

BAB I. PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi informasi dewasa ini semakin pesat dan masif, kemajuan teknologi informasi ini memicu berbagai bidang untuk menerapkan teknologi informasi di setiap aktivitasnya. Hampir semua instansi baik pendidikan, pemerintahan, maupun swasta berlomba-lomba menerapkan teknologi informasi untuk memfasilitasi proses pelayanan sehingga mampu memberikan pelayanan yang terbaik bagi pelanggannya atau mempermudah pengelolaan internal instansinya. Salah satu bentuk penerapan teknologi informasi yang umum dijumpai di setiap instansi adalah sistem informasi manajemen pegawai. Sistem informasi manajemen merupakan sebuah sistem yang menyediakan fungsi manajemen seperti perencanaan, pengendalian, dan operasional pada sebuah instansi dengan cara yang efektif sehingga menghasilkan informasi yang dibutuhkan secara cepat, tepat dan akurat untuk membantu proses pengambilan keputusan dengan mudah (Sadikin & Wiranda, 2022). Penerapan sistem informasi manajemen kepegawaian dapat meningkatkan efektivitas pengelolaan sumber daya manusia. Sistem informasi manajemen kepegawaian tersebut dapat mempercepat dan mempermudah pengelolaan data pegawai, serta memudahkan dalam pengambilan keputusan terkait pengelolaan sumber daya manusia (A. Wibisono, 2020).

Pengelolaan arsip merupakan salah satu bagian dari sistem manajemen informasi yang penting dilakukan di sebuah instansi. Arsip yang dikelola dengan baik dan teratur mampu meningkatkan kinerja instansi dalam kegiatan administrasi dan pengambilan keputusan. Apabila arsip tidak dikelola dengan baik maka akan menghambat instansi dalam mengambil keputusan dan memperlambat proses administrasi (Hendriyani, 2021). Arsip yang dikelola di sebuah instansi meliputi semua arsip yang berkaitan dengan instansi tersebut, salah satunya yaitu arsip kepegawaian. Untuk itu, optimalisasi pengelolaan arsip kepegawaian menjadi penting dilakukan pada PT XYZ untuk mempermudah proses administrasi baik perubahan data seperti mutasi dan cuti, maupun pengarsipan data kepegawaian lainnya yang sebelumnya dilakukan secara manual menjadi otomatis menggunakan sebuah sistem informasi manajemen.

PT XYZ merupakan Badan Usaha Milik Negara (BUMN) yang bertanggung jawab membangkitkan dan menyediakan listrik di Sumatera bagian tengah dan selatan. Instansi tersebut merupakan pusat administrasi semua pembangkitan yang ada di provinsi Lampung. Saat ini pengelolaan arsip pada PT XYZ masih dilakukan dengan cara manual mulai dari pegawai diterima sampai dengan pegawai pensiun. Pemberkasan arsip pegawai masih disimpan di dalam ordner map dan dikelompokkan berdasarkan tahun masuk pegawai. Proses perubahan data, mutasi, cuti dan pencarian berkas pegawai yang memakan waktu lama menyebabkan menurunnya kinerja proses administrasi pegawai di PT XYZ.

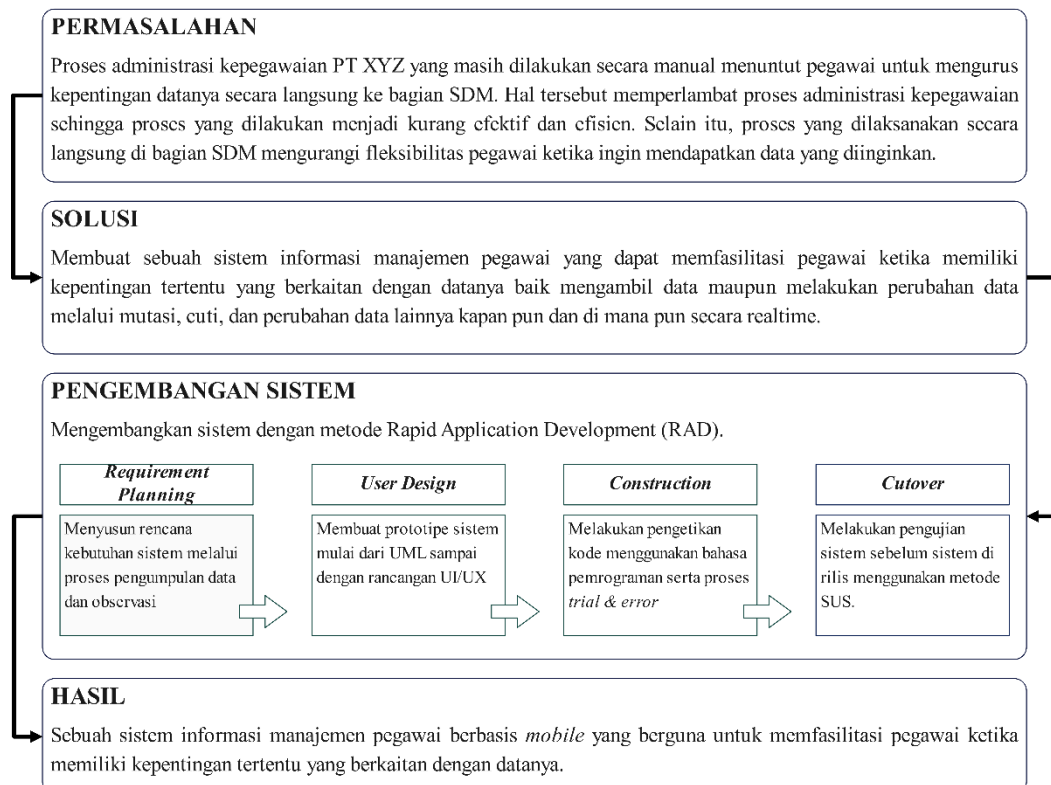
Permasalahan dalam proses administrasi tersebut dapat diselesaikan dengan membangun sebuah sistem informasi manajemen pegawai berbasis *mobile*. Sistem ini secara utuh terdiri dari 2 bagian utama yaitu *web* dan *mobile*. Penulis hanya berfokus pada pengembangan sistem berbasis *mobile* yang menggunakan *back-end* (REST API) yang sama dengan sistem *web*. Sistem ini dapat mempermudah pegawai ketika ingin mendapatkan data yang diinginkan tanpa harus menunggu proses pencarian di bagian SDM kapan saja dan dimana saja. Selain itu, sistem ini dapat menyediakan data pegawai yang dibutuhkan secara *real-time* ketika proses perubahan data seperti mutasi dan cuti. Sistem akan dibangun menggunakan bahasa pemrograman *JavaScript*, dengan kerangka kerja pengembangan aplikasi *mobile react native* dan komponen *user interface nativebase*. Manfaat yang diharapkan dari tugas akhir ini adalah untuk memfasilitasi kebutuhan data pegawai mulai dari data pribadi, data lamaran, riwayat pendidikan, riwayat mutasi sampai dengan riwayat cuti.

1.2. Tujuan

Tujuan tugas akhir ini adalah membangun sebuah sistem informasi manajemen pegawai berbasis *mobile* untuk diterapkan di PT XYZ. Sistem informasi ini bertujuan untuk memfasilitasi pegawai dalam proses pencarian data dan memudahkan pegawai ketika ingin mendapatkan data kepegawaian seperti data pribadi, data lamaran, riwayat pendidikan, riwayat mutasi dan riwayat cuti yang berkaitan dengan dirinya secara *real-time*.

1.3. Kerangka Pemikiran

Pengelolaan arsip kepegawaian pada PT XYZ saat ini masih dilakukan secara manual. Penyimpanan datanya masih disimpan di dalam lemari dan belum dicadangkan sehingga berisiko hilang. Hal ini dapat menurunkan kinerja proses administrasi instansi yang dilakukan oleh admin SDM ketika melayani pegawai yang membutuhkan datanya. Selain itu, proses pelayanan yang hanya dapat dilakukan di hari kerja dapat mengurangi fleksibilitas akses pegawai terhadap perubahan datanya. Untuk mengatasi permasalahan tersebut perlu adanya sebuah sistem informasi yang dapat membantu pegawai untuk mengakses data yang diinginkan dengan cepat kapan saja dan di mana saja secara *real-time*. Sebuah solusi yang muncul untuk menyelesaikan permasalahan tersebut adalah “Rancang Bangun Sistem Informasi Manajemen Pegawai Berbasis Mobile Pada PT XYZ Menggunakan Framework React Native” dengan menggunakan metode pengembangan RAD (*Rapid Application Development*). Tahapan kerangka pemikiran dalam pembuatan sistem ini disajikan pada Gambar 1.



Gambar 1. Kerangka pemikiran

1.4. Kontribusi

Sistem informasi manajemen pegawai berbasis *mobile* ini mampu memberikan kontribusi untuk beberapa pihak sebagai berikut:

1. Kontribusi bagi PT XYZ
 - a. Memfasilitasi proses administrasi pada PT XYZ menggunakan sebuah sistem informasi yang sebelumnya manual sehingga data yang disajikan telah sesuai dengan data yang terbaru secara *real-time*.
 - b. Mempermudah pegawai dalam mengakses datanya melalui perangkat *mobile* mereka tanpa harus mengurusnya langsung ke bagian SDM.
2. Kontribusi bagi Politeknik Negeri Lampung
 - a. Meningkatkan dan mengembangkan ilmu pengetahuan
 - b. Menjadi bahan rujukan dan referensi bagi mahasiswa yang akan atau sedang melakukan penelitian tugas akhir dengan tema sejenis

BAB II. TINJAUAN PUSTAKA

2.1. Sistem Informasi Manajemen

Sistem informasi manajemen merupakan suatu metode pengorganisasian yang saling berkaitan dan saling berinteraksi antar komponen dalam sebuah kesatuan untuk menghasilkan informasi yang dibutuhkan manajer. Sistem informasi manajemen memfasilitasi proses pengambilan keputusan dalam menerapkan fungsi manajemen seperti perencanaan, pengendalian, dan pengorganisasian suatu instansi supaya dapat dilaksanakan dengan cepat dan efektif. Sistem informasi manajemen bertujuan untuk mengatasi berbagai permasalahan terkait dengan pengaturan suatu instansi dalam menjalankan proses bisnisnya seperti permasalahan layanan, biaya operasional, strategi bisnis, dan proses operasional instansi lainnya. Terdapat 5 komponen utama yang saling berkaitan dalam satu kesatuan dalam sistem informasi manajemen yaitu manusia (*brainware*), prosedur bisnis, data, perangkat keras (*hardware*) dan perangkat lunak (*software*) (Sadikin & Wiranda, 2022). Berikut ini manfaat penggunaan sistem informasi manajemen (Hutahaeen dkk., 2021).

1. Mempermudah dalam perencanaan sehingga lebih efektif dan efisien.
2. Menjadi sarana untuk menganalisis pelaksanaan dan keperluan.
3. Meningkatkan produktivitas dan kinerja instansi.
4. Mengurangi biaya operasional instansi.
5. Menghasilkan informasi yang aktual dan *real-time* bagi pengguna yang membutuhkan tanpa perantara.

2.2. Arsip

Arsip adalah semua rekaman yang terekam dalam berbagai media baik tertulis, berupa gambar, atau berupa rekaman (dalam bentuk audio atau video) yang dihasilkan oleh suatu instansi. Suatu dokumen dapat dianggap sebagai arsip jika dalam dokumen tersebut mengandung informasi yang penting bagi sebuah instansi pada masa lalu, masa kini, maupun di masa yang akan datang. Pengelolaan arsip di sebuah instansi harusnya dikelola dengan baik dan teratur untuk menghindari

kesalahpahaman informasi yang disajikan sehingga pengambilan keputusan dapat dilakukan dengan benar (Rosalin, 2017).

Menurut Undang-Undang Republik Indonesia No. 43 Tahun 2009 tentang Kearsipan, Arsip dibagi menjadi beberapa jenis yaitu :

1. Arsip dinamis merupakan arsip yang dapat digunakan secara langsung dan dalam jangka waktu tertentu.
2. Arsip statis merupakan arsip yang memiliki nilai kesejarahan atau yang sudah habis masa retensinya.
3. Arsip vital merupakan arsip yang dipergunakan sebagai syarat dasar dalam proses operasional. Arsip ini tidak dapat diperbarui, dan keberadaannya tidak dapat digantikan apabila hilang atau rusak.
4. Arsip aktif merupakan arsip yang sering digunakan secara terus-menerus.
5. Arsip terjaga merupakan arsip yang berkaitan dengan negara yang keberadaannya mempengaruhi keberlangsungan hidup bangsa dan negara yang harus dilindungi, dijaga keamanannya, dan keutuhannya.
6. Arsip umum merupakan arsip yang berkategori selain arsip terjaga.

Berikut ini beberapa jenis arsip lainnya (Sugiarto & Wahyono, 2014).

1. Arsip berdasarkan subjeknya yaitu arsip kepegawaian, arsip pemasaran, arsip keuangan, arsip pendidikan, dan sebagainya.
2. Arsip berdasarkan bentuk medianya yaitu berkas surat, arsip digital, rekaman baik audio maupun video, berkas gambar, dan lain-lain.
3. Arsip berdasarkan nilai kegunaannya yaitu :
4. Arsip yang bernilai administrasi seperti prosedur kerja.
5. Arsip yang bernilai hukum seperti akta kelahiran, akta tanah, dll.
6. Arsip yang bernilai pendidikan seperti silabus, kurikulum, dll.
7. Arsip yang bernilai sejarah seperti laporan bulanan, tahunan, dll.
8. Arsip yang bernilai keuangan seperti kuitansi, nota pembayaran, dll.
9. Arsip yang bernilai informasi seperti pengumuman dan undangan.
10. Arsip yang bernilai ilmiah seperti laporan penelitian, skripsi, dll.
11. Arsip berdasarkan fungsinya yaitu arsip dinamis dan arsip statis.
12. Arsip berdasarkan tempat pengelolaannya yaitu arsip pusat dan arsip unit.

13. Arsip berdasarkan tingkatan keasliannya yaitu arsip asli, arsip tembusan, arsip salinan, dan arsip petikan.
14. Arsip berdasarkan kekuatan hukumnya yaitu arsip otentik, dan arsip tidak otentik.
15. Arsip berdasarkan kepentingannya yaitu :
 - a. Arsip yang tidak berguna seperti surat undangan.
 - b. Arsip yang berguna seperti surat cuti, surat izin, dan presensi pegawai.
 - c. Arsip yang penting seperti laporan keuangan, surat keputusan, dll.
 - d. Arsip yang vital seperti akta kelahiran, ijazah, sertifikat pelatihan, dll.

Salah satu jenis arsip berdasarkan subjeknya yaitu arsip kepegawaian. Arsip kepegawaian merupakan semua arsip yang ada hubungannya dengan masalah kepegawaian, seperti surat lamaran, daftar riwayat hidup, data pribadi, surat-surat keputusan pegawai, absensi pegawai, dan lain-lain (Suparman, 2020).

2.3. Aplikasi *Mobile*

Perangkat lunak (*software*) merupakan serangkaian detail instruksi yang ditulis menggunakan bahasa pemrograman untuk memberitahu komputer apa yang harus dilakukan guna mencapai sebuah tujuan tertentu. Keberadaan perangkat lunak berperan penting sebagai penghubung antara pengguna dengan perangkat keras (*hardware*) komputer, karena tanpa adanya perangkat lunak komputer hanyalah kumpulan perangkat keras yang tidak dapat dioperasikan (Fox, 2013). Secara umum perangkat lunak dibagi menjadi dua jenis yaitu:

1. Perangkat lunak sistem

Perangkat lunak sistem merupakan kumpulan program atau instruksi yang dibuat untuk membentuk sebuah sistem operasi yang menghubungkan perangkat keras komputer dengan perangkat lunak aplikasi. Komponen inti dari sebuah perangkat lunak sistem disebut sebagai *kernel* yang dijalankan saat pertama kali komputer dihidupkan. *Kernel* memiliki beberapa tugas yaitu menangani manajemen proses, manajemen sumber daya, dan manajemen memori pada komputer. Tanpa adanya *kernel* perangkat lunak aplikasi tidak dapat dijalankan secara efisien atau bahkan tidak dapat dijalankan sama sekali. Beberapa contoh sistem operasi yang umum digunakan adalah *linux*, *windows*, dan *macOS*.

2. Perangkat lunak aplikasi

Perangkat lunak aplikasi adalah kumpulan program yang dijalankan pengguna untuk menyelesaikan beberapa tugas atau fungsi-fungsi lainnya yang berjalan di atas perangkat lunak sistem. Pengelompokan perangkat lunak aplikasi bergantung pada kegunaannya bagi pengguna.

- a. Aplikasi pendukung produktivitas yang digunakan semua orang mencakup aplikasi pengolah kata, aplikasi pengolah angka, aplikasi presentasi, sistem manajemen data, aplikasi kontak, aplikasi pengatur data dll.
- b. Aplikasi desain yang digunakan para desainer mencakup aplikasi menggambar, aplikasi edit gambar, aplikasi pengolah *vector*, aplikasi 3D, dan aplikasi pengolah video.
- c. Aplikasi pengolah suara yang digunakan para musisi.
- d. Aplikasi pengembangan yang digunakan para *programmer* mencakup aplikasi editor kode, aplikasi IDE, aplikasi pelacakan perubahan kode, *library* atau *framework*, aplikasi *debugging*, aplikasi visualisasi kode, dll.
- e. Aplikasi daring mencakup aplikasi yang menggunakan konektivitas internet ketika penggunaannya seperti aplikasi email, browser web, aplikasi FTP, aplikasi *remote*, dsb.
- f. Aplikasi lainnya mencakup aplikasi di luar kategori aplikasi yang dijelaskan di atas seperti aplikasi hiburan (pemutar musik dan video), dan permainan.

Perangkat lunak dapat berjalan di beberapa *platform* seperti komputer, perangkat *mobile*, perangkat jaringan, atau bahkan perangkat *embedded system*. Perangkat lunak yang dijalankan menggunakan perangkat *mobile* disebut aplikasi *mobile*. Aplikasi *mobile* merupakan kumpulan beberapa perangkat lunak atau program yang berjalan pada perangkat *mobile* yang mampu menjalankan tugas-tugas tertentu berdasarkan keinginan pengguna. Aplikasi *mobile* merupakan terobosan terbaru akibat berkembang pesatnya teknologi informasi dan komunikasi. Kemudahan penggunaan, tampilan yang *user-friendly*, mudah didapat, mudah diunduh, dan dapat dijalankan di berbagai perangkat *mobile* merupakan kelebihan aplikasi *mobile*. Penggunaannya yang luas mulai dari sarana komunikasi, menjelajah internet, jejaring sosial, pendidikan, bisnis, sampai dengan hiburan

mampu terpenuhi oleh aplikasi *mobile*. Jumlah pasar aplikasi *mobile* yang besar berbanding lurus dengan bertambahnya jumlah pengembang, penerbit, dan penyedia aplikasi menjadikan aplikasi *mobile* sebuah teknologi baru yang menguntungkan penggunanya (Islam dkk., 2010).

Aplikasi *mobile* saat ini dapat dikembangkan dengan berbagai bahasa pemrograman seperti *Java*, *Kotlin*, *Dart*, *Objective-C*, *Swift*, dan *JavaScript*. Selain itu para pengembang telah berhasil mengembangkan berbagai *framework* yang dapat digunakan untuk mengembangkan aplikasi *mobile* baik Android, IOS, maupun lintas *platform* dengan stabil dan lebih cepat. Salah satu *framework* yang bisa digunakan untuk mengembangkan aplikasi *mobile* yaitu *React Native* dengan bahasa pemrograman *JavaScript*.

2.3.1. JavaScript

JavaScript merupakan bahasa pemrograman tingkat tinggi dinamis yang mampu dikembangkan dengan gaya pemrograman fungsional atau gaya pemrograman berorientasi objek. *JavaScript* merupakan bahasa pemrograman yang memerlukan interpreter untuk menjalankan kodenya. Dahulu *JavaScript* hanya dapat dijalankan di dalam lingkungan browser, namun satu dekade terakhir muncul *runtime-environment* yang memfasilitasi *JavaScript* supaya dapat berjalan di luar lingkungan browser yaitu *Node.js*. Keberhasilan *Node.js* membawa *JavaScript* keluar dari lingkungan browser membuat *JavaScript* sekarang menjadi bahasa pemrograman yang paling banyak digunakan di kalangan pengembang perangkat lunak (Flanagan, 2020). Selain pengembangan web, saat ini para pengembang *JavaScript* telah mengembangkan *framework* yang dapat digunakan untuk mengembangkan aplikasi *mobile*. *Framework* yang populer yaitu *React Native* dan *Ionic*.

2.3.2. React Native

React Native adalah salah satu kerangka kerja (*framework*) *JavaScript* besutan Facebook yang digunakan untuk mengembangkan aplikasi baik aplikasi *mobile* (Android dan IOS) maupun aplikasi web. Kerangka kerja *React Native* dibuat berdasarkan *library React*. Dengan kata lain, *React Native* memungkinkan pengembang web untuk membuat aplikasi *mobile* yang mirip seperti aplikasi *mobile* yang dibuat secara *native* tanpa harus beralih ke kerangka kerja dan bahasa

pemrograman yang lain. Mirip seperti *React*, *React Native* ditulis menggunakan campuran bahasa pemrograman *JavaScript* dan *markup XML-esque* atau lebih dikenal sebagai *JSX*. *React Native* menghubungkan kode *JSX* yang ditulis dengan *API rendering* asli milik masing-masing *platform* sehingga aplikasi yang dibuat akan ditampilkan menggunakan komponen UI masing-masing *platform* dan akan terasa seperti aplikasi *mobile* lainnya (Eisenman, 2015).

2.3.3. *NativeBase*

Library komponen UI/UX (*User Interface/User Experience*) merupakan kumpulan serangkaian kode komponen yang siap digunakan pengguna untuk kebutuhan tertentu. Dengan menggunakan komponen yang sudah disediakan memungkinkan pengguna untuk menyusun tata letak tampilan dan pengalaman pengguna dengan cepat tanpa membuang waktu untuk merancanginya dari awal. Selain itu keuntungan menggunakan *library* komponen sering kali menghasilkan hasil yang lebih stabil dan konsisten baik dari segi UI maupun UX. Salah satu *library* yang populer digunakan dalam pengembangan aplikasi *React Native* adalah *NativeBase* (Boduch dkk., 2022).

2.3.4. *Application Programming Interface*

Application Programming Interface (API) merupakan kumpulan aturan yang ditentukan oleh pengembang guna memfasilitasi aplikasi untuk dapat berkomunikasi dengan aplikasi lain. API bertindak sebagai perantara yang memproses pertukaran data antar sistem atau aplikasi. Hal ini memungkinkan perusahaan membuka akses dan fungsionalitas aplikasinya untuk dapat dikembangkan atau dihubungkan dengan pengembang pihak ketiga, mitra bisnis, atau divisi internal perusahaan. Manfaat mengembangkan API adalah untuk membantu proses bisnis perusahaan dengan menghubungkan banyak aplikasi berbeda yang digunakan sehingga menghemat waktu pengerjaan dan memudahkan pengembang berkolaborasi dan berinovasi (IBM Cloud Education, 2022).

API menggunakan beberapa protokol yang sering dijumpai dalam pengembangan aplikasi yaitu:

1. SOAP (*Simple Object Access Protocol*)

API SOAP dibangun dengan menggunakan XML yang memungkinkan *endpoint* untuk mengirim dan menerima data melalui SMTP dan HTTP.

2. XML-RPC (*XML-Remote Procedure Call*)

Protokol XML-RPC menggunakan format XML tertentu untuk mentransfer datanya. XML-RPC dibuat sebelum adanya SOAP, akan tetapi jauh lebih sederhana, dan relatif ringan karena menggunakan *bandwidth* yang kecil.

3. JSON-RPC (*JSON-Remote Procedure Call*)

JSON-RPC hampir sama dengan XML-RPC, pembedanya hanya penggunaan JSON (*JavaScript Object Notation*) untuk mentransfer data dan tidak lagi menggunakan XML.

4. REST (*Representational State Transfer*)

REST adalah seperangkat prinsip pengembangan arsitektur web API.

2.3.5. Representational State Transfer

Representational State Transfer atau REST merupakan arsitektur web yang menyediakan proses pertukaran data antara *server* dan *client* dengan menggunakan API yang terhubung secara *point-to-point*. Arsitektur REST menggunakan prinsip *client-server*. Dengan kata lain *server* REST bertanggung jawab untuk menyediakan sumber daya yang diminta oleh *client* REST ketika melakukan pembuatan, pengambilan, pengubahan, dan penghapusan sumber daya. Arsitektur REST menggunakan format XML dan JSON (*JavaScript Object Notation*) sebagai media pertukaran datanya (Chatterjee & Mamatha, 2020). Ada beberapa prinsip yang perlu diperhatikan ketika mengembangkan REST API yaitu:

1. *Addressability*

Setiap sumber daya harus memiliki setidaknya satu URI (*Uniform Resource Identifier*) yang terkait. URI digunakan untuk menentukan sumber daya atau sekumpulan sumber daya.

2. *Statelessness*

Layanan REST adalah layanan yang independen. Setiap permintaan yang dikirimkan menggunakan protokol HTTP (*Hypertext Transfer Protocol*) tidak berhubungan dengan permintaan sebelumnya.

3. *Cacheable*

Sumber daya yang ditandai sebagai *cache* dapat disimpan sementara dalam sistem dan dapat digunakan kembali ketika permintaan menghasilkan hasil yang sama. Penandaan sumber daya dapat dilakukan ketika permintaan dimulai.

4. *Uniform Interface*

Layanan REST menggunakan kumpulan metode HTTP standar untuk menentukan permintaan yang dibuat. Berikut ini beberapa metode HTTP sederhana yang dapat digunakan dalam layanan REST. Berikut ini metode HTTP yang umum digunakan.

Tabel 1. Metode HTTP

Metode	Penjelasan
GET	Digunakan untuk mengambil data (<i>Read</i>).
POST	Digunakan untuk <i>input</i> data (<i>Create</i>).
PUT	Digunakan untuk <i>input</i> data, apabila data sudah ada maka data tersebut akan diperbarui (<i>Create/Update</i>).
PATCH	Digunakan untuk memperbarui data (<i>Update</i>)
DELETE	Digunakan untuk menghapus data (<i>Delete</i>)

Sumber: (MDN Mozilla, 2022)





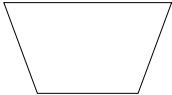

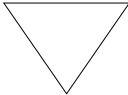
2.4. Perancangan Sistem

Perancangan sistem merupakan proses merinci pembuatan sistem menggunakan berbagai teknik, mulai dari deskripsi arsitektur pendukung dan detail komponen sampai dengan kendala pengerjaan yang bertujuan untuk memberikan gambaran yang jelas dalam memenuhi kebutuhan pengguna. Perancangan sistem bisanya digambarkan dalam bentuk diagram alir yang menampilkan proses secara runtut dan diagram-diagram pendukung lainnya (Aziz, N., Pribadi, G., & Nurcahya, 2020).


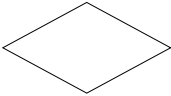

2.4.1. Mapping Chart

Mapping chart merupakan diagram alir yang memvisualisasikan langkah-langkah dalam bentuk simbol-simbol grafis yang digunakan untuk membantu proses analisis, penelitian, atau untuk memecahkan masalah tertentu. *Mapping chart* berfungsi untuk menyederhanakan proses agar lebih mudah dilihat dan dipahami menggunakan media simbol-simbol (Mukodimah dkk., 2019).

Tabel 2. Simbol-simbol *mapping chart*

Simbol	Nama	Fungsi
(1)	(2)	(3)
	Dokumen	Menyatakan <i>input</i> dokumen yang dicetak
	Dokumen rangkap	Menyatakan dokumen yang lebih dari satu
	<i>Connector</i>	Menyatakan sambungan satu dengan yang lainnya pada proses berikutnya
	Proses	Proses pengolahan yang ditugaskan oleh komputer
	Proses manual	Proses pengolahan yang dilakukan secara manual
	<i>Display</i>	Menyatakan output yang digunakan
	Arsip	Mengarsipkan data di dalam program

Tabel 2. Lanjutan simbol-simbol *mapping chart*

Simbol	Nama	Fungsi
(1)	(2)	(3)
	Terminal	Awalan dan akhiran pada program
	<i>Desicion</i>	Keputusan ya atau tidak dalam proses pengolahan data
	<i>Connecting Line</i>	Menghubungkan simbol dengan simbol lainnya dengan menyatakan suatu alur proses

Sumber: (Fitriyana & Susianto, 2018)



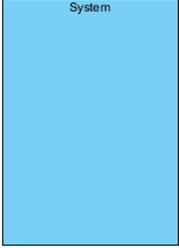

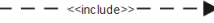

2.4.2. *Unified Modelling Language*

Unified Modelling Language atau UML merupakan pemodelan visual yang digunakan untuk mendefinisikan, memvisualisasikan, dan mendokumentasikan rancangan analisis dan desain aplikasi, serta memvisualisasikan arsitektur dalam OOP (*Object Oriented Programming*). UML menjadi sarana yang digunakan untuk memberikan gambaran jelas terkait analisis sistem sebelum dibangun baik secara struktural maupun secara fungsional (Putra & Andriani, 2019). Dalam UML terdapat beberapa diagram yang sering digunakan yaitu:

1. *Use Case Diagram*

Use case diagram merupakan diagram yang memodelkan perlakuan (*behavior*) sistem yang akan dibuat kepada penggunanya. *Use case* digunakan untuk mendeskripsikan interaksi antara satu atau lebih pengguna dengan sebuah sistem. Melalui *use case* pengembang mendapat gambaran bagaimana interaksi pengguna dengan sistem dan fungsi apa saja yang ada pada sistem tersebut. Notasi *use case* dibagi menjadi beberapa bagian yaitu pengguna (*actor*), sistem atau sub-sistem (*use case*), dan hubungan (*relationship*).

Tabel 3. Simbol-simbol diagram *use case*




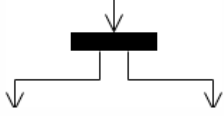
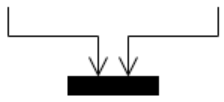


Simbol	Nama	Fungsi
	Aktor	Menggambarkan pengguna yang berinteraksi dan berperan dalam sistem bisnis.
	<i>Use Case</i>	Menggambarkan fungsi yang dapat dilakukan sebuah sistem bisnis.
	Batas Sistem	Menggambarkan batasan sistem bisnis untuk kasus penggunaan khusus.
	<i>Association</i>	Relasi asosiasi antara <i>use case</i> dengan pengguna.
	<i>Includes</i>	Relasi ketika sebuah <i>use case</i> digambarkan menggunakan fungsionalitas dari <i>use case</i> yang lain.
	<i>Extends</i>	Relasi yang digunakan untuk menyertakan perilaku opsional dari <i>use case</i> yang diperluas.

Sumber: (Visual Paradigm, 2022b)

2. *Activity Diagram*

Activity diagram merupakan diagram yang memvisualisasikan alir kerja (*workflow*) atau aktivitas dari sebuah sistem yang menjalankan proses bisnis atau menu yang ada pada perangkat lunak. Dengan kata lain *activity* diagram hanya menggambarkan aktivitas yang berjalan pada sistem bukan aktivitas yang dilakukan pengguna (*actor*).

Tabel 4. Simbol-simbol *activity diagram*

Simbol	Nama	Fungsi
	<i>Initial Node</i>	Menggambarkan titik awal dari serangkaian tindakan atau kegiatan.
	<i>Activity</i>	Menggambarkan serangkaian tindakan atau aktivitas sistem.
	<i>Decision Node</i>	Menggambarkan seleksi kondisi untuk memastikan bahwa aliran kontrol atau aliran aktivitas hanya turun satu jalur.
	<i>Fork Node</i>	Membagi aktivitas sistem menjadi serangkaian aktivitas (atau tindakan) paralel atau bersamaan.
	<i>Join Node</i>	Menyatukan kembali aktivitas atau tindakan sistem yang paralel atau bersamaan.
	<i>Swimlane and Partition</i>	Mengelompokkan aktivitas yang dilakukan oleh aktor yang sama atau pengelompokan aktivitas dalam satu partisi.
	<i>Activity Final Node</i>	Menghentikan semua aliran kontrol dan aliran aktivitas atau tindakan

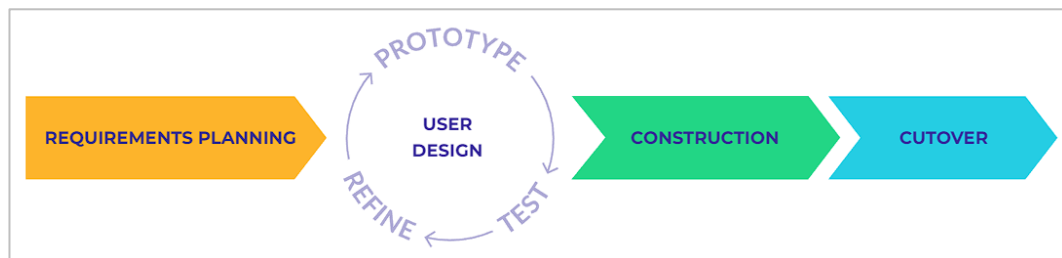
Sumber: (Visual Paradigm, 2022a)

2.5. *Rapid Application Development (RAD)*

Rapid Application Development atau RAD merupakan salah satu metode pengembangan sistem perangkat lunak yang bersifat bertahap (*incremental*). Dalam penerapannya, metode RAD mementingkan waktu siklus pengembangan yang relatif singkat, cepat, pendek, dan berulang (*iterative*) (Kaban dkk., 2022). Metode RAD menggunakan pendekatan yang berfokus pada tahap perancangan dan pembuatan prototipe dengan tujuan mendapatkan umpan balik pengguna secara instan. Perulangan yang konstan dalam mengembangkan sistem berdasarkan umpan balik pengguna dan pembaruan yang cepat membantu mencapai hasil yang

sesuai dengan keinginan pengguna. Berikut ini keuntungan menggunakan metode RAD (Glaschenko, 2021).

1. Menghasilkan kualitas sistem yang tinggi
Keterlibatan pengguna dalam tahap pembuatan prototipe, sistem yang dihasilkan akan lebih relevan dengan keinginan dan ekspektasi pengguna.
2. Meminimalisir biaya dan risiko pengembangan
Ketika menggunakan metode *waterfall*, pengguna hanya dapat melihat hasil dan memberikan umpan balik ketika proyek diluncurkan. Perubahan yang terjadi pada proses ini memakan biaya yang mahal dan memakan lebih banyak waktu. Dengan metode pengembangan aplikasi yang cepat, risiko penulisan ulang dari proyek yang telah diluncurkan menjadi lebih minim.



Gambar 2. Fase metode RAD

Metode RAD memiliki beberapa fase dalam pengerjaannya. Berikut ini fase yang harus dilalui ketika mengembangkan sebuah sistem menggunakan metode RAD.

1. *Requirements Planning*

Pada fase ini, pengguna dan tim proyek mengidentifikasi tujuan sistem yang diinginkan. Pengembangan nantinya berfokus untuk mencapai tujuan bisnis dengan kebutuhan yang telah ditetapkan. Kebutuhan yang sebelumnya sudah ditetapkan dapat diubah dan disesuaikan kembali pada tahap pembuatan prototipe.

2. *User Design*

User design merupakan fase penting dari metode RAD. Pada fase ini, pengembang dimulai dengan mengerjakan prototipe. Tujuan membuat prototipe adalah untuk mendemonstrasikan sebuah gagasan tentang sistem yang akan dibuat kepada klien dengan cepat. Tim mengumpulkan semua

umpan balik dari pengguna berdasarkan prototipe yang telah disampaikan. Pada proses inilah kebutuhan awal dapat disesuaikan kembali berdasarkan keinginan pengguna. Dengan umpan balik yang telah dikumpulkan, pengembang akan mengulang langkah pembuatan prototipe, hingga pengguna puas dengan hasilnya.

3. *Construction*

Pada fase ini, pengembangan dan pengujian dilakukan untuk mempersiapkan sistem menuju ke tahap produksi. Dalam pengerjaannya fokus utamanya adalah pada kualitas, skalabilitas, pemeliharaan, dll. Namun, pengguna terus berpartisipasi pada fase ini dan terus memberikan umpan balik saat fitur diimplementasikan. Sedikit penyempurnaan dimungkinkan pada fase ini.

4. *Cutover*

Cutover merupakan fase terakhir. Dalam fase ini mencakup *acceptance testing*, peluncuran (*rollout*) dan pelatihan pengguna.

2.6. *System Usability Scale (SUS)*

Pengujian sistem merupakan proses mencari kesalahan pada setiap komponen sistem, melakukan pencatatan, mengevaluasi semua aspek dan fitur-fitur sistem yang sedang dikembangkan. Berikut ini beberapa aturan dalam pengujian sistem (W. Wibisono & Baskoro, 2002).

1. Pengujian dilakukan dengan mengeksekusi program dengan tujuan menemukan kesalahan
2. Sebuah kasus pengujian dikatakan baik jika dalam proses pengujiannya berpotensi menemukan kesalahan lebih tinggi.
3. Pengujian dikatakan berhasil jika menemukan kesalahan.

System Usability Scale merupakan metode evaluasi sebuah aplikasi untuk menilai dan mengukur tingkat kegunaannya dengan menggunakan sebuah kuesioner sederhana yang memiliki sepuluh pernyataan. Pernyataan metode SUS yang bernomor ganjil menggunakan kalimat positif dan pernyataan yang bernomor genap menggunakan kalimat negatif. Selanjutnya untuk menilai hasil kuesioner tersebut menggunakan skala *likert* yang telah ditentukan (Muhammad Nur Fauzi dkk., 2022).

Kelebihan menggunakan SUS adalah:

1. SUS mudah digunakan karena tidak menggunakan perhitungan yang rumit.
2. Rentang skor SUS bernilai 0-100.
3. Akurat walau menggunakan sampel yang kecil.
4. Gratis tanpa biaya.

2.7. Jurnal terkait

Berikut ini beberapa jurnal terkait yang menjadi bahan pertimbangan dan menjadi bahan referensi pengembangan sistem.

1. Adam dkk. (2022). Dengan judul “Aplikasi Jasa Titip Belanja Berbasis Mobile di Minahasa Utara”. Karya ilmiah ini bertujuan untuk membangun sebuah aplikasi jasa titip belanja supermarket di daerah Minahasa Utara. Aplikasi yang akan dibangun akan tersedia di platform Android untuk Pelanggan dan Kurir serta *website* untuk admin dan juga toko, dengan sisi *front-end* dikembangkan dengan bahasa pemrograman Javascript menggunakan *framework* React Native, React JS, dan sisi *back-end* dikembangkan lewat bahasa pemrograman Javascript dengan memanfaatkan teknologi firebase *real-time database*.
2. Karim & Adriansyah (2022). Dengan judul “Analisis dan Perancangan Aplikasi Mobile untuk Donasi menggunakan Metode Hybrid berbasis React Native”. Karya ilmiah ini dibuat untuk memudahkan pengguna untuk berdonasi barang, dimana pun dan kapan pun. Aplikasi dibangun menggunakan *framework* React Native dengan RESTful API dan MySQL sebagai bahasa pemrograman *database*. Pengujian yang dilakukan yaitu pengujian fungsional sistem dengan menggunakan teknik *black-box testing* dan *User Acceptance Test* (UAT) serta pengujian dengan menggunakan kuesioner.
3. Nursaid dkk. (2020). Dengan judul “Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus : Toko Uda Fajri)”. Karya ilmiah ini dibuat untuk mempermudah pedagang yang memiliki toko Uda Fajri tanpa harus mencatat barang masuk dan keluar dengan menggunakan kertas lagi, selain itu juga dapat melakukan transaksi barang dengan menggunakan telepon genggam yang sudah terpasang aplikasi sistem pengelolaan barang.

4. Mukodimah dkk. (2019). Dengan Judul “Aplikasi Penentuan Bengkel TSM Berkualitas Untuk UKK Siswa SMK Kabupaten Pringsewu Berbasis Mobile”. *Website* untuk mengukur kelayakan lab/bengkel uji kompetensi kejuruan TSM dapat dijadikan sebagai solusi dalam menyelesaikan permasalahan pengukuran kelayakan bengkel TSM SMK secara tepat dan akurat. Dengan menerapkan 5 kriteria yang ada seperti area kerja mesin otomotif, area kerja kelistrikan, area kerja *chasis* dan pemindah tenaga, ruang penyimpanan dan instruktur dan kelengkapan peralatan. Proses pengukuran kelayakan dapat dilakukan dengan lebih tepat, akurat dan efisien.
5. Prastio & Ani (2018). Dengan judul “Aplikasi Self Service Menu Menggunakan Metode Scrum Berbasis Android (Case Study : Warkobar Café Cikarang)”. Sebuah aplikasi yang diharapkan membantu pelanggan *café*, pelayan, kasir, dapur, dan pemilik *café*. Perancangan sistem akan dibuat dengan tampilan sederhana dan mudah dipahami oleh pelanggan. Dengan begitu pemesanan menu bisa berjalan dengan cepat dan praktis, tanpa melalui proses panjang serta mengurangi kemungkinan terjadinya kesalahan komunikasi antara pelayan dan pelanggan. Sistem ini juga akan meminimalisir adanya antrean tempat duduk dalam *café*, dengan adanya fungsi reservasi dengan pemilihan tempat dan waktu yang sudah ditentukan oleh pelanggan. Sistem ini juga akan menggunakan OS Mobile populer saat ini, yaitu Android.