

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Pengertian Website**

*Website* adalah sebutan bagi sekelompok halaman *web* (*web page*), yang umumnya merupakan bagian dari suatu nama *domain* atau *sub domain* di *World Wide Web* (WWW) di Internet (Zaki & Community, 2011). Sebuah *webpage* adalah dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*), yang hampir selalu bisa diakses melalui HTTP, yaitu protokol yang menyampaikan informasi dari server website untuk ditampilkan kepada para pemakai melalui web browser baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*).

*Website* bersifat statis apabila isi informasi *website* tetap, jarang berubah, dan isi informasinya searah hanya dari pemilik website. *Website* bersifat dinamis apabila isi informasi *website* selalu berubah-ubah, dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna website. Halaman-halaman sebuah situs web diakses dari sebuah URL yang menjadi “akar” (*root*), yang disebut homepage (halaman induk; sering diterjemahkan menjadi “beranda”, “halaman muka”), dan biasanya disimpan dalam server yang sama. *Website* ini dibuka melalui sebuah program penjelajah (*Browser*) yang berada di sebuah komputer. Program penjelajah yang bisa digunakan dalam komputer diantaranya: IE (*Internet Explorer*), Mozilla, Firefox, Netscape, Opera dan Google Chrome.

#### **2.2 Back End Website**

*Back end* merupakan segala sesuatu bagian dari website yang berhubungan dengan server dan database. Back end adalah mesin yang bekerja di balik layar, semua yang tidak dilihat oleh pengguna akhir atau berinteraksi langsung dengan pengguna. Pada dasarnya, cara kerja *back end* adalah memastikan agar suatu *website* atau aplikasi bisa bekerja dengan baik. Mereka memiliki tanggung jawab agar website bisa terintegrasi dengan baik. Pada akhirnya, tugas *back end* adalah fokus pada fungsi dari website bisa berjalan sesuai yang diinginkan.

#### **2.3 Domain Model**

Domain Model adalah cara untuk mengidentifikasi objek pada kata benda yang ada dalam daftar requirement yang diklasifikasikan pada area

domain permasalahan yang sama untuk dijadikan kandidat class pada class diagram. Class diagram sendiri memiliki fungsi untuk menunjukkan struktur dari sistem sehingga meningkatkan pemahaman atau gambaran umum sistem serta membantu business analyst untuk membuat model sistem dari sisi bisnis dan memberikan gambaran mengenai sistem atau perangkat lunak serta relasi yang terkandung di dalamnya.

## 2.4 Pejabat Pengelola Informasi dan Dokumentasi (PPID)

PPID adalah kepanjangan dari Pejabat Pengelola Informasi dan Dokumentasi, yang berfungsi sebagai pengelola dan penyampai dokumen yang dimiliki oleh Badan Publik sesuai dengan amanat UU 14/2008 tentang Keterbukaan Informasi Publik (*Profil PPID*, n.d.). Dengan keberadaan PPID maka masyarakat yang akan menyampaikan permohonan informasi lebih mudah dan tidak berbelit karena dilayani lewat satu pintu. Pejabat Pengelola Informasi dan Dokumentasi (PPID) adalah pejabat yang bertanggung jawab dibidang penyimpanan, pendokumentasian, penyediaan dan/ atau pelayanan informasi di badan public.

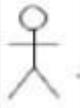
## 2.5 Framework Laravel

Laravel merupakan sebuah kerangka kerja pemrograman yang berbasis open source yang dipakai oleh banyak developer dari seluruh dunia. Kemudahan penggunaan dan dokumentasi yang lengkap menjadi salah satu faktor mengapa Laravel menjadi primadona dalam beberapa tahun terakhir (Mulyadi, 2015). Laravel dapat membantu memaksimalkan penggunaan PHP dalam pengembangan website dengan beberapa fiturnya antara lain template engine, routing dan modularity.

## 2.6 Use Case Diagram

Dalam buku “System Analysis and Design in a Changing World” Use case diagram adalah sebuah kegiatan yang dilakukan oleh sistem, biasanya dalam menanggapi permintaan pengguna sistem (Satzinger et al., 2012). *Use Case Diagram* menunjukkan tiga aspek dari sistem yaitu actor, use case dan sistem atau subsistem boundary (Evi Triandini, 2012). Actor mewakili peran orang, system mewakili alat ketika berkomunikasi dengan *use case*.

**Tabel 2. 1 Simbol Use Case Diagram**

Simbol	Nama	Keterangan
	<i>Actor</i>	Mewakili User

	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya
	<i>Generalization</i>	Hubungan dimana objek anak (Descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor)
	<i>Include</i>	Menspesifikasikan bahwa Use Casesumber secara eksplisit
	<i>Extend</i>	Merupakan perluasan dari Use Case lain jika kondisi atau Syarat terpenuhi

## 2.7 Database

Database atau basis data adalah kumpulan data yang dikelola sedemikian rupa berdasarkan ketentuan tertentu yang saling berhubungan sehingga mudah dalam pengelolaannya. Melalui pengelolaan tersebut pengguna dapat memperoleh kemudahan dalam mencari informasi, menyimpan informasi dan membuang informasi.

## 2.8 Robustness Diagram

*Robustness* merupakan gambar objek dari suatu use case yang tujuannya untuk menyempurnakan use case dan domain model. Robustness memiliki 3 struktur yaitu *Boundary object*, *entity object* dan *controller*.

- a. *Boundary Object* didefinisikan sebagai antarmuka antara sistem dan dunia luar. Boundary biasanya menggambarkan layar atau halaman web.
- b. Entity Object didefinisikan sebagai kelas dari domain model.
- c. Controller menggambarkan penghubung antara boundary object dan entity object.