

Arsitektur Model-View-Controller Untuk Pengembangan Aplikasi Bersasis Web

STEVEN LOLONG

(Program Studi Ilmu Komputer Universitas Klabat Manado)

Abstrak

Berkembangnya teknologi informasi dan komunikasi (TIK) mendorong peningkatan kebutuhan akan perangkat lunak (aplikasi). Pengembangan aplikasi membutuhkan suatu penanganan khusus supaya aplikasi tersebut mudah dalam perawatan dan bisa dikembangkan apabila dibutuhkan. Salah satu konsep yang digunakan dalam pengembangan aplikasi adalah konsep yang digunakan dalam pengembangan aplikasi adalah konsep Model-View-Controller (MVC). Konsep MVC merupakan teknik pengembangan aplikasi yang memisahkan antara kode program untuk antar muka, kode program inti dan kode program untuk manipulasi basis data. Setiap bahasa pemrograman yang mengimplementasi konsep MVC mempunyai struktur dan logika yang berbeda. Konsep MVC membantu tim dalam pengembangan dan perawatan aplikasi karena penataan dan pembagian tugas untuk anggota tim yang jelas sesuai dengan keahlian masing-masing.

Kata Kunci: MVC, Pengembangan Aplikasi

Pendahuluan

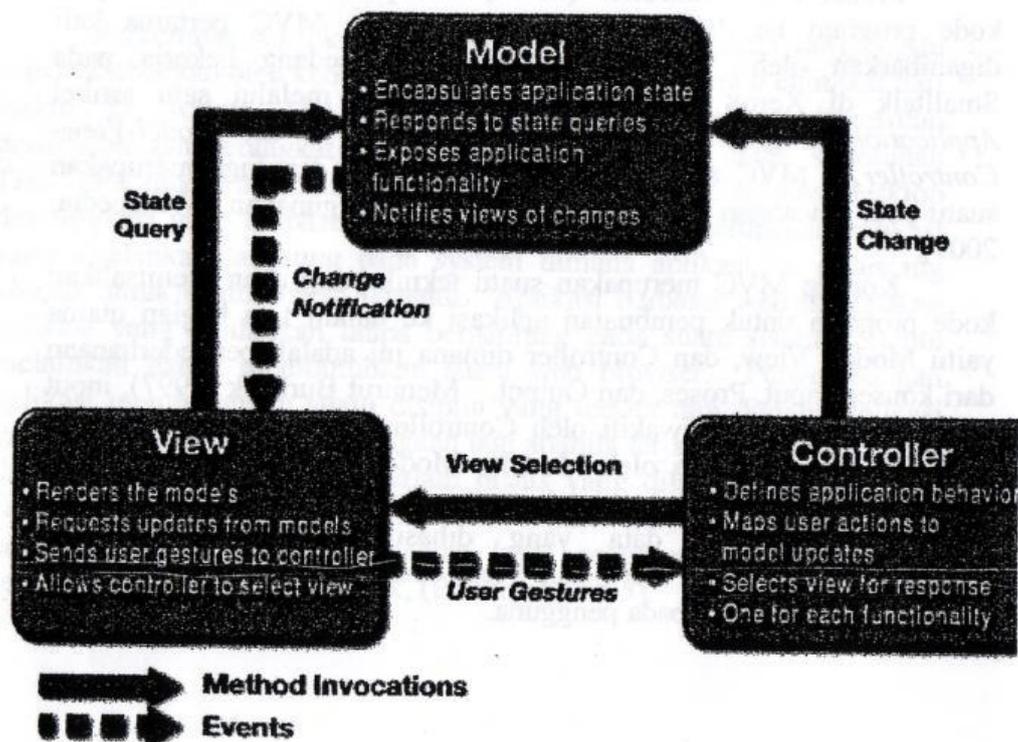
Perkembangan teknologi informasi dan komunikasi yang begitu cepat disebabkan oleh kebutuhan akan teknolo ini meningkat dari waktu ke waktu. hal ini harus diimbangi dengan kecepatan pengembangan perangkat lunak dan perangkat keras untuk kebutuhan TIK. Dua model aplikasi yang sering digunakan yaitu aplikasi desktop dan aplikasi yang berbasis web. Aplikasi desktop merupakan aplikasi yang dijalankan langsung pada system dimana aplikasi ini dirancang khusus untuk suatu sistem tertentu. Aplikasi berbasis web merupakan aplikasi yang dijalankan tanpa bergantung pada suatu sistem tertentu melainkan hanya membutuhkan suatu web browser. Pengembangan aplikasi membutuhkan suatu disiplin yang tertata rapi dimana aplikasi yang dihasilkan bisa dikembangkan lagi apabila terjadi perubahan aturan dan proses yang digunakan dalam basis yang dilayani oleh aplikasi. Salah satu penekanan yang ditekankan dalam rekayasa perangkat lunak adalah pengaturan dan pembagian kode program yang teratur dan terkomentasikan dengan baik, (Pressman, 2005).

Model-View-Controller

Model-View-Controller (MVC) merupakan teknik pembagian kode program ke dalam tiga bagian utama. MVC pertama kali digambarkan oleh Trygve Reenskaug yang sedang bekerja pada Smalltalk di Xerox PARC pada tahun 1979 melalui suatu artikel *Application Programming in Smalltalk-80: How to use Model-View-Controller*. MVC adalah suatu pola arsitektur dan juga merupakan suatu rancangan tergantung bagaimana itu digunakan, (Wikipedia, 2009).

Konsep MVC merupakan suatu teknik yang akan memisahkan kode program untuk pembuatan aplikasi ke dalam tiga bagian utama yaitu Model, View, dan Controller dimana ini adalah penyederhanaan dari konsep Input, Proses, dan Output. Menurut Burbeck (1997), input dan sebagian proses diwakili oleh Controller, Output oleh View, dan sebagian proses lainnya oleh model/ model dalam implementasinya adalah proses yang berhubungan langsung dengan basis data, Controller bertugas memproses data yang dihasilkan oleh Model dan menampilkannya pada View, dan View bertugas memformat tampilan yang akan ditampilkan pada pengguna.

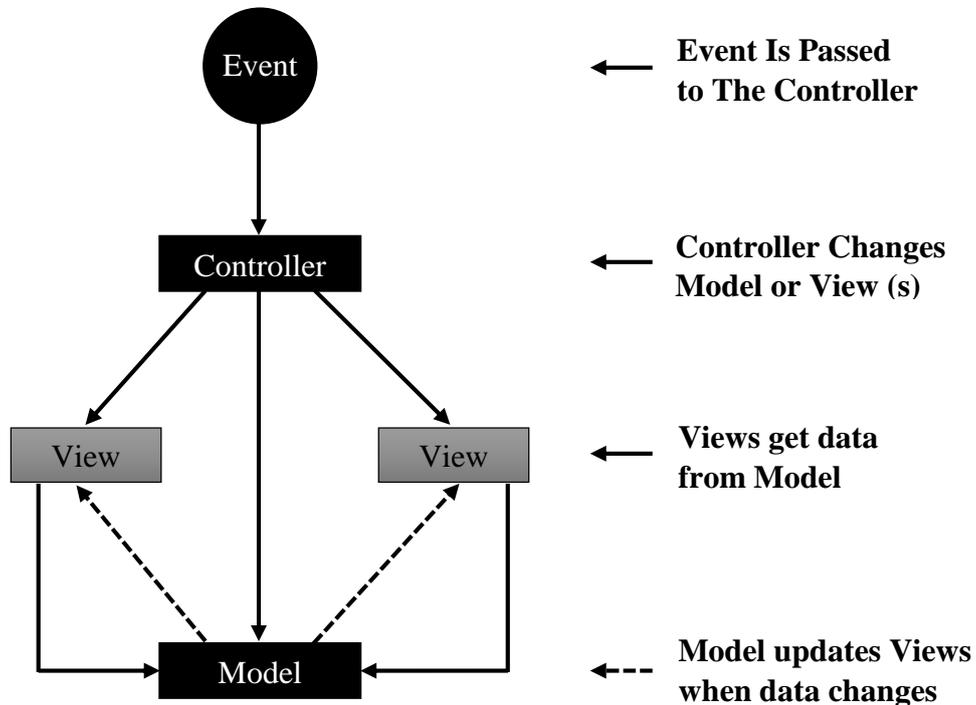
Java.sun.com (2002) menggambarkan hubungan antara MVC sebagai berikut:



Gambar 2.1 – hubungan MVC

- Model, merupakan perwakilan dari data dan aturan bisnis. Model terbagi atas dua jenis yaitu push model dan pull model.
- View, menampilkan isi dari suatu model dimana view mengakses data melalui model.
- Controller, bagian ini berfungsi menerjemahkan aksi yang dihasilkan oleh model ataupun dari view.

Endo.com (2002) menjelaskan hubungan yang terjadi dalam MVC tidak sebatas antara model-view-controller saja tetapi bisa juga terjadi antara controller ke controller yang ada dalam aplikasi. Hal ini akan memudahkan programmer dalam merancang dan mengembangkan aplikasi.



Gambar 2.2 – Pola MVC (Enode.com, 2002)

Implementasi MVC pada Pengembangan Aplikasi Berbasis Web

Bahasa pemrograman web yang sudah menyediakan framework untuk MVC diantaranya adalah Java, NET, PHP dan Ruby. Tiap-tiap bahasa pemrograman tersebut menyediakan framework yang mendukung MVC dengan struktur yang didasarkan pada arsitektur masing-masing bahasa.

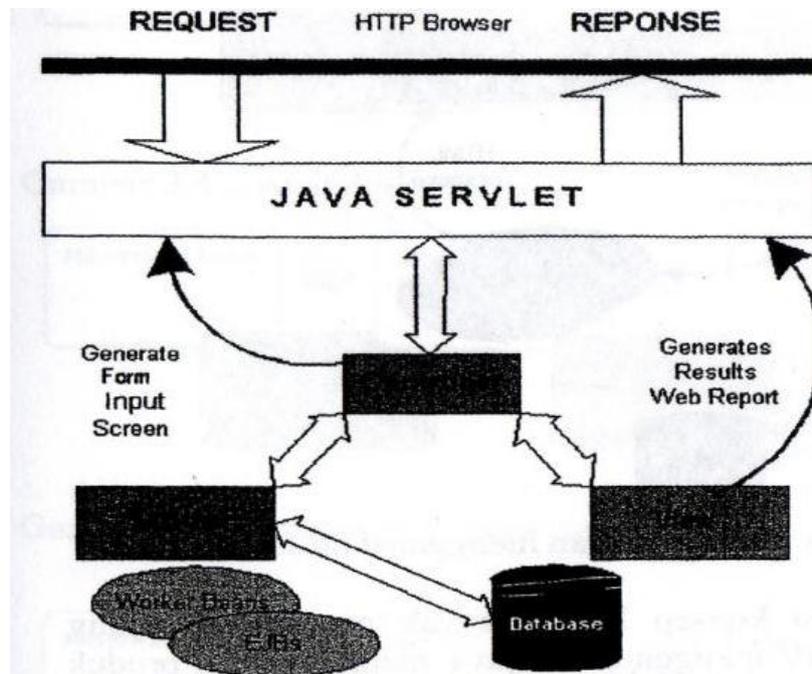
Implementasi MVC pada Java

MVC pada pengembangan aplikasi web menggunakan bahasa pemrograman java dimulai pada saat Sun membuat Java Server Page (JSP) yang akan mempresentasi data pengguna. Menurut Thamara, Leo, & Endy (2006), pembagian konsep MVC pada pemrograman web berbasis java adalah:

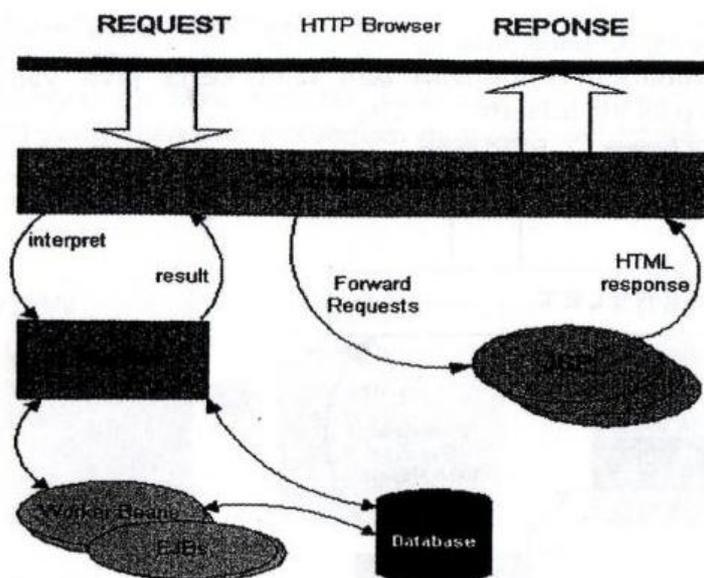


Gambar 3.1 – Implementasi MVC pada Java

Lloyd (2004) menggambarkan hubungan dan cara kerja java yang menimplementasi konsep MVC adalah:

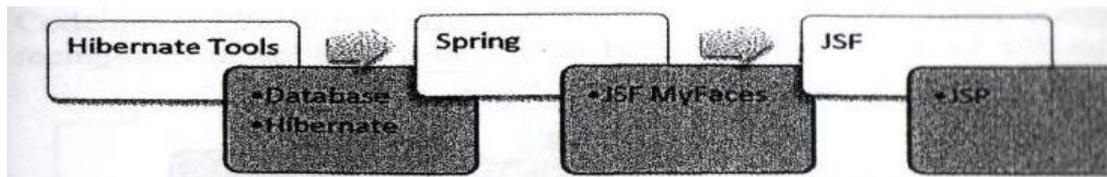


Gambar 3.2 – Java dan MVC

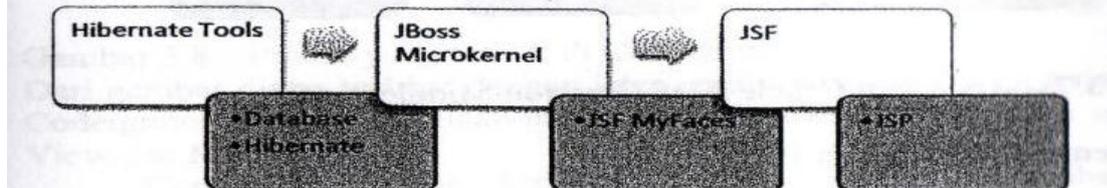


Gambar 3.3 – Implementasi pengembangan menggunakan Java

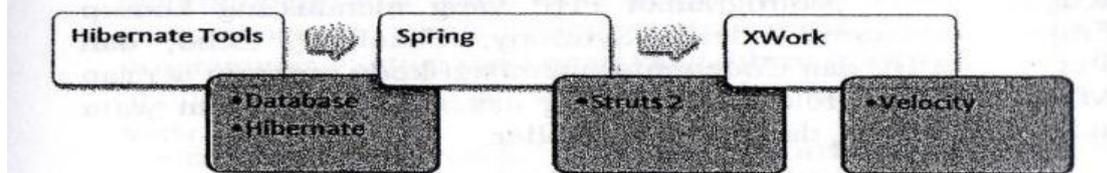
Dengan mengadopsi konsep MVC banyak pengembang yang mengembangkan framework menggunakan java mengeluarkan produk mereka baik yang berlisensi ataupun yang bebas. Perbedaan yang muncul diantara produk-produk adalah teknik rangkaian arsitektur yang didukung oleh java. Empat contoh arsitektur yang mengambill konsep MVC yang sering digunakan oleh pengembang yaitu arsitektur Alfresco, arsitektur Jboss Seam, arsitektur BlueOxygen Cimande, dan arsitektur Oracle ADF (Fussion Middleware), (Thamura, Leo, & Endy, 2006).



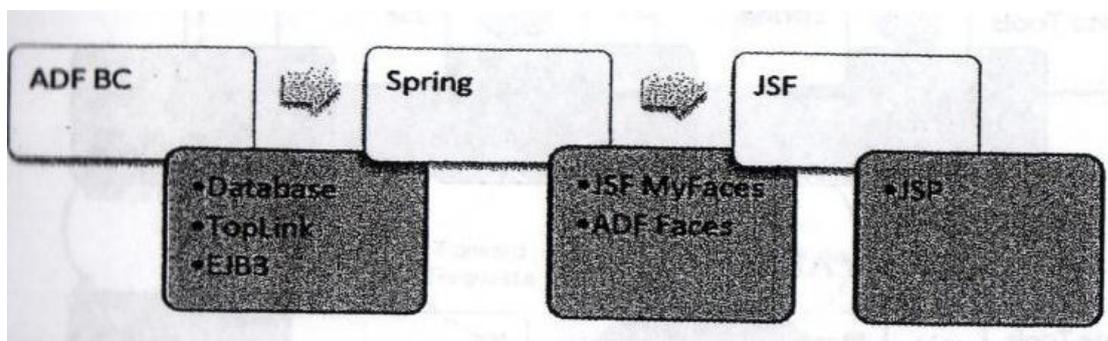
Gambar 3.4 – Arsitektur Alfresco



Gambar 3.5 – Arsitektur J. Boss Seam



Gambar 3.6 – Arsitektur Blue Oxygen Cimande



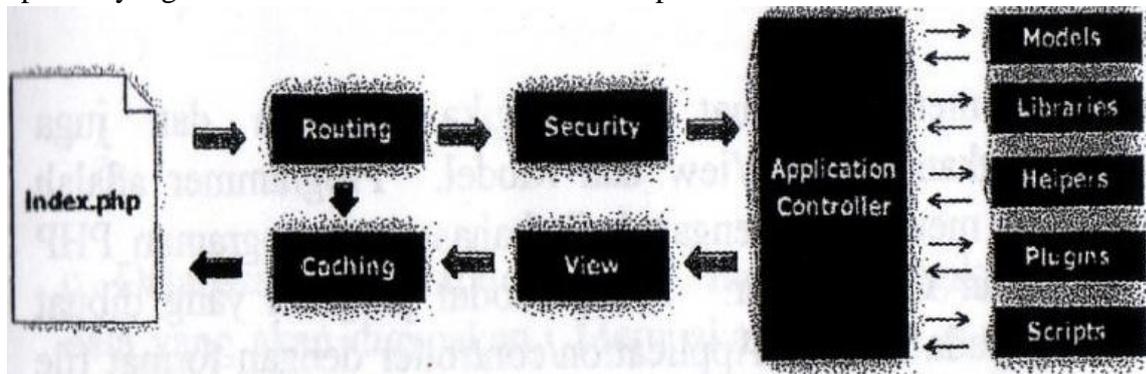
Gambar 3.7 – Arsitektur Oracle ADF (Fusion Middleware)

Implentasi MVC pada PHP

Arsitektur PHP untuk mengimplementasikan konsep MVC lebih ditekankan pada pembagian folder, file, dan kelas untuk tiap-tiap kebutuhan. menurutPhpwatc.org (2009), ada 65 frame work yang dibuat dengan bahasa pemrograman PHP yang mendukung konsep MVC. Empat diantaranya adalah Symfony, CakePHP, Zend, dan Codeigniter. cakePHP dan Codeigniter membagi kode program dengan konsep MVC ke dalam folder-folder yang namanya disesuaikan yaitu folder model, folder view, dan folder controller.

MVC pada Codeigniter

Codeigniter (2008) menggambarkan proses yang terjadi setiap pengguna menggunakan aplikasi yang dibuat berdasarkan framework ini seperti:



Gambar 3.8 – Proses yang terjadi di Codeigniter

Dari gambar diatas terlihat dengan jelas pembagian kerja yang terjadi di Codeigniter dimana Application Controller menjadi pengantara antara View dan Model.

Contoh pembagian kerja programmer untuk pengembangan aplikasi menggunakan Codeigniter adalah:

- a. Perancang antarmuka (Designer), membuat antarmuka dalam bahasa HTML dan PHP tanpa harus menguasai teknik pemrograman PHP yang lebih dalam. Designer merupakan orang yang menguasai teknik perancangan antarmuka bisa bukan seorang programmer. Setiap rancangan yang dibuat diletakkan pada folder: application/view dengan format file php-html. Contoh file untuk view:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"
<html>
<head>
<title>login System</title>
</head>
<body>
<?php echo form_open('login/index');?>
</label>Nama:
<input type="text" id="name" name="name" value="<?php echo set_value('nama');?."
Size="25"/>
</label><br/>
<label>password:
<input type="password" name="password" value="<?php echo set_value('password'); ?>"
size="20"/>
</label><br/>
<?php echo form_submit("Submit","Login");?>
<?php echo form_close(); ?>
</body>
</html>
```

b. Programmer, membuat alur logika program dan juga menghubungkan antara View dan Model. Programmer adalah orang yang menguasai dengan baik bahasa pemrograman PHP dan arsitektur Codeigniter. Setiap modul program yang dibuat diletakkan pada folder: Application/controller dengan file php. Contoh modul untuk controller:

```
<?php
/*
 * Modul for authentication system
 * copyright ©2008 by SL under GPL
 */
Class Login extends Controller{

    function __construct()
    {
        parent::Controller();
        $this->load->library("form_validation");
        $this->load->model('login_model');
    }

    function index()
    {
        $this->form_validation->set_rules('name','Name',required');
        $this->form_validation->set_rules('password','password',required');

        if($this->form_validation->run() == FALSE) {
            $this->load->view('login_view');
        }
        else{
            $dat = $this->login_model($this->input->post('name),$this->input-
            >post('password'));
            if($dat != 'false'){
                $data['dat']=$dat;

                $this->load->view(view('welcome_view',$date);
            }
            else{
                $date['login_error'] = "Error to login";
                $this->load->view('login_view',$data);
            }
        }
    }
}
?>
```

c. Database administrator (DBA), membuat logika untuk basis data yang akan digunakan. Merupakan seseorang yang memahami dengan benar konsep basis data dan mengerti bahasa pemrograman PHP. Setiap model yang dibuat diletakkan pada folder: Application/model dengan format file php. Contoh model yang dibuat:

```
<?php
/*
 *Model for authentication system
 * copyright ©2008 by SL under GPL
 */
class Login_Model extends Model{

    function __constructor(){
        parent::Model();
    }

    function login($id, $password){
        $this->db->where('user_id',$id);
        $this->db->where('password, $password);
        $user_=$this->db->get('user');
        $total= $this->db->count_all_results();
        echo %total;
        if($total > 0){
            return $user_;
        }
        else{
            return 'false';
        }
    }
}
?>
```

Kesimpulan

Implementasi konsep MVC dalam pengembangan aplikasi desktop sudah dijalankan sejak tahun 80an dimana teknik ini dipelopori oleh tim Smalltalk dan kemudian diikuti oleh bahasa pemrograman lainnya seperti Microsoft foundation Class (MFC) untuk C++ yang dibuat tim Microsoft. MVC untuk pemrograman web mulai diminati semenjak ditetapkan standart baru untuk XHTML pada tahun 2002. Dengan mengimplementasikan konsep MVC dalam pemrograman web muka keuntungan yang didapat:

- a. Pembagian tugas antara designer, programmer, dan DBA menjadi sangat jelas. Hal ini memudahkan projek manajer dalam mengontrol perkembangan pekerjaan anggota tim.
- b. Struktur aplikasi menjadi teratur karena dibagi kedalam modul-modul yang lebih kecil yang disesuaikan dengan fungsinya.

- c. Perawatan aplikasi menjadi lebih muda karena apabila ada kesalahan (error) pada aplikasi maka bisa langsung membuka kode program untuk modul yang error pada aplikasi.
- d. Kode program yang terpisah antara rancangan logika program, dan basisdata sehingga memudahkan untuk pengeditan apabila terjadi perubahan logika bisnis.

Setiap bahasa program mempunyai struktur file dan logika yang berbeda dalam mengimplementasikan konsep MVC tetapi mempunyai tujuan yang sama yaitu untuk memisahkan kode program utama, kode program untuk antar muka dan kode program untuk manipulasi basis data.

DAFTAR PUSTAKA

Burbeck, S. (1997, March 4). *Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller (MVC)*. Retrieved January 29, 2006, from st-www.cs.uiuc.edu: <http://st-www.cs.uiuc.edu/users/smarch/st-dics/mvc.html>

Codeigniter. (2005). *Application Flow Chart*. Retrieved December 15, 2005, from Codeigniter Official Web Site: http://codeigniter.com/user_guide/overview/appflow.html

Lloyd, D. (2004). *(What is MVC?)* Retrieved January 20, 2006, from Jcorporate Web site: http://javacorporate.com/expresso/doc/edg/edg_WhatIsMVC.html

Phpwact.org. (2006, January 5). *PHP MVC Frameworks*. Retrieved January 31, 2006, from www.phpwact.org: http://www.phpwact.org/php/mvc_frameworks

Pressman, R.S. (2005). *Software Engineering a Practical Approach* (6th ed.). Grandvile: Mc Graw hill.

Thamura, F., L. H., & E. M. (2006). *Cara Cepat Mengembangkan Aplikasi Java dengan Arsitektur MVC*. Jakarta: Bambumas.

Wikipedia. (2006, January 28). *Model-View-Controller*. Retrieved February 1, 2006, from Wikipedia.org: <http://en.wikipedia.org/wiki/Model-view-controller>