

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Pada penelitian ini penulis menggunakan beberapa tujuan Pustaka untuk mendukung penelitian yang sedang dilakukan. Berikut adalah tinjauan Pustaka yang digunakan penulis yang dapat dilihat di Tabel 2.1.

**Table 2.1** Tinjauan Pustaka

No	Peneliti/Tahun	Judul Penelitian	Perbedaan	Hasil Penelitian
1	Fajar Nyfantoro Tamara Adriani Salim Anon Mirani (2019)	<i>Perkembangan Pengelolaan Arsip Elektronik di Indonesia: Tinjauan Pustaka Sistematis</i>	Sistem yang menggunakan tinjauan pustaka sistematis sedangkan penelitian ini sistem berbasis web.	Penelitian ini menghasilkan sebuah tinjauan sistematis yang telah dilakukan melalui review artikel jurnal online yang membahas mengenai pengelolaan arsip elektronik di Indonesia

**Tabel 2.1** Tinjauan Pustaka (Lanjutan)

No	Peneliti/Tahun	Judul Penelitian	Perbedaan	Hasil Penelitian
2	Wiha Nur Alim dan Yudi ramdhani (2021)	Perancangan Aplikasi Document Management System Himpana Bandung Berbasis Web	Sistem yang menggunakan metode prototype sedangkan menggunakan metode extreme programming.	Aplikasi ini memudahkan penyimpanan data dapat dilakukan dengan scanning data dalam bentuk softcopy sehingga memiliki backup yang dapat meminimalisir terjadinya kehilangan data keika suatu waktu terjadi hal-hal yang tak terduga.

**Tabel 2.1** Tinjauan Pustaka (Lanjutan)

<b>No</b>	<b>Peneliti/Tahun</b>	<b>Judul Penelitian</b>	<b>Perbedaan</b>	<b>Hasil Penelitian</b>
3	Mahzuro Supianti P, Muhammad Dedi Irawan, dan Ahyat Perdana Utama (2022)	Implementasi RAD (Rapid Application Development) dan Uji Black Box pada Administrasi E-Arsip	Sistem menggunakan pengujian black box sedangkan penelitian ini menggunakan pengujian black box dan iso 25010	Metode RAD pada E-Arsip ini, dapat mempermudah sistem kerja yang sebelumnya menggunakan manual menjadi efektif, dengan metode RAD yang terstruktur dalam penyusunannya dan diperkuat oleh pengujian black box texting dimana pada metode pengujian ini tidak terdapat identifikasi kesalahan dalam perancangan sistem tersebut.

**Tabel 2.1** Tinjauan Pustaka (Lanjutan)

4	Reni Kurniah (2023)	Penerapan Sistem Informasi Berbasis Web Untuk Pengelolaan dan Pengarsipan Dokumen	Sistem pengelolaan dan pengarsipan dokumen berbasis web menggunakan framework codeigniter. Sedangkan penelitian ini sistem berbasis web menggunakan framework laravel.	Memudahkan dalam penyimpanan, pengolahan dan pencarian arsip dimana pada sistem sebelumnya proses penyimpanan, pengolahan dan pencarian arsip masih dilakukan secara manual sehingga memerlukan waktu yang lama dalam proses pencarian berkas.
---	---------------------	---	--	--

**Tabel 2.1** Tinjauan Pustaka (Lanjutan)

No	Peneliti/Tahun	Judul Penelitian	Perbedaan	Hasil Penelitian
5	Billy Ardi dan Yono Cahyono (2024)	Rancang Bangun Penyimpanan Data Elektronik Berbasis Web Menggunakan Metode Waterfall Pada Yayasan Raudhatut Tauhid	Metode yang digunakan adalah Metode Waterfall. Sedangkan penelitian ini menggunakan metode extreme programming.	Sistem yang dibangun dapat digunakan untuk melakukan penyimpanan data secara elektronik dan memudahkan guru serta admin tu dalam menyimpan dan mengambil data surat masuk, surat keluar ijazah dan nilai.

## 2.2 Pengertian Arsip

Menurut kamus besar bahasa Indonesia arsip /ar-sip/ dokumen (surat, akta dan sebagainya), lisan (pidato, ceramah, dan sebagainya), atau bergambar foto, film, dan sebagainya) dari waktu yang lampau, disimpan dalam media tulis (kertas), elektronik (pita kaset, pita video, disket computer, dan sebagainya), biasanya dikeluarkan oleh instansi resmi, disimpan dan dipelihara ditempat khusus untuk referensi, untuk pembakuan, pengaturan, dan pengawetan yang diperlukan supaya bahan arsip dapat dikenal dan disusun sebagaimana aslinya tanpa ada yang dirusak dan di ubah.

### 2.3 Pengertian Arsip Elektronik

Arsip elektronik didefinisikan sebagai arsip yang terdapat dalam media penyimpanan elektronik dan dibuat, dikirimkan, disimpan, atau diakses dengan menggunakan alat elektronik. Arsip elektronik dibuat langsung di teknologi komputer atau dalam bentuk yang telah diubah dari format aslinya menjadi format digital (Erlianti *et al.*, 2022)

### 2.4 Aplikasi Web

*Website* adalah sekumpulan halaman web dengan topik saling terkait, terkadang disertai dengan gambar, video, atau jenis file lainnya. Itu dapat diakses melalui jaringan seperti internet atau jaringan area lokal (*LAN*). Halaman web adalah file yang ditulis sebagai file teks biasa dan dikomplikasi serta digabungkan dengan cara ini dengan instruksi berbasis *HTML* atau *XHTML*. (Larno, Razi and Anggraini, 2019)

Aplikasi berbasis Web adalah aplikasi yang dikembangkan dengan menggunakan bahasa pemrograman *HTML*, *PHP*, *CSS*, *JS* serta memerlukan web server dan browser untuk menjalankannya seperti *Chrome*, *Firefox* atau *Opera*, *internet explorer*, *Microsoft edge* dan lain-lainnya. Aplikasi ini dapat berjalan pada jaringan maupun internet (Jaringan *LAN*). Data terpusat dan akses mudah adalah fitur utama yang membuat aplikasi web populer dan mudah diimplementasikan di berbagai bidang kehidupan. (Rahma Novria *et al.*, 2022)

### 2.5 Extreme Programming

*Extreme Programming* adalah metodologi dari penelitian dari pendekatan *Agile Software Development*. Metode ini berfokus pada pengkodean sebagai aktivitas utama pada semua tahap pengembangan sistem untuk memenuhi kebutuhan pelanggan dan membangun *software* yang lebih baik (Fattah Nadin and Ananda Raharja, 2022)

Alasan penggunaan metode *Extreme Programming* yang digunakan memiliki empat tahapan dimulai *Extreme Programming* yang terdiri dari 4 tahapan yaitu : perencanaan, perancangan, pengkodean, dan pengujian. (Satria *et al.*, 2021)

Berikut ini merupakan tahapan-tahapan dalam metode *Extreme Programming*:(Fattah Nadin and Ananda Raharja, 2022)

### **2.5.1 *Planning* (Perencanaan)**

Tahapan ini merupakan langkah awal dalam pembangunan sistem dimana dalam tahapan ini dilakukan beberapa kegiatan perencanaan yaitu, identifikasi permasalahan, menganalisa kebutuhan sampai dengan penetapan jadwal pelaksanaan pembangunan sistem.

### **2.5.2 *Design* (Perancangan)**

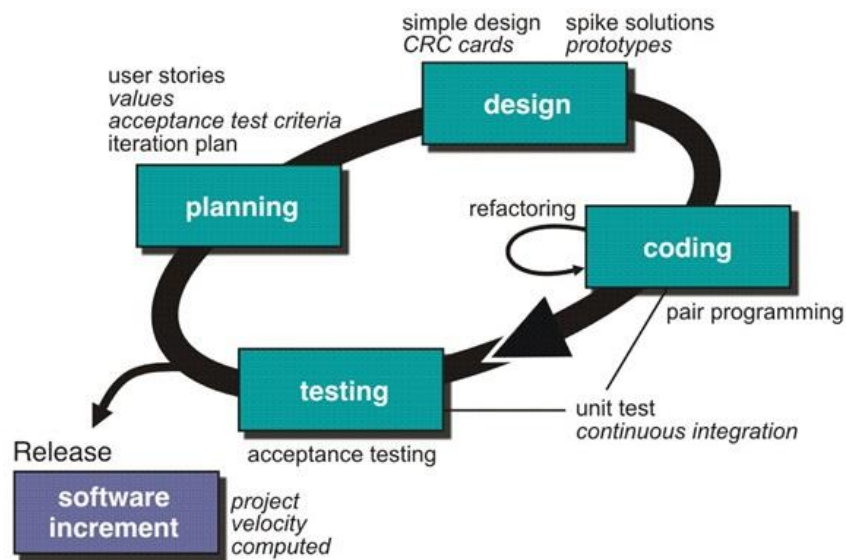
Tahapan berikutnya adalah perancangan dimana pada tahapan ini dilakukan kegiatan pemodelan yang dimulai dari pemodelan sistem, pemodelan arsitektur sampai dengan pemodelan basis data.

### **2.5.3 *Coding* (Pengkodean)**

Tahapan ini merupakan kegiatan penerapan pemodelan yang sudah dibuat kedalam bentuk user interface dengan menggunakan bahasa pemrograman.

### **2.5.4 *Testing* (Pengujian)**

Setelah tahapan pengkodean selesai, kemudian dilakukan tahapan pengujian sistem untuk mengetahui kesalahan apa saja yang timbul saat aplikasi sedang berjalan serta mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan pengguna. dan penekanan pada jawaban atas permasalahan.



**Gambar 2.1** Tahapan Extreme Programming  
 Sumber: (Priskila, 2018)

## 2.6 Unified Modelling Language (UML)

*Unified Modelling language (UML)* adalah penerapan visual yang mendeskripsikan hasil desain, termasuk sintak untuk permodelan visual sistem, dan mewakili sekumpulan gaya permodelan yang mendeskripsikan sistem perangkat lunak dalam bentuk objek (P, Muhammad Dedi Irawan and Ahyat Perdana Utama, 2022). Dikarenakan telah mempunyai standar dalam membangun perangkat lunak yang berbasis objek maka *UML* memiliki banyak diagram (Fadilah, Aris Tantowi and Budiarto, 2023). Dalam penelitian ini melakukan desain hanya 4 diagram yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*

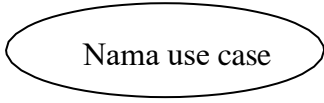
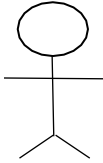

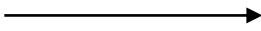
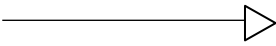
### 2.5.1 Use Case Diagram

*Use Case diagram* merupakan pemodelan kelakuan (*behavior*) pada sebuah sistem informasi yang akan dibuat. Secara umum *use case* diagram digunakan untuk mengetahui apa saja fungsi yang ada didalam sebuah sistem informasi dan siapa saja yang dapat menggunakan fungsi – fungsi dalam sistem tersebut (Fadilah, Aris Tantowi and Budiarto, 2023).



Dengan kata lain, *Use Case Diagram* menggambarkan hubungan antara Aktor dan *Use Case*. Simbol – simbol *Use Case* dapat dilihat pada tabel 2.1.

**Table 2. 2.** Simbol – simbol *use case diagram*







No.	Simbol	Deskripsi
1.	<p><i>Use case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit- unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .
2.	<p>Aktor</p> 	Merupakan simbol yang melambangkan Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	<p>Assosiasi/<i>association</i></p> 	Merupakan simbol yang melambangkan Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.	<p>Exstensi</p> 	Relasi <i>use case</i> tambahan sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.
5.	<p>Generalisasi</p> 	Hubungan generalisasi dan spesialisasi (umum- khusus) antara dua <i>buah use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

### 2.5.2 Activity Diagram

*Activity diagram* menggambarkan alur kerja atau aktivitas yang dilakukan oleh suatu sistem atau proses bisnis. Perlu diingat bahwa diagram 11 aktivitas ini hanyalah menggambarkan kegiatan aktivitas sistem bukan apa

yang dilakukan oleh aktor (Fadilah, Aris Tantowi and Budiarto, 2023) Simbol – simbol *Activity diagram* dapat dilihat pada tabel 2.2.

**Table 2. 3.** Simbol – simbol *Activity Diagram*

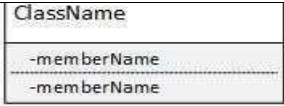






No.	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukansistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlan 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

### 2.5.3 *Class Diagram*

*Class Diagram* menjelaskan hubungan antar kelas dan memberikan deskripsi rinci dari setiap kelas sistem. *Class Diagram* merupakan sebuah 10 gambaran yang menggambarkan struktur sistem dalam hal mendefinisikan kelas – kelas dibuat untuk beroperasi dalam sistem sesuai dengan persyaratan fungsional untuk membangun sistem (Fadilah et al., 2023).

Kelas memiliki apa yang disebut dengan atribut dan metode atau biasanya disebut dengan operasi. Simbol – simbol yang ada pada *Class Diagram* ditunjukkan oleh Tabel 2.3

**Table 2. 4.** Simbol – simbol *Class diagram*

No.	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur system
2.	Antarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.	Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum - khusus)
6.	Kebergantungan/ <i>dependensi</i> 	Relasi antar kelas dengan kebergantungan makna antar kelas
7.	Agregasi/ <i>aggregation</i> 	Relasi antar kelas dengan semua-bagian makna ( <i>whole-part</i> )

## 2.7 *Laravel*

*Laravel* adalah *framework PHP* yang digunakan untuk membangun aplikasi berbasis *website*. *Laravel* banyak digunakan oleh banyak *developer* karena performa, fitur dan skalabilitasnya. *Laravel* dibangun dengan konsep *MVC (Model View Controller)*. Konsep *MVC* ini merupakan cara untuk memisahkan data dari tampilan berdasarkan komponen aplikasi seperti manipulasi data, *controller*, dan *user interface* (Fattah Nadin and Ananda Raharja, 2022) .

## 2.8 *PHP*

*PHP* adalah salah satu bahasa pemrograman yang berjalan dalam sebuah *webserver* dan bertindak sebagai pengolah data pada sebuah server. Bahasa ini memungkinkan pengembang aplikasi web dengan cepat dan mudah menampilkan halaman *HTML* yang dinamis dan interaktif yang dibuat oleh server, selain itu *PHP* juga bersifat *server side script* yang mampu dijalankan di beberapa sistem operasi seperti *linux*, *windows*, dan lainnya. Dalam *database PHP* memiliki kedinamisan yang bisa dihubungkan langsung dengan *MySQL*, *Oracle* (Larno et al., 2019).

## 2.9 *HTML*

*HTML* adalah bahasa pemrograman yang digunakan untuk menulis dan membuat halaman web yang dapat menampilkan berbagai informasi seperti teks dan gambar web *browser* (Augustine and Matondang, 2023) . Saat ini bahasa *HTML* masih terus dikembangkan. Hal ini dikarenakan pengguna internet semakin hari semakin berkembang pesat. Oleh karena itu bahasa *HTML* harus ditingkatkan lagi agar bias menciptakan halaman web yang lebih berkualitas. (andy antonius setiawan, 2019).

## 2.10 *CSS (Cascading Style Sheet)*

*Cascading Style Sheets (CSS)* adalah suatu bahasa pemrograman yang digunakan untuk mendukung pembuatan website agar memiliki tampilan yang lebih menarik dan terstruktur. *CSS* dikembangkan oleh *W3C*. organisasi yang mengembangkan teknologi internet. Tujuannya tak lain untuk mempermudah proses penataan halaman web. (andy antonius setiawan, 2019).

Kesimpulannya adalah CSS merupakan bahasa-bahasa yang merepresentasikan halaman web yang bertujuan untuk menghias dan mengatur gaya tampilan/*layout* halaman web supaya lebih elegan dan menarik.

### 2.11 *Java Script*

*JavaScript* adalah bahasa pemrograman web yang bersifat *Client Side Programming Language*. *Client Side Programming Language* adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh client. Bahasa pemrograman *Client Side* berbeda dengan bahasa pemrograman *Server Side* seperti *PHP*, dimana untuk server side seluruh kode program dijalankan di sisi *server* (andy antonius setiawan, 2019).

### 2.12 *Database*

Basis data (*Database*) merupakan suatu kumpulan data yang disimpan didalam *computer* atau perangkat lunak yang dapat diolah atau dimanipulasi dan kemudian ditampilkan menjadi sebuah informasi (andy antonius setiawan, 2019).

### 2.13 *SQL*

*SQL* dapat diartikan sebagai suatu bahasa yang digunakan untuk mengakses suatu data dalam database relasional dan terstruktur sedangkan *MySQL* adalah *software* atau *tools* untuk mengelola atau memajemen *SQL* dengan menggunakan *Query* atau bahasa khusus. *MySQL* juga merupakan salah satu software yang bersifat open source (andy antonius setiawan, 2019).

Dapat ditarik kesimpulan bahwa *MySQL* merupakan aplikasi yang digunakan untuk mengolah basis data yang banyak digunakan untuk membangun aplikasi yang menggunakan database.

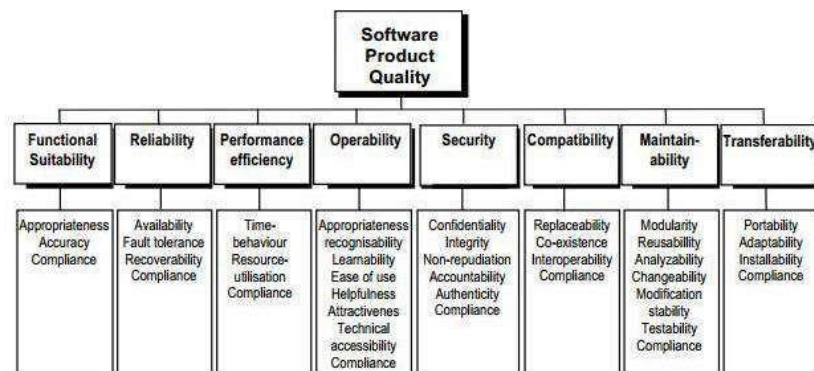
### 2.14 *Black Box Testing*

*Blackbox Testing* adalah metode merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program (Shadiq *et al.*, 2021). Proses *Black Box Testing* dengan cara mencoba

program yang telah dibuat dengan mencoba memasukkan data pada setiap formnya. *Black Box Testing* sendiri merupakan pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak (Zidan *et al.*, 2022).

### 2.15 ISO 25010

*ISO 9126* dan *ISO 25010* adalah dua standar Internasional yang digunakan untuk pengujian perangkat lunak *ISO 25010* dikembangkan sebagai alternatif dari standar *ISO 9126* dengan mempertimbangkan perkembangan teknologi informasi dan komunikasi (*ICT*). Standar *ISO 25010* terdiri dari delapan karakteristik utama, yaitu kompatibilitas fitur, keandalan, efisiensi, ketersediaan, keamanan, *kompatibilitas*, pemeliharaan, dan *probabilitas*. Dua standar ini digunakan sebagai acuan dalam melakukan pengujian terhadap perangkat lunak untuk memastikan kualitas dan kesesuaian fungsionalitasnya. (Lamada, Sa'ban Miru and Amalia, 2020)



**Gambar 2. 2.** Karakteristik *ISO 25010*

Berikut adalah penjelasan dari delapan karakteristik *ISO 25010* yang terlihat pada gambar di atas:

1. *Functionality Suitability*: Pengujian ini menggunakan *test case* dengan *Guttman scale*, yang memberikan jawaban unik untuk pertanyaan yang diujikan. Tujuan dari pengujian ini adalah untuk memastikan kesesuaian fungsionalitas aplikasi.

2. *Performance Efficiency*: Pengujian ini dilakukan untuk menguji kinerja aplikasi yang telah dikembangkan, memastikan bahwa aplikasi berjalan dengan efisien dan responsif.
3. *Compatibility*: Pengujian ini menguji kemampuan suatu komponen atau sistem untuk bertukar informasi dengan baik dengan sistem lainnya.
4. *Usability*: Pengujian ini dilakukan untuk menganalisis umpan balik dari pengguna menggunakan skala dengan 5 pilihan, memastikan bahwa aplikasi mudah digunakan dan sesuai dengan kebutuhan pengguna.
5. *Reliability*: Pengujian ini bertujuan untuk menguji keandalan atau keterpercayaan sistem, memastikan bahwa aplikasi berfungsi secara konsisten dan dapat diandalkan.
6. *Security*: Pengujian ini dilakukan untuk menguji sejauh mana sistem atau produk yang menyediakan layanan melindungi dari akses, penggunaan, modifikasi, gangguan, atau pengungkapan bahaya yang tidak diinginkan.
7. *Maintainability*: Pengujian ini melibatkan alat ukur yang telah diuji oleh peneliti langsung di lapangan kegiatan, dengan tiga aspek utama, yaitu alat ukur, perhitungan konsistensi, dan kesederhanaan dalam pemeliharaan aplikasi.
8. *Portability*: Pengujian ini bertujuan untuk menguji sejauh mana suatu sistem atau produk dapat dipindahkan dari satu ruangan ke ruangan lainnya tanpa mengalami masalah.