

BAB II

LANDASAN TEORI

2.1. Sistem Informasi

Sistem merupakan kumpulan elemen – elemen yang saling terkait dan bekerja sama untuk memproses masukan (*input*) yang ditujukan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (*output*) yang di inginkan. Suatu sistem merupakan suatu unsur-unsur yang berhubungan satu sama lain yang berfungsi untuk mencapai tujuan tertentu. Sistem informasi bisa dimaknai sebagai sebuah sistem yang bertujuan menampilkan informasi [7].

2.2. Arsip

Arsip merupakan sebuah kumpulan dokumen atau catatan sejarah yang dapat memberikan informasi terkait suatu lembaga, kelompok orang, atau tempat tertentu. Arsip dokumen dapat dikelompokkan menjadi surat, akta, dan sebagainya. Secara umum arsip adalah sebuah catatan atau rekaman yang diketik, dicetak, atau ditulis dalam wujud gambar, angka dan huruf yang memiliki arti serta tujuan tertentu yang akan digunakan sebagai suatu bahan informasi dan juga komunikasi yang direkam dalam berbagai media, seperti media komputer, kertas, atau kertas film. Dalam sebuah organisasi atau perusahaan, arsip sering dijadikan sebagai alat untuk menyimpan dan mengatur catatan-catatan organisasi [8].

2.3. Website

Website merupakan suatu kumpulan dari banyak halaman yang memuat suatu informasi berupa teks dan data multimedia. Data multimedia dapat berupa audio, video, animasi, gambar yang disajikan dalam sebuah halaman baik statis maupun dinamis sehingga semua konten tersebut saling terkait dan semua itu bisa diakses oleh pengguna internet melalui peramban web [9]. *Website* statis merupakan sebuah *website* yang memiliki tampilan selalu tetap dan tidak mengalami perubahan. Perubahan yang dimaksud

tidak termasuk desain *website*. Contoh *website* bertipe statis adalah *website* yang menampilkan profil suatu perusahaan atau organisasi. *Website* dinamis merupakan *website* dengan halaman yang selalu mengalami perubahan sesuai dengan kebutuhan dari *website* tersebut baik dari segi bisnis maupun karena mengikuti tren atau perkembangan zaman. Biasanya *website* ini memiliki fitur tampilan yang interaktif, kolom komentar dan *chat* [10].

Sebuah *website* dalam bekerja memiliki dua sisi yang saling bekerja sama untuk bisa berfungsi dengan baik dan memiliki jenis *script* yang berbeda. Kedua sisi tersebut adalah sisi *client* dan server. Sisi *client* merupakan sisi dari pengguna internet yang mengakses *website* dengan mengirim permintaan ke web server menggunakan web browser bisa secara lokal maupun melalui internet dan kode program di eksekusi pada sisi *client*. Server merupakan sisi yang memberikan kepada *client* halaman web yang diminta melalui web browser. Terdapat kode program khusus yang hanya bisa berjalan pada sisi server bersamaan saat halaman web diminta *client*, biasanya dalam format PHP dan bekerja untuk menyiapkan konten yang dibutuhkan untuk ditampilkan di halaman web dengan mengambil datanya di *database* server sebelum halaman web dikirimkan ke *client* [11].

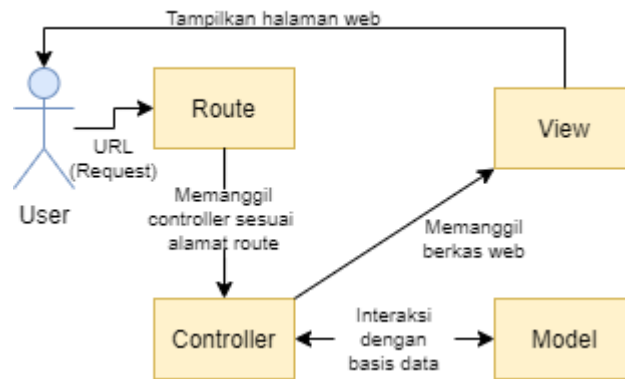
2.4. Database MySQL

Database atau basis data merupakan suatu kumpulan informasi yang dikumpulkan dan di simpan dalam sebuah perangkat komputer. Data yang disimpan disusun dalam format atau struktur tabel. Sebuah basis data terdapat sebuah sistem manajemen yang bernama DBMS. DBMS (*Database Management Sistem*) merupakan sebuah perangkat lunak yang dirancang untuk mengelola sebuah *database*, mulai dari membuat, membaca, edit, tambah, *update* dan hapus data dengan tampilan yang *user friendly*. Dalam membangun *website*, *database* biasanya dibuat dan dikelola secara otomatis oleh *framework* jika dasar dari pembuatan *website* menggunakan bantuan *framework* [12].

2.5. *Framework* Laravel

Framework merupakan sebuah perangkat lunak yang dirancang untuk mempermudah dalam membangun dan berkolaborasi sebagai tim dalam pembuatan *website*. *Framework* membantu pengembang web dalam mengoptimalkan kode program agar lebih efisien dan tidak berulang, lebih mudah dalam bekerja secara kelompok dikarenakan aturan pemrograman yang sudah diatur oleh *framework* tersebut sehingga tidak terjadi perbedaan model pemrograman yang berbeda dan membingungkan. *Framework* lebih optimal untuk membuat *website* yang kompleks dan bukan untuk *website* sederhana. *Framework* yang biasa digunakan untuk membangun *website* salah satunya adalah Laravel [13].

Laravel merupakan sebuah *framework* web yang menggunakan pemrograman berbasis PHP yang bersifat *open-source* dan gratis untuk diunduh dan digunakan, diciptakan oleh Taylor Otwell dan ditujukan untuk membangun ataupun mengembangkan aplikasi berbasis web yang menggunakan sistem MVC. MVC yang dimaksud adalah *Model* yang mengurus masalah pengelolaan basis data, *View* untuk tampilan dan *Controller* untuk menjembatani *model* dan *view* [13]. Pada Laravel terdapat *routing* yang digunakan atau berfungsi sebagai perantara antara permintaan dari *user* dengan *controller* melalui url. *controller* akan berinteraksi dengan basis data melalui *model* dan mencari berkas web dan memrosesnya kemudian akhirnya akan ditampilkan ke *user*. Ilustrasi MVC digambarkan pada Gambar 2.1 yang menjelaskan alur kerja MVC pada Laravel.



Gambar 2. 3 Ilustrasi MVC Pada Laravel [14].

Dengan menggunakan Laravel, proses *development* web mendapat keuntungan daripada *development* secara *native* (membuat seluruh pemrograman dari dasar). Kelebihannya adalah *website* menjadi mudah dikembangkan, terdapat *namespace* dan tampilan yang memudahkan dalam mengelola dan mengatur sumber daya *website* dan proses pengembangan bisa dilakukan dengan cepat karena bisa dikombinasikan dengan komponen *framework* lain. Laravel memiliki *tools* utama yang bisa digunakan yaitu Composer dan Artisan [15].

Composer merupakan sebuah *tools* yang digunakan sebagai *dependencies management*. Di dalam Composer terdapat *dependencies* dan *library* yang disimpan dalam format *file composer.json* yang ditempatkan di folder utama *website*. *Library* yang terdapat di *dependencies management* diantaranya adalah *library* untuk implementasi validasi dan proteksi dari *spamming* menggunakan *Google reCaptcha* [15].

Artisan merupakan sebuah perintah yang membantu dalam membangun *website*. *Artisan* berjalan menggunakan *command line* yang didalamnya memuat banyak fungsi untuk mengelola komponen *website* di dalam *project* secara otomatis menggunakan perintah yang sudah disediakan. Untuk melihat semua perintah yang tersedia digunakan perintah *php artisan list*. Laravel juga memiliki *built-in* HTTP Server yang bisa dijalankan dengan menggunakan perintah *php artisan serve* [16].

Pertama kali membuat proyek Laravel terdapat *file* dan *folder* yang

sudah ada secara *default* yang membantu pembuatan *website* agar terstruktur. *File* dan *folder* tersebut tidak boleh dihapus dan di modifikasi secara sembarangan karena dibutuhkan agar *website* bisa bekerja. Hal inilah yang membuat ukuran proyek *website* yang dibuka berukuran cukup besar saat baru pertama kali dibuka. Berikut merupakan beberapa dasar dari Laravel [17].

1. *Routing*

Routing merupakan proses yang membuat sebuah *request* mencapai tujuan. Dengan menggunakan *routing* dapat menentukan halaman awal dari *website* saat pertama kali diakses atau mengarahkan *request* dari *user* berdasarkan *url*. *File* untuk konfigurasi *routing* terdapat di *web.php* di dalam folder *routes*.

2. *Middleware*

Middleware berfungsi untuk menyimpan *class* dari *middleware php*. *Class middleware* dijalankan sebelum *HTTP request* dari *user* di jalankan oleh *controller* yang berfungsi sebagai *filter*. Proses *filter* yang dimaksud adalah menolak akses dari *user* yang belum melakukan otentifikasi atau *login* ke sistem dengan cara mengarahkan *user* ke halaman beranda atau halaman *login*.

3. *Controller*

Controller merupakan suatu proses 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 menggunakan perintah *php artisan make controller nama_file_controller* melalui *command line*. 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* di dalam *folder views*.

5. *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*.

2.6. Visual Studio Code

Visual Studio Code adalah sebuah teks editor *multiplatform* yang komplet dan Handal buatan *Microsoft*. Teks editor mendukung banyak bahasa pemrograman seperti JavaScript, Typescript, dan Node.js, serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat dipasang di Visual Studio Code seperti C++, C#, Python, Go, Java, dll. Visual Studio Code menyediakan *Intellisense*, *Git Integration*, *Debugging*, dan fitur ekstensi. Fitur-fiturnya akan terus bertambah seiring dengan perkembangan versi *Visual Studio Code*. Perkembangan versi *Visual Studio Code* ini juga dilakukan berkala setiap bulan, dan inilah yang membuat *Visual Studio Code* unggul dibandingkan teks editor lainnya [18].

2.7. PHP

PHP (*Hypertext Preprocessor*) adalah sebuah bahasa pemrograman *serverside scripting* yang bersifat *open-source*. PHP merupakan bahasa pemrograman *server-side*, maka *script* dari PHP nantinya akan diproses di server. Jenis server yang sering digunakan bersama dengan PHP antara lain Apache, Nginx, dan LiteSpeed. Sebagai sebuah *scripting language*, PHP menjalankan instruksi pemrograman saat proses *runtime*. *PHP* juga merupakan bahasa pemrograman yang bersifat *open source*, artinya

pengguna bebas memodifikasi dan mengembangkan sesuai dengan kebutuhan mereka [19].

PHP awalnya diciptakan oleh Rasmus Lerdorf untuk memantau orang-orang yang mengunjungi halaman web miliknya. Seiring waktu, bahasa ini menjadi makin populer, dan Lerdorf akhirnya merilisnya sebagai proyek *open-source*. Kemudian para pengembang mulai menggunakan, memperbaiki, dan menyempurnakan kode ini, hingga kemudian menjadi bahasa penulisan skrip yang kini banyak digunakan. PHP sering kali digunakan untuk pengembangan web. Hal ini dikarenakan PHP memiliki sebuah fitur yang dirasa sangat berguna, yaitu kemampuannya untuk disematkan ke *file* HTML. Fitur ini digunakan saat pengembang web tidak ingin memperlihatkan *source code* dari web yang dikerjakannya dengan alasan keamanan. Fitur ini juga akan berguna ketika harus menggunakan kode HTML yang sama berulang kali, tidak perlu menulis kodenya lagi [20].

2.8. PhpMyAdmin

PhpMyAdmin adalah aplikasi berbasis web yang digunakan untuk melakukan pengelolaan basis data MySQL dan atau alat yang paling populer untuk mengelola basis data MySQL. Aplikasi ini banyak digunakan dalam pengembangan situs web, seperti WordPress. Di beberapa template WordPress terkadang juga memerlukan akses ke basis data. Kegunaan dari PhpMyAdmin diantaranya adalah sebagai berikut [21].

- a. Membuat, menghapus, mengelola *user*,
- b. Membuat dan menghapus basis data, tabel dan baris,
- c. Mencari objek di *database* dan tabel,
- d. *Import* dan *Export* data dengan format berbeda seperti SQL, XML dan CSV,
- e. Mengeksekusi perintah atau *query* SQL,
- f. *Backup* basis data MySQL.

2.9. XAMPP

XAMPP adalah perangkat lunak atau aplikasi komputer yang banyak digunakan dalam dunia web developer yang juga bisa dipelajari untuk membuat *website*. XAMPP adalah perangkat lunak berbasis web server yang bersifat *open source* serta mendukung di berbagai sistem operasi seperti Linux, Windows, Mac OS, dan juga Solaris. XAMPP biasa digunakan untuk menghemat anggaran karena mampu menggantikan peran web *hosting* dengan cara menyimpan *file* web ke dalam *hosting* lokal agar bisa dipanggil lewat *browser*. *Software* XAMPP dikembangkan oleh tim bernama Apache Friends pada tahun 2002, yang bisa didapatkan secara gratis dengan label GNU (*General Public License*) [22].

2.10. VCS

Version Control System (VCS) adalah sebuah sistem yang melakukan *source code management* (SCM) untuk mengelola perubahan di setiap dokumen, program komputer, *website*, dan kumpulan pemrograman lainnya. Sebagai contoh saat mengerjakan skripsi, akan ada saat melakukan revisi. Berkas yang pertama akan tetap di simpan dan berkas hasil revisi akan disimpan dengan nama yang lain. Jadi peran *version control* disini adalah mempertahankan perubahan (histori) dari perubahan yang telah dilakukan, sehingga jika sewaktu-waktu ingin mengembalikan ke keadaan sebelumnya akan lebih mudah. Dari contoh tersebut, bisa diketahui setiap proses akan memenuhi kapasitas penyimpanan, oleh karena itu sistem VCS memainkan peran untuk membantu penyimpanan berupa histori tanpa menyimpan *file* baru, yang tersimpan hanya perubahan data. Sehingga kapasitas penyimpanan *file* menjadi ringan. Layanan aplikasi VCS yang sering digunakan adalah Gitlab dan Github [23].

2.11. Git

Git merupakan perangkat lunak berbasis *Version Control System* (VCS) yang bertugas untuk mencatat perubahan seluruh *file* atau *repository* suatu proyek. Developer perangkat lunak biasa menggunakan Git untuk

distributed revision (VCS terdistribusi), hal ini bertujuan untuk menyimpan basis data tidak hanya ke satu tempat. Namun semua orang yang terlibat dalam penyusunan kode dapat menyimpan ke dalam basis data ini. Prosedur yang diterapkan ini dapat membantu antar divisi proyek untuk memantau dan menghubungkan (merger) antar ekstensi yang berbeda dengan mudah. Sehingga aplikasi yang dibuat oleh sebuah tim proyek dapat berfungsi tanpa menghubungkan secara manual. Terdapat istilah *commit* pada Git yang berfungsi untuk menyimpan riwayat perubahan data pada sebuah berkas. Melalui *commit*, developer dapat kembali ke *source code* sebelumnya dengan istilah *checkout* [23].