

## BAB II

### LANDASAN TEORI

#### 1.1. Tinjauan Pustaka

Dalam penelitian ini, penulis melakukan analisis literatur dari penelitian-penelitian sebelumnya sebagai sumber dukungan untuk penelitian yang akan dilakukan oleh penulis. Daftar literatur yang telah diteliti sebelumnya dapat dilihat pada tabel 2.1.

**Tabel 2.1.** Daftar Literatur

No	Penulis	Tahun	Judul
1	Edy, Daniel Adhinugraha, Likodemus Priko	2022	Implementasi <i>Web Service</i> pada Aplikasi Pemesanan Makanan Berbasis <i>Android</i>
2	Rizky Ramadhan, Purwanto	2022	Implementasi <i>web Service rest api</i> untuk merancang Aplikasi pusat informasi masjid al muhajirin larangan Indah
3	Mochamad Hasbi Maula, Dewi Kusumaningsih	2022	Implementasi <i>web Service</i> pada aplikasi pesan pakaian Toko snack collection
4	Herfandi, M. Julkarnain, Muhammad Hanif	2022	Desain dan implementasi <i>restful web Service</i> untuk integrasi data dan aplikasi
5	Baharuddin, Hamka Wakkang, Bambang Irianto	2022	Implementasi <i>web Service</i> dengan metode <i>rest api</i> untuk Integrasi data covid 19 di sulawesi selatan

### **1.1.1. Tinjauan Literatur 1**

Pada tahun 2022, Edy, Daniel Adhinugraha, dan Likodemus Priko telah melakukan penelitian yang berjudul "Implementasi *Web Service* pada Aplikasi Pemesanan Makanan Berbasis Android" yang dipublikasikan di Jurnal Algor. Penelitian tersebut membahas tentang masalah dalam mengelola data menu makanan pada sebuah perusahaan yang menyediakan konsumsi bagi para karyawannya. Saat ini, pengelolaan data menu makanan dilakukan melalui file yang disebarluaskan melalui email secara manual satu per satu. Namun, semakin banyaknya jumlah karyawan dan ragamnya menu yang tersedia membuat pengelolaan data semakin sulit dilakukan. Oleh karena itu, penelitian ini menawarkan solusi dengan menggunakan pengelolaan terpusat melalui smartphone yang memudahkan pengelolaan data menu makanan di mana saja. Namun, untuk melakukan komunikasi yang dinamis dalam aplikasi Android, diperlukan sebuah *Web Service* yang dapat berfungsi sebagai tempat untuk pertukaran data. *Web Service* juga harus dapat mendukung *interoperabilitas* antara mesin atau platform melalui jaringan. Dalam penelitian ini, *Web Service* digunakan untuk mengirimkan data menu makanan yang ditawarkan secara tepat waktu dan relevan dengan menu yang tersedia pada saat itu. Dengan adanya solusi ini, diharapkan dapat mempermudah pengelolaan data menu makanan pada perusahaan dan meningkatkan efisiensi pada pengelolaannya.

### **1.1.2. Tinjauan Literatur 2**

Pada tahun 2022, Rizky Ramadhan dan Purwanto dari SENAFTI Universitas Budi Luhur telah mengadakan sebuah penelitian yang berjudul

"Implementasi *Web Service* REST API untuk Merancang Aplikasi Pusat Informasi Masjid Al-Muhajirin Larangan Indah". Penelitian ini bertujuan untuk menyelesaikan permasalahan teknologi yang belum memadai di Masjid Al-Muhajirin Larangan Indah. Informasi seputar jadwal kajian, laporan keuangan, dan kegiatan-kegiatan lainnya masih disampaikan melalui papan pengumuman, spanduk, dan pengeras suara, serta pencatatan laporan keuangan masih dilakukan secara manual. Hal ini tentunya kurang efektif karena informasi yang disampaikan belum merata dan pencatatan manual rentan terhadap kehilangan data. Melalui implementasi *Web Service* dengan menggunakan metode REST API, penelitian ini memberikan solusi untuk mengatasi permasalahan tersebut dengan cara menyediakan akses informasi yang mudah dan terpadu untuk jamaah Masjid Al-Muhajirin Larangan Indah.

### **1.1.3. Tinjauan Literatur 3**

Pada tahun 2022, Mochamad Hasbi Maula dan Dewi Kusumaningsih dari SENAFTI Universitas Budi Luhur melakukan penelitian dengan judul "Implementasi *Web Service* pada Aplikasi Pemesanan Pakaian Toko Snack Collection". Penelitian ini dilakukan untuk mengatasi kesulitan yang dihadapi oleh toko tersebut dalam memasarkan produknya karena pandemi Covid-19 yang masih berdampak pada perilaku konsumen. Dalam penelitian ini, para peneliti merancang sebuah aplikasi toko online yang menggunakan teknologi *web Service*. Implementasi *web Service* bertujuan untuk memungkinkan perangkat yang berbeda platform dapat saling berkomunikasi dengan mudah. Aplikasi toko online yang dirancang dalam penelitian tersebut dapat diakses melalui website dan aplikasi Android. Dengan adanya aplikasi toko online yang menggunakan *web Service*,

diharapkan dapat memudahkan konsumen dalam melakukan pemesanan produk, serta meningkatkan efektivitas pemasaran dan penjualan produk untuk toko Snacks Collection.

#### **1.1.4. Tinjauan Literatur 4**

Pada tahun 2022, Herfandi, M. Julkarnain, dan Muhammad Hanif melaksanakan sebuah penelitian yang berjudul "Desain dan implementasi *restful web Service* untuk integrasi data dan aplikasi". Penelitian yang dipublikasikan di JINTEKS (Jurnal Informatika Teknologi dan Sains) ini membahas tentang masalah yang dihadapi oleh Geprek Chicken Dinner dalam terhambatnya melakukan manajemen bisnis karena menggunakan dua aplikasi yang berbeda dalam proses bisnisnya. Masalah tersebut mengakibatkan proses bisnis menjadi tidak optimal karena membutuhkan lebih banyak waktu dan tenaga. Dalam penelitian ini, para peneliti merancang aplikasi RESTful *web Service* dengan fitur access token dan API key sebagai sistem keamanan dengan metode pengembangan extreme programming. Aplikasi tersebut diharapkan dapat menjadi pangkalan untuk semua data yang diproses dan sebagai penghubung antar aplikasi yang digunakan oleh manajemen sehingga aplikasi-aplikasi yang digunakan dapat saling terintegrasi secara efektif. Dengan adanya aplikasi tersebut, manajemen Geprek Chicken Dinner diharapkan dapat mengatasi masalah integrasi data dan aplikasi dengan lebih baik sehingga proses bisnis dapat berjalan dengan lebih lancar dan efisien.

#### **1.1.5. Tinjauan Literatur 5**

Pada tahun 2022, Baharuddin, Hamka Wakkang, dan Bambang Irianto melakukan penelitian yang berjudul "Implementasi *Web Service* dengan Metode REST API untuk Integrasi Data COVID-19 di Sulawesi Selatan". Penelitian ini

dipublikasikan di Jurnal Sintaks Logika dan membahas masalah pembuatan sebuah *Web Service* yang dapat mengintegrasikan aplikasi web dan aplikasi mobile yang ada di Sistem Informasi Perkembangan COVID-19 di Sulawesi Selatan. Dalam penelitian ini, penulis berhasil menghasilkan sebuah website yang bersifat terpusat dengan memanfaatkan *Web Service* metode REST API dalam proses pengambilan data. Tujuan utama penelitian ini adalah untuk menggabungkan atau menyatukan data yang berasal dari database yang berbeda-beda menjadi sebuah sistem informasi yang terintegrasi. Dalam implementasinya, *Web Service* ini berperan sebagai sebuah jembatan komunikasi antara aplikasi web dan aplikasi mobile dalam mengambil data yang dibutuhkan. Dengan adanya solusi ini, diharapkan dapat mempermudah pengambilan data dan mempercepat proses integrasi data COVID-19 di Sulawesi Selatan.

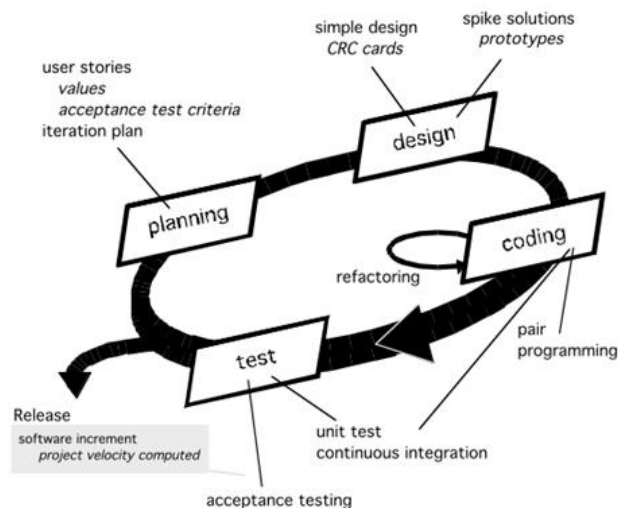
## **1.2. Metode Pengumpulan Data**

Metode pengumpulan data merupakan suatu hal yang penting dalam penelitian, karena metode ini merupakan strategi untuk mendapatkan data yang diperlukan. Keberhasilan penelitian sebagian besar tergantung pada teknik-teknik pengumpulan data yang digunakan. Untuk memperoleh bahan-bahan, keterangan, kenyataan- kenyataan dan informasi yang dapat dipercaya. Untuk memperoleh data seperti yang dimaksudkan itu, dalam penelitian digunakan teknik-teknik, prosedur-prosedur, alat- alat serta kegiatan yang nyata. Proses pengumpulan data dapat dilakukan melalui wawancara, observasi dan dokumentasi (Nurdiansyah dan Siti Rugoyah 2021).

### 1.3. *Extreme Programming (XP)*

*XP* merupakan metodologi yang digunakan untuk pengembangan perangkat lunak yang ditujukan dalam meningkatkan kualitas perangkat lunak terhadap perubahan serta kebutuhan pelanggan. Pada pengembangan ini jenis perangkat lunak dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pos pemeriksaan yang mana persyaratan pelanggan baru dapat diadopsi (Ariyanti dkk. 2020).

Ada beberapa tahapan yang ada pada *Extreme Programming* yaitu terdiri dari Perencanaan (*Planning*), *designing*, pengkodean dan testing. Dibawah ini adalah gambar kerangka kerja *XP* dapat dilihat pada Gambar 2.1.



**Gambar 2.1.** Kerangka Kerja *XP*

### 1.4. *Web Service*

*Web Service* adalah aplikasi sekumpulan data (database), perangkat lunak (software) atau bagian dari perangkat lunak yang dapat diakses secara remote oleh berbagai piranti dengan sebuah perantara tertentu. Secara umum, *web Service* dapat diidentifikasi dengan menggunakan URL. Penting untuk diketahui bahwa *web Service* harus bersifat stateless, yang berarti layanan ini tidak dapat menyimpan

informasi mengenai status sebelumnya, sehingga dapat dibaca dan diakses oleh berbagai platform. Dalam pembangunan arsitektur pada *web Service*, terdapat dua opsi yaitu menggunakan *Simple Object Access Protocol* (SOAP) atau *Representational State Transfer* (REST). Layanan tersebut juga dapat direpresentasikan dalam format teks seperti JSON atau XML.

Pemilihan antara SOAP dan REST harus dilakukan dengan mempertimbangkan kebutuhan dan keterbatasan sistem yang akan dibangun. Sehingga, pemilihan arsitektur pada *web Service* yang tepat dapat memastikan kinerja dan interoperabilitas yang maksimal (Rizal dan Rahmatulloh 2019). Dalam perkembangan teknologi saat ini, *web Service* menjadi salah satu teknologi yang banyak digunakan dalam pengembangan aplikasi. Oleh karena itu, pemahaman yang baik mengenai *web Service* sangat penting bagi para pengembang aplikasi dan teknologi informasi secara umum. Adapun perbedaan SOAP dan REST, dapat dilihat pada Tabel 2.2.

**Tabel 2.2** Perbedaan SOAP dan REST

Uraian	SOAP	REST
Protokol komunikasi	HTTP, HTTPS, SMTP, FTP	HTTP, HTTPS
Penggunaan bandwidth	Dalam jumlah request yang banyak, relatif boros bandwidth. Hal ini karena banyaknya markup dalam penulisan format XML	Relatif hemat bandwidth, karena markup-markup ekstra seperti pada XML tidak dipakai
Tren penggunaan	Banyak mulai beralih ke REST, meski masih tetap ada yang	Mulai populer, banyak dipakai oleh penyedia web servis terkemuka,

	mempertahankan, misalnya untuk integrasi aplikasi ke sistem legasi pada sebuah perusahaan.	seperti twitter, yahoo!, Flickr, loglines, technorati, google, amazon, eBay, dsb
Aturan penulisan	Ketat, mengikuti spesifikasi XML (SOAP v1.2)	Tidak ada spesifikasi khusus
Format respon	XML dengan spesifikasi SOAP. Agak sulit bagi kita untuk membaca langsung dan memahaminya.	XML, JSON, atau format plain teks lainnya. Hal ini memudahkan penerima respon membaca dan memahaminya.
Attachment file	Bisa (karena dapat mengembalikan respon dalam format binary)	Tidak bisa
Sifat <i>web Service</i> pada umumnya	Tertutup, lebih ditujukan untuk vendor atau perusahaan tertentu	Terbuka, bisa diakses siapa saja
Caching web	Relatif sulit	Mudah, karena menggunakan URI
Penggunaan standar	Standar lama (XML, HTTP) dan baru (SOAP) digunakan bersamaan	Standar yang sudah ada, seperti XML dan HTTP
Tool pengembangan	Banyak, baik komersial maupun opensource	Beberapa, karena tidak begitu dibutuhkan
Tool manajemen	Perlu, bahkan kadang harganya mahal	Menggunakan tool yang sudah ada pada sistem jaringan
Ekstensibel	Bisa, banyak ekstensi termasuk standar WS-*	Relatif tidak ekstensibel



Kemudahan implementasi	Mudah jika kita sudah memiliki lingkungan berbasis SOAP	Mudah
------------------------	---	-------

### 1.5. Aplikasi

Aplikasi merupakan salah satu jenis perangkat lunak (software) yang memiliki tugas penting sebagai antarmuka pengguna (front-end) dalam sistem komputer. Aplikasi memungkinkan pengguna untuk mengolah berbagai jenis data dan menghasilkan informasi yang bermanfaat sesuai dengan kebutuhan mereka. Selain itu, aplikasi juga dapat terintegrasi dengan sistem lainnya yang berkaitan.

Seperti yang dikemukakan oleh (Soraya dan Wahyudi 2021), aplikasi berperan penting dalam mengoptimalkan penggunaan sistem komputer. Dalam konteks bisnis atau organisasi, aplikasi dapat digunakan untuk meningkatkan efisiensi operasional, mengoptimalkan pemanfaatan sumber daya, dan menghasilkan informasi yang akurat dan relevan bagi pengambilan keputusan.

Aplikasi juga memiliki peran yang krusial dalam memenuhi kebutuhan pengguna, terutama dalam era digital yang semakin berkembang pesat. Pengguna saat ini membutuhkan akses cepat dan mudah terhadap informasi yang dibutuhkan, dan aplikasi merupakan solusi yang efektif untuk memenuhi kebutuhan tersebut.

Dalam rangka memaksimalkan manfaat dari aplikasi, pengembang perangkat lunak perlu memperhatikan aspek-aspek seperti keamanan, performa, dan user experience (pengalaman pengguna). Dengan memperhatikan hal-hal tersebut, pengguna akan merasa nyaman dan terbantu dalam mengolah data dan informasi yang dibutuhkan melalui aplikasi yang mereka gunakan.

## **1.6. Analisis SWOT**

Analisis SWOT adalah kerangka kerja yang digunakan untuk menganalisis kekuatan (*strengths*), kelemahan (*weaknesses*), peluang (*opportunities*), dan ancaman (*threats*) yang terkait dengan suatu bisnis atau proyek. Analisis ini dapat diterapkan dalam pengembangan perangkat lunak untuk membantu mengidentifikasi faktor internal dan eksternal yang dapat mempengaruhi keberhasilan pengembangan perangkat lunak.

Analisis SWOT dalam pengembangan perangkat lunak dapat membantu pengembang untuk memperkuat kekuatan mereka, mengatasi kelemahan mereka, mengeksplorasi peluang yang ada, dan mengatasi ancaman yang mungkin timbul. Dalam pengembangan perangkat lunak yang sukses, penting untuk mempertimbangkan faktor internal dan eksternal yang dapat mempengaruhi keberhasilan proyek dan untuk mengembangkan strategi untuk mengatasi faktor-faktor tersebut.

## **1.7. Unified Modeling Language (UML)**

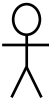


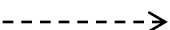
*UML* adalah bahasa spesifikasi standar yang digunakan untuk membangun, mendokumentasikan, dan menspesifikasi perangkat lunak. UML adalah metodologi yang digunakan dalam mengembangkan sistem berorientasi objek dan alat yang dapat mendukung pengembangan sistem. UML berfungsi sebagai bahasa yang berbasis grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan sistem pengembangan software berbasis OO (*Object-Oriented*). Dalam UML, terdapat standar penulisan sistem blue print yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang dibutuhkan dalam sistem



software (Suendri 2018). Terdapat beberapa jenis diagram dalam UML seperti diagram use case, diagram *Class* dan diagram activity yang membantu dalam memodelkan dan menggambarkan interaksi dan komponen-komponen dalam sebuah sistem. Adapun penjelasan dari jenis-jenis diagram tersebut sebagai berikut:

### 1.7.1. Use case diagram

*Use case diagram* merupakan pemodelan untuk menggambarkan perilaku (behavior) sistem yang akan dibuat. Diagram use case mendeskripsikan sebuah interaksi satu atau lebih aktor dengan sistem. Diagram use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Sari dan Alita 2022). Adapun simbol-simbol yang terdapat pada *Use case diagram* terdapat pada tabel 2.3 dibawah ini.

**Tabel 2.3** Tabel *Use case diagram*



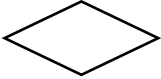
Nama	Simbol	Keterangan
<i>Aktor</i>		Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
<i>Use Case</i>		Abstraksi dan interaksi antara sistem dan aktor
<i>Association</i>		Abstraksi dari penghubung antara aktor dengan use case
<i>Generalisasi</i>		Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case



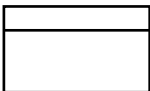
<i>Include</i>	<pre>&lt;&lt;include&gt;&gt;</pre> 	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
<i>Extend</i>	<pre>&lt;&lt; extend &gt;&gt;</pre> 	Menunjukkan bahwa suatu use case merupakan fungsional dari use case lainnya jika suatu kondisi terpenuhi

### 1.7.2. Activity diagram

*Activity Diagram* menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi, bagaimana masing-masing aksi tersebut dimulai, keputusan yang mungkin terjadi hingga berakhirnya aksi. *Activity diagram* juga dapat menggambarkan proses lebih dari satu aksi dalam waktu bersamaan (Suendri 2018). Adapun simbol-simbol yang terdapat pada *Activity diagram* terdapat pada tabel 2.4 dibawah ini.

**Tabel 2.4** Tabel *Activity diagram*


Nama	Simbol	Keterangan
Status awal		Sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan		Percabangan dimana ada pilihan aktivitas yang lebih dari satu




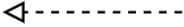
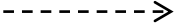

Penggabungan		Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi sat
Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
<i>Swinlane</i>		<i>Swinlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

### 1.7.3. *Class diagram*

*Class diagram* merupakan model yang menggambarkan struktur dan deskripsi *Class* serta dapat menghubungkan antara *Class* yang lain. *Class diagram* menjelaskan model yang digunakan dalam perancangan atribut dan fungsi-fungsi yang akan digunakan untuk membangun sistem baru (Ramadhanu dan Priandika 2021). Adapun simbol-simbol yang terdapat pada *Class diagram* terdapat pada tabel 2.5 dibawah ini.

**Tabel 2.5** *Tabel Class diagram*

Nama	Simbol	Keterangan
<i>Generalication</i>		Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> )

<i>Navy Association</i>		Upaya untuk menghindari asosiasi dengan lebih dari 2 objek
<i>Class</i>		Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama
<i>Collaboration</i>		Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
<i>Realication</i>		Operasi yang benar-benar dilakukan oleh suatu objek
<i>Depedency</i>		Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
<i>Association</i>		Apa yang menghubungkan antara objek satu dengan objek lainnya

#### 1.7.4. CRC Cards

*Class Responsibility Collaborator Cards* atau *CRC Cards* berguna untuk mendeskripsikan kelas apa saja yang akan dibuat beserta fungsionalitas yang dibutuhkan (Armanda dan Putra 2020). CRC memiliki 3 bagian yaitu *Class*, *Responsibilities*, dan *Collaborator*. Mungkin kata-kata *Class* teman-teman sudah mengetahui apa itu *Class* sehingga tidak akan saya bahas, selanjutnya saya akan

membahas apa itu *Responsibilities* dan *Collaborator*. *Responsibilities* merupakan hal-hal yang dikerjakan dan yang diketahui oleh suatu *Class*. *Collaborator* merupakan *Class* lain yang berinteraksi untuk dapat memenuhi *Responsibilities* suatu *Class*. Untuk lebih jelasnya, berikut adalah *CRC Card* dari Sistem Manajemen Informasi Destinasi Wisata Bagi Turis. Untuk skema bagaimana bentuk dari *CRC Card* ini sendiri dapat dilihat pada tabel 2.6 di bawah ini.

**Tabel 2.6** *Tabel Skema Bentuk CRC Card*

<i>Class Name</i>	
<i>Responsibilities</i>	Collabolator

### **1.8. Java**

Java adalah bahasa pemrograman dan platform komputasi pertama kali di rilis oleh Sun Microsystems pada tahun 1995. Pengguna aplikasi biasanya menggunakan *Java Runtime Environment (JRE)* diinstal pada mesin mereka sendiri untuk menjalankan aplikasi Java, atau dalam browser web untuk applet Java. Untuk pembuatan dan pengembangan aplikasi berbasis Java diperlukan *Java Development Kit (JDK)*, dimana saat ini pemilik lisensi dari JDK adalah Oracle Corporation yang telah secara resmi mengakuisisi Sun Microsystem pada awal tahun 2010 (Muharam dan Fauziah 2020).

### **1.9. Apache Maven**

Apache Maven adalah manajemen proyek perangkat lunak dan alat pemahaman. Berdasarkan konsep *project object model (POM)*, Maven dapat mengelola pembangunan, pelaporan, dan dokumentasi proyek dari informasi utama (Muharam dan Fauziah 2020).

### **1.10. Spring Boot**

Spring Boot adalah kerangka kerja (*framework*) yang memudahkan pengembangan aplikasi web berbasis Java. Spring Boot memungkinkan pengembang untuk membuat aplikasi web yang dapat dijalankan secara mandiri (*standalone*) dengan konfigurasi yang minimal. Dengan menggunakan Spring Boot, pengembang dapat memfokuskan pada logika bisnis daripada konfigurasi atau konvensi yang rumit. Spring Boot adalah kerangka kerja (*framework*) yang memudahkan pengembangan aplikasi web berbasis Java. Spring Boot memungkinkan pengembang untuk membuat aplikasi web yang dapat dijalankan secara mandiri (*standalone*) dengan konfigurasi yang minimal. Dengan menggunakan Spring Boot, pengembang dapat memfokuskan pada logika bisnis daripada konfigurasi atau konvensi yang rumit (Fauzi dkk. 2021).

### **1.11. Representational State Transfer (REST)**

REST (*Representational State Transfer*) adalah suatu arsitektur komunikasi berbasis web yang digunakan dalam pengembangan layanan berbasis web. REST umumnya menggunakan protokol HTTP (*Hypertext Transfer Protocol*) sebagai protokol untuk komunikasi data. Roy Fielding memperkenalkan REST pertama kali pada tahun 2000. Format umum yang digunakan dalam REST adalah JSON dan XML (Muharam dan Fauziah 2020).

*Restful web Service* merupakan salah satu kriteria desain *web Service* yang paling sering digunakan. Restful bekerja dengan cara *resource-oriented*, di mana client (*requester*) mengakses layanan yang ditawarkan oleh server dengan mengakses URL dari resource menggunakan method pada HTTP. Dalam dunia web API, protokol yang digunakan adalah HTTP. API *client* dapat berinteraksi dengan



API dengan mengirimkan berbagai jenis pesan HTTP yang memiliki standar pesan yang ditentukan. Standar-standar pesan tersebut yaitu:

1. *GET (Method Get* mengambil data dari web server dengan menentukan parameter di bagian URL dari permintaan.
2. *DELETE (Method Delete* menghapus sumber daya)
3. *POST (Method Post* memanfaatkan badan pesan untuk mengirim data ke server web)
4. *PUT (Method Put* mirip dengan post memanfaatkan badan pesan untuk mentransfer data)

*HTTP response code* merupakan kode yang dikirim ketika client mengirimkan request yang berisi informasi tentang respons server terhadap request yang dikirimkan. Beberapa contoh kode *respons status* HTTP dapat dilihat pada tabel 2.7.

**Tabel 2.7** Tabel Skema Bentuk *CRC Card*

200 ( <i>OK</i> )	berhasil melakukan tindakan yang di <i>request client</i> .
400 ( <i>Bad Request</i> )	Tidak bisa memahami tindakan yang di <i>request client</i> .
404 ( <i>Not Found</i> )	Tidak bisa memetakan URI <i>client</i> ke <i>resource</i> .

### 1.12. *Java Script Object Notation (JSON)*

*Java Script Object Notation (JSON)* adalah sebuah format data yang memungkinkan aplikasi untuk saling berkomunikasi di dalam sebuah jaringan, yang melalui RESTful API. JSON menyediakan semua bahasa pemrograman

modern dengan bantuan untuk menghasilkan dan menerima data (Muharam dan Fauziah 2020).

### **1.13. Database**

Database adalah sebuah system yang di buat untuk mengorganisasi, menyimpan dan menarik data dengan mudah. Database terdiri dari kumpulan data yang terorganisir untuk 1 atau lebih penggunaan, dalam bentuk digital. Database digital di manage menggunakan Database *Management System* (DBMS), yang menyimpan isi database, mengizinkan pembuatan dan maintenance data dan pencarian dan akses yang lain (Ramadhan dan Mukhaiyar 2020).

### **1.14. *My Structured Query Language (MySQL)***

MySQL adalah salah satu jenis database server yang sangat terkenal. MySQL menggunakan bahasa SQL untuk mengakses database nya. Lisensi Mysql adalah FOSS License Exception dan ada juga yang versi komersial nya. Tag Mysql adalah “*The World's most popular open-source database*”. MySQL tersedia untuk beberapa platform, di antara nya adalah untuk versi windows dan versi linux. Untuk melakukan administrasi secara lebih mudah terhadap Mysql, anda dapat menggunakan software seperti phpmyadmin dan mysql yog (Ramadhan dan Mukhaiyar 2020).

### **1.15. JSON Web Token (JWT)**

JWT adalah sebuah token yang digunakan untuk otentikasi dan pertukaran informasi. Token ini berbentuk string JSON yang sangat padat sehingga dapat dikirim melalui URL, parameter HTTP POST atau dalam *Header* HTTP. Keuntungan dari penggunaan token JWT adalah ukurannya yang kecil sehingga dapat ditransmisikan dengan lebih cepat dan memiliki informasi mandiri yang tidak

perlu query ke basis data lebih dari satu kali. Proses login menggunakan session, tetapi API hanya menggunakan konsep JWT yang disebut "jot". Token JWT dapat di-sign secara digital menggunakan secret atau pasangan *public/private key*. Struktur JWT terdiri dari tiga bagian: header, payload, dan signature. Cara kerja JWT sama seperti password, ketika pengguna berhasil login maka server akan memberikan token yang disimpan di local storage atau cookies browser (Yoda Gustiegan 2022).

#### **1.16. *Insomnia***

*Insomnia* adalah aplikasi desktop lintas platform gratis yang digunakan untuk berinteraksi dengan dan merancang API berbasis HTTP (Satyra Fajar Kusumah dkk. 2019).

#### **1.17. *Black Box Testing***

*Black box testing* adalah metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja (Pradana Putra dkk. 2020).

#### **1.18. *Unit Testing***

*Unit testing* merupakan bagian dari *Test-Driven Development* yang sangat penting. Fungsional aplikasi akan terukur dan terjaga kualitasnya mulai dari bagian terkecil (unit). Sementara itu untuk menjaga validitas fungsi terhadap *requirement* dari penggunaannya (Rizkyana dkk. 2021).