

## BAB II

### LANDASAN TEORI

#### 2.1. Tinjauan Pustaka

##### 2.1.1. Pengertian Framework Laravel

**Laravel** adalah sebuah framework PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep MVC (model view controller). Laravel adalah pengembangan website berbasis MVP yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu. MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen- komponen aplikasi, seperti : manipulasi data, controller, dan user interface.

1. Model, Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.
2. View, View adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web.
3. Controller, Controller merupakan bagian yang menjembatani model dan view.

Beberapa fitur yang terdapat di Laravel :

- Bundles, yaitu sebuah fitur dengan sistem pengemasan modular dan tersedia beragam di aplikasi.
- Eloquent ORM, merupakan penerapan PHP lanjutan menyediakan metode internal dari pola “active record” yang mengatasi masalah pada hubungan objek database.
- Application Logic, merupakan bagian dari aplikasi, menggunakan controller atau bagian Route.
- Reverse Routing, mendefinisikan relasi atau hubungan antara Link dan Route.
- Restful controllers, memisahkan logika dalam melayani HTTP GET and POST.
- Class Auto Loading, menyediakan loading otomatis untuk class PHP.
- View Composer, adalah kode unit logikal yang dapat dieksekusi ketika view sedang loading

Framework adalah kerangka kerja. Framework digunakan oleh developer untuk memudahkan pembangunan aplikasi web yang dapat berupa sekumpulan library yang berisi fungsi, tools, ataupun class-class, dan digunakan sebagai kerangka dalam pembangunan aplikasi web. Umumnya didalam framework tersebut telah menyediakan solusi untuk akses database, authentication, templating, controls, dan fungsi-fungsi lainnya. Penggunaan framework diharapkan membuat pengembangan aplikasi menjadi rapi dan bersih, memiliki struktur yang optimal, dan reusable. Struktur aplikasi yang lebih rapi dan teratur, biasanya menggunakan struktur MVC yang melakukan pemisahan antara business logic dengan presentation. Pemisahan dilakukan dengan tujuan agar setiap perubahan yang terjadi pada presentation logic atau business logic tidak memberikan pengaruh satu sama lainnya yang kompleks[3].

### **2.1.2. Pengertian Aplikasi**

”Aplikasi adalah sekelompok atribut yang terdiri dari beberapa form,report yang disusun sedemikian rupa sehingga dapat mengakses data” (Kusu Sutu, 2001:5) [4].Dari pengertian diatas, dapat disimpulkan bahwa aplikasi merupakan software yang ditransformasikan ke komputer yang berisikan perintah-perintah yang berfungsi untuk melakukan berbagai bentuk pekerjaan atautugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data.

### **2.1.3. Pengertian Aplikasi Berbasis Web**

Aplikasi web merupakan sebuah aplikasi yang menggunakan teknologi browser untuk menjalankan aplikasi dan di akses dengan jaringan komputer (Remik 2011). Sedangkan menurut (Rouse 2011) Aplikasi web adalah sebuah program yang di simpan di server dan dikirim melalui internet dan di akses melalui antarmuka browser. Dari pengertian di atas dapat disimpulkan aplikasi web merupakan aplikasi yang di akses menggunakan web browser melalui jaringan internet atau intranet. Aplikasi web juga merupakan suatu perangkat lunak komputer yang dikodekan dalam bahasa pemograman yang mendukung perangkat lunak berbasis web seperti HTML,JavaScript,CSS, Ruby, Python, Php, Java dan Bahasa lainnya[5].

### **2.1.4. Pengertian UML(Unifield Modeling Language)**

*Unifield Modelling Language* (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah system [6]. Jenis jenis diagram dalam UML adalah :

#### a. Use CaseDiagram

*Use Case Diagram* menggambarkan secara ringkas siapa yang menggunakan sistem dan apa saja yang bisa dilakukannya, Use Case Diagram tidak menjelaskan detail penggunaan usecase, Use Case Diagram hanya memberikan gambaran singkat hubungan antar usecase, sistem dan actor.

#### b. ClassDiagram

*Class Diagram* merupakan sebuah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sebuah sistem. Kelas mempunyai 3 bagian utama yakni *attribute*, *operation*, dan *name*. kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem

#### c. StateDiagram

StateDiagram adalah sebuah diagram yang menggambarkan daur hidup (*behaviorpattern*) dari suatu objek, dari objek tersebut diinisialisasi sampai didestroy, StateDiagram Menggambarkan Transisi dan perubahan keadaan dari suatu state ke state yang lainnya dalam suatu objek pada sistem.

#### d. Sequence Diagram

*Sequence Diagram* merupakan sebuah diagram yang menggambarkan interaksi antar objek dan mengindikasikan komunikasi antara objek-objek tersebut. *Sequence Diagram* juga menunjukkan serangkaian pesan yang dipertukarkan oleh objek-objek yang melakukan tugas atau interaksi tertentu. Objek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya diletakan pada bagian paling kiri dari diagram.

#### e. Collaboration Diagram

Diagram ini menggambarkan kolaborasi dinamis seperti *sequence diagram*. Dalam menunjukkan pertukaran pesan, *collaboration diagram* menggambarkan objek dan hubungannya, jika penekanannya pada waktu dan urutan, gunakan *sequence diagram* namun jika penekanannya pada konteks gunakanlah *collaboration diagram*.

#### f. Activity Diagram

*Activity diagram* adalah sebuah diagram yang menggambarkan aliran kerja (*workflow*) atau aktifitas dari sebuah sistem / proses bisnis. *Activity diagram* menggambarkan aktifitas yang dilakukan oleh sistem, bukan apa yang dilakukan aktor.

### **g. Component Diagram**

*Component diagram* menggambarkan suatu struktur dan hubungan antar piranti lunak, termasuk ketergantungan diantaranya. Komponen piranti lunak adalah modul berisi kode, baik sumber kode, ataupun binary code, baik executable maupun library, baik yang muncul pada compiletime, link time, atau run time.

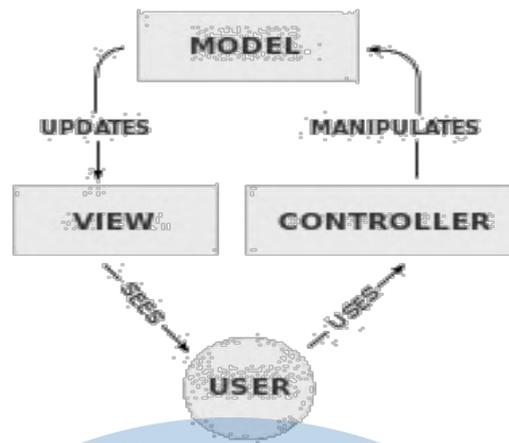
### **h. Deployment Diagram**

*Deployment diagram* menggambarkan detail bagaimana komponen *dideploy* dalam infrastruktur suatu sistem, dimana komponen akan terletak (pada mesin, server atau piranti keras), bagaimana kemampuan suatu jaringan pada lokasi tersebut, spesifikasi server dan hal lain yang bersifat fisik.

## **2.1.5 Pengertian MVC (Model-View-Controller)**

Model View Controller merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman Small Talk, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, user interface, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC pattern dalam suatu aplikasi yaitu [7] :

- Model, biasanya berhubungan langsung dengan database untuk memanipulasi data (insert, update, delete, search), menangani validasi dari bagian controller, namun tidak dapat berhubungan langsung dengan bagian view.
- View, merupakan bagian yang menangani presentation logic. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh controller. View berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.
- Controller, merupakan bagian yang mengatur hubungan antara bagian model dan bagian view, controller berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.



*Gambar 2.1. MVC Architectur*

Secara umum control flow arsitektur MVC melakukan hal seperti :

- Pengguna berinteraksi dengan user interface dengan beberapa cara (sebagai contoh, menekan tombol mouse).
- Controller menangani event input melalui user interface, biasanya via registered handler atau callback, kemudian mengkonversi event menjadi user action yang tepat dan dapat dimengerti oleh model.
- Controller memberitahukan kepada model mengenai user action yang memungkinkan terjadinya perubahan pada state model. (Sebagai contoh, controller meng-update shopping cart user.)
- Query model digunakan untuk men-generate user interface dengan tepat (contohnya, list konten yang ada pada shopping cart). View mendapatkan data melalui model, dalam beberapa implementasi, controller memberikan instruksi kepada view untuk me-render hasil query. Namun ada juga model yang secara otomatis memberitahukan view mengenai perubahan state yang membutuhkan update pada tampilan.
- User Inteface akan menunggu interaksi lebih lanjut dari pengguna, dimana akan memulai kembali siklus control flow.

### 2.1.6. Pengertian Black Box Testing

Black box testing, dilakukan tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. juga disebut sebagai behavioral testing, specification-based testing, input/output testing atau functional testing. Black box testing berfokus pada kebutuhan fungsional pada software, berdasarkan pada spesifikasi kebutuhan dari software. Black-box testing merupakan pendekatan pengujian yang ujinya diturunkan dari spesifikasi program atau komponen. Sistem merupakan 'kotak hitam' yang perilakunya hanya dapat ditentukan dengan mempelajari input dan output yang berkaitan. Nama lain untuk pengujian ini ialah pengujian fungsional karena penguji hanya berkepentingan dengan fungsionalitas dan bukan implementasi perangkat lunak (Sommerville 2001) [8].

### 2.1.7. UAT (User Acceptance Test) & Metode Pengujian Sistem

adalah suatu proses pengujian yang dilakukan oleh pengguna dengan hasil output sebuah dokumen hasil **uji** yang dapat dijadikan bukti bahwa software sudah diterima dan sudah memenuhi kebutuhan yang diminta. **UAT** tidak jauh beda dengan kusioner pada tahap awal pembuatan aplikasi.

### 2.1.8. Perangkat Lunak Pendukung Perancangan Aplikasi

Dalam pembuatan program aplikasi, untuk laporan tugas akhir ini Penulis menggunakan beberapa perangkat lunak yang menunjang pembuatan PSB Online ini, yaitu *Windows* sebagai sistem operasi yang akan dipakai, *CSS*, *Javascript*, dan *HTML* sebagai Bahasa pemrograman, *mysql* sebagai database dan *Laravel* sebagai pembangunan aplikasi.

#### 1. **CSS**

CSS Merupakan kumpulan kode-kode yang bertujuan untuk menghias dan mengatur gaya tampilan/layout halaman web supaya lebih elegan dan menarik. *CSS* adalah sebuah teknologi internet yang direkomendasikan oleh *World WideWeb Consortium* atau *W3C* pada tahun 1996. Awalnya, *CSS* dikembangkan di *SGML* pada tahun 1970, dan terus dikembangkan hingga saat ini. *CSS* telah mendukung banyak bahasa markup seperti *HTML*, *XHTML*, *XML*, *SVG* (Scalable Vector Graphics) dan *Mozilla XUL* (XML User Interface Language)[9].

#### 2. **Javascript**

JavaScript adalah bahasa pemrograman tingkat tinggi dan dinamis. JavaScript merupakan sebuah merk dagang yang dimiliki oleh perusahaan *Oracle*, akan tetapi

anda juga akan mendengar *Javascript* versi 1.5 atau versi 1.8. Versi tersebut merupakan versi yang dikeluarkan oleh *Mozilla Firefox*. *Javascript* versi 1.5 adalah *ECMAScript 3* dan *Javascript* versi 1.8 adalah *ECMAScript* yang sudah diperbaharui oleh *Mozilla Firefox*[10].

### 3. HTML

*HyperText Markup Language (HTML)* adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah Penjelajah web Internet dan formatting *hypertext* sederhana yang ditulis kedalam berkas format *ASCII* agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan kedalam format *ASCII* normal sehingga menjadi *home page* dengan perintah-perintah *HTML*. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan *SGML* (Standard Generalized Markup Language), *HTML* adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. *HTML* saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium (W3C)*. *HTML* dibuat oleh kolaborasi Caillau TIM dengan Berners-Lee Robert ketika mereka bekerja di *CERN* (lembaga penelitian fisika energi tinggi di Jenewa) pada tahun 1989.

Tahun 1980, *IBM* memikirkan pembuatan suatu dokumen yang akan mengenali setiap elemen dari dokumen dengan suatu tanda tertentu. *IBM* kemudian mengembangkan suatu jenis bahasa yang menggabungkan teks dengan perintah-perintah pemformatan dokumen. Bahasa ini dinamakan Markup Language, sebuah bahasa yang menggunakan tanda-tanda sebagai basisnya. *IBM* menamakan sistemnya ini sebagai *Generalized Markup Language* atau *GML*.

Tahun 1986, *ISO* menyatakan bahwa *IBM* memiliki suatu konsep tentang dokumen yang sangat baik, dan kemudian mengeluarkan suatu publikasi (*ISO 8879*) yang menyatakan markup language sebagai standar untuk pembuatan dokumen-dokumen. *ISO* membuat bahasa ini dari *GML* milik *IBM*, tetapi memberinya nama lain, yaitu *SGML* (Standard Generalized Markup Language).

*ISO* dalam publikasinya meyakini bahwa *SGML* akan sangat berguna untuk pemrosesan informasi teks dan sistem-sistem perkantoran. Tetapi diluar perkiraan *ISO*, *SGML* dan terutama subset dari *SGML*, yaitu *HTML* juga berguna untuk menjelajahi internet. Khususnya bagi mereka yang menggunakan *World Wide Web*. Versi terakhir dari *HTML* adalah *HTML 4.01*, meskipun saat ini telah berkembang *XHTML* yang merupakan pengembangan dari *HTML*[10]

## 2.2. Penelitian Terkait

Berikut ini penulis menjabarkan beberapa penelitian yang berkaitan dengan penelitian yang dilakukan oleh penulis yang disajikan pada tabel 1.1 :

Tabel 1 Penelitian Terkait

NO	Nama Penulis	Judul	Tahun	Uraian Penulisan
1.	Nurlela (naskah Publikasi)	Sistem Informasi Penerimaan Siswa Baru SMA Negri 1 Merauke Menggunakan Netbeans	2013	Pembuatan Aplikasi dengan menggunakan netbeans untuk penerimaan siswa baru pada SMA Negri 1 Merauke.
2.	Hescaryo Doni .C	Sistem informasi Pendaftaran siswa baru online SMP N 2 Tawang sari	2014	Pembuatan web dengan menggunakan PHP dan sql untuk pendaftaran online pada SMP N 2 tawang sari.
3.	Enda Suhendar (STT Nurul Fikri)	Perancangan Aplikasi Penerimaan Siswa Baru MTS AL- Hidayah menggunakan Yii Framework 2	2015	Pembuatan Aplikasi dengan menggunakan Yii framework untuk pendaftaran Online untuk MTS Al- Hidayah.
4.	Reza Purnama Ibrahim (STT Nurul Fikri)	Perancangan Web Aplikasi PSB Onlie SMP IT Raudhatul Jannah dengan Framework Laravel	2018	Pembuatan sistem pendaftaran online untuk SMP IT Raudhatul Jannah dengan menggunakan Framework Laravel

### 2.3. Posisi Penelitian

Tabel 2 Posisi Penelitian

N O	Judul	Tahun	Metode	Platform	Bahasa Pemogra man
1	Sistem Informasi Penerimaan Siswa Baru SMA Negri 1 Merauke Menggunakan Netbeans	2013	Waterfall	Dekstop	Php
2	Sistem informasi Pendaftaran siswa baru online SMP N 2 Tawang sari	2014	Waterfall	Web	CSS java skript
3	Perancangan APlikasi Penerimaan Siswa Baru MTS AL- Hidayah menggunakan YII Framework 2	2015	Waterfall	Web	YII Framewo rk
4	Perancangan Web Aplikasi PSB Onlie SMP IT Raudhatul Jannah dengan Framework Laravel	2018	Waterfall	Web	Laravel5 Framewo rk

STT - NF

## 2.4.Perbedaan Aplikasi Dengan Aplikasi lainnya.

Tabel 3 Perbedaan aplikasi

N O	Judul	Tahun	Metode	Perbedaan
1	Sistem Informasi Penerimaan Siswa Baru SMA Negri 1 Merauke Menggunakan Netbeans	2013	Waterfall	Aplikasi menggunakan Desktop
2	Sistem informasi Pendaftaran siswa baru online SMP N 2 Tawang sari	2014	Waterfall	Web dengan menggunakan CSS jva script.
3	Perancangan Aplikasi Penerimaan Siswa Baru MTS AL-Hidayah menggunakan Yii Framework 2	2015	Waterfall	Web dengan menggunakan Bahasa Pemograman Y11 Framework

STT - NF