDLC)

Menurut (Hisham, 2021) metode “*Web Development Life Cycle* yaitu metode yang digunakan dalam membangun sebuah *website*”. Menurut (Yudianto & Sulisty, 2022) metode “*Web Development Life Cycle* merupakan sebuah proses pengembangan atau perancangan dalam pembuatan sebuah *website* dan dapat menggambarkan proses yang akan disampaikan kepada pengguna”.



**Gambar 2.1** Fase WDLC

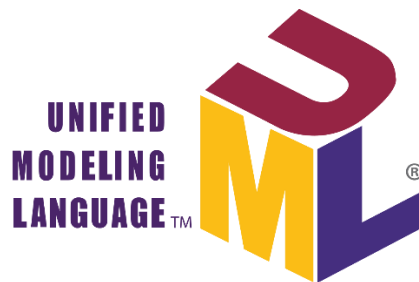
Sumber : (Yudianto & Sulisty, 2022)

Dapat disimpulkan bahwa Metode *Web Development Life Cycle* merupakan sebuah proses pengembangan atau perancangan dalam pembuatan sebuah website dan dapat menggambarkan proses yang akan disampaikan kepada pengguna. Metode ini mencakup serangkaian tahapan yang digunakan dalam membangun sebuah *website*, yang dapat dilihat sebagai berikut :

- a. *Planning*, perencanaan awal hingga peluncuran dan pemeliharaan. Dalam tahap awal, kebutuhan pengguna diidentifikasi.
- b. *Analysis*, analisis untuk memastikan bahwa *website* yang dibangun akan memenuhi kebutuhan dan harapan mereka. Setelah itu,
- c. *Design*, desain visual dan arsitektur informasi dibuat untuk memberikan gambaran tentang tata letak dan navigasi *website*.
- d. *Development*, Tahap pengembangan melibatkan penulisan kode dan integrasi sistem, di mana semua elemen yang direncanakan sebelumnya diwujudkan dalam bentuk nyata.

- e. *Testing*, Pengujian dilakukan untuk memastikan bahwa website berfungsi dengan baik dan bebas dari bug atau kesalahan.
- f. *Implementation*, Setelah website siap dilakukan peluncuran di mana *website* diperkenalkan kepada pengguna.
- g. *Maintenance*, Tahap terakhir adalah pemeliharaan, di mana *website* terus dipantau dan diperbarui untuk memastikan tetap relevan dan berfungsi dengan baik. Dengan mengikuti metode *Web Development Life Cycle*, pengembang dapat memastikan bahwa setiap langkah dalam pembuatan *website* dilakukan secara sistematis dan efektif, sehingga menghasilkan *website* yang berkualitas tinggi dan *user-friendly*.

## 2.8 Unified Modeling Language (UML)



**Gambar 2. 2** Logo UML

Sumber : (Kurniasih, 2024)

Menurut (Sumiati et al., 2021) “*Unified Modeling Language (UML)* adalah sebuah bahasa pemodelan perangkat lunak yang telah distandardisasi sebagai media penulisan cetak biru (*blueprints*) perangkat lunak (Pressman). UML bisa saja digunakan untuk visualisasi, spesifikasi, konstruksi dan dokumentasi beberapa bagian-bagian dari system yang ada dalam perangkat lunak”. Menurut (Andraini & Bella, 2022) “*Unified Modeling Language (UML)* adalah salah satu standar bahasa

yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”. Dengan demikian dapat disimpulkan *Unified Modeling Language* (UML) adalah bahasa pemodelan perangkat lunak yang telah distandardisasi sebagai media penulisan cetak biru (*blueprints*) perangkat lunak. UML digunakan untuk visualisasi, spesifikasi, konstruksi, dan dokumentasi berbagai bagian dari sistem perangkat lunak. Sebagai salah satu standar bahasa yang banyak digunakan di dunia industri, UML berperan penting dalam mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Dengan UML, pengembang perangkat lunak dapat membuat representasi visual dari struktur dan perilaku sistem, yang memudahkan komunikasi dan pemahaman antar tim pengembang serta pemangku kepentingan.

Implementasi UML membantu memastikan bahwa semua aspek penting dari sistem perangkat lunak telah dianalisis dan didokumentasikan dengan baik, sehingga meminimalkan kesalahan dan mempercepat proses pengembangan. Pada penelitian ini penulis menggunakan diagram UML diantaranya sebagai berikut:





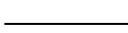
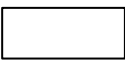
1. *Use Case Diagram*, memodelkan proses bisnis.
2. *Class Diagram*, memodelkan struktur kelas.
3. *Activity Diagram*, memodelkan perilaku *use case* dan *objects* di dalam *system*.

### **2.8.1 Use Case Diagram**

*Use Case* adalah model pemodelan untuk perilaku sistem informasi yang akan dibangun. Mereka menggambarkan interaksi antara aktor (pengguna atau sistem eksternal) dengan sistem itu sendiri. Fungsinya adalah untuk

mengidentifikasi dan menggambarkan fungsi-fungsi yang dapat diakses oleh aktor-aktor tersebut. Dengan menggunakan *Use Case*, pengembang dan analis sistem dapat memahami secara detail bagaimana setiap aktor berinteraksi dengan sistem, serta fungsi apa saja yang tersedia dan bagaimana mereka saling terkait. Ini membantu memastikan bahwa sistem yang dikembangkan akan memenuhi kebutuhan pengguna secara efektif dan efisien. Simbol *use case* dapat dilihat sebagai berikut:

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

No	Nama	Simbol	Keterangan
1	<i>Actor</i>		Segala sesuatu yang berinteraksi dengan sistem aplikasi komputer.
2	<i>Extends</i>		Relase <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
3	<i>Include</i>		Relasi <i>use case</i> tambahan kesebuah <i>use case</i> yang ditambahkan melalui <i>use case</i> ini untuk menjalankan fungsinya untuk sebagai syarat dijalankan <i>use case</i> ini.
4	<i>Use case</i>		Menjelaskan yang dilakukan aktor dari sistem untuk mencapai tujuan tertentu.
5	<i>Association</i>		Apa yang menghubungkan antara objek satu dengan objek lainnya.
6	<i>System</i>		Menspesifikasikan data paket yang menampilkan system secara terbatas


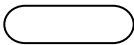



Sumber : (Dias & Muhallim, 2022)

### 2.8.2 *Activity Diagram*

*Activity diagram* atau diagram aktivitas adalah alat visual yang digunakan untuk menggambarkan workflow atau aliran kerja dalam sebuah sistem, proses bisnis, atau menu yang ada dalam perangkat lunak. Penting untuk dicatat bahwa diagram aktivitas menggambarkan aktivitas yang dilakukan oleh sistem itu sendiri,

bukan aktivitas yang dilakukan oleh aktor atau pengguna. Hal ini memungkinkan para pengembang untuk memahami secara jelas bagaimana berbagai aktivitas dalam sistem berinteraksi dan saling terkait, sehingga mempermudah analisis dan pengembangan untuk mencapai efisiensi dan tujuan yang diinginkan dalam desain perangkat lunak tersebut. Simbol *activity diagram* dapat dilihat pada tabel berikut:

**Tabel 2. 4** Tabel Simbol *Activity Diagram*

No	Nama	Simbol	Keterangan
1	<i>Activity</i>		Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2	<i>Action</i>		State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3	<i>Initial Node</i>		Bagaimana objek dibentuk atau diawali.
4	<i>Activity Final Node</i>		Bagaimana objek dibentuk dan dihancurkan
5	<i>Fork Node</i>		Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

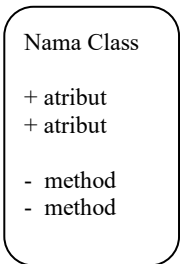
Sumber : (Dias & Muhallim, 2022)

### 2.8.3 *Class Diagram*

*Class diagram* adalah jenis diagram yang digunakan untuk menggambarkan struktur sistem dari segi definisi kelas-kelas yang dibutuhkan dalam pembangunan sistem. Kelas dalam konteks ini merupakan kumpulan objek-objek yang memiliki struktur, perilaku, relasi, dan semantik yang serupa. Diagram ini membantu para pengembang dan analis sistem untuk memvisualisasikan secara jelas bagaimana kelas-kelas saling terkait dan berinteraksi dalam sistem yang akan dibangun, serta menentukan atribut dan metode yang dimiliki oleh setiap kelas. Dengan menggunakan class diagram, tim pengembang dapat merancang struktur sistem

secara sistematis dan efisien, sehingga memastikan bahwa sistem yang dikembangkan sesuai dengan kebutuhan dan tujuan yang telah ditetapkan. Simbol *class diagram* dapat dilihat pada tabel berikut:

**Tabel 2. 5** Tabel Simbol *Class Diagram*

No	Nama	Simbol	Keterangan
1	<i>Dependency</i>	----->	Penggunaan dependency digunakan untuk menunjukkan operasi pada suatu class yang menggunakan class yang lain.
2	<i>Class</i>		<i>Class</i> adalah blok-blok pembangun pada pemrograman berorientasi objek. <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan atribut <i>class</i> . Bagian akhir mendefinisikan method-method
3	<i>Association</i>	_____	Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 class dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 class. Garis ini bisa melambangkan tipe-tipe relationship dan juga dapat menampilkan hukum - hukum multiplisitas pada sebuah relationship

Sumber : (Malius et al., 2021)

## 2.9 Framework Codeigniter

*Codeigniter* adalah sebuah aplikasi open source yang berupa kerangka kerja atau framework untuk membangun website menggunakan bahasa pemrograman PHP. Framework ini dirancang untuk memudahkan dan mempercepat proses pengembangan aplikasi web. Dengan menggunakan *Codeigniter*, pengembang tidak perlu menulis kode dari awal untuk setiap proyek karena framework ini menyediakan banyak library dan fungsi siap pakai yang biasanya digunakan dalam pengembangan aplikasi web. Library yang disediakan oleh *Codeigniter* mencakup



berbagai aspek penting dalam pengembangan web, seperti koneksi database, pengelolaan sesi, validasi form, pengaturan URL, dan banyak lagi. Hal ini memungkinkan pengembang untuk fokus pada logika bisnis dan fungsionalitas aplikasi daripada memikirkan detail teknis yang berulang.



**Gambar 2. 3** Logo CodeIgniter

Selain itu, *Codeigniter* juga dikenal karena performanya yang cepat dan ukuran yang ringan, menjadikannya pilihan populer bagi pengembang yang membutuhkan framework yang efisien. Dengan dokumentasi yang lengkap dan komunitas pengguna yang aktif, *Codeigniter* menawarkan dukungan yang luas bagi pengembang dari berbagai tingkat keahlian. Kesederhanaan dalam struktur dan penggunaan framework ini juga membuatnya cocok bagi pemula yang ingin mempelajari pengembangan aplikasi web dengan PHP. Pada akhirnya, *Codeigniter* membantu pengembang untuk menciptakan aplikasi web yang robust dan scalable dengan lebih sedikit usaha dan waktu (Devianty et al., 2021).

## 2.10 Xampp



**Gambar 2. 4** Logo Xampp

Xampp adalah sebuah aplikasi *open source* yang dikembangkan oleh Apache Friends untuk pengelolaan server. Karena sifatnya yang *open source*, Xampp dapat digunakan secara gratis dengan mengunduh pada website resminya yaitu <https://www.apachefriends.org/download.html>. Selain itu, keunggulan Xampp terletak pada kemampuannya sebagai *platform* lintas sistem operasi, yang mendukung Windows, macOS, dan Linux. Nama Xampp sendiri mengandung arti bahwa aplikasi ini dirancang untuk menjadi *cross-platform*, sehingga pengguna dapat dengan mudah menginstalnya di berbagai sistem operasi tanpa masalah kompatibilitas (Nadhirah et al., 2024). Xampp terdiri dari komponen-komponen utama seperti Apache sebagai *web server*, MariaDB (yang berasal dari MySQL) sebagai sistem manajemen basis data, serta PHP dan Perl sebagai bahasa pemrograman. Kombinasi ini memberikan solusi lengkap untuk membangun *web server* lokal yang dapat digunakan untuk menguji dan mengembangkan *website* secara efisien. Meskipun berfungsi sebagai aplikasi yang kuat, Xampp dikenal karena kesederhanaannya dalam penggunaan dan kemampuannya untuk dijalankan dengan ringan di berbagai jenis komputer (Sari et al., 2022).

### 2.11 MySQL



Gambar 2. 5 Logo MySQL

MySQL adalah DBMS (*Database Management System*) yang bersifat *open source*, menyediakan dua bentuk lisensi yaitu *Free Software* (perangkat lunak

bebas) dan *Shareware* (perangkat lunak berpemilik dengan penggunaan terbatas). Dalam bentuk *Free Software*, MySQL didistribusikan dengan lisensi GNU *General Public License* (GPL), yang memungkinkan pengguna untuk menggunakan, memodifikasi, dan mendistribusikan perangkat lunak ini secara gratis. Lisensi ini memberikan kebebasan kepada pengguna untuk memanfaatkan MySQL baik untuk keperluan pribadi maupun komersial tanpa dikenakan biaya lisensi, menjadikannya pilihan yang ekonomis dan fleksibel untuk berbagai jenis proyek (Yahfizham et al., 2024).

Di sisi lain, lisensi *Shareware* dari MySQL menawarkan fitur tambahan dan dukungan teknis yang lebih komprehensif bagi pengguna yang membutuhkan layanan tersebut. Dengan menggunakan MySQL, pengguna dapat memanfaatkan kemampuan database server yang handal dan populer tanpa harus mengeluarkan biaya besar. Dukungan komunitas yang luas dan dokumentasi yang lengkap juga membantu pengguna dalam mengatasi berbagai permasalahan teknis, sehingga MySQL menjadi salah satu DBMS yang sangat diminati di berbagai kalangan, baik di lingkungan akademis, bisnis, maupun pengembangan perangkat lunak.

## **2.12 ISO/IEC 25010**

Menurut yang dikutip dari (Andikasari et al., 2024) ISO/IEC 25010 adalah bagian dari kerangka kerja *Systems and Software Quality Requirements and Evaluation (SQuaRE)* yang dikembangkan dalam disiplin rekayasa sistem dan perangkat lunak. Standar ini merupakan kelanjutan dari ISO/IEC 9126 dan secara resmi menggantikan ISO/IEC 9126. ISO/IEC 25010 pertama kali muncul pada tahun 2011 setelah mengalami revisi menyeluruh, menjadikannya sebagai standar internasional terbaru yang sangat valid untuk pengujian sistem informasi yang

sedang dikembangkan. Standar ISO/IEC 25010 menjadi tolak ukur yang diadopsi secara luas oleh perusahaan, instansi, dan organisasi untuk mengevaluasi kualitas perangkat lunak. Dengan menggunakan ISO/IEC 25010, evaluasi kualitas sistem perangkat lunak dapat dilakukan secara mendetail.



**Gambar 2. 6** Gambar ISO 25010

ISO/IEC 25010 mengklasifikasikan karakteristik kualitas produk perangkat lunak ke dalam delapan aspek utama: *functional suitability*, *performance efficiency*, *compatibility*, *usability*, *reliability*, *security*, *maintainability*, dan *portability*.

1. *Functional Suitability*, Karakteristik ini mewakili sejauh mana suatu produk atau sistem menyediakan fungsi yang memenuhi kebutuhan yang dinyatakan dan tersirat ketika digunakan dalam kondisi tertentu. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:
  - a. *Functional completeness* - Sejauh mana rangkaian fungsi mencakup semua tugas dan tujuan pengguna yang ditentukan.
  - b. *Functional correctness* - Sejauh mana suatu produk atau sistem memberikan hasil yang benar dengan tingkat presisi yang diperlukan.
  - c. *Functional appropriateness* - Sejauh mana fungsi-fungsi tersebut memfasilitasi pencapaian tugas dan tujuan tertentu.

2. *Performance efficiency*, Karakteristik ini mewakili kinerja relatif terhadap jumlah sumber daya yang digunakan dalam kondisi tertentu. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:
  - a. *Time behaviour* - Sejauh mana respons dan waktu pemrosesan serta tingkat keluaran suatu produk atau sistem, ketika menjalankan fungsinya, memenuhi persyaratan.
  - b. *Resource utilization* - Sejauh mana jumlah dan jenis sumber daya yang digunakan oleh suatu produk atau sistem, ketika menjalankan fungsinya, memenuhi persyaratan.
  - c. *Capacity* - Sejauh mana batas maksimum parameter produk atau sistem memenuhi persyaratan.
3. *Compatibility*, Sejauh mana suatu produk, sistem, atau komponen dapat bertukar informasi dengan produk, sistem, atau komponen lain, dan/atau menjalankan fungsi yang diperlukan sambil berbagi lingkungan perangkat keras atau perangkat lunak yang sama. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:
  - a. *Co-existence* - Sejauh mana suatu produk dapat menjalankan fungsi-fungsi yang diperlukan secara efisien sambil berbagi lingkungan dan sumber daya yang sama dengan produk lain, tanpa dampak merugikan pada produk lain.
  - b. *Interoperability* - Sejauh mana dua atau lebih sistem, produk atau komponen dapat bertukar informasi dan menggunakan informasi yang telah dipertukarkan.
4. *Usability*, Sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan efektivitas, efisiensi,

dan kepuasan dalam konteks penggunaan tertentu. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:

- a. *Appropriateness recognizability* – Sejauh mana pengguna dapat mengenali apakah suatu produk atau sistem sesuai dengan kebutuhan mereka.
  - b. *Learnability* - Sejauh mana suatu produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan pembelajaran tertentu untuk menggunakan produk atau sistem dengan efektif, efisien, bebas dari risiko dan kepuasan dalam konteks penggunaan tertentu.
  - c. *Operability* - Sejauh mana suatu produk atau sistem memiliki atribut yang membuatnya mudah dioperasikan dan dikendalikan.
  - d. *User error protection* - Sejauh mana suatu sistem melindungi pengguna dari kesalahan.
  - e. *User interface aesthetics* - Sejauh mana antarmuka pengguna memungkinkan interaksi yang menyenangkan dan memuaskan bagi pengguna.
  - f. *Accessibility* - Sejauh mana suatu produk atau sistem dapat digunakan oleh orang-orang dengan karakteristik dan kemampuan terluas untuk mencapai tujuan tertentu dalam konteks penggunaan tertentu.
5. *Reliability*, Sejauh mana suatu sistem, produk, atau komponen menjalankan fungsi tertentu dalam kondisi tertentu untuk jangka waktu tertentu. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:
- a. *Maturity* - Sejauh mana suatu sistem, produk atau komponen memenuhi kebutuhan keandalan dalam operasi normal. Ketersediaan - Sejauh mana

suatu sistem, produk atau komponen dapat beroperasi dan dapat diakses ketika diperlukan untuk digunakan.

- b. *Fault tolerance* - Sejauh mana sistem, produk, atau komponen beroperasi sebagaimana mestinya meskipun terdapat kesalahan perangkat keras atau perangkat lunak.
  - c. *Recoverability* - Sejauh mana, jika terjadi gangguan atau kegagalan, produk atau sistem dapat memulihkan data yang terkena dampak langsung dan memulihkan kondisi sistem yang diinginkan.
6. *Security*, Sejauh mana suatu produk atau sistem melindungi informasi dan data sehingga orang atau produk atau sistem lain memiliki tingkat akses data yang sesuai dengan jenis dan tingkat otorisasinya. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:
- a. *Confidentiality* - Sejauh mana suatu produk atau sistem memastikan bahwa data hanya dapat diakses oleh mereka yang berwenang untuk memiliki akses.
  - b. *Integrity* - Sejauh mana sistem, produk, atau komponen mencegah akses tidak sah, atau modifikasi, program atau data komputer.
  - c. *Non-repudiation* - Sejauh mana suatu tindakan atau peristiwa dapat dibuktikan telah terjadi sehingga peristiwa atau tindakan tersebut tidak dapat disangkal lagi di kemudian hari.
  - d. *Accountability* - Sejauh mana tindakan suatu entitas dapat ditelusuri secara unik ke entitas tersebut.
  - e. *Authenticity* - Sejauh mana identitas subjek atau sumber daya dapat dibuktikan sesuai dengan yang diklaim.

7. *Maintanability*, Karakteristik ini mewakili tingkat efektivitas dan efisiensi dimana suatu produk atau sistem dapat dimodifikasi untuk memperbaikinya, memperbaikinya atau menyesuaikannya dengan perubahan lingkungan, dan persyaratan. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:
  - a. *Modularity* - Sejauh mana suatu sistem atau program komputer terdiri dari komponen-komponen terpisah sedemikian rupa sehingga perubahan pada satu komponen mempunyai dampak minimal pada komponen lainnya.
  - b. *Reusability* - Sejauh mana suatu aset dapat digunakan di lebih dari satu sistem, atau dalam membangun aset lain.
  - c. *Analysability* - Tingkat efektivitas dan efisiensi yang memungkinkan untuk menilai dampak pada produk atau sistem dari perubahan yang direncanakan pada satu atau lebih bagian-bagiannya, atau untuk mendiagnosis kekurangan atau penyebab kegagalan produk, atau untuk mengidentifikasi bagian-bagian yang perlu dianalisa. dimodifikasi.
  - d. *Modifiability* - Sejauh mana suatu produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa menimbulkan cacat atau menurunkan kualitas produk yang ada.
  - e. *Testability* - Tingkat efektivitas dan efisiensi kriteria pengujian yang dapat ditetapkan untuk suatu sistem, produk atau komponen dan pengujian dapat dilakukan untuk menentukan apakah kriteria tersebut telah dipenuhi.
8. *Portability*, Tingkat efektivitas dan efisiensi sistem, produk, atau komponen yang dapat ditransfer dari satu perangkat keras, perangkat lunak, atau lingkungan operasional atau penggunaan lainnya ke lingkungan lain. Ciri-ciri ini terdiri dari sub ciri-ciri sebagai berikut:



- a. *Adaptability* - Sejauh mana suatu produk atau sistem dapat diadaptasi secara efektif dan efisien untuk perangkat keras, perangkat lunak, atau lingkungan operasional atau penggunaan lainnya yang berbeda atau terus berkembang.
- b. *Installability* - Tingkat efektivitas dan efisiensi dimana suatu produk atau sistem dapat berhasil diinstal dan/atau dihapus instalasinya dalam lingkungan tertentu.
- c. *Replaceability* - Sejauh mana suatu produk dapat menggantikan produk perangkat lunak lain yang ditentukan untuk tujuan yang sama di lingkungan yang sama.

Setiap karakteristik ini memberikan panduan yang jelas untuk mengukur dan mengevaluasi berbagai aspek kualitas perangkat lunak yang diperlukan. Dengan demikian, ISO/IEC 25010 tidak hanya memberikan *framework* umum untuk pengujian kualitas, tetapi juga menjadi acuan yang dapat diandalkan dalam menjamin bahwa perangkat lunak yang dikembangkan memenuhi standar yang diharapkan dalam berbagai konteks penggunaan. **Skala Likert**

Skala pengukuran adalah kesepakatan yang berfungsi sebagai acuan untuk menentukan panjang pendeknya interval pada alat ukur. Hal ini memastikan bahwa alat ukur dapat menghasilkan data kuantitatif yang akurat. Dalam penelitian ini, penulis menggunakan skala pengukuran dengan metode skala Likert. Skala Likert digunakan untuk mengukur sikap, pendapat, dan persepsi individu atau kelompok terhadap fenomena sosial. Fenomena sosial ini ditentukan secara spesifik oleh peneliti dan disebut sebagai variabel penelitian. Dengan menggunakan skala Likert, peneliti dapat mengumpulkan data yang menggambarkan sejauh mana responden

setuju atau tidak setuju dengan berbagai pernyataan terkait variabel penelitian. Skala ini membantu dalam mengkonversi sikap subyektif menjadi data numerik yang bisa dianalisis secara statistik. Oleh karena itu, penggunaan skala Likert sangat efektif dalam penelitian yang bertujuan untuk memahami pandangan atau sikap masyarakat terhadap isu-isu sosial yang telah ditetapkan oleh peneliti. Dengan demikian, alat ukur ini memungkinkan pengumpulan data yang konsisten dan valid, sehingga dapat mendukung kesimpulan yang dapat diandalkan dalam penelitian sosial (Sugiyono, 2022).