

Tutorial Codeigniter – Membuat Multi-level Login

Dimas Edu Prasada

edudimas1@gmail.com

<http://samidcorner.blogspot.com>

Lisensi Dokumen:

Copyright © 2003-2013 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Codeigniter menjadi sangat populer berkat kehandalan dan kemampuan untuk membuat sebuah aplikasi. Terlebih lagi dengan Codeigniter pengembangan sebuah aplikasi menjadi mudah. Termasuk adalah membuat sistem login dimana merupakan elemen penting sebuah aplikasi. Tutorial ini menjelaskan bagaimana membuat sistem login menggunakan Codeigniter serta membuat multilevel login dengan Codeigniter. Semoga Bermanfaat.

1. Mengenal Library Session

Library yang paling penting dalam proses login adalah SESSION. Sebenarnya PHP sudah menyediakan sebuah function untuk melakukan proses SESSION, tetapi Framework telah “membungkusnya” menjadi lebih mudah digunakan serta banyak fitur yang bisa kita gunakan.

Pada Framework Codeigniter juga terdapat sebuah library yang dapat mengelola SESSION dengan baik yaitu Session Class. Dengan Session Class ini anda akan dimudahkan dalam mengelola SESSION dari user yang mengakses aplikasi anda. Dalam user guide / dokumentasi Codeigniter sudah sangat jelas di jelaskan mengenai Session Class. Untuk menampilkan atau memanggil Session Class anda cukup dengan

```
$this->load->library('session');
```

Ya, diatas adalah cara memanggil Session Class yang akan kita butuhkan nantinya. Kemudian untuk melakukan set sebuah session kita cukup saja dengan :

```
$this->session->set('nama_session');
```

Session yang sudah kita set akan melakukan “maintenance” terhadap user yang sudah menggunakan session tersebut. Dan session tersebut sudah tentu punya masa berlaku nah di Codeigniter pun sudah memiliki fitur untuk melakukan pengaturan umur dari session tersebut. Bagaimana caranya ??? anda cukup membuka pada file config.php. Kalo anda tidak tahu saya kasih tau monggoh anda masuk ke direktori **application/config/config.php**. Kemudian cari pada baris 247 anda akan menemui pengaturan session di situ. Namun, saya

tidak akan menjelaskan secara detail, anda dapat mengakses dokumentasi untuk mendapatkan lebih lengkap mengenai pengaturan session.

Ya pada baris 248 ada pengaturan masa berlaku session, dengan ukuran second secara default Session dapat basi di Codeigniter selama 2 jam atau 7200 detik. Anda dapat melakukan setting sesuka hati asal semua dikonversi ke bentuk detik. Dan jika anda memberikan angka 0 (nol / zero) maka session tidak akan pernah basi alias expired.

Untuk pengecekan session kita dapat menggunakan :

```
if(!$this->session->userdata('username'))
{
    redirect('admin/login');
}
else
{
    redirect('admin/home');
}
```

Oke cuplikan kode diatas adalah bagaimana kita akan melakukan pengecekan, apakah session sudah terpakai atau belum. Jika tidak ada alias tidak diaktifkan oleh user atau digunakan oleh pengguna aplikasi (ditandai dengan ! tanda seru) maka akan langsung diarahkan pada halaman login. Dan begitu sebaliknya akan langsung diarahkan ke halaman home.

Kurang lebih seperti itulah nantinya cara kerja dari library Session milik Codeigniter. Kita akan mulai dengan membuat sistem login dan kedua adalah membuat sistem login multi level alias setiap level dapat login disesuaikan dengan level si pengguna alias user.

2. Membuat Sistem Login

Berikut ini akan dibahas mengenai bagaimana membuat sistem login menggunakan Codeigniter. Oke lah kita langsung saja menuju TKP...

2.1. Membuat Tabel

Kita awali dengan membuat tabel terlebih dahulu, tabel ini berfungsi untuk menampung data pengguna yang nantinya akan kita gunakan pada latihan kita membuat sistem login.

Langsung saja anda buat databasenya, bikin pake phpMyAdmin, SQLyog ataupun aplikasi apapun lah ya. Buat database misalkan dengan nama **db_login**, berikut query-nya :

```
CREATE DATABASE db_login;
```

Kalo sudah anda buat databasenya, buatlah tabel sebagai berikut :

```
CREATE TABLE `user` (
  `username` VARCHAR( 50 ) NOT NULL ,
  `password` VARCHAR( 32 ) NOT NULL ,
  `nama` VARCHAR( 100 ) NULL ,
  `level` INT( 1 ) NOT NULL ,
  `status` INT( 1 ) NOT NULL ,
  PRIMARY KEY ( `username` )
) ENGINE = MYISAM ;
```

Dan masukkan / insert data yang akan kita gunakan dengan query berikut :

```
INSERT INTO user VALUES ('fergie', MD5( 'manutd' ) ,  
Sir Alex Fergie', '1', '1');  
  
INSERT INTO user VALUES ('moyes', MD5( 'david' ) ,  
David Moyes', '2', '1');
```

2.2. Membuat Model

Selanjutnya buat file baru, kasih nama misalkan **m_login.php**, berikut adalah potongan kode dari script **m_login.php** :

```
<?php if(!defined('BASEPATH')) exit('Hacking Attempt :  
Keluar dari sistem...!');  
  
class M_login extends CI_Model  
{  
    public function __construct()  
    {  
        parent::__construct();  
    }  
  
    public function ambilPengguna($username, $password,  
$status)  
    {  
        $this->db->select('*');  
        $this->db->from('user');  
        $this->db->where('username', $username);  
        $this->db->where('password', $password);  
        $this->db->where('status', $status);  
        $query = $this->db->get();  
  
        return $query->num_rows();  
    }  
  
    public function dataPengguna($username)  
    {  
        $this->db->select('username');  
        $this->db->select('name');  
        $this->db->where('username', $username);  
        $query = $this->db->get('user');  
  
        return $query->row();  
    }  
}
```

2.3. Membuat View untuk Login

Jika sudah membuat model maka selanjutnya adalah kita akan membuat file view, yang fungsinya adalah untuk menampilkan form login. Buat file baru lagi, anda kasih nama saja misalkan **form_login.php**. Seperti ini scriptnya :

```
<html>
<head>
  <title>Form Login</title>
  <style>
    body
    {
      font-family:Calibri;
      margin:50px;
    }

    #form-login{
      margin:auto;
      width:500px;
      padding:10px;
      border:1px #ccc solid;
      font-size:18px;
      font-weight:bold;
      color:#FF6600;
    }

    .inputan
    {
      padding:3px;
      font-family:Calibri;
      border:1px solid #ccc;
    }

    .tombol
    {
      padding:5px;
      background:#FF6600;
      color:#FFF;
      font-weight:bold;
      font-family:Calibri;
      font-size:15px;
      border:#eee 1px solid;
    }

    .error
    {
      color:#FF6600;
      font-size:11px;
    }

  </style>
</head>
```

```
<body>
  <form action="<?php echo
base_url();?>index.php/login/login_form" method="post"
name="login">
    <div id="form-login">
        Administrator Page - Plase Login First
        <br><br>
        <table border="0" cellpadding="4">
            <tr>
                <td>Username</td>
                <td>:</td>
                <td><input type="text" size="40" name="username"
value="<?php echo set_value('username');?>"
class="inputan"> <?php echo form_error('username');?></td>
            </tr>
            <tr>
                <td>Password</td>
                <td>:</td>
                <td><input type="password" size="40"
name="password" value="<?php echo
set_value('password');?>" class="inputan"> <?php echo
form_error('password');?></td>
            </tr>
            <tr>
                <td>&nbsp;</td>
                <td>&nbsp;</td>
                <td><input type="submit" name="login"
value="Login" class="tombol"> </td>
            </tr>
        </table>
    </div>
  </form>
</body>
</html>
```

2.4. Membuat Controller

Gimana panjang banget kan, ga usah khawatir. Mari lanjutkan dengan membuat file baru untuk controller, misalkan beri nama dengan login.php, berikut scriptnya :

```
<?php if(!defined('BASEPATH')) exit('Hacking Attempt :
Keluar dari sistem...!!');

class Login extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();

        $this->load->model('m_login');

        $this->load-
>library(array('form_validation','session'));

        $this->load->database(); //berlanjut ke bawah ini ...
```

```
//lanjutan yang diatas

    $this->load->helper('url');

}

public function index()
{
    $session = $this->session->userdata('isLogin');

    if($session == FALSE)
    {
        redirect('login/login_form');
    }else
    {
        redirect('home');
    }
}

public function login_form()
{
    $this->form_validation->set_rules('username',
'Username', 'required|trim|xss_clean');
    $this->form_validation->set_rules('password',
'Password', 'required|md5|xss_clean');
    $this->form_validation->set_error_delimiters('<span
class="error">', '</span>');

    if($this->form_validation->run()==FALSE)
    {
        $this->load->view('form_login');
    }else
    {
        $username = $this->input->post('username');
        $password = $this->input->post('password');

        $cek      =      $this->m_login->ambilPengguna($username,
$password, 1)

        if($cek <> 0)
        {
            $this->session->set_userdata('isLogin', TRUE);
            $this->session-
>set_userdata('username',$username);

            redirect('home');
        }else
        {
            //masih berlanjut ke bawah lagi gan ...

```

```
?>
                                <script>
                                alert('Gagal Login: Cek username
dan password anda!');
                                history.go(-1);
                                </script>
                                <?php
                                }
                                }
                                }

                                public function logout()
                                {
                                    $this->session->sess_destroy();

                                    redirect('login/login_form');
                                }
                                }

?>
```

2.5. Menjalankan sistem Login

Bagaimana masih semangat? ☺Tenang sudah saya siapkan dalam lampiran bersama ebook ini. Oke sekarang silahkan coba saja di browser, jika benar anda akan dibawa pada halaman seperti tampak pada gambar berikut :

Administrator Page - Plase Login First

Username :
Password :

Dan buat file baru lagi dengan nama home.php, apa fungsinya? Coba cek kembali pada login.php ada sebaris kode untuk mengarahkan langsung pada controller dengan nama home. Nah, tulis script home sebagai berikut :

```
<?php if(!defined('BASEPATH')) exit('Hacking Attempt. Keluar dari sistem.');
```

```
class Home extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();

        $this->load->library(array('session'));
    }

    public function index()
    {
        if($this->session->userdata('isLogin') == TRUE)
        {
            redirect('login/login_form');
        }else
        {
            $this->load->view('home_v');
        }
    }
}
```

Dan buat lagi file home_v.php,

```
<html>
<head>
    <title>Selamat Datang Friend</title>
<style>
body
    {
        font-family:Calibri;
    }
</style>
</head>
<body>
    <h2>Hai, Selamat Datang di Halaman Administrator</h2>
</body>
</html>
```

Coba pada browser, jika apa yang anda buat benar. Maka anda akan mendapatkan tampilan berikut :



Hai, Selamat Datang di Halaman Administrator

3. Membuat Multilevel Login

Setelah kita membuat bagaimana membuat sistem login dengan Codeigniter, kali ini kita akan beranjak pada sistem yang lebih kompleks dari sistem login, yaitu Multi-level Login. Oke kita mulai dengan persiapan terlebih dahulu.

3.1. Persiapan Awal

Ya, persiapan disini adalah kita akan merancang terlebih dahulu apa yang akan kita buat nantinya pada sistem login. Desain yang akan kita buat adalah kita mempunyai misalkan 2 level login, yaitu Admin dan Operator. Dimana jika salah satu login maka akan diarahkan ke halaman tertentu. Bagaimana sudah paham kan desain nya?

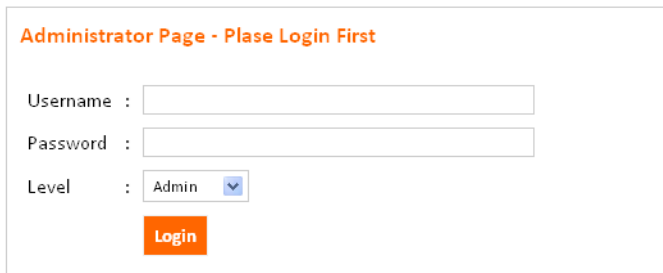
3.2. Membuat Login

Halaman login sudah kita buat pada pembahasan sebelumnya, kita hanya akan menambahkan sebuah combobox yang bertujuan untuk level user. Modifikasi lah file login dengan script berikut :

```
<form action="php echo<br/base_url();?>index.php/login/login_form" method="post"  
name="login">  
    <div id="form-login">  
        Administrator Page - Plase Login First  
        <br><br>  
        <table border="0" cellpadding="4">  
            <tr>  
                <td>Username</td>  
                <td>:</td>  
                <td><input type="text" size="40" name="username"  
value="php echo set_value('username');?&gt;"<br/class="inputan"> <?php echo form_error('username');?></td>  
            </tr>  
            <tr>  
                <td>Password</td>  
                <td>:</td>  
                <td><input type="password" size="40"  
name="password" value="php echo<br/set_value('password');?>" class="inputan"> <?php echo  
form_error('password');?></td>  
            </tr>  
            <tr>  
                <td>Level</td>  
                <td>:</td>  
                <td><select name="level" class="inputan">  
                    <option value="1">Admin</option>  
                    <option value="2">Operator</option>  
                </select></td>  
            </tr>
```

```
<tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td><input type="submit" name="login" value="Login"
class="tombol"> </td>
</tr>
</table>
</div>
</form>
```

Sehingga akan muncul tampilan login sebagai berikut :



Administrator Page - Plase Login First

Username :

Password :

Level : ▼

3.3. Modifikasi pada file Model m_login.php

Modifikasi Model diperlukan karena kita akan mencocokkan username dan password sesuai dengan levelnya ☺. Cukup ubah pada baris berikut :

```
public function ambilPegguna($username, $password,
$status, $level)
{
    $this->db->select('*');
    $this->db->from('user');
    $this->db->where('username', $username);
    $this->db->where('password', $password);
    $this->db->where('status', $status);
    $this->db->where('level', $level);
    $query = $this->db->get();

    return $query->num_rows();
}
```

3.4. Modifikasi file Controller login.php

Selanjutnya lakukan modifikasi pada controller login. Disini kita akan menambahkan session untuk level. Dan nanti level ini yang menjadi acuan untuk membedakan antar user.

```
public function login_form()
{
    $this->form_validation->set_rules('username',
'Username', 'required|trim|xss_clean');
    $this->form_validation->set_rules('password',
'Password', 'required|md5|xss_clean');
    $this->form_validation->set_error_delimiters('<span
class="error">', '</span>');

    if($this->form_validation->run()==FALSE)
    {
        $this->load->view('form_login');
    }else
    {
        $username = $this->input->post('username');
        $password = $this->input->post('password');
        $level = $this->input->post('level');

        $cek      =      $this->m_login->ambilPengguna($username,
$password, 1, $level);

        if($cek <> 0)
        {
            $this->session->set_userdata('isLoggedIn', TRUE);
            $this->session-
>set_userdata('username',$username);
            $this->session->set_userdata('level',$level);

            redirect('home');
        }else
        {
            echo " <script>
                                alert('Gagal Login: Cek username ,
password dan level anda!');
                                history.go(-1);
                                </script>";
        }
    }
}
```

Oke sudah ketemu perubahannya ? ya bagi yang belum ketemu, coba liat pada baris ini **\$level = \$this->input->post('level');** dan **\$this->session->set_userdata(\$level);** ya itulah penambahan script baru sebagai “senjata” untuk membuat multilevel login. 😊

3.5. Mengubah halaman Home (Controller dan View)

Ubahlah file controller home.php menjadi :

```

public function __construct()
{
    parent::__construct();

    $this->load->library('session');

    $this->load->helper('url');

    $this->load->model('m_login');

    $this->load->database();

}

public function index()
{
    if($this->session->userdata('isLoggedIn') == FALSE)
    {
        redirect('login/login_form');
    }else
    {
        $this->load->model('m_login');

        $user = $this->session->userdata('username');

        $data['level'] = $this->session->userdata('level');
        $data['pengguna'] = $this->m_login->dataPengguna($user);

        $this->load->view('home_v', $data);
    }
}

```

Kemudian lakukan modifikasi pada file `home_v.php`. Disini kita akan melakukan perubahan tampilan sesuai dengan level user nya.

```
<html>
  <head>
    <title>Selamat Datang Friend</title>
  <style>
    body
    {
      font-family:Calibri;
    }
  </style>
</head>
```

```
<body>
  <?php
    if($level == "1")
    {
      ?>
      <h2>Hai <?php echo $pengguna->nama;?>, anda login sebagai
      Administrator <?php echo anchor('login/logout', 'Keluar
      &raquo;');?></h2>
      <?php
    }else
    {
      ?>
      <h2>Hai <?php echo $pengguna->nama;?>, anda login sebagai
      Operator <?php echo anchor('login/logout', 'Keluar
      &raquo;');?></h2>
      <?php
    }
    ?>
  </body>
</html>
```

Dan lakukan ujicoba pada browser anda, berikut adalah tampilan dari halaman home :



Hai Dimas Edubuntu Samid, anda login sebagai Administrator [Keluar »](#)

Gambar 1. Halaman Home level Admin (1)



Hai Sir Alex Fergie, anda login sebagai Operator [Keluar »](#)

Gambar 2. Halaman Home level Operator(2)

Oke intinya adalah kita dapat membedakan level user melalui setting session yang sudah kita buat tadi. Jadi, anda dapat melakukan modifikasi melalui level yang sudah kita set melalui Controller. Dan menjadikan multi-level yang anda buat jauh lebih memenuhi kebutuhan aplikasi anda.

Demikian tutorial mengenai sistem login dengan Codeigniter, dan mohon maaf bila ada kesalahan bagi anda yang ingin rikues tutorial ataupun bertanya mengenai semua tutorial saya, silahkan kirim email ke **edudimas1@gmail.com**. Semoga bermanfaat.
^_^.

Tentang Penulis

Dimas Edu Prasada (Dimas Edubuntu Samid) / Edu, lahir di Kota Tegal, 27 Januari 1988. Bujangan yang hobi dengan music koplo ini , masih aktif sebagai praktisi IT. Menggemari “mazhab” web based application dan PHP menjadi bahasa yang paling disukainya.

Setelah sekian lama berkecimpung di dunia IT saat ini saya ingin sekali mewujudkan cita – cita yaitu memiliki sekolah gratis untuk anak-anak yang tidak mampu. Namun, saat ini masih belum bisa mewujudkannya semoga dikemudian hari dapat terwujud keinginan tersebut. Mohon doanya ya.... ☺

Saat ini penulis masih aktif mengajar di STMIK Tasimalaya serta pada LPK Mita College sebagai pengajar bahasa pemrograman. Semoga apa yang saya tulis dapat bermanfaat untuk anda semua, seperti quote yang menjadi prinsip hidup penulis, “Sebaik - baiknya orang adalah mereka yang bermanfaat untuk orang lain.”

Penulis dapat dihubungi melalui :

HP : 0857-42100-454

Email : edudimas1@gmail.com

FB : <http://facebook.com/eduaying>

Twitter : @edu_aying

Blog / URL : samidcorner.blogspot.com