



UNIVERSITAS INDONESIA

**IMPLEMENTASI PORTAL PEMBELAJARAN
OBJECT ORIENTED PROGRAMMING
BERBASIS *RUBY ON RAILS* DAN *PODCASTING***

SKRIPSI

ARDIATI UTAMI S.

0606042286

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
DESEMBER 2008**



UNIVERSITAS INDONESIA

**IMPLEMENTASI PORTAL PEMBELAJARAN
OBJECT ORIENTED PROGRAMMING
BERBASIS *RUBY ON RAILS* DAN *PODCASTING***

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana

ARDIATI UTAMI S.

0606042286

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
DESEMBER 2008**

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : ARDIATI UTAMI S.

NPM : 0606042286

Tanda Tangan :

Tanggal : 19 Desember 2008



HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : ARDIATI UTAMI S.

NPM : 0606042286

Program Studi : TEKNIK ELEKTRO EKSTENSI

Judul Skripsi : IMPLEMENTASI PORTAL PEMBELAJARAN

OBJECT ORIENTED PROGRAMMING BERBASIS *RUBY ON RAILS* DAN
PODCASTING

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Ekstensi Teknik Elektro Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Dr. Ir. Riri Fitri Sari MM, M.Sc (.....)

Penguji : Prof. Dr. Ir. Bagio Budiarjo M.Sc (.....)

Penguji : Dr. Ir. Kalamullah Ramli M.Eng (.....)

Ditetapkan di : Depok

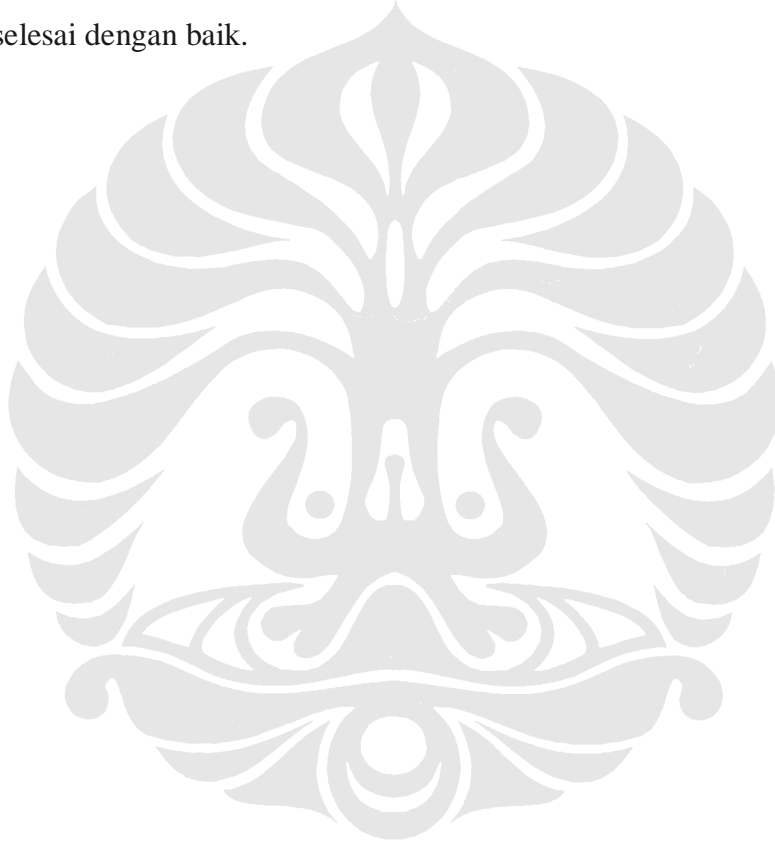
Tanggal : 19 Desember 2008

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada:

Dr. Ir. Riri Fitri Sari MM, M.Sc

selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberi pengarahan, diskusi, dan bimbingan serta persetujuan sehingga skripsi ini dapat selesai dengan baik.



**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : ARDIATI UTAMI S.
NPM : 0606042286
Program Studi : Teknik Elektro Ekstensi
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul:

IMPLEMENTASI PORTAL PEMBELAJARAN *OBJECT ORIENTED PROGRAMMING* BERBASIS *RUBY ON RAILS* DAN *PODCASTING*.

Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok, Jawa Barat

Pada Tanggal : 19 Desember 2008

Yang Menyatakan

(**ARDIATI UTAMI S.**)

ABSTRAK

Nama : ARDIATI UTAMI S.
Program Studi : Teknik Elektro
Judul : IMPLEMENTASI PORTAL PEMBELAJARAN *OBJECT ORIENTED PROGRAMMING* BERBASIS *RUBY ON RAILS* DAN *PODCASTING*

Teknologi informasi menjadi pendorong utama bagi perubahan dan pengembangan di bidang pendidikan. Salah satu implementasi teknologi informasi adalah portal yang menyediakan akses khusus mengenai suatu materi yang disajikan secara *online* di Internet. Pada tugas akhir ini dibuat suatu portal yang menyajikan materi pembelajaran dengan studi kasus *Object Oriented Software Engineering* menggunakan PHP.

Dalam implementasinya, portal ini menggunakan Ruby on Rails (RoR) sebagai bahasa pemrograman dan *framework*-nya. RoR merupakan bahasa pemrograman yang berjalan pada *framework* Rails. RoR telah menerapkan konsep pemrograman berorientasi object sehingga mempermudah pemeliharaan dan pengembangan aplikasi portal pembelajaran selanjutnya.

Pada tugas akhir ini, materi pembelajaran dikemas dalam bentuk tulisan (teks), *slide* presentasi, bahan pendukung materi, serta video pembelajaran. Video pembelajaran ditujukan agar dapat diimplementasikan menggunakan alat multimedia sejenis Zune/iPod yang dapat menjadi standar pembelajaran dengan *podcasting*. Dalam membuat video digunakan *software* pendukung seperti Total Video Converter, Ulead Video Studio, dan iTunes untuk mengolah video agar dapat didukung oleh media *player*.

Aplikasi portal pembelajaran ini kemudian diuji coba oleh sepuluh orang pengguna sistem. Hal-hal yang diujikan diantaranya adalah pengetahuan penguji tentang bahasa pemrograman, hasil aplikasi dilihat dari karakteristik sistem, dan mengenai pendapat umum penguji terhadap teknologi yang digunakan dalam aplikasi portal pembelajaran ini. Dari hasil pengujian didapatkan informasi bahwa pengetahuan mengenai Ruby on Rails masih sangat sedikit meskipun konsep *Object Oriented Programming* sudah banyak dipahami. Tetapi untuk pengembangan ke depan, sistem portal berbasis *Ruby on Rails* dan *podcasting* ini layak untuk dikembangkan lebih lanjut.

Kata kunci:

Portal, *Ruby on Rails*, *Podcast*, *Zune/iPod*.

ABSTRACT

Name : ARDIATI UTAMI S.
Study Program : Electrical Engineering
Title : IMPLEMENTATION OF OBJECT ORIENTED
PROGRAMMING LEARNING PORTAL BASED ON
RUBY ON RAILS AND PODCASTING

Information technology becomes main factor for regulation and development of education. One of the information technology implementation is portal technology which is provide particular access about particular matery presented online on the Internet. In this Final Project created a portal which provide learning materials with Object Oriented Software Engineering case study with PHP.

In the implementation of this portal, we use Ruby on Rails (RoR) as programming language and framework. Ruby is a programming language which is run on Rails framework. RoR has implemented object oriented programming which make it easier to maintain and develop for next learning portal application.

In this Final Project, learning material has been packages in the format of text, presentation slide, learning material support, and video content. Video content is designated to be implemented using multimedia devices such as Zune/iPod which is one of the podcasting standar learning. We use video converter to make video such as Total Video Converter, Ulead Video Studio, and iTunes to manage video to be supported by media player.

This learning portal application subsequently tested by ten users. Some features such as basic knowledge about programming language and application result from the characteristic system are tested by user. And finally, general opinion about the technology that has been used on the application. From the testing result show that user knowledge about Ruby on Rails still very low though concept about Object Oriented Programming have comprehended by many user. However, in the case of forward development, this portal system which based on Ruby on Rails and podcasting competent to be developed furthermore. .

Keywords:

Portal, Ruby on Rails, Podcast, Zune/iPod

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN	iii
UCAPAN TERIMA KASIH	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
DAFTAR SINGKATAN	xi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Penulisan	2
1.3 Batasan Masalah	2
1.4 Metodologi Penelitian	3
1.5 Sistematika Penelitian	3
1.6 Sistematika Penulisan	3
BAB 2 LANDASAN TEORI.....	5
2.1 Portal Pembelajaran	5
2.1.1 Definisi dan Tipe-tipe Portal.....	5
2.1.1.1 Portal <i>Corporate/Enterprise</i> (Intranet)	5
2.1.1.2 Portal <i>e-Business</i> (Extranet)	6
2.1.1.3 Portal Personal (WAP)	6
2.1.2 Komponen Teknik Portal.....	7
2.1.3 Portal dan Infrastruktur.....	8
2.1.4 Portal dan Keamanan.....	8
2.2 Ruby on Rails	12
2.2.1 Pemrograman Ruby	12
2.2.2 <i>Framework</i> Ruby on Rails (RoR)	13
2.2.3 Arsitektur Rails	14
2.2.3.1 <i>Model, View, dan Controller</i>	14
2.2.3.2 Prinsip Pengembangan Aplikasi RoR[5]	15
2.2.3.3 Komponen Ruby on Rails	17
2.2.3.4 Struktur Direktori Aplikasi Rails	19
2.2.4 Radrails	19
2.3 Web v2.0	20
2.3.1 Definisi AJAX.....	20
2.3.1.1 Model Tradisional.....	21
2.3.1.2 Model AJAX	22
2.3.2 Teknologi di balik AJAX.....	23
2.3.3 Teknik Penerapan AJAX	25
2.3.3.1 Teknik Hidden Frame dan IFrame	25
2.3.3.2 Teknik Cache Control (Cookie).....	26

2.3.3.3 Teknik HTTP Request	26
2.3.4 Kelebihan dan Kekurangan AJAX.....	26
2.3.5 Keistimewaan AJAX.....	28
2.4 Alat-alat Pemodelan.....	28
2.4.1 <i>Use-Case Diagram</i>	28
2.4.2 <i>Activity Diagram</i>	29
BAB 3 PERANCANGAN SISTEM.....	30
3.1 Analisa <i>Requirement</i> Sistem.....	30
3.1.1 <i>Requirement State</i> dan <i>Mode</i>	30
3.1.2 <i>Requirement</i> Kemampuan Perangkat Lunak	31
3.2 Perancangan.....	31
3.2.1 Cakupan Perancangan Perangkat Lunak	31
3.2.2 Batasan Perancangan.....	32
3.2.3 Struktur dan Perilaku Sistem dari Sudut Pandang <i>User</i>	32
3.2.4 Perancangan Detil Sistem Portal.....	35
BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM.....	37
4.1 Implementasi	37
4.1.1 <i>Prototyping</i>	37
4.1.2 <i>Graphical User Interface (GUI)</i>	38
4.1.2.1 Halaman Utama untuk <i>User</i> Umum	38
4.1.2.2 Halaman Materi untuk <i>User</i> Umum	39
4.1.2.3 Halaman Utama <i>Login Admin</i>	39
4.1.2.4 Halaman Utama untuk <i>User Admin/Dosen</i>	40
4.1.2.5 Halaman Materi untuk <i>User Admin/Dosen</i>	41
4.1.3 Pengembangan Konten <i>e-Learning</i>	43
4.1.4 Pemanfaatan Teknologi Zune/iPod sebagai Sarana Penunjang Pembelajaran.....	44
4.2 Pengujian Aplikasi PP-OOP oleh <i>User</i>	46
4.2.1 Kategori Tanggapan terhadap Bahasa Pemrograman	47
4.2.2 Kategori Tanggapan terhadap Karakteristik Sistem.....	48
4.2.3 Kategori Pendapat Umum <i>User</i>	50
BAB 5 KESIMPULAN.....	52
DAFTAR ACUAN.....	53
DAFTAR REFERENSI.....	54

DAFTAR TABEL

Tabel 2.1	Perbandingan Fitur IDE	19
Tabel 2.2	Teknik-teknik dalam AJAX beserta kelebihan dan kekurangannya	26
Tabel 2.3	Komponen <i>Use-Case Diagram</i>	28
Tabel 2.4	Komponen <i>Activity Diagram</i>	29
Tabel 3.1	<i>Requirement State</i> dan <i>Mode</i>	30
Tabel 3.2	Penjelasan <i>Use-Case</i> Melihat Materi Pembelajaran.....	33
Tabel 3.3	Penjelasan <i>Use-Case Download</i> Materi	33
Tabel 3.4	Penjelasan <i>Use-Case</i> Pengolahan Data Materi	34
Tabel 3.5	Penjelasan <i>Use-Case Upload</i> Materi.....	34
Tabel 3.6	Penjelasan <i>Use-Case Login</i>	35
Tabel 4.1	Data Tanggapan Penguji terhadap Bahasa Pemrograman	47
Tabel 4.2	Hasil Pengolahan Data Berdasarkan Bahasa Pemrograman.....	47
Tabel 4.3	Data Tanggapan Penguji terhadap Karakteristik Sistem	48
Tabel 4.4	Hasil Pengolahan Data Berdasarkan Bahasa Pemrograman.....	49
Tabel 4.5	Data Tanggapan Umum Aplikasi dan Teknologi.....	50
Tabel 4.6	Hasil Pengolahan Data Berdasarkan Pendapat Umum.....	50

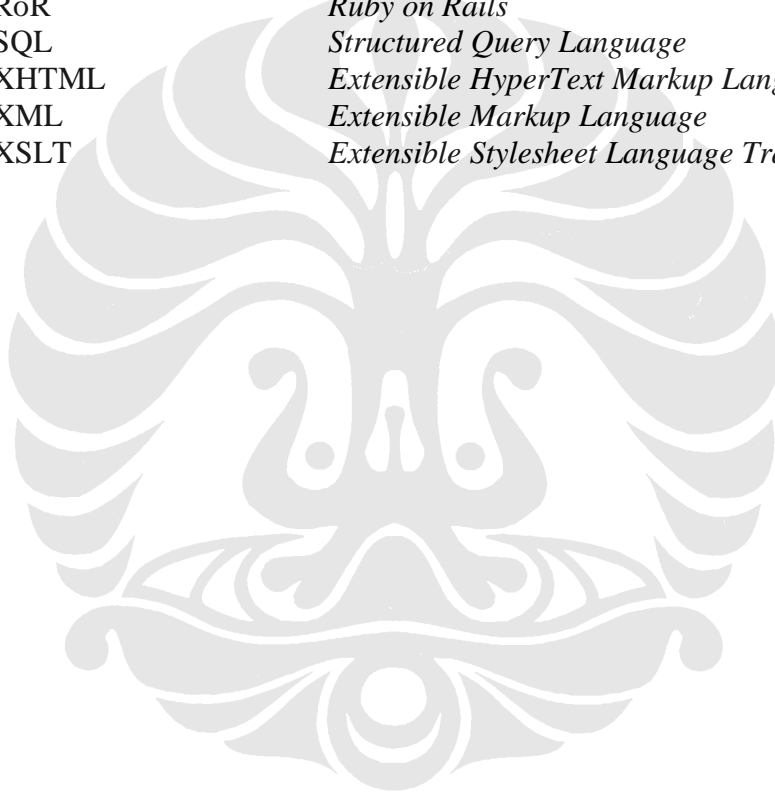


DAFTAR GAMBAR

Gambar 2.1	Arsitektur Model-View-Controller	14
Gambar 2.2	Komponen Ruby on Rails	17
Gambar 2.3	Diagram Ruby on Rails[5].....	18
Gambar 2.4	Struktur Direktori Rails	19
Gambar 2.5	Arsitektur Model Tradisional	21
Gambar 2.6	Arsitektur Model AJAX	22
Gambar 2.7	Teknologi di balik AJAX	23
Gambar 2.8	Teknik Hidden Frame AJAX.....	25
Gambar 3.1	<i>Use-Case Diagram</i>	32
Gambar 3.2	<i>Activity Diagram</i>	35
Gambar 4.1	Halaman Utama untuk <i>User</i> Umum.....	38
Gambar 4.2	Halaman Materi untuk <i>User</i> Umum.....	39
Gambar 4.3	Halaman Utama <i>Login Admin</i>	40
Gambar 4.4	Halaman Utama untuk <i>User Admin/Dosen</i>	41
Gambar 4.5	Halaman Materi untuk <i>User Admin/Dosen</i>	42
Gambar 4.6	Grafik Tanggapan Berdasarkan Bahasa Pemrograman.....	47
Gambar 4.7	Grafik Tanggapan Berdasarkan Karakteristik Sistem.....	49
Gambar 4.8	Grafik Tanggapan Berdasarkan Pendapat Umum	51

DAFTAR SINGKATAN

AJAX	<i>Asynchronous JavaScript And XML</i>
API	<i>Application Program Interface</i>
CSS	<i>Cascading Style Sheets</i>
DOM	<i>Document Object Model</i>
HTML	<i>HyperText Markup Language</i>
JSON	<i>JavaScript Object Notation</i>
MVC	<i>Model View Controller</i>
OOP	<i>Object Oriented Programming</i>
OO-SE	<i>Object Oriented Software Engineering</i>
PP-OOP	<i>Portal Pembelajaran – Object Oriented Programming</i>
RoR	<i>Ruby on Rails</i>
SQL	<i>Structured Query Language</i>
XHTML	<i>Extensible HyperText Markup Language</i>
XML	<i>Extensible Markup Language</i>
XSLT	<i>Extensible Stylesheet Language Transformations</i>



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Di dalam era global sekarang ini, teknologi informasi menjadi pendorong utama dari perubahan dan pengembangan di bidang pendidikan dan ekonomi dunia. Oleh karena itu, semua negara di dunia berlomba untuk menguasai teknologi informasi ini agar tidak tertinggal dari negara lain atau ditinggal oleh keadaan. Berbagai negara tetangga juga sudah memanfaatkan teknologi informasi tersebut agar pekerjaan mereka menjadi lebih efektif dan efisien, serta dapat berhubungan akrab dengan dunia luar.

Dalam menguasai teknologi informasi ini, media web memegang peranan penting karena merupakan salah satu pendukung teknologi informasi tersebut. Untuk mengetahui seluk beluk dunia teknologi informasi, agar nantinya teknologi tersebut benar-benar dapat diterapkan secara benar, dibutuhkan wawasan yang luas untuk menguasai web tersebut.

Kemajuan teknologi khususnya Internet membawa pengaruh terhadap perkembangan teknik-teknik dalam membangun halaman web. Awalnya, web bersifat statis yang artinya isi web tidak dapat berubah ketika diakses pada waktu yang berbeda. Kemudian, muncul web dinamis yang dapat menampilkan isi yang berbeda ketika mengakses pada waktu yang berbeda pula.

Pada dasarnya pemrograman web dibagi menjadi dua kelompok besar, yaitu pemrograman di sisi klien dan pemrograman di sisi *server*. Skrip pemrograman di sisi klien yang populer adalah HTML, CSS, JavaScript, Jscript, VBScript, sedangkan pemrograman di sisi *server* meliputi PHP, ASP, JavaServelet, SGI, Perl, dan Ruby on Rails.

Kemunculan skrip-skrip tersebut membawa kemajuan yang istimewa terhadap teknik pemrograman web. Namun, terkait dengan kebutuhan pengguna yang menginginkan aplikasi web yang lebih interaktif, lebih cepat, dan mudah digunakan seperti aplikasi *desktop*, akhir-akhir ini muncullah teknik pemrograman

web yang disebut AJAX.

Asynchronous JavaScript And XML (AJAX) adalah suatu teknik pemrograman berbasis web untuk menciptakan aplikasi web lebih interaktif dan dinamis. Tujuannya adalah memindahkan sebagian besar interaksi pada komputer *web server*, melakukan pertukaran data dengan *server* di belakang layar, sehingga halaman web tidak harus dibaca ulang secara keseluruhan setiap kali seorang pengguna melakukan perubahan.

1.2 Tujuan Penulisan

Tujuan dari tugas akhir ini adalah:

- Membuat portal pembelajaran berbasis web dengan memanfaatkan teknik bahasa pemrograman *Ruby on Rails*.
- Membuat metode pembelajaran (*e-learning*) dengan memanfaatkan alat multimedia *portable* sebagai fitur *podcasting* pada portal pembelajaran.
- Membuat contoh materi pembelajaran dengan judul *Object Oriented Software Engineering* menggunakan PHP 5 dan mengimplementasikannya pada portal pembelajaran.

1.3 Batasan Masalah

Pada tugas akhir ini, ruang lingkup proses terbatas pada:

- Aplikasi yang dibangun dapat menyajikan berbagai macam materi pembelajaran. Tetapi dalam tugas akhir ini hanya akan membahas satu contoh materi pembelajaran dengan judul *Object Oriented Software Engineering* menggunakan PHP 5.
- Aplikasi juga dapat menyajikan informasi tambahan lainnya yang terkait dengan materi yang sedang disajikan dalam bentuk halaman web dan *podcasting* yang dapat di-*download* dan dijalankan menggunakan alat multimedia Zune.

- Aplikasi yang dibangun akan menangani pengolahan data materi (baik *create*, *update*, *delete*, dan *upload* materi) yang akan disajikan dalam bentuk web yang dibangun menggunakan bahasa pemrograman *Ruby on Rails*.

1.4 Metodologi Penelitian

Penelitian ini dimulai dengan studi literatur, perancangan portal pembelajaran menggunakan Ruby on Rails, perancangan konten materi pembelajaran, implementasi dari perancangan yang dibuat, dan evaluasi hasil kerja.

1.5 Sistematika Penelitian

Hasil implementasi tugas akhir ini dievaluasi oleh pengguna dalam hal pengetahuan tentang bahasa pemrograman, karakteristik sistem aplikasi portal pembelajaran, tanggapan terhadap teknologi yang digunakan, dan pengembangan sistem aplikasi portal pembelajaran di masa mendatang. Pada tugas akhir ini juga dibuat konten tentang materi "*Object Oriented Software Engineering* menggunakan PHP 5" dan video pembelajaran materi tersebut. Video pembelajaran ini akan ditampilkan pada portal pembelajaran sebagai bahan *podcasting* untuk diimplementasikan pada Microsoft Zune.

1.6 Sistematika Penulisan

Pada penulisan tugas akhir ini dibagi menjadi lima bagian, yaitu:

BAB I PENDAHULUAN

Bagian ini berisi latar belakang masalah, tujuan penelitian, batasan masalah, metodologi penelitian, sistematika penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bagian ini berisi penjelasan tentang konsep dan prinsip dasar yang diperlukan untuk memecahkan permasalahan dalam tugas akhir ini.

BAB III PERANCANGAN SISTEM

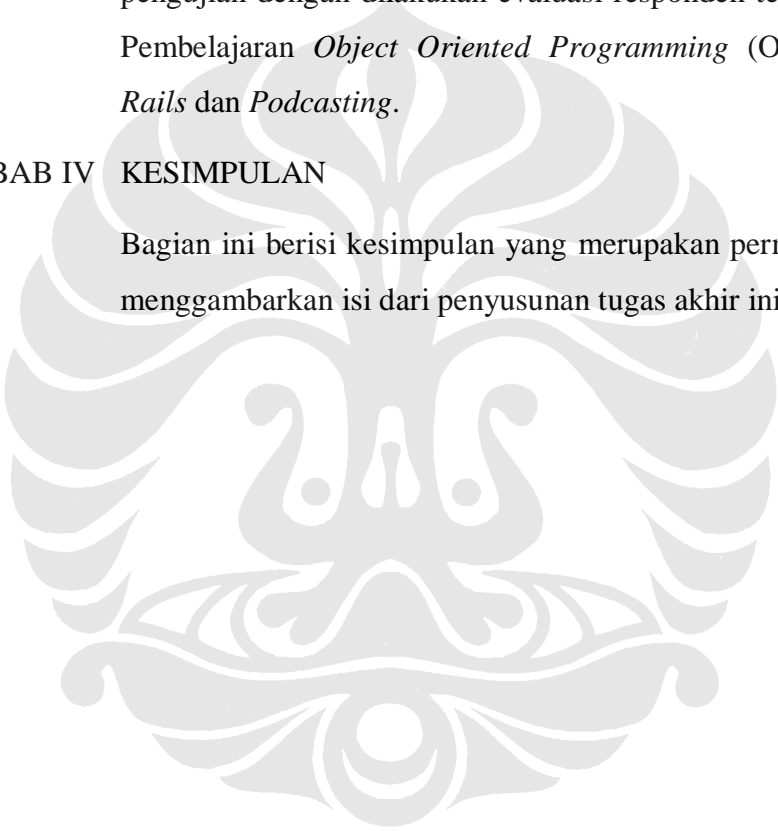
Bagian ini berisi penjelasan tentang rancangan sistem yang memuat analisa kebutuhan (*requirement*) sistem dan langkah-langkah yang akan dikerjakan untuk tahap implementasi sistem Portal Pembelajaran *Object Oriented Programming* pada *Ruby on Rails*.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bagian ini berisi penjelasan tentang implementasi program dan hasil pengujian dengan dilakukan evaluasi responden terhadap sistem Portal Pembelajaran *Object Oriented Programming* (OOP) pada *Ruby on Rails* dan *Podcasting*.

BAB IV KESIMPULAN

Bagian ini berisi kesimpulan yang merupakan pernyataan singkat yang menggambarkan isi dari penyusunan tugas akhir ini.



BAB 2 LANDASAN TEORI

2.1 Portal Pembelajaran

2.1.1 Definisi dan Tipe-tipe Portal

Portal merupakan salah satu contoh dari aplikasi berbasis web. Aplikasi ini menyediakan akses suatu titik tunggal dari informasi *online* terdistribusi, seperti dokumen yang didapat melalui pencarian, kanal berita, dan link ke situs khusus. Untuk memudahkan pengguna, biasanya disediakan kemampuan pencarian dan pengorganisasian informasi-informasi[1].

Menurut PortalsCommunity.com, portal dapat dikelompokkan menjadi empat kategori yaitu[2]:

- Portal *Corporate/Enterprise* (Intranet)
- Portal *e-Business* (Extranet)
- Portal Personal (WAP)
- Portal Mega/Publik (Internet)

2.1.1.1 Portal *Corporate/Enterprise* (Intranet)

Sering disebut Portal Perusahaan atau Business2Employee (B2E). Contoh dari portal ini adalah:

- *Business Intellegent Portal* ~ Portal perusahaan yang memungkinkan pengguna mengakses dan membuat laporan untuk tujuan pembuatan keputusan pada perusahaan yang biasanya menggunakan *database* yang besar.
- *Business Area (Intranet) Portal* ~ Portal ini menyediakan fungsi spesifik atau proses dan aplikasi-aplikasi dalam suatu perusahaan.
- Portal Horizontal ~ Portal ini biasa digunakan dalam hubungan antar organisasi/perusahaan. Contoh dari portal ini diantaranya *Enterprise Collaborative Portal (ECP)*, *Enterprise Expertise Portal (EEP)*,

Enterprise Knowledge Portal (EKP), Content Management, dan Manajemen Dokumen.

- *Role Portals* ~ Portal ini mengembangkan pelayanan terhadap 3 model bisnis yaitu *Business to Employee (B2E), Business to Corporate (B2C), dan Business to Business (B2B).*

2.1.1.2 Portal *e-Business* (Extranet)

Portal ini mempunyai tiga sub-kategori, yaitu:

- *Extended Enterprise Portal* (Portal Perluasan Perusahaan) ~ Portal ini digunakan sebagai perluasan perusahaan untuk para pelanggannya dengan tujuan pemesanan, tagihan, layanan pelanggan, *self service*, dsb. Perusahaan juga dapat memanfaatkan portal ini untuk memperluas perusahaan pada para *supplier* (pemasok barang) dan rekan kerjanya.
- *e-Marketplace Portal* (Portal Pemasaran melalui *net*) ~ Portal ini menyediakan perniagaan yang berhubungan dengan pelayanan kepada komunitas pembeli, penjual, maupun pembuat pasar melalui *net*. Portal akan menyajikan informasi yang dibutuhkan secara cepat dan membeli produk-produk serta jasa secara *online*.
- *ASP Portal* ~ ASP Portal merupakan portal B2B yang mengizinkan pelanggan bisnis menyewa produk maupun jasa.

2.1.1.3 Portal Personal (WAP)

Portal Pribadi ini dapat dibagi menjadi dua tipe utama, yaitu:

- *Pervasive portal atau mobility portal* ~ Portal ini terintegrasi dalam telepon web, telepon seluler, *wireless* PDA, pager, dan sebagainya.
- *Appliance portals* ~ Portal yang tergabung dalam TV (Web TV), otomobil (OnStar), dan sebagainya.

2.1.1.4 Portal Mega/Publik (Internet)

Organisasi yang masuk dalam kategori portal ini akan berfokus membangun pemirsa *online* yang besar dengan demografi yang luas atau berorientasi secara profesional. Ada dua tipe utama Portal Publik, yaitu[2]:

- *General public portals atau Mega Portals* ~ Portal ini sepenuhnya beralamatkan di Internet dengan masing-masing memiliki kepentingan khusus, contohnya seperti Yahoo, Google, MSN, dll.
- *Industrial Portals* ~ Portal ini berfokus pada komunitas yang sempit seperti konsumen suatu barang, bank, asuransi, dsb.

Berdasarkan uraian tipe-tipe portal di atas, maka Portal Pembelajaran *Object Oriented Programming* (OOP) Berbasis Ruby on Rails yang akan dibuat dalam tugas akhir ini termasuk ke dalam kategori *General Public Portals* atau disebut juga *Mega Portals*.

2.1.2 Komponen Teknik Portal

Portal melibatkan berbagai macam komponen teknik yang mendukung berfungsinya portal tersebut[1]. Sebuah portal komprehensif menggabungkan berbagai macam Internet dan aplikasi yang terkait dengan komponen-komponen teknologi. Karena tujuan dari portal adalah menyediakan tampilan tunggal untuk *end user* terhadap informasi yang datang dari berbagai sumber, daftar dari teknologi yang mungkin digunakan tak ada habis-habisnya. Beberapa dari teknologi yang digunakan (seperti *web service*) secara konstan berkembang, sementara yang lainnya (seperti *database* portal, yang biasanya tergabung dalam *SQL database*) secara relatif merupakan teknologi yang stabil[2].

Komponen-komponen teknik portal diantaranya adalah *server* aplikasi, *server web*, *database*, taksonomi, *Crawler* (perayap), gudang metadata, portlet, *Categorization Engine*, *filter*, *index*, *virtual card*, *web service*, protokol dan standar *development* (XML, XSL- XSLT, DTD dan XSD, WDSL, SOAP, UDDI, WSUI), profil pengguna, *Content Management System* (CMS), dan *Enterprise Application Integration* (EAI)[2].

2.1.3 Portal dan Infrastruktur

Portal terdiri dari berbagai macam infrastruktur yang mendukung portal itu berdiri. Tanpa infrastruktur ini portal tidak akan dapat memberikan layanan[1]. Beberapa infrastruktur yang terkait dengan portal diantaranya adalah sebagai berikut[2]:

- *Hosting Server Provider*
- *Service Level Agreement*
- *Platform* (lapisan klien, *server* presentasi, *server* aplikasi, *server* integrasi, *server* data)
- Manajemen Sistem
- *Network* (jaringan)
- *Interoperability* (lintas operasi)
- Penyebaran Teknologi (performansi, ketersediaan)
- *Unified Development Environment* (penyeragaman dalam lingkungan pembangunan)
- *Unity* (kesatuan)

2.1.4 Portal dan Keamanan

Portal adalah kumpulan layanan dari banyak *provider* dan menempatkannya ke dalam presentasi yang terorganisasi yang sesuai atau cocok dengan aliran kerja para pelanggannya (*user*). Para *provider* menggunakan berbagai sistem dan paradigma aplikasi yang berbeda, dimana semuanya mempunyai *hardware* yang berbeda, sistem operasi (*operating system*) yang berbeda, dan paradigma aplikasi yang berbeda untuk mengatur keamanannya[2].

Teknologi-teknologi yang dapat dimanfaatkan untuk mengatur keamanan pada aplikasi portal diantaranya adalah sebagai berikut [2]:

- *Single Sign-on* (SSO) ~ Teknologi *Single Sign on* sangat penting bagi portal. Singkatnya, portal membutuhkannya untuk mengkoordinasikan informasi dari beberapa *website*, penyimpanan data, XML, dan sistem transaksi lainnya. Kesemua ini mempunyai paradigma keamanan yang berbeda dimana solusi *Single Sign-on* akan diterapkan. Contoh dari *vendor*

yang memanfaatkan teknologi ini adalah Netegrity, Oblix, IBM, dan Entrust.

- Pendelegasian Manajemen ~ Sebuah evolusi dari teknologi *Single Sign-on*. Ketika SSO mencoba untuk memfasilitasi aktivitas, Sistem Pendelegasian manajemen mencoba untuk bertindak sebagai titik tunggal bagi pengaturan semua aplikasi dan sistem operasi tingkat keamanan. Sistem Pendelegasian manajemen pada akhirnya akan menggantikan sistem SSO seiring pendewasaannya. Contoh dari *vendor* yang memanfaatkan teknologi ini adalah Netegrity dan IBM.
- *Firewalls* ~ *Firewalls* adalah komputer yang menjalankan *software* yang menganalisis dan menyaring paket jaringan dan membuat keputusan sekuritas berdasar rekomendasinya.
- *Intrusion Detection* (Pendeteksian Penyusup) ~ *Software* ini menganalisis pola aktifitas dalam suatu jaringan untuk mengetahui jika sistem “diserang”.
- Kriptografi ~ Ilmu pengetahuan ini menyediakan perhitungan matematis yang teliti dalam rangka otentifikasi, enkripsi, dan *non-repudiation*. Keamanan portal yang tinggi menerapkan kriptografi untuk semua kemampuannya.
- *Access controls* ~ Sistem kendali akses menyediakan aturan berdasar daftar identitas untuk menetapkan sebuah identitas, dimana bagian dari sebuah peranan/tugas dari group/kelompok, harus mempunyai level akses yang tepat untuk menjalankan operasi yang menggunakan sumber daya. Ilmu pengetahuan keamanan komputer adalah kombinasi dari kendali akses dan teknologi kriptografi. Semua portal umumnya menggunakan kendali akses.
- Otentifikasi ~ Otentifikasi mempunyai dua macam format, yaitu format kriptografi dan format kendali akses. Format kriptografi dalam otentifikasi menggunakan skema dasar sertifikasi untuk memastikan identitas. Format

kendali akses lebih sederhana, dimana biasanya menggunakan mandat seperti *user-id* (identitas pengguna) dan *password*.

- *Non Repudiation* ~ Tindakan mempercayakan data pada suatu sistem kunci yang susah dirusak disebut *non repudiation* yang menggunakan teknologi *public key* dan fungsi-fungsi lama dari kriptografi. Portal Finansial, Portal Perawatan Kesehatan akan memperoleh manfaat lebih dari teknologi ini.
- Otorisasi ~ Sebuah fungsi kendali akses yang sangat penting dimana portal akan memelihara daftar otorisasi (daftar kontrol akses) untuk menetapkan level akses yang tepat pada setiap identitas yang bisa mengakses sumber daya. Seperti sebuah sistem akan menetapkan jika *user* diotorisasi terhadap tindakan yang dilakukan terhadap sumber daya.
- *Policy* ~ Sebuah kebijakan keamanan perlu ditetapkan dalam organisasi. Outline kebijakan keamanan dalam bisnis dibutuhkan untuk keamanan dan prosedur organisasi untuk mempertemukan kebutuhan-kebutuhan bisnis. Seperti sebuah kebijakan yang digunakan untuk mendefinisikan kendali akses dan kebijakan sertifikasi.
- *Groups* ~ *Groups* adalah kumpulan identitas yang diorganisasikan. *Groups* diatur atau dikonfigurasi oleh personil administratif dan diatur di atas basis harian (hari per hari). Portal selalu butuh untuk mengatur group sebagai alat ekonomi untuk mengatur privasi, integritas, dan aksesibilitas yang tepat dari data.
- *Roles* ~ *Roles* mengorganisasikan kumpulan kemampuan. Kumpulan dari kemampuan ini cenderung dipelihara oleh *developer*. *Roles* dapat berupa group dan atau *user* sebagai anggota yang mempunyai akses untuk kemampuan yang didefinisikan oleh *developer*. Keanggotaan *roles* cenderung diatur oleh administrator.
- *Lightweight Directory Access Protocol (LDAP)* ~ Sebuah struktur direktori umum yang diterima oleh sebagian besar industri. Portal

menggunakan ini untuk mengatur informasi pengguna, informasi organisasi, sebagai kontrol akses dan informasi sertifikasi kriptografi.

- *Certificates Authorities atau CA* (Otoritas Sertifikasi) ~ Otoritas sertifikasi adalah penentu keputusan dalam pembuktian, identitas digital, meskipun pada kenyataannya cenderung tidak dapat dipertanggungjawabkan atas hasil kerjanya. Berdasarkan hal ini, dan aksi tanda tangan digital, otoritas sertifikasi tidak dapat diadopsi secara luas. Otoritas sertifikasi dapat menghasilkan sertifikasi. Walaupun ada publik CA, seperti Valicert dan Verisign, tetap banyak perusahaan yang membuat sertifikasinya sendiri. CA bermanfaat bagi portal yang menyediakan servis perdagangan dengan nilai tinggi atau jasa pelayanan kesehatan. Mereka juga menyediakan mekanisme *third party* (pihak ketiga) untuk memvalidasi identitas. Aplikasi portal yang lebih kecil akan membuat sertifikasinya sendiri. Tanda tangan digital memungkinkan sertifikasi-diri sendiri. Sertifikasi diri sendiri (*self certified*) ini legal dan valid untuk bertransaksi.
- *Validation Authorities atau VA* (Otoritas validasi) ~ Standar X509 meragukan dan tidak semua sertifikat yang dibuat dari semua *vendor* sama. Dalam hal ini, ketika perusahaan-perusahaan saling bertukar sertifikat sebelum melakukan e-Business, perusahaan “sumber“ membuat sertifikat dan akan mengontrol pemeliharaan sertifikat. Dengan kata lain, jika sumber pengguna “menjadi jahat“ sumber perusahaan pengguna akan meninjau kembali sertifikat tersebut. Otoritas validasi memungkinkan perusahaan tujuan melakukan “penerbitan sertifikasi lokal“, hal ini mengurangi kebutuhan komunikasi organisasi yang kuat diantara dua perusahaan yang menerapkan sertifikasi transaksi secara kriptografi.
- *Public Key Infrastructur* (Infrastruktur Kunci Publik) ~ Kriptografi kunci publik menyediakan implementasi elegan (elok) dari implementasi enkripsi, *non repudiation*, dan otentifikasi yang membutuhkan aktifitas manajemen kunci yang minimum. Ini membuat infrastruktur kunci publik lebih efisien untuk diatur bila dibandingkan dengan infrastruktur kunci

simetrik yang tradisional. Portal yang membutuhkan kriptografi akan menggunakan *Public Key Infrastructure*(PKI).

- *Secure Socket Layer* (SSL) ~ Standar bagi transaksi-transaksi yang butuh keamanan dengan menggunakan kriptografi public key dan X509. Ini dikhususkan bagi otentifikasi (dua arah) dan enkripsi informasi yang dikirim melalui socket TCP/IP. Portal yang membutuhkan transaksi finansial atau data perawatan kesehatan akan menggunakan SSL.
- *Secure Access Markup Language* (SAML) ~ Terinspirasi oleh Netegrity, bahasa ini dibangun untuk memfasilitas Strategi Manajemen Pendelegasian. Berisi transaksi-transaksi yang tak bereputasi untuk mengatur kontrol aksesnya. Dengan ini diharapkan *vendor-vendor software* akan merangkul SAML untuk memfasilitasi strategi SSO miliknya (segera akan dikenal Manajemen Pendelegasian). Portal akan mengurangi biayanya dalam jangka menengah dengan mengadopsi SAML, sebagai sebuah integrasi dengan paradigma keamanan lainnya yang akan semakin sederhana.
- *Digital Signature* (Tanda tangan Digital) ~ Tanda tangan digital memanfaatkan kemampuan kunci (*non repudiation*) yang susah ditembus dari *public key infrastructure* untuk menyediakan kriptografi yang memastikan data dikelola sebagai integritas.

2.2 Ruby on Rails

2.2.1 Pemrograman Ruby

Bahasa Ruby lahir pada tanggal 23 Februari 1993 dan masuk ke Amerika pada tahun 2000. Pembuat bahasa ini adalah **Yukihiro “Matz” Matsumoto** yang semasa kuliahnya sangat menyenangi pemrograman berorientasi objek dan bahasa *scripting*. Matsumoto kemudian melakukan riset terhadap bahasa *scripting* Perl dan Python, Menurutnya, bahasa Perl kurang *powerful* dan bahasa Python kurang *object-oriented*. Maka lahirlah bahasa Ruby (berarti sejenis batu permata) yang lebih *powerful* daripada Perl dan lebih *object-oriented* dari Python[3].

Fitur-fitur Ruby diantaranya adalah sebagai berikut:

- Merupakan bahasa interpreter.
- Memiliki sintak yang sederhana, mudah dipelajari dan dipahami.
- Mendukung *exception handling* seperti halnya Java dan Python.
- Mendukung *single-inheritance* dan modul *mix-in* yang serupa dengan interface di Java. (Multiple-inheritance dapat digantikan dengan menggunakan fitur *mix-in* ini).
- Memiliki portabilitas yang tinggi antar-*platform* sehingga dapat berjalan di berbagai sistem operasi seperti UNIX, Linux, DOS, Windows 95/98/Me/NT/2000/XP, MacOS, BeOS, OS/2, dan sebagainya.
- Mendukung *dynamic-typing*, seperti halnya pada Python tidak diperlukan pendeklarasian tipe untuk suatu variabel.
- Mendukung *garbage collection* seperti halnya pada Java dan Python tidak diperlukan membebaskan memory yang dialokasi (mis. `free()` di C). Variabel yang tidak lagi digunakan akan segera dibebaskan oleh *garbage collector* sehingga tidak diperlukan manajemen memori.
- Mudah dikembangkan dengan bahasa C seperti halnya Python misalnya dengan menggunakan interface SWIG.
- Lahir dari komunitas, sehingga Ruby memiliki dukungan komunitas yang siap membantu jika menemui kesulitan.
- Gratis, bahkan untuk aplikasi komersial.

2.2.2 *Framework* Ruby on Rails (RoR)

RoR ditulis dalam bahasa pemrograman Ruby. Menurut Russel Kay RoR adalah *full-stack web application framework open-source*[10]. *Full-stack* mengacu pada tingkat kemampuan yang disediakan oleh *framework* RoR. *Web application* adalah *software* aplikasi yang menggunakan *browser* (IE, Firefox, Opera, dll) sebagai aksesnya dalam suatu jaringan Internet atau Intranet. *Framework* dapat diasumsikan sebagai pondasi dari sebuah *software* aplikasi web.

RoR diciptakan oleh David Heinemeier Hansson[11] asal Denmark, seorang mahasiswa perguruan tinggi. RoR pertama kali ditulisnya pada sebuah *project*

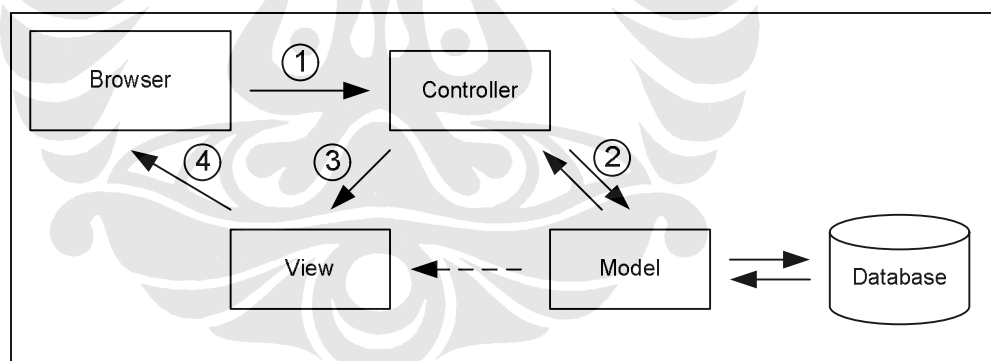
management tool Basecamp[12]. RoR versi beta pertama dilepaskan kepada publik pada bulan Juli 2004, kemudian diikuti dengan versi 1.0 pada tanggal 13 Desember 2005. Hingga saat ini lebih dari 300.000 *framework* RoR telah di *download* dan terus meningkat.

RoR juga termasuk *framework open-source*, dirancang untuk membuat pekerjaan para pengembang web menjadi lebih ringan dengan menyediakan segala hal yang perlu untuk membuat aplikasi berbasis web. RoR dirancang menjadi kolaborasi dari *quick-n-dirty* PHP dengan metodologi *slow-n-clean* JSP[13]. Ini berarti performansi RoR cepat seperti PHP dengan struktur sintak seperti JSP.

2.2.3 Arsitektur Rails

2.2.3.1 Model, View, dan Controller

Pada tahun 1979, Trygve Reenskaug datang dengan arsitektur yang baru untuk membangun aplikasi yang interaktif. Dalam rancangannya, aplikasi dibagi menjadi tiga komponen, yaitu *model*, *view*, dan *controller*[4].



Gambar 2.1 Arsitektur Model-View-Controller

Dari Gambar 2.1 di atas dapat disimpulkan bahwa proses yang terjadi dalam model MVC secara umum adalah sebagai berikut:

1. Browser akan mengirimkan *request* ke *controller*.
2. *Controller* akan merespon dan berkomunikasi dengan *model*. Komunikasi ini dapat berupa mengakses data, ataupun mengubah data yang disimpan oleh *model*. (Namun tidak semua *model* harus berhubungan dengan *database*).

3. *Controller* akan membuat *view* yang bersesuaian.
4. *Browser* akan me-render *view* yang ada.

Rails adalah sebuah *framework* MVC untuk pengembangan aplikasi berbasis web. Rails ditulis dengan menggunakan Ruby yang dikenal sangat *Object Oriented*. Di dalam pengembangan aplikasi menggunakan Rails dikenal sebuah paradigma yang disebut *convention over configuration*. Misalnya, untuk penamaan tabel di *database*, nama tabel harus menggunakan kata benda/jamak dari nama model yang dimiliki. Sebagai contoh, model *Article*, maka diberikan tabel *articles* di *database*. Jika tidak diikuti maka harus dilakukan konfigurasi secara manual untuk melakukan *mapping* dari model *Article* ke tabel *articles* di *database*.

Arsitektur dengan tiga komponen pada Gambar 2.1 biasa disebut dengan arsitektur MVC. Berikut penjelasan ketiga komponen tersebut:

- a. *Model* - bertanggung jawab mengatur kondisi aplikasi. Terkadang kondisi ini bersifat sementara, hanya untuk berinteraksi dengan *user*. Terkadang juga bersifat permanen dan akan disimpan di luar dari aplikasi (biasa disebut *database*).
- b. *View* - bertanggung jawab menampilkan *user interface* berdasarkan data yang ada pada *model*.
- c. *Controller* - mengendalikan jalannya aplikasi. *Controller* menerima *event* dari luar (input *user*), berinteraksi dengan *model*, dan menampilkan *view* yang sesuai untuk *user*.

2.2.3.2 Prinsip Pengembangan Aplikasi RoR[5]

RoR lebih menonjol jika dibandingkan dengan *framework* lainnya. Hal ini disebabkan RoR memiliki prinsip pengembangan *framework* seperti berikut:

- a. *Convention Over Configuration*

Prinsip ini mengacu pada fakta bahwa RoR diasumsikan sebagai kumpulan sejumlah aplikasi web yang khas yang sudah menjadi standarisasi. Pada *framework* lain seperti *Java-based Struts* atau *Python-based Zope*, sebelum dimulai diperlukan suatu konfigurasi yang panjang.

Informasi konfigurasi biasanya disimpan ke dalam *file* XML yang besar dan sulit untuk di-*maintain*. Atau jika akan memulai suatu proyek baru wajib mengulangi seluruh proses konfigurasi. Heinemeier Hansson dengan sengaja menciptakan RoR sedemikian sehingga tidak memerlukan konfigurasi berlebihan, sepanjang beberapa konvensi yang standar diikuti.

b. *Don't Repeat Your Self*

RoR mendukung prinsip DRY (*Don't Repeat Yourself*) programming. Untuk mengubah perilaku suatu aplikasi yang didasarkan pada prinsip DRY, tidak diperlukan memodifikasi terlalu banyak sintak. Contoh bagaimana RoR mendukung prinsip DRY adalah bahwa, tidak seperti Java, RoR tidak memaksa untuk mengulangi definisi *database* di dalam aplikasi yang dibuat.

RoR juga menerapkan teknik Ajax (*Asynchronous Javascript and XML*). Ajax adalah satu pendekatan yang mengizinkan aplikasi web menggantikan isi di dalam *browser* secara dinamis. Misalnya, untuk menukar data sebuah *form* tanpa harus men-*download* seluruh halaman dari *server*. Para pengembang sering kali harus menyalin sintak selagi menciptakan aplikasi-aplikasi Ajax. RoR membuatnya mudah untuk memperlakukan masing-masing generasi *browser* tanpa harus menyalin setiap sintak untuk masing-masing *browser*.

c. *Agile Development*

Pada pendekatan pengembangan *software* tradisional pembuatan sketsa perencanaan biasanya lebih panjang dan kaku. Pendekatan ini biasanya memakai pendekatan aplikasi dari bawah-ke atas, langkah pertamanya berdasarkan data.

Sebaliknya, metode *Agile Development* menggunakan pendekatan adaptif. Pendekatan adaptif ini hanya membutuhkan sejumlah kecil tim sebagai pengembang. Para pengembang *Agile* juga merancang aplikasinya dari atas-ke bawah yang dimulai dari desain *layout*.

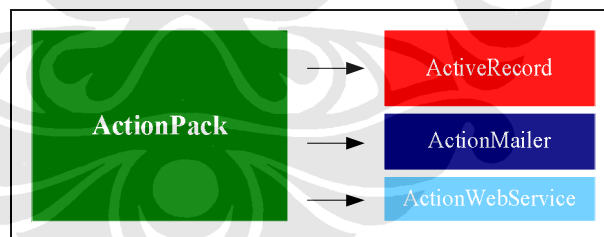
Menurut Patrick Lentz[14] ada beberapa contoh ilustrasi RoR menganut metode *Agile Development*.

- Pekerjaan dapat dimulai dengan mendesain *layout* aplikasi RoR sebelum memutuskan mengenai *database*. Tidak diperlukan membuat ulang *layout* tersebut ketika menambahkan fungsi ke desain, semuanya telah diatur secara dinamis berdasarkan kebutuhan.
- Tidak seperti sintak yang dibuat dalam bahasa C atau Java, aplikasi RoR tidak memerlukan kompilasi sehingga mempercepat proses pengembangan.
- RoR menyediakan otomatisasi uji coba sintak aplikasi pada proses pengembangan.
- *Refactoring (rewriting code with an emphasis on optimization)*.

Berdasarkan prinsip-prinsip di atas, RoR merupakan *framework* yang benar-benar menghemat waktu dalam usaha para pengembang membuat sebuah aplikasi web berbasis RoR.

2.2.3.3 Komponen Ruby on Rails

Secara umum garis besar, *framework* RoR terdiri dari beberapa komponen seperti pada Gambar 2.2 berikut ini:



Gambar 2.2 Komponen Ruby on Rails

a. *ActionPack*

Komponen ini merupakan jantung dari *framework* RoR. Terdiri atas dua sub-komponen:

- *ActionController*, modul ini memastikan pemakaian aksi yang tepat berdasarkan *web-request* setiap pengguna. Aksi ini diambil berdasarkan URL. Contohnya *.../books/list* akan memanggil metode 'list' pada *BookController-class*.

- *ActionView*, komponen ini digunakan oleh *ActionPack* untuk mengembalikan *request-view*. *ActionView* menggunakan *template* RHTML, RJS dan RXML. *Template* ini dapat digunakan untuk *men-generate* HTML/JS/XML.

b. *ActiveRecord*

ActiveRecord merupakan ORM (*Object Relational Mapper*) yang *powerful* dan salah satu kekuatan RoR. Kekuatannya adalah mudah digunakan, tidak memerlukan konfigurasi, hanya mewariskan *ActiveRecord* base class maka secara otomatis *object-model* dipetakan ke dalam *database-model*. Semua fungsi umum dari relasi *database* telah tersedia seperti proses pembuatan, modifikasi dan pencarian ke dalam *database*. Syaratnya untuk setiap tabel harus memiliki nama *field* id sebagai *primary key*.

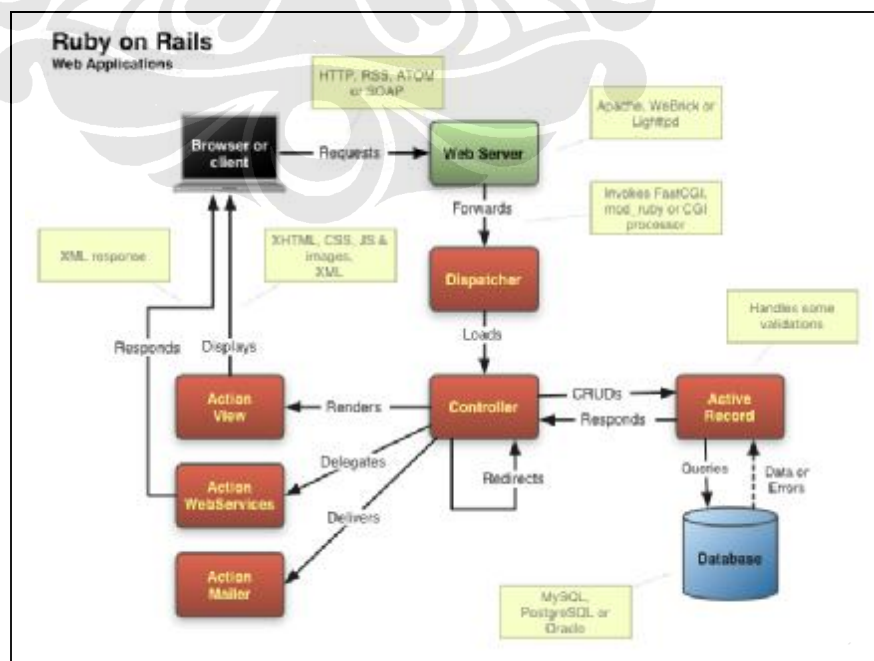
c. *ActionMailer*

Disediakan RoR untuk komunikasi melalui email.

d. *ActiveWebService*

Dukungan RoR terhadap XML-RPC dan SOAP *webservice*.

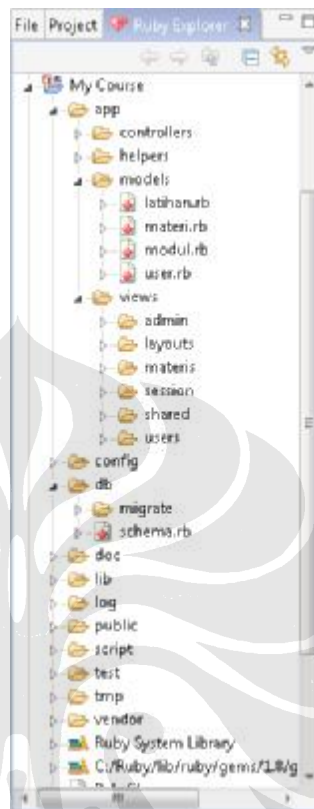
Berdasarkan komponen-komponen di atas, dapat digambarkan aliran kerja dari aplikasi web berbasis Ruby on Rails pada Gambar 2.3 berikut ini[5]:



Gambar 2.3 Diagram Ruby on Rails[5]

2.2.3.4 Struktur Direktori Aplikasi Rails

Gambar 2.4 di bawah ini menunjukkan struktur direktori dari aplikasi RoR.



Gambar 2.4 Struktur Direktori Rails

2.2.4 Radrails

RadRails pada Aptana merupakan salah satu *open source editor* (IDE) untuk Ruby dalam *framework* RoR. Ada banyak IDE yang dapat mendukung *framework* RoR, namun penulis lebih memilih RadRails karena RadRails memiliki fitur yang lebih lengkap jika dibandingkan dengan IDE lainnya. Perbandingan antara RadRails dengan IDE lainnya diberikan pada Tabel 2.1[6].

Tabel 2.1 Perbandingan Fitur IDE

General	RadRails	NetBeans	3rdRail
Harga	Gratis	Gratis	\$399
Tipe Lisensi	<i>Open Source</i>	<i>Open Source</i>	Komersial
Available Standalone or as Eclipse Plugin	√	×	×
Interpreter Support/Bundling			
Bundled Jruby Interpreter	√	√	×
Interpreter Support	√ Ruby, Jruby, Rubinius	√ Ruby, Jruby	√ Ruby, Jruby
Scriptability/Extensibility			
Scriptable via Ruby	√	×	×

Debugging / Profiling			
Debugger	√ classic and ruby-debug for MRI; ruby debug bundled with Jruby	√ classic and ruby-debug for MRI; ruby debug bundled with Jruby	x
JavaScript Debugging	√	x	√
Profiler	√ (pro)	x	x
Editor			
Editor HTML	√	√	√
Editor CSS	√	√	√
Editor JavaScript	√	√	√
Editor JSON	√	√	x
Editor SQL	√	√	√
Editor YML	√	√	√
Editor RHTML/Erb	√	√	√
Editor XML	√	√	√
Ruby Editing			
Code Completion	√	√	√
Type Inferencing	√	√	√
Ruby-specific search engine (Find usages)	√	√	√
Code analysis(warnings/errors/hints)	√	√	x
Type Hierarchy View	√	x	√
Call Hierarchy View	√	x	√
Mylyn Integration	√	x	√
Reguler Expressions Tester	√	x	√
Quick Outline	√	x	√
Spell Checking Support	√	√	x
Smart Indent	√	√	x
Mark Occurences	√	√	x
Refactoring			
Rename	√ (1)	√	√
Konversi Variabel Local menjadi Field	√	x	x
Encapsulate Field	√	x	x
Extract Method	√	√	x
Extract Constant	√	√	x
Inline Class	√	x	x
Inline Local Variable	√	x	x
Inline Method	√	x	x
Merge Class Parts (internal to <i>file</i> & external)	√	x	x
Move Field	√	x	x
Move Method	√	x	x
Push Down Method	√	x	x
Pull Up Method	√	x	x
Split Local Variable	√	x	x
Pengujian			
Menampilkan Test :: Unit	√	√	x
Auto Test	√	√	x
Mendukung Rspec	√	√	x
Rails Specific Functionality			
Integrated rails-specific "shell"	√	x	√
Log Tail View	√	x	√
Embedded browser	√	x	√

2.3 Web v2.0

2.3.1 Definisi AJAX

AJAX atau *Asynchronous JavaScript XML* diperkenalkan oleh Jesse James Garret dari Adaptive Path pada tahun 2005. AJAX adalah teknologi *browser (client-side)* yang tidak tergantung pada *software web server* tertentu. Melalui AJAX,

JavaScript dapat dikomunikasikan secara langsung dengan *server* menggunakan obyek JavaScript XMLHttpRequest. Obyek JavaScript ini dapat *men-trade* data sebuah web *server* tanpa harus *me-reload* (*refresh*) halaman web[7].

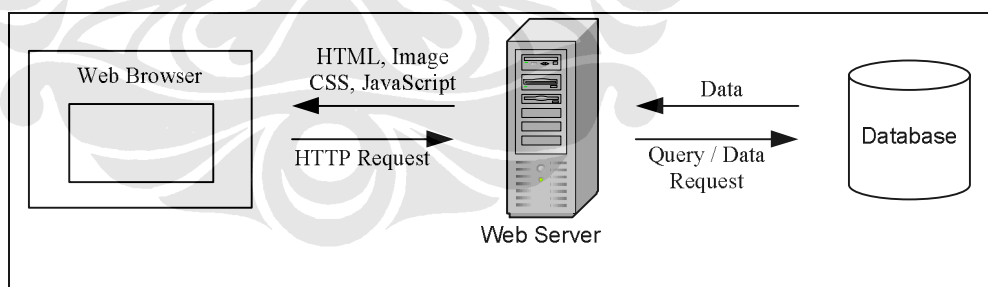
AJAX merupakan aplikasi web yang lebih baik. Aplikasi web menambah keuntungan dibanding aplikasi *desktop* karena hal-hal berikut ini:

- Dapat menjangkau pengguna yang luas
- Mudah diinstall
- Mudah dikembangkan
- Mudah dipelihara

Aplikasi Internet tidak selalu susah dan *user friendly* seperti aplikasi *desktop*. Dengan AJAX, aplikasi Internet menjadi semakin kecil, cepat dan mudah digunakan.

2.3.1.1 Model Tradisional

Pengembangan web secara tradisional bekerja secara *synchronous*, antara aplikasi dan *server*, setiap kali melakukan link atau melakukan operasi “submit” pada form. Model tradisional adalah model yang sering digunakan tanpa AJAX.

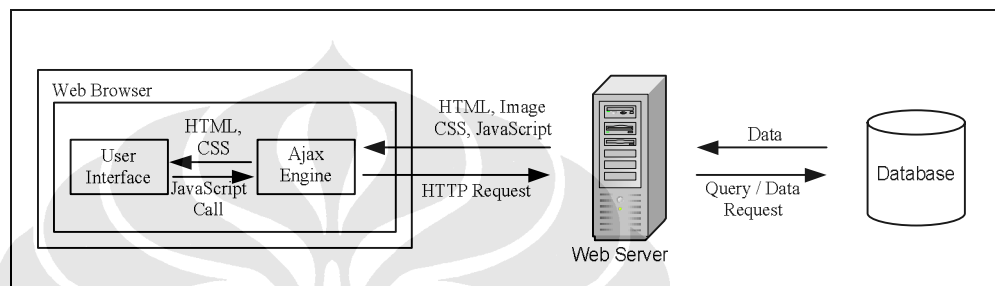


Gambar 2.5 Arsitektur Model Tradisional

Gambar 2.5 merupakan arsitektur model tradisional. Pada model ini (lihat Gambar 2.5), *server* mengirimkan respon berisi seluruh halaman termasuk *header*, logo, navigasi, *footer*, dll. Ketika mengklik *next* maka akan menampilkan halaman baru lagi (artinya *header*, logo, navigasi, *footer* dikirim ulang) dan seterusnya akan mengirimkan data halaman baru lagi setiap diminta *request* dari *user*.

2.3.1.2 Model AJAX

Aplikasi web yang bekerja dengan AJAX bekerja secara *asynchronous*, yang berarti mengirim dan menerima data dari *user* ke *server* tanpa perlu *me-load* kembali seluruh halaman, melainkan hanya melakukan penggantian pada bagian web yang ingin diubah. Dalam model AJAX, aksi dari sisi klien dibagi menjadi dua bagian, yaitu layer *User Interface* (UI) dan layer AJAX.



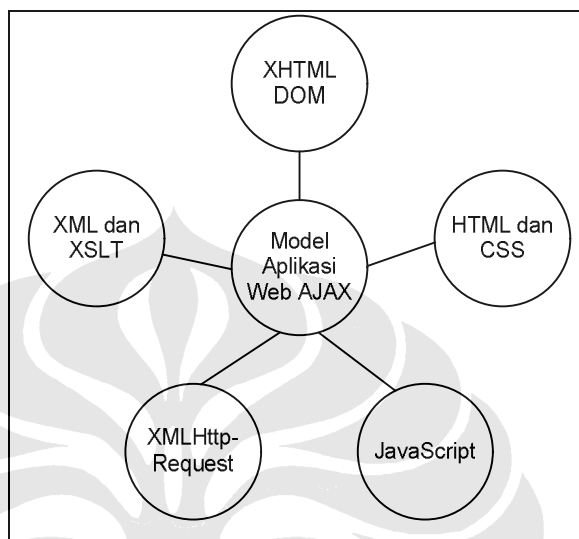
Gambar 2.6 Arsitektur Model AJAX

Gambar 2.6 merupakan arsitektur model AJAX. Berikut ini poin-poin yang penting untuk menggambarkan AJAX:

- Layer AJAX tidak memerlukan komunikasi dengan *server* (contohnya untuk validasi form karena dapat ditangani sepenuhnya oleh sisi klien).
- *Request* antara layer AJAX dan *server* berupa bagian kecil dari informasi (tidak lengkap satu halaman), maka sering digunakan untuk interaksi dengan *database* sehingga waktu *render* dan waktu pengiriman menjadi pendek.
- Layer UI secara langsung tergantung pada respon *server* sehingga *user* dapat melanjutkan interaksi dengan sebuah halaman selama aktivitas dikerjakan di *background* (*background process*). Berarti, untuk beberapa interaksi, waktu tunggu *user* hampir tidak ada.
- Komunikasi antar halaman dan *server* tidak selalu memerlukan AJAX untuk mengubah perubahan UI. Contoh, beberapa aplikasi menggunakan AJAX untuk notifikasi dengan halaman, tetapi tidak melakukan apapun terhadap respon dari *server*.

2.3.2 Teknologi di balik AJAX

Teknologi AJAX merupakan paduan dari beberapa teknologi yang telah dikenal sebelumnya. Adapun komponen-komponen yang terdapat di dalam teknologi ini digambarkan pada Gambar 2.6 berikut ini:



Gambar 2.7 Teknologi di balik AJAX

- **HyperText Markup Language (HTML)** digunakan dalam membuat halaman web dan dokumen-dokumen lain yang dapat ditampilkan dalam *browser*. HTML merupakan standar internasional dengan spesifikasi yang ditetapkan oleh *World Wide Web Consortium (W3C)*.
- **Extensible HyperText Markup Language (XHTML)** adalah bahasa *markup* sebagaimana HTML, tetapi dengan gaya bahasa lebih baik.
- **Cascading Style Sheets (CSS)** adalah sebuah mekanisme sederhana untuk memberikan *style* (seperti font, warna, jarak spasi, dll) kepada dokumen web yang ditulis dalam HTML atau XML (termasuk beberapa variasi bahasa XML seperti XHTML dan SVG).
- **JavaScript** adalah bahasa *scripting* kecil, ringan, berorientasi-objek dan lintas platform. JavaScript tidak dapat berjalan dengan baik sebagai bahasa mandiri, melainkan dirancang untuk ditanamkan pada produk dan aplikasi lain seperti web.

- **Document Object Model (DOM)** adalah sebuah API (*Application Program Interface*) untuk dokumen HTML dan XML. DOM menyediakan representasi dokumen secara terstruktur, dimungkinkan untuk merubah isi dan presentasi visual. Pada dasarnya, DOM menghubungkan halaman web dengan script atau bahasa pemrograman.
- **Extensible Markup Language (XML)** adalah bahasa *markup* untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen *markup* keperluan khusus. Keperluan utama XML adalah untuk pertukaran data antar sistem yang beraneka ragam.
- **Extensible Stylesheet Language Transformations (XSLT)** adalah sebuah bahasa berbasis-XML untuk transformasi dokumen XML. Walaupun proses merujuk pada transformasi, dokumen asli tidak berubah melainkan dokumen XML baru dibuat dengan basis isi dokumen yang sudah ada. XSLT biasanya digunakan untuk merubah skema XML ke halaman web atau dokumen PDF.
- Objek **XMLHttpRequest** untuk melakukan pertukaran data secara asinkron dengan web *server*. Ajax menggunakan obyek XMLHttpRequest untuk melakukan pertukaran data dengan web *server*.
- **JavaScript Object Notation (JSON)** yaitu format pertukaran data komputer yang ringan dan mudah. Keuntungan JSON dibandingkan dengan XML adalah pada proses penterjemahan data menggunakan JavaScript. JavaScript dapat menterjemahkan JSON menggunakan *built-in* procedure **eval()**.

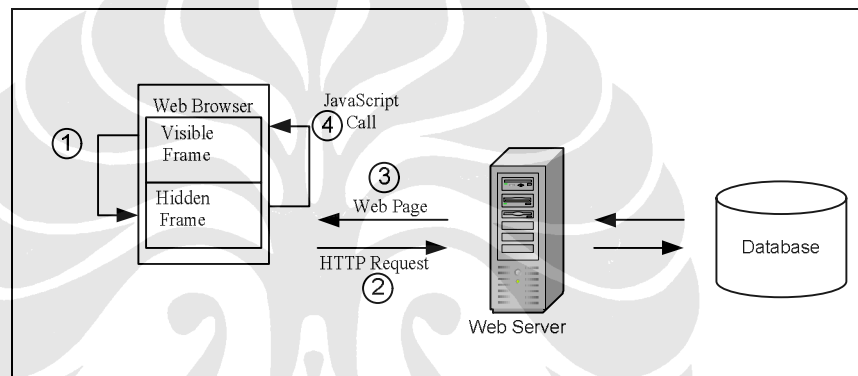
Dalam kenyataannya, semua teknologi dapat digunakan untuk AJAX, tetapi hanya tiga teknologi yang dibutuhkan yaitu HTML/XHTML, DOM dan JavaScript. XHTML diperlukan untuk menampilkan informasi, sedangkan DOM diperlukan untuk halaman XHTML tanpa di-*reload* ulang. JavaScript dibutuhkan untuk komunikasi *client-server*, sementara manipulasi DOM untuk meng-*update* halaman web. Teknologi yang lain digunakan agar AJAX lebih bagus, tetapi bukan merupakan hal yang pokok.

2.3.3 Teknik Penerapan AJAX

Teknik AJAX memungkinkan pengembang web membuat halaman seorang *user* tidak lagi menunggu untuk melakukan aksi selanjutnya. Hal ini berarti memungkinkan komunikasi ke *server* dalam setiap waktu. AJAX memiliki beberapa teknik untuk berkomunikasi yaitu: Hidden Frame dan IFrame, Cache Control (Cookie), dan HTTP Request.

2.3.3.1 Teknik Hidden Frame dan IFrame

Teknik ini secara spesifik memiliki empat pola, seperti terlihat pada Gambar 2.8 berikut ini:



Gambar 2.8 Teknik Hidden Frame AJAX

Metode ini memanfaatkan frame yang tersembunyi. Biasanya salah satu frame di-*set* dengan ukuran tinggi/lebar menjadi 0 pixel sehingga tidak terlihat di halaman. Frame tersembunyi inilah yang sebenarnya melakukan *request* ke dan menerima respon dari *server* sehingga frame yang tampil tidak tampak melakukan *post-back* ke *server*. JavaScript digunakan untuk mengambil data dan mengisi data yang ada di frame yang tersembunyi ini.

Setting hidden frame dapat dilihat pada skrip berikut ini:

```
<frameset rows="100%,*">
```

atau,

```
<iframe height="0" width="0" src="hidden.htm">
```

Hidden IFrame hampir sama dengan hidden frame. Perbedaannya terletak pada elemen yang digunakan, yaitu IFrame, bukan Frame.

2.3.3.2 Teknik Cache Control (Cookie)

Biasanya, ketika membuka halaman yang sama akan terlihat semakin cepat. Hal tersebut disebabkan ketika mengakses pertama kali, *browser* melakukan *caching*. Tujuan *caching* adalah menyimpan informasi yang telah dibuka untuk disimpan di *browser*. Teknik *caching* berfungsi untuk mempercepat dalam menampilkan halaman (yang telah di-load sebelumnya).

Penggunaan teknik pemanggilan AJAX beberapa kali akan dapat menyebabkan masalah. Dengan demikian, jalan terbaik adalah menambahkan *no-cache header* agar *browser* tidak melakukan *caching* (`Cache-Control:no-cache`).

2.3.3.3 Teknik HTTP Request

Metode HTTP Request memanfaatkan ActiveX Objek (IE) atau objek melakukan *post-back* ke *server* dan menerima respon balik berupa data (bukan halaman). Data yang didapat dari *server* kemudian diolah di klien untuk ditampilkan di halaman.

XMLHttpRequest adalah metode yang banyak digunakan AJAX untuk berkomunikasi karena memiliki dua fitur yang unik. Fitur utama adalah mempunyai kemampuan *me-load* isi data baru tanpa mengubah seluruhnya dan ini merupakan hal yang berbeda dibanding dengan cara konvensional. Fitur kedua adalah memperbolehkan JavaScript melakukan pemanggilan secara *asynchronous*.

2.3.4 Kelebihan dan Kekurangan AJAX

Dari teknik-teknik yang telah diutarakan pada sub-bab 2.4.3 di atas, terdapat kelebihan dan kekurangan dari masing-masing teknik tersebut. Tabel 2.2 menunjukkan kelebihan dan kekurangan tersebut.

Tabel 2.2 Teknik-teknik dalam AJAX beserta kelebihan dan kekurangannya

Teknik	Kelebihan	Kekurangan
XHR (XMLHttpRequest)	<ul style="list-style-type: none"> Dapat mengirim dan menerima (send/get) semua HTTP <i>header</i>. 	<ul style="list-style-type: none"> Pada IE5 dan 6 harus ada ActiveX <i>request</i>. Hanya berjalan pada Opera

	<ul style="list-style-type: none"> • Dapat membuat <i>request</i> HTTP menggunakan tipe (GET, POST, PROPEND). • Dapat mengontrol semua <i>request</i> POST, dan semua tipe. 	<ul style="list-style-type: none"> • dan Safari versi baru. • Tiap <i>browser</i> memiliki perbedaan cara implementasinya.
HF/IF (Hidden Frame/IFrame)	<ul style="list-style-type: none"> • Dapat menjalankan <i>request</i> GET dan POST. • Didukung oleh semua <i>browser</i> modern. • Mendukung <i>upload file</i> secara <i>asynchronous</i>. 	<ul style="list-style-type: none"> • Tidak boleh menggunakan sistem <i>synchronous</i>. • Implementasi berbeda pada tiap <i>browser</i>. • Meninggalkan <i>history</i> tambahan (tergantung dari <i>browser</i> dan implementasinya) • Semua <i>request</i> data adalah <i>encode-URL</i>, dan dapat menambah ukuran.
CK (Cookie)	<ul style="list-style-type: none"> • Didukung oleh banyak <i>browser</i>. • Sedikit perbedaan pada implementasi tiap <i>browser</i>. 	<ul style="list-style-type: none"> • Tidak boleh menggunakan <i>synchronous requests</i>. • Tidak boleh bekerja dengan <i>request</i> atau hasil yang besar. • Membutuhkan halaman <i>server</i> yang mendukung <i>cookie</i>. • Hanya menggunakan <i>request</i> GET HTTP.

Secara umum, teknik pemrosesan halaman ada dua, yaitu:

a. Pembuatan/manipulasi objek dokumen menggunakan JavaScript

Klien mengirimkan data dalam format XML/JSON kepada *server* dan mendapatkan data dari *server* berupa XML/JSON. Data tersebut kemudian diolah untuk memanipulasi objek dokumen menggunakan DOM dan JavaScript.

b. Parsial Rendering

User Interface (UI) dan perilaku UI tidak diproses di klien melainkan di *server*. Klien menerima UI dan perilakunya kemudian melakukan *rendering* pada bagian halaman tersebut.

Dengan AJAX, pertukaran data antara klien dan *server* lebih ringan karena hanya data yang dipertukarkan (bukan halaman) sehingga aplikasi web dapat berjalan lebih cepat.

2.3.5 Keistimewaan AJAX

AJAX memiliki beberapa keistimewaan, antara lain:

- Membuat permintaan kepada *server* tanpa memuat kembali (*reload*) kembali.
- Mengurai (*parse*) dan bekerja dengan dokumen XML dan atau JSON.
- Data yang dikirim sedikit sehingga menghemat *bandwidth* dan mempercepat koneksi.
- Proses dilakukan di belakang layar.
- Banyak didukung oleh *browser-browser* moder yang populer.
- Aplikasi yang dibangun semakin interaktif dan dinamis.

Keuntungan-keuntungan yang akan didapatkan dalam penerapan AJAX diantaranya adalah:

- *High Usability*: *Update* data tidak *me-reload* keseluruhan halaman, melainkan hanya yang relevan.
- *High Speed*: Aplikasi AJAX lebih cepat dibandingkan dengan aplikasi web konvensional.

2.4 Alat-alat Pemodelan


Alat untuk memodelkan sistem yang digunakan dalam tugas akhir ini adalah *Use-Case Diagram* dan *Activity Diagram*.



2.4.1 Use-Case Diagram

Use-Case Diagram secara grafik menggambarkan perilaku sistem (*Use-Case*). Diagram ini digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem [8].

Komponen dari *Use-Case Diagram* ditunjukkan pada Tabel 2.3.

Tabel 2.3 Komponen *Use-Case Diagram*

Gambar	Penjelasan
	<i>Actor</i> merepresentasikan orang yang akan mengoperasikan atau orang yang akan berinteraksi dengan sistem aplikasi.

 Use Case	Use-Case merepresentasikan operasi-operasi yang dilakukan oleh actor.
	Interaksi atau hubungan antara Actor dan Use-Case di dalam sistem yang mencakup hubungan asosiasi dan generalisasi.

2.4.2 Activity Diagram

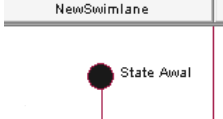

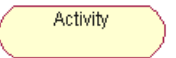
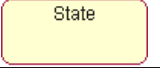


Activity Diagram menggambarkan cara untuk memodelkan *workflow* dari proses bisnis atau cara untuk memodelkan pengoperasian class. Diagram ini mirip dengan *flowchart* karena dalam memodelkan *workflow* dapat dilakukan dari *activity* ke *activity* atau dari *activity* ke *state*.

Workflow

Masing-masing *activity* menggambarkan performansi dari kelompok “action” dalam sebuah *workflow*. Ketika *activity* sudah lengkap, *flow control* pindah ke *activity* berikutnya atau kondisi transisi. Jika pada kondisi transisi tidak terpenuhi, maka akan dipenuhi oleh action lainnya di dalam *activity*. Keunikan fitur *Activity Diagram* adalah pada swimlane yang mendefinisikan “siapa” atau “apa” yang bertanggung jawab untuk menjalankan *activity* atau state. Workflow berhenti ketika transisi sampai pada state akhir.

Pada *Activity Diagram*, dapat ditampilkan banyak elemen pemodelan kecuali atribut, asosiasi, atau model elemen lainnya yang tampil dalam komponen tampilan. Komponen *Activity Diagram* ditunjukkan pada Tabel 2.4.

Tabel 2.4 Komponen *Activity Diagram*

Gambar	Penjelasan
	Swimlane yaitu pemisahan tanggung jawab dari setiap aktifitas.
	Kondisi Awal dan Kondisi Akhir dari proses aktifitas.
	<i>Activity</i> menggambarkan proses aktifitas yang terjadi.
	<i>State</i> menyatakan suatu kondisi.
	<i>Decisions</i>
	Transisi menggambarkan arah alur aktifitas.

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"> - Melihat materi pembelajaran - <i>Download</i> materi pembelajaran 	<ul style="list-style-type: none"> - Pengolahan materi pembelajaran - <i>Upload</i> materi pembelajaran
Administrator / Dosen	<ul style="list-style-type: none"> - Melihat materi pembelajaran - <i>Download</i> materi pembelajaran - Pengolahan materi pembelajaran - <i>Upload</i> materi pembelajaran 	- 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.

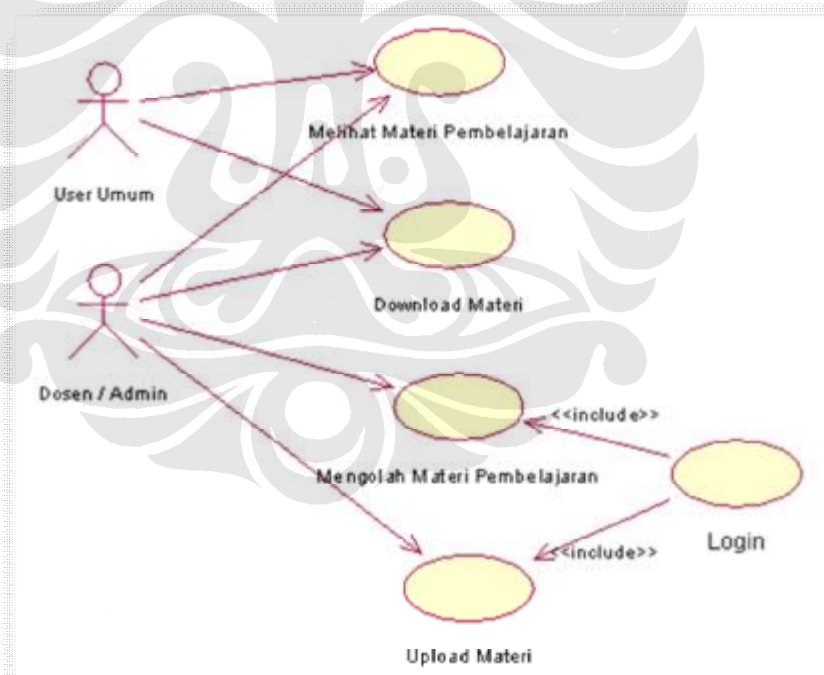
3.2.2 Batasan Perancangan

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

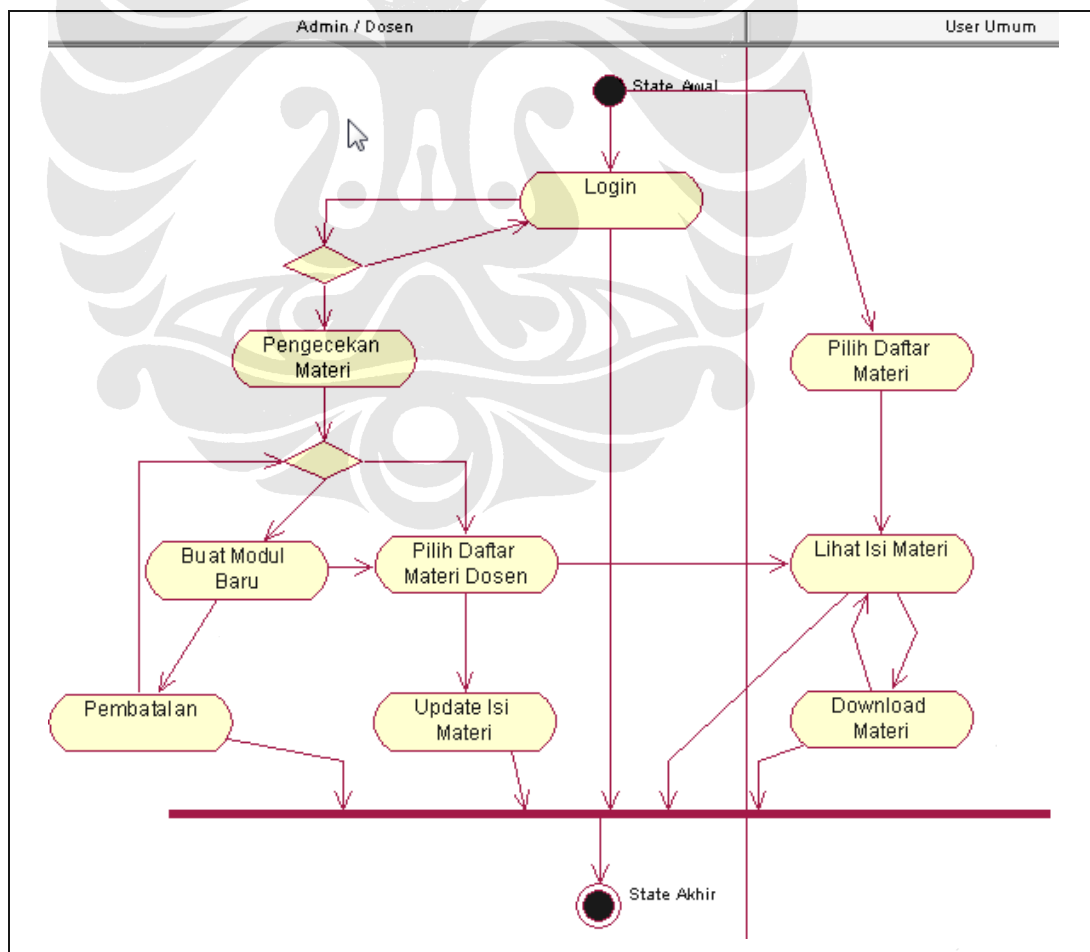
Nama	<i>Upload</i> Materi
Deskripsi Singkat	Dosen melakukan <i>upload</i> materi
Aktor	Dosen / Admin
Kondisi awal	Materi belum ter- <i>upload</i>
Kondisi akhir	Mater berhasil di- <i>upload</i>

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.

BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi

Pada tahap ini, dilakukan pembuatan perangkat lunak dengan mengikuti dan menerapkan semua yang dirancang pada tahap perancangan ke dalam bahasa pemrograman *Ruby on Rails* yang berbasis web. Selain itu, pada tahap ini juga dilakukan pembuatan materi pembelajaran sebagai konten *e-learning* dari aplikasi perangkat lunak yang dibuat serta pemanfaatan teknologi Zune/iPod sebagai alat pendukung pembelajaran.

4.1.1 *Prototyping*

Untuk tujuan implementasi dibuat *prototype* program yang dapat mewakili proses input data, pemrosesan data, dan proses output data. Pada pembuatan *prototype* program juga dilakukan pengisian data materi pembelajaran ke dalam *database* yang diperlukan dalam pengujian program aplikasi yang dibuat.

Sasaran dari aplikasi Portal Pembelajaran ini adalah untuk siapa saja yang membutuhkan ilmu pembelajaran dengan tidak terbatas pada tempat dan waktu. Oleh karena itu, aplikasi ini harus memiliki karakteristik tertentu. Karakteristik sistem yang harus dimiliki diantaranya adalah adanya kemampuan dari sistem untuk melayani proses akses dari banyak pengguna dan mendukung kerjasama antara pengguna sistem. Untuk memenuhi hal tersebut, maka *prototype* aplikasi portal ini harus menggunakan teknologi yang mampu menangani kebutuhan tersebut. Alternatif teknologi yang memungkinkan pada saat ini dalam menangani kebutuhan tersebut adalah teknologi Internet.

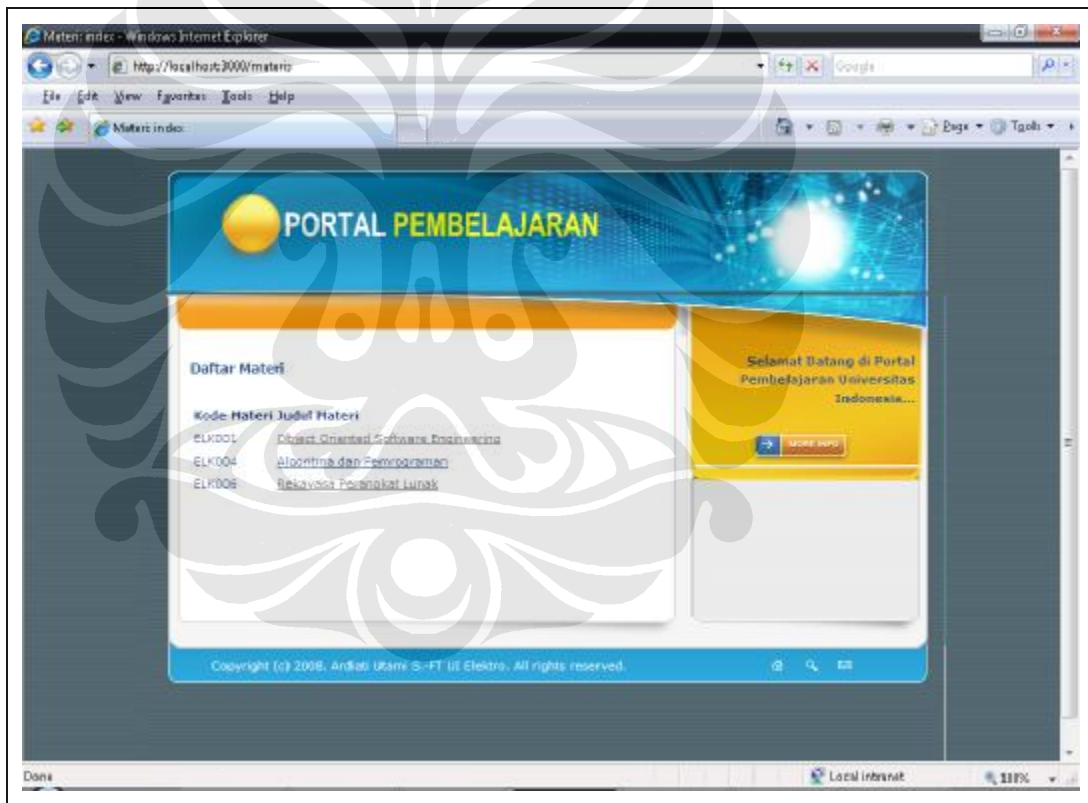
Teknologi Internet dibutuhkan untuk mengembangkan sistem ini karena teknologi ini melibatkan jaringan komputer dalam skala yang sangat besar serta dapat diakses kapanpun pada hampir setiap tempat yang ada di bumi ini. Pengembangan aplikasi ini melibatkan usaha-usaha untuk membangun aplikasi berbasis Internet, membangun *server web* dengan konfigurasi yang sesuai.

Software yang mendukung pembuatan aplikasi ini meliputi:

- Database Engine MySQL 5.0
- Aptana RadRails sebagai *editor script* (IDE) pemrograman Ruby
- Mozilla Firefox/Internet Explorer/*browser* lainnya sebagai *browser* untuk menampilkan halaman web portal.
- Webrick sebagai web *server* dalam *Internetworking* (*localhost*)
- Windows 9x atau versi di atasnya sebagai *platform* sistem operasi yang digunakan dalam pembuatan aplikasi

4.1.2 Graphical User Interface (GUI)

4.1.2.1 Halaman Utama untuk *User* Umum



Gambar 4.1 Halaman Utama untuk *User* Umum

Halaman ini dapat diakses oleh semua *user*, dimana dari halaman ini *user* dapat mengakses materi-materi yang telah disajikan. Khusus pada halaman utama Admin/Dosen disediakan modul khusus untuk *login*. *Login* ini dapat digunakan oleh *user* Admin/Dosen jika akan mengolah data materi yang akan disajikan pada Portal Pembelajaran.

4.1.2.2 Halaman Materi untuk *User* Umum

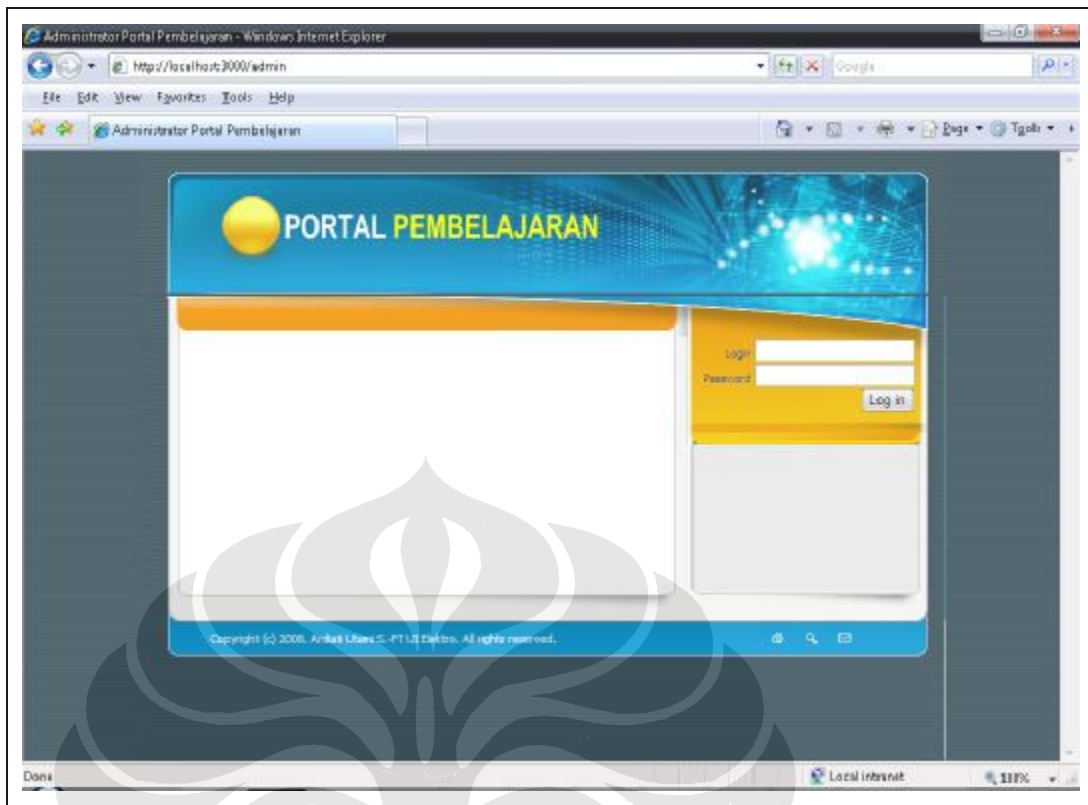


Gambar 4.2 Halaman Materi untuk *User* Umum

Halaman materi ini akan menyajikan detail materi yang telah dipilih oleh *user* pada halaman utama. Detail materi ditampilkan dalam bentuk menu yang dapat diakses selama *user* tidak keluar dari halaman materi atau kembali ke halaman utama *user*. Pada menu materi salah satunya disediakan akses bagi *user* untuk melakukan *download* terhadap materi yang diinginkan.

4.1.2.3 Halaman Utama *Login* Admin

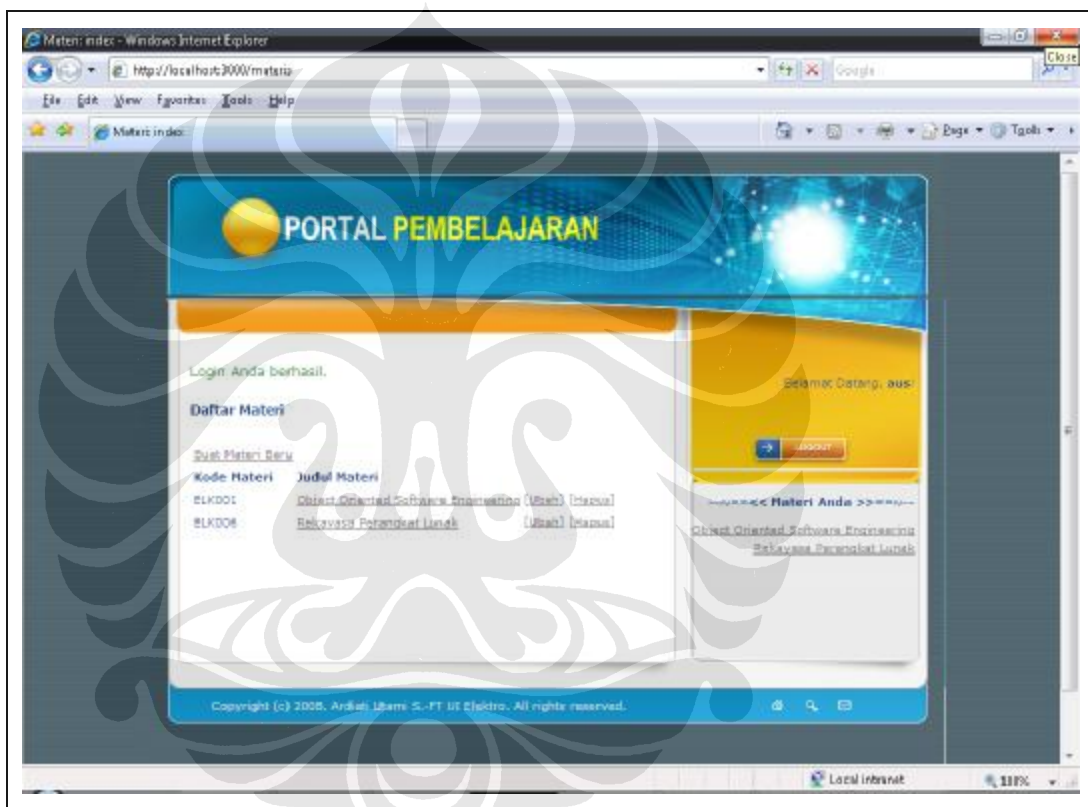
Halaman utama untuk Admin/Dosen ini menyediakan modul khusus untuk *login*. *Login* ini dapat digunakan oleh *user* Admin/Dosen untuk mengolah data materi yang akan disajikan dalam Portal Pembelajaran sesuai dengan materi yang dibuat oleh *user* tersebut. Pada halaman utama *login* admin ini akan menampilkan semua materi yang ada di *database*. Data materi yang ditampilkan sama dengan materi yang ditampilkan pada Halaman Utama *User* Umum sehingga disini admin hanya memiliki hak akses sebagai *user* umum.



Gambar 4.3 Halaman Utama *Login Admin*

4.1.2.4 Halaman Utama untuk *User Admin/Dosen*

Halaman ini hanya dapat diakses oleh *user* admin/dosen dengan cara melakukan *login* pada halaman utama *login* admin. Halaman utama untuk admin ini akan menampilkan sejumlah materi yang telah dibuat oleh *user* admin tersebut. Daftar materi yang dibuat oleh *user* admin lainnya tidak akan ditampilkan pada halaman ini. Pada setiap materi yang ditampilkan disediakan link untuk melakukan pengolahan terhadap materi baik ubah data materi maupun hapus data materi. Selain link untuk melakukan pengolahan terhadap materi, disediakan pula link untuk membuat materi baru.



Gambar 4.4 Halaman Utama untuk *User* Admin/Dosen

4.1.2.5 Halaman Materi untuk *User* Admin/Dosen



Gambar 4.5 Halaman Materi untuk *User Admin/Dosen*

Halaman materi ini akan menyajikan detail materi yang telah dipilih oleh *user* pada halaman utama. Detail materi ditampilkan dalam bentuk menu yang dapat diakses selama *user admin* tidak keluar dari halaman materi atau kembali ke halaman utama *user admin*. *User admin* diizinkan untuk melakukan perubahan terhadap isi materi sehingga pada menu materi ini disediakan link untuk melakukan pengolahan terhadap isi materi.

4.1.3 Pengembangan Konten *e-Learning*

Untuk memenuhi kebutuhan (*requirement*) pengguna (*user*), sistem portal ini menyediakan fitur-fitur berikut ini:

- a) Informasi unit-unit dalam materi pembelajaran
 - Pendahuluan (tujuan dan sasaran)
 - Jadwal pembelajaran
 - Sistem penilaian
- b) Isi materi pembelajaran
 - Modul dalam bentuk tulisan dan gambar pada halaman web
 - Latihan dalam bentuk tulisan dan gambar pada halaman web
 - Bahan materi pembelajaran dalam bentuk *file* pdf/word, presentasi (ppt), dan video yang dapat dijalankan menggunakan alat multimedia *portable* seperti iPod/Zune.
 - Informasi sumber referensi

Dalam tugas akhir ini, studi kasus materi pembelajaran yang dibahas adalah tentang “*Object Oriented Software Engineering* menggunakan PHP 5” yang kemudian akan disingkat menjadi OO-SE. Materi pembelajaran yang disajikan akan lebih ditekankan pada *file* presentasi dan video pembelajaran untuk *podcasting*.

File presentasi yang disajikan sama seperti *file* presentasi pada umumnya. Namun pada tugas akhir ini, materi OO-SE lebih memaksimalkan fungsi fitur yang telah disediakan oleh Microsoft Office Powerpoint 2007 sebagai *tools* untuk membuat *file* presentasi. Tujuan dari memanfaatkan fitur-fitur ini adalah untuk menyajikan materi yang menarik dan memudahkan pengguna untuk mempelajari materi yang disajikan tersebut.

File presentasi dirancang dengan memadukan antara *file* video dengan slide presentasi. Video disisipkan ke dalam slide dan diatur agar berjalan terus selama slide presentasi berjalan. Slide presentasi dirancang menggunakan animasi tampilan serta diatur waktu untuk setiap perpindahan slide-nya. Hal ini bertujuan supaya pengguna tidak perlu menjalankan slide secara manual.

Video yang telah dibahas sebelumnya merupakan video yang berisi penjelasan tentang materi yang diberikan. Dalam studi kasus materi OO-SE ini, pembuatan video masih dilakukan dengan cara yang sederhana. Video dibuat dengan menggunakan kamera digital Sony dengan resolusi 7,1 MP. Kamera ini memuat eksternal memori berkapasitas 1 GB. Untuk kapasitas 1 GB tersebut hanya dapat menghasilkan video (dengan format .avi) yang berdurasi 8 s.d 9 menit. Karena keterbatasan memori yang tersedia tersebut, dalam uji coba membuat video berdurasi 27 menit dilakukan pengambilan video sampai dengan empat kali sehingga menghasilkan empat buah *file* video. Untuk menggabungkan empat *file* video tersebut menjadi satu *file*, perlu digunakan sebuah *software* khusus untuk pengolahan video. Dalam hal ini, penulis menggunakan Ulead Video Studio versi 11 untuk menggabungkan *file-file* video tersebut menjadi satu *file* dengan format mp4.

Video yang telah dijelaskan sebelumnya tidak hanya digunakan sebagai pelengkap slide presentasi materi, tetapi juga dapat di-*download* oleh pengguna melalui fitur *podcasting* untuk dijalankan pada berbagai alat multimedia khususnya Zune/iPod. Untuk pembahasan mengenai teknologi Zune/iPod akan dibahas pada sub bagian selanjutnya.

4.1.4 Pemanfaatan Teknologi Zune/iPod sebagai Sarana Penunjang Pembelajaran

Pada umumnya, video pembelajaran dijalankan menggunakan media *player* pada *Personal Computer* (PC) yang bersifat *standalone*. Hal ini dapat membatasi waktu dan tempat pengguna untuk mempelajari materi yang disajikan. Untuk mengatasi keterbatasan tersebut, dibutuhkan suatu media yang dapat menunjang pembelajaran tanpa terbatas waktu dan tempat.

Zune/iPod adalah salah satu alat multimedia yang mudah dibawa kemana-mana. Zune maupun iPod memiliki fungsi utama yang sama yaitu sebagai alat *media player*. Perbedaan hanya terletak pada perusahaan yang memproduksi alat tersebut dimana Zune diproduksi oleh *Microsoft*, sedangkan iPod oleh *Apple*. Selain itu, masing-masing alat dikemas dengan berbagai fitur agar dapat menarik perhatian masyarakat pengguna. Sampai sejauh ini, Zune/iPod dimanfaatkan oleh sebagian besar masyarakat sebagai sarana hiburan yaitu untuk menjalankan musik maupun

video kesukaan. Pada tugas akhir ini penulis menjadikan Zune/iPod tidak hanya sebagai sarana hiburan, tetapi juga sebagai salah satu alternatif sarana pembelajaran yang *portable* (mudah dibawa) serta tidak terbatas waktu dan tempat.

Fitur yang dimiliki oleh Zune/iPod sangat bermanfaat dalam dunia pembelajaran. Kelebihan Zune/iPod yang mudah dibawa kemana-mana memungkinkan pengguna umum sistem PP-OOP ini menjalankan video pembelajaran secara *mobile* dan tidak terbatas waktu serta tempat. Sedangkan bagi pengguna yang bertindak sebagai admin/dosen, terdapat beberapa hal yang harus diperhatikan untuk membuat suatu video pembelajaran agar dapat didukung oleh Zune/iPod.

Dalam membuat video ini diperlukan pengetahuan mengenai tipe *file* dengan format apa saja yang didukung oleh Zune. Hal ini disebabkan tidak semua tipe format multimedia didukung oleh Zune. Tipe *file* multimedia yang didukung oleh Zune diantaranya adalah m3u, m4u, m4b, mp3, mp4, wma, wmv, zpl. Kemudian tidak semua *file* tersebut dapat langsung disinkronisasikan ke Zune, perlu dilakukan proses konversi kembali ke dalam format Zune. Proses konversi ini membutuhkan waktu yang cukup lama.

Sebagai contoh, penulis melakukan pembuatan video pembelajaran untuk mendukung *file* presentasi yang telah dibahas pada bagian sebelumnya. *File* yang telah dibuat berformat mp4 dengan durasi 27 menit dengan ukuran *file* 180 MB. Jika *file* bertipe mp4 ini langsung disinkronisasi ke Zune, *file* ini tidak terdeteksi. Hal ini disebabkan *file* bertipe mp4 ini masih belum dikonversi ke dalam format Zune. Untuk melakukan konversi ke dalam format Zune, diperlukan *software* khusus untuk mengonversi *file* video. Penulis melakukan dua uji coba menggunakan dua *software* yaitu Total Video Converter dan iTunes. Dengan menggunakan Total Video Converter dibutuhkan waktu 29 menit untuk melakukan konversi *file* berdurasi 27 menit dengan hasil akhir *file* berukuran 103 MB. Sedangkan menggunakan iTunes, dibutuhkan waktu 38 menit dengan hasil akhir *file* berukuran 165 MB.

Video yang telah dibuat di atas kemudian dimanfaatkan sebagai isi dari fitur yang diberikan oleh sistem PP-OOP. Bagi *user* yang bertindak sebagai admin/dosen,

file video yang berisi materi pembelajaran akan di-*upload* ke sistem PP-OOP melalui Internet. Sedangkan bagi *user* umum, *file* video pembelajaran tersebut akan di-*download*. Oleh karena itu, dalam membuat *file* video perlu dipertimbangkan ukuran dari *file* video yang akan dijadikan materi pembelajaran. Semakin besar ukuran *file* video yang akan di-*upload/download*, semakin tinggi kebutuhan *bandwidth* yang dibutuhkan baik dari sisi klien maupun *server* untuk memenuhi proses tersebut.

4.2 Pengujian Aplikasi PP-OOP oleh *User*

Pada tahap ini, dilakukan pengujian terhadap sistem PP-OOP. Pengujian dilakukan dengan melakukan simulasi aplikasi oleh 10 orang *user* yang bertindak sebagai penguji. Sepuluh penguji ini kemudian memberikan umpan balik terhadap aplikasi sistem PP-OOP dengan menjawab pernyataan pada kuisisioner yang disediakan. Pernyataan pada kuisisioner dibagi menjadi tiga kategori, yaitu tanggapan mengenai pengetahuan terhadap bahasa pemrograman, karakteristik sistem, dan terakhir mengenai pendapat umum aplikasi dan teknologi yang digunakan pada tugas akhir ini. Nilai pada setiap kategori kuisisioner disajikan dalam skala empat.

Tujuan dari pengumpulan data umpan balik ini adalah untuk mencari nilai *confidence interval* 95% berdasarkan rata-rata sampel populasi. Untuk mencari nilai *confidence interval* 95% ini digunakan rumus sebagai berikut :

$$\bar{x} \pm 1.96 \left(\frac{\sigma}{\sqrt{n}} \right)$$

Dimana: x = nilai rata-rata (*mean*)
 σ = standar deviasi
 n = jumlah sampel penguji

Setelah nilai *confidence interval* 95% diketahui berdasarkan jumlah sampel n=10, hasil yang diperoleh digambarkan dalam bentuk grafik. Grafik yang ditampilkan akan dikelompokkan berdasarkan masing-masing kategori.

4.2.1 Kategori Tanggapan terhadap Bahasa Pemrograman

Data yang diperoleh dari 10 user tentang pengetahuan user mengenai bahasa pemrograman ditampilkan pada Tabel 4.1 berikut ini:

Tabel 4.1 Data Tanggapan Penguji terhadap Bahasa Pemrograman

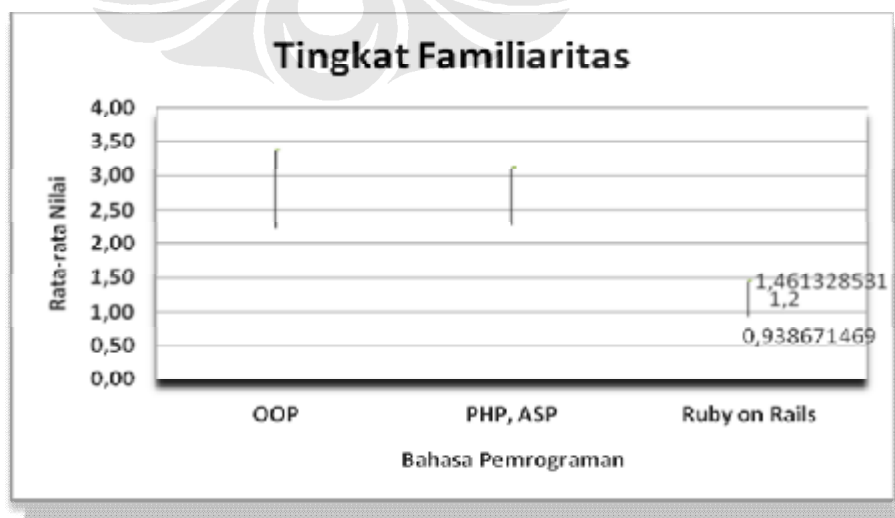
No	Pernyataan	Tanggapan Penguji Berdasarkan Nilai			
		1	2	3	4
1.	Anda familiar dengan bahasa pemrograman berorientasi objek (<i>Object Oriented Programming</i>)	0	5	2	3
2.	Anda familiar dengan bahasa pemrograman berbasis web (seperti PHP, ASP)	0	4	5	1
3.	Anda familiar dengan Ruby on Rails	8	2	0	0
Skala Nilai:					
1. Sangat tidak familiar (tidak pernah dengar)					
2. Tidak familiar (pernah dengar, tapi tidak pernah menggunakan)					
3. Familiar (pernah dengar, pernah menggunakan)					
4. Sangat familiar (sering menggunakan)					

Berdasarkan data tanggapan pada Tabel 4.1, maka diperoleh hasil perhitungan sebagai berikut:

Tabel 4.2 Hasil Pengolahan Data Berdasarkan Bahasa Pemrograman

No	Pernyataan	Hasil Perhitungan Nilai		
		Mean (x)	Σ	Confidence Interval 95%
1.	<i>Object Oriented Programming (OOP)</i>	2,8	0,918936583	2,8 ± 0,569552329
2.	PHP, ASP	2,7	0,674948558	2,7 ± 0,418329763
3.	Ruby on Rails	1,2	0,421637021	1,2 ± 0,261328531

Dari data pada Tabel 4.2, maka dapat digambarkan grafik sebagai berikut:



Gambar 4.6 Grafik Tanggapan Berdasarkan Bahasa Pemrograman

Dari grafik pada Gambar 4.1 dapat dilihat bahwa rata-rata *user* pengujian memiliki pengetahuan mengenai bahasa pemrograman. Dalam grafik terlihat bahwa sebagian besar *user* mengetahui dan paham konsep OOP. Tidak sedikit pula *user* pernah, bahkan sering menggunakan bahasa pemrograman web seperti PHP dan ASP. Namun bahasa pemrograman Ruby on Rails terlihat penurunan yang sangat tinggi yang menandakan bahwa kebanyakan *user* belum pernah mendengar bahasa Ruby on Rails meskipun sebagian besar paham konsep OOP. Hal ini disebabkan kurangnya informasi umum seperti buku referensi. Pengetahuan mengenai Ruby on Rails umumnya banyak diperoleh dari forum komunitas di Internet dimana banyak informasi didapat dari pengalaman banyak orang yang telah menggunakan Ruby baik dalam negeri maupun dari luar negeri. Sehingga hanya sebagian kecil orang yang menggali informasi mengenai Ruby di Internet.

4.2.2 Kategori Tanggapan terhadap Karakteristik Sistem

Data yang diperoleh dari 10 *user* teterhadap karakteristik sistem portal ditampilkan pada Tabel 4.3 berikut ini:

Tabel 4.3 Data Tanggapan Pengujian terhadap Karakteristik Sistem

No.	Karakteristik Sistem	Tanggapan Pengujian Berdasarkan Nilai			
		1	2	3	4
4.	Menu Pembuatan menu dari sisi fungsi, pemilihan kata, dan tampilan (GUI) sudah baik.	0	2	6	2
5.	Window Posisi window, perpindahan antar window serta kecepatan perpindahannya sudah baik.	0	0	5	5
6.	Control Control (input keyboard, field input text, button) sudah baik.	0	0	3	7
7.	Warna Pemilihan warna serta pemanfaatannya dalam aplikasi sudah baik.	0	0	6	4
8.	Perilaku Sistem Aplikasi Perilaku sistem mempermudah pengoperasian aplikasi.	0	0	6	4
9.	Bahasa dan Tulisan (Font) Pemilihan bahasa serta penggunaan tulisan baik ukuran, perilaku (tebal, miring, garis bawah) sudah baik.	0	2	5	3
10.	Fitur-fitur Fitur yang disajikan (materi, <i>file</i> pendukung, video pembelajaran dan pemanfaatannya pada teknologi Zune) sudah baik.	0	0	6	4
Skala Nilai: 1. Sangat Buruk					

- | |
|----------------|
| 2. Buruk |
| 3. Baik |
| 4. Sangat Baik |

Berdasarkan data pada Tabel 4.3, maka diperoleh hasil perhitungan sebagai berikut:

Tabel 4.4 Hasil Pengolahan Data Berdasarkan Bahasa Pemrograman

No.	Karakteristik Sistem	Hasil Perhitungan Nilai		
		Mean (x)	Σ	Confidence Interval 95%
4.	Menu	3,0	0,666666667	$3,0 \pm 0,413196688$
5.	Window	3,5	0,527046277	$3,5 \pm 0,326660664$
6.	Control	3,7	0,483045892	$3,7 \pm 0,299389444$
7.	Warna	3,4	0,516397779	$3,4 \pm 0,320060778$
8.	Perilaku Sistem	3,4	0,516397779	$3,4 \pm 0,320060778$
9.	Bahasa dan Tulisan	3,1	0,737864787	$3,1 \pm 0,45732493$
10.	Fitur-fitur	3,4	0,516397779	$3,4 \pm 0,320060778$

Dari data pada Tabel 4.4, maka dapat digambarkan grafik sebagai berikut:



Gambar 4.7 Grafik Tanggapan Berdasarkan Karakteristik Sistem

Berdasarkan grafik hasil tanggapan *user* berdasarkan karakteristik sistem, dapat dilihat bahwa rata-rata berada di atas 3,00. Ini dapat disimpulkan bahwa aplikasi yang dibuat memenuhi kriteria dari persyaratan karakteristik suatu sistem. Namun masih ada kekurangan dari sisi tampilan menu yang masih minimal karena hanya memanfaatkan fasilitas link yang ada pada Ruby on Rails.

4.2.3 Kategori Pendapat Umum *User*

Data yang diperoleh dari 10 *user* terhadap karakteristik sistem portal ditampilkan pada Tabel 4.5 berikut ini:

Tabel 4.5 Data Tanggapan Umum Aplikasi dan Teknologi

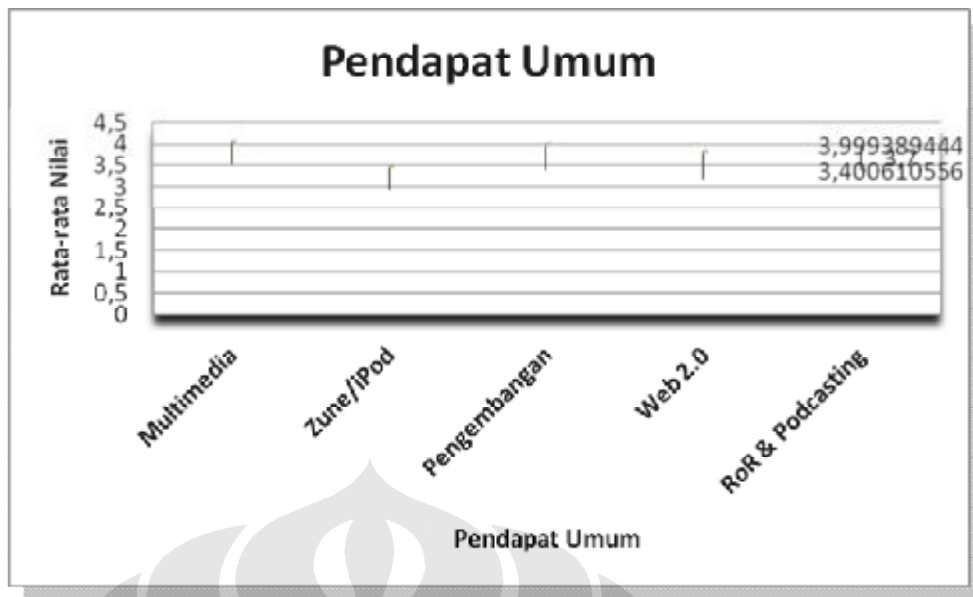
No.	Pernyataan	Tanggapan Penguji Berdasarkan Nilai			
		1	2	3	4
Teknologi Zune/iPod					
11.	Alat multimedia tidak hanya berfungsi sebagai hiburan, tetapi bisa dimanfaatkan di dalam dunia pendidikan.	0	0	2	8
12.	Zune/iPod merupakan salah satu sarana pembelajaran yang baik, praktis, dan mudah dibawa.	0	0	8	2
13.	Zune/iPod bisa dikembangkan di masa depan sebagai sarana pembelajaran <i>offline</i> yang praktis.	0	0	3	7
Teknologi Web 2.0					
14.	Internet berbasis Ruby on Rails merupakan salah satu sarana pembelajaran yang sangat baik	0	0	5	5
15.	Sistem Pembelajaran berbasis Ruby on Rails sangat baik untuk dikembangkan di masa depan.	0	0	3	7
Skala Nilai: 1. Sangat Tidak Setuju 2. Tidak Setuju 3. Setuju 4. Sangat Setuju					

Berdasarkan data tanggapan pada Tabel 4.5, maka diperoleh hasil perhitungan sebagai berikut:

Tabel 4.6 Hasil Pengolahan Data Berdasarkan Pendapat Umum

No	Pernyataan	Hasil Perhitungan Nilai		
		Mean (x)	Σ	Confidence Interval 95%
11.	Multimedia	3,8	0,421637021	3,8 ± 0,261328531
12.	Zune/iPod	3,2	0,421637021	3,2 ± 0,261328531
13.	Pengembangan	3,7	0,483045892	3,7 ± 0,299389444
14.	Web 2.0	3,5	0,527046277	3,5 ± 0,326660664
15.	RoR dan Podcasting	3,7	0,483045892	3,7 ± 0,299389444

Dari data pada Tabel 4.6, maka dapat digambarkan grafik sebagai berikut:



Gambar 4.8 Grafik Tanggapan Berdasarkan Pendapat Umum

Berdasarkan pada grafik di atas, dapat dilihat bahwa umumnya *user* setuju bahwa teknologi web 2.0 serta implementasinya pada berbagai multimedia merupakan sarana yang baik dalam memajukan dunia pendidikan terutama dalam hal penyampaian materi pembelajaran.

BAB 5 KESIMPULAN

Berdasarkan hasil perancangan, implementasi serta pengujian dari sistem Portal Pembelajaran *Object Oriented Programming* berbasis *Ruby on Rails* dan *Podcasting* ini, maka dapat disimpulkan bahwa:

1. *Ruby on Rails* merupakan bahasa pemrograman berbasis web yang mengimplementasikan konsep *Object Oriented Programming* (OOP) dan teknologi AJAX dalam membangun aplikasi.
2. Zune bermanfaat sebagai sarana penunjang pembelajaran. Bagi *user* pembelajaran, Zune sangat bermanfaat sebagai salah satu wadah pembelajaran yang menarik, mudah dibawa kemana-mana serta tidak terbatas pada waktu dan tempat. Sedangkan bagi *developer/administrator*, (misalnya dosen), Zune dapat membantu proses pengajaran menjadi lebih efektif baik dari segi waktu maupun tenaga.
3. Dalam membangun sistem pembelajaran berbasis *podcasting* dibutuhkan sarana dan prasarana yang memadai bagi pembangun sistem untuk menunjang pembuatan video dengan kualitas yang baik.
4. Pada tugas akhir ini telah berhasil dibuat portal pembelajaran *Object Oriented Programming* berbasis *Ruby on Rails* dan *Podcasting* dengan studi kasus *Object Oriented Software Engineering* dalam bentuk teks materi pembelajaran, *slide* presentasi yang menarik, serta video pembelajaran yang menjadi fitur dalam sistem portal.
5. Hasil pengujian yang dilakukan pada 10 koresponden didapatkan bahwa tingkat pengenalan terhadap konsep bahasa pemrograman berorientasi objek sudah tinggi, namun dalam implementasinya terutama pemrograman berbasis web yang berorientasi objek, umumnya masih sangat kecil. Sedangkan tanggapan terhadap pengembangan sistem ini di masa depan bernilai rata-rata di atas 3,00; yang artinya semua koresponden setuju proyek ini dapat dikembangkan lebih lanjut.

DAFTAR ACUAN

- [1] Prihatna, Hengky. *Kiat Praktis Menjadi Webmaster Profesional*. PT Elex Media Komputindo, Jakarta. 2005.
- [2] Fundamentals of Portal, <http://ilmukomputer.com>, diakses tanggal 4 Maret 2008.
- [3] Ruby on Rails, <http://ilmukomputer.com>, diakses tanggal 2 Februari 2008.
- [4] Sam, Ruby., Dave Thomas, and David Heinemeier Hansson. *Agile Web Development with Rails Third Edition*. The Pragmatic Bookshelf, Texas. 2008.
- [5] Ruby on Rails, dipresentasikan pada workshop *Bandung Ruby on Rails*, tanggal 28-29 Juli 2008 di Bandung.
- [6] RadRails, <http://www.apтана.com/studio/download/>, diakses tanggal 4 Maret 2008
- [7] Sunyoto, Andi. *AJAX, Membangun Web dengan Teknologi Asynchronous JavaScript & XML*. PT Andi Offset, Yogyakarta. 2007.
- [8] Aziz, M. Farid. *Object Oriented Programming dengan PHP 5*. PT Elex Media Komputindo, Jakarta. 2006.

DAFTAR REFERENSI

- Aziz, M. Farid. *Object Oriented Programming dengan PHP 5*. PT Elex Media Komputindo, Jakarta. 2006.
- Booch, Grady. James Rumbaugh, and Ivar Jacobson. *The Unified Modeling Language User Guide*. Addison Wesley Longman.Inc, Massachusetts. 1998.
- Prihatna, Hengky. *Kiat Praktis Menjadi Webmaster Profesional*. PT Elex Media Komputindo. Jakarta. 2005.
- Sam, Ruby., Dave Thomas, and David Heinemeier Hansson. *Agile Web Development with Rails Third Edition*. The Pragmatic Bookshelf, Texas. 2008.
- Siswoutomo, Wiwit. *PHP Undercover, Mengungkap Rahasia Pemrograman PHP*. PT Elex Media Komputindo, Jakarta. 2005.
- Sunyoto, Andi. *AJAX, Membangun Web dengan Teknologi Asynchronous JavaScript & XML*. PT Andi Offset, Yogyakarta. 2007.
- Suteja, Bernald Renaldy., Jimmy Anthony Sarapung, and Wilfridus Bambang Triadi Handaya. *Memasuki Dunia e-Learning*. Informatika, Bandung. 2008.