

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Beberapa penelitian yang terkait dengan penelitian ini adalah sebagai berikut.

Tabel 2.1 Perbandingan Penelitian

Parameter Penulis	Objek	Framework	Studi Kasus
Mohamad Rizky Adhiasta (2016)	Informasi Batik Solo	Bootstrap	Kota Solo
Nugroho Ari Saputro (2016)	Manajemen Presensi	Bootstrap	SMP Muhammadiyah 1 Kartasura
Adentya Maryo Santosa (2016)	Pengecekan Transaksi Internet Banking	Laravel dan Bootstrap	Bank Negara Indonesia dan Bank Mandiri
Tyar Kurnia Putra (2016)	E-Commerce	Bootstrap	CV. Kamajaya Yogyakarta
Yusuf Dzikrianto (2016)	Mobile Learning	jQuery Mobile dan Bootstrap	Program Studi Teknik Informatika STMIK AKAKOM Yogyakarta
Agus Irawan (2017)	Informasi Penunjang Perkuliahan Dengan Notifikasi Email	Laravel dan Bootstrap	STMIK AKAKOM Yogyakarta

Perbedaan penelitian yang sekarang dengan 5 penelitian sebelumnya terletak pada objek yang diteliti yaitu penyampaian informasi penunjang perkuliahan dari bagian akademik dan dosen kepada mahasiswa yang di gunakan untuk semua program studi di STMIK AKAKOM Yogyakarta dan juga penggunaan *framework* untuk membuat aplikasi.

## **2.2 Dasar Teori**

### **2.2.1 Penyampaian Informasi Penunjang Perkuliahan di**

#### **STMIK AKAKOM Yogyakarta**

Informasi penunjang perkuliahan yang diterima oleh mahasiswa di STMIK AKAKOM Yogyakarta masih diperoleh secara manual. Informasi dari bagian akademik diperoleh melalui pengumuman yang ditempel di depan ruangan kuliah dan informasi dari dosen diperoleh pada saat perkuliahan berlangsung. Hal tersebut menyebabkan informasi perkuliahan tidak langsung diketahui oleh mahasiswa jika belum datang ke ruangan kuliah ataupun tidak hadir pada perkuliahan.

### **2.2.2 Pengertian Framework**

*Framework* adalah struktur konseptual dasar yang berisi

kumpulan fungsi untuk tujuan tertentu yang sudah siap untuk digunakan, sehingga pembuatan aplikasi dapat dilakukan dengan lebih cepat karena kode programnya tidak di buat dari awal.

Beberapa alasan dari digunakannya *framework* dalam membuat aplikasi adalah sebagai berikut.

1. Aplikasi akan memiliki standar pemrograman yang universal.
2. Menghindari *repetitive work*.
3. Memudahkan dalam *team work*.
4. Memudahkan dalam *maintenance* dan pengembangan aplikasi di masa mendatang.
5. Hemat waktu dan biaya.

### **2.2.3 Laravel**

Laravel adalah *framework* bahasa pemrograman *Hypertext Preprocessor (PHP)* yang ditujukan untuk pengembangan aplikasi berbasis web dengan menerapkan konsep *Model View Controller (MVC)*. *Framework* ini dibuat oleh Taylor Otwell dan pertama kali dirilis pada tanggal 9 Juni 2011. Laravel berlisensi *open source* yang artinya bebas digunakan tanpa harus melakukan pembayaran. Alamat *website* resmi dari *framework* Laravel adalah <https://laravel.com>. Fitur-fitur modern Laravel yang sangat

membantu *developer* dalam membuat aplikasi adalah *Bundles*, *Eloquent ORM (Object-Relational Mapping)*, *Query Builder*, *Application Logic*, *Reverse Routing*, *Resource Controller*, *Class Auto Loading*, *View Composers*, *Blade*, *IoC Containers*, *Migration*, *Database Seeding*, *Unit Testing*, *Automatic Pagination*, *Form request*, dan *Middleware*.

*Framework* Laravel juga memiliki beberapa keunggulan sebagai berikut.

1. Menggunakan *Command Line Interface (CLI) Artisan*.
2. Menggunakan *package manager PHP Composer*.
3. Penulisan kode program lebih singkat, mudah dimengerti, dan ekspresif.

Kemudian untuk cara instalasi *framework* Laravel dapat dilakukan dengan 3 cara yaitu.

1. Melalui *Installer* Laravel.
2. Menggunakan *Composer* dengan mengetikkan perintah *create-project*.
3. *Download source code* Laravel secara lengkap melalui GitHub dengan alamat <https://github.com/laravel/laravel/>.

Pada tanggal 23 Agustus 2016 *framework* Laravel versi 5.3 resmi dirilis. Versi 5.3 akan mendapatkan dukungan *bug fixes* selama enam bulan dan *security fixes* selama satu tahun. Untuk menggunakan Laravel versi 5.3 komputer atau server yang digunakan harus memenuhi persyaratan sebagai berikut.

1. *PHP*  $\geq$  5.6.4
2. *OpenSSL PHP Extension*
3. *PDO PHP Extension*
4. *Mbstring PHP Extension*
5. *Tokenizer PHP Extension*
6. *XML PHP Extension*

Fitur *framework* Laravel yang ditekankan pada penelitian ini adalah *Blade*, *Migration*, *Eloquent ORM*, *Resource Controller*, dan *Middleware*. Berikut adalah penjelasan mengenai lima fitur tersebut.

1. *Blade*

*Blade* adalah *template engine*. Pada dasarnya *Blade* adalah *view* namun dengan menggunakan *Blade* akan mempermudah untuk mengatur tampilan *website* dan menampilkan data. Cara untuk membuat file *view* menjadi file *Blade* adalah dengan

menambahkan ekstensi `.blade.php` pada file *view*. Dan cara untuk memanggil file *Blade* sama dengan cara untuk memanggil file *view* biasa. Contoh program berikut adalah perbandingan antara file *view* biasa dengan file *Blade*.

<b>mahasiswa.php</b>
<pre> &lt;div id="Mahasiswa"&gt;   &lt;h1&gt;Mahasiswa&lt;/h1&gt;   &lt;?php if(!empty(\$mahasiswa)): ?&gt;     &lt;ul&gt;       &lt;?php foreach(\$mahasiswa as \$mhs): ?&gt;         &lt;li&gt; &lt;?= \$mhs ?&gt; &lt;/li&gt;       &lt;?php endforeach ?&gt;     &lt;/ul&gt;   &lt;?php else: ?&gt;     &lt;p&gt;Tidak ada data Mahasiswa.&lt;/p&gt;   &lt;?php endif ?&gt; &lt;/div&gt; </pre>
<b>mahasiswa.blade.php</b>
<pre> &lt;div id="Mahasiswa"&gt;   &lt;h1&gt;Mahasiswa&lt;/h1&gt;   @if(!empty(\$mahasiswa))     &lt;ul&gt;       @foreach(\$mahasiswa as \$mhs)         &lt;li&gt; {{ \$mhs }} &lt;/li&gt;       @endforeach     &lt;/ul&gt;   @else     &lt;p&gt;Tidak ada data Mahasiswa.&lt;/p&gt;   @endif &lt;/div&gt; </pre>

Dari kedua contoh program diatas dapat disimpulkan jika dengan menggunakan file *Blade* maka penulisan program akan menjadi lebih singkat dan rapi.

## 2. Migration

*Migration* adalah fitur yang menyediakan cara baru untuk membuat *database*. Dengan menggunakan *migration* cara membuat *database* melalui *Command Line Interface (CLI) database* atau dengan menggunakan aplikasi *database manager* digantikan dengan menggunakan *class*. Tahapan menggunakan *migration* adalah membuat *class* kemudian melakukan perintah *migrate* melalui *Command Line Interface (CLI) artisan*.

Keuntungan menggunakan *migration* adalah *class* yang dibuat bisa dipakai untuk membuat *database* pada berbagai macam *Relation Database Management System (RDBMS)* yang didukung oleh Laravel. Sebagai contoh misalnya aplikasi yang digunakan selama ini menggunakan *database MySQL*, kemudian karena alasan pengembangan aplikasi maka akan dilakukan penggantian *database* ke PostgreSQL. Dalam proses penggantian tersebut tidak perlu membuat *class* lagi, tinggal melakukan perintah *migrate* melalui *Command Line Interface (CLI) artisan*.

Keuntungan lain dari menggunakan *migration* adalah semua perubahan yang dilakukan pada *database* akan disimpan pada suatu tabel. Sehingga bisa dilakukan pembatalan (*rollback*) pada

*database* jika melakukan perubahan yang tidak benar.

### 3. Eloquent ORM

*Eloquent ORM* adalah implementasi dari *ActiveRecord* yang digunakan untuk mengatur relasi antar tabel di *database*. Pada *Eloquent ORM* tabel direpresentasikan dalam bentuk kelas dan data yang tersimpan didalam tabel direpresentasikan dalam bentuk objek. Relasi yang dapat diatur menggunakan *Eloquent ORM* adalah sebagai berikut.

- a. *One-to-One* yaitu relasi satu ke satu. Pada relasi ini digunakan *method hasOne* dan *belongsTo*.
- b. *One-to-Many* yaitu relasi satu ke banyak. Pada relasi ini digunakan *method hasMany* dan *belongsTo*.
- c. *Many-to-One* yaitu relasi banyak ke satu. Pada relasi ini digunakan *method belongsTo* dan *hasMany*.
- d. *Many-to-Many* yaitu relasi banyak ke banyak. Pada relasi ini digunakan *method belongsToMany*.

### 4. Resource Controller

*Resource Controller* adalah fitur yang digunakan untuk mempercepat pembuatan *controller*. Sebagai contoh misalnya ada *controller* yang menangani semua *HTTP request* terhadap data



dosen, untuk membuat *controller* tersebut hanya perlu mengetikkan perintah berikut.

```
php artisan make:controller DosenController --resource
```

Perintah diatas akan menghasilkan *controller* DosenController.php yang disimpan pada folder app/Http/Controllers. Tabel dibawah ini adalah daftar *action* yang dapat dilakukan oleh *controller* DosenController.php.

Tabel 2.2 Daftar Action DosenController.php

No	Verb	URI	Action	Route Name
1	GET	/dosen	index	dosen.index
2	GET	/dosen/create	create	dosen.create
3	POST	/dosen	store	dosen.store
4	GET	/dosen/{dosen}	show	dosen.show
5	GET	/dosen/{dosen}/edit	edit	dosen.edit
6	PUT/PATCH	/dosen/{dosen}	update	dosen.update
7	DELETE	/dosen/{dosen}	destroy	dosen.destroy

Setelah membuat *controller* DosenController.php hal yang harus dilakukan selanjutnya adalah membuat satu baris kode program pada *route*.

```
Route::resource('dosen', 'DosenController');
```

Satu baris kode program pada *route* diatas akan menangani semua *route* untuk melihat, menambah, mengedit, dan menghapus data dosen.

Jadi dapat disimpulkan dengan menggunakan fitur *Resource Controller* dapat mempercepat pembuatan *controller* serta dapat menyederhanakan *route* untuk *controller*.

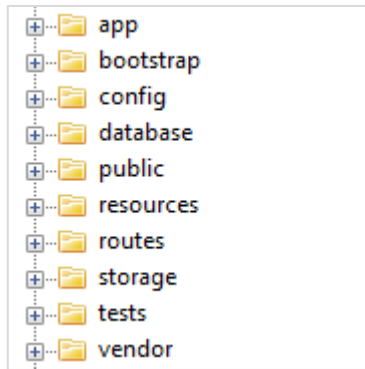
## 5. Middleware

*Middleware* adalah fitur yang menyediakan mekanisme untuk memfilter *HTTP request* yang masuk ke aplikasi. Laravel memiliki beberapa *Middleware* yaitu *Authenticate*, *EncryptCookies*, *RedirectIfAuthenticated*, dan *VerifyCsrfToken*.

Sebagai pembahasan akan dibahas *Middleware Authenticate*. *Middleware* tersebut akan memeriksa apakah *user* sudah *login* atau belum. Jika *user* sudah *login* maka *request* akan dilanjutkan ke halaman yang dikehendaki oleh *user*. Tetapi jika *user* belum *login* maka *Middleware Authenticate* akan mengarahkan *user* ke halaman *login*.

Jika *Middleware* yang sudah ada pada Laravel kurang sesuai dengan kebutuhan ataupun tidak sesuai dengan kebutuhan maka dapat dibuat sendiri *Middleware* yang sesuai dengan kebutuhan.

Struktur folder dari *framework* Laravel 5.3 yang masih *default* dapat dilihat pada gambar 2.1.



Gambar 2.1 Struktur Folder Laravel 5.3

- a. Folder *app* adalah folder yang berisi kode program inti dari aplikasi yang akan dibuat. *Model* dan *controller* tersimpan pada folder ini.
- b. Folder *bootstrap* adalah folder yang berisi konfigurasi *autoloading* dan terdapat juga folder *cache* yang menyimpan file-file yang dihasilkan secara otomatis oleh Laravel untuk mengoptimasi kinerja dari sistem yang dihasilkan.
- c. Folder *config* adalah folder yang berisi semua file konfigurasi aplikasi.
- d. Folder *database* adalah folder yang berisi file *database migration* dan *seeds*.

- e. Folder *public* adalah folder yang berisi file *index.php*. File tersebut digunakan sebagai *entry point* untuk menangani semua *request* yang masuk ke aplikasi. Pada folder ini juga dapat disimpan beberapa aset dari aplikasi seperti gambar, *JavaScript*, dan *CSS*.
- f. Folder *resources* adalah folder yang berisi file *view* dari aplikasi yang dibuat. Selain itu terdapat juga file *language* yang digunakan aplikasi.
- g. Folder *routes* adalah folder yang berisi file yang digunakan untuk mendefinisikan semua *route* ke aplikasi. Secara *default* ada tiga file *route* yang disediakan Laravel yaitu *api.php*, *console.php*, dan *web.php*.
- h. Folder *storage* adalah folder yang berisi *template Blade* yang dikompilasi, file *session*, file *cache*, dan file lainnya yang dihasilkan secara otomatis oleh Laravel.
- i. Folder *tests* adalah folder yang berisi semua file *test* yang dibuat untuk aplikasi.
- j. Folder *vendor* adalah folder yang menyimpan semua *library* yang digunakan.

#### **2.2.4 Bootstrap**

Bootstrap adalah *framework* bahasa pemrograman *Cascade Style Sheet (CSS)*, *Hyper Text Markup Language (HTML)*, dan *JavaScript* yang ditujukan untuk membuat tampilan aplikasi berbasis web menjadi responsif. Maksud responsif adalah tampilan aplikasi web akan menyesuaikan dengan ukuran layar dari perangkat yang mengaksesnya. *Framework* ini dibuat oleh Mark Otto dan Jacob Thornton. Bootstrap pertama kali dirilis pada tanggal 19 Agustus 2011 dan berlisensi *open source* yang artinya bebas digunakan tanpa harus melakukan pembayaran. Alamat *website* resmi dari *framework* Bootstrap adalah <http://getbootstrap.com>. Untuk mengunduh *framework* Bootstrap dapat dilakukan melalui *website* tersebut atau dapat melalui GitHub dengan alamat <https://github.com/twbs/bootstrap/>.

#### **2.2.5 MySQL**

MySQL adalah *database server* yang digunakan untuk menyimpan dan manajemen data, dalam bahasa Inggris disebut *Database Management System (DBMS)*. MySQL merupakan implementasi dari sistem manajemen basis data relasional, dalam bahasa Inggris disebut *Relation Database Management System*

(RDBMS). Secara umum *Structured Query Language (SQL)* pada MySQL dibagi menjadi dua yaitu.

1. *Data Definition Language (DDL)* yang digunakan untuk membuat objek pada basis data seperti tabel, *indeks*, *sequence*, dan *view*. Yang termasuk dalam perintah *DDL* adalah *CREATE*, *ALTER*, dan *DROP*.
2. *Data Manipulation Language (DML)* yang digunakan untuk memanipulasi objek pada basis data. Yang termasuk dalam perintah *DML* adalah *SELECT*, *INSERT*, *UPDATE*, dan *DELETE*.

*Structured Query Language (SQL)* yang dipakai pada aplikasi ini adalah *CREATE*, *DROP*, *SELECT*, *INSERT*, *UPDATE*, dan *DELETE*.