

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB LANJUT**

Jobsheet-5: Multi Level User



Disusun oleh:
Livia Yurike Khuril Maula
D4 TI – 2A / 16
NIM : 1841720025

**POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI D4 TEKNIK INFORMATIKA**

MARET 2020



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-5: Multi Level User
Mata Kuliah Pemrograman Web Lanjut
Pengampu: Tim Ajar Pemrograman Web Lanjut
Februari 2020

Topik

Membuat sebuah login dengan beberapa level user.

Tujuan

Mahasiswa diharapkan dapat:

1. Memahami cara membuat halaman login
2. Memahami cara membuat halaman login dengan beberapa level user yang berbeda

Praktikum – Bagian 1: Membuat Login dengan Multi Level User

Langkah	Keterangan
1	<p>Buat tampilan login (views/login/index.php), dan tambahkan code berikut pada views/login/index.php:</p> <pre> 1 <?php 2 form_open('login/proses_login'); 3 > 4 <div class="container py-5"> 5 <div class="row"> 6 <div class="col-md-12"> 7 <div class="row"> 8 <div class="col-md-6 mx-auto"> 9 10 <!-- form card login --> 11 <div class="card rounded-0"> 12 <div class="card-header"> 13 <h3 class="mb-0">Login</h3> 14 </div> 15 <div class="card-body"> 16 <form class="form" role="form" autocomplete="off" id="formLogin" novalidate="" method="POST"> 17 <div class="form-group"> 18 <label for="uname1">Username</label> 19 <input type="text" class="form-control form-control-lg rounded-0" name="uname1" id="uname1" 20 required=""> 21 <div class="invalid-feedback">Oops, you missed this one.</div> 22 </div> 23 <div class="form-group"> 24 <label>Password</label> 25 <input type="password" class="form-control form-control-lg rounded-0" name="pwd1" id="pwd1" 26 required="" autocomplete="new-password"> 27 <div class="invalid-feedback">Enter your password too!</div> 28 </div> 29 <button type="submit" class="btn btn-success btn-lg float-right" id="btnLogin">Login</button> 30 </form> 31 </div> 32 </div> 33 </div> 34 </div> 35 </div> 36 </div> 37 <!--/row--> 38 </div> 39 <!--/col--> 40 </div> 41 <!--/row--> 42 </div> 43 <!--/container--> 44 45 <?php 46 form_close(); 47 > </pre>

2

Buatlah sebuah **controller** dengan nama **login (controller/login)**

```

application > controllers > login.php
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class login extends CI_Controller {
6
7      public function index()
8      {
9          $data['title']='Login';
10         $this->load->view('template/header_login',$data);
11         $this->load->view('login/index');
12         $this->load->view('template/footer');
13     }
14
15     public function proses_login() {
16         $username=$this->input->post('uname1');
17         $password=$this->input->post('pwd1');
18
19         $ceklogin=$this->login_model->login($username,$password);
20
21         if($ceklogin) {
22
23         }
24     }
25 }
26
27
28 /* End of file login.php */
29

```

3

Tambahkan sebuah **code** pada **views/template** dengan nama **header_login (views/template/header_login)**

```

application > views > template > header_login.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <!-- Required meta tags -->
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8      <!-- Bootstrap CSS -->
9      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
10      integrity="sh384-MCw98/SFnGE8fJT3GXwE0ngsV7Zt27NXFoaoApmYm81iuXoPkF03wJ8ERdknLPM0" crossorigin="anonymous">
11
12      <!-- style CSS -->
13      <style>
14          .badge{
15              margin-left:3px;
16          }
17      </style>
18
19      <!-- <title>Hello, world</title> -->
20      <title><?= $title ?></title>
21
22  </head>
23  <body>
24
25  </body>
26  </html>

```

4

Ketika dijalankan di browser, hasilnya akan seperti gambar berikut:

5

Setelah berhasil membuat sebuah form login maka selanjutnya kita akan membuat logic nya. Logic ini digunakan untuk proses validasi data antara inputan user dengan database yang sudah ada.

6

Sebelum membuat *logic*-nya terlebih dahulu buat tabelnya pada **database Kuliah_CI** (yang pernah dibuat sebelumnya) dengan nama **tabel user** dan tambahkan 2 data, sebagai contoh yaitu **admin** dan **user**

[Edit dikotak] [Ubah] [Buat]

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	id	int(20)		Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
<input type="checkbox"/>	2	username	varchar(20)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	3	password	varchar(20)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	4	level	varchar(20)	utf8mb4_general_ci	Tidak	Tidak ada			Ubah Hapus Lainnya

☐ Pilih Semua Dengan pilihan: Jelajahi Ubah Hapus Utama Unik Indeks Teks penuh

+ Opsi

				id	username	password	level
<input type="checkbox"/>	Ubah	Salin	Hapus	1	Admin	admin	admin
<input type="checkbox"/>	Ubah	Salin	Hapus	2	user	user	user

☐ Pilih Semua Dengan pilihan: Ubah Salin Hapus Ekspor

7

Setelah kita berhasil membuat sebuah tabel untuk menyimpan *username*, *password* dan *level* maka selanjutnya kita akan membuat model dengan nama **login_model** (**models/login_model**)

```

application > models > login_model.php
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class login_model extends CI_Model {
6
7      // var_dump($username);
8      // var_dump($password);
9      // die();
10
11     $this->db->select('username,password,level');
12     $this->db->from('user');
13     $this->db->where('username',$username); //maksudnya adalah username
14     $this->db->where('password'$password);
15     $this->db->limit(1); //maksudnya data yang diambil cuma 1
16
17     $query=$this->db->get();
18     if($query->num_rows()==1)//jika data disimpan
19     {
20         return $query->result();
21     } else {
22         return false;
23     }
24 }
25
26
27 /* End of file login_model.php */
28

```

Keterangan :

1. Line 6 sebuah function *login* dimana terdapat 2 parameter yaitu *\$username* dan *\$password*
2. Line 11 untuk melakukan *select* mana saja yang akan digunakan. Jika kita lihat terdapat 3 parameter yaitu *username*, *password*, dan *level* -> untuk penulisan ini semua harus sesuai dengan nama *field* yang ada pada *tabel*-nya (**tabel user**).
3. Line 12 adalah nama *tabel*-nya (**tabel user**)
4. Line 13 adalah selama inputan *user* yang tersimpan pada *\$username* sama dengan *username* pada *tabel user*
5. Line 14 adalah selama inputan *user* yang tersimpan pada *\$password* sama dengan *password* pada *tabel user*
6. Line 15 adalah data yang diambil cuma 1
7. Line 16 adalah sebuah variabel baru dengan nama *\$query* yang digunakan untuk mengambil semua data
8. Line 18 – 24 adalah sebuah pengkondisian dimana jika *\$query* menemukan 1 *row* pada *tabel* atau dengan kata lain data berhasil ditemukan maka nilai pengembalian querynya akan mencetak isi dari *\$query(\$query->result())* dan jika tidak menemukan maka nilai kembaliannya kita set menjadi *false*

8

Langkah berikutnya tambahkan **form_open** dan **form_close** pada *code* **views/login/index.php**

```
application > views > Login > index.php
1  <?php
2  form_open('login/proses_login');
3  ?>
4  <div class="container py-5">
5      <div class="row">
6          <div class="col-md-12">
7              <div class="row">
8                  <div class="col-md-6 mx-auto">
40                  </div>
41              <!--/row-->
42          </div>
43      <!--/container-->
44
45  <?=
46  form_close();
47  ?>
```

Keterangan:

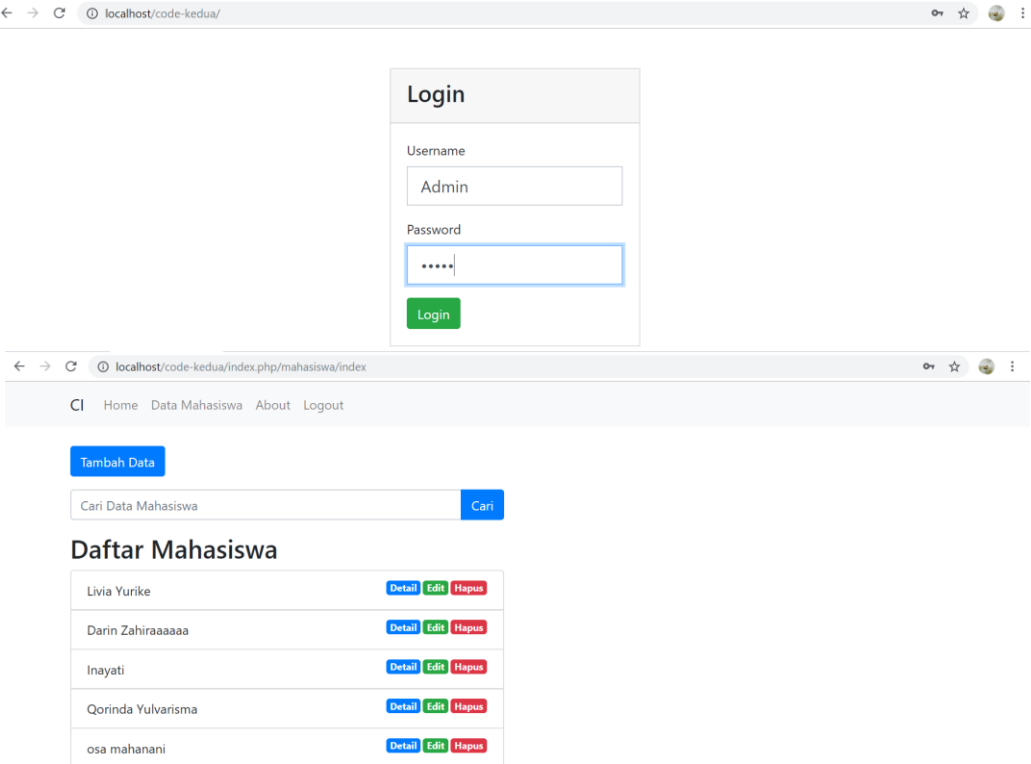
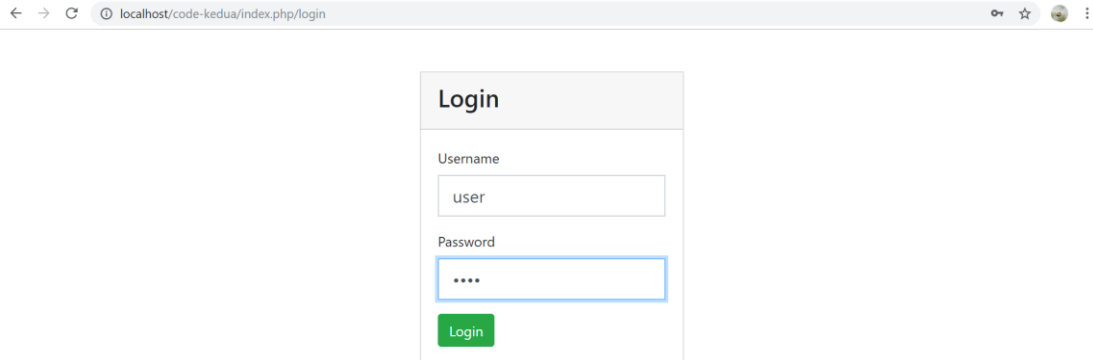
1. Line 2 artinya adalah *form* yang digunakan untuk membuka halaman proses *login* pada *controller login*
2. Line 44 artinya adalah untuk menutup *form* dikarenakan tadi terdapat perintah *form_open*

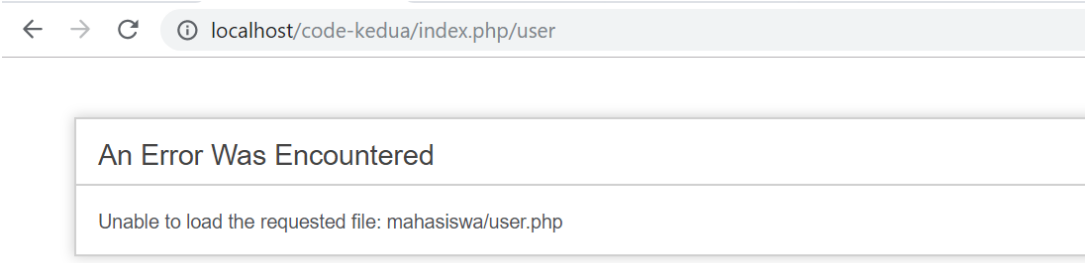

9

Buatlah sebuah *function* dengan nama **proses_login** pada **controller login**. Tapi sebelum kita membuat **function proses_login** maka terlebih dahulu kita tambahkan sebuah **function construct**

```
application > controllers > login.php
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class login extends CI_Controller {
6
7      public function __construct() {
8          parent::__construct();
9          $this->load->helper('url');
10         $this->load->helper('form');
11         $this->load->model('login_model');
12         // $this->load->library('session');
13     }
14
15     public function index()
16     {
17         $data['title']='Login';
18         $this->load->view('template/header_login',$data);
19         $this->load->view('login/index');
20         $this->load->view('template/footer');
21     }
22 }
```

	<p>Keterangan : Jika pada config/autoload sudah kalian setting maka kita tidak perlu menambahkan lagi seperti pada line 11. Dikarenakan pada config/autoload/library sudah diset maka pada <i>function</i> construct tidak perlu ditambahkan lagi.</p>
10	<p>Langkah berikutnya menambahkan function proses_login pada controller login</p> <pre> 23 public function proses_login() { 24 \$username=\$this->input->post('uname1'); 25 \$password=\$this->input->post('pwd1'); 26 27 \$ceklogin=\$this->login_model->login(\$username,\$password); 28 29 if(\$ceklogin) { 30 31 foreach(\$ceklogin as \$row); 32 // kita set userdata pada session dengan nama user dan isi 33 \$this->session->set_userdata('user',\$row->username); 34 \$this->session->set_userdata('level',\$row->level); 35 36 // var_dump(\$row) 37 if(\$this->session->userdata('level')=="admin") 38 { 39 redirect('mahasiswa/index'); 40 } 41 elseif(\$this->session->userdata('level')=="user") 42 { 43 redirect('mahasiswa/user'); 44 } 45 } 46 else { 47 redirect('login/index',"refresh"); 48 } 49 } </pre> <p>Keterangan :</p> <ol style="list-style-type: none"> 1. Line 25 dan 26 adalah sebuah variabel baru dimana isinya adalah inputan dari <i>user</i>. <i>User</i> akan mengirimkan data yang terkirim dengan metode POST yang tersimpan sementara pada uname1 dan pwd1 (uname1 dan pwd1 adalah name pada inputan <i>username</i> dan <i>password</i> pada page login/index pada tag input). 2. Line 28 adalah variabel baru(\$cek_login) isinya adalah sebuah model(login_model) dimana model(login_model) tersebut akan menjalankan sebuah <i>function</i>(function login) dengan nama login(yang telah kita buat sebelumnya) dan mempunyai 2 paramater yaitu \$username dan \$password dimana isinya adalah isi dari line 25 dan 26, yang nanti akan dikirimkan ke <i>function</i> login pada login_model. 3. Line 30-51 adalah sebuah pengkondisian jika \$cek_login bernilai benar maka cek_login akan dipecah menjadi beberapa row (\$cek_login as \$row). setelah itu kita set_userdata menggunakan <i>session</i> dengan nama variabel barunya adalah <i>user</i> sedangkan isinya adalah <i>username</i> (line 34) dan kita juga set_userdata menggunakan <i>session</i> dengan nama variabelnya level (line 35) sedangkan isinya adalah level. Setelah kita pecah menjadi beberapa row dan mempunyai isi yang berbeda-beda maka langkah berikutnya kita buat pengkondisian jika <i>session</i> userdata('level') berisi admin maka akan me-<i>redirect</i> pada halaman mahasiswa/index(line 38-41). Apabila levelnya berupa user maka akan meredirect ke halaman <i>user</i> (line 42-45). Jika \$cek_login bernilai salah maka akan me-<i>redirect</i> ke halaman <i>login</i> kembali (line 47 - 51).

11	<p>Pada <i>line</i> 25 dan 26 tambahkan htmlspecialchars agar <i>user</i> tidak dapat memasukkan <i>char</i> yang special.</p> <pre><code>\$username=htmlspecialchars(\$this->input->post('uname1')); \$password=htmlspecialchars(\$this->input->post('pwd1'));</code></pre>
12	<p>Agar halaman <i>login</i> yang pertama kali di-load maka kita harus ubah konfigurasinya pada application/config/routes</p> <pre><code>application > config > routes.php 52 \$route['default_controller'] = 'login'; //rubah menjadi login yang sebelumnya home 53 \$route['404_override'] = ''; 54 \$route['translate_uri_dashes'] = FALSE; 55</code></pre>
13	<p>Hasilnya jika kita masuk menggunakan admin</p> 
14	<p>Hasilnya jika kita masuk menggunakan <i>user</i> (jika kita lihat maka hasilnya <i>page not found</i> hal ini dikarenakan kita belum membuat halaman tersebut)</p> 

	
15	<p>Hasilnya jika kita salah memasukkan <i>password</i> dan <i>username</i> (jika salah maka redirect ke halaman login tanpa ada pemberitahuan jika <i>username</i> dan <i>password</i> salah)</p> <div data-bbox="691 528 1066 943"> <h3>Login</h3> <p>Username</p> <input type="text" value="a"/> <p>Password</p> <input type="password" value="."/> <p>Login</p> </div> <div data-bbox="649 981 1104 1485"> <h3>Login</h3> <p>Username</p> <input type="text" value="a"/> <p>Password</p> <input type="password"/> <p>Log  Please fill out this field.</p> </div>
16	<p>Kita tambahkan alert agar pada saat user salah memasukkan username dan password, maka akan muncul pemberitahuan bahwa username dan password anda salah. Tambahkan code berikut pada login/index</p> <pre data-bbox="525 1653 1235 2016"><!--/form card login --> <!--cek pesan --> <div class="alert alert-info" role="alert"> <?php if(isset(\$pesan)) { echo \$pesan; }else{ echo"Masukkan username dan password anda"; } ?> </div></pre>

dan pada bagian **controller/login/proses_login** kita ubah pada *else*-nya

```
else
{
    $data['pesan']="username dan password anda salah";
    $data['title']="Login";
    $this->load->view('template/header_login',$data);
    $this->load->view('login/index',$data);
    $this->load->view('template/footer');
    //redirect('login/index','refresh');
}
```

Sehingga hasilnya akan tampak seperti berikut

Login

Username

Password

Login

Masukkan username dan password anda

Login

Username

Password

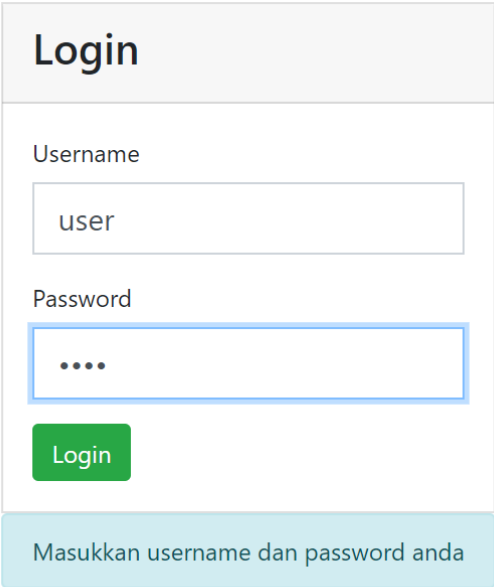
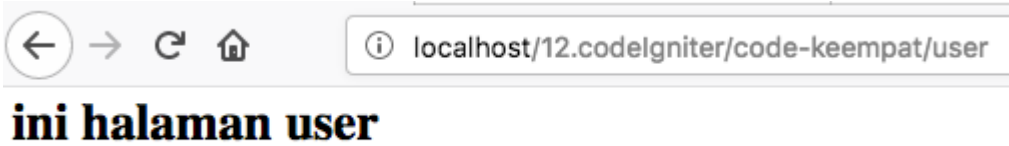
Login

username dan password anda salah

17

Jika kita lihat pada percobaan sebelumnya, pada saat kita *login* dengan menggunakan level *user* maka halaman tidak akan tampil atau *page not found*. Untuk kita akan membuat halaman *user* sederhana

18	<p>Jika dilihat pada function proses_login terdapat <i>code</i> sebagai berikut:</p> <pre> } elseif(\$this->session->userdata('level')== "user") { redirect('mahasiswa/user'); } } </pre> <p>Untuk lebih mudahnya pada line 44 ganti redirect ('mahasiswa/index') menjadi redirect('user');</p> <pre> elseif(\$this->session->userdata('level')== "user") { redirect('user'); } } </pre>
19	<p>Setelah mengganti redirect, maka yang harus dilakukan adalah membuat sebuah view dengan nama user pada folder mahasiswa. sebelum kita membuat view dengan nama user kita terlebih dahulu membuat sebuah controller dengan nama user. Buat controller user dengan <i>code</i> sebagai berikut:</p> <pre> application > controllers > user.php 1 <?php 2 3 defined('BASEPATH') OR exit('No direct script access allowed'); 4 5 class user extends CI_Controller { 6 7 public function __construct() { 8 parent::__construct(); 9 \$this->load->helper('url'); 10 \$this->load->helper('form'); 11 \$this->load->model('login_model'); 12 // \$this->load->library('session'); 13 } 14 15 public function index() 16 { 17 \$this->load->view('mahasiswa/user'); 18 } 19 20 } 21 22 /* End of file user.php */ 23 </pre>

20	<p>Membuat tampilan pada <i>view</i> dengan nama <i>user</i></p> <pre>application > views > user.php 1 <h1>ini halaman user</h1></pre> <p>Sehingga hasilnya jika kita <i>login</i> pada level <i>user</i> maka tampilan akan seperti gambar berikut</p>  
21	<p>Setelah berhasil membuat halaman <i>user</i>, maka langkah berikutnya kita membuat menu <i>logout</i>. Tambahkan <i>code</i> berikut pada views/template/header line 36</p> <pre>32 <div class="collapse navbar-collapse" id="navbarNavAltMarkup"> 33 <div class="navbar-nav"> 34 <a class="nav-item nav-link" href="<?= base_url(); ?>">Home (current) 35 <a class="nav-item nav-link" href="<?= base_url(); ?>">mahasiswa Data Mahasiswa 36 <a class="nav-item nav-link" href="<?= base_url(); ?>">About 37 <a class="nav-item nav-link" href="<?= base_url(); ?>">login/logout Logout 38 <!-- Disabled --> 39 </div> 40 </div></pre> <p>line 33 diubah menjadi seperti pada gambar</p>
22	<p>Tambahkan <i>function logout</i> pada <i>controller login</i> <i>code</i>-nya sebagai berikut</p> <pre>56 public function logout(){ 57 \$this->session->sess_destroy(); 58 redirect('login','refresh'); 59 }</pre>
23	<p>Hasilnya jika kita klik <i>logout</i> maka akan muncul halaman <i>login</i>. Pada halaman <i>user</i> kita juga tambahkan <i>code</i> untuk proses <i>logout</i> sederhana</p> <pre>application > views > user.php 1 <h1>ini halaman user</h1> 2 <a href="<?= base_url(). 'login/logout'; ?>">Logout</pre>

Jika kita lakukan perubahan langsung pada url <http://localhost/12.codelgniter/code-keempat/login/> menjadi <http://localhost/12.codelgniter/codekeempat/mahasiswa/> seperti pada gambar dibawah ini:

The top screenshot shows a 'Login' form with the following fields and buttons:

- Username:
- Password:
- Login:
- Masukkan username dan password anda

The bottom screenshot shows the 'Daftar Mahasiswa' page with the following elements:

- Navigation bar: CI, Home, Data Mahasiswa, About, Logout
- Tambah Data:
- Cari Data Mahasiswa:
- Daftar Mahasiswa table:

Livia Yurike	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
Darin Zahiraaaaa	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
Inayati	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
Qorinda Yulvarisma	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>
osa mahanani	<input type="button" value="Detail"/> <input type="button" value="Edit"/> <input type="button" value="Hapus"/>

maka kita akan dapat masuk secara langsung tanpa perlu *login* terlebih dahulu. Untuk itu kita harus menambahkan beberapa *code* pada *controller/mahasiswa* seperti pada gambar dibawah ini (line 15-19)

```

8     public function __construct()
9     {
10         // digunakan untuk menjalankan fungsi construct pa
11         parent::__construct();
12         $this->load->model('mahasiswa_model');
13         $this->load->library('form_validation');
14
15         //validasi level
16         if($this->session->userdata('level')!="Admin") {
17             redirect('login','refresh');
18         }
19     }

```

Keterangan :

Line 15-17 adalah pengkondisian dimana jika levelnya bukan admin maka halaman akan me-redirect ke halaman *login*

25

Lakukan hal yang sama pada **user (controller user)**

```

application > controllers > user.php
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class user extends CI_Controller {
6
7      public function __construct() {
8          parent::__construct();
9          $this->load->helper('url');
10         $this->load->helper('form');
11         $this->load->model('login_model');
12         // $this->load->library('session');
13
14         //validasi level
15         if($this->session->userdata('level')!="user") {
16             redirect('login','refresh')
17         }
18     }
19
20     public function index()
21     {
22         $this->load->view('mahasiswa/user');
23     }
24
25 }
26
27 /* End of file user.php */
28

```

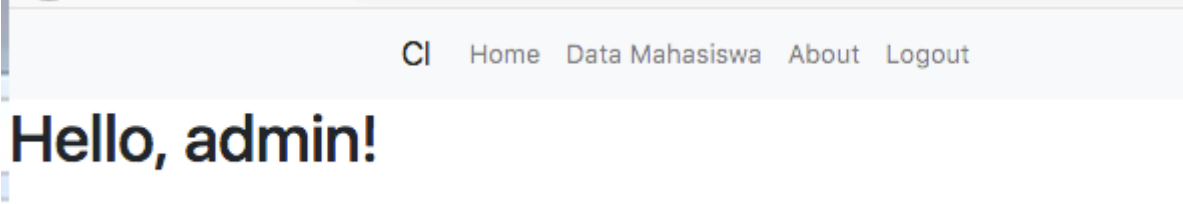
26

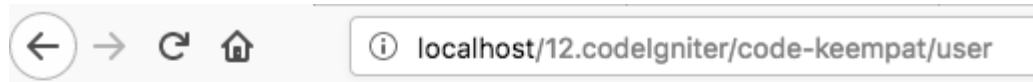
Lakukan hal yang sama pada **home (controller home)**

```

application > controllers > Home.php
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class Home extends CI_Controller {
6
7      //tambahkan parameter nama pada index
8      public function index($name='')
9      {
10         $data['title']='Home';
11         // $data adalah sebuah array dengan isi arraynya adalah nama dan diisi $name
12         $data['name']=$name;
13         $this->load->view('template/header', $data);
14         //tambahkan $data pada home/index
15         $this->load->view('home/index',$data);
16         $this->load->view('template/footer');
17
18         //validasi level
19         if($this->Session->userdata('level')!="Admin") {
20             redirect('login','refresh');
21         }
22     }
23
24 }
25
26 /* End of file Controllername.php */
27
28 ?>

```

27	<p>Tambahkan <i>code</i> berikut agar pada halaman admin tampil level <i>user</i>nya (views/home/index)</p> <pre> application > views > home > index.php 1 2 <h1>Hello, <?php \$this->session->userdata('level'); ?>!</h1> 3 4 </pre>
28	<p>Comment beberapa <i>code</i> pada <i>controller home</i> (line 12 dan hapus parameter name pada index)</p> <pre> application > controllers > Home.php 1 <?php 2 3 defined('BASEPATH') OR exit('No direct script access allowed'); 4 5 class Home extends CI_Controller { 6 7 //tambahkan parameter nama pada index 8 public function index(\$name='') 9 { 10 \$data['title']='Home'; 11 // \$data adalah sebuah array dengan isi arraynya adalah nama dan diisi \$name 12 // \$data['name']=\$name; 13 \$this->load->view('template/header', \$data); 14 //tambahkan \$data pada home/index 15 \$this->load->view('home/index',\$data); 16 \$this->load->view('template/footer'); 17 \$this->load->library('session'); 18 19 //validasi level 20 if(\$this->Session->userdata('level')!="Admin") { 21 redirect('login','refresh'); 22 } 23 } 24 25 } 26 27 /* End of file Controllername.php */ 28 29 ?> </pre> <p>Sehingga hasilnya adalah sebagai berikut:</p> 
29	<p>Lakukan hal yang sama pada <i>user</i></p> <pre> application > views > user.php 1 <h1>ini halaman user</h1> 2 3 <h1>Hello, <?php echo \$this->session->userdata('level'); ?>!</h1> 4 <a href=" <?php echo base_url(). 'login/logout' ; ?>">Logout </pre>



ini halaman user

Hello, user!

[Logout](#)