

## BAB II

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Andri dan Tri Sutrisno (2013) dalam penelitiannya yang berjudul “Rancang Bangun Sistem Informasi Pengajuan Judul Tugas Akhir dan Skripsi Berbasis Web *Service*” mengatakan bahwa proses pengajuan judul skripsi dan tugas akhir membutuhkan sistem yang terintegrasi karena selama ini mahasiswa mengajukan judul skripsi dan tugas akhir secara manual. Penelitian didasarkan pada perkembangan teknologi informasi yang begitu pesat, sehingga dapat mempermudah penyampaian informasi yang akurat dan tepat.

Rizky Vickyan Kusuma, et. al. (2011) dalam penelitiannya yang berjudul “Sistem Informasi Manajemen Tugas Akhir dan Kerja Praktek Jurusan Teknik Informatika PENS-ITS Berbasis Web” mengatakan bahwa proses manajemen tugas akhir dan kerja praktek yang dilakukan secara manual sudah tidak relevan dengan tuntutan perbaikan dan kecepatan layanan proses, karena membutuhkan SDM yang banyak dan proses tidak berjalan lebih cepat. Oleh karena itu dibutuhkan sistem yang mampu mengelola proses manajemen tugas akhir dan kerja praktek secara komputerisasi. Sehingga diharapkan proses berjalan lebih cepat, akurat, dan efisien. Sistem dibangun dengan menggunakan Bahasa PHP dan database MySQL.

Indra Adikusuma, et al. (tanpa tahun) dalam penelitiannya yang berjudul “Aplikasi Pengajuan Judul Skripsi Mahasiswa Jurusan Ilmu Komputer Universitas Pakuan Berbasis Web”. Dalam penelitiannya, sistem informasi berbasis *website* dibangun untuk melayani proses pengajuan judul skripsi secara online. Indra Adikusuma, et. al. mengatakan bahwa dengan menggunakan sistem ini dapat mempercepat proses pengolahan data pengajuan judul skripsi mahasiswa dan mempercepat proses validasi penelitian yang sudah disetujui. Dalam membangun sistem, digunakan metode *waterfall* dan implementasinya menggunakan Bahasa pemrograman HTML, PHP, dan CSS.

Dari beberapa penelitian yang telah dilakukan, Bahasa pemrograman PHP dan database MySQL banyak digunakan dalam membangun *website*. Dengan begitu, pada penelitian ini akan dibangun Sistem Pengambilan Ide Skripsi dengan menggunakan Bahasa pemrograman PHP.

Adapun perbedaan yang ada dalam penelitian ini, yaitu terletak pada pemanfaatan fitur yang ada dalam sistem. Pada penelitian sebelumnya, mahasiswa mengajukan judul skripsinya sendiri. Sedangkan, pada penelitian ini dosen menawarkan ide skripsi yang dapat diambil oleh mahasiswa.

## **2.2 Landasan Teori**

### **2.2.1 Website**

*Website* atau situs adalah kumpulan halaman yang saling berkaitan satu sama lain yang digunakan untuk menampilkan konten multimedia berupa teks, gambar, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis (Hidayat, 2010). Hubungan antara satu halaman web dengan halaman web yang lainnya disebut *Hyperlink*. Sedangkan, teks yang dijadikan media penghubung disebut *Hypertext*.

### **2.2.2 Unified Markup Language**

*Unified Modelling Language* (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual (Braun, et. al. 2001). UML juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek (Whitten, et. al. 2004).

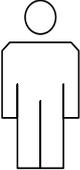
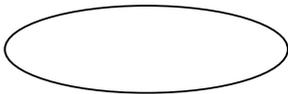
UML yang digunakan dalam pengembangan *website* sistem pengambilan ide skripsi, antara lain:

#### **a. Use Case Diagram**

*Use case diagram* menggambarkan actor, *use case* dan relasinya dengan sistem. *Use case diagram* menggambarkan siapa yang menggunakan sistem

dan apa yang bisa dilakukan dalam sistem. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*. Simbol-simbol dalam *use case diagram* dapat dilihat pada Tabel 2.1.

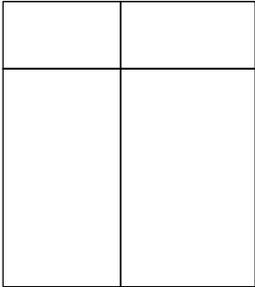
**Tabel 2. 1** Simbol-simbol *use case*

No.	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Suatu peran yang dimainkan oleh pengguna dalam berinteraksi dengan <i>use case</i> .
2.	--<<include>>->	<i>Include</i>	Menspesifikasikan fungsionalitas suatu <i>use case</i> dari <i>use case</i> lainnya.
3.	<--<<extend>>---	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memiliki tambahan fungsional dari <i>use case</i> sumber jika suatu kondisi terpenuhi.
4.	_____	<i>Association</i>	Hubungan antara objek satu dengan objek lainnya.
5.		<i>Use case</i>	Aksi dalam sistem yang dilakukan oleh actor.

b. *Activity Diagram*

*Activity diagram* berisi aktivitas-aktivitas, objek, state, transisi state dan event. *Activity diagram* menggambarkan aktivitas atau alur kerja sistem, bukan apa yang dilakukan aktor. Simbol-simbol dalam *activity diagram* dapat dilihat pada Tabel 2.2.

**Tabel 2. 2** Simbol-simbol *activity diagram*

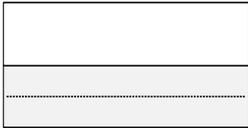
No.	Gambar	Nama	Keterangan
1.		<i>Swimlane</i>	Memperlihatkan siapa yang bertanggung jawab atas setiap langkah dalam <i>activity diagram</i> .
2.		<i>Action</i>	Langkah-langkah yang terjadi antar aliran <i>activity diagram</i> .
3.		<i>Initial node</i>	Awal dari aliran <i>activity diagram</i> .
4.		<i>Final node</i>	Akhir dari aliran <i>activity diagram</i> .

No.	Gambar	Nama	Keterangan
5.		<i>Join node</i>	Menggabungkan beberapa aliran menjadi satu aliran.

c. *Class Diagram*

*Class diagram* menggambarkan struktur statis dari kelas dalam sistem dengan menunjukkan kelas, atribut, operasi, dan hubungan antar objek. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. *Class diagram* terdiri atas 4 elemen, yaitu: nama kelas, atribut, operasi, dan relasi. Simbol-simbol dalam *class diagram* dapat dilihat pada Tabel 2.3.

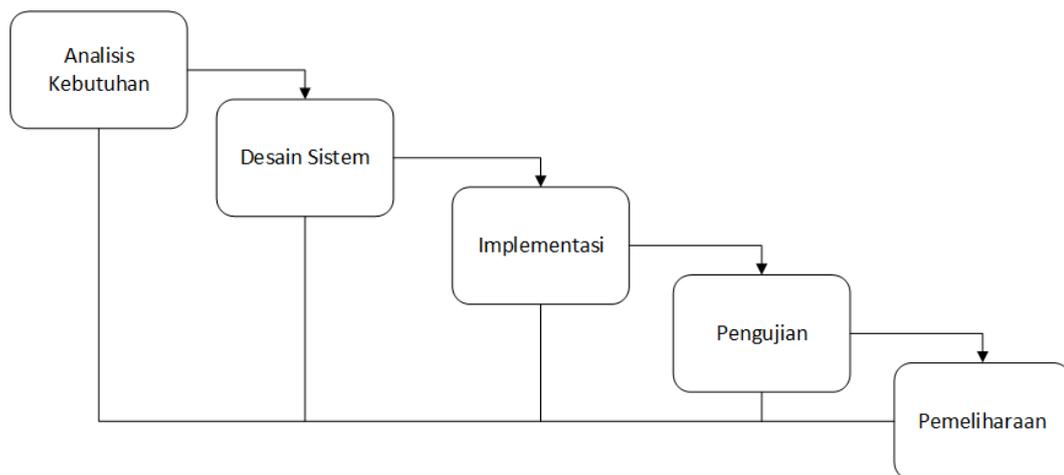
**Tabel 2. 3** Simbol-simbol *class diagram*

No.	Gambar	Nama	Keterangan
1.		<i>Class</i>	Kumpulan objek yang memiliki atribut dan operasi yang sama.
2.		<i>Composition</i>	Hubungan antar kelas, dimana suatu kelas menjadi bagian dari kelas lain.
3.		<i>Association</i>	Hubungan antara suatu objek kelas dengan objek kelas lainnya.

### 2.2.3 Software Development Life Cycle

SDLC (*Software Development Life Cycle*) adalah suatu kerangka yang menggambarkan beberapa kegiatan yang dilakukan melalui beberapa tahap dalam pembuatan sebuah *software* (Fatta, 2007).

Model SDLC yang digunakan dalam pembuatan *website* ini adalah *waterfall*. Model *waterfall* adalah proses pengembangan perangkat lunak tradisional yang paling umum digunakan. Model ini melakukan pendekatan secara sistematis dan sekuensial, sehingga penyelesaian satu set kegiatan menyebabkan dimulainya aktivitas berikutnya. Hal ini disebut *waterfall* karena proses mengalir secara sistematis dari satu tahap ke tahap lainnya dalam mode ke bawah. Secara umum, model ini memiliki lima tahapan yang dapat dilihat pada Gambar 2.1. Tahapan dalam model *waterfall* meliputi analisis kebutuhan, desain sistem, implementasi, pengujian, dan pemeliharaan.



Gambar 2. 1 Tahapan model waterfall

### 2.2.4 HTML

*Hyper Text Markup Language* atau disingkat HTML adalah bahasa pemrograman yang digunakan dalam web sebagai bahasa untuk pertukaran dokumen web (Sibero, 2011).

*Hypertext* adalah teks yang membawa konten halaman web dan informasi pemrograman yang dibutuhkan untuk menampilkan halaman tersebut dan

menghubungkannya ke halaman lain. Hypertext dibentuk dengan *markup language*, yaitu sebuah kumpulan symbol dan kode terstandar yang dapat diinterpretasikan oleh semua *browser*.

*Markup* digunakan untuk menyampaikan dua macam informasi tentang teks atau konten lainnya dalam halaman web, yaitu mengidentifikasi struktur yang dibutuhkan konten dan menentukan penyajian elemen-elemen yang ada.

### 2.2.5 CSS

*Cascading Style Sheet* (CSS) merupakan salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web, sehingga akan lebih terstruktur dan seragam (Slamet, 2009).

CSS menggunakan sebuah aturan *style* untuk menentukan penyajian visual dokumen HTML. Aturan *style* ini diintegrasikan dengan halaman web dengan berbagai cara, antara lain:

- *Internal style*: CSS disimpan di area <head> pada suatu halaman HTML dan *style* hanya digunakan untuk halaman tersebut.
- *External style*: CSS disimpan di file terpisah yang berformat .css dan *style* dapat digunakan untuk beberapa halaman web.
- *Inline style*: *Style* hanya diterapkan pada elemen tertentu dalam sebuah dokumen.

### 2.2.6 PHP

PHP adalah singkatan dari *PHP: Hypertext Preprocessor*, yaitu Bahasa pemrograman *web server-side* yang dapat digunakan secara gratis. PHP merupakan *script* yang digunakan bersamaan dengan HTML yang berada pada server (*server side HTML embedded scripting*). PHP digunakan untuk membuat halaman *website* yang dinamis yang artinya halaman dibuat pada saat diminta oleh klien. Mekanisme ini menyebabkan informasi yang diterima klien selalu *up to date* (Anhar, 2010).

Beberapa kelebihan PHP, antara lain:

- Sederhana, PHP sangat mudah dipelajari karena memiliki banyak fungsi *built in* yang dapat menangani kebutuhan standar pembuatan aplikasi web.
- Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang mudah dipahami memiliki banyak referensi.
- PHP adalah bahasa open source yang dapat digunakan di berbagai sistem operasi. Selain itu, PHP dapat dijalankan secara runtime melalui *console* atau menjalankan perintah-perintah sistem. *Source code* yang tersedia secara gratis memungkinkan para *developer* dalam melakukan perbaikan, pengembangan, dan menemukan bug dalam Bahasa PHP.
- *Web server* yang mendukung PHP sangat banyak, seperti Apache, IIS, Lighttpd, dan Xitami dengan konfigurasi yang mudah.
- PHP juga dilengkapi dengan berbagai macam pendukung lain seperti dukungan langsung untuk beberapa *database*, seperti Oracle, PostgreSQL, MySQL, dan lain-lain.

### 2.2.7 Codeigniter

Menurut Supono dan Vidiandry Putratama (2016), Codeigniter adalah aplikasi *open source* berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP. Codeigniter memudahkan pengembang web untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuat dari awal.

Kelebihan CodeIgniter (CI) dibandingkan dengan *framework* lain:

- Performa yang cepat: CodeIgniter merupakan *framework* yang paling cepat dibanding *framework* lain.
- Konfigurasi yang sedikit (*nearly zero configuration*): Untuk menggunakan codeigniter dengan pengaturan yang standar, hanya perlu mengubah sedikit *file* pada folder *config*.

- Menggunakan konsep MVC: Pengerjaan antara logika dengan *layout* telah dipisahkan, sehingga *programmer* dan *designer* dapat mengerjakan tugas masing-masing dengan mudah.
- Banyak komunitas: Banyaknya komunitas CI memudahkan *developer* berdiskusi satu sama lain.
- Dokumentasi lengkap: Setiap paket instalasi codeigniter sudah disertai *user guide* yang lengkap dan mudah dipahami.

### 2.2.8 XAMPP

Menurut Wicaksono (2008), XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolahan data MySQL di komputer lokal. XAMPP berperan sebagai *web server* pada komputer. XAMPP juga dapat disebut sebagai sebuah *Cpanel server* yang *virtual*, yang dapat membantu melakukan pratinjau sehingga dapat memodifikasi *website* tanpa harus terkoneksi dengan internet.

### 2.2.9 Sublime Text

Menurut Supono dan Vidiandry Putratama (2016), Sublime Text merupakan perangkat lunak *text editor* yang digunakan untuk membuat atau mengedit suatu aplikasi. Sublime Text mempunyai fitur plugin tambahan yang memudahkan programmer. Selain itu, Sublime Text merupakan IDE yang ringan, cepat dalam menyimpan dan membuka file. Sublime text juga memiliki desain terkesan elegan untuk sebuah *syntax editor*.