

Single Site Login Portal Web dan Blogging Sederhana dengan CodeIgniter

DOKUMENTASI

Oleh :

**Amelia Riani
132 07 127
Divisi Komputer HME ITB**



**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2009**

LEMBAR PENGESAHAN

Single Site Login Portal Web dan Blogging Sederhana dengan CodeIgniter

DOKUMENTASI

untuk

Oleh :

Amelia Riani
132 07 127
Divisi Komputer HME ITB

Telah disetujui dan disahkan

Bandung, xx xxxx 2010

Telah diperiksa, diuji, disetujui dan disahkan oleh :

Pengaji I

Irham Nurhalim
NIM : 132 07 010

Pengaji II

Mamamama Surama
NIM : 132 05 456

Pengaji III

Nananana Surana
NIM : 132 05 456

Mengetahui,

Ketua Divisi Komputer

Arief Ibrahim
NIM : 132 06 121



**HME
ITB**

Kata Pengantar

Puji syukur dipanjatkan kepada Tuhan Yang Maha Esa segala rahmat dan karunia-Nya sehingga dokumentasi tugas akhir ini dapat diselesaikan. Dokumentasi tugas akhir ini berjudul "Single Site Login Portal Web dan Blogging Sederhana dengan CodeIgniter". Tugas Akhir ini disusun sebagai dokumentasi oprekan kru Divkom Himpunan Mahasiswa Elektro, Institut Teknologi Bandung.

Semoga Tuhan meridhoi pekerjaan saya dan pekerjaan saya dapat bermanfaat bagi khalayak umum.

Bandung, Desember 2010

Penulis

Amelia Riani

Daftar Isi

Lembar Pengesahan	ii
Kata Pengantar	iv
Daftar Isi	V
Bab I Pendahuluan	1
1.1 Latar Belakang	
1.2 Identifikasi Masalah	
1.3 Maksud dan Tujuan Penelitian	
1.4 Batasan Masalah	
1.5 Metodologi Penelitian	
1.6 Sistematika Penelitian	
Bab II Teori Dasar	
Bab III Perancangan dan Implementasi	
Bab IV Hasil Implementasi	
Bab V Kesimpulan dan Saran	
Daftar Pustaka	



**HME
ITB**

Bab I

Pendahuluan

Pada bab ini akan dijelaskan mengenai latar belakang penelitian dan pengembangan, identifikasi masalah, serta batasan-batasan permasalahan. Akan dijelaskan juga mengenai maksud dan tujuan penelitian, serta metodologi dan penjelasan sistematika penulisan.

1.1 Latar Belakang

Untuk membuat sebuah web utuh, akan sulit dan memakan waktu yang lama jika dimulai dari nol. Berbagai solusi telah ditawarkan, diantaranya penggunaan CMS (Content Management System) seperti Wordpress, Joomla, atau Drupal. Akan tetapi penggunaan CMS tersebut kurang fleksibel sehingga diperlukan tools lain. Tool tersebut diantaranya adalah framework . Framework adalah suatu kerangka kerja yang berupa sekumpulan folder yang memuat file-file php yang menyediakan class libraries, helpers, plugins, dan lainnya. Salah satu framework yang sering digunakan adalah CodeIgniter.

1.2 Identifikasi Masalah

Rumusan masalah untuk penelitian dan pengembangan ini akan dijabarkan sebagai berikut :

- a. Apa saja yang diperlukan untuk membuat website menggunakan CodeIgniter?
- b. Bagaimana cara membuat single site login portal web dengan CodeIgniter?
- c. Bagaimana cara membuat blog sederhana dengan CodeIgniter?

1.3 Maksud dan Tujuan Penelitian

Maksud dan tujuan dari penelitian ini akan dijabarkan sebagai berikut :

- a. Mempelajari mengenai Model, View, dan Controller (MVC) pada framework
- b. Mempelajari bagaimana membuat website menggunakan CodeIgniter

1.4 Batasan Masalah

Masalah yang akan diselesaikan adalah pembuatan modul login, logout, entry new blog, dan entry comments.

1.5 Metodologi Penelitian

Dokumentasi ini diselesaikan dengan beberapa tahap yang dijabarkan sebagai berikut :

a. Identifikasi Masalah

Pada tahap ini, penulis merumuskan masalah latar belakang permasalahan yang ada dengan tujuan-tujuan dan batasan masalah.

b. Studi Literatur

Membaca buku dan sumber-sumber on line yang sesuai.

c. Instalasi Tools

Hal pertama yang harus dilakukan adalah menginstal tools yang digunakan seperti XAMPP.

d. Pengujian Hasil

Hasil yang dibuat diuji apakah setelah log in dengan menggunakan username dan password yang ditentukan, pemakai dapat mengupload atau mendownload file yang diinginkan. Untuk mengupload dibutuhkan password untuk mengupload ke file tertentu agar pemakai tidak dapat sembarangan mengupload ke folder orang lain.

1.6 Sistematika Penulisan

Sistematika penulisan dokumentasi ini dijabarkan sebagai berikut :

a. Bab I : Pendahuluan

Bab pertama berisi latar belakang permasalahan dari dokumentasi, pengidentifikasian masalah, maksud dan tujuan penelitian, batasan masalah dalam penelitian. Dan sistematikan penelitian.

b. Bab 2 : Teori Dasar

Bab kedua berisi tentang teori dasar yang melandasi penelitian ini. Teori yang dibahas adalah pengetahuan tentang XAMPP dan PHP.

c. Bab 3: Perancangan dan Implementasi

Bab ketiga ini berisi penjelasan mengenai instalasi XAMPP dan pembuatan *source code* yang digunakan.

d. Bab 4 : Hasil Implementasi

Bab empat berisi tentang hasil dari proses implementasi sistem.

e. Bab 5 : Kesimpulan

Bab terakhir berisi tentang kesimpulan dari penelitian secara keseluruhan.

The logo consists of a shield-shaped emblem. Inside the shield, there is a stylized graphic element resembling a megaphone or a speaker, with a dark gray gradient on the left and a light gray gradient on the right. Below this graphic, the letters "HME" are stacked above "ITB", both in large, bold, black, sans-serif capital letters.

HME
ITB

Bab II

Teori Dasar

Pada bab ini akan dijelaskan teori dasar di antaranya mengenai konsep dasar framework CodeIgniter

2.1 CodeIgniter

2.1.1 Apa itu CodeIgniter?

CodeIgniter adalah sebuah framework pengembangan aplikasi untuk membuat web. CodeIgniter menyediakan berbagai library untuk mempermudah dan mepercepat pembuatan suatu proyek. Keuntungan-keuntungan menggunakan CodeIgniter:

1. Memiliki kompatibilitas luas dengan account hosting standar yang menjalankan berbagai versi PHP dan konfigurasi
2. Tidak perlu menggunakan command line
3. Tidak memerlukan banyak konfigurasi
4. Ada dokumentasi lengkap

2.1.2 Konsep Model-View-Controller (MVC)

Konsep MVC adalah konsep pemisahan antara logic dengan tampilan dan database. Manfaat konsep ini adalah, membuat coding logic lebih simple, karena sudah dipisah dengan code untuk tampilan dan membuat programmer dapat bekerja secara terpisah dengan designer. Programmer mengerjakan logic, sedangkan designer berkutat dengan design dan tampilan.

Model → Merupakan code struktur data. Model berisi fungsi di dalam pengolahan database. Script SQL masuk di sini.

View → Merupakan code untuk menampilkan tampilan suatu program. Tampilan dapat berupa webpage, header, footer dan apa saja yang berjenis tampilan.

Controller → Merupakan code untuk logic, algoritma dan sebagai penghubung antara model, view, dan sumber lain yang diperlukan untuk mengolah HTTP request dan generate web page.

CI menerapkan pola MVC yang flexible, karena model dapat tidak digunakan. Anda dapat hanya menggunakan Controller dan View saja dalam menggunakan CI tanpa Model. Jika anda tidak memerlukan pemisahan di dalam struktur data dan database atau menganggap penggunaan model hanya menambah kompleks aplikasi dengan keuntungan yang kurang sebanding, maka anda dapat tidak menggunakan model.



Bab III

Peracangan dan Implementasi

3.1. Download CodeIgniter (CI)

Hal yang pertama dilakukan adalah mengaktifkan modul Apache dan MySQL pada web server. Disini digunakan XAMPP. Setelah itu download CI di <http://codeigniter.com/>. File tersebut diekstrak ke htdocs XAMPP.

3.2. Pembuatan Database

Dibuat 1 buah database bernama 'Tutorial' yang berisi 3 tabel.

1. Tabel 'users'

Server: localhost > Database: tutorial > Table: users								
Field	Type	Collation	Attributes	Null	Default	Extra	Action	
<input type="checkbox"/> user_id	int(11)			No	None	auto_increment		
<input type="checkbox"/> username	varchar(50)	latin1_swedish_ci		No	None			
<input type="checkbox"/> password	varchar(200)	latin1_swedish_ci		No	None			
<input type="checkbox"/> email	varchar(50)	latin1_swedish_ci		No	None			
<input type="checkbox"/> name	varchar(50)	latin1_swedish_ci		No	None			

2. Tabel 'entries'

Server: localhost > Database: tutorial > Table: entries								
Field	Type	Collation	Attributes	Null	Default	Extra	Action	
<input type="checkbox"/> id	int(11)			No	None	auto_increment		
<input type="checkbox"/> title	varchar(128)	latin1_swedish_ci		Yes	NULL			
<input type="checkbox"/> body	text	latin1_swedish_ci		Yes	NULL			

3. Tabel 'comments'

Server: localhost > Database: tutorial > Table: comments								
Field	Type	Collation	Attributes	Null	Default	Extra	Action	
<input type="checkbox"/> id	int(11)			No	None	auto_increment		
<input type="checkbox"/> entry_id	int(11)			Yes	NULL			
<input type="checkbox"/> body	text	latin1_swedish_ci		Yes	NULL			
<input type="checkbox"/> author	varchar(100)	latin1_swedish_ci		Yes	NULL			

3.3. Pembuatan Modul Login dan Logout

Untuk memudahkan, kita simpan folder CodeIgniter tersebut dengan nama ci.

Pastikan pada root ci dibuat file .htaccess yang berisi:

```
<IfModule mod_rewrite.c>
```

```
RewriteEngine On
```

```

RewriteBase /ci/
RewriteCond %{REQUEST_URI} ^system.*
RewriteRule ^(.*)$ /index.php/$1 [L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php/$1 [L]
</IfModule>

<IfModule !mod_rewrite.c>
    ErrorDocument 404 /index.php
</IfModule>

```

Pada ci->system->application->config->config.php, ubahlah base url menjadi:

```
$config['base_url']      = "http://localhost/ci/";
```

Dan index page menjadi:

```
$config['index_page'] = "";
```

Kemudian lakukan konfigurasi pada ci->system->application->config->routes.php sebagai berikut:

```
$route['default_controller'] = "user";
$route['scaffolding_trigger'] = "";
```

Pada ci->system->application->config->database.php, lakukan perubahan berikut.

```
$db['default']['hostname'] = "localhost";
$db['default']['username'] = "root";
$db['default']['password'] = "";
$db['default']['database'] = "tutorial";
$db['default']['dbdriver'] = "mysql";
```

Untuk membuat modul ini, pertama kita buat file yang disimpan pada folder controller. File tersebut diberi judul user.php. Judul ini harus sesuai dengan class dan function pada file tersebut.

```
<?php
class User extends Controller {
    function User()
    {
        parent::Controller();
    }

    function index()
    {
```

```

        $this->load->view('view_login');
    }

    function main_page()
    {
        if ($this->session->userdata('logged_in'))
        {
            $this->load->view('view_mainpage');;
        }
        else
        {
            redirect ('user/login');
        }
    }

    function login()
    {

        $this->form_validation-
>set_rules('username','Username','required|trim|max_length[50]|xs
s_clean');
        $this->form_validation-
>set_rules('password','Password','required|trim|max_length[200]|x
ss_clean');

        if($this->form_validation->run ()==FALSE)
        {
            $this->load->view('view_login');
        }
        else
        {
            //process their input and login the user
            extract($_POST);

            $user_id = $this->User_model->check_login($username,
$password);

            if(! $user_id)
            {
                //login failed error
                $this->session->set_flashdata('login_error',
TRUE);

                redirect ('user/login');
            }
            else
            {
                $this->session->set_userdata(array(

```

```

        'logged_in'=>TRUE,
        'user_id'=>$user_id
    )) ;

    redirect ('user/main_page') ;
}

}

function logout()
{
    $this->session->sess_destroy();
    redirect ('user/login');
}

?

}

?>

```

Pada ci->system->application->config->autoload.php masukkan library seperti ini:

```
$autoload['libraries'] =
array('form_validation', 'database', 'session', );
```

Dan helper menjadi:

```
$autoload['helper'] = array('form', 'url');
```

Kemudian buat file view_login.php pada folder view.

```
<html>
<head>
<title>Welcome to CodeIgniter</title>

<style type="text/css">

body {
background-color: #fff;
margin: 180px;
font-family: Lucida Grande, Verdana, Sans-serif;
font-size: 14px;
color: #4F5155;
}

a {
color: #003399;
background-color: transparent;
font-weight: normal;
}
```

```
h1 {
    color: #444;
    background-color: transparent;
    border-bottom: 1px solid #D0D0D0;
    font-size: 16px;
    font-weight: bold;
    margin: 24px 0 2px 0;
    padding: 5px 0 6px 0;
}

code {
    font-family: Monaco, Verdana, Sans-serif;
    font-size: 12px;
    background-color: #f9f9f9;
    border: 1px solid #D0D0D0;
    color: #002166;
    display: block;
    margin: 14px 0 14px 0;
    padding: 12px 10px 12px 10px;
}
div#login_form
{
    width:270px;
    margin:40px inherit;
    border:1px solid #acacac;
    padding:10px;
}
</style>
</head>
<body background="../../../Web-Page-Background3.jpg"
style="background-repeat:no-repeat" >

<div id="login_form">
<h1>Welcome to Our Site</h1>

<p>Please log in</p>

<?php echo form_open(base_url() . 'User/login') ;?>

<label>Username</label>
<div>
<?php echo form_input(array('id'=>'username',
'name'=>'username')) ; ?>
</div>
```

```

<label>Password</label>
<div>
<?php echo form_password(array('id'=>'password',
'name'=>'password')) ;?>
</div>
<?php
if ($this->session->flashdata('login_error'))
{
    echo 'you entered an incorrect username or password';
}

echo validation_errors();?>
</br>
<?php echo form_submit(array('name'=>'submit'), 'Login');?>
<?php echo form_close();?>

<?php echo form_open(base_url() . 'signup/index');?>
<?php echo form_submit('name', 'Signup');
//<button name="name" type="button">Signup</button>
?>

<?php echo form_close();?>
</div>
</body>
</html>

```

User.php digunakan untuk memvalidasi masukan. Masukan berupa username dan password. Syarat masukan valid adalah kedua kolom tersebut terisi dengan kapasitas username maksimal 50 karakter dan password (setelah dienkripsi) 200 karakter. Sebelumnya pada database tutorial tabel users telah diisi input yang diinginkan yaitu username ‘amel’ dan password ‘admin’. Masukan dari user akan dicocokkan dengan isi tabel oleh function check_login pada user_model.php. Berikut adalah scriptnya:

```

<?php
class User_model extends Model{

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

    function check_login($username, $password)
    {
        $shai_password = sha1($password);

        $query_str = "SELECT user_id FROM users WHERE username= ?
and password= ?";

```

```

        $result = $this->db->query($query_str, array($username,
$sha1_password));

        if($result->num_rows()==1)
        {
            return $result->row(0)->user_id;
        }
        else
        {
            return false;
        }
    }
?>

```

Setelah berhasil login, kemudian masuk ke mainpage yang berisi pilihan untuk logout dan new blog entry. Tampilan tersebut dibuat pada file view_mainpage.php yang disimpan dalam folder view.

```

<html>
<head>
<title>Welcome to CodeIgniter</title>

<style type="text/css">

body {
background-color: #fff;
margin: 80px;
font-family: Lucida Grande, Verdana, Sans-serif;
font-size: 14px;
color: #4F5155;
}

a {
color: #003399;
background-color: transparent;
font-weight: normal;
}

h1 {
color: #444;
background-color: transparent;
border-bottom: 1px solid #D0D0D0;
font-size: 16px;
font-weight: bold;
margin: 70px 0 2px 0;
padding: 5px 0 6px 0;
}

```

```
code {
    font-family: Monaco, Verdana, Sans-serif;
    font-size: 12px;
    background-color: #f9f9f9;
    border: 1px solid #D0D0D0;
    color: #002166;
    display: block;
    margin: 14px 0 14px 0;
    padding: 12px 10px 12px 10px;
}
div#login_form
{
    width:270px;
    margin:0px auto;
    border:1px solid #acacac;
    padding:10px;
}
</style>
</head>
<body background="../../../../Web-Page-Background3.jpg">

<h1>BlogShare</h1>

<p>Welcome
<?php
//membuat koneksi ke database
$conn=mysql_connect("localhost", "root", "");
mysql_select_db("tutorial");
//membuat query select
$sql="select name from users";
//membaca data
$hasil=mysql_query($sql);
//menampilkan data
$row=mysql_fetch_array($hasil);
//menampilkan field nama
echo "".$row['name'];
?>

</p>
<br>
<br>
<p><?php echo anchor('blog/index/', 'New Blog Entry');?></p>

<p><?php echo anchor('user/logout/', 'Logout');?></p>

</body>
```

```
</html>
```

3.4. Pembuatan Modul Blogging Sederhana

Untuk membuat new blog entry, dibuat file blog.php pada controller sebagai berikut:

```
<?php  
class Blog extends Controller {  
  
    function Blog()  
    {  
        parent::Controller();  
  
        $this->load->helper('url');  
        $this->load->helper('form');  
    }  
  
    function index()  
    {  
        $data['title']="My Blog";  
        $data['heading']="Selamat Datang Dunia!";  
        $data['query']=$this->db->get('entries');  
  
        $this->load->view('blog_view', $data);  
    }  
  
    function blog_insert()  
    {  
        $this->db->insert('entries', $_POST);  
  
        redirect('blog/index/'. $_POST['id']);  
    }  
  
    function comments()  
    {  
        $data['title']="My Comment";  
        $data['heading']="Kata Mereka";  
        $this->db->where('entry_id', $this->uri->segment(3));  
        $data['query']=$this->db->get('comments');  
  
        $this->load->view('comment_view', $data);  
    }  
  
    function comment_insert()  
    {  
        $this->db->insert('comments', $_POST);  
  
        redirect('blog/comments/'. $_POST['entry_id']);  
    }  
} ?>
```

Blog.php terdiri dari beberapa fungsi, diantaranya function index untuk menampilkan data dari tabel entries, function blog_insert untuk memasukkan data ke database, function comments untuk menampilkan komentar dan function comment_insert untuk memasukkan komentar ke dalam database.

1. blog_view.php

```
<html>
<head>
    <title><?php echo $title; ?></title>

    <style type="text/css">

body {
    background-color: #fff;
    margin: 60px;
    font-family: Lucida Grande, Verdana, Sans-serif;
    font-size: 14px;
    color: #4F5155;
}

a {
    color: #003399;
    background-color: transparent;
    font-weight: normal;
}

h1 {
    color: #444;
    background-color: transparent;
    border-bottom: 1px solid #D0D0D0;
    font-size: 24px;
    font-weight: bold;
    margin: 24px 100px 2px 0;
    padding: 5px 0 6px 0;
}

h2 {
    color: #444;
    background-color: transparent;
    font-size: 12px;
    font-weight: bold;
    margin: 24px 0 2px 0;
    padding: 5px 0 6px 0;
}

code {
```

```

font-family: Monaco, Verdana, Sans-serif;
font-size: 12px;
background-color: #f9f9f9;
border: 1px solid #D0D0D0;
color: #002166;
display: block;
margin: 14px 0 14px 0;
padding: 12px 10px 12px 10px;
}

div#login_form
{
    width:270px;
    margin:40px inherit;
    border:1px solid #acacac;
    padding:10px;
}

</style>
</head>
<body background="../../../Web-Page-Background3.jpg"
style="background-repeat:no-repeat">

<h1><?php echo $heading; ?></h1>
<br>

<h2><?php echo anchor('user/logout/', 'Logout'); ?></h2>
<br>

<?php foreach($query->result() as $row) :?>

<h3><?php echo $row->title?></h3>
<p><?php echo $row->body?></p>

<p><?php echo anchor('blog/comments/'.$row->id,
'Comments'); ?></p>
<hr>

<?php endforeach;?>
<?php echo form_open('blog/blog_insert'); ?>

<?php echo form_hidden('id', $this->uri->segment(3)); ?>

New Blog Title
<p><input type="text" name="title"/></p><br>

New Blog Entry
<p><textarea name="body" rows="10"></textarea></p>

```

```
<p><input type="submit" value="Submit Entry"/></p>

</form>

</body>
</html>
2. comment_view.php
<html>
<head>
    <title><?php echo $title;?></title>
    <style type="text/css">

body {
    background-color: #fff;
    margin: 60px;
    font-family: Lucida Grande, Verdana, Sans-serif;
    font-size: 14px;
    color: #4F5155;
}

a {
    color: #003399;
    background-color: transparent;
    font-weight: normal;
}

h1 {
    color: #444;
    background-color: transparent;
    border-bottom: 1px solid #D0D0D0;
    font-size: 24px;
    font-weight: bold;
    margin: 24px 100px 2px 0;
    padding: 5px 0 6px 0;
}

h2 {
    color: #444;
    background-color: transparent;
    font-size: 12px;
    font-weight: bold;
    margin: 24px 0 2px 0;
    padding: 5px 0 6px 0;
}

code {
    font-family: Monaco, Verdana, Sans-serif;
    font-size: 12px;
```

```

background-color: #f9f9f9;
border: 1px solid #D0D0D0;
color: #002166;
display: block;
margin: 14px 0 14px 0;
padding: 12px 10px 12px 10px;
}
div#login_form
{
    width:270px;
    margin:40px inherit;
    border:1px solid #acacac;
    padding:10px;
}
</style>
</head>
<body background="../../../Web-Page-Background3.jpg"
style="background-repeat:no-repeat">
<h1><?php echo $heading; ?></h1>

<?php if ($query->num_rows ()>0) :?>
<?php foreach($query->result() as $row) :?>

<p><?php echo $row->body?></p>
by: <p><?php echo $row->author?></p>

<hr>

<?php endforeach;?>
<?php else: ?>
<p>No comment has been posted for this blog
entry</p>
<hr/>
<?php endif;?>

<p><?php echo anchor('blog', 'Back to Blog'); ?></p>

<?php echo form_open('blog/comment_insert'); ?>

<?php echo form_hidden('entry_id', $this->uri-
>segment(3)); ?>

<br>
Leave your comment<br>
<textarea name="body" rows="10"></textarea>
<br>Your identity, please...<br>
<input type="text" name="author"/>

```

```
<p><input type="submit" value="Submit Comment"/></p>

</form>

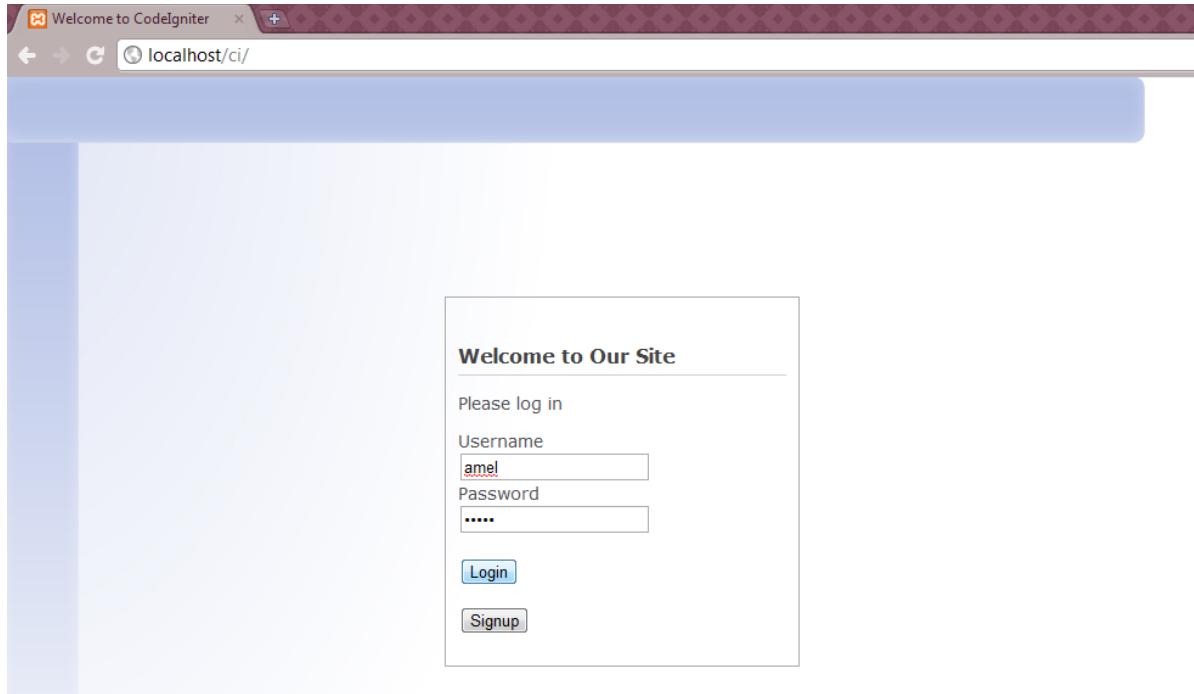
</body>
</html>
```



Bab IV

Pengujian Sistem

3.2 Modul Login dan Logout



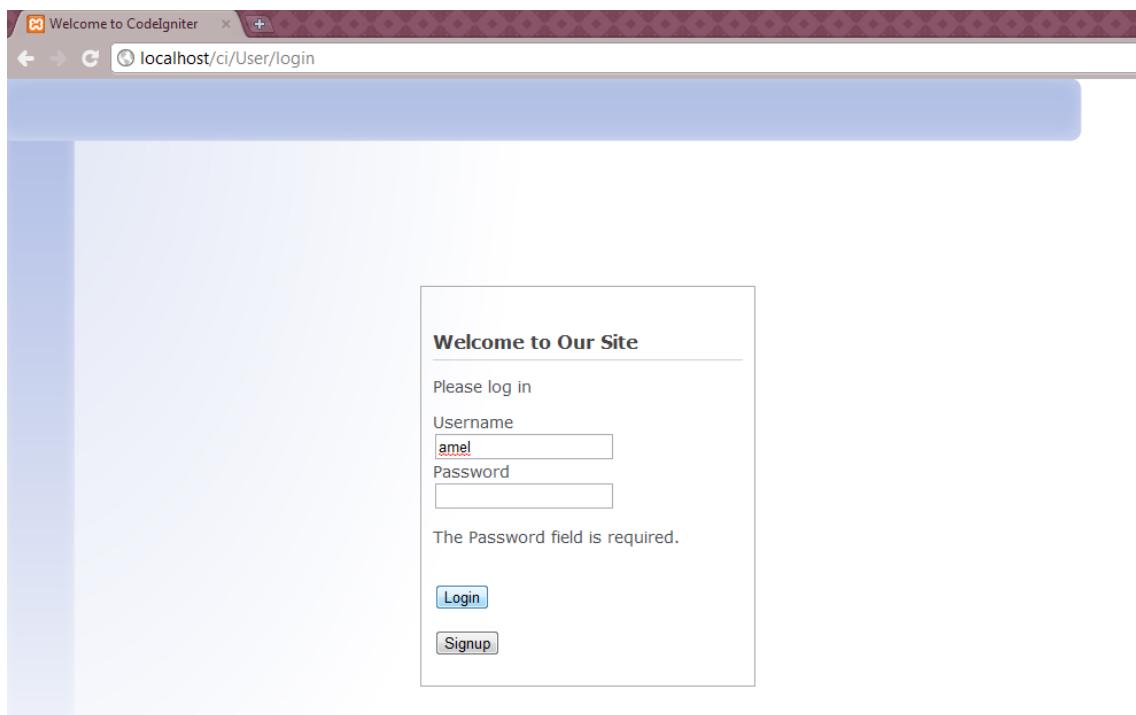
Welcome to Our Site

Please log in

Username

Password

Jika salah satu kolom tidak diisi:



Welcome to Our Site

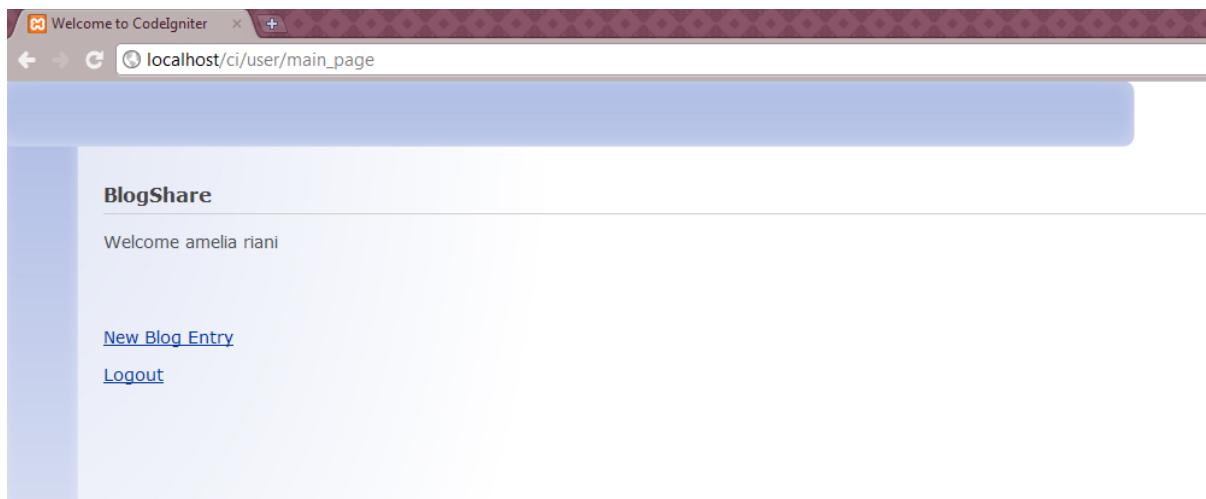
Please log in

Username

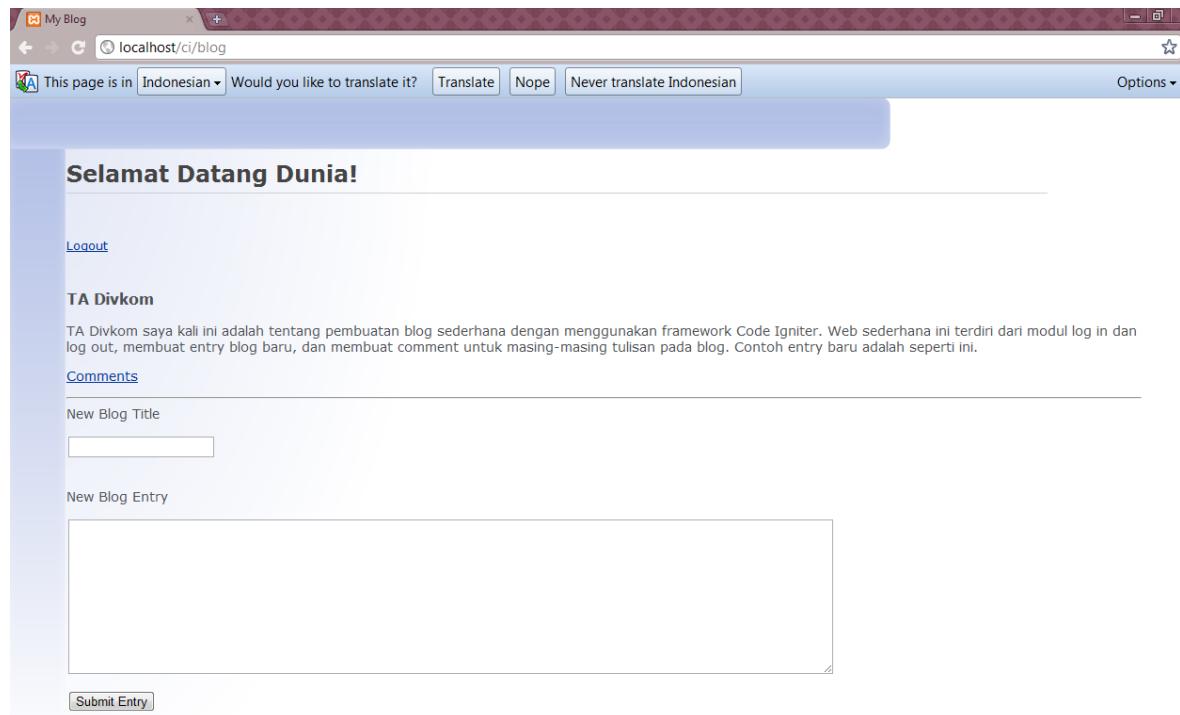
Password

The Password field is required.

Masuk ke mainpage:



Jika memilih logout maka tampilan akan kembali ke gambar 1, jika memilih new blog entry:



Pada page tersebut terdapat form untuk mengisi Judul dan textarea untuk menulis blog. Terdapat pula link untuk logout dan menulis komentar. Jika link logout diklik maka sesi dengan username 'amel' akan didestroy dan page kembali ke gambar pertama. Jika comments diklik, maka akan muncul:

My Comment

localhost/ci/blog/comments/6

Kata Mereka

ini adalah komentar kedua pada tulisan TA Divkom
by:
Riani

ini adalah komentar pertama pada tulisan "TA Divkom"
by:
Amelia

[Back to Blog](#)

Leave your comment

Your identity, please...



**HME
ITB**

Bab V

Kesimpulan dan Saran

Kesimpulan

- Antara Controller, view, dan model pada CI selalu berkaitan satu sama lain, sehingga harus diperhatikan masalah link dari ketiganya.
- Pada CI terdapat banyak library yang dapat digunakan pada pembuatan tugas ini, diantaranya form_validation, database, dan session. Juga terdapat helper diantaranya form dan url.

Saran

- Pembuatan database seefektif harus mungkin
- Sebelum memulai CI, ada baiknya membaca user guidenya terlebih dahulu karena cara mengoding di CI cukup berbeda dengan mengoding biasa.