

BAB II TINJUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian dilakukan Rifky Lana Rahardian, Ni Luh Gede Pivin Suwirmayanti (2018). Membuat Studi Kasus Portal wisata, yang dijadikan sebagai *referensi* mengenai pengalaman yang disimpan disebuah *portal* wisata, dengan menggunakan *Platform Website*, dan yang dihasilkan dari penelitian ini adalah sebuah portal informasi wisata untuk *tourist* yang ingin melakukan perjalanan wisata.

Penelitian dilakukan Eko Piyanto (2019). Membuat sebuah studi kasus Sistem Informasi dengan menggunakan *Platform Website* dan informasi yang di dihasilkan dari penelitian ini adalah sebuah sistem informasi wisata yang ada di Kabupaten Purworejo dengan menerapkan penggunaan Teknologi *framework laravel*.

Penelitian dilakukan Indra Soraya, Asep Deddy Supriyatna (2017). Membuat studi kasus mengembangkan sistem informasi di Garut. *Platform* digunakan dalam penelitian dilakukan secara online dan yang dihasilkan dari penelitian ini adalah sebuah informasi pemesanan paket wisata yang dilakukan secara *online*.

Penelitian dilakukan Sandy Kautsar, Tri Ferga Prasetyo (2019). Membuat Studi kasus tentang perancangan *E-Travel*, dan *Platform* yang digunakan yaitu *Website* dan akan menghasilkan sebuah aplikasi *E-travel* yang bisa digunakan untuk mempermudah perencanaan, pengelolaan dan mengatur transaksi diperusahaan *Travel* dan digunakan untuk acuan dalam pengambilan sebuah keputusan.

Penelitian dilakukan Wildan Hamdani, Suharnawi (2018). Membuat studi kasus pengembangan sistem informasi pariwisata Kabupaten Tegal, dengan menggunakan Platform *Website* dan yang dihasilkan dari penelitian ini adalah membuat sebuah informasi wisata yang ada di Kabupaten Tegal.

Penelitian dilakukan SIWA Nada Ramadhanty (2022). Membuat studi kasus yang bertujuan memperkenalkan informasi wisata yang ada di Kota Pontianak dengan menggunakan *Platform Website*, dengan menggunakan teknologi *framework laravel* dan yang dihasilkan dari penelitian ini adalah sebuah aplikasi wisata yang mempermudah pengunjung melihat informasi wisata yang ada di Kota Pontianak. Perbandingan dapat dilihat pada Tabel 2.1 Tinjauan Pustaka.

Tabel 2. 1 Tinjauan Pustaka

No	Penulis	Studi Kasus	Platform	Hasil
1	Rifky Lana Rahardian, Ni Luh Gede Pivin Suwirmayanti (2018)	Memberikan informasi wisata di Bali dan membuat sebuah portal untuk berbagi sebuah pengalaman <i>Tourism</i> melakukan perjalanan wisata.	Website	Akan menghasilkan sebuah portal untuk <i>tourism</i> berbagi informasi wisata.
2	Eko Piyanto (2019)	Membuat sebuah Sistem Informasi.	Website	Akan menghasilkan sebuah sistem informasi yang ada di kabupaten Purworejo dengan <i>framework laravel</i> .
3	Indra Soraya, Asep Deddy Supriyatna (2017)	Mengembangkan sebuah sistem informasi yang ada di Garut.	Online	Akan menghasilkan sebuah informasi pemesanan paket wisata yang dilakukan secara online.
4	Sandy Kautsar, Tri Ferga Prasetyo(2019)	Perancangan Aplikasi E-Travel	Website	Akan menghasilkan aplikasi E-Travel .
5	Wildan Hamdani, suharnawi (2018)	Pengembangan Sistem Informasi pariwisata kabupaten Tegal	Website	Akan menghasilkan sebuah informasi yang ada di Kabuten Tegal.

6	SIWA Nada Ramadhanty	Menampilkan dan memperkenalkan informasi wisata di Kota Pontianak.	Website	Akan menghasilkan informasi website pariwisata yang ada di Kota Pontinak
---	----------------------	--	---------	--

2.2 Dasar Teori

2.2.1 Laravel 8.0

Laravel sering dikenal pada konsepnya yaitu *MVC (Model View Contraller)*, disertai beberapa *command line tool* digunakan beberapa instalasi *bundle* serta *packaging bundle* yang dilakukan melalui *command prompt*. Kesederhanaan dan *fleksibilitas* pada *interface* adalah salah satu fungsi terdapat pada *framework Laravel*.

Laravel merupakan *framework PHP* yang *expressif*, dimana sebagai *programmer* ketika melihat suatu *sintaks Laravel* dapat mengetahui sesuai kegunaan *sintaks*. Dengan *Eloquent ORM* menjadikan lebih simple *Laravel* ketika digunakan oleh *programmer*.

Berikut ini fitur-fitur yang dimiliki *framework Laravel*.

- a) *Bundles* yaitu fitur dengan sistem pengemasan *modular*, dan berbagai *bundler* tersedia untuk digunakan dalam pembuatan aplikasi.
- b) *Eloquent ORM* merupakan bagian internal penerapan *PHP* lanjutan yang berguna untuk mengatasi kendala yang terdapat pada *script* dan *database*.
- c) *Aplication Logic* merupakan perkembangan aplikasi dengan menggunakan *controller* bagian maupun dari deklarasi *router*.

- d) *Reverse Routing*, definisi cara keterkaitan antara *link* dan *router*.
- e) *Restfull Controllers*, sebagai pemberi pilihan pada programmer untuk memisahkan logika dalam menjalankan *HTTP Get*, melakukan *request POST*, permintaan melakukan perubahan *PUT/PATCH*, dan *DELETE* untuk menghapus yang tidak digunakan.
- f) *Migrations*, sistem yang berfungsi untuk mengatur skema *database*, sehingga menjadi penghubung antar skema database dengan *framework Laravel*.
- g) *View Compressor* kode yang berjalan ketika *view* akan ditampilkan.

Kelebihan *Framework Laravel* :

- a) *Template layout* yang ringan dimana pada *Laravel* memiliki *template engine* yang memudahkan *developer* dalam membentuk *template* yang bagus namun ringan di dalam *server* dan *hosting*.
- b) *Laravel* didukung *framework MVC (Model View Controller)*, *MVC* merupakan metode aplikasi dengan memisahkan data dengan tampilan berdasarkan komponen-komponen aplikasi seperti: memanipulasi data, *controller* dan *user interface*.
- c) *Laravel* mempunyai banyak *Library Object Oriented*, keunggulan *Laravel* dibanding *framework* lain mempunyai banyak *library object oriented* di mana pada *framework* lainnya tidak mempunyai sebanyak yang dimiliki *Laravel*, salah satu *library* sangat penting dalam *Laravel* adalah *library authentication*. *Library*

ini menyediakan fitur *Bcrypt hashing*, Fitur tersebut lebih canggih dibidang *otentifikasi* saat ini.

- d) Modul dalam *Laravel* bersifat *individual* dan *independen*. Dalam satu aplikasi *Laravel* terdapat lebih dari 20 *library* yang digunakan dan masing-masing modul bersifat *individu*. Selain bersifat *individu* modul tersebut bersifat *independen* dalam menjalankan prinsip *PHP* modern. Dengan begitu modul-modul membuat aplikasi *website* menjadi *informative*, *modular*, dan *responsive*.
- e) Tersedia *tool artisan* yaitu fitur *command line interface* yang dimiliki *Laravel*, dengan *tools* dapat mengetikkan perintah yang membantu dalam membangun aplikasi *website*. Terdapat juga fitur *Add on* yang memungkinkan untuk menambahkan berbagai fitur baru dalam *Laravel* dengan mudah. Fitur *Add on* merupakan perintah *Artisan* dan penggabungan dengan *framework Symphony*.
- f) Komunitas yang banyak dan tetap *update versi* hingga saat ini *Laravel* mempunyai banyak tambahan yang membantu *developer* seperti *Vue*, *Jetstream*, dan *Blade*.
- g) *Website* atau *project* mudah untuk diwariskan kepada *programmer* dengan adanya *source code* yang *readable* sehingga seorang *developer* mewariskan programnya kepada orang lain sangat mudah *dimaintenance*.
- h) Dalam segi pasar *Laravel* banyak peminat dari pada *framework* lainnya dikarenakan dinilai lebih ringan dan *fiturnya* melimpah.
- i) *Laravel* mendukung *restful api*, Jadi pengembang tidak perlu menulis ulang *kode* untuk membuat *restful api*.

Kekurangan Framework *Laravel* :

- a) *Laravel update* terlalu cepat.
- b) Aplikasi yang jalan di *browser* tidak langsung *update style cssnya* setelah *cssnya* diubah. Perlu dilakukan *clear cache* terlebih dahulu untuk merubahnya.
- c) *Fitur* yang terlalu luas, sehingga membuat *developer* bingung menggunakan *fitur* yang mana.
- d) Belajar untuk pemula yang sulit karena harus belajar *codeigniter* dahulu sebelum *Laravel*.
- e) Penulisan *script laravel* tergolong rumit dibanding *framework* lainnya.
- f) *Laravel* tidak mendukung *implicit routing*, hal ini membuat pengembang yang menggunakan *Laravel* perlu waktu lebih untuk mendaftarkan model binding ke dalam *library route service provider*.

2.2.2 Framework

Framework menurut Basuki (2010) adalah koleksi atau kumpulan potongan-potongan program disusun atau diorganisasikan dan digunakan untuk membuat aplikasi tanpa membuat kodenya dari awal.

2.2.3 Bootstrap

Menurut Ignas (2016) *Bootstrap* sebuah *framework CSS twitter* yang menyediakan komponen-komponen antarmuka siap dipakai dan telah dirancang untuk desain halaman *website* yang *artistik*. *Bootstrap* menyediakan *HTML*, *CSS* dan *Java Script* siap digunakan dan dikembangkan. *Bootstrap* digunakan untuk mempermudah

dan mempercepat dalam membangun sebuah *aplikasi berbasis website yang responsif* artinya tampilan yang dibuat *Bootstrap* akan menyesuaikan layar dari *browser* baik di *desktop tablet* maupun *mobile device*.