

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terkait

Penelitian yang dilakukan oleh Indriawan Muhammad Akbar dan rekan-rekannya dalam pembuatan sistem ini mengadopsi kerangka kerja Laravel. Kerangka kerja Laravel dikenal dengan penggunaan arsitektur MVC. MVC memisahkan komponen aplikasi menjadi model, view, dan controller, memfasilitasi pengembang dalam memahami dan mengelola kode aplikasi dengan lebih efisien. Pendekatan pembuatan backend aplikasi dilakukan dengan menggunakan metode *waterfall*, yakni suatu pengembangan sistem yang telah diuji dan umumnya diterapkan dalam industri.[7]

Dalam penelitian yang dilakukan oleh Ahsan Mubariz dan timnya. Teknologi yang digunakan untuk mengatasi keterbatasan hubungan antar layanan client yang tidak sinkron adalah *application programming interfaces* (API). yang dimaksud API merupakan suatu interface yang memfasilitasi program supaya mengakses aplikasi atau layanan spesifik. Dengan API, pengembang dapat memanfaatkan fitur-fitur yang sudah tersedia dari aplikasi lain tanpa harus membuatnya sendiri dari awal.[8]

Dalam penelitian yang dilakukan oleh Hendrik, Bagian back end dijelaskan sebagai ruang logis yang menangani fungsionalitas dan operasi dari sebuah aplikasi perangkat lunak atau sistem informasi salah dapat diimplementasikan dengan menggunakan sistem backend berbasis *RESTful*,

dimana sebuah arsitektur yang memungkinkan pertukaran data dengan metode protokol HTTP. tidak semua framework PHP memiliki kemampuan untuk mendukung fitur itu sendiri. Framework Codeigniter menyajikan cara untuk mengembangkan REST Server melalui sebuah library yang disebut Rest Server yang secara menyeluruh mendukung implementasi RESTful server. Dalam penelitian ini, REST Server dibangun untuk melaksanakan fungsi dasar CRUD, dan hasil dari endpoint adalah respon dari server dalam format JSON. Penelitian ini berhasil menghasilkan RESTful API atau REST Server yang mampu menangani empat jenis permintaan HTTP, yakni GET, POST, PUT dan DELETE.[9]

Dalam penelitian yang dilakukan oleh Eli Nurhayati dan rekannya menunjukan bahwa penggunaan API dapat meningkatkan efisiensi kinerja pengembang dengan memungkinkan mereka untuk memanfaatkan fungsi yang tersedia tanpa perlu mengembangkannya kembali sejak awal. salah satu arsitektur back end yang terkenal adalah Representational State Transfer (REST). Dalam proses mengkomunikasikannya, REST menerapkan pola permintaan-response dengan menggunakan protokol HTTP.[10]

## **2.2 Landasan Teori**

Pada bagian landasan teori ini, akan dijelaskan mengenai teori-teori yang mendukung pembahasan dan cara kerja rangkaian ini. Beberapa teori pendukung yang digunakan antara lain: PHP, MySQL, Xampp, Visual Studio Code, Laravel, UML ( *Unified Modeling Language* ).

### **2.2.1 PHP**

PHP (PHP: Hypertext Preprocessor) merupakan bahasa pemrograman sering diaplikasikan dalam pengembangan situs web. Script PHP dijalankan di sisi server dan diinterpretasikan Web Server. Web Server adalah entitas, baik dalam bentuk hardware maupun software, yang memberikan fasilitas kepada pengguna melalui protokol komunikasi HTTP atau HTTPS, dengan menggunakan berbagai file yang tersedia di situs web. peran PHP adalah untuk melakukan pemrograman di sisi server, dimana skripnya dieksekusi oleh web server. Sedangkan untuk desain tampilan, umumnya menggunakan HTML,CSS,Java Script, serta elemen grafis lainnya.[11]

### **2.2.2 MySQL**

MySQL merupakan salah satu sistem manajemen basis data (DBMS) cukup terkenal dan berperan sebagai sistem manajemen basis data relasional (RDBMS). Selain itu, MySQL merupakan perangkat lunak bersifat terbuka yang sangat cepat, andal, dan praktis untuk digunakan. Server basis data MySQL mendukung dua arsitektur utama, yaitu client server dan sistem tertanam (embedded systems). Karena MySQL merupakan perangkat lunak sifatnya yang sumber terbuka dan banyak digunakan, sehingga sesuai mengilustrasikan proses replikasi basis data.[12]

### 2.2.3 Xampp

XAMPP adalah perangkat lunak gratis yang dapat dioperasikan pada berbagai sistem operasi. Perangkat lunak berperan sebagai server lokal (localhost) yang terdiri dari Apache HTTP Server, MySQL database, serta menafsirkan bahasa PHP dan Perl. Singkatan XAMPP diambil huruf pertama dalam X (mengacu pada berbagai sistem operasi), Apache, MySQL, PHP dan Perl.[13]

### 2.2.4 Visual Studio Code

*Visual Studio Code (VS Code)* merupakan suatu program pengedit teks dibuat oleh *microsoft* dan dapat dioperasikan di berbagai platform sistem operasi. Termasuk *Linux, Mac, dan Windows*. Teks editor ini memiliki fitur bawaan untuk bahasa pemrograman *JavaScript, Typescript, dan Node.js*. selain itu, dukungan untuk bahasa pemrograman lainnya dapat ditambahkan melalui plugin yang tersedia marketplace *Visual Studio Code*, termasuk bahasa seperti *C++, C#, Python, Go, Java*, dan sebagainya.[14]

### 2.2.5 Codeigniter

Codeigniter merupakan web framework yang menggunakan bahasa pemrograman PHP, yang tersedia secara gratis serta bersifat open source. Framework ini dirilis dibawah lisensi MIT dan kode sumbernya dapat diakses melalui platform Github, Codeigniter dikembangkan oleh *Taylor Otwell* dengan tujuan supaya

mempermudah pengembangan aplikasi web mengadopsi sistem rancangan *model view controller (MVC)*. [15]

### 2.2.6 UML(*Unified Modelling Language*)

Unified Modelling Language (UML) merupakan standar industri dapat diaplikasikan untuk pemetaan, perancangan dan dokumentasi sistem perangkat lunak. UML memberikan bahasa yang konsisten dan efektif bagi para pengembang untuk mengkomunikasikan konsep, desain, dan struktur dari suatu sistem perangkat lunak. [16]

Terdapat empat model UML yang paling sering digunakan untuk menggambarkan suatu desain sistem *Usecase* diagram, *Class* diagram, Behavioral State machine diagram, dan juga *Sequence* diagram. Teknik-teknik pemodelan *Unified Modeling Language* ini disebut juga dengan 4 teknik dasar. Dalam proyek berorientasi objek, keempat teknik UML ini sangat mendominasi penggunaannya.

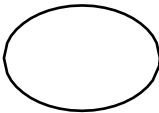
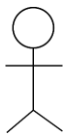

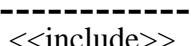

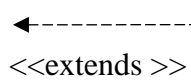
Dalam perancangan sistem terdapat *UML* yang sering digunakan sebagai berikut:

#### 1. *UseCase* Diagram

Dalam pembuatan sistem informasi, *use case* diagram adalah pemodelan untuk perilaku atau tingkah laku sistem informasi. Diagram ini juga bersifat statis.

Untuk simbol *UseCase* Diagram dapat di lihat pada Tabel 2.1.

Tabel 2. 1 *UseCase Diagram*



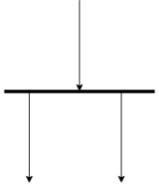

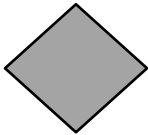
No	Simbol	Keterangan
1.		<i>Usecase</i> merupakan deskripsi dari fungsionalitas-fungsionalitas telah ditetapkan atau tersedia dalam suatu sistem.
2.		<i>Actor</i> merupakan himpunan peran untuk berinteraksi dengan <i>UseCase</i> .
3.		<i>Association</i> merupakan garis yang menghubungkan objek satu dengan objek lainnya.
4.		<i>Include</i> , merupakan gambaran jika <i>use case</i> dipanggil oleh <i>usecase</i> lain.
5.		<i>Dependency</i> merupakan garis panah yang menunjukkan jika aktor berinteraksi secara pasif.
6.		<i>Extend</i> , merupakan gambaran jika memperluas <i>usecase</i> target.

## 2. Activity Diagram

*Activity Diagram* didefinisikan penggambaran grafis dari aktivitas-aktivitas yang terjadi dalam suatu sistem. Diagram ini membantu dalam memvisualkan urutan langkah-langkah atau proses yang terjadi dalam suatu aktivitas atau aliran kerja.

Untuk simbol dari diagram aktivitas dapat dilihat di Tabel 2.2.

Tabel 2. 2 Activity Diagram


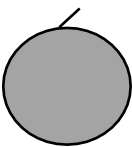
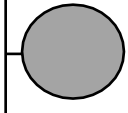

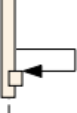

No	Simbol	Keterangan
1.		<i>End Point atau Final Node</i> , merupakan gambaran akhir dari suatu aktivitas
2.		<i>Start Point</i> , merupakan awal dari suatu aktivitas yang peletakkanya pada pojok kiri atas.
3.		<i>Fork</i> atau <i>Join</i> , digunakan untuk memarallelkan suatu kegiatan atau penggabungan 2 kegiatan paralel menjadi satu.
4.		<i>Activity</i> , merupakan gambaran dari suatu proses.
5.		<i>Decision</i> , merupakan pilihan pengambilan suatu keputusan <i>false or true</i> .

### 3. *Sequence Diagram*


*Sequence Diagram* atau Diagram Urutan mendeskripsikan diagram interaksi yang mengirimkan pesan dan diterima antar objek.

Untuk simbol diagram urutan dapat dilihat pada Tabel 2.3.

Tabel 2.3 *Sequence Diagram*

No	Simbol	Keterangan
1.		<i>Entity Class</i> , merupakan bentuk deskripsi awal sistem dan berisi kumpulan kelas dalam bentuk unsur yang mendasari untuk membuat database.
2.		<i>ControlClass</i> , merupakan gambaran penghubung antara Boundaryclass dengan suatu tabel
3.		<i>Boundary Class</i> , merupakan gambaran dari penggambaran form.
4.		Pesan atau <i>message</i> , menunjukkan pengiriman pesan antar <i>class</i> .
5.		<i>Self Message</i> , menunjukkan pengiriman suatu pesan yang akan dikirim ke objek itu sendiri.
6.		<i>Activation</i> , menggambarkan suatu objek yang melakukan sebuah aksi/eksekusi operasi.



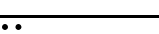
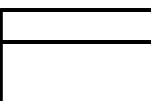
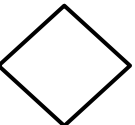


No	Simbol	Keterangan
7.		<i>Lifeline</i> , Garis titik terkoneksi keobjek di sepanjang garis <i>lifeline</i> memilikiaktivasi.



#### 4. *Class Diagram*

Diagram Class merupakan diagram yang memiliki karakteristik statis, diagram ini menampilkan kumpulan kelas, antarmuka, serta relasi.

untuk simbol dari *Class Diagram* dapat dilihat pada Tabel 2.4.

Tabel 2. 4 *Class Diagram*

No	Simbol	Keterangan
1.		<i>Generallization</i> , merupakan dimana Objek turunan ( <i>descendent</i> ) membagikan perilaku dan struktur data objek induknya.
2.		<i>Class</i> , adalah kumpulan objek yang saling berbagi.
3.		<i>Nary Assocation</i> , digunakan untuk asosiasi terhindar dengan objek lainnya.
4.		<i>Collaboration</i> , merupakan deskripsi urutan aksi yang ditampilkan suatu sistem yang memiliki konsekuensi terukur bagi aktor.
5.		Merupakan operasi yang <i>valid</i> dijalankan dari suatu objek.

No	Simbol	Keterangan
6.		Merupakan garis panah yang menunjukkan jika aktor berinteraksi secara pasif.
7.		<i>Association</i> merupakan garis yang menghubungkan objek satu dengan objek lainnya.