



# MODUL LOGIN MULTILEVEL PADA CODEIGNITER

KELAS XII

TIARA KUSUMA DEWI S.Tr.

# DAFTAR ISI

PENGENALAN .....	2
STEP 1 - PERSIAPAN .....	2
STEP 2 - DATABASE .....	3
STEP 3 - KONFIGURASI .....	3
STEP 4 - BUAT MODEL .....	5
STEP 5 - BUAT CORE SENDIRI .....	5
STEP 6 - BUAT CONTROLLER AUTH .....	6
STEP 7 - BUAT FORM LOGIN .....	8
STEP 8 - BUAT CONTROLLER PAGE .....	9
STEP 9 - BUAT TEMPLATE LOGIN .....	10
STEP 10 - BUAT TEMPLATE BACKEND .....	11
STEP 11 - HOME .....	13
STEP 12 - BERITA .....	14
LATIHAN STEP 12 – PENILAIAN UH 1 .....	16
STEP 13 - PENGGUNA .....	17
LATIHAN STEP 13 – PENILAIAN UH 2 .....	18
STEP 14 - KONTAK .....	18
LATIHAN AKHIR .....	19
PENGETESAN .....	19

# LOGIN MULTILEVEL PADA CODEIGNITER

## PENGENALAN

Login multilevel adalah sebuah fitur pada web yang berguna untuk membuat beberapa hak akses pada sebuah aplikasi web. Hal ini berguna sebagai pengatur halaman apa saja yang akan digunakan sesuai dengan hak akses yang telah dibuat. Dalam praktik kali ini, akan dibagi level user menjadi 2 (Admin dan Operator), berikut detailnya:

### Admin

Level user admin hak aksesnya adalah berikut ini :

1. Akses Menu Home. Aksi yang bisa di lakukan : Read
2. Akses Menu Berita. Aksi yang bisa di lakukan : Create, Read, Update, Delete
3. Akses Menu Pengguna. Aksi yang bisa di lakukan : Create, Read, Update, Delete
4. Akses Menu Kontak. Aksi yang bisa di lakukan : Read

### Operator

Level user operator hak aksesnya adalah berikut ini :

1. Akses Menu Home. Aksi yang bisa di lakukan : Read
2. Akses Menu Berita. Aksi yang bisa di lakukan : Read, Update, Delete
3. Akses Menu Pengguna. Aksi yang bisa di lakukan : Read
4. Akses Menu Kontak. Aksi yang bisa di lakukan : Read

### Akun User

Disini telah disediakan 2 data user untuk tutorial, diantaranya :

1. User Admin

Username : rizaldi

Password : 12345678

2. User Operator

Username : siska

Password : siska123

## STEP 1 - PERSIAPAN

Berikut ini adalah hal-hal yang harus dipersiapkan :

1. Download Framework Codeigniter,
2. Download Bootstrap.
3. Download jQuery,
4. Buat sebuah folder baru dengan nama login\_role\_ci, lalu simpan pada folder xampp/htdocs/.
5. Buat sebuah folder dengan nama css, lalu simpan pada folder xampp/htdocs/login\_role\_ci/.
6. Buat sebuah folder dengan nama js, lalu simpan pada folder xampp/htdocs/login\_role\_ci/.


7. Ekstrak codeigniter yang telah di download tadi.
8. Copy semua file dan folder yang ada di folder codeigniter\_v3 yang telah di ekstrak tadi ke folder xampp/htdocs/login\_role\_ci/.
9. Ekstrak file bootstrap.7z yang telah didownload tadi.
10. Copy and paste folder fonts dari folder bootstrap yang telah diekstrak tadi ke folder xampp/htdocs/login\_role\_ci/.
11. Copy and paste file bootstrap.min.css dari folder bootstrap/css/ yang telah diekstrak tadi ke folder xampp/htdocs/login\_role\_ci/css/.
12. Copy and paste file bootstrap.min.js dari folder bootstrap/js/ yang telah diekstrak tadi ke folder xampp/htdocs/login\_role\_ci/js/.
13. Copy and paste file jquery.min.js yang telah di download tadi ke folder xampp/htdocs/login\_role\_ci/js/.

## STEP 2 - DATABASE

Pada step ini, kita akan membuat databasenya. ikuti langkah-langkah berikut ini :

1. Buat sebuah database dengan nama **multilevel**.
2. Buat sebuah tabel dengan nama **user**. Struktur tabelnya sebagai berikut :

Struktur Tabel - Cara Membuat Login Multi Level User dengan Codeigniter dan MySQL

#	Name	Type
1	id 	int(11)
2	username	varchar(20)
3	password	varchar(50)
4	nama	varchar(200)
5	role	enum('admin', 'operator')

3. Insert 2 data user :

```
INSERT INTO `user` (`id`, `username`, `password`, `nama`, `role`)
VALUES (1, 'rizaldi', md5('12345678'), 'Rizaldi Maulidia Achmad', 'admin'), (2, 'siska', md5('siska123'), 'Siska Melina Rachman', 'operator');
```

## STEP 3 - KONFIGURASI

Karena dibuat dengan Codeigniter, pertama kita harus melakukan konfigurasi terlebih dahulu pada framework codeigniternya.

1. Buka folder xampp/htdocs/login\_role\_ci/application/config/
2. Buka file **config.php**

Cari kode berikut \$config['base\_url'] = ""; Ubah kode tersebut jadi seperti ini :

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

### 3. Buka file **autoload.php**

Cari kode berikut ini :

```
$autoload['libraries'] = array();  
$autoload['helper'] = array();
```

Ubah jadi seperti ini :

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

Kode diatas digunakan untuk memuat (menload) class database dan url. Lalu simpan file tersebut.

### 4. Buka file **routes.php**

Cari kode berikut ini :

```
$route['default_controller'] = 'welcome';
```

Ubah jadi seperti ini :

```
$route['default_controller'] = 'auth';
```

Kode diatas digunakan untuk menset controller mana yang akan di load pertama kali. Secara default, Codeigniter telah menset default controller yaitu welcome. Disini kita set default controller menjadi auth. Lalu simpan file tersebut.

### 5. Buka file **database.php**

Cari kode berikut ini :

```
'hostname' => 'localhost',  
'username' => '',  
'password' => '',  
'database' => '',
```

Ubah jadi seperti ini :

```
'hostname' => 'localhost',  
'username' => 'root', // Username  
'password' => '', // Jika menggunakan password isi  
'database' => 'multilevel', // Nama databasenya
```

Kode diatas digunakan untuk koneksi ke database. Lalu simpan file tersebut.

## STEP 4 - BUAT MODEL

Langkah selanjutnya adalah membuat modelnya. Buat sebuah file dengan nama **UserModel.php**, Lalu simpan pada folder `xampp/htdocs/login_role_ci/application/models/`. Berikut kodenya .

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class UserModel extends CI_Model {
    public function get($username){
        $this->db->where('username', $username); // Untuk menambahkan
Where Clause : username='$username'
        $result = $this->db->get('user')->row(); // Untuk mengeksekusi
dan mengambil data hasil query
        return $result;
    }
}
```

Fungsi [get\(\)](#) berguna untuk melakukan query ke tabel user berdasarkan username yang diinput pada form login. Fungs ini akan menghasilkan query sebagai berikut : `SELECT * FROM user WHERE username = '$username'`.

## STEP 5 - BUAT CORE SENDIRI

Selanjutnya kita akan membuat sebuah core. Core ini berfungsi untuk mengkontrol akses ke halaman setelah login dan juga untuk mengatur template mana yang akan di pakai (Backend / Frontend). Buat sebuah file dengan nama **MY\_Controller.php**, lalu simpan pada folder `xampp/htdocs/login_role_ci/application/core/`. Berikut ini kodenya

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class MY_Controller extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->authenticated(); // Panggil fungsi authenticated
    }

    public function authenticated(){ // Fungsi ini berguna untuk
mengecek apakah user sudah login atau belum
        // Pertama kita cek dulu apakah controller saat ini yang sedang
diakses user adalah controller Auth apa bukan
        // Karena fungsi cek login hanya kita berlakukan untuk
controller selain controller Auth
        if($this->uri->segment(1) != 'auth' && $this->uri->segment(1)
!= ''){
            // Cek apakah terdapat session dengan nama authenticated
            if( ! $this->session->userdata('authenticated')) // Jika
tidak ada / artinya belum login
                redirect('auth'); // Redirect ke halaman login
        }
    }

    public function render_login($content, $data = NULL){
        /*
        * $data['contentnya']
        * variabel diatas ^ nantinya akan dikirim ke file
views/template/login/index.php
        */
        $data['contentnya'] = $this->load->view($content, $data, TRUE);
    }
}
```

```

        $this->load->view('template/login/index', $data);
    }
    public function render_backend($content, $data = NULL){
        /*
        * $data['headernya'] , $data['contentnya']
        * variabel diatas ^ nantinya akan dikirim ke file
        views/template/backend/index.php
        * */
        $data['headernya'] = $this->load-
>view('template/backend/header', $data, TRUE);
        $data['contentnya'] = $this->load->view($content, $data, TRUE);
        $this->load->view('template/backend/index', $data);
    }
}

```

Core akan kita manfaatkan untuk mengontrol akses ke halaman setelah login dan juga untuk mengatur template. Core ini nanti harus di gunakan pada semua controller.

*public function \_\_construct()* Kode diatas berguna sebagai fungsi awal yang akan dieksekusi setiap kali Core MY\_Controller dipanggil. Jadi sebelum mengeksekusi fungsi lainnya, kode yang ada didalam fungsi *\_\_construct()* akan terlebih dahulu dieksekusi diawal.

*if( ! \$this->session->userdata('authenticated'))* Fungsi dari kode diatas adalah untuk mengecek apakah session dengan nama authenticated ada atau tidak. Pada if kondisi diatas saya menambahkan tanda seru "!". Tanda seru tersebut artinya kebalikannya, jadi If disitu jika di terjemahkan adalah jika session authenticated tidak ada, maka redirect ke halaman login.

Disini juga kita buat 2 fungsi yakni *render\_login()* dan *render\_backend()*. Fungsi *render\_login()* sendiri untuk memanggil template login sehingga ketika user mengakses form login, maka template yang digunakan adalah template untuk form login. Begitupun dengan *render\_backend()*, fungsi ini untuk memanggil template backend (setelah login).

## STEP 6 - BUAT CONTROLLER AUTH

Pada tahap ini, kita akan membuat controller untuk proses login dan logout. Buat sebuah file dengan nama **Auth.php**, lalu simpan pada folder `xampp/htdocs/login_role_ci/application/controllers/`. Berikut ini kodenya

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Auth extends MY_Controller {
    public function __construct(){
        parent::__construct();
        $this->load->model('UserModel');
    }
    public function index(){
        if($this->session->userdata('authenticated')) // Jika user sudah
login (Session authenticated ditemukan)
        redirect('page/home'); // Redirect ke page home
        // function render_login tersebut dari file core/MY_Controller.php
        $this->render_login('login'); // Load view login.php
    }
    public function login(){

```

```

        $username = $this->input->post('username'); // Ambil isi dari
        inputan username pada form login
        $password = md5($this->input->post('password')); // Ambil isi dari
        inputan password pada form login dan encrypt dengan md5
        $user = $this->UserModel->get($username); // Panggil fungsi get
        yang ada di UserModel.php
        if(empty($user)){ // Jika hasilnya kosong / user tidak ditemukan
            $this->session->set_flashdata('message', 'Username tidak
            ditemukan'); // Buat session flashdata
            redirect('auth'); // Redirect ke halaman login
        }else{
            if($password == $user->password){ // Jika password yang diinput
            sama dengan password yang didatabase
                $session = array(
                    'authenticated'=>true, // Buat session authenticated dengan
                    value true
                    'username'=>$user->username, // Buat session username
                    'nama'=>$user->nama, // Buat session nama
                    'role'=>$user->role // Buat session role
                );
                $this->session->set_userdata($session); // Buat session sesuai
                $session
                redirect('page/home'); // Redirect ke halaman home
            }else{
                $this->session->set_flashdata('message', 'Password salah'); //
                Buat session flashdata
                redirect('auth'); // Redirect ke halaman login
            }
        }
    }
    public function logout(){
        $this->session->sess_destroy(); // Hapus semua session
        redirect('auth'); // Redirect ke halaman login
    }
}

```

Satu hal yang harus diperhatikan pada skrip diatas, biasanya ketika kita buat controller di codeigniter, controller tersebut akan meng-extends ke CI\_Controller. Tapi dalam tutorial ini kita ubah menjadi extends MY\_Controller agar skrip yang ada di MY\_Controller bisa di eksekusi oleh controller tersebut.

Dari kode diatas, anda bisa lihat bahwa controller Auth meng-extends ke MY\_Controller (class Auth extends MY\_Controller O. Secara default ketika kita menggunakan Framework Codeigniter, maka setiap controller harus meng-extends ke CI\_Controller. Tapi karena pada kasus ini kita telah membuat Core sendiri dengan nama MY\_Controller, maka ketika kita akan menggunakan core tersebut, controllernya harus meng-extends ke MY\_Controller.

Jadi setiap kita membuka salah satu fungsi di dalam controller Auth, maka dia akan mengeksekusi terlebih dahulu kode yang ada di dalam fungsi \_construct() pada core MY\_Controller, baru setelah itu dia akan mengeksekusi kode yang ada didalam fungsi yang kita buka / panggil. Kesimpulannya, dengan cara ini, Kode kita akan lebih efsiensi karena kita tidak perlu lagi menambahkan fungsi if untuk mengecek apakah user sudah login atau belum pada setiap fungsi. Karena sudah kita letakan kode tersebut pada core MY\_Controller.



`if($this->session->userdata('authenticated'))` Kode diatas berfungsi untuk mengecek apakah session dengan nama authenticated ada atau tidak. Jika ada, artinya user sudah login. Jika tidak ada, berarti user belum login.

`$user = $this->UserModel->get($username)` Kode diatas berfungsi untuk memanggil fungsi get() yang ada di UserModel.php untuk mendapatkan data user yang ada didatabase berdasarkan username.

`$this->session->set_flashdata('message', 'Username tidak ditemukan')` Kode diatas untuk membuat session flashdata. Session ini sifatnya hanya sementara, jadi ketika sudah dibaca, maka akan otomatis terhapus. Struktur dasar kodenya adalah `$this->session->set_flashdata('nama_session', 'isi_session')`.

`$this->session->set_userdata($session)` Kode diatas berfungsi untuk membuat session. Sebenarnya di codeigniter ada 2 opsi cara membuat array antara lain :

1. Jika anda ingin membuat session lebih dari 1, sebaiknya pakai cara ini : (dalam tutorial ini, kita menggunakan cara ini)

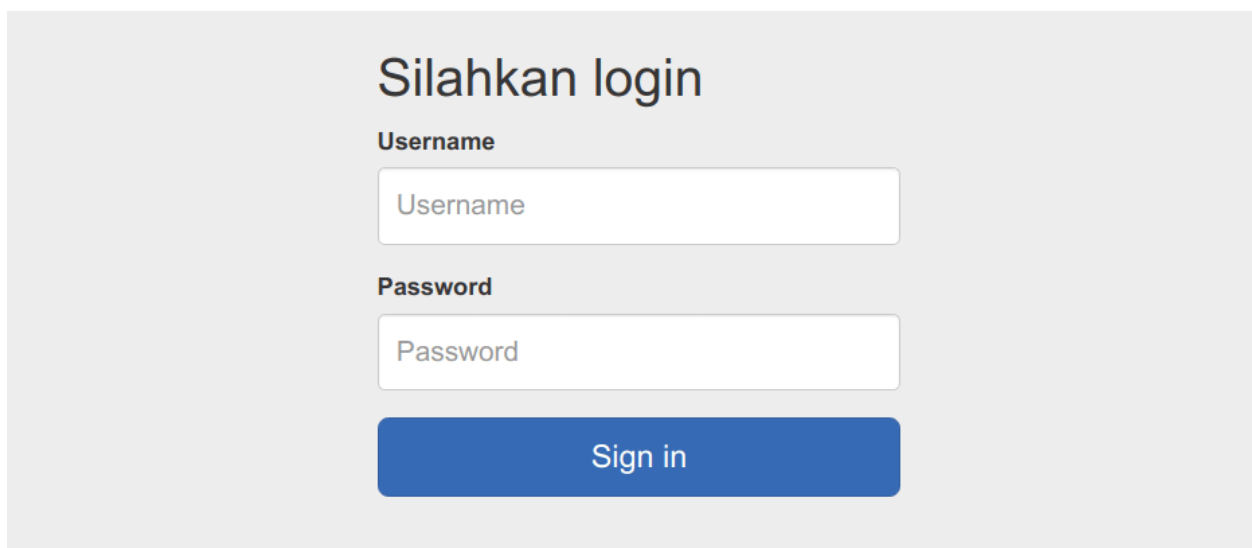
```
$session = array(
    'nama_session1'=>'isi_session1',
    'nama_session2'=>'isi_session2',
    'nama_session3'=>'isi_session3' );
$this->session->set_userdata($session);
```

2. Jika anda ingin membuat hanya 1 session, pakai cara ini :

```
$this->session->set_userdata('nama_session1', 'isi_session1');
```

## STEP 7 - BUAT FORM LOGIN

Pada tahap ini, kita akan membuat form loginnya. Buat sebuah file dengan nama **login.php**, lalu simpan pada folder `xampp/htdocs/login_role_ci/application/views/`. Berikut ini tampilan dan kodenya :



The image shows a login form with a light gray background. At the top, the text "Silahkan login" is displayed in a large, bold, black font. Below this, there are two input fields. The first is labeled "Username" in a bold black font, and the input field itself has the placeholder text "Username" in a light gray font. The second is labeled "Password" in a bold black font, and the input field has the placeholder text "Password" in a light gray font. Below these two fields is a blue button with the text "Sign in" in white.

```

<?php
// Cek apakah terdapat session nama message
if($this->session->flashdata('message')){ // Jika ada
    echo '<div class="alert alert-danger">'. $this->session->flashdata('message'). '</div>'; // Tampilkan pesannya
}
?>
<form method="post" action="<?php echo
base_url('index.php/auth/login'); ?>">
    <div class="form-group">
        <label>Username</label>
        <input type="text" class="form-control" name="username"
placeholder="Username" required autofocus>
    </div>
    <div class="form-group">
        <label>Password</label>
        <input type="password" class="form-control" name="password"
placeholder="Password" required>
    </div>
    <button class="btn btn-lg btn-primary btn-block" type="submit">Sign
in</button>
</form>

```

*if(\$this->session->flashdata('message'))* Kode diatas berfungsi untuk memanggil session flashdata dengan nama message yang dibuat pada controller auth fungsi login (jika data user tidak ditemukan). Struktur dasar kodenya adalah \$this->session->flashdata('nama\_session').

## STEP 8 - BUAT CONTROLLER PAGE

Selanjutnya, kita akan membuat controller page. Buat sebuah file dengan nama **Page.php**, lalu simpan pada folder xampp/htdocs/login\_role\_ci/application/controllers/. Berikut ini kodenya :

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Page extends MY_Controller {
    public function home(){
        // function render_backend tersebut dari file
        core/MY_Controller.php
        $this->render_backend('home'); // load view home.php
    }
    public function berita(){
        // function render_backend tersebut dari file
        core/MY_Controller.php
        $this->render_backend('berita_view'); // load view berita.php
    }
    public function pengguna(){
        //if($this->session->userdata('role') != 'admin') // Jika user yg
        login bukan admin
        //show_404(); // Redirect ke halaman 404 Not found
        // function render_backend tersebut dari file
        core/MY_Controller.php
        $this->render_backend('pengguna_view'); // load view pengguna.php
    }
    public function kontak(){

```

```

        // function render_backend tersebut dari file
        core/MY_Controller.php
        $this->render_backend('kontak'); // load view kontak.php
    }
}

```

Sama seperti controller Auth, Controller Page juga meng-extends ke MY\_Controller.

## STEP 9 - BUAT TEMPLATE LOGIN

Kemudian kita akan membuat template untuk halaman login. Buat sebuah folder dengan nama **template**, lalu simpan pada folder xampp/htdocs/login\_role\_ci/application/views/. Kemudian buat sebuah folder lagi dengan nama **login** pada folder xampp/htdocs/login\_role\_ci/application/views /template/.

Terakhir buat sebuah file dengan nama **index.php**, lalu simpan pada folder xampp/htdocs/login\_role\_ci/application/views/template/login/. Berikut ini kodenya :

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other
head content must come *after* these tags -->
    <meta name="description" content="">
    <meta name="author" content="">
    <title>Login + Hak Akses (PHP)</title>
    <!-- Load File CSS Bootstrap -->
    <link href="<?php echo base_url('css/bootstrap.min.css'); ?>"
rel="stylesheet">
    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements
and media queries -->
    <!--[if lt IE 9]>
    <script
src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
    <script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
    <style>
    body {
        padding-top: 40px;
        padding-bottom: 40px;
        background-color: #eee;
    }
    .form-signin {
        max-width: 330px;
        padding: 15px;
        margin: 0 auto;
    }
    .form-signin .form-signin-heading{
        margin-bottom: 10px;
    }
    .form-signin .form-control {
        position: relative;
        height: auto;
        -webkit-box-sizing: border-box;

```

```

        -moz-box-sizing: border-box;
        box-sizing: border-box;
        padding: 10px;
        font-size: 16px;
    }
    .form-signin .form-control:focus {
        z-index: 2;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="form-signin">
            <h2 class="form-signin-heading">Silahkan login</h2>
            <?php
            /*
            * Variabel $contentnya diambil dari core MY_Controller
            * (application/core/MY_Controller.php)
            * */
            echo $contentnya;
            ?>
        </div>
    </div>
</body>
</html>

```

## STEP 10 - BUAT TEMPLATE BACKEND

Kemudian kita akan membuat template untuk halaman setelah login. Buat sebuah folder dengan nama **backend** pada folder `xampp/htdocs/login_role_ci/application/views/template/`. Kemudian buat sebuah file dengan nama **index.php**, lalu simpan pada folder `xampp/htdocs/login_role_ci/application/views/template/backend/`. Berikut ini kodenya :

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other
head content must come *after* these tags -->
    <meta name="description" content="">
    <meta name="author" content="">
    <title>Login + Hak Akses (PHP)</title>
    <!-- Load File CSS Bootstrap -->
    <link href="<?php echo base_url('css/bootstrap.min.css'); ?>"
rel="stylesheet">
    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements
and media queries -->
    <!--[if lt IE 9]>
    <script
src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>

```

```

<script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
<![endif]-->
<style>
body {
    min-height: 2000px;
    padding-top: 70px;
}
</style>
</head>
<body>
    <!-- Fixed navbar -->
    <nav class="navbar navbar-default navbar-fixed-top">
        <?php
        /*
        * Variabel $headernya diambil dari core MY_Controller
        * (application/core/MY_Controller.php)
        * */
        echo $headernya;
        ?>
    </nav>
    <div class="container">
        <?php
        /*
        * Variabel $contentnya diambil dari core MY_Controller
        * (application/core/MY_Controller.php)
        * */
        echo $contentnya;
        ?>
    </div>
    <!-- Load file Javascript Bootstrap & jQuery -->
    <script src="<?php echo base_url('js/jquery.min.js'); ?>"></script>
    <script src="<?php echo base_url('js/bootstrap.min.js');
?>"></script>
</body>
</html>

```

Selanjutnya buat lagi sebuah file dengan nama **header.php**, lalu simpan pada folder xampp/htdocs/login\_role\_ci/application/views/template/backend/. Berikut ini kodenya :

```

<div class="container">
    <div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#navbar" aria-expanded="false" aria-
controls="navbar">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand" href="#">My Notes Code</a>
    </div>
    <div id="navbar" class="navbar-collapse collapse">
        <ul class="nav navbar-nav">
            <li><a href="<?php echo base_url('index.php/page/home');
?>">Home</a></li>

```

```

        <li><a href="<?php echo base_url('index.php/page/berita');
?>">Berita</a></li>
        <?php
        // Cek role user
        if($this->session->userdata('role') == 'admin'){ // Jika
role-nya admin
            ?>
            <li><a href="<?php echo
base_url('index.php/page/pengguna'); ?>">Pengguna</a></li>
            <?php
            }
            ?>
            <li><a href="<?php echo base_url('index.php/page/kontak');
?>">Kontak</a></li>
        </ul>
        <ul class="nav navbar-nav navbar-right">
            <li><a href="<?php echo base_url('index.php/auth/logout');
?>">Logout</a></li>
        </ul>
    </div><!--/.nav-collapse -->
</div>

```

## STEP 11 - HOME

Tahap selanjutnya kita akan membuat file untuk halaman home (halaman setelah login). Silahkan buat file dengan nama **home.php**, lalu simpan pada folder `xampp/htdocs/login_role_ci/application/views/`. Berikut tampilan dan kodenya :

My Notes Code
Home
Berita
Pengguna
Kontak
Logout

Selamat datang  
**Rizaldi Maulidia Achmad**

**Role**  
Admin

**Hak Akses**  
1. Akses menu home. Aksi yang diperbolehkan : Read  
2. Akses menu berita. Aksi yang diperbolehkan : Create, Read, Update, Delete  
3. Akses menu pengguna. Aksi yang diperbolehkan : Create, Read, Update, Delete  
4. Akses menu Kontak. Aksi yang diperbolehkan : Read

```

<h2 style="margin-top: 0;">
    <small>Selamat datang</small>
    <br />
    <?php echo $this->session->userdata('nama') ?>
</h2>
<hr />
<div class="form-group">
    <label>Role</label>
    <br /><?php echo ucwords($this->session->userdata('role')) ?>
</div>
<?php
// Cek role user
if($this->session->userdata('role') == 'admin'){ // Jika role-nya admin

```

```

?>
<div class="form-group">
    <label>Hak Akses</label>
    <br />
    <ol style="margin-left: -25px;">
        <li>
            Akses menu home. Aksi yang diperbolehkan : Read
        </li>
        <li>
            Akses menu berita. Aksi yang diperbolehkan : Create,
Read, Update, Delete
        </li>
        <li>
            Akses menu pengguna. Aksi yang diperbolehkan : Create,
Read, Update, Delete
        </li>
        <li>
            Akses menu Kontak. Aksi yang diperbolehkan : Read
        </li>
    </ol>
</div>
<?php
}else{ // Jika role-nya operator
?>
<div class="form-group">
    <label>Hak Akses</label>
    <br />
    <ol style="margin-left: -25px;">
        <li>
            Akses menu home. Aksi yang diperbolehkan : Read
        </li>
        <li>
            Akses menu berita. Aksi yang diperbolehkan : Read,
Update, Delete
        </li>
        <li>
            Akses menu Kontak. Aksi yang diperbolehkan : Read
        </li>
    </ol>
</div>
<?php
}
?>

```







Dalam kode diatas terdapat sebuah kode kondisi *if(\$this->session->userdata ('role') = 'admin')*. Kode itu untuk menampilkan data jika user yang login adalah admin. Sebaliknya jika bukan admin akan ditampilkan apa.

## STEP 12 - BERITA

Tahap selanjutnya kita akan membuat file untuk halaman berita (halaman setelah login). Silahkan buat file dengan nama **berita\_view.php**, lalu simpan pada folder `xampp/htdocs/login_role_ci/application/views/`. Berikut tampilan dan kodenya :

## Berita

TAMBAH DATA

No	Judul	Deskripsi	Tanggal	Aksi
1	Lorem ipsum dolor sit consectetur adipiscing elit	Duis arcu augue, tincidunt id posuere suscipit, aliquam a libero. Cras varius neque quis odio varius commodo. Nullam in tellus gravida, porttitor tortor nec, bibendum nisi...	12 Januari 2019	 
2	Lorem ipsum dolor sit consectetur adipiscing elit	Duis arcu augue, tincidunt id posuere suscipit, aliquam a libero. Cras varius neque quis odio varius commodo. Nullam in tellus gravida, porttitor tortor nec, bibendum nisi...	12 Januari 2019	 
3	Lorem ipsum dolor sit consectetur adipiscing elit	Duis arcu augue, tincidunt id posuere suscipit, aliquam a libero. Cras varius neque quis odio varius commodo. Nullam in tellus gravida, porttitor tortor nec, bibendum nisi...	12 Januari 2019	 

```

<?php
// Cek role user
if($this->session->userdata('role') == 'admin'){ // Jika role-nya admin
    ?>
    <div class="pull-right">
        <a href="" class="btn btn-success">TAMBAH DATA</a>
    </div>
    <?php
}
?>
<h2 style="margin-top: 0;margin-bottom: 0;">Berita</h2>
<div class="clearfix"></div>
<hr />
<div class="table-responsive">
    <table class="table table-bordered table-hover">
        <thead>
            <tr>
                <th>No</th>
                <th>Judul</th>
                <th>Deskripsi</th>
                <th>Tanggal</th>
                <th style="width: 80px;">Aksi</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>1</td>
                <td>Lorem ipsum dolor sit consectetur adipiscing
elit</td>
                <td>Duis arcu augue, tincidunt id posuere suscipit,
aliquam a libero. Cras varius neque quis odio varius commodo. Nullam in
tellus gravida, porttitor tortor nec, bibendum nisi...</td>
                <td>12 Januari 2019</td>
                <td>
                    <a href="" class="btn btn-default btn-xs"><i
class="glyphicon glyphicon-pencil"></i></a>
                    <a href="" class="btn btn-danger btn-xs"><i
class="glyphicon glyphicon-trash"></i></a>
                </td>
            </tr>
            <tr>
                <td>2</td>
                <td>Lorem ipsum dolor sit consectetur adipiscing
elit</td>

```



```

        <td>Duis arcu augue, tincidunt id posuere suscipit,
        aliquam a libero. Cras varius neque quis odio varius commodo. Nullam in
        tellus gravida, porttitor tortor nec, bibendum nisi...</td>
        <td>12 Januari 2019</td>
        <td>
            <a href="" class="btn btn-default btn-xs"><i
            class="glyphicon glyphicon-pencil"></i></a>
            <a href="" class="btn btn-danger btn-xs"><i
            class="glyphicon glyphicon-trash"></i></a>
        </td>
    </tr>
    <tr>
        <td>3</td>
        <td>Lorem ipsum dolor sit consectetur adipiscing
        elit</td>
        <td>Duis arcu augue, tincidunt id posuere suscipit,
        aliquam a libero. Cras varius neque quis odio varius commodo. Nullam in
        tellus gravida, porttitor tortor nec, bibendum nisi...</td>
        <td>12 Januari 2019</td>
        <td>
            <a href="" class="btn btn-default btn-xs"><i
            class="glyphicon glyphicon-pencil"></i></a>
            <a href="" class="btn btn-danger btn-xs"><i
            class="glyphicon glyphicon-trash"></i></a>
        </td>
    </tr>
</tbody>
</table>
</div>

```

```

if($this->session->userdata('role') == 'admin'){ // Jika role-nya admin
?><div class="pull-right">
<a href="" class="btn btn-success">TAMBAH DATA</a> </div>}

```

Kode diatas berfungsi untuk jika user yang login adalah admin. Mekan tombol tambah akan dimunculkan, Tapi jika user yang login bukan admin, maka tombol tambah akan disembunyikan.

## LATIHAN STEP 12 – PENILAIAN UH 1

Berdasarkan materi CRUD CodeIgniter yang telah dipraktikkan sebelumnya, buat aksi CRUD berita dapat dijalankan!

Arahan:

1. Buat tabel baru didalam dengan nama **berita** didalam database multilevel dengan struktur dibawah ini:

no	int(11)	primary key	AI
judul	varchar(100)		
deskripsi	text		
tanggal	varchar(50)		

2. Buat controller dengan nama **Berita.php**, model dengan nama **Berita\_model.php**, dan view yang berisi form inputan untuk mengisi data dengan nama **add\_berita\_view.php** dan form editnya dengan nama **edit\_berita\_view.php**.





3. Isikan masing-masing file tersebut seperti pada modul CRUD CodeIgniter dengan beberapa modifikasi, sesuai dengan struktur tabel yang telah dibuat.
4. Modifikasi pula file **berita\_view.php** sehingga dapat melakukan load data tabel dari tabel yang telah dibuat, dan masukkan link pada button delete dan updatenya.
5. Matikan fungsi aksi button update dan hapusnya jika hak aksesnya adalah operator menggunakan perintah if berdasarkan role nya.

## STEP 13 - PENGGUNA

Tahap selanjutnya kita akan membuat file untuk halaman berita (halaman setelah login). Silahkan buat file dengan nama **pengguna\_view.php**, lalu simpan pada folder `xampp/htdocs/login_role_ci/application/views/`. Berikut tampilan dan kodenya :

My Notes Code	Home	Berita	Pengguna	Kontak	Logout
---------------	------	--------	----------	--------	--------

Pengguna				TAMBAH DATA
No	Username	Nama	Role	Aksi
1	rizaldi	Rizaldi Maulidia Achmad	Admin	 
2	siska	Siska Melina Rachman	Operator	 

```
<div class="pull-right">
    <a href="" class="btn btn-success">TAMBAH DATA</a>
</div>
<h2 style="margin-top: 0;margin-bottom: 0;">Pengguna</h2>
<div class="clearfix"></div>
<hr />
<div class="table-responsive">
    <table class="table table-bordered table-hover">
        <thead>
            <tr>
                <th>No</th>
                <th>Username</th>
                <th>Nama</th>
                <th>Role</th>
                <th style="width: 80px;">Aksi</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>1</td>
                <td>rizaldi</td>
                <td>Rizaldi Maulidia Achmad</td>
                <td>Admin</td>
                <td>
                    <a href="" class="btn btn-default btn-xs"><i
class="glyphicon glyphicon-pencil"></i></a>
```

```

                <a href="" class="btn btn-danger btn-xs"><i
class="glyphicon glyphicon-trash"></i></a>
            </td>
        </tr>
        <tr>
            <td>2</td>
            <td>siska</td>
            <td>Siska Melina Rachman</td>
            <td>Operator</td>
            <td>
                <a href="" class="btn btn-default btn-xs"><i
class="glyphicon glyphicon-pencil"></i></a>
                <a href="" class="btn btn-danger btn-xs"><i
class="glyphicon glyphicon-trash"></i></a>
            </td>
        </tr>
    </tbody>
</table>
</div>

```

## LATIHAN STEP 13 – PENILAIAN UH 2

Berdasarkan materi CRUD CodeIgniter yang telah dipraktikkan sebelumnya, buat aksi CRUD pengguna dapat dijalankan!

Arahan:

1. Buat controller dengan nama **Pengguna.php**, model dengan nama **Pengguna\_model.php**, dan view yang berisi form inputan untuk mengisi data dengan nama **add\_pengguna\_view.php** dan form editnya dengan nama **edit\_pengguna\_view.php**.
3. Isikan masing-masing file tersebut seperti pada modul CRUD CodeIgniter dengan beberapa modifikasi, sesuai dengan struktur tabel **user** yang telah dibuat diatas.
4. Modifikasi pula file **pengguna\_view.php** sehingga dapat melakukan load data tabel dari tabel yang telah dibuat, dan masukkan link pada button delete dan updatenya.
5. Matikan fungsi aksi button update dan hapusnya jika hak aksesnya adalah operator menggunakan perintah if berdasarkan role nya.

## STEP 14 - KONTAK

Tahap selanjutnya kita akan membuat file untuk halaman berita (halaman setelah login). Silahkan buat file dengan nama **kontak.php**, lalu simpan pada folder xampp/htdocs/login\_role\_ci/application/views/. Berikut tampilan dan kodenya :

My Notes Code	Home	Berita	Pengguna	Kontak	Logout
<p>My Notes Code</p> <p>Telepon : 022-0123456</p> <p>Jl.Jend. Gatot Subroto No.123 Bandung</p>					

```
<h3>My Notes Code</h3>  
<br>Telepon : 022-0123456  
<br>Jl.Jend. Gatot Subroto No.123 Bandung
```

## LATIHAN AKHIR

Buat 1 hak akses tambahan dengan nama **subscriber** dengan hak akses sebagai berikut:

- 1 .Akses Menu Home. Aksi yang bisa di lakukan : Read
- 2.Akses Menu Berita. Aksi yang bisa di lakukan : Read
- 3.Akses Menu Kontak. Aksi yang bisa di lakukan : Read

## PENGETESAN

Silahkan lakukan pengetesan menggunakan kedua hak akses yang telah dibuat sebelumnya dengan login satu persatu dan lihat perbedaan antara kedua hak akses tersebut.