

BAB II

KAJIAN LITERATUR

2.1 *Unified Process*

Unified Process merupakan salah satu *software development process* yang digunakan untuk menerjemahkan permintaan pengguna ke dalam sistem perangkat lunak. *Unified Process* bersifat *component-based* yang berarti perangkat lunak yang dibangun dibuat dari sejumlah komponen yang diinterkoneksi melalui *interface* yang dibuat dengan baik (Ryan Adiwinata, 2011). Pada *Unified Process*, *Unified Model Language* (UML) digunakan untuk mempersiapkan rancangan dari sistem yang akan dibangun, dengan UML yang dikembangkan secara berkesinambungan. Pengembangan perangkat lunak yang menggunakan *unified process* dilakukan dalam satu atau lebih siklus yang berurutan. Siklus tersebut terbagi menjadi empat fase yaitu *requirement, analysis, design, implementation*, dan *test*. Selanjutnya, *workflow* menentukan urutan aktivitas yang perlu dilakukan dan artifak apa yang harus dihasilkan.

2.2 Sistem Berbasis Website

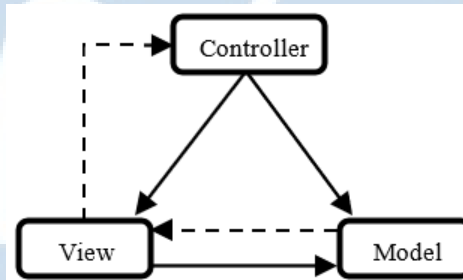
Menurut Alter, sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi. Sedangkan menurut Hall, sistem informasi adalah sebuah rangkaian prosedur formal di mana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai (Kadir, 2014). Kegiatan pemrosesan informasi ini mencakup menyimpan, mengambil, memanipulasi, dan menampilkan informasi. Berdasarkan dua pengertian tersebut dapat disimpulkan bahwa sistem informasi melibatkan prosedur kerja, teknologi informasi, dan manusia dalam kegiatan memproses informasi.

Sistem berbasis web atau bisa juga disebut sebagai aplikasi berbasis web sendiri merupakan sebuah sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis web (Simarmata, 2010). Saat ini sistem berbasis web sudah banyak digunakan di berbagai sektor kehidupan seperti pada dunia pendidikan, perusahaan, pemerintahan, dan lain sebagainya. Suatu sistem berbasis web biasanya dirancang untuk dapat saling terintegrasi dengan sistem lainnya yang ada di suatu instansi sehingga memudahkan instansi tersebut dalam melakukan pengelolaan data.

2.3 Model-View-Controller (MVC)

Mode-View-Controller atau MVC adalah sebuah arsitektur untuk membuat sebuah aplikasi yang memisahkan pengembangan aplikasi berdasarkan komponen utama sebuah aplikasi seperti manipulasi data, *user interface*, dan bagian kontrol aplikasi (Supono, 2018). *Model* adalah kelas yang merepresentasikan atau memodelkan tipe data untuk kemudian mengatur, memanipulasi, dan mengorganisasikan data dari *database* berdasarkan instruksi tertentu.

View bertugas menyajikan informasi kepada pengguna berdasarkan instruksi. Pada dasarnya, halaman web yang di tampilkan adalah bagian dari *view* itu sendiri. Karena itulah kode yang ditulis pada *view* menggunakan kode-kode HTML, CSS, Javascript, dan PHP itu sendiri. Sedangkan *controller* bertugas untuk mengatur dan menginstruksikan apa yang harus ditampilkan oleh pengguna. Jika diibaratkan lalu lintas, *controller* adalah polisi yang mengatur lalu lintas dari permintaan pengguna yang harus dijalankan.



Gambar 1 Diagram Model View Controller

Dalam pemrograman berorientasi objek, MVC adalah suatu metodologi untuk merelasikan antara data dan *user interface* dari aplikasi secara efisien. Metodologi ini awalnya digunakan untuk pengembangan aplikasi desktop. Namun, saat ini metodologi tersebut didopsi untuk pengembangan aplikasi berbasis web, dan hampir semua *framework* untuk pengembangan aplikasi web saat ini mengimplementasikan arsitektur MVC. Tidak hanya pada PHP, tetapi untuk *framework* dengan bahasa pemrograman lain seperti Python dan Ruby juga menerapkan konsep dasar arsitektur yang sama.

2.3.1 Framework CodeIgniter

Framework adalah sebuah alat yang digunakan untuk mempermudah para pengembang aplikasi untuk dapat membangun, membuat, dan mengembangkan

aplikasi berbasis web. Di dalam *framework* tersebut terdapat kumpulan kode pustaka atau biasa disebut *library* yang dapat mempermudah dan mempercepat proses pengembangan aplikasi. Dengan menggunakan *framework*, pengkodean suatu aplikasi akan menjadi terstruktur dan rapih sehingga akan lebih mudah apabila ingin melakukan pengembangan lebih lanjut ke depannya pada sebuah aplikasi yang dibangun.

Sedangkan CodeIgniter sendiri merupakan salah satu jenis web *framework* yang bersifat *open source* dengan arsitektur MVC untuk membangun website dinamis dengan menggunakan bahasa pemrograman PHP (Supono, 2018). Pada awalnya, CodeIgniter dibuat oleh Rick Ellis pada tahun 2006 bersama dengan EllisLab. EllisLab ini berisikan para programmer yang mengembangkan pembuatan *software* dan *tool* untuk para pengembang web. Hingga kemudian hak kepemilikan CodeIgniter diserahkan pada 2014 lalu kepada *British Columbia Institute of Technology* (BCIT) untuk pengembangan lebih lanjut.

Di dalam CodeIgniter sendiri sudah menyediakan kumpulan *library* untuk tugas-tugas yang sering dilakukan dalam aktivitas pengkodean sehingga mudah dalam mengakses *library* yang tersedia di CodeIgniter (Raharjo, 2018). Salah satu keunggulan lain dari *framework* ini yaitu CodeIgniter dapat digunakan secara sebagian maupun keseluruhan. Artinya, pengembang diberikan kebebasan untuk dapat memilih menulis kode-kode tertentu dalam aplikasi menggunakan cara konvensional (tanpa *framework*).

2.4 Digital Signage

Papan Informasi Digital (*Digital Signage*) merupakan salah satu media informasi elektronik yang di dalamnya memuat informasi dan konten-konten seperti gambar, video, dan tulisan. Umumnya *digital signage* dapat digunakan untuk menampilkan informasi seperti antrian elektronik, jadwal transportasi, denah lokasi, dan konten interaktif museum. Sistem *digital signage* modern memungkinkan konten yang ditampilkan lebih beragam, dan terdapat fitur yang dapat menentukan bagaimana, di mana, dan kapan konten tersebut akan ditampilkan.

Pada sistem *digital signage*, pengelola konten telah diprogram untuk dapat menampilkan urutan informasi kepada audiens secara efektif, efisien, dan terarah. Konsep *digital signage* yang didasarkan pada berbagai metode menggunakan layar komputer dan televisi serta jenis perangkat penampil lainnya dengan cara yang

seefisien mungkin dalam menyediakan iklan dan informasi kepada orang-orang di area publik yang ditargetkan (Lundstrom, 2008). Contoh paling sederhana untuk penerapan *digital signage* ini adalah penggunaan videotron yang sering terdapat di pinggir jalan raya, kemudian terdapat juga panel monitor yang terdapat di beberapa gerbong *commuter line*.



Gambar 2 Contoh Digital Signage

STT-NF

Tabel 1 Penelitian Terkait

No.	Nama dan Tahun	Judul	Viewer	Data dan Informasi
1	Agus Mulyana, Muhammad Aria, 2015	Perancangan Digital Signage sebagai Papan Informasi Digital	Mahasiswa dan Dosen Program Studi Sistem Komputer Universitas Komputer Indonesia	Informasi konten (video, gambar, dan running text), informasi skala program studi (jadwal kuliah, UTS, UAS, magang, dan tugas akhir), informasi pencarian ruang kelas, dosen, dan staff jurusan
2	Rizal Panuntun, Adian Fatchur Rochim, Kurniawan Teguh Martono, 2015	Perancangan Papan Informasi Digital Berbasis Web pada Raspberry pi	Mahasiswa Program Studi Teknik Sistem Komputer Universitas Diponegoro	Informasi konten (video, gambar, dan running text), informasi berita
3	Iwan Wijaya, Teguh Sutanto, Anjik Sukmaaji, 2016	Digital Signage Sistem Antrian Elektronik Secara Interaktif Dan Real-Time Monitoring Pada Koperasi Setia Bhakti Wanita	Koperasi Setia Bhakti Wanita	Informasi antrian, informasi konten (gambar, video, dan running text), informasi layanan
4	Khairul Anwar Hafizd, Melda Aolia, 2017	Angkasa Pura Digital Signage pada Bandar udara Syamsudinor Banjarmasin	Pegawai Bandar udara Syamsudinor Banjarmasin	Informasi jadwal kegiatan, informasi laporan kegiatan, data kehadiran pegawai, data pegawai, data galeri