

BAB I. PENDAHULUAN

1.1 Latar Belakang

Perkembangan era ekonomi informasi telah menjadikan teknologi informasi sebagai cara yang kompetitif dan efektif untuk membantu perusahaan bertahan dalam masyarakat modern dan mencapai efek pembangunan yang lebih besar (Ashshidiqy & Ali, 2019). Teknologi memang diperlukan untuk membuat sistem informasi manajemen kepegawaian, namun tidak hanya itu, teknologi yang digunakan dalam sistem tersebut harus dirancang untuk memenuhi kebutuhan perusahaan. Teknologi informasi dapat menyebabkan perubahan dalam praktik dan pemikiran. Perkembangan teknologi informasi di bidang komputer sangat pesat, diawali dengan progress yang efisiensi komputer dengan arsitektur yang sering disebut dengan *client/server*. Pengaturan *client/server* yang menciptakan teknologi informasi lebih mudah digunakan, meningkatkan kapasitas, dan membuat penyimpanan lebih andal (Mahardika, 2021)

PT.XYZ termasuk dalam badan usaha milik negara yang bergerak dibidang transmisi, pembangkitan dan distribusi tenaga listrik. Perusahaan ini memiliki stuktur organisasi, salah satunya terdapat jabatan manajer. Manajer memiliki tanggung jawab mengarahkan usaha pada PT.XYZ dalam mencapai tujuan. Manajer dibantu oleh 3 pejabat pelaksana yaitu : Manajer bagian *Engineering*, Manajer bagian operasi dan pemeliharaan, manajer bagian keuangan dan umum (Sitorus, 2020). Sistem informasi manajemen kepegawaian adalah rangkaian sub sistem informasi yang lengkap dan tersusun secara teratur, yang dapat memperbaiki sebuah informasi sedemikian rupa untuk mengembangkan produktivitas sesuai dengan gaya dan karakter manajer (Dede, 2020). Sistem informasi manajemen kepegawaian ini berguna untuk mendapatkan informasi mengenai kondisi pegawai (profil kepegawaian) yang cepat dan akurat, dapat merencanakan penyebaran (mutasi) pegawai sesuai pendidikan dan kompetensinya, memudahkan pegawai untuk mengajukan cuti dan mutasi.

Terdapat permasalahan mengenai sistem informasi manajemen kepegawaian yang sedang berjalan pada perusahaan saat ini yaitu manajer sulit untuk mengetahui aktivitas setiap pegawai khususnya divisi SDM. Permasalahan

semua data masih berbentuk *hardcopy* dan disimpan pada lemari tanpa membuat *backup* data atau arsip berbentuk *softcopy* sehingga mengakibatkan pengerjaan memakan waktu yang cukup lama dan harus menunggu lebih lama lagi untuk menyediakan laporan yang diperlukan. Data-data tersebut sangat diperlukan untuk diketahui oleh *staff* SDM maupun pegawai yang bersangkutan. Risiko ketika menyimpan secara *hardcopy* sangat besar yaitu kehilangan dokumen, terjadinya bencana alam, keadaan tempat penyimpanan lembab, dari risiko-risiko tersebut diperlukannya arsip atau wadah untuk pengarsipan yang lebih aman, jika pegawai memerlukan data dari tidak perlu untuk menemui atau bertatap muka langsung pada bagian *staff* SDM, dan bagian SDM juga mengetahui informasi pegawai sehingga tidak perlu memeriksa langsung pada karyawan melainkan berinteraksi pada aplikasi yang sudah disediakan.

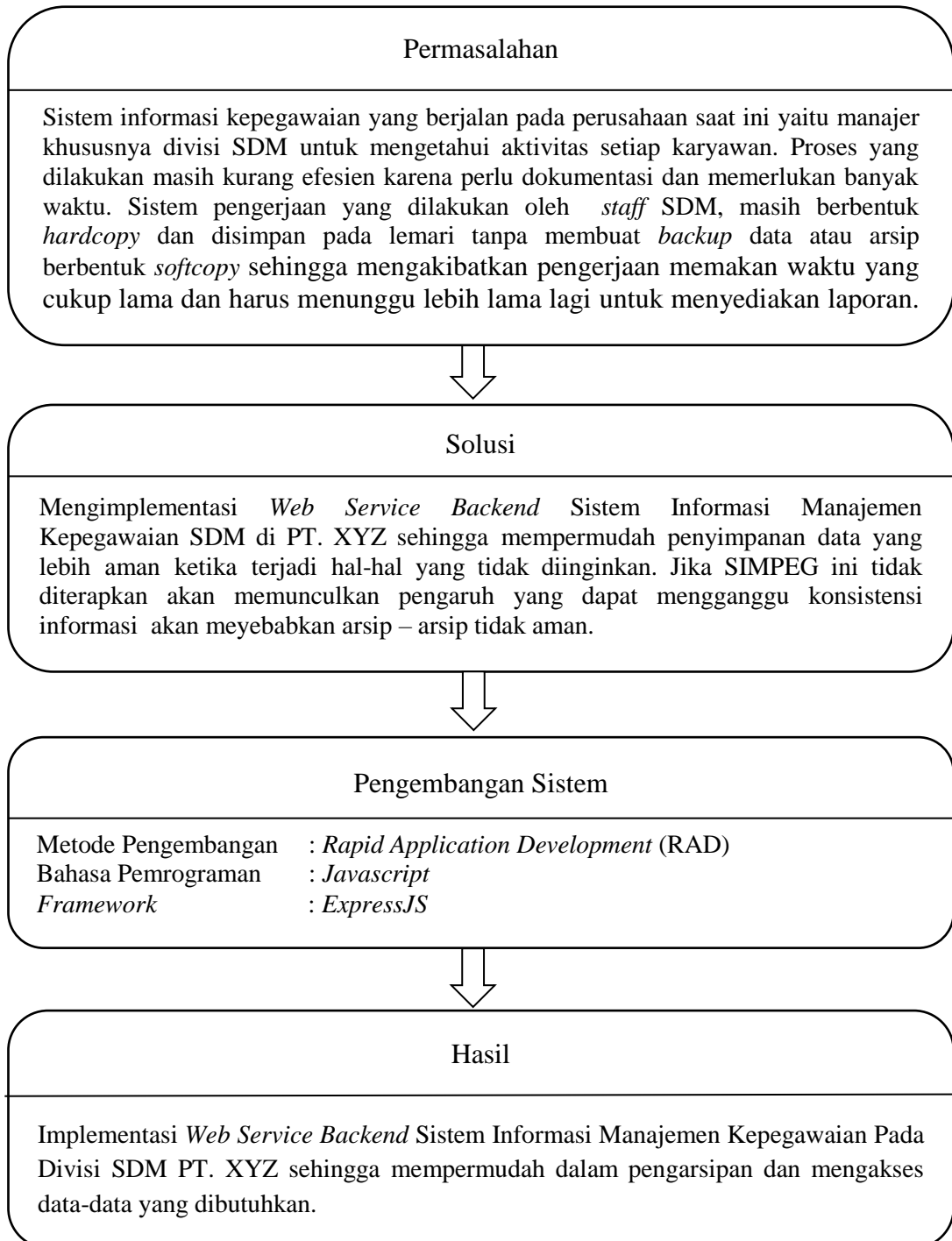
Berdasarkan uraian diatas permasalahan yang dihadapi pegawai di PT. XYZ, dengan menyediakan aplikasi sistem informasi manajemen kepegawaian yang terdiri dari *frontend*, *backend*, dan *database*. Pada penulisan laporan akhir ini penulis hanya memfokuskan pada bagian *backend*. Aplikasi tersebut sangat membantu bagian SDM dalam menginput data dan mengakses data jika pegawai memerlukan data tersebut. Pada menu admin menampilkan history data cuti dan data mutasi terbaru. Ada banyak cara berbeda untuk membuka dokumen atau informasi dengan mudah membangun aplikasi sistem manajemen dapat mempermudah proses pengerjaan pegawai dalam mendata dan mengakses data-data yang dibutuhkan. Data pegawai seperti data pribadi, SK mutasi, dan lain-lain. Alat bantu yang dimaksud adalah *JavaScript*. Sistem ini dapat dibangun dengan mudah terfokus pada pegawai instansi tersebut, tujuan dari pemakaian aplikasi ini untuk mempermudah pegawai pada instansi dalam pencarian dokumen tanpa harus mencari dokumen instansi secara manual. Sistem ini diharapkan dapat bekerja sesuai perencanaan dan berjalan dengan ketentuan yang telah ada berdasarkan wawancara dengan bidang SDM serta hasil dari aplikasi ini dapat membantu PT.XYZ, jika sistem informasi manajemen kepegawaian ini tidak diterapkan akan memunculkan dampak yang dapat mengganggu kestabilan informasi akan menyebabkan arsip – arsip tidak aman.

1.2 Tujuan

Tujuan pada tugas akhir ini ialah implementasi *web service backend* sistem informasi manajemen kepegawaian SDM di PT. XYZ karena proses pengerjaan pendataan pegawai masih belum efisien dan memakan waktu yang banyak. Dengan adanya aplikasi sistem manajemen memudahkan pegawai dalam mengupload dan mengakses data-data penting, meminimalisir kehilangan data seperti terjadi nya bencana alam, jika aplikasi ini diterapkan di PT. XYZ maka proses pengerjaan pegawai lebih mudah dan pengarsipan data pegawai akan aman jika terjadi hal-hal yang tidak diinginkan.

1.3 Kerangka Pemikiran

Dalam proses pengolahan data pegawai pada PT. XYZ saat ini pengerjaan yang dilakukan oleh *staff* SDM, masih kurang efisien karena perlu dokumentasi dan memerlukan banyak waktu, oleh karena itu memperlambat pekerjaan dan harus menemui pegawai yang bertanggung jawab dalam mengurus data-data tersebut untuk pengecekan. Sistem pengerjaan yang dilakukan oleh *staff* SDM, masih berbentuk *hardcopy* dan disimpan pada lemari tanpa membuat *backup* data atau arsip berbentuk *softcopy* sehingga menyebabkan pengerjaan memakan waktu yang cukup lama dan harus menunggu lebih lama lagi hingga laporan siap disajikan. Berdasarkan pemaparan tersebut, maka diperlukan solusi untuk mempermudah dalam sistem manajemen data pegawai, yaitu dengan membuat “Implementasi *Web Service Backend* Sistem Informasi Manajemen Kepegawaian Pada Divisi SDM PT. XYZ”. Tahapan kerangka pemikiran disajikan pada Gambar 1.



Gambar 1. Kerangka Pemikiran

1.4 Kontribusi

Adapun harapan dengan adanya sistem manajemen kepegawai dapat memberikan kontribusi kepada beberapa pihak. Berikut dibawah ini adalah kontribusi yang dapat diberikan :

1. Kontribusi bagi *staff* SDM

Diharapkan dapat bermanfaat bagi perusahaan khususnya pada divisi SDM sehingga mempermudah manajemen kepegawaian di PT. XYZ.

2. Kontribusi bagi karyawan

Diharapkan aplikasi ini dapat bermanfaat bagi karyawan dan dapat berguna dalam mengajukan dokumen kepada divisi SDM.

BAB II. TINJAUAN PUSTAKA

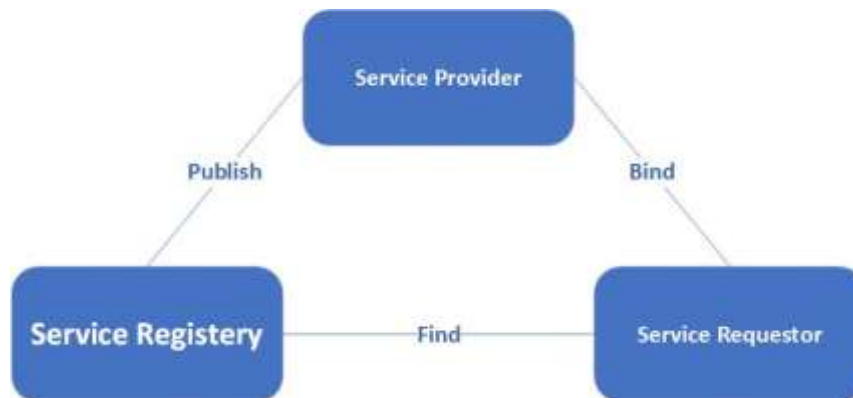
2.1 Implementasi *Web Service*

2.1.1 *Web Service*

Web service merupakan cara sistem perangkat lunak yang dirancang untuk membantu komunikasi dengan mesin di jaringan komputer. Ini berarti semua jenis perangkat lunak dapat dihubungkan dan bertukar informasi yang diperlukan. *Web Service* adalah sebuah fasilitas yang tersedia di *internet*. Selain itu, *web service* juga melingkupi *Web API* yang bisa diakses melalui jaringan, dan dapat dijalankan dengan cara metode *online* sesuai dengan fasilitas yang diminta (Firdaus dkk., 2019).

2.1.2 *Arsitektur Web Service*

Arsitektur *web service* dilihat secara umum menurut (Harjono dkk., 2020) dapat dilihat pada Gambar 2 dibawah ini.



Gambar 2. *Arsitektur Web Service*

Pada gambar diatas, ada tiga entitas utama dari *web service* yaitu:

- *Service provider* : penyedia *web service* aktif menyediakan serangkaian *web service* yang dapat digunakan pengguna.
- *Service requestor* : aplikasi yang bertindak sebagai pengguna membuat permintaan layanan (dalam bentuk layanan *web*) ke *service provider*.
- *Service registry*: tempat di mana *service provider* membuat layanannya. Dalam arsitektur *web service*, *service registry* bersifat opsional.

Pada gambar diatas, *Web service* memiliki tiga operasi yang terlibat di dalamnya :

- *Publish* : Mempublikasikan layanan ke dalam atau dari *registry*.
- *Find* : *Service requestor* memeriksa dan menemukan layanan yang diperlukan.
- *Bind* : *Service requestor* setelah mendapatkan layanan yang mereka cari dan berinteraksi dengan *service provider* untuk berkomunikasi dan mengakses layanan yang di fasilitasi oleh *service provider*.

2.1.3 Komponen Web Service

Menurut (Choirudin & Adil, 2019) Ada tiga komponen standar *web service*, sebagai berikut :

- a. *Simple Object Access Protocol* (SOAP), seperti protokol akan berhubungan dengan pegantian informasi dalam jaringan terdistribusi.
- b. *Web Service Definition Language* (WSDL), mengartikan keberadaan *web service* dari jaringan dan dokumen standar yang ditulis dalam format XML.
- c. *Universal Description, Discovery, and Integration* (UDDI), ditulis dalam XML dan dapat dicapai oleh entitas jaringan, ini juga menunjukkan lokasi direktori *independen platform* yang berisi *service*.

2.1.4 REST

REST adalah salah satu jenis *web service* yang mengimplementasikan peralihan antar *state*. *Web service* bekerja dengan memindahkan tautan yang diberikan, misalnya mengubah *state* halaman *web*. Sementara itu, REST bekerja melalui pengikatan HTTP untuk operasi tertentu, jadi seperti beralih antar *state*. Beberapa fungsi dan metode HTTP secara umum dipakai pada REST, yaitu :

1. *GET*, berguna untuk membaca atau *resource* dari REST *server*.
2. *POST*, berguna untuk membuat sebuah data atau *resource* baru di REST *server*.
3. *PUT*, berguna untuk membaharui atau mengupdate data atau *resource* dari REST *server*.
4. *DELETE*, berguna untuk menghapus data atau *resource* dari REST *server*.

Agar lebih mudah membaca dan menganalisis data yang diterima di sisi *client*, maka *respons* yang dikirim oleh *server* API dalam format JSON (Choirudin & Adil, 2019).

2.1.5 API

Application Programming Interface (API) merupakan rancangan yang berguna sebagai penghubung pemrograman aplikasi yang memungkinkan pihak lain untuk menggunakan aplikasi tanpa mengubah 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 HTTP (*Hypertext Transfer Protocol*) (*Presentation State Transfer*), ini berisi file *Javascript Object Notation* (JSON) yang disediakan untuk pengguna saat mereka mengakses API (Muri dkk., 2019).

2.2 Aplikasi Backend

Aplikasi merupakan sebuah program yang sudah siap dipakai dengan mengikuti instruksi-intruksi pengguna aplikasi dengan hasil yang setara dengan sesuatu yang akan tuju pada aplikasi tersebut. Aplikasi memiliki makna yang dapat memecahkan masalah dan dapat diambil manfaatnya dengan cara yaitu memproses data aplikasi yang berfokus pada perhitung yang menggunakan komputer yang dilakukan sesuai dengan urutan langkah yang diharapkan maupun pada pemrosesan data, aplikasi adalah perangkat komputer yang siap digunakan oleh pengguna (Novendri dkk., 2019).

Backend adalah sistem yang berkerja di belakang layar dari sebuah *website*. Setiap *backend* dari situs *web* meliputi dari tiga macam : *server*, *database*, dan aplikasi. Pengembang *backend* mencatat kode yang mengharuskan ketiga komponen ini berkomunikasi dan bekerja sama untuk menyelesaikan fungsi dan menyampaikan informasi kepada *end user* (Salim & Ishaq, 2020).

2.2.1 Javascript Object Notation (JSON)

JSON (*JavaScript Object Notation*) adalah format *transfer* data yang ringan, mudah untuk dibaca dan ditulis, serta mempermudah untuk diterjemahkan oleh mesin. JSON ialah format dalam bentuk teks yang sepenuhnya mandiri tetapi menggunakan konvensi yang dikenal bagi pemrogram dalam berbagai bahasa C, yaitu C, C, C#, Java, *JavaScript*, *Perl*, dan *Python*. Fitur-fitur ini membentuk JSON sebagai bahasa komunikasi yang bisa dijadikan sebagai acuan (Susetyo dkk., 2018). JSON terbentuk dari dua struktur utama, yaitu:

a. kumpulan pasangan nama/nilai. Dalam beragam bahasa, ini direpresentasikan

sebagai objek, rekaman, struktur, kamus, tabel hash, daftar kunci, tabel asosiatif.

- b. Daftar nilai terurutkan (*an ordered list of values*). Dalam beragam bahasa, ini direpresentasikan sebagai larik, vektor, daftar, atau urutan.

2.2.2 JavaScript

JavaScript adalah bahasa *scripting* yang menganimasikan halaman HTML. *JavaScript* berfungsi di hampir semua *platform*. *JavaScript* identik dengan *browser*, sehingga terintegrasi dengan baik dengan HTML. Saat *browser* memuat halaman, *server* mengirimkan seluruh konten dokumen, termasuk pernyataan HTML dan *JavaScript* (Siahaan & Hasiholan Sianipar, 2018). Program *javascript* digunakan untuk mendeteksi dan merespons tindakan pengguna. *JavaScript* dapat digunakan untuk menentukan tata letak halaman. *JavaScript* juga digunakan untuk memvalidasi formulir pengguna sebelum mengirimkan formulir ke *server*. *JavaScript* bekerja dengan struktur dasar seperti variabel dan tipe data seperti bahasa pemrograman. *JavaScript* juga dapat menangani peristiwa yang diprakarsai pengguna dan pengaturan waktu. Kombinasi HTML, CSS, dan *javascript* membuat situs *web* menarik bagi pengguna (Putawa, 2022).

2.2.3 Supabase

Supabase adalah alternatif *open source* terkenal untuk *backend* Google sebagai layanan (BaaS) *Firebase*. Jangan hanya berpaku pada *database NoSQL*, *supabase* menggunakan *database* relasional *Postgres open-source* dan terpelihara dengan baik. *Supabase* tersedia sebagai solusi yang dihosting sendiri atau dikelola oleh tim *Supabase*. Sebagai sistem aplikasi BaaS, *Supabase* menyediakan beberapa fitur inti dan fitur menarik seperti dukungan data *real-time*, integrasi dengan beberapa layanan autentikasi populer dan layanan *email* untuk membuat dan mengoperasikan *backend* serta menyederhanakan proses (Sobri dkk., 2022).

2.3 Sistem Informasi Manajemen Kepegawaian

Sistem informasi manajemen kepegawaian terdiri dari *software*, perangkat penyimpanan seperti *database* dan perangkat komunikasi, akan saling berhubungan, bergantung dan ditentukan sesuai rancangan transmisi informasi di bidang kepegawaian. Sistem informasi manajemen adalah kombinasi sumber daya manusia dan komputer yang memungkinkan pengumpulan, penyimpanan,

pengambilan, transmisi, dan penerapan informasi untuk sesuatu yang akan tuju manajemen dan perancangan bisnis yang efektif (Sijabat & Purba, 2020).

Keunggulan sistem informasi manajemen kepegawaian (SIMPEG) antara lain:

1. Dapat melacak data karyawan dengan mudah dan cepat.
2. Penyusunan laporan dapat dikerjakan dengan mudah.
3. Memfasilitasi pekerjaan terkait SDM.
4. Anda bisa mendapatkan informasi mengenai kondisi karyawan (profil pribadi) dengan cepat dan akurat.
5. Dapat dengan mudah merencanakan (mutasi) karyawan sesuai dengan pelatihan dan keterampilannya.
6. Perencanaan kebutuhan karyawan (keseimbangan kebutuhan karyawan).

2.4 Object-Oriented Programming (OOP)

Object-Oriented Programming (OOP) merupakan metode pemrograman berorientasi objek. Tujuan OOP ialah memfasilitasi peningkatan program dengan mempelajari pola yang ada dalam kehidupan sehari-hari. Prinsip pemrograman *Object Oriented programming* (OOP) dan pola desain diperkenalkan sebagai cara untuk menangani kompleksitas desain. *Object Oriented programming* (OOP) ialah cara pemrograman yang menggunakan *object* dan *class* (Ningrum, 2022).

2.4.1 Framework

Framework merupakan sebuah kumpulan perintah-perintah yang disatukan dalam *class* dan *function-function* dengan fungsi yang berbeda dengan tujuan agar dapat membantu *developer* dalam memanggil tanpa harus mengetik ulang karna *syntax* program yang digunakan dalam bentuk sama sehingga dapat menghemat waktu (Sallaby & Kanedi, 2020). Adapun beberapa fungsi dari *framework* ketika mengembangkan sebuah aplikasi yang dibuat, antara lain :

1. Membuat kode program lebih sistematis, dengan begitu program yang dibuat akan sesuai dengan fungsi-fungsinya.
2. Meningkatkan keamanan dari *software* yang sudah dibuat. Keamanan tersebut sudah diidentifikasi oleh banyak *developer* ahli. *Framework* juga sering menerima perbaikan yang di dalamnya terdapat komponen keamanannya.

2.4.2 NodeJS

NodeJs merupakan *engine javascript* yang sudah dikembangkan dan dibuat oleh google untuk *browser*, dan menambahkan beberapa pustaka internal lainnya. Sebuah *platform* pengembangan *source* terbuka bertujuan untuk mengimplementasi sisi *server* kode pada *javascript*. *Nodejs* juga berfungsi untuk meningkatkan aplikasi yang memerlukan koneksi berkepanjangan dari *browser* ke *server*. Pada dasarnya, *nodejs* ialah sebuah *runtime environment* dan *script library*. Yang dimaksud dengan *runtime environment* ialah perangkat lunak yang berfokus pada eksekusi, menerapkan dan implementasi. Sedangkan *script* pada *library* adalah sebuah kumpulan, gabungan dari data yang berisi *script* atau kode-kode pemograman (Fajrin, 2017).

2.4.3 Express.JS

Express.JS merupakan *framework* atau kerangka yang terletak dalam *nodejs* yang sangat mudah untuk dikembangkan dan pengembangan pada aplikasi *web, routing, service API, dan security*. *ExpressJs* juga berfungsi pada pemakaian *design pattern* yang dengan mudah disesuaikan dengan arsitektur apapun sehingga mendapatkan hasil yang diinginkan dan fleksibel. Pembuatan desain pada *expressjs* akan lebih mudah dan *resource* yang sangat banyak sehingga dapat meminimalisir biaya yang akan digunakan dalam pengembangan *website* berikutnya (Yuliana & Zahra, 2022).



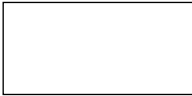




2.5 Perancangan dan Desain Sistem

Desain perangkat lunak ialah metode yang bisa digunakan untuk perancangan desain dan berfokus pada desain pembuatan *software*, adapun struktur data, arsitektur *software*, penggambaran secara *online* oleh pengguna, dan membuat sebuah pengkodean. Pada proses ini keperluan *software* ditransformasikan dari fase analisis kebutuhan menjadi representasi desain sehingga bisa dilakukan implementasi sebagai program pada fase berikutnya. Desain perangkat lunak yang dihasilkan dalam proses ini juga harus didokumentasikan. Sistem yang akan dibuat membutuhkan perancangan peralatan berupa *process planner* dan *information planner* Alat proses meliputi diagram aliran data dan diagram aliran sistem (Bachri dkk., 2022).

2.5.1 Mapping Chart

Mapping chart ialah sebuah bagan alir yang mengartikan mengenai rincian dari tiap langkah proses pada sistem dari awal sampai selesai. Penggambaran tahapan tahapan menggunakan simbol-simbol yang masing-masing memiliki fungsi yang berbeda (Andjani dkk., 2019) disajikan pada Tabel 1.

Tabel 1. Simbol-simbol *Mapping Chart*

Simbol	Nama	Fungsi
	Dokumen	Menjelaskan tentang penginputan pada <i>output</i> atau dokumen yang dicetak.
	<i>Display</i>	Menyatakan <i>output</i> yang dipakai itu <i>layer</i>
	Proses	Menunjukkan pengolahan yang dikerjakan oleh komputer
	Arsip	Berfungsi untuk mengarsipkan data pada program.
	Terminal	Berfungsi untuk awalan dan akhiran program
	Garis Alir	Berfungsi untuk titik sambung pada tempat yang sama
	<i>One page reference</i>	Berfungsi untuk menunjukkan aliran proses.


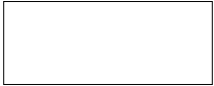

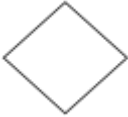
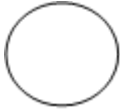

Sumber : (Fitriyana I, Susianto D, 2018)

2.5.2 Flowchart

Flowchart atau diagram alir ialah langkah-langkah dan uraian secara sistematis. *Flowchart* membantu untuk memecahkan suatu masalah dimana perlu melihat masalah lebih dekat, selain itu *flowchart* digunakan sebagai alat dokumentasi dan komunikasi. Menjelaskan metode menyelesaikan masalah

dengan menunjukkan simbol-simbol spesifik yang mudah dipahami, mudah digunakan, dan umum. (Siboro & Tiur Gantini, 2022) disajikan pada tabel 2.

Tabel 2. Simbol-simbol *flowchart*

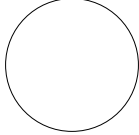

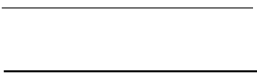


Simbol	Nama	Fungsi
	Terminal	Berfungsi sebagai mengawali dan mengakhir program.
	<i>Processing</i>	Berfungsi untuk menunjukkan sebuah proses yang akan dibuat.
	<i>Input/output</i>	Berfungsi untuk melihat input dan <i>output</i> tanpa melihat jenisnya.
	<i>Decision</i>	Berfungsi untuk menentukan pilihan sebuah proses berdasarkan kondisi tertentu.
	<i>Connector</i>	Berfungsi untuk menyatakan sambungan satu dengan sambungan lainnya ke proses berikutnya pada halaman yang sama.
	<i>Connecting line</i>	Berfungsi untuk menghubungkan suatu simbol dengan simbol lainnya dengan menyatakan suatu arus proses.

Sumber : (Santoso & Nurmalina, 2017)

2.5.3 *Data flow diagram (DFD)*

Data Flow Diagram ialah sebuah diagram yang digunakan untuk mengilustrasikan sistem otomatis atau terkomputerisasi, panduan atau kombinasi keduanya, DFD adalah format aliran data atau ditentukan untuk menentukan dari mana data bersumber dan menunjukkan data pergi dari sistem (Handayani dkk., 2019) disajikan pada tabel 3.

Tabel 3. Simbol-simbol *Data flow diagram*

Simbol	Nama	Fungsi
	Proses	Memberitahu perubahan dari masukan menjadi keluaran pada sistem.
	Entitas eksternal	Berfungsi dimana entitas saling berkomunikasi pada sistem.
	Penyimpanan	Berfungsi untuk menunjukkan sebuah penyimpanan dalam <i>database</i> .
 	Aliran data	Berfungsi untuk menggambarkan aliran data yang masuk dan keluar pada suatu proses.

Sumber : (Budiman dkk., 2021)

2.6 Metode Pengembangan

2.6.1 *Rapid Application Development (RAD)*

Rapid Application Development (RAD) merupakan model proses *software* linier dan berurutan dengan mengutamakan siklus pengembangan yang sangat singkat. RAD adalah pendekatan menuju kepada objek untuk proses pengembangan sistem yang meliputi metode pengembangan dan perangkat lunak. Tujuan dari RAD ialah untuk mempercepat waktu yang pada umumnya dihabiskan dalam siklus hidup pengembangan sistem tradisional antara perencanaan dan implementasi sistem informasi. (Sintawati, 2022). Adapun keuntungan dalam menggunakan metode RAD sebagai berikut:

1. Cocok untuk mengembangkan sistem informasi jangka pendek.
2. Metode RAD lebih baik dan efisien dibandingkan dengan metode *waterfall/sequential linear* dalam menghasilkan sistem yang dibutuhkan.
3. Metode ini memungkinkan Anda untuk menggunakan komponen yang sudah

ada, jadi tidak perlu membuatnya kembali dari awal, sehingga tidak memakan banyak waktu.

Adapun kekurangan menggunakan metode RAD adalah sebagai berikut :

1. Membutuhkan banyak pekerjaan jika proyek pengembangan sistem informasinya besar.
2. Karena mempersingkat waktu pembuatan sistem, maka harus ada kewajiban antara pengembang dan *clien*, jika tidak ada kewajiban maka proyek akan berjalan dengan cara ini.
3. Tidak semua pengembangan sistem dapat menggunakan metode RAD.
4. Apabila terjadi perubahan di tengah pengembangan sistem, harus dibuat kontrak baru.



Gambar 3. RAD

Sumber : (Putri dkk., 2018)

Berikut tahapan dari RAD menurut dibagi menjadi empat, yaitu :

1. *Requirement Planning* (Merencanakan Kebutuhan)

Pada tahapan ini, pengguna menentukan fungsi apa yang harus dikarakteristikan oleh aplikasi tersebut sehingga terdapat informasi beserta proses data bisnis yang terkait.

2. *Design Workshop* (Membuat Desain)

Tahapan ini ialah solusi alternatif yang mana pengguna akan berfokus pada sudut pandang desain dari sistem melalui analisis. Selanjutnya, membuat desain dari alur bisnis dan pemograman bertujuan agar data-data yang telah diperoleh dan diimplementasikan dalam arsitektur sistem informasi.

3. *Implementation*

Tahapan ini berfokus pada desain yang sudah dibuat dalam tahapan sebelumnya. Berikutnya mengembangkan pengkodean sistem.

4. *Cutover*

Pada tahapan terakhir, dimana aplikasi akan diuji coba dan pengenalan terhadap aplikasi. Untuk melihat apakah aplikasi yang dibangun sudah sesuai dengan sistem yang diperlukan. Jika aplikasi terdapat kesalahan maka akan direvisi dari pengguna.

2.7 **Dokumentasi Web Service**

2.7.1 *Postman*

Postman adalah *platform* yang memudahkan pengujian API. Salah satu fitur Postman adalah dapat melakukan pengujian API dengan baik dan cepat. Cara kerja Postman dengan mengklasifikasi *request* berdasarkan metode *request* dan URL (Dody Firmansyah, 2022).

2.8 **Pengujian**

Pengujian adalah sebuah alat untuk mengukur kualitas atau pengetahuan perangkat lunak yang dibutuhkan sebelum perangkat lunak digunakan. Hal tersebut diupayakan untuk meyakinkan pengguna bahwa perangkat lunak dapat terlaksana dalam kinerja yang sesuai dengan keperluan pengguna. Pada tugas akhir ini penulis menggunakan dua jenis pengujian yaitu *system usability testing*.

2.8.1 *System Usability Scale (SUS)*

System Usability Scale (SUS) adalah kuesioner yang dapat digunakan untuk mengukur kegunaan sistem dari perspektif holistik pengguna komputer. SUS dikembangkan oleh *John Brooke* pada tahun 1986 (Setiawan & Rafianto, 2020). Hingga saat ini SUS banyak digunakan untuk mengukur *usability* dan memiliki beberapa keunggulan, diantaranya:

1. SUS mudah digunakan, dengan skor mulai dari 0 hingga 100 poin.
2. SUS sangat mudah digunakan, tidak memerlukan perhitungan yang rumit.
3. SUS gratis, tidak perlu biaya tambahan.
4. SUS ditemukan valid dan dapat diandalkan dengan ukuran yang kecil.

Usability adalah metode analisis kualitatif yang dapat digunakan untuk menentukan seberapa mudah pengguna menggunakan perangkat lunak. Aspek *usability* merupakan aspek yang ditujukan untuk keberhasilan perangkat lunak dan

tanda diterimanya perangkat lunak di kalangan pengguna. *Usability* dalam proses pengujian dapat mempengaruhi pengguna akhir secara langsung atau tidak langsung, dengan kata lain dapat disesuaikan dengan keadaan objek dan pilihan perangkat lunak penguji. dapat dilakukan untuk perangkat lunak kecil atau besar dan untuk berbagai *platform* seperti *desktop*, *web*, dan seluler. Fokus pengujian kegunaan adalah keramahan pengguna dari perangkat lunak (Ependi, 2019).

2.9 Jurnal Terkait

Jurnal terkait ialah teori dari beberapa sumber riset yang dijadikan pedoman penyusunan tugas akhir ini. Keterkaitan yang dimaksud kesamaan dalam metode pengembangan, maupun studi kasus yang diangkat oleh penulis. Adapun referensi karya ilmiah yang di dapatkan oleh penulis sebagai berikut :

1. Rulloh dkk., (2017) dengan judul “Implementasi REST API pada Aplikasi Panduan Kepaskibraan Berbasis *Android*” karya tulis ini dibuat untuk meningkatkan perkembangannya khususnya pada smartphone dengan sistem operasi *android*, dari aplikasi yang berdiri sendiri, kini telah menjadi aplikasi yang dapat terhubung ke aplikasi lain melalui jaringan global. Salah satu jenis *web service* adalah REST atau RESTful (*Representational State Transfer*). REST sendiri memungkinkan permintaan sistem untuk mengakses dan memodifikasi teks layanan *web*. *Web service* API yang menggunakan REST disebut dengan RESTful API Oleh karena itu, peneliti tertarik untuk meneliti implementasi REST API dengan *prototype* aplikasi Panduan kepaskibrakaan. Metode pengembangan yang digunakan pada tugas akhir ini yaitu *Rapid Application Development*
2. Laipaka, (2019) dengan judul “Penerapan *Web Service* JSON pada *Backend-Developer Summary Report Executive* Menggunakan Arsitektur MVC *CodeIgniter*” karya ilmiah ini dibuat untuk membangun aplikasi khusus untuk *report executive*. Akses data menggunakan JSON *web service*, pengembangan sistem ini adalah model *waterfall* yang terdiri dari *Communication, Design, Modeling, Construction, Development*. Studi menciptakan sistem informasi yang membantu manajemen puncak mendapatkan informasi verifikasi secara *real-time*.
3. Ridha dkk., (2021) dengan judul “Pengembangan *Web Service* Sistem Pembayaran Multibank Universitas Negeri Jakarta” karya ilmiah ini dibuat

mengembangkan dan menghasilkan layanan *online* yang dapat digunakan untuk mengimplementasikan multi-banking di Universitas Negeri Jakarta dan dapat beradaptasi dengan perubahan masa depan. Multibank *web service* dikembangkan dengan konsep *service-oriented architecture*, *spiral development model*, *Node.js scope*, *Express framework* dan *database MySQL*. Metode pengembangan yang digunakan pada tugas akhir ini yaitu *waterfall*.

4. Susilo (2018) dengan judul “Pembangunan Sistem Informasi Perekrutan Pegawai Berbasis *Web* Dengan *Spring Framework*” karya ilmiah ini dibuat untuk merekrut karyawan baru untuk memperkuat perusahaan dalam hal sumber daya manusia dan mempromosikan *startup XYZ*, karena rekrutmen pegawai pada awalnya masih menggunakan aplikasi lama, dimana fungsinya masih sangat terbatas, aplikasi hanya bisa mengirim CV dari luar sistem, dan dari sistem internal hanya CV yang dikirim yang bisa di *review* kemudian HRD (PSDM), Form Penerimaan Karyawan (FPK) dan proses rekrutmen masih manual. Ini dapat menyebabkan kesalahan dan pemrosesan yang lambat. Untuk mengatasi masalah tersebut maka dibangunlah sebuah sistem informasi rekrutmen karyawan, dimana rekrutmen tidak bergantung pada keterbatasan manusia, serta pengelolaannya lebih cepat dan efisien dibandingkan dengan cara manual. *Database* yang digunakan adalah *PostgreSQL*.

5. Sijabat dkk., (2020) dengan judul “Pengaruh Penerapan Sistem Informasi Manajemen Kepegawaian terhadap Pengembangan Karir di Satuan Polisi Pamong Praja Kota” karya ilmiah ini dibuat untuk kegiatan yang berupa pengelolaan data pegawai, pengumpulan data pegawai serta pelaporan data kepegawaian. Penelitian ini bertujuan dalam penerapan sistem informasi manajemen kepegawaian diharapkan dapat memberikan informasi yang cepat dan akurat bagi pegawai dalam mengembangkan karirnya. Metode pengembangan yang digunakan dalam tugas akhir yaitu SDLC.

BAB III. METODE PELAKSANAAN

3.1 Tempat dan Waktu

Tugas akhir dengan judul “Implementasi *Web Service Backend* Sistem Informasi Manajemen Kepegawaian Pada Divisi SDM PT. XYZ” ini dilaksanakan di Politeknik Negeri Lampung, Jl. Soekarno Hatta Rajabasa Bandar Lampung.

3.2 Alat dan Bahan

3.2.1 Alat

Alat yang digunakan dikategorikan menjadi 2 perangkat yaitu perangkat keras dan Perangkat lunak. Berikut ini merupakan alat dan bahan yang digunakan dalam pembuatan tugas akhir tersaji pada Tabel 4.

Tabel 4. Alat

Perangkat Keras	Perangkat Lunak
Laptop	<ol style="list-style-type: none">1. Sistem Operasi <i>Windows</i>2. <i>Supabase</i>3. <i>Microsoft Visio</i>4. <i>Microsoft Word</i>5. <i>Web Browser</i>6. <i>Visual Studio Code</i>7. <i>Node.js</i>8. <i>Swagger</i>9. <i>ExpressJS</i>10. Aplikasi <i>Balsamiq Wireframes</i>

3.2.2 Bahan

Dalam pembuatan tugas akhir ini membutuhkan bahan untuk menyelesaikan tugas akhir ini. Materi berupa data yang dikumpulkan untuk pembuatan aplikasi ini merupakan hasil observasi sistem manajemen kepegawaian dan wawancara dengan karyawan dan manajer lapangan PT.XYZ.

3.2 Metode Pengumpulan Data

Pengumpulan data yang dilakukan dalam implementasi *web service backend* sistem informasi manajemen kepegawaian pada divisi SDM PT. XYZ yaitu dengan cara wawancara kebagian SDM serta mencari referensi mengenai proyek yang akan dibuat dengan dasar pemecahan masalah atau tujuan dari Tugas Akhir ini.

3.3 Metode Pengembangan Sistem

Pengembangan sistem yang dilakukan dalam implementasi *web service backend* sistem informasi manajemen kepegawaian pada divisi SDM PT. XYZ yaitu metode *Rapid Application Development* (RAD). Tahapan yang dilakukan pada metode RAD dalam pengembangan sistem yaitu dengan, *requirement planning, design workshop, implementation* dan *cutover*.

Adapun penjelasan mengenai pengembangan sistem yang dilakukan menggunakan metode RAD adalah sebagai berikut :

3.3.2 Requirement Planning

Pada tahapan ini dilakukan proses perancangan yang menghasilkan tujuan dari sistem yang dibuat yaitu implementasi *web service backend* sistem informasi manajemen kepegawaian pada divisi SDM PT. XYZ. Tahapan pertama yang dilakukan dengan wawancara pada bidang SDM yang dilampirkan pada lampiran 1. Dari hasil perencanaan tersebut, data yang dihasilkan kemudian di analisa untuk mencapai tujuan yang nantinya akan bermanfaat bagi instansi tersebut.

3.3.3 Design Workshop

Fase *design* adalah tahapan kedua, tahapan ini hanya befokus pada perancangan. Perancangan tersebut berdasarkan hasil dari analisa yang telah dilakukan yaitu :

1. Rancangan *Flowchart*
2. Rancangan *Mapping chart*
3. Rancangan DFD (*Data Flow Diagram*)

3.3.4 Implementation

Pada langkah ini, desain sistem diimplementasikan sebagai kode program. Pengkodean dilakukan sesuai dengan yang direncanakan sebelumnya yang menggunakan bahasa pemograman *javascript* dan *framework expressjs*. Tujuan

akhir dari tahap ini adalah untuk menghasilkan implementasi *web service backend* sistem informasi manajemen kepegawaian pada divisi SDM PT. XYZ.

3.3.5 Cutover

Cutover ialah tahap pengujian setelah aplikasi selesai dibuat. *System Usability Testing* digunakan dalam pengujian aplikasi yang tujuannya adalah untuk mengevaluasi aspek-aspek aplikasi, apakah program yang dibuat memenuhi kebutuhan sistem yang dibangun karena metode ini lebih menekankan pada sudut pandang penggunaan akhir sehingga hasil pengujian akan lebih sesuai dengan apa yang dihadapi oleh pengguna.