

BAB II

TINJAUAN PUSTAKA

Dalam bab ini peneliti akan menjelaskan hal-hal berkenaan pada peninjauan kembali terhadap suatu pustaka yang berhubungan dalam topik penelitian sebagai bahan perbandingan dan referensi terhadap beberapa teori terkait dalam tugas akhir yang berjudul Pengembangan Fitur Mobile-Apps Laporan-Mak yang terintegrasi Dashboard Application berbasis Web. Pengembangan aplikasi yang dimaksud adalah aplikasi mobile-apps dan dashboard berbasis web, maka teori-teori dasar yang akan dijelaskan konsep pengembangan aplikasi, perancangan aplikasi dan sistem database yang akan diterapkan.

2.1. Teori Dasar

2.1.1. Definisi Pengembangan Aplikasi

Definisi dari Pengembangan yang dikutip dari UU RI No.18 Tahun 2002 yang membahas Tentang Sistem Nasional Penelitian, Pengembangan, Penerapan IPTEK (Ilmu Pengetahuan dan Teknologi) menjelaskan bahwa *“Pengembangan merupakan kegiatan IPTEK atau ilmu pengetahuan dan teknologi yang memiliki tujuan untuk memberikan manfaat terhadap teori ilmu pengetahuan yang telah terbukti kebenarannya dalam meningkatkan manfaat, fungsi dan aplikasi ilmu pengetahuan dan teknologi yang sudah ada atau menghasilkan suatu teknologi baru”*. Kata Aplikasi berasal dari bahasa *“To Applicate”* yang artinya menerapkan. Pengertian Aplikasi secara umum merupakan sekumpulan sistem yang terhubung dan dapat digunakan. Menurut Jogiyanto (1999), *“Aplikasi adalah sesuatu yang dapat digunakan*

dalam komputer yang berisi pernyataan (statement) atau instruksi (instruction) yang telah disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output”.

Pengembangan Aplikasi dapat dikembangkan artinya sebagai serangkaian kegiatan dalam IPTEK (ilmu pengetahuan dan teknologi) pada tahap menyusun atau membuat perancangan dalam pernyataan (*statement*) atau instruksi (*instruction*) pada komputer dalam menghasilkan proses masukan (*input*) menjadi keluaran (*output*) dalam meningkatkan manfaat, fungsi dan ilmu pengetahuan dan teknologi yang ada saat ini untuk menghasilkan teknologi baru.

2.1.2. Pengertian Analisis dan Desain Sistem

Pengertian analisis dan desain sistem menurut ahli Mulyani (2017) adalah *“Penentuan suatu tahapan terhadap kebutuhan data yang diperlukan oleh sistem baru dan merupakan tujuan dari perancangan sistem dalam memenuhi kebutuhan pengguna sistem (user systems requirement) serta memberikan gambaran yang detail terhadap rancang bangun yang kumulit”.*

Menurut Muharto (2016) definisi analisis dan desain sistem adalah *“Tahapan dimana diperlukannya suatu keahlian perancangan dalam elemen-elemen komputer yang akan menggunakan sistem yaitu pemilihan peralatan dan program komputer untuk sistem yang baru”.*

Dalam kegiatan analisis dan desain sistem terdapat teknik dalam memecahkan suatu permasalahan. Menurut Whitten & Bentley (2007), terdapat beberapa pendekatan masalah terhadap analisis dan desain sistem, yaitu:

- *Model-Driven Approaches (Pendekatan Berbasis Model)*

Merupakan pendekatan pemecahan masalah berfokus pada gambaran model pembelajaran dan pengembangan sebuah sistem kedalam dokumen dengan memvalidasikan sistem.

- *Structured Analysis (Analisis Terstruktur)*

Pendekatan pemecahan masalah ini merupakan teknik *model-driven* yang fokus dalam proses yang digunakan yaitu mendefinisikan persyaratan-persyaratan bisnis dalam sebuah sistem.

- *Object Oriented Analysis (Analisis Berorientasi Objek)*

Pendekatan pemecahan masalah ini merupakan teknik yang berelasi terhadap data dan proses ke dalam konstruksi yang disebut Object dengan mengilustrasikan objek-objek sistem terhadap beragam interaksi objek, perspektif, struktur dan operasional.

- *Information Engineering (Teknik Informasi)*

Pendekatan pemecahan masalah ini merupakan teknik teknik *model-driven* yang terpusat pada data dan sensitif pada proses dalam mengilustrasikan data dan proses sistem.

- *Discovery Prototyping (Prototyping Penemuan)*

Pendekatan pemecahan masalah ini merupakan teknik dalam mengidentifikasi persyaratan bisnis terhadap user atau pengguna dengan membuat para pengguna bereaksi pada implementasi yang efektif tanpa efek samping dan kesalahan terhadap semua persyaratan.

2.1.3. Elisitasi

Hasil wawancara dan pengamatan akan dikumpulkan dan

diklasifikasikan dan dilakukan melalui tiga tahapan Elisitasi. Menurut (Sarah Riwanda Shofroh, 2015) pada artikel i-learning teknik elisitasi memiliki tiga tahapan, yaitu:

✓ *Tahap Pertama*; Tahapan awal dalam merencanakan pengembangan pada sistem yang diusulkan terhadap hasil pengamatan dan wawancara.

✓ *Tahap Kedua*; Tahapan Selanjutnya mengklasifikasikan hasil tahap pertama untuk dipisahkan berdasarkan *metode MDI* antara rencana pengembangan sistem baru dengan rancangan yang disanggupi oleh penulis untuk dieksekusi. penjelasan mengenai Metode MDI:

- **M** dalam MDI atau **Mandatory (Penting)**. Maksudnya kebutuhan (*requirement*) tersebut harus ada dan tidak dapat dihilangkan ketika pembuatan sistem baru.
- **D** dalam MDI atau **Desirable (Diperlukan)**. Maksudnya kebutuhan (*requirement*) tersebut dapat dihilangkan atau tidak terlalu penting. Tetapi jika kebutuhan tersebut di implementasikan dalam pembentukan sistem, akan membuat sistem tersebut lebih baik dan sempurna.
- **I** dalam MDI atau **Inessential (Tidak Penting)**. Maksudnya bahwa kebutuhan (*requirement*) tersebut tidak termasuk pada bagian dari sistem yang dibahas atau bagian dari luar sistem.

✓ *Tahap Ketiga*; hasil penyusutan dari tahapan elisitasi kedua Selanjutnya adalah mengeliminasi penyesuaian kebutuhan sistem yang dilanjutkan akan semua kebutuhan (*requirement*) yang tersisa akan diklasifikasikan lagi menggunakan *metode TOE*, yaitu:

- **Technical (T)**, merupakan tata cara atau teknik pembuatan kebutuhan terhadap sistem yang diusulkan?

- **Operational (O)**, merupakan tata cara penggunaan kebutuhan dalam sistem yang akan dikembangkan?
- **Economic (E)**, merupakan kesimpulan terhadap biaya yang diperlukan terhadap pembangunan pada sistem.

Pada teknik TOE ini terbagi kembali menjadi beberapa bagian, yaitu:

- High (H)**: Sulit dalam pengerjaannya, karena teknik pembuatan dan pemakaiannya sulit serta biayanya tinggi. Maka kebutuhan tersebut dapat dieliminasi.
- Middle (M)**: Mampu dikerjakan.
- Low (L)**: Mudah dikerjakan.

• 2.1.4. Metode Penelitian

Metode Penelitian merupakan jalan atau cara yang memiliki rencana dan arahan dalam memperoleh data dan fakta yang akan digunakan untuk kepentingan penelitian.

Dijelaskan oleh Sugiono (2012) “Metode penelitian pada hakikatnya merupakan suatu cara ilmiah dalam mendapatkan data dengan tujuan dan kegunaan tertentu. Secara umum terdapat tiga (3) jenis penelitian, yaitu:

- **Bersifat Penemuan**

Data yang diperoleh pada penelitian betul-betul baru yang sebelumnya belum pernah ada atau belum pernah diketahui.

- **Bersifat Pembuktian**

Data yang telah didapatkan digunakan dengan tujuan mendapatkan pembuktian terhadap keragu-raguan informasi.

- ***Bersifat Pengembangan***

Pengumpulan data pada penelitian memiliki tujuan dalam mempertajam dan memperluas pengetahuan sebelumnya.

Menurut Ety Rochaety (2007) *“Penelitian deskriptif merupakan suatu kegiatan penelitian untuk mengetahui nilai variable mandiri, baik satu variable (independent) atau lebih tanpa membuat perbandingan atau menghubungkannya dengan variable lain”*.

Prof. DR. Lexy J. Moleong (2018) menuliskan bahwa *“Metodologi kualitatif merupakan prosedur yang dapat menghasilkan suatu data deskriptif berupa kalimat tertulis atau lisan dari orang-orang dan perilaku yang dapat diamati”*.

McMillan & Schumacher (2012) dalam pendapatnya menyatakan bahwa *“Pendekatan kualitatif adalah pendekatan investigasi karena biasanya peneliti mengumpulkan data dengan cara bertatap muka langsung dan berinteraksi dengan orang-orang di tempat penelitian”*.

2.1.5. Full Stack Development

Menurut Simon Holmes (2016), *“Full stack is a technology that involves a database and web server on back end which contains application logic and control in the middle as well as a user interface on the front end. Back-end developers focus on the mechanics that run behind them, front-end developers focus on building user experience”* dalam bahasa Indonesia *“Full stack adalah teknologi yang melibatkan database dan web server pada back-*

end yang berisi logika pada aplikasi terhadap antar muka. Pengembang back-end akan menitikberatkan pada mekanisme yang berjalan pada transaksi data di belakang, pengembang front-end mementingkan dalam membangun pengalaman pengguna (*User Experience*)". Front-end menyediakan data dalam bentuk tampilan rancangan antar muka (*User Interface*) dan Back-end lebih mengarah pada business logic dan data pada web aplikasi. Development adalah pembangunan atau pengembangan, jadi Full Stack Development adalah pembangunan atau pengembangan yang menggunakan teknologi yang melibatkan database dan web server pada back-end pada mekanisme dibelakang layar dan front-end fornt-end membangun penyediaan UI serta

- membangun UX yang lebih baik. User Interface Design (*UI Design*) merupakan tahapan desain dalam menampilkan sebuah hasil rancangan antar muka dalam bentuk tampilan yang dapat dilihat oleh pengguna (*user*). *User Experience* (UX) atau pengalaman pengguna (*user*) merupakan metode evaluasi penggunaan pengguna terhadap perangkat lunak dengan cara mendapatkan informasi beberapa tanggapan atau persepsi dari penggunaan pada perangkat lunak sehingga masalah mengenai kegunaan, pengalaman pengguna yang berhubungan dengan desain antar muka *User Interface* (UI) dapat di nilai untuk menentukan suatu aplikasi sudah sesuai dengan kebutuhan yang dapat diterima oleh pengguna (*user*) atau belum.

2.1.5.1. Front-end

Front-end adalah segala yang dalam menampilkan suatu desain antarmuka atau *user interface* (UI Design) serta fitur yang ada pada suatu aplikasi. Front-and dalam membangun mobile application

menggunakan bahasa pemrograman standar *Hypertext Markup Language* (HTML), pengaturan elemen dengan bahasa markup *Cascading Style Sheets* (CSS) dan pengembangan tampilan secara dinamis dan interaktif dengan *JavaScript*. Pada beberapa tampilan halaman akan berinteraksi pada database dan menerapkan *Single Page Application* (SPA) artinya sistem handling suatu transaksi data pada satu halaman UI dengan *JavaScript* menggunakan bahasa *Asynchronous JavaScript* (AJAX) dan *Framework AngularJS* sehingga tampilan dan fitur akan dapat berjalan secara dinamis.

2.1.5.2. Back-end

Back-end merupakan bagian belakang layar yang menyediakan kebutuhan yang tak terlihat yaitu handling setiap proses ketika suatu aplikasi dapat beroperasi. Pada back-end dapat handling proses database yang dapat digunakan dalam membuat (*CREATE*), menampilkan (*READ*), mengubah (*UPDATE*) dan menghapus (*DELETE*). Dalam halaman website akan mampu berinteraksi dengan database dengan bantuan pemrograman JavaScript menggunakan bahasa *Asynchronous JavaScript* (AJAX) dan *Framework AngularJS* sehingga tampilan dan fitur dapat berjalan secara dinamis. Back-end mampu membangun *Database Management System* (DBMS) menggunakan perintah *My Structured Query Language* (MySQL) dengan penerapan *Representational State Transfer Application Programming Interface* (*RestAPI*) dengan metode transaksi data secara asinkron terhadap data yang terstruktur dalam tampilan *JavaScript Object Notation* (JSON)

yang mengurai database dengan *Hypertext Preprocessor (PHP Fetch and Parse)*. Pengembangan *Database Management System (DBMS)* dalam hal ini menggunakan perintah yang dikenal dengan *My Structured Query Language (MySQL)*. *JavaScript Object Notation* atau kepanjangan dari JSON merupakan format transaksi data yang sangat ringan, mudah dibaca serta dapat diterjemahkan dan digenerate oleh komputer yang terhubung dengan jaringan. JSON menggunakan bahasa standar dan umum yang digunakan oleh programmer pada bahasa keluarga C termasuk C, C++, C#, Java, JavaScript, PHP, Perl, Python tidak bergantung pada bahasa pemrograman apapun yang menjadikan JSON sangat baik digunakan sebagai bahasa pertukaran data.

2.1.6. Fitur

Dalam KBBI (Kamus Besar Bahasa Indonesia) menjelaskan bahwa Fitur merupakan karakteristik dari sesuatu. Secara umum Fitur merupakan karakteristik khusus terkait kemampuan dan fungsionalitas yang terdapat pada perangkat lunak atau perangkat keras.

2.1.7. Aplikasi Mobile (Mobile-Apps)

Aplikasi Mobile dalam bahasa yang dikenal *Mobile Application* atau disebut juga mobile-apps adalah merupakan aplikasi perangkat lunak (software) yang beroperasi secara mandiri dan independent terhadap perangkat keras lain atau standalone yang dalam pengoperasiannya memiliki kemampuan dapat dipasangkan dan berjalan pada perangkat mobile seperti telepon pintar (*smartphone*). Pada Mobile Application terdapat empat (4)

jenis mobile application yang dikenal secara luas, yaitu:

- *Hybrid Mobile Applications*

Tipe ini menggunakan *single code* tanpa native engine dengan teknologi standar web (HTML, CSS dan JavaScript.) yang dibungkus pada satu web container atau satu paket instalasi dan dapat diakses dari smartphone berbasis OS Android maupun iOS.

- *Native Mobile Applications*

Pada tipe ini menggunakan basis kode yang ditulis langsung pada bahasa pemrograman asli, misalkan akan membuat aplikasi untuk android maka menggunakan bahasa pemrograman Java dan jika untuk IOS ialah bahasa pemrograman dalam Objective-C dan C.

- *Cross-platform Native Mobile Applications*

Pada tipe ini sama dengan hybrid mobile applications menggunakan single code namun menggunakan native engine untuk melakukan rendering sehingga dapat digunakan pada platform android dan IOS atau lainnya. Pendekatan cross platform merupakan salah satu metode yang paling efisien dengan hambatan yang rendah dan support bahasa HTML, CSS, dan JavaScript yang sejalan dengan paradigma Web.

- *Progressive Web Applications*

Pada tipe merupakan gabungan dari web application dan mobile apps yang dapat mengakses native mobile experience pada web browser tanpa harus mengunduh dan menginstal aplikasi pada perangkat anda.

2.1.8. Aplikasi Mobile Android

Aplikasi Mobile Android adalah salah satu jenis sistem operasi yang didesain secara khusus dengan berbasis *Linux kernel* yang dapat beroperasi pada perangkat keras telepon pintar (*smartphone*) dan tablet. Rancangan atau keseluruhan sistem atau arsitektur pada android terdiri dari tumpukan komponen perangkat lunak (*software stack*). Komponen architecture android terbagi menjadi lima (5), yaitu:

1. Aplikasi (Application).

Dalam lapisan ini terdapat tiga (3) lapisan:

- Lapisan UI (*User Interface*) adalah tampilan data aplikasi di layar.
- Lapisan Domain ialah lapisan penghubung interaksi antara lapisan UI dan data.
- Lapisan Data merupakan tampilan yang berisi logika bisnis aplikasi dan mengekspos data aplikasi.

2. Kerangka Aplikasi (Framework Layer).

Pada komponen ini menyediakan class dan interface dalam bentuk Java class untuk pengembangan aplikasi android dan services terhadap aplikasi layanan lokasi, layanan telepon, manajer notifikasi, layanan NFC, sistem tampilan.

3. Penggerak Android (Android Runtime).

Komponen mesin yang mengoperasikan aplikasi bersama dengan libraries dan membentuk dasar terhadap framework aplikasi. Dalam Android Runtime tersedia komponen utama yaitu DVM (*Dalvik Virtual Machine*) atau Mesin Virtual Java khusus untuk Android.

4. *Perpustakaan Program (Platform Libraries).*

Komponen ini merupakan kumpulan atau cakupan kemampuan program bahasa dari bermacam-macam bahasa, antara lain: Java, C/C+, libc, Graphics, SQLite, Webkit, Media, Surface Manger, OpenGL, dll.

5. *Linux Kernel.*

Linux Kernel adalah komponen terpenting yang bertugas sebagai driver terhadap perangkat seperti jaringan network, GPS, camera, layar LCD, tombol atau keyboard, bluetooth dan mengelola memori sehingga semua dapat berjalan dengan baik dan benar.

● **2.1.9. Aplikasi Berbasis Web (Web-based Application)**

Aplikasi Berbasis Web dinamakan *Web-based* adalah aplikasi web atau program yang diakses dengan melalui aplikasi pada komputer melalui browser (aplikasi untuk membuka halaman internet) yang mana aplikasi web tersimpan dalam storage web server yang dapat diakses pada jaringan internet dalam bentuk sebuah situs website.

Dalam membangun atau pengembangan aplikasi berbasis web bahasa pemrograman yang dapat digunakan adalah bahasa markup (HTML), Cascading Style Sheet (CSS), JavaScript dan dapat menggunakan beberapa framework yang telah memiliki fitur dan interface yang menarik dan lengkap seperti Bootstrap, IONIC, ChartJS, Cordova, jQuery, PhoneGap, Framework7 dan ada banyak framework lainnya.

2.1.10. Web Development Framework

Web Development Framework merupakan bentuk kerangka kerja yang

dapat digunakan dalam memudahkan *programmer* atau *developer* mengembangkan website dalam penulisan baris kode agar penulisan baris kode menjadi terstruktur, rapih, mudah, dan cepat.

2.1.10.1. Framework Bootstrap

Framework Bootstrap merupakan aplikasi web yang telah tersusun dan terstruktur untuk memudahkan para programmer dalam mengaplikasikan aplikasi web yang telah memiliki fungsi-fungsi variable, class-class, komponen properti (navigation, grids, forms, buttons, datatables) hingga plugins yang telah menyimpan beragam pustaka CSS (Cascading Style Sheet) yang dapat disesuaikan dalam tampilan web yang responsive dan dinamis.

2.1.10.2. Framework IONIC

Framework Ionic adalah framework opensource lainnya seperti Bootstrap yang dapat digunakan secara gratis tetapi pada IONIC telah mengembangkan lebih luas lagi yaitu dapat menampilkan halaman web pada aplikasi mobile dan pemrogramannya menggunakan bahasa HTML5, CSS dan AngularJS.

2.1.10.3. Framework ChartJS

Framework ChartJS merupakan framework yang mampu menampilkan grafis chart pada halaman web yang hadir dengan pengetahuan pemrograman open-source yang dikembangkan oleh Microsoft atau dinamakan TypeScript dan mampu berinteraksi pada

bahasa pemrograman JavaScript termasuk React, Vue, Svelte dan Angular.

2.1.10.4. Framework Cordova

Framework Cordova yang dikeluarkan oleh Apache merupakan salah satu *platform* yang dapat digunakan untuk membangun native mobile-apps dalam bahasa program HTML, CSS dan JavaScript yang mendukung OS Android, iOS, Windows Phone dan Blackberry. Framework Cordova ini terdiri dari sekumpulan *Application Programming Interface* (API) dalam mengakses device dari perangkat mobile seperti kamera, GPS (*Global Positioning System*), storage dan lain-lain. Dengan menggunakan UI (*User Interface*) framework seperti jQuery Mobile, maka dapat mengakses API ini atau dapat dikatakan Cordova dapat membangun aplikasi dengan menggunakan HTML, CSS dan Javascript sehingga mobile-apps dapat dibangun tanpa menggunakan native code, Java, Objective-C dan lainnya. Aplikasi yang dihasilkan dari Cordova akan dikemas dalam aplikasi yang menggunakan *Software Development Kit* (SDK) masing-masing platform.

2.1.10.5. Framework jQuery

jQuery merupakan kumpulan kode yang dapat dipergunakan dalam mempercepat akses, mempermudah akses, memanipulasi dan memanggil elemen tertentu pada pustaka atau library JavaScript yang mampu memudahkan interaksi antara kode dalam HTML dan

JavaScript sehingga dapat beroperasi pada sisi klien.. Framework ini pertama kali diperkenalkan pada tahun 2006 oleh Jhon Resig.

2.1.11. Dashboard

Menurut Few (2006) pengertian Dashboard ialah tampilan atau gambaran terhadap informasi-informasi penting yang dibutuhkan dalam menghasilkan tujuan yang tergabung dan tersusun pada satu layar tunggal sehingga visualisasi tersebut dapat dipantau secara sekilas. Dashboard biasanya menampilkan antar muka suatu halaman dengan berbagai bentuk tampilan seperti chart, diagram, laporan, indikator, alert, yang dipadukan dengan informasi yang dinamis dan relevan (Gonzales, 2006).

Menurut Rasmussen, et al (2009), jenis-jenis Dashboard dapat dikenali pada tiga (3) jenis, yaitu:

1. *Strategic Dashboard (Dashboard Strategis)*

Dashboard yang memberikan auto informasi dalam memudahkan pada pengambilan keputusan bisnis, gambaran dalam pencapaian bisnis serta prediksi peluang bisnis yang mendukung operasional pada tingkatlevel manajemen strategis.

2. *Tactical Dashboard (Dashboard Taktis)*

Dashboard tipe taktis yang memiliki perhatian pada proses analisis dalam menentukan penyebab terhadap suatu kejadian atau kondisi tertentu.

3. *Operational Dashboard (Dashboard Operasional)*

Dashboard ini menggambarkan status proses bisnis yang berguna dalam memonitor proses bisnis, aktivitas bisnis, dan hal yang kompleks

yang biasanya memberikan update harian/mingguan atau grafik real time.

2.1.12. HTML5

HTML adalah bahasa markup substandard pada web. Dalam mendukung pengembangan website, HTML (HyperText Markup Language) telah memperbaharui bahasa penulisan substandard pada web menjadi HTML5 atau HTML versi kelima (5) yang telah menambahkan tambahan fitur dan kemampuan baru dan kemudahan dalam penulisan kode.

• 2.1.13. CSS

CSS kepanjangan dari *Cascading Style Sheet* yang dikembangkan oleh W3C (*World Wide Web Consortium*). CSS merupakan bahasa penulisan style pada HTML dalam mengatur tampilan seperti pengaturan background, warna, ukuran, jarak antar baris, teks hingga desain image sehingga tampilan web menjadi menarik dan terstruktur.

2.1.14. PHP

PHP (*Hypertext Preprocessor*) dikenalkan pada tahun 1994 oleh pengembang perangkat lunak yang bernama Rasmus Lerdorf. Enterprise, Jubilee (2017) menjelaskan bahwa PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis web. PHP merupakan bahasa pemrograman server-side, sehingga dokumen atau script PHP akan diproses oleh server yang hasilnya akan dikirim kembali ke browser.

2.1.15. JavaScript

JavaScript merupakan bahasa pemrograman web yang bersifat client-side programming language (bahasa pemrograman pemrosesannya dilakukan oleh *client*). JavaScript dapat beroperasi pada hampir semua browser dan dapat digunakan pada aplikasi mobile. Penulisan JavaScript dapat disisipkan dalam kode script menggunakan tag mulai `<script>` dan tag akhir `</script>`.

2.1.16. JSON

JSON kepanjangan dari *Java Script Object Notation* adalah salah satu teknik pengelolaan data yang dapat digunakan dalam mengambil data dari database dan menampilkan data pada web dengan menggunakan bantuan penulisan kode pemrograman PHP dan JavaScript. Dalam penggunaan pengelolaan JSON dapat menggunakan PHP dan JavaScript untuk membuat komunikasi data yang cepat serta komunikasi data dapat beroperasi secara langsung (*real time*). Berikut ini adalah struktur data universal JSON:

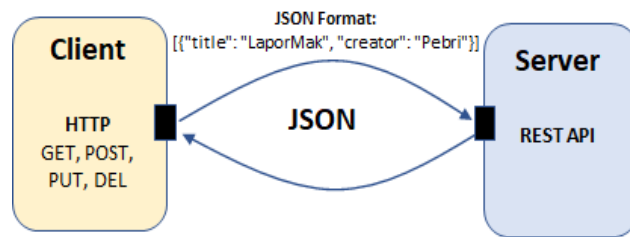
- *Set of Name/Value.*

Ini merupakan pernyataan terhadap *object* (objek), *record* (rekaman), *struct* (struktur), *dictionary* (kamus), *hash table*, *keyed list* dan *associative array*.

- *Ordered list of values.*

Ini pernyataan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Berikut ini adalah komunikasi JSON antara Client dan Server:



Gambar 2.1.1 - Komunikasi JSON

2.1.17. REST API

REST API adalah kependekan nama dari *Representational State Transfer Application Programming Interface* yang diperkenalkan pertama kali oleh Roy Fielding pada tahun 2000 dan menyatakan bahwa “REST dimaksudkan dalam suatu rancangan bagaimana aplikasi Web yang dirancang dengan baik beroperasi: dalam jaringan halaman web (a virtual state-machine), tempat pengguna melanjutkan melalui aplikasi dengan memilih tautan (state transitions), menghasilkan halaman berikutnya (mewakili status berikutnya dari aplikasi) yang ditransfer ke pengguna dan diberikan untuk mereka gunakan”.

Menurut Patni (2017) “REST atau RESTful adalah sebuah gaya arsitektur yang memiliki fungsi untuk membuat dan mengelola sebuah sistem distribusi data”.

Menurut Jin, Sahni, & Shevat (2018) “API merupakan sebuah antarmuka dalam sebuah sistem perangkat lunak yang disajikan kepada program lain, kepada manusia dan dalam API web, untuk dunia melalui internet”.

Menurut Pautasso (2008), terdapat 4 (empat) prinsip utama dalam REST, yaitu:

- *Resource identifier through Uniform Resource Identifier (URI).*

REST Web Services mengidentifikasi interaksi antar client dalam mencari sekumpulan sumber daya.

- *Uniform interface.*

Sumber daya yang di manipulasi yaitu CRUD (Create, Read, Update, Delete) dan menggunakan operasi PUT, GET, POST, dan DELETE.

- *Self-descriptive messages.*

Sumber daya terhadap informasi tidak terhubung, sehingga penggunaannya untuk mengakses berbagai format seperti (HTML, XML, PDF, JPEG, Plain text dan lainnya). Metadata pun dapat digunakan.

- *Stateful interactions through hyperlinks.*

Setiap interaksi pada suatu sumber daya bersifat statis (*stateless*), yaitu *request messages* tergantung pada jenis.

POX	RSS	JSON	...
HTTP GET	HTTP POST	HTTP PUT	HTTP DEL
<i>Resource URI</i>			
<i>Application</i>			

Gambar 2.1.2 - REST : the Web is the universe of globally accessible information (Pautasso, SOA Symposium)

2.1.18. Sistem Basis Data

Menurut Sukamto, R. A., & Shalahuddin (2018) “*Sistem basis data merupakan sistem terkomputerisasi yang tujuan utamanya untuk memelihara*

data yang telah diolah atau informasi tersedia saat dibutuhkan. Pada intinya basis data merupakan media yang diperuntukan pada penyimpanan data agar dapat diakses dengan mudah dan cepat". Bentuk spesifikasi basis secara umum berupa tabel yang berisi informasi field name, field type, field size, dan keterangan field yang menjadi field unik (unique key) field kunci (primary key).

2.1.19. Unified Modelling Language (UML)





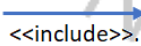
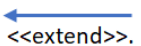
Menurut (Booch, 2005:7), *Unified Modelling Language (UML)* adalah bahasa standar dalam pemrograman yang digunakan pada tahapan perancangan suatu sistem. Pemodelan ini merupakan standarisasi yang digunakan dalam pembangunan perangkat lunak berorientasi object. UML hadir karena kebutuhan pemodelan visual dalam menggambarkan, menggambarkan spesifikasi dan membuat dokumentasi dengan menggunakan visualisasi diagram dan teks-teks pendukung. Diagram UML yang dikenal luas terdiri dari 13 diagram, yaitu: *Use-case diagram, Activity diagram, Sequence diagram, Class diagram, State-machine diagram, Communication diagram, Deployment diagram, Component diagram, Object diagram, Composite structure diagram, Interaction overview diagram, Package diagram dan Diagram timing*. Dalam praktiknya dalam suatu pengembangan aplikasi menggunakan 3 sampai 5 diagram. Berikut ini tiga (3) diagram UML yang sering digunakan dalam pengembangan aplikasi.

2.1.19.1. Use-case Diagram

Use-case diagram merupakan visualisasi interaksi antar aktor

dari sebuah proses atau kegiatan tertentu, Seorang aktor yang dimaksud merupakan sebuah *entity* (entitas) yaitu pengguna atau mesin yang berinteraksi dengan sistem dalam melakukan pekerjaan-pekerjaan tertentu. Notasi-notasi yang digunakan dalam menggambarkan use-case diagram adalah sebagai berikut:

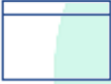





Tabel 2.1.1 - Notasi dan Simbol Use-case Diagram

SIMBOL	KETERANGAN
System Boundry 	Menyatakan batasan sistem dalam relasi dengan <i>aktor-aktor</i> yang menggunakannya (di luar sistem) dan fitur-fitur yang harus disediakan (dalam sistem). Digambarkan dengan segi empat yang membatasi semua use case dalam sistem terhadap pihak mana sistem akan beraksi.
Aktor 	<i>Aktor</i> adalah segala sesuatu yang perlu berinteraksi dengan sistem untuk pertukaran informasi. Bisa merupakan manusia, sistem atau <i>device</i> .
Use-case 	Mengidentifikasi fitur kunci dari sistem. Tanpa fitur ini, sistem tidak akan memenuhi permintaan <i>user</i> /aktor. Setiap <i>use case</i> mengekspresikan goal dari sistem yang harus dicapai. Diberi nama sesuai dengan goalnya dan digambarkan dengan elips dan nama di dalamnya.
Assosiation 	Mengidentifikasi interaksi antara setiap <i>aktor</i> tertentu dengan setiap <i>use case</i> tertentu. Digambarkan sebagai garis antara <i>aktor</i> terhadap <i>use case</i> yang bersangkutan.
Include 	Mengidentifikasi kelakuan yang harus terpenuhi agar sebuah <i>event</i> dapat terjadi, kondisi ini adalah hubungan dua <i>use case</i> dimana yang satu memanggil yang lain.
Extend 	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber secara <i>eksplisit</i> .

2.1.19.2. Activity Diagram

Activity Diagram atau diagram aktivitas merupakan jenis diagram *Unified modelling Language* (UML) untuk menggambarkan proses suatu sistem Langkah demi Langkah dari mulai *input* (masukan) hingga *output* (keluaran).

Tabel 2.1.2 - Notasi dan Simbol Activity Diagram






SIMBOL	KETERANGAN
Swimlane 	Menunjukkan siapa yang bertanggung jawab melakukan aktivitas dalam suatu diagram.
Start State 	Menunjukkan dimana aliran kerja itu dimulai.
End State 	Menunjukkan dimana aliran berakhir.
Action State 	Action state adalah langkah-langkah dalam sebuah activity. Action bisa terjadi saat memasuki activity, meninggalkan activity, atau pada event yang spesifik.
Decision 	Menunjukkan dimana sebuah keputusan perlu di buat dalam aliran kerja.
Synchronization 	Synchronization menunjukkan dua atau lebih langkah dalam aliran kerja berjalan secara serentak.

2.1.19.3. Sequence Diagram

Sequence Diagram atau diagram urutan merupakan diagram

dalam menjelaskan dan menggambarkan interaksi antar objek-objek pada sistem secara terperinci pada sekumpulan aktivitas dalam menggambarkan tahapan alur kerja terhadap suatu sistem dari mulai masukan hingga keluaran. Dalam sequence diagram memiliki beberapa komponen, yaitu:

Tabel 2.1.3 - Komponen Sequence Diagram


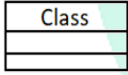
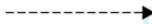

SIMBOL	KETERANGAN
Object 	Komponen ini menggambarkan sebuah object atau class. Biasanya objek berfungsi untuk mendokumentasikan perilaku sebuah objek pada sebuah sistem.
Activation Box 	Komponen ini merepresentasikan waktu yang dibutuhkan suatu objek untuk menyelesaikan tugasnya.
Actor 	Komponen ini menggambarkan user atau pengguna yang berinteraksi dalam sistem. Dalam sequence diagram, aktor biasanya digambarkan dengan simbol stick figure.
Lifeline 	Garis hidup sebuah object. Menggambarkan aktifitas dari objek.
Message 	Komponen ini menggambarkan pesan atau interaksi antar object.

2.1.19.4. Class Diagram

Class Diagram adalah diagram yang menggambarkan struktur sistem atau menggambarkan struktur kelas-kelas dari sistem. Pada

Class diagram memperlihatkan kumpulan dari kelas-kelas, collaboration, interface dan relasi dalam sistem. Gambaran dari Class diagram digambarkan dalam bentuk simbol-simbol yang ada pada tabel dibawah ini:

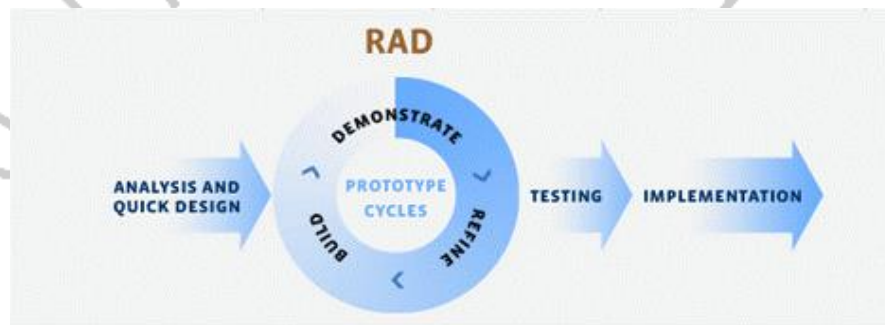
Tabel 2.1.4 - Simbol Class Diagram

SIMBOL	KETERANGAN
Generalization 	Menunjukkan hubungan objek anak (descendent) dan induk (ancestor) dalam hal berbagai perilaku dan struktur datanya.
Class 	himpunan dari objek-objek dalam sistem, yang kemudian berbagi atribut dan operasi yang persis sama.
Include 	Spesifikasi bahwa use-case sumber secara eksplisit.
Association 	Bagian yang menghubungkan objek yang satu dengan yang lainnya
Multiplicity 1, 1,*	Jumlah banyaknya Object sebuah Class berelasi dengan sebuah Object lain yang berasosiasi dengan Class tersebut.

2.1.20. Rapid Application Development (RAD)

Model RAD merupakan kependekan dari *Rapid Application Development* (RAD) adalah salah satu metode pengembangan dalam waktu

cepat namun mendapatkan hasil dengan kualitas yang lebih baik dibandingkan dengan hasil yang dicapai melalui siklus tradisional. Model ini mampu mempercepat dalam tahapan pengembangan yang dalam model ini menggabungkan beberapa teknik terstruktur dari teknik joint application dan teknik prototyping. Terdapat empat (4) tahapan dalam Pengembangan RAD yaitu:



Gambar 2.1.3 - Tahapan RAD

1. *Requirements Planning* (Perencanaan syarat-syarat).

Pada tahapan awal ini Fase ini pengembang akan menganalisa dalam mendapatkan kebutuhan sistem dan persyaratan sistem.

2. *RAD Design Workshop* (Workshop desain RAD).

Tahapan ini adalah penerapan dari persyaratan sistem yaitu membuat dan mengembangkan sebuah *prototype* berdasar pada persyaratan sistem yang telah ditentukan. tahapannya membuat perancangan dari front-end dan back-end dan UML.

3. *Testing* (Pengujian).

Tahapan ini adalah kegiatan evaluasi dalam memastikan sistem atau aplikasi telah sesuai dengan kebutuhan.

4. *Implementation* (implementasi).

Ini adalah tahapan terakhir dari metode ini yaitu saat pengembang melakukan penerapan langsung terhadap sistem sesuai dengan perancangan yang telah dilakukan, sistem akan masuk pada pengujian akhir.

2.1.21. User Acceptance Test (UAT)

User Acceptance Test (UAT) merupakan proses validasi pada tahap terakhir sebelum akhirnya aplikasi akan direlease. Testing tidak fokus pada kesalahan tampilan, kesalahan text atau pengujian sistem. Pengujian penerimaan pengguna dilakukan di lingkungan pengujian terpisah dengan penyiapan data kebutuhan (*requirement*) dan pengujian di akan di uji oleh dua atau lebih pengguna akhir yang akan terlibat.

2.2. Tinjauan Studi

Peneliti dalam mencari referensi atau penggalian informasi dalam penelitian melakukan kajian terdahulu yang telah dilakukan oleh beberapa peneliti sebelumnya menggunakan kajian deduktif yaitu mempelajari contoh yang dilakukan dan mengambil kesimpulan pada pembahasan yang telah dilakukan. Beberapa kajian yang akan digunakan sebagai referensi peneliti adalah sebagai berikut.

2.2.1. Rapid Application Development

Pada penerapan metode RAD terdapat penelitian sebelumnya yang

dilakukan oleh Nurman Hidayat dan Kusuma Hati pada tahun 2021 yang berjudul Penerapan Metode *Rapid Application Development* (RAD) dalam Rancang Bangun Sistem Informasi Rapor Online (SIRALINE) [7] yang di publikasikan dalam Jurnal Sistem Informasi STMIK Antar Bangsa menjelaskan bahwa penerapan metode RAD merupakan proses pengembangan *software* (perangkat lunak) secara waterfall atau *linear sequential* dalam pendekatan *System Development Life Cycle* (SDLC) yang unggul dalam kecepatan, ketepatan dan biaya yang lebih rendah.

2.2.2. Mobile Hybrid Application

Peneliti dalam tahapan pengembangan aplikasi mobile-apps dan dashboard berbasis web memilih teknik pembuatan aplikasi *cross platform* yang bersifat *hybrid* (dapat digunakan ke beberapa platform) dan teknik ini telah dituliskan pada penelitian terkait sebelumnya oleh penulis Eksen Hidihsah pada Journal Pengembangan IT (JPIT) tahun 2018 dengan judul *Mobile Hybrid Application Sebagai Solusi Dalam Pelaporan Bencana Menggunakan Framework Cordova* [1].

2.2.3. View board Berbasis Charts

Dalam pengembangan dashboard application berbasis web, tujuan peneliti adalah menggambarkan informasi praktis dan ringkas yaitu dengan memvisualisasikan data menjadi graphic atau chart yang menarik dan mudah dimengerti. Pada studi terdahulu mengenai visualisasi tampilan informasi praktis menggunakan chart telah dilakukan oleh Untung Rahardja, Qurotul Aini, Fitri Faradilla di tahun 2018 dengan judul *Implementasi Viewboard*

Berbasis Interaktif Javascript Charts Pada Sistem Penilaian Perkuliahan yang telah dipublikasikan pada Jurnal Ilmiah Teknologi Informasi Asia (JITIA) [2] dan Deviana Rahmadhani Sofyan, Ni Made Ria Rolina, Hetti Hidayati dengan judul *Employee Dashboard Telkom University Based On Web* dalam *Journal e-Proceeding of Applied Science* pada tahun 2020 [3]. Pada perancangan dan implementasi dashboard penulisan coding menggunakan HTML, CSS dan PHP yang ditampilkan dalam bentuk halaman website dan hal ini terdahulu telah dituliskan dalam sebuah jurnal oleh Jennifer S.Woo, Peter Suslow , Russell Thorsen, Rosaline Ma, Sara Bakhtary, Morvarid Moayeri, Ashok Nambiar pada tahun 2019 pada *Journal of Pathology Informatics* volume 10 dengan judul *Development and Implementation of Real-Time Web-Based Dashboards in a Multisite Transfusion Service* [4].

2.2.4. Full Stack Development

Konsep perancangan *Full Stack Development* telah dilakukan oleh peneliti sebelumnya dalam pengembangan penyediaan fitur pada aplikasi kepada jemaat untuk melihat lokasi komunitas sel yang dituliskan dalam Jurnal Strategi pada tahun 2020 yang di teliti oleh William, Billy Susanto Panca, Robby Tan dengan judul *Full Stack Development in Purpose for Digitalized Data on Connect Group* [5].

2.2.5. Web Service

Dalam penelitian oleh Rifki Indra Perwira, Budi Santosa pada tahun 2017 pada Jurnal Telematika volume 14 yang berjudul *Implementasi Web*

Service Pada Integrasi Data Akademik Dengan Replica Pangkalan Data Dikti [6] yang telah mengimplementasikan integrasi data CBIS akademik dengan replika pangkalan DIKTI di UPN telah di gabungkan pada web service. Pada penelitiannya mempermudah pnggabungan database dalam mengintegrasikan atau synchronize database dari aplikasi feeder ke aplikasi pusat.

