

PEMROGRAMAN WEB LANJUT
JOBSHEET 5
MULTI LEVEL USER



Oleh:
SILVI INDAH NOVITA SARI
1841720006

Program Studi D-IV Teknik Informatika
Jurusan Teknologi Informasi
Politeknik Negeri Malang
2020



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 <? form_open('login/proses_login'); 2 ? 3 <div class="container py-5"> 4 <div class="row"> 5 <div class="col-md-12"> 6 <div class="row"> 7 <div class="col-md-6 mx-auto"> 8 <!--form card login--> 9 <div class="card rounded-0"> 10 <div class="card-header"> 11 <h3 class="mb-0">Login</h3> 12 </div> 13 <div class="card-body"> 14 <form class="form" role="form" autocomplete="off" id="formlogin" novalidate="" method="POST"> 15 <div class="form-group"> 16 <label for="uname1">Username </label> 17 <input type="text" class="form-control form-control-lg rounded-0" name="uname1" id="uname1" required=""> 18 <div class="invalid-feedback">Oops, you missed this one.</div> 19 </div> 20 <div class="form-group"> 21 <label>Password</label> 22 <input type="password" class="form-control form-control-lg rounded-0" name="pwd1" id="pwd1" required="" autoc 23 <div class="invalid-feedback">Enter your password too!</div> 24 </div> 25 <button type="submit" class="btn btn-success btn-lg float-right" id="btnLogin">Login</button> 26 </form> 27 </div> 28 </div> 29 </div> 30 </div> 31 </div> 32 </div> 33 </div> 34 <? form_close(); 35 </pre>
2	<p>Buatlah sebuah <i>controller</i> dengan nama <i>login</i> (controller/login)</p>

	<pre> application > controllers > login.php > login > index 1 <?php 2 defined('BASEPATH') or exit('No direct script access allowed'); 3 4 class login extends CI_Controller 5 { 6 public function index() 7 { 8 \$data['title'] = 'Login'; 9 \$this->load->helper('form'); 10 \$this->load->view('template/header_login', \$data); 11 \$this->load->view('login/index'); 12 \$this->load->view('template/footer'); 13 } 14 } 15 </pre>
3	<p>Tambahkan sebuah <i>code</i> pada views/template dengan nama header_login (views/template/header_login)</p> <pre> application > views > template > header_login.php > html > head > link 1 <!doctype html> 2 <html lang="en"> 3 4 <head> 5 <!-- Required meta tags --> 6 <meta charset="utf-8"> 7 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"> 8 9 <!-- Bootstrap CSS --> 10 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" 11 integrity="sha384-MCw98/SFYo6E8fJ13GXXwE0ngsV7Zt27NXPFeacApeYmB11uXoPkFO3wJ3BERdnLPNO" crossorigin="anonymous"> 12 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 24 <body> </pre>
4	<p>Ketika dijalankan di browser, hasilnya akan seperti gambar berikut:</p>

Login

Username

Password

Login

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**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	username	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
3	password	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
4	level	varchar(10)	latin1_swedish_ci		No	None			Change Drop More

Server: 127.0.0.1 » Database: kuliah_ci » Table: user

Browse Structure SQL Search Insert Export

✓ Showing rows 0 - 1 (2 total, Query took 0.0016 seconds.)

SELECT * FROM `user`

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

		id	username	password	level
<input type="checkbox"/>	Edit Copy Delete	1	admin	admin	admin
<input type="checkbox"/>	Edit Copy Delete	2	user	user	user

↑ ☐ Check all With selected: Edit Copy Delete Export


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

```

	<p>Keterangan :</p> <ol style="list-style-type: none"> 1. <i>Line 6</i> sebuah <i>function login</i> dimana terdapat 2 parameter yaitu <code>\$username</code> dan <code>\$password</code> 2. <i>Line 11</i> untuk melakukan <i>select</i> mana saja yang akan digunakan. Jika kita lihat terdapat 3 parameter yaitu <i>username</i>, <i>password</i>, dan <i>level</i> -> untuk penulisan ini semua harus sesuai dengan nama <i>field</i> yang ada pada <i>tabel</i>-nya (tabel user). 3. <i>Line 12</i> adalah nama <i>tabel</i>-nya (tabel user) 4. <i>Line 13</i> adalah selama inputan <i>user</i> yang tersimpan pada <code>\$username</code> sama dengan <i>username</i> pada <i>tabel user</i> 5. <i>Line 14</i> adalah selama inputan <i>user</i> yang tersimpan pada <code>\$password</code> sama dengan <i>password</i> pada <i>tabel user</i> 6. <i>Line 15</i> adalah data yang diambil cuma 1 7. <i>Line 16</i> adalah sebuah variabel baru dengan nama <code>\$query</code> yang digunakan untuk mengambil semua data 8. <i>Line 18 – 24</i> adalah sebuah pengkondisian dimana jika <code>\$query</code> menemukan 1 <i>row</i> pada <i>tabel</i> atau dengan kata lain data berhasil ditemukan maka nilai pengembalian querynya akan mencetak isi dari <code>\$query(\$query->result())</code> dan jika tidak menemukan maka nilai kembaliannya kita set menjadi <i>false</i>
8	<p>Langkah berikutnya tambahkan form_open dan form_close pada <i>code views/login/index.php</i></p>  <pre> application > views > login > index.php > div.container.py-5 > di 1 <?= 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"> 35 <?= form_close(); 36 ?> </pre> <p>Keterangan:</p> <ol style="list-style-type: none"> 1. <i>Line 2</i> artinya adalah <i>form</i> yang digunakan untuk membuka halaman proses <i>login</i> pada <i>controller login</i> 2. <i>Line 44</i> artinya adalah untuk menutup <i>form</i> dikarenakan tadi terdapat perintah <code>form_open</code>
9	<p>Buatlah sebuah <i>function</i> dengan nama proses_login pada <i>controller login</i>. Tapi sebelum kita membuat <i>function proses_login</i> maka terlebih dahulu kita tambahkan sebuah <i>function construct</i></p>

application > controllers > login.php > ...

```
1  <?php
2  defined('BASEPATH') or exit('No direct script access allowed');
3
4  class login extends CI_Controller
5  {
6      public function __construct()
7      {
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     public function index()
15     {
16         $data['title'] = 'Login';
17         $this->load->helper('form');
18         $this->load->view('template/header_login', $data);
19         $this->load->view('login/index');
20         $this->load->view('template/footer');
21     }
22 }
23
```

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 *function* construct tidak perlu ditambahkan lagi.

10

Langkah berikutnya menambahkan **function proses_login** pada **controller login**

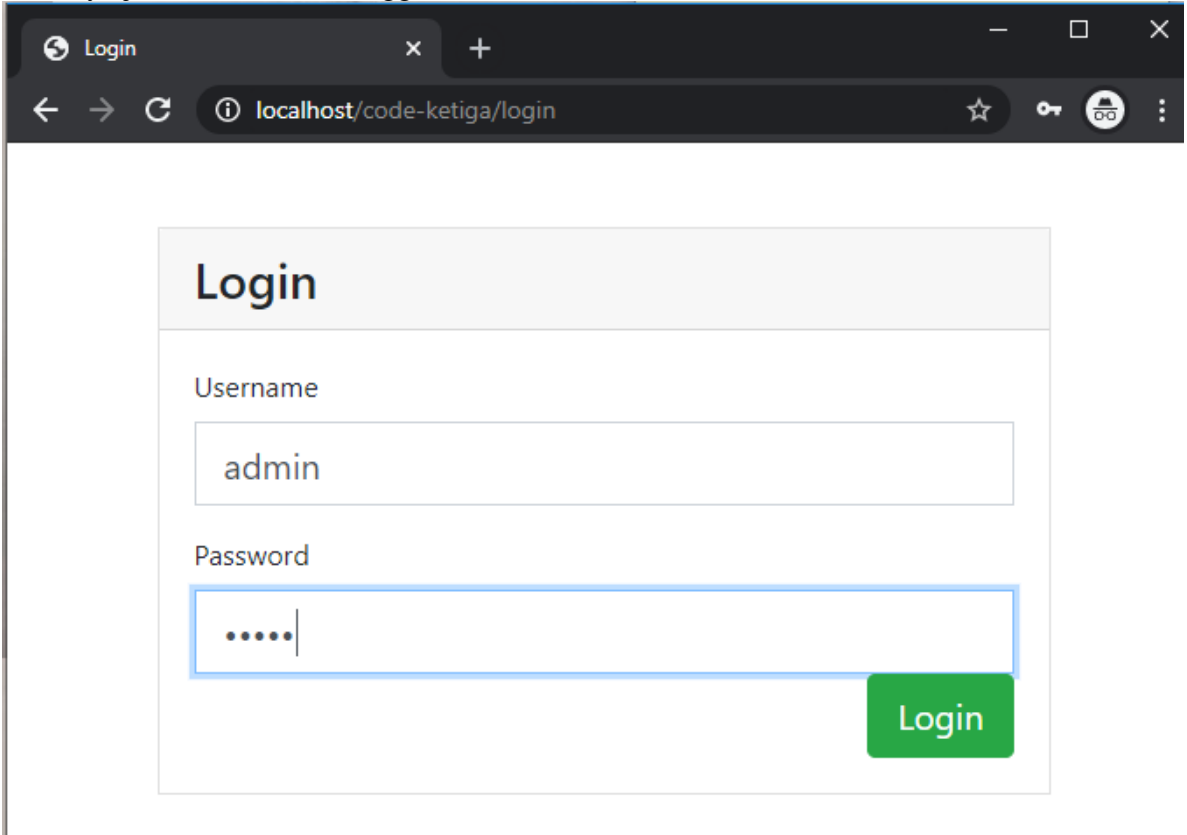
```

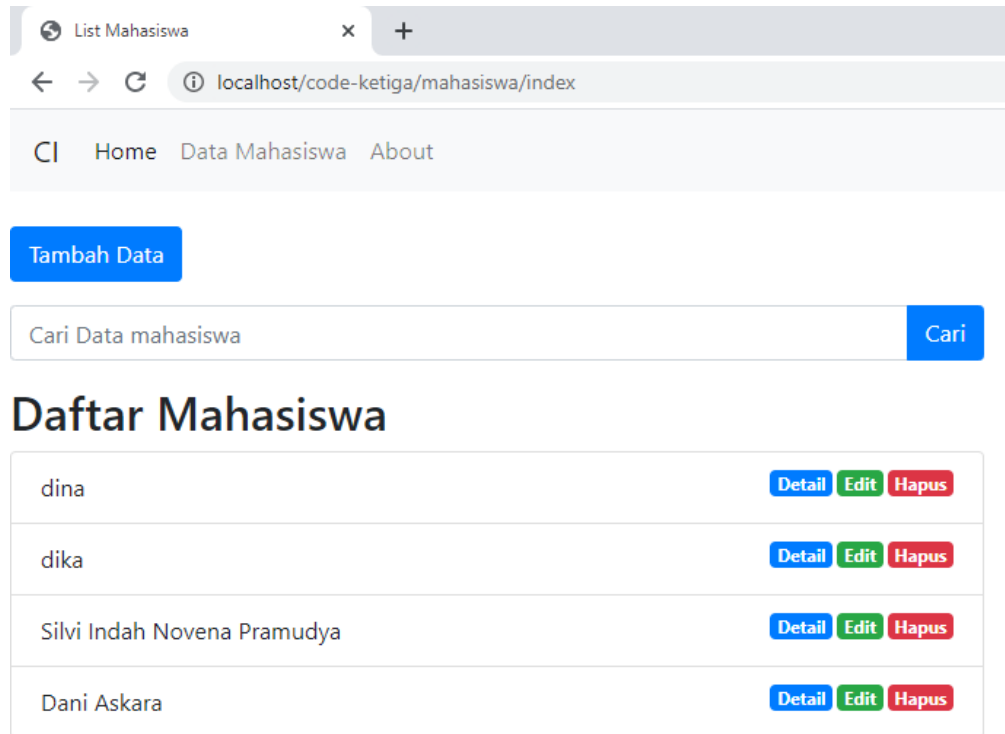
22 public function proses_login()
23 {
24     $username = $this->input->post('uname1');
25     $password = $this->input->post('pwd1');
26
27     $ceklogin = $this->login_model->login($username, $password);
28
29     if ($ceklogin) {
30         foreach ($ceklogin as $row);
31         $this->session->set_userdata('user', $row->username);
32         $this->session->set_userdata('level', $row->level);
33
34         //var_dump($row);
35         if ($this->session->userdata('level') == "admin") {
36             redirect('mahasiswa/index');
37         } else if ($this->session->userdata('level') == "user") {
38             redirect('mahasiswa/user');
39         }
40     } else {
41         redirect('login/index', 'refresh');
42     }
43 }
44 }

```

Keterangan :

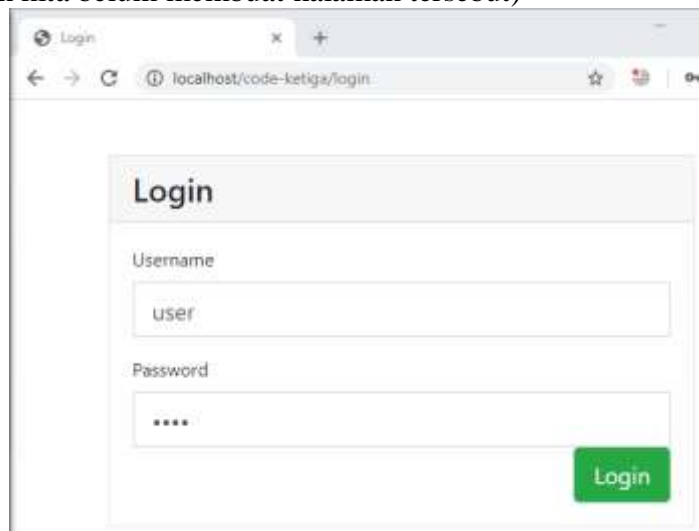
1. Line 25 dan 26 adalah sebuah variabel baru dimana isinya adalah inputan dari *user*. *User* akan mengirimkan data yang terkirim dengan metode **POST** yang tersimpan sementara pada **uname1** dan **pwd1** (**uname1** dan **pwd1** adalah **name** pada inputan *username* dan *password* pada page login/index 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 *function*(**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 *function* 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 *session* dengan nama variabel barunya adalah *user* sedangkan isinya adalah *username* (line 34) dan kita juga set_userdata menggunakan *session* 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 *session* userdata('level') berisi admin maka akan me-*redirect* pada halaman mahasiswa/index(line 38-41). Apabila levelnya berupa user maka akan meredirect ke halaman *user* (line 42-45). Jika \$cek_login bernilai salah maka akan me-*redirect* ke halaman *login* kembali (line 47 - 51).

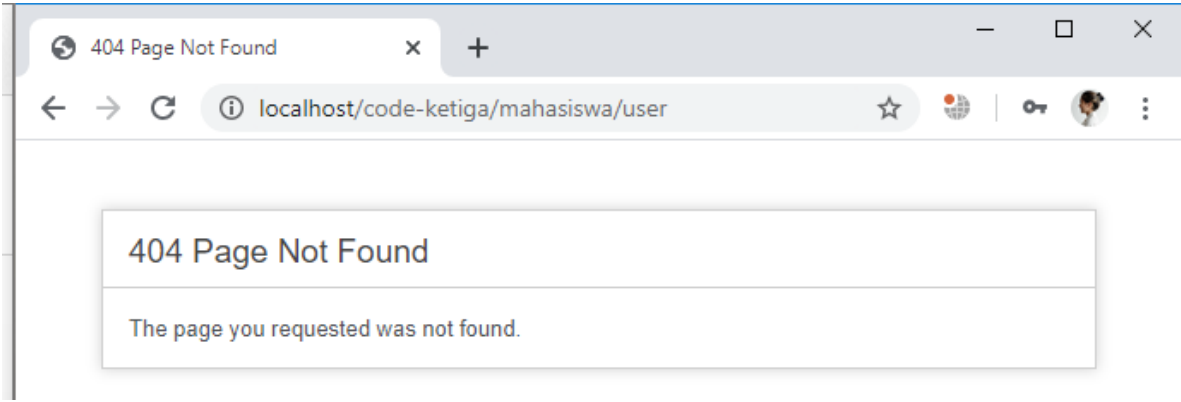
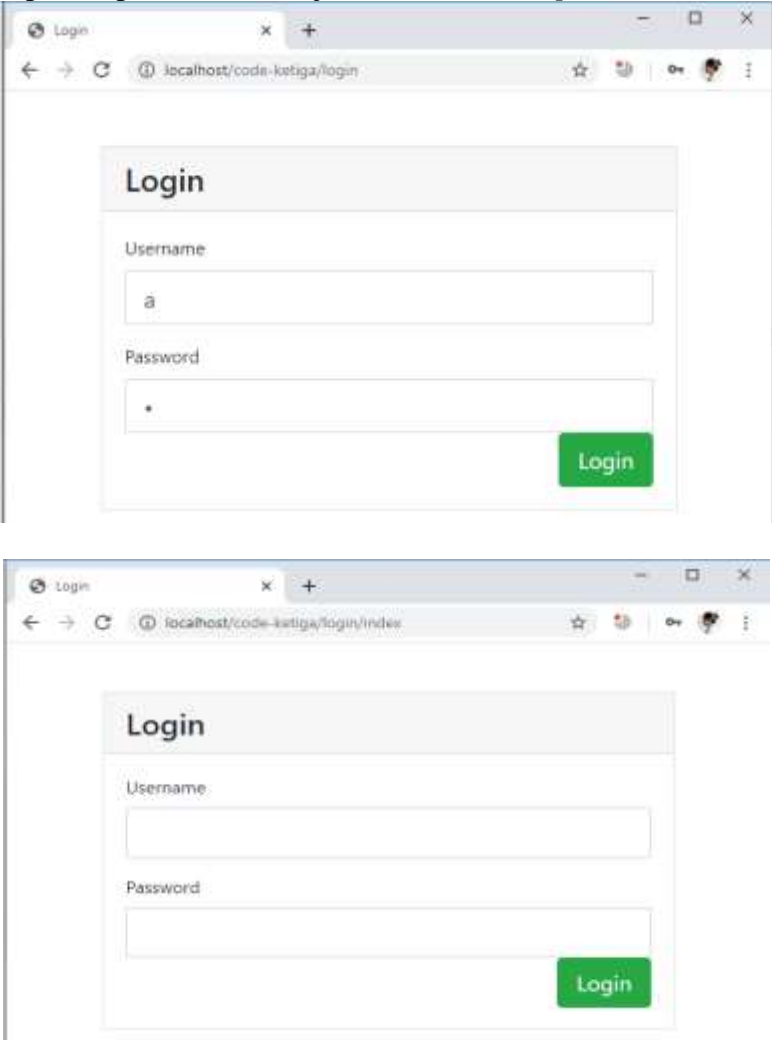
11	<p>Pada <i>line</i> 25 dan 26 tambahkan htmlspecialchars agar <i>user</i> tidak dapat memasukkan <i>char</i> yang special.</p> <pre><code>\$username = htmlspecialchars(\$this->input->post('uname1')); \$password = htmlspecialchars(\$this->input->post('pwd1'));</code></pre>
12	<p>Agar halaman <i>login</i> yang pertama kali di-load maka kita harus ubah konfigurasinya pada application/config/routes</p> <pre><code>52 \$route['default_controller'] = 'login'; //rubah menjadi home 53 \$route['404_override'] = ''; 54 \$route['translate_uri_dashes'] = FALSE; 55</code></pre>
13	<p>Hasilnya jika kita masuk menggunakan admin</p> 



14

Hasilnya jika kita masuk menggunakan *user* (jika kita lihat maka hasilnya *page not found* hal ini dikarenakan kita belum membuat halaman tersebut)



	
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> 
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>

```

<div class="alert alert-info" role="alert">
    <?php
    if (isset($pesan)) {
        echo $pesan;
    } else {
        echo "Masukkan username dan password anda";
    }
    <?>
</div>

```

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

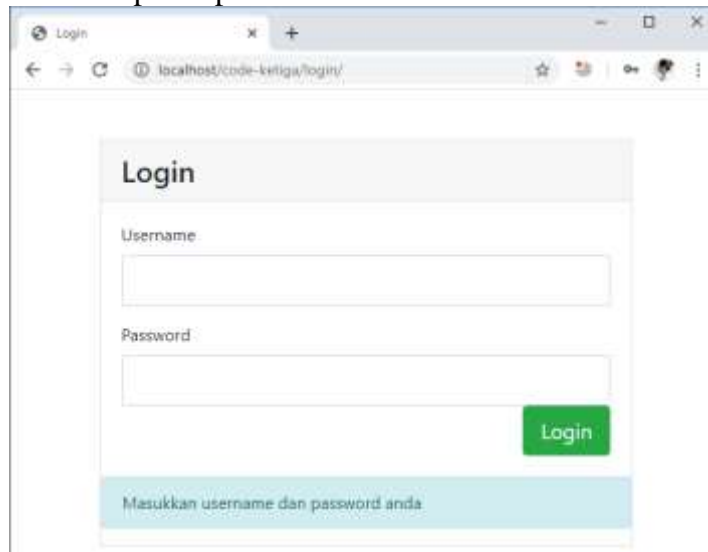
```

} else {
    $data['pesan'] = "username dan password anda salah";
    $data['title'] = 'Login';

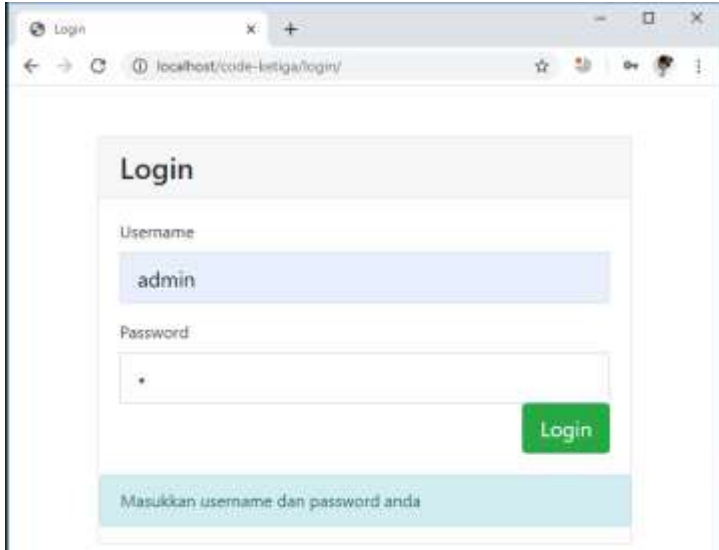
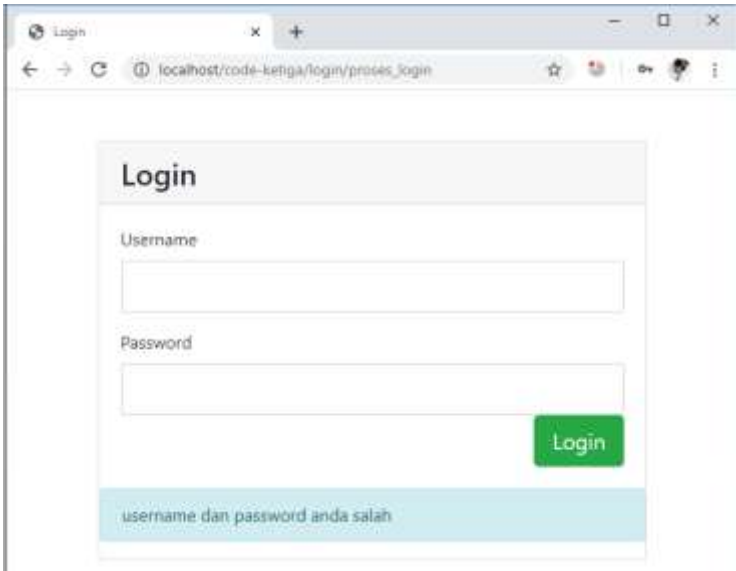
    $this->load->view('template/header_login', $data);
    $this->load->view('login/index');
    $this->load->view('template/footer');
    //redirect('login/index', 'refresh');
}

```

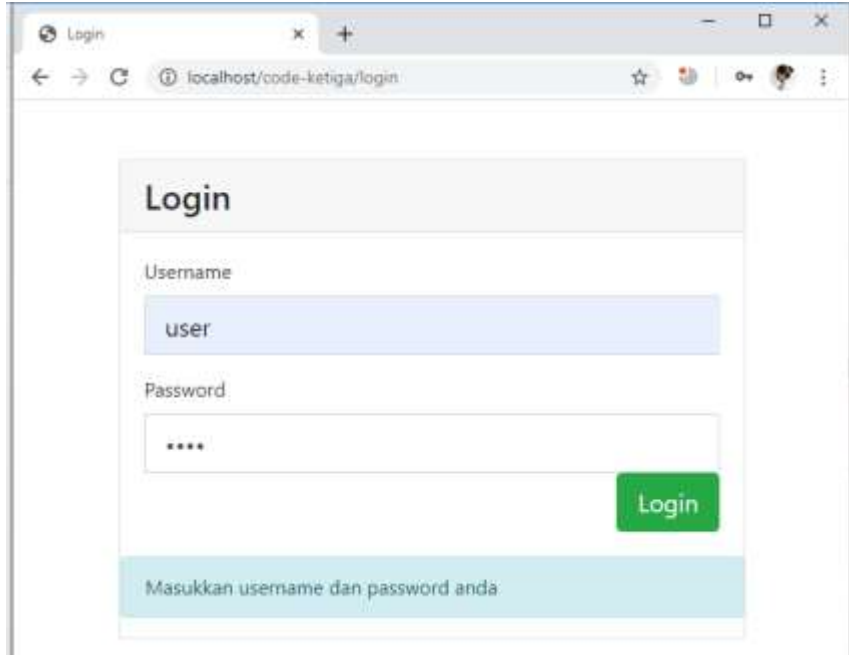
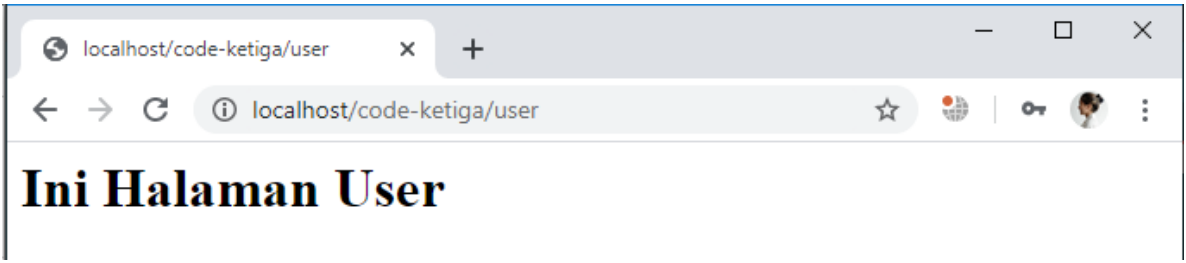
Sehingga hasilnya akan tampak seperti berikut



The screenshot shows a web browser window with the title "Login". The address bar shows "localhost/code-ketiga/login/". The page content includes a "Login" header, two input fields labeled "Username" and "Password", and a green "Login" button. Below the input fields, there is a light blue alert box with the text "Masukkan username dan password anda".

	 
17	<p>Jika kita lihat pada percobaan sebelumnya, pada saat kita <i>login</i> dengan menggunakan level <i>user</i> maka halaman tidak akan tampil atau <i>page not found</i>. Untuk kita akan membuat halaman <i>user</i> sederhana</p>
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 <i>redirect</i> ('mahasiswa/index') menjadi <i>redirect</i>('user');</p>

	<pre> } else if (\$this->session->userdata('level') == "user") { redirect('user'); } </pre>
19	<p>Setelah mengganti <i>redirect</i>, maka yang harus dilakukan adalah membuat sebuah <i>view</i> dengan nama <i>user</i> pada <i>folder mahasiswa</i>. sebelum kita membuat <i>view</i> dengan nama <i>user</i> kita terlebih dahulu membuat sebuah <i>controller</i> dengan nama <i>user</i>. Buat <i>controller</i> user dengan <i>code</i> sebagai berikut:</p> <pre> application > controllers > user.php > user > index 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 { 9 parent::__construct(); 10 //Do your magic here 11 \$this->load->helper('url'); 12 \$this->load->helper('form'); 13 \$this->load->model('login_model'); 14 //\$this->load->library('session'); 15 } 16 17 public function index() 18 { 19 \$this->load->view('mahasiswa/user'); 20 } 21 22 } 23 24 /* End of file Controllername.php */ 25 </pre>
20	<p>Membuat tampilan pada <i>view</i> dengan nama <i>user</i></p> <pre> application > views > mahasiswa > user.php > h1 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><div class="navbar-nav"> Home (current) Data Mahasiswa About Login/Logout <!--Disabled --> </div></pre> <p>line 33 diubah menjadi seperti pada gambar</p>
22	<p>Tambahkan function <i>logout</i> pada controller <i>login</i> <i>code</i>-nya sebagai berikut</p> <pre>public function logout() { \$this->session->sess_destroy(); redirect('login', 'refresh'); }</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>

```

application > views > mahasiswa > user.php > a
1  <h1>Ini Halaman User</h1>
2  <a href="<?=> base_url() . 'login/logout'; ?>">Logout</a>

```

24

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

The screenshot shows two browser windows. The top window displays a 'Login' form with fields for 'Username' and 'Password', a 'Login' button, and a message 'Masukkan username dan password anda'. The bottom window displays a page titled 'Daftar Mahasiswa' (Student List) with a navigation bar containing 'CI', 'Home', 'Data Mahasiswa', 'About', and 'Logout'. Below the navigation bar is a 'Tambah Data' button and a search bar with the text 'Cari Data mahasiswa' and a 'Cari' button. The main content area shows a table of students with their names and action buttons (Detail, Edit, Hapus).

Daftar Mahasiswa	
ari rahmat bagio	Detail Edit Hapus
diana ayu lestari	Detail Edit Hapus
ari untung	Detail Edit Hapus
Dika mahendra	Detail Edit Hapus
ayu gusti	Detail Edit 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)

	<pre> public function __construct() { parent::__construct(); \$this->load->model('mahasiswa_model'); \$this->load->library('form_validation'); //validasi level if (\$this->session->userdata('level') != "admin") { redirect('login', 'refresh'); } } </pre> <p>Keterangan : <i>Line 15-17</i> adalah pengkondisian dimana jika levelnya bukan admin maka halaman akan me-<i>redirect</i> ke halaman <i>login</i></p>
25	<p>Lakukan hal yang sama pada <i>user (controller user)</i></p> <pre> public function __construct() { parent::__construct(); //Do your magic here \$this->load->helper('url'); \$this->load->helper('form'); \$this->load->model('login_model'); //\$this->load->library('session'); //validasi level if (\$this->session->userdata('level') != "user") { redirect('login', 'refresh'); } } </pre>
26	<p>Lakukan hal yang sama pada <i>home (controller home)</i></p>

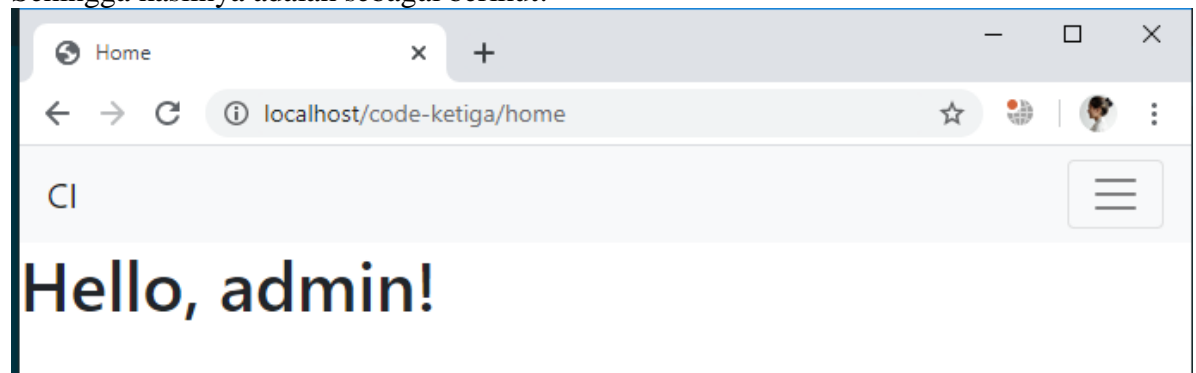
	<pre> //tambahkan parameter name pada index public function index(\$name = '') { \$data['title'] = 'Home'; \$data['name'] = \$name; \$this->load->view('template/header', \$data); //echo "Selamat Datang Di Halaman Home"; \$this->load->view('home/index', \$data); \$this->load->view('template/footer'); //validasi level if (\$this->session->userdata('level') != "admin") { redirect('login', 'refresh'); } } </pre>
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 > h1 1 <h1>Hello, <?= \$this->session->userdata('level'); ?>!</h1> </pre>
28	<p>Comment beberapa code pada controller home (line 12 dan hapus parameter name pada index)</p>

```

application > controllers > Home.php > Home > index
1  <?php
2
3  defined('BASEPATH') or exit('No direct script access allowed');
4
5  class Home extends CI_Controller
6  {
7
8      //tambahkan parameter name pada index
9      public function index()
10     {
11         $data['title'] = 'Home';
12         //$data['name'] = $name;
13         $this->load->view('template/header', $data);
14         //echo "Selamat Datang Di Halaman Home";
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 /* End of file Controllername.php */
27

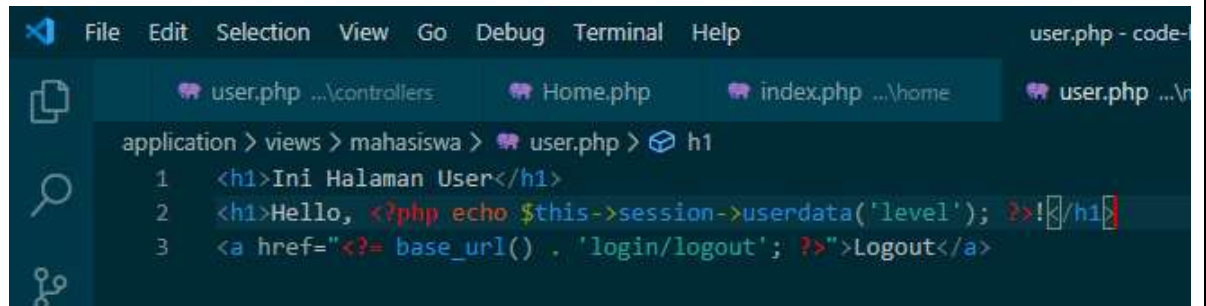
```

Sehingga hasilnya adalah sebagai berikut:



29

Lakukan hal yang sama pada *user*



The screenshot shows a code editor with a dark theme. The menu bar includes File, Edit, Selection, View, Go, Debug, Terminal, and Help. The breadcrumb path is application > views > mahasiswa > user.php > h1. The code is as follows:

```
1 <h1>Ini Halaman User</h1>
2 <h1>Hello, <?php echo $this->session->userdata('level'); ?>!</h1>
3 <a href="<?php echo base_url() . 'login/logout'; ?>">Logout</a>
```

