

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Website**

*Website* merupakan suatu situs yang dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara dan gabungan dari semuanya baik bersifat statis maupun dinamis sehingga membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan pada halaman[5]. *Website* memiliki dua sisi yaitu pada sisi server dan sisi client. Pada sisi client maka pada prosesnya disebut dengan front-end karena bersinggungan langsung dengan *client* sehingga erat kaitannya dengan design.

#### **2.2. Front-End**

*Front-end* adalah bagian dari pengembangan situs web di mana tampilan atau desain situs web dibuat menggunakan alat Pengembangan *Front-end*[6]. Tampilan diharapkan dapat berfungsi sebagaimana semestinya dan penggunaan alat yang dimaksud adalah penggunaan HTML, CSS, dan *framework* serta penggunaan *library*.

#### **2.3. Visual Studio Code**

Visual studio code merupakan *software* yang sangat ringan, namun kuat editor kode sumbernya yang berjalan dari desktop. Muncul dengan *builts-in* dukungan untuk JavaScript, naskah dan Node.js dan memiliki array beragam ekstensi yang tersedia untuk bahasa lain, termasuk C++, C#, Python dan PHP. Hal ini didasarkan sekitar Github merupakan Elektron, yang merupakan versi *cross-platform* dari Atom komponen kode-editing, berdasarkan JavaScript dan HTML5. Editor ini merupakan fitur lengkap lingkungan pengembangan terpadu (IDE) dirancang untuk pengembang yang bekerja dengan teknologi

*cloud* yang terbuka Microsoft. Visual Studio Code menggunakan *open source* NET perkakas untuk memberikan dukungan untuk ASP.NET C# kode, membangun alat pengembang omnisharp NET dan compiler Roslyn. Antarmuka yang mudah untuk bekerja didasarkan pada gaya *explorer* umum, dengan panel di sebelah kiri untuk menunjukkan semua file dan folder. Sehingga memiliki akses panel editor di sebelah kanan untuk menunjukkan isi dari file yang telah dibuka. Hal tersebut memiliki fungsi yang baik, dengan *intellisense* dan *autocomplete* bekerja dengan baik untuk JSON, CSS, HTML, {kurang}, dan Node.js [7].

#### 2.4. Laravel

Laravel merupakan sebuah *framework* web berbasis PHP yang *open-source* dan tidak berbayar, diciptakan oleh Taylor Otwell dan diperuntukan untuk mengembangkan aplikasi web yang menggunakan pola MVC. Struktur pola MVC pada laravel sedikit berbeda dengan struktur pola pada umumnya. Pada laravel terdapat routing yang menjembatani antara *request* dari *user* dan *controller*. *controller* tidak langsung menerima request tersebut. Ilustrasi dari konsep MVC pada Laravel yang saling terhubung antara satu sama lain dalam proses pembuatan *website* melalui laravel [8].

Menurut Naista (2017) *framework* adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah yang kompleks. Singkatnya, *framework* adalah wadah atau kerangka kerja dari sebuah *website* yang akan dibangun. Melihat kerangka tersebut waktu pengerjaan *website* akan lebih singkat dan mudah dalam melakukan perbaikan. Salah satunya *framework* laravel, dalam penggunaannya laravel terdapat file yang sifatnya default seperti *vendor*. File tersebut tidak boleh dihapus sembarangan sehingga ukuran *website* yang dibuka berukuran cukup besar. Selain itu dibutuhkan koneksi internet dalam instalasi dan mengunduh *library* laravel. Berikut merupakan dasar-dasar laravel [9].

### 1. *Artisan*

*Artisan* adalah *command line* atau perintah yang dijalankan melalui terminal dan disediakan beberapa perintah yang dapat digunakan selama melakukan pengembangan serta dalam pembuatan aplikasi. Salah satu fungsi dari php artisan yaitu “php artisan serve”. Php artisan serve mempunyai fungsi untuk membuka *website* yang telah dibuat tanpa menggunakan web server lokal.

### 2. *Routing*

*Routing* merupakan suatu proses agar suatu item yang diinginkan dapat sampai pada tujuan. Dengan menggunakan *routing* dapat ditentukan halaman-halaman yang akan muncul ketika dibuka oleh *user*. Pengaturan tersebut terletak pada file *web.php* dan pada folder *routes*.

### 3. *Controller*

*Controller* adalah suatu proses yang bertujuan untuk mengambil permintaan, inisialisasi, dan memanggil model untuk dikirim ke *view*. Terdapat dua cara membuat *controller* pada *laravel*, pertama dibuatlah *file controller* secara manual dan dituliskan *code extends controller* di dalamnya. Cara kedua yaitu dibuat *file controller* menggunakan *command line* dengan menuliskan “php artisan make controller nama\_file\_controller”. Permintaan yang dibuat dalam *laravel* harus berada di dalam *controller*, setelah itu dilempar melalui *routing* untuk dapat permintaan yang diinginkan.

### 4. *View (blade templating)*

*Blade* merupakan *template engine* bawaan dari *laravel*, dengan memiliki kode-kode yang lebih mudah untuk menghasilkan *laravel*. Cara membuat *file.blade* dilakukan secara manual dengan membuat nama\_file.php.blade didalam *folder views*.

### 5. *Middleware*

*Middleware* adalah penengah antara *request* yang masuk dengan *controller* yang akan dituju. Proses membuat *middleware* yaitu menggunakan *artisan* dengan mengetikkan “php artisan make:middleware nama\_file”. Oleh karena itu, *file middleware* berada di dalam folder *middleware*.

### 6. *Session*

*Session* adalah cara yang digunakan untuk menyimpan data pada server dan menyimpan data tersebut digunakan pada beberapa halaman itu sendiri. Pada *session* dapat dibuat menggunakan *request* dan fungsi global *helper session*.

## 2.5. Xampp

*XAMPP* merupakan suatu *tools* yang menyediakan paket perangkat lunak ke dalam sebuah paket. Berfungsi sebagai *server* yang berdiri sendiri (*localhost*) yang terdiri dari beberapa program yaitu Apache, HTTP Server, MySQL database dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl [10].