

## BAB 3 PERANCANGAN SISTEM

Perancangan sistem bertujuan untuk memberikan gambaran secara umum tentang sistem yang akan dibuat. Rancangan sistem ini secara umum mengidentifikasi komponen-komponen sistem yang akan digambarkan dengan rinci. Teknik perancangan sistem dimulai dari analisa kebutuhan (*requirement*) sistem, perancangan arsitektural sistem dan perancangan detil sistem. Metode pemodelan dalam perancangan ini menggunakan *Unified Modelling Language* (UML) yang diantaranya terdiri dari *Use-Case Diagram* dan *Activity Diagram* seperti yang telah dijelaskan pada Bab 2 Landasan Teori.

### 3.1 Analisa Requirement Sistem

Pada bagian ini berisi penjelasan secara terperinci tentang segala sesuatu yang dibutuhkan untuk membuat sistem aplikasi Portal Pembelajaran *Object Oriented Programming* (PP-OOP).

#### 3.1.1 Requirement State dan Mode

Pada sistem ini terdapat dua mode yang didasarkan pada hak akses *user* (*multi-user*) terhadap fungsi-fungsi yang disediakan oleh aplikasi PP-OOP. Kedua mode itu adalah mode *user* umum dan mode administrator/dosen. *Requirement* mode dan state akan ditunjukkan pada Tabel 3.1.

**Tabel 3.1 Requirement State dan Mode**

Mode	State	
	Fungsi Aktif	Fungsi Tak Aktif
User umum	<ul style="list-style-type: none"> <li>- Melihat materi pembelajaran</li> <li>- <i>Download</i> materi pembelajaran</li> </ul>	<ul style="list-style-type: none"> <li>- Pengolahan materi pembelajaran</li> <li>- <i>Upload</i> materi pembelajaran</li> </ul>
Administrator / Dosen	<ul style="list-style-type: none"> <li>- Melihat materi pembelajaran</li> <li>- <i>Download</i> materi pembelajaran</li> <li>- Pengolahan materi pembelajaran</li> <li>- <i>Upload</i> materi pembelajaran</li> </ul>	- Tidak ada

### 3.1.2 *Requirement* Kemampuan Perangkat Lunak

Berikut ini *requirement* kemampuan sistem PP-OOP berbasis Ruby on Rails dilihat dari sudut pandang pengguna sistem.

- a. *User* Umum
  - Aplikasi dapat menampilkan isi dari materi pembelajaran OOP.
  - Aplikasi dapat menyediakan fasilitas *download* materi pembelajaran.
- b. Dosen / Admin
  - Aplikasi dapat menyediakan fasilitas *login*.
  - Aplikasi dapat menyediakan form untuk pengolahan(*input*, *edit*, dan *delete*) materi pembelajaran.
  - Aplikasi dapat menyediakan fasilitas *upload* materi pembelajaran.

## 3.2 Perancangan

### 3.2.1 Cakupan Perancangan Perangkat Lunak

Perancangan yang akan dibuat pada tugas akhir ini adalah membuat suatu sistem portal pembelajaran dimana *user* umum dapat mempelajari materi pembelajaran yang dibutuhkan dengan efektif dan efisien. Sasaran yang ingin dicapai dalam pembuatan perangkat lunak pada tugas akhir ini adalah adanya suatu sistem yang dapat menangani pengolahan materi yang akan disajikan dalam bentuk portal pembelajaran.

Ruang lingkup proses dari perangkat lunak ini yaitu:

- a. Penyampaian materi

Aplikasi yang dibangun dapat menyajikan berbagai macam materi pembelajaran. Selain itu, aplikasi juga dapat menyajikan informasi tambahan lainnya yang terkait dengan materi yang sedang disajikan dalam bentuk halaman web dan *podcasting*.
- b. Pengolahan data materi

Aplikasi yang dibangun akan menangani pengolahan data materi (baik *create*, *update*, *delete* dan *upload* materi) yang akan disajikan dalam bentuk web.

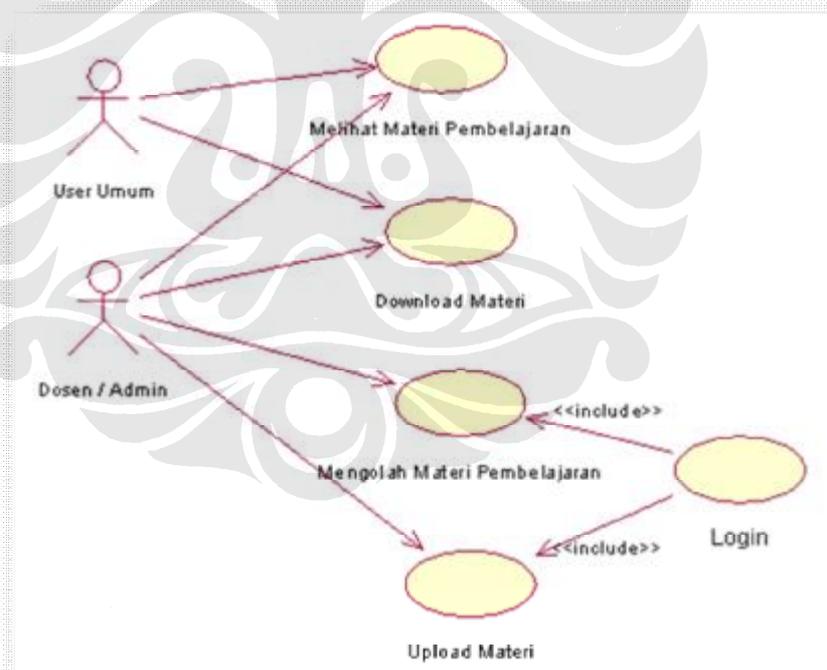
### 3.2.2 Batasan Perancangan

Beberapa batasan yang mempengaruhi perancangan aplikasi PP-OOP berbasis Ruby on Rails ini adalah sebagai berikut:

- Platform Sistem Operasi: Microsoft Windows 9x atau lebih dan Linux
- Database Engine : MySQL 5.0
- Web Server : Webrick
- Web Browser : Mozilla FireFox v.5.0, IE (Internet Explorer) v.7.0, Opera dan Safari versi terbaru
- IDE : RadRails (Aptana)
- Bahasa Pemrograman : Ruby on Rails (Ruby 1.8.6, Rails 2.1.0)

### 3.2.3 Struktur dan Perilaku Sistem dari Sudut Pandang *User*

Struktur dan perilaku sistem dari sudut pandang *user* akan dimodelkan dengan menggunakan *Use-Case Diagram*. *Use-Case Diagram* untuk sistem PP-OOP ini ditunjukkan pada Gambar 3.1.



**Gambar 3.1** *Use-Case Diagram*

*Use-Case Diagram* adalah deskripsi dari fungsionalitas yang dimiliki oleh sistem yang terdiri dari *use case-use case*, aktor dan hubungan interaksinya. *Use-Case*

digambarkan sebagai suatu cara pandang terhadap sistem dilihat dari perspektif aktor.

Penjelasan Use-Case Diagram:

Dari *Use-Case Diagram* pada Gambar 3.1, diketahui dalam perancangan ini memiliki dua aktor yaitu *User* Umum dan Dosen/Admin.

a. *User* Umum

Target dari aktor *User* Umum ini khususnya adalah mahasiswa, umumnya adalah masyarakat umum baik dalam maupun luar negeri yang ingin mengetahui lebih banyak lagi mengenai materi yang diberikan. *User* Umum memiliki dua buah *Use-Case* yaitu *Melihat Materi Pembelajaran* dan *Download Materi*. Pada *Use-Case* *Melihat Materi Pembelajaran*, *User* Umum akan mendapatkan sejumlah materi yang diberikan sesuai dengan pilihannya. Disediakan juga fasilitas untuk *User* Umum jika ingin melakukan *download* materi yang diberikan dalam halaman aplikasi.

Berikut penjelasan dari setiap *Use-Case* yang terhubung dengan *User* Umum maupun Dosen/Admin:

**Tabel 3.2** Penjelasan *Use-Case* *Melihat Materi Pembelajaran*

Nama	Melihat Materi Pembelajaran
Deskripsi Singkat	Menyajikan materi pembelajaran yang telah di- <i>request</i> oleh <i>user</i>
Aktor	<i>User</i> , Dosen/Admin
Kondisi awal	Aplikasi sudah dijalankan
Kondisi akhir	Materi pembelajaran ditampilkan berdasarkan materi pilihan <i>user</i>

**Tabel 3.3** Penjelasan *Use-Case* *Download Materi*

Nama	<i>Download Materi</i>
Deskripsi Singkat	<i>User</i> mengajukan permintaan untuk <i>download</i> materi yang diinginkan
Aktor	<i>User</i> , Dosen/Admin
Kondisi awal	Materi yang akan di- <i>download</i> sudah dipilih

Kondisi akhir	<i>User</i> berhasil melakukan <i>download</i> materi
---------------	---

b. Dosen / Admin

Target dari aktor Dosen/Admin adalah khususnya dosen dari suatu universitas. Dosen/Admin terhubung ke semua *Use-Case* yang ada dalam sistem ini karena sewaktu-waktu Dosen/Admin dapat bertindak sebagai *User* Umum maupun sebagai Dosen yang menyediakan materi untuk dipublikasikan. Ada tiga buah *Use-Case* yang tidak terkait dengan *User* Umum, yaitu Pengolahan Data Materi, *Upload* Materi dan *Login*. *Login* dibutuhkan untuk melakukan verifikasi terhadap pengolah suatu materi tertentu sehingga hanya pemilik pengolah materi yang dapat melakukan perubahan terhadap data materi.

Berikut penjelasan dari setiap *Use-Case* yang terhubung dengan Dosen/Admin dan tidak terhubung dengan *User* Umum:

**Tabel 3.4** Penjelasan *Use-Case* Pengolahan Data Materi

Nama	Pengolahan Data Materi
Deskripsi Singkat	Dosen melakukan pengolahan data materi berupa fungsi <i>create</i> , <i>edit (update)</i> , dan <i>delete</i> .
Aktor	Dosen / Admin
Kondisi awal	Materi belum ada, atau sudah ada
Kondisi akhir	Materi pembelajaran telah di- <i>update</i>

**Tabel 3.5** Penjelasan *Use-Case Upload* Materi

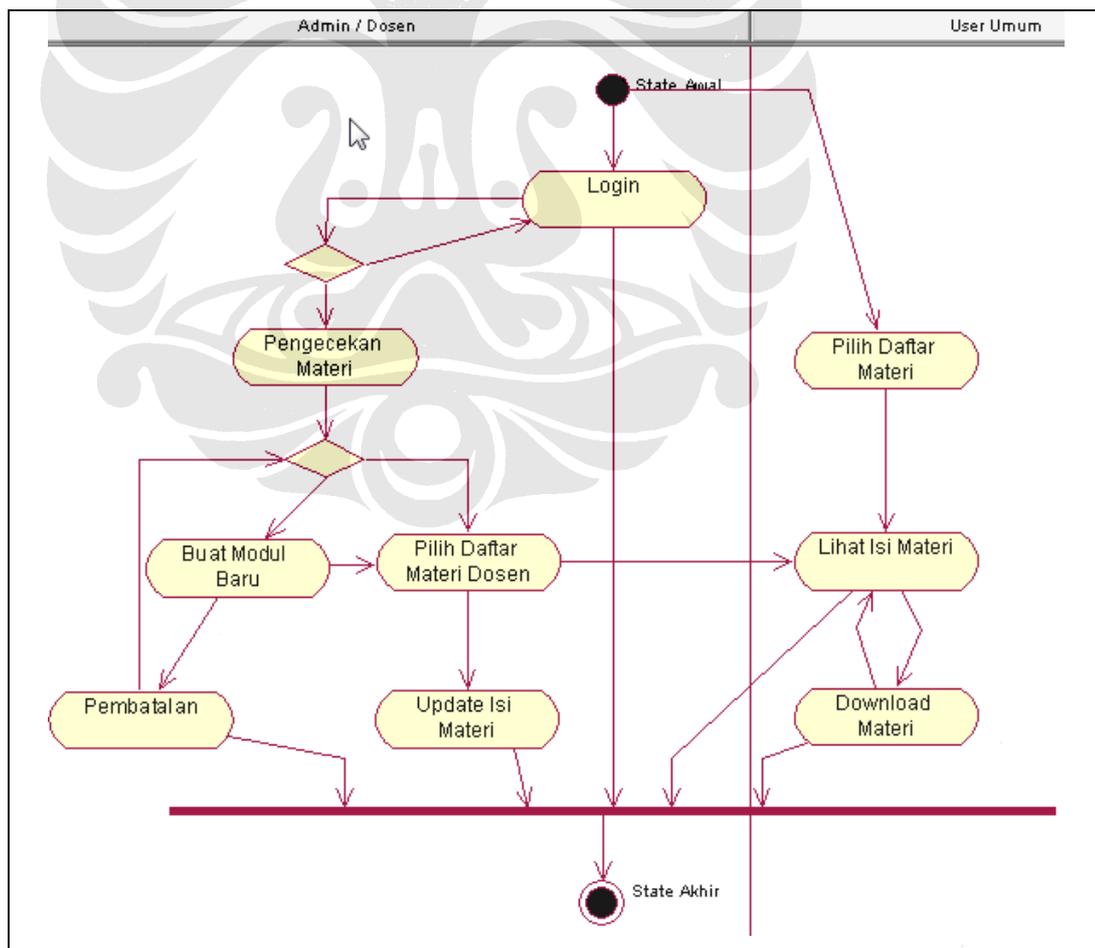
Nama	<i>Upload</i> Materi
Deskripsi Singkat	Dosen melakukan <i>upload</i> materi
Aktor	Dosen / Admin
Kondisi awal	Materi belum ter- <i>upload</i>
Kondisi akhir	Materi berhasil di- <i>upload</i>

**Tabel 3.6** Penjelasan *Use-Case Login*

Nama	<i>Login</i>
Deskripsi Singkat	Dosen / Admin melakukan <i>login</i> untuk melakukan pengolahan data materi
Aktor	Dosen / Admin
Kondisi awal	<i>Username</i> dan <i>password</i> sudah diisi pada kolom <i>login</i>
Kondisi akhir	Dosen / Admin berhasil melakukan <i>login</i>

### 3.2.4 Perancangan Detil Sistem Portal

Perancangan detil sistem portal digambarkan dengan menggunakan pemodelan *Activity Diagram*. *Activity Diagram* adalah salah satu diagram dari UML untuk menggambarkan aspek dinamis dari sebuah sistem. Diagram ini sebenarnya merupakan flowchart yang menggambarkan aliran kontrol dari satu aktivitas ke aktivitas lain. *Activity Diagram* untuk sistem ini dapat terlihat pada Gambar 3.2.

**Gambar 3.2** *Activity Diagram*

### Penjelasan Activity Diagram

Dari Gambar 3.2 dapat dilihat bahwa saat State Awal terdapat dua buah jalur yaitu ke *Activity Login* dan ke *Activity Pilih Daftar Materi*. Hal ini disebabkan pada diagram di atas dilakukan pemisahan hak akses. Untuk pemisahan hak akses ini dibedakan menggunakan SwimLane. Terdapat dua buah SwimLane pada diagram di atas berdasarkan *user* yang terlibat, yaitu Admin/Dosen dan *User Umum*.

#### a. Swimlane Dosen/Admin

Pada Swimlane Dosen/Admin, saat state dimulai, aplikasi segera meminta *user* dan password untuk melakukan *login* terhadap aplikasi. Jika *user* yang dimasukkan tidak ditemukan atau *user* ditemukan namun password yang dimasukkan salah, maka aplikasi akan meminta pengguna aplikasi untuk mengulangi melakukan *login* atau memilih keluar dari aplikasi. Jika pengguna aplikasi telah berhasil masuk ke dalam aplikasi menggunakan *user* dan password yang benar, maka aplikasi akan melakukan pengecekan terhadap *database* mengenai ketersediaan materi. Jika di dalam *database* belum terdapat materi, maka pengguna diminta untuk melakukan pengajuan materi baru. Jika telah terdapat materi pada saat pengguna melakukan *login*, maka aplikasi akan mengizinkan pengguna untuk memilih antara membuat materi baru kembali atau memilih untuk membuka materi yang sudah ada. Jika pengguna memilih untuk membuka materi yang sudah ada, maka aplikasi akan memberikan layanan kepada pengguna untuk memilih apakah akan menampilkan isi materi atau melakukan pengolahan terhadap materi yang sudah ada. Pengolahan ini dapat berupa edit dan delete isi materi. Jika pengguna memilih untuk menampilkan isi materi, maka tampilan akan masuk ke tampilan yang mirip dengan SwimLine *User Umum* namun terdapat link untuk melakukan perubahan terhadap isi materi tersebut.

#### b. Swimline *User Umum*

Pada swimline ini, saat state dimulai, aplikasi akan mengizinkan pengguna untuk memilih topik materi yang diinginkan. Jika telah dipilih, maka aplikasi akan menampilkan seluruh isi dari materi yang ada. Sewaktu-waktu pengguna dapat melakukan *download* terhadap materi yang disajikan.