

BAB I. PENDAHULUAN

1.1 Latar Belakang

Front-End adalah pengembangan antarmuka pengguna *grafis* dari suatu situs *web*. Ini melibatkan penggunaan HTML, CSS, dan *JavaScript* untuk memungkinkan pengguna melihat dan berinteraksi dengan situs *web*. Seorang *Front-End* adalah individu yang bertanggung jawab untuk mengelola tampilan yang dapat dilihat oleh pengguna di *browser* mereka. Mereka harus memastikan bahwa tampilan dan fungsionalitas situs *web* sesuai dengan kesepakatan *desainer*. Secara sederhana, pekerjaan *Front-End* adalah tentang pemrograman yang mengelola dan mengembangkan tampilan dari sebuah aplikasi atau situs *web*.

Tugas *Front-End* adalah memastikan bahwa pengunjung dapat dengan lancar mengakses dan menggunakan aplikasi atau situs *web*. Sebagai contoh, setiap kali Anda mengunjungi sebuah situs *web*, apa pun yang Anda lihat, klik, atau gunakan merupakan hasil dari pekerjaan *Front-End* (Mufti Prasetyo dkk., 2022).

Arsip merupakan dokumen yang berfungsi sebagai memori di dalam sebuah perusahaan. Keberadaan arsip ini bertujuan untuk mengumpulkan informasi. Arsip memegang peran dalam sebuah perusahaan sebagai sumber informasi. Informasi arsip bisa digunakan untuk bukti dalam pengambilan keputusan yang tepat (Nyfantoro dkk., 2020).

PT. XYZ adalah salah satu perusahaan yang bergerak dibidang sosial merupakan cabang dari instansi perusahaan yang banyak melakukan komunikasi dengan masyarakat. Saat ini, proses Arsip penerima Upah masih dilakukan secara manual dan belum adanya pengembangan dalam pengelolaan data Arsip Penerima Upah di PT. XYZ dibagian *Front-end*. Dengan adanya Pengembangan *Front-End* Aplikasi Arsip Penerima Upah Pada PT. XYZ ini lebih memudahkan pengguna dalam proses Arsip Penerima Upah. dengan adanya tampilan yang menarik dengan berbagai fitur didalamnya. Pengembangan *Front-End* Aplikasi Arsip Penerima Upah Pada PT. XYZ membantu pengguna memudahkan proses pengelolaan dokumen Arsip lebih cepat karena fitur yang tersedia mudah dipahami.

Solusi yang dibutuhkan pada permasalahan diatas adalah dengan membuat Pengembangan *Front-End* Aplikasi Arsip Penerima Upah Pada PT. XYZ, sistem

ini terdiri dari *Front-End*, *backend*, dan *database*. Pada penulisan tugas akhir ini penulis hanya memfokuskan pada desain *Front-End* yang digunakan untuk menampilkan tampilan terlihat menarik dan mudah di pahami. *Software* akan dibangun dengan menggunakan *bootstrap*, *javascript*, *cascading style sheet*, *hypertext markup language* dan *JavaScript Object Notation (JSON)*. Sistem ini akan mempermudah pengguna yang ada di PT. XYZ dalam menggunakan Pengembangan *Front-End* Aplikasi Arsip Penerima Upah Pada PT. XYZ.

1.2 Tujuan

Tujuan Tugas Akhir ini untuk Pengembangan *Front-End* Aplikasi Arsip Penerima Upah Pada PT. XYZ. Dengan adanya Pengembangan *Front-End* Aplikasi Arsip Penerima Upah Pada PT. XYZ proses pembuatan *backend* menjadi lebih mudah dengan sudah dibuatnya *Front-end* Arsip Penerima Upah pada PT. XYZ.

1.3 Kerangka Pemikiran

PT. XYZ adalah salah satu perusahaan yang bergerak dibidang sosial dan merupakan cabang dari instansi perusahaan yang banyak melakukan komunikasi dengan masyarakat. Saat ini, proses Arsip penerima Upah masih dilakukan secara manual dan belum adanya pengembangan dalam pengelolaan data Arsip Penerima Upah di PT. XYZ dibagian *Front-end*. Dengan adanya Pengembangan *Front-End* Aplikasi Arsip Penerima Upah Pada PT. XYZ ini lebih memudahkan pengguna dalam proses Arsip Penerima Upah. Dengan adanya tampilan tampilan yang menarik dengan berbagai banyak fitur didalamnya.

Untuk membantu proses pengembangan Arsip Penerima Upah, yaitu dengan membuatkan “Pengembangan *Front-End* Arsip Penerima Upah Pada PT. XYZ”. Dengan menggunakan metode *Waterfall*. Bagan kerangka pemikiran akan disajikan pada Gambar 1.

Permasalahan	
<p>PT. XYZ adalah salah satu perusahaan yang bergerak dibidang sosial dan merupakan cabang dari instansi perusahaan yang banyak melakukan komunikasi dengan masyarakat. Saat ini, proses Arsip penerima Upah masih dilakukan secara manual dan belum adanya pengembangan dalam pengelolaan data Arsip Penerima Upah di PT. XYZ dibagian <i>Front-end</i>. Dengan adanya Pengembangan <i>Front-End</i> Aplikasi Arsip Penerima Upah Pada PT. XYZ ini lebih memudahkan pengguna dalam proses Arsip Penerima Upah. Dengan adanya tampilan tampilan yang menarik dengan berbagai banyak fitur didalamnya.</p>	
Solusi	
<p>Membuat sebuah “Pengembangan <i>Front-End</i> Arsip Penerima Upah Pada PT. XYZ”.</p>	
Perencanaan	
Metode pengembangan	: Metode <i>Waterfall</i>
<i>Requirements Definition</i>	: Pengumpulan data dan teknik wawancara.
<i>System and Software Design</i>	: Membuat desain <i>Mapping chart</i> , <i>Flowchart</i> , dan <i>Data flow diagram</i> (DFD).
<i>Implementation and Unit Testing</i>	: Menuliskan kode program.
<i>Integration and System Testing</i>	: menguji sistem
<i>Operation and Maintance</i>	: Melakukan perawatan atau perbaikan pada sistem.
Hasil	
<p>Pengembangan <i>Front-End</i> Arsip Penerima Upah Pada PT. XYZ</p>	

Gambar 1. Kerangka pemikiran

1.4 Kontribusi

Kontribusi Pengembangan *Front-End* Aplikasi Arsip Penerima Upah Pada PT. XYZ memiliki beberapa kontribusi yaitu :

1. Kontribusi bagi PT. XYZ
 - a. Mempermudah karyawan PT. XYZ yang berfokus pada pencatatan administrasi untuk mencari data, menyimpan data, mengubah data dan menghapus data.
 - b. Mempermudah penanggung jawab dalam monitoring pengelolaan data.
 - c. Menghindari kehilangan, kerusakan dan kecacatan dari fisik arsip dokumen.
 - d. Membantu manajemen dalam meningkatkan kecepatan dan ketepatan bekerja.
2. Kontribusi bagi Politeknik Negeri Lampung
 - a. Memberikan pengetahuan dan pengembangan ilmu
 - b. Menjadikan referensi bagi para mahasiswa yang akan atau sedang melakukan tugas akhir dengan tema sejenis.

BAB II. TINJAUAN PUSTAKA

2.1 *Application Programming Interface (API)*

Antarmuka Pemrograman Aplikasi adalah aplikasi yang dapat diakses oleh pihak lain tanpa perlu melakukan perubahan pada struktur kode inti atau basis data sistem. Hal ini mempermudah komunikasi dengan sistem yang dapat berjalan dengan *platform* yang berbeda. *Web Service*, sebagai bentuk API, berfungsi untuk memberikan akses kepada proses pengambilan data. Ini dilakukan melalui arsitektur *Representational State Transfer* yang beroperasi melalui *Hypertext Transfer Protocol*, dan menyajikan informasi dalam format *Javascript Object Notation (JSON)* kepada pengguna ketika mengakses Antarmuka Pemrograman Aplikasi (API) (Muri dkk., 2019).

2.2 *Front-End*

Front-End adalah bagian dari sebuah *website* yang menampilkan fungsi kepada pengguna dan berhubungan langsung kepada pengguna. *Front-End* mencakup pengalaman pengguna saat berinteraksi dengan *website*. Proses pengembangan tampilan *website* di bagian *Front-End* dikenal sebagai Pengembangan *Front-End*. Dalam praktiknya, seorang pengembang *front-end* juga dapat menggunakan kerangka kerja *framework* atau *library* seperti *react*, *angular*, atau *vue.js* dan *bootstrap* untuk mempermudah pengembangan dan mempercepat proses pembuatan antar muka pengguna yang kompleks. Secara singkat *front-end* adalah bagian dari aplikasi atau *situs web* yang terlihat oleh pengguna dan berurusan dengan antarmuka (Chastro & Darmawan, 2020).

2.3 *Bootstrap*

Bootstrap merupakan *framework* CSS yang didesain untuk mempermudah proses pengembangan bagian depan *front-end* dari sebuah *website*. Ini termasuk dalam kategori *framework front-end* karena mencakup HTML, CSS, dan *JavaScript* (Chastro & Darmawan, 2020). *Bootstrap* adalah sebuah kerangka kerja untuk CSS yang merupakan produk *open source* yang dikembangkan oleh *Mark Otto* dan *Jacob Thorton*. *Bootstrap* diciptakan untuk menghasilkan standar *front-end* yang

konsisten bagi semua pengembang di perusahaan. *Bootstrap* telah berkembang dari proyek yang awalnya berfokus pada CSS menjadi sebuah *platform* yang menyediakan berbagai *plugin JavaScript* dan ikon yang mudah digunakan untuk membuat tombol (Pahlevi dkk., 2018).

2.4 Website

Situs *web* merupakan halaman *web* yang saling terhubung dan disimpan di *server web* dengan jenis dan topik tertentu. *Webpage* adalah halaman khusus dari suatu situs *web* yang disimpan dalam bentuk *file*. Konsep dasar dari situs *web* ini merupakan dasar yang membentuk semua situs *web* (Belinda dkk., 2018). Ada beberapa hal yang menjadikan perusahaan membuat *website* adalah :

1. *Website* adalah wadah wadah yang bisa mempresentasikan produk dan instasi kepada public seperti latar belakang, visi misi perusahaan, serta jenis produk yang ditawarkan.
2. Internet adalah media informasi yang dapat dilakukan tanpa mengetahui batas waktu yang tersedia dimana saja.
3. Memiliki tampilan yang didesain dengan rapi sehingga meningkatkan citra bisnis perusahaan.

2.5 Xampp

XAMPP adalah *server* yang terdiri dari *Apache HTTP Server*. Singkatan XAMPP berasal dari keempat komponen dari sistem operasi utamanya, yaitu *Apache*, *MySQL*, *PHP*, dan *Perl*. XAMPP berfungsi sebagai *database MySQL* yang dapat dioperasikan dengan bantuan bahasa pemrograman *PHP*. Program ini dapat digunakan secara bebas berdasarkan lisensi *GNU*, mudah dioperasikan, dan mampu menyajikan halaman web dinamis (Sarwindah, 2018).

2.6 HyperText Markup Language (HTML)

Bahasa pemrograman yang digunakan untuk membuat *website* adalah Bahasa *HyperText Markup Language (HTML)*. *HTML* digunakan untuk pembuatan struktur dasar halaman *website* atau sebagai dasar awal dalam mengatur kerangka halaman *website* secara lebih terstruktur (Sari dkk., 2022). Menggunakan bahasa

untuk menggambarkan sebuah struktur pada halaman *web*. HTML berperan dalam dokumen *online*. Pernyataan dasar dalam HTML disebut dengan tag, yang ditandai dengan adanya kurung siku (<>). Tag menunjukkan bagian dari dokumen (Pahlevi dkk., 2018).

2.7 *Cascading Style Sheet (CSS)*

Bahasa pemrograman yang dapat digunakan mengatur sebuah tampilan dan desain dari sebuah situs *web*, seperti warna, jenis huruf, garis luar, dan latar belakang. *Cascading Style Sheet (CSS)* digunakan dalam pembuatan situs *web* untuk berkolaborasi dengan *HyperText Markup Language (HTML)* untuk menciptakan tampilan yang menarik. CSS beroperasi dengan prinsip *cascading*, yang berarti aturan-aturan yang didefinisikan dapat berasal dari berbagai sumber. Dengan menggunakan HTML, CSS, dan *JavaScript*, pengembang dapat menciptakan antarmuka pengguna yang menarik dan berfungsi dengan baik di berbagai perangkat dan *platform* (Sari dkk., 2022).

2.8 *JavaScript Object Notation (JSON)*

Format pertukaran data yang memiliki penulisan yang sederhana sehingga menghasilkan *file* dengan ukuran yang kecil. JSON menghasilkan *file* yang berukuran sangat kecil, memungkinkannya diakses dengan cepat. JSON berperan sebagai penghubung dalam komunikasi antara aplikasi klien dan *server*. *Web service* menggunakan format data tertentu untuk mengirimkan sebuah data. Baik dari klien ke *server* maupun sebaliknya, data harus diubah ke dalam sebuah format yang sesuai dengan *web service* yang digunakan (Rima Maulini., 2018). JSON mempunyai 2 struktur sebagai berikut :

1. Pasangan name dan value sebagai *object*, *struct*, *record*, *hashtable*, *dictionary*, *keyed-list*.
2. Daftar nilai biasanya dinyatakan sebagai *vector*, *array*, *list* atau *sequence*.

2.9 JavaScript

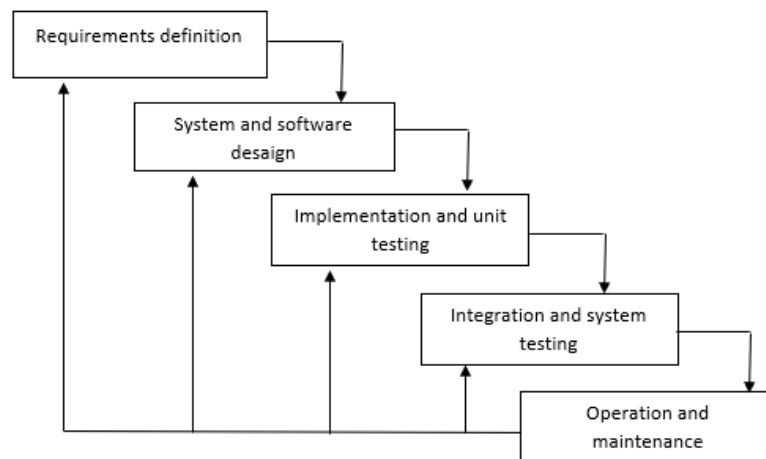
Bahasa pemrograman yang berbentuk sebuah kumpulan *script* yang berjalan pada dokumen *HyperText Markup Language* (HTML). *Javascript* dapat menyempurnakan sebuah tampilan halaman *web-based application* yang dikembangkan. *Javascript* memiliki kemampuan untuk menambahkan interaktivitas ke halaman *web*, seperti *animasi*, efek visual, validasi formulir, manipulasi dan berintraksi dengan server memperbarui konten secara dinamis tanpa perlu memuat ulang halaman. Secara keseluruhan *javascript* adalah alat yang kuat untuk membangun antarmuka pengguna interaktif dan aplikasi *web modern* (Mariko, 2019).

2.10 Command Prompt (CMD)

Command Prompt adalah antarmuka baris perintah yang tersedia di sistem operasi *Windows*. Digunakan untuk eksekusi perintah dan *input*. Dari perintah sederhana hingga tugas melalui skrip dan *file batch*, hingga menjalankan fungsi administratif atau memecahkan masalah umum pada *Windows*. Secara resmi disebut sebagai *Windows Command Processor*, namun sering juga disebut sebagai *shell command* atau *cmd* (Tri Yusnanto, 2019).

2.11 Metode Pengembangan Waterfall

Metode *waterfall* adalah konsep yang menggambarkan pendekatan pembangunan perangkat lunak secara bertahap. Dimulai dari spesifikasi kebutuhan hingga ke pengguna, kemudian tahapan perencanaan, termasuk perencanaan, pemodelan, konstruksi, pengujian sistem, dan akhirnya penyerahan sistem kepada pengguna. Metode ini juga mencakup dukungan penuh terhadap perangkat lunak yang telah selesai dibuat (Kurniawan dkk., 2021).



Gambar 2. *Waterfall* model

Sumber : (R. Kristiana., 2023)

Metode *waterfall* mempunyai lima tahap yaitu :

1. *Requirements Definition*

Pada fase ini, data dikumpulkan untuk selanjutnya dianalisis dan kebutuhan yang harus dipenuhi oleh program yang akan dikembangkan didefinisikan. Dalam proses pengumpulan data, digunakan metode wawancara.

2. *System and Software Design*

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

3. *Implementation and Unit Testing*

Proses desain program ke dalam kode dalam bahasa pemrograman tertentu. Kode program disusun sesuai dengan desain yang telah direncanakan, sehingga menghasilkan aplikasi yang memberikan manfaat bagi perusahaan.

4. *Integration and System Testing*

Melakukan pengujian sistem untuk memastikan bahwa sistem tersebut *valid* dan dapat digunakan dengan benar.

5. *Operation and Maintenance*

Melakukan perawatan pada sistem yang sudah terintegrasi atau perbaikan kalau ada kekeliruan pada sistem yang dibuat.

2.12 Perencanaan dan Desain

Tahap perencanaan adalah tahap mengumpulkan semua data dan informasi yang dibutuhkan untuk Pengembangan *Front-End* Arsip Penerima Upah Pada PT. XYZ. Pengumpulan data ini dilakukan dengan wawancara secara terstruktur.

Sedangkan desain adalah tahap dalam membuat suatu desain sebuah perangkat lunak yaitu sistem yang dapat digunakan di PT. XYZ. Pembuatan desain ini bertujuan untuk menyelesaikan masalah yang dihadapi. Pada tahapan desain ini difokuskan untuk pembuatan desain UML, dan *Use Case Diagram*, *Flowchart* (Adam dkk.,2018).


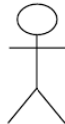


2.12.1 *Unified Modeling Language* (UML)

bahasa visual yang dipakai untuk memodelkan dan berkomunikasi tentang sistem dengan menggunakan diagram dan teks pendukung (Syarif & Nugraha, 2020).

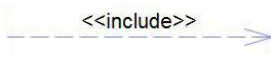

2.12.2 *Use Case Diagram*

Diagram kelas, atau class diagram, memvisualisasikan struktur sistem dengan mengidentifikasi kelas-kelas yang diperlukan untuk membangun sistem tersebut (Syarif & Nugraha, 2020). Simbol *Use Case* dapat dilihat pada Tabel 1.

Tabel 1. Simbol *Use Case* diagram

Simbol	Nama	Keterangan
	<i>Use Case</i>	Menggambarkan intraksi sistem dan aktor.
	<i>Actor</i>	Aktor individu, prosedur, sistem lain yang terlibat dalam <i>use case</i> .
	<i>Asociation</i>	Untuk komunikasi antar aktor dan <i>use case</i> .
	<i>Generalisasi relationship</i>	Asosiasi antar aktor dan kasus penggunaan yang secara pasif berintraksi dengan sistem menggunakan panah terbuka.

Lanjutan Tabel 1.


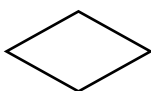


Simbol	Nama	Keterangan
	<i>Include</i>	Keterkaitan antara use case tambahan dan use case aslinya terletak pada ketergantungan use case tambahan pada use case utama untuk dapat beroperasi.
	<i>Extend</i>	yang dapat diakses secara langsung tanpa <i>login</i> .

Sumber : (Julianto & Setiawan, 2019)


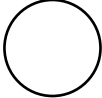


2.12.3 Flowchart

Flowchart adalah jenis diagram yang menggambarkan urutan algoritma atau langkah-langkah yang dijalankan secara berurutan dalam suatu sistem. Seorang analis sistem menggunakan *flowchart* sebagai dokumen pendukung untuk menjelaskan deretan *logis* dari sistem yang sedang dikembangkan. Ini memungkinkan flowchart untuk membantu dalam menemukan solusi terhadap potensi masalah yang mungkin muncul selama pengembangan sistem. *Flowchart* menggunakan sebuah simbol untuk merepresentasikan setiap proses. Untuk menghubungkan satu proses dengan proses berikutnya, digunakan garis penghubung. Dengan adanya *flowchart*, setiap proses dapat dijelaskan secara lebih rinci (Rosaly & Prasetyo, 2019). Simbol *flowchart* dapat dilihat pada Tabel 2.

Tabel 2. Simbol *Flowchart*

Simbol	Nama	Keterangan
	<i>Proses</i>	Proses pengolahan data
	<i>Decision</i>	Menyatakan kondisi yang menentukan 2 kemungkinan seperti iya atau tidak
	<i>Predefined Procces</i>	Sub program
	<i>Input/Output Data</i>	Menghasilkan masukan/keluaran pada data

Lanjutan Tabel 2.

Simbol	Nama	Keterangan
	<i>Terminator</i>	Permulaan/akhir program
	<i>On Page Connector</i>	Penghubung pada bagian <i>flowchart</i>
	<i>Off Page Connector</i>	Perhubungan pada bagian <i>flowchart</i> yang letaknya dihalaman yang berbeda
	<i>Flow Line</i>	Arah aliran program

Sumber : (Fitriyana & Susianto, 2018)

2.13 *BlackBox Testing*

Teknik pengujian yang berfokus pada spesifikasi fungsional perangkat lunak. Dalam blackbox testing, struktur kontrol dianggap tidak relevan, dan perhatian terpusat pada karakteristik domain informasi. Pendekatan ini memastikan bahwa perangkat lunak memiliki kondisi input yang dapat menguji semua persyaratan fungsional dari program tersebut.

Manfaat dari menggunakan *blackbox testing* adalah penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman spesifik. Pengujian dilakukan dari perspektif pengguna, dan ini membantu dalam menetapkan persyaratan bagi programmer dan penguji karena keduanya saling bergantung satu sama lain (Jaya, 2018).