

BAB I. PENDAHULUAN

1.1 Latar Belakang

PT. XYZ adalah perusahaan badan hukum publik yang didirikan dengan tujuan memberikan perlindungan sosial kepada pekerja Indonesia. Program-program PT. XYZ dikembangkan dengan menggunakan dana yang berasal dari peserta, baik mereka bekerja secara formal maupun informal. Kehadiran lembaga ini mencerminkan tanggung jawab dan kewajiban negara dalam memberikan perlindungan sosial kepada masyarakat, khususnya pekerja. Sistem jaminan sosial ini beroperasi berdasarkan prinsip gotong royong, artinya dana berasal sepenuhnya dari peserta dan digunakan sepenuhnya untuk meningkatkan kesejahteraan mereka.

Arsip dokumen pada instansi ini akan semakin meningkat setiap harinya, sedangkan kecepatan dan ketepatan dalam pelayanan adalah faktor yang utama. Perubahan data dan penghapusan data saat ini dilakukan dengan pencarian arsip secara manual. Pergarsipan ini juga sangat dibutuhkan dari berbagai divisi di PT. XYZ sehingga sangat rentan terhadap human error karna banyak orang yang terlibat didalamnya. Dari uraian masalah tersebut, penulis mencoba membuat suatu Aplikasi E-Archive (E-Arsip) berbasis website yang terdiri dari *frontend* dan *backend*. Pada laporan akhir ini penulis hanya memfokuskan pada bagian *backend*. Menurut (Salim & Ishaq, 2021) *Backend* adalah bagian belakang layar dari sebuah *website*. *Backend* merupakan bagian yang berfungsi dibalik layar untuk mengelola berbagai operasi, pengolahan data, dan interaksi dengan database.

Aplikasi tersebut dapat membantu menggantikan pengelolaan fisik arsip dokumen tersebut sehingga terbentuk manajemen yang efektif, efisien dan produktif. Begitu juga dengan fisik arsip yang akan terjaga keutuhan informasi dan keamanannya, mempermudah semua orang yang terlibat dan membutuhkan pengelolaan dokumen tersebut.

maka diharapkan dengan adanya penelitian yang berjudul Aplikasi E-arsip berbasis Web pada PT. XYZ dapat mempermudah pengelolaan fisik arsip dokumen sehingga kinerja karyawan PT. XYZ dapat lebih efektif dan efisien. Aplikasi E-Arsip berbasis web pada PT. XYZ dibuat menggunakan metode *Waterfall* dan pada pengaplikasiannya aplikasi ini menggunakan basis data sehingga fisik arsip

dokumen tersebut dapat tersimpan di wadah secara komputerisasi dan dapat dikelola dan dipergunakan sesuai peruntukan dan kebutuhannya.

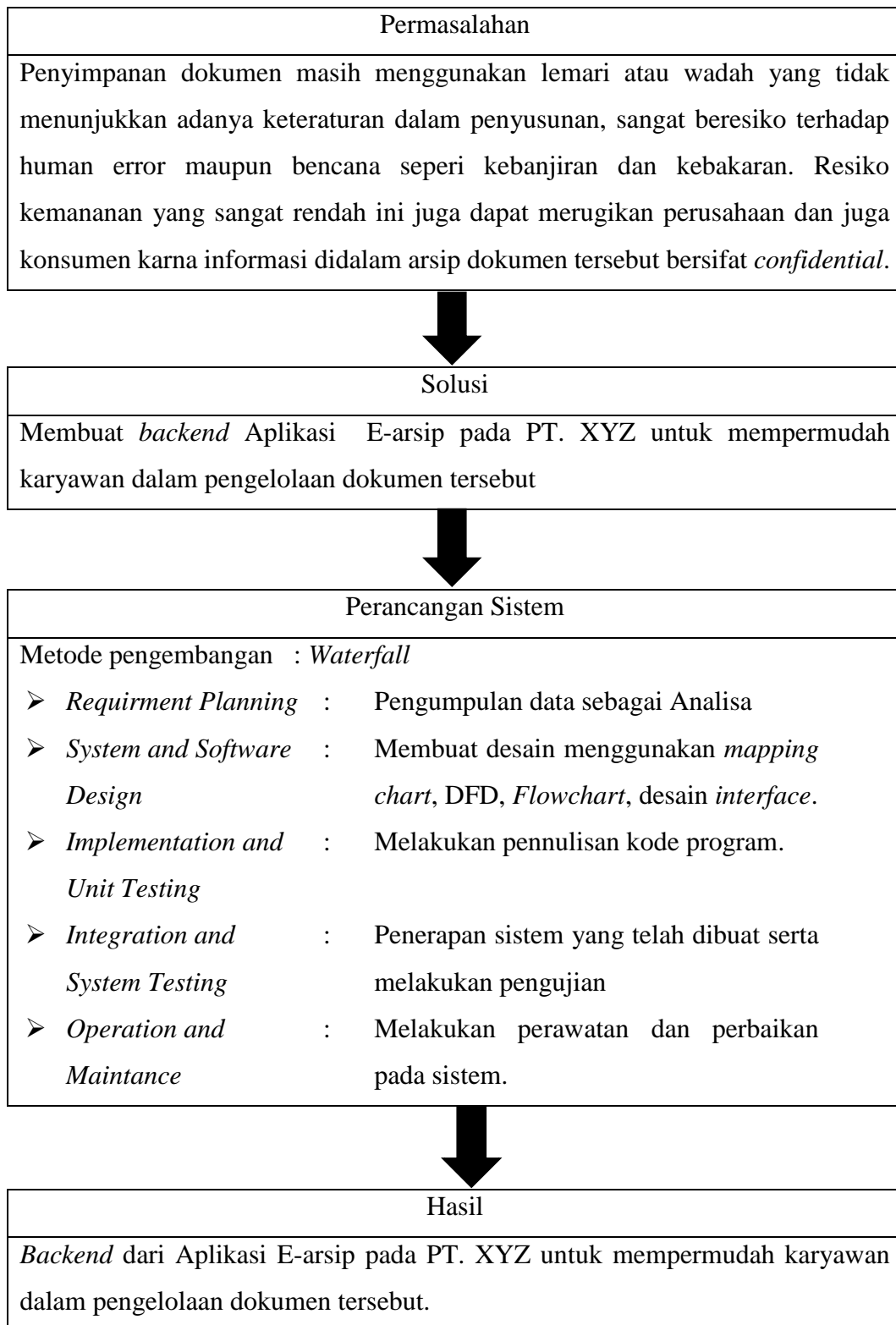
1.2 Tujuan

Tujuan dari penulisan Tugas Akhir ini adalah menghasilkan sebuah *backend* Aplikasi berbasis *website* yang dapat digunakan oleh PT. XYZ untuk mengelola arsip baik informasinya maupun bentuk fisiknya.

1.3 Kerangka Pemikiran

Pengelolaan arsip dokumen pada PT. XYZ yang belum tersimpan dalam basis data secara terkomputerisasi membuat pengelolaan menjadi tidak efektif dan efisien. Penyimpanan dokumen masih menggunakan lemari atau wadah yang tidak menunjukkan adanya keteraturan dalam penyusunan, sangat beresiko terhadap human error maupun bencana seperti banjir dan kebakaran. Resiko keamanan yang sangat rendah ini juga dapat merugikan perusahaan dan juga konsumen karena informasi didalam arsip dokumen tersebut bersifat *confidential*.

Dari permasalahan diatas, penulis akan membuat Aplikasi berbasis website dan menggunakan basis data sebagai media penyimpanan arsip dokumen tersebut. Aplikasi ini selain dapat digunakan oleh divisi pencatatan administrasi juga diharapkan dapat digunakan dari berbagai divisi yang membutuhkan dengan menambahkan fitur permintaan akses *user*. Sehingga, divisi lain yang juga membutuhkan arsip dokumen juga terbantu dalam pencarian data secara efektif dan efisien. Divisi pencatatan administrasi dengan adanya aplikasi ini dapat membantu dalam pencarian, penyimpanan, pembaruan data hingga penghapusan data yang sudah tidak digunakan. Aplikasi ini akan dibuat menggunakan banyak manajemen *user* sehingga masing masing *user* akan mendapat akses yang berbeda-beda. Berdasarkan uraian yang dijelaskan diatas, maka kerangka pikiran disajikan pada Gambar 1.



Gambar 1. Kerangka Pemikiran

1.4 Kontribusi

Aplikasi ini memberikan kontribusi kepada beberapa pihak seperti yang dijelaskan berikut ini. :

- a. Mempermudah karyawan PT. XYZ yang berfokus pada pencatatan arsip untuk mencari data, menyimpan data, mengubah data dan menghapus data sehingga pekerjaan akan lebih cepat dan efisien.
- b. Mempermudah karyawan PT. XYZ di divisi lain dalam pencarian dan penyimpanan yang membutuhkan data dan informasi tersebut
- c. Menghindari kehilangan, kerusakan dan kecatatan dari fisik arsip dokumentasi dari resiko human error dan bencana alam.
- d. Membantu konsumen atau klien dalam menjaga kerahasiaan data informasi dalam arsip tersebut.
- e. Membantu manajemen dalam meningkatkan kecepatan dan ketepatan bekerja.
- f. Memberikan alternatif bagi instansi terkait lainnya dalam memperbaiki penyimpanan fisik arsip dokumentasi yang manual menjadi terkomputerisasi.

BAB II. TINJAUAN PUSTAKA

2.1 Pengembangan *Backend*

Backend adalah sistem yang berjalan dibelakang layar dari sebuah aplikasi atau *website* (Salim & Ishaq, 2021). *Backend* adalah bagian dari sebuah aplikasi atau sistem komputer yang tidak terlihat oleh pengguna. Ini merupakan bagian yang berfungsi untuk mengelola berbagai operasi, pengelolaan data, dan interaksi dengan database. Dalam pembuatan aplikasi penulis menggunakan konsep MVC. MVC adalah singkatan dari model, view, controller, yang merupakan metode untuk mengembangkan aplikasi dengan memisahkan data (*Model*) dari tampilan (*View*) dan cara pengolahannya (*Controller*). (Kosasi dkk, 2021).

- a. *Model*, yang bertindak sebagai pengelola data dalam domain aplikasi, merespons permintaan informasi dan menjalankan instruksi untuk mengubah kondisi tertentu. Contoh dari fungsi-fungsi ini mencakup operasi-operasi terkait basis data, seperti membuat, membaca, memperbarui, dan menghapus data.
- b. *View*, umumnya berupa antarmuka yang berinteraksi langsung dengan pengguna. *View* mengambil informasi dari model dan mengubahnya menjadi format yang sesuai untuk berkomunikasi dengan pengguna.
- c. *Controller*, berperan dalam mengelola input dari pengguna, lalu mengirim instruksi kepada model dan view untuk menjalankan proses sesuai dengan input tersebut. *Controller* merupakan penghubung tindakan pengguna dengan respons dari sistem.

2.2 *Website*

Website adalah sekumpulan halaman web yang ada dalam satu domain dan berisi berbagai informasi. (Asmara, 2019). *Website* merupakan sebuah aplikasi yang berisikan berbagai dokumen multimedia dan diakses melalui *protokol transfer teks hyper* (HTTP) dengan bantuan perangkat lunak yang disebut sebagai *browser*. (Irawan & Simargolang, 2018). Dari uraian diatas penulis simpulkan *website* ialah kumpulan halaman *web* yang terhubung dan dapat diakses melalui internet. *Website*

dapat berisikan teks, gambar, video, dan berbagai jenis konten lainnya yang disajikan dalam format yang terstruktur.

2.3 *Hypertext Preprocessor (PHP)*

Hypertext Preprocessor (PHP) adalah aplikasi perangkat lunak *opensource* yang diatur dalam *General Purpose Licenses* (GPL). PHP adalah bahasa pemrograman yang sangat sesuai untuk pengembangan dalam lingkungan web karena dapat digunakan baik dalam script HTML maupun sebaliknya. PHP juga sudah mendukung pemrograman berbasis objek, yang merupakan model yang banyak digunakan dalam berbagai bahasa pemrograman modern. (Herpendi & Habibah, 2019).

2.4 *MySQL (Structured Query Language)*

Menurut (Julianti dkk, 2019) *MySQL* adalah sebuah *server database* yang memiliki kemampuan untuk mengirim dan menerima data dengan sangat cepat, *multi user* serta menggunakan perintah standar *Structured Query Language* (SQL). *MySQL* adalah database pertama yang mendukung bahasa pemrograman script untuk digunakan di internet. *MySQL* umumnya digunakan dalam pengembangan aplikasi berbasis web, yang biasanya menggunakan bahasa pemrograman PHP.

2.5 *CodeIgniter*

CodeIgniter adalah sebuah kerangka kerja PHP yang mengikuti pola MVC (*Model, View, Controller*) untuk membantu pengembang atau programmer dalam membuat aplikasi berbasis *website* dengan lebih mudah. (Sallaby & Kanedi, 2020). Salah satu kelebihan penggunaan *framework* ini adalah ukurannya yang ringan jika dibandingkan dengan *framework* lain yang memerlukan sumber daya yang besar untuk beroperasi. *CodeIgniter* menyediakan berbagai pustaka (*library*) yang dapat digunakan secara fleksibel sesuai kebutuhan, sehingga sangat membantu para pengembang dalam membangun aplikasi.

2.6 *E-arsip*

Menurut (Irawan & Simargolang, 2018), *Arsip* adalah sesuatu yang sangat penting dalam segala bentuk pekerjaan administratif dan manajemen dalam sebuah organisasi. Ini karena *arsip* adalah tempat penyimpanan semua informasi yang

mencakup segala aktivitas yang kita lakukan dalam kehidupan sehari-hari kita.. Arsip elektronik atau E-arsip adalah sistem atau prosedur untuk mengumpulkan informasi dalam bentuk dokumen yang telah di-scan atau disimpan menggunakan teknologi komputer. Tujuan utamanya adalah untuk memudahkan pencarian dan penggunaan ulang dokumen tersebut. (Putra & Nelisa, 2020)

2.7 *Application Programming Interface (API)*

Application Programming Interface (API) merupakan rancangan yang berguna sebagai penghubung pemrograman aplikasi yang memungkinkan pihak lain untuk menggunakan aplikasi tanpa merubah struktur kode atau basis data sistem yang mendasarinya, dan memfasilitasi komunikasi antar sistem bahkan pada *platform* yang berbeda. *Web service* merupakan API yang akan memberi pengguna akses ke proses pencarian informasi. Melalui arsitektur *Hypertext Transfer Protocol (HTTP)* ini berisi file *Javascript Object Notion (JSON)* yang disediakan untuk pengguna saat mereka mengakses API (Muri dkk., 2019)

2.8 *Javascript Object Notion (JSON)*

Javascript Object Notion (JSON) adalah sebuah format pesan balasan yang memiliki ukuran kecil sehingga mudah dibaca dan dipahami oleh mesin. Ini adalah salah satu jenis respons yang dapat digunakan dalam *REST API*. (Mardzotillah & Ridwan, 2020).

2.9 Perancangan dan Desain Sistem


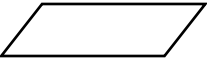
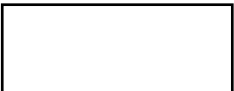
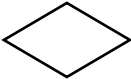


Perancangan adalah proses menggambarkan dengan sebuah teknik yang berbeda, termasuk komponen dan deskripsi arsitektur, serta kendala yang dihadapi selama proses tersebut. Untuk memenuhi kebutuhan pengguna dan memberikan tampilan yang jelas merupakan tujuan perancangan sistem (Aziz, N., Pribadi, G., & Nurcahya, 2020).

Desain sistem adalah seperangkat komponen yang saling berhubungan kemudian diklasifikasikan dan diintegrasikan dengan standarisasi yang jelas Sistem. Desain sistem dapat digunakan oleh pengembang dan berkali-kali disesuaikan dengan kebutuhan organisasi. Ini diperlukan saat merancang sistem. (Kurniawan & Fadlia Adiwijaya, 2021).

2.9.1 Mapping Chart

Mapping chart merupakan diagram alir yang memiliki proses untuk menampilkan langkah dalam bentuk simbol-simbol grafis. *Mapping Chart* memudahkan dalam menganalisis penelitian untuk memecahkan satu masalah. Fungsi dari *mapping chart* adalah untuk menjelaskan bagaimana berbagai elemen terkait satu sama lain. Ini mencakup bagian yang terlibat dalam suatu proses, proses itu sendiri, dan bagaimana dokumen atau data bergerak di antara mereka. (Bagir & Putro, 2018). Simbol *mapping chart* ditampikan pada Tabel 1.

Tabel 1. Simbol-simbol *Mapping Chart*


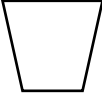

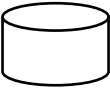


Nama	Simbol	Keterangan
(1)	(2)	(3)
Terminal		menggambarkan awal atau akhir suatu proses
<i>Input/Output</i>		menjelaskan apa yang masuk dan keluar dari proses.
<i>Process</i>		menunjukkan apa yang sedang dilakukan oleh suatu fungsi atau aktivitas.
<i>Decision</i>		mencerminkan kondisi yang dapat mengarah ke cabang hasil ya atau tidak.
<i>Connector</i>		menunjukkan bagaimana aliran berpindah dari satu proses ke proses lain atau menunjukkan arah keputusan yang akan diambil.
<i>Document</i>		Data dapat berbentuk informasi, bisa berupa dokumen tertulis atau berupa salinan digital.

Sumber : (Likhar & Purwanto, 2021)

2.9.2 Flowchart

Flowchart diartikan sebagai penyelesaian pada suatu masalah dengan menggambarkan langkah-langkah menjadi simbol-simbol. Menurut (Syamsiah, 2019), *flowchart* ialah bagian dengan simbol tertentu yang menggambarkan keterkaitan antara suatu instruksi dengan proses lainnya dalam program tersebut. Simbol-simbol *flowchart* disajikan pada Tabel 2.

Tabel 2. Simbol-simbol *Flowchart*

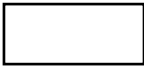
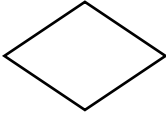


Nama	Simbol	Keterangan
(1)	(2)	(3)
Keyboard		Menyatakan inputan yang melibatkan proses komputerisasi.
Manual Activity		Menunjukkan proses <i>Input</i> atau <i>Output</i> .
Process		Menunjukkan proses yang dilakukan oleh fungsi.
Database		Menunjukkan penyimpanan data yang terkomputerisasi.
Connector		Menyatakan sambungan dari satu proses ke yang lain pada halaman yang sama.
Document		Merupakan bentuk data yang berbentuk dalam bentuk dokumen tertulis ataupun <i>softcopy</i> .

Sumber : (Sari & Sari siregar, 2021)

2.9.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah gambaran grafis dalam bentuk notasi yang digunakan dalam pembuatan basis data untuk menghubungkan entitas atau elemen satu dengan yang lainnya. (Afifah dkk., 2022). ERD adalah gambaran grafis dan konsep logika basis data yang menampilkan informasi rinci tentang semua entitas, hubungan, dan pembatasan yang terkait. (Sari & Sari siregar, 2021). Simbol-simbol ERD disajikan pada Tabel 3.

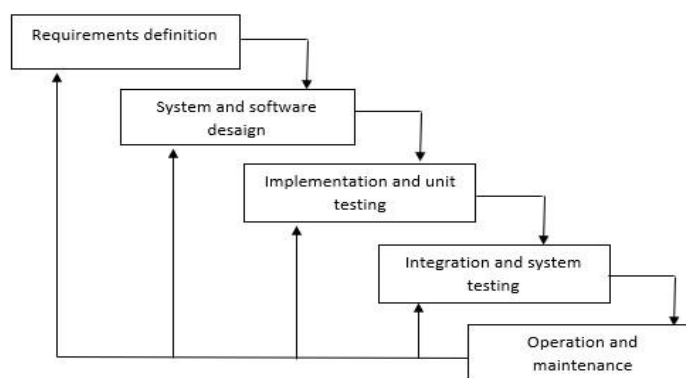
Tabel 3. Simbol-simbol Entity Relationship Diagram

Nama	Simbol	Keterangan
(1)	(2)	(3)
Entitas		Entitas merupakan suatu yang nyata atau abstrak dimana kita akan menyimpan data.
Relasi		Relasi adalah hubungan alami yang terjadi antara satu entitas dengan entitas lainnya.
Atribut		Atribut adalah karakteristik umum atau sebagian besar informasi yang dimiliki oleh suatu entitas.
Garissari		Garis adalah penghubung yang menghubungkan antara relasi dengan entitas atau antara entitas dengan atribut.

Sumber : (Sari & Sari siregar, 2021)

2.10 Metode Pengembangan Waterfall

Metode Waterfall adalah model pengembangan perangkat lunak yang umumnya digunakan. Model pengembangan ini bersifat berurutan dari tahap awal perencanaan hingga tahap akhir pemeliharaan sistem. Setiap tahap berikutnya tidak akan dimulai sebelum tahap sebelumnya diselesaikan, dan tidak ada kemungkinan untuk kembali ke tahap sebelumnya. (Pricillia, 2021). Arsitektur metode pengembangan *waterfall* disajikan pada Gambar 2.



Gambar 2. Metode pengembangan Waterfall

Sumber : (Pricillia, 2021)

Dalam melakukan penerapan pengembangan menggunakan metode Waterfall, memiliki 5 tahapan yaitu :

1. *Requirement Definition*

Dalam tahap ini, data dikumpulkan terlebih dahulu, kemudian dilakukan analisis, dan kebutuhan yang harus dipenuhi oleh program yang akan dikembangkan didefinisikan.

2. *System and Software Design*

Proses tahapan design akan melakukan perancangan design pada perangkat lunak, dari hasil data analisis tersebut. Rancangan yang dibuat berbentuk *Mapping chart*, *Flowchart*, dan data *flow* diagram (DFD).

3. *Implementation and Unit Testing*

Tahap ini dimulai dari membuat program yang sudah dirancang pada tahap kedua, dengan melakukan *coding* program, pengujian pada sistem, dan perbaikan jika terdapat kesalahan.

4. *Integration and System Testing*

Pada tahap ini, dilakukan pengujian sistem untuk memastikan bahwa sistem tersebut valid dan dapat digunakan dengan baik..

5. *Operation and Maintance*

Dan pada tahap terakhir yaitu perawatan pada sistem yang sudah terintegrasi dan perbaikan jika ada kekeliruan pada sistem yang dibangun.

2.9 Pengujian *Black Box* Testing

Pengujian Black Box adalah teknik pengujian perangkat lunak yang berpusat pada spesifikasi fungsional dari perangkat lunak. (Jaya, 2018). *Black Box Testing* Metode ini berfokus pada aspek spesifikasi fungsional perangkat lunak, yang menggambarkan kondisi input dan pengujian yang berhubungan dengan fungsionalitas program. Penggunaan metode ini cukup umum karena cukup sederhana, hanya memerlukan batas bawah dan batas atas dari data yang digunakan.