



MODUL CODEIGNITER #6

 <https://github.com/yysofiyan>



Session

Session merupakan variabel yang diletakkan pada sebuah web server dan dapat diakses dari halaman mana saja pada website tersebut. Session biasa digunakan untuk proses login pada aplikasi berbasis web.



Jenis Session yang akan digunakan

1

SESSION USERDATA

2

SESSION FLASHDATA

Session Userdata

Session userdata berfungsi untuk menampung variabel yang disimpan oleh cookie pada browser, yang akan digunakan untuk menyimpan data sementara ketika berhasil melakukan autentikasi seperti login.



Session Flashdata



Session flashdata berfungsi untuk menampilkan pesan sementara yang ditampilkan melalui browser, tujuan utamanya yaitu sebagai validasi setelah melakukan sebuah aksi.

Misal : jika melakukan login dengan input password yang salah, maka akan muncul pesan : Password salah!

Integrasi Session di CodeIgniter

Konfigurasi **autoload.php** di folder
application/config/ **autoload.php**

Bagian **\$autoload ['libraries']** isi dengan
'database','session' di dalam **array()**

```
● ● ●  
1 $autoload['libraries'] = array('database','session');  
2
```

Bagian **\$autoload ['helper']** isi dengan
'url','form' di dalam **array()**

```
● ● ●  
1 $autoload['helper'] = array('url','form');  
2
```





Konfigurasi **config.php** di folder application/config/ **config.php**

Bagian **\$config['base_url']** isi dengan '**http://localhost/mardira/**' diapit menggunakan kutip satu ('')

```
● ● ●  
1 $config['base_url'] = 'http://localhost/mardira/'  
2
```

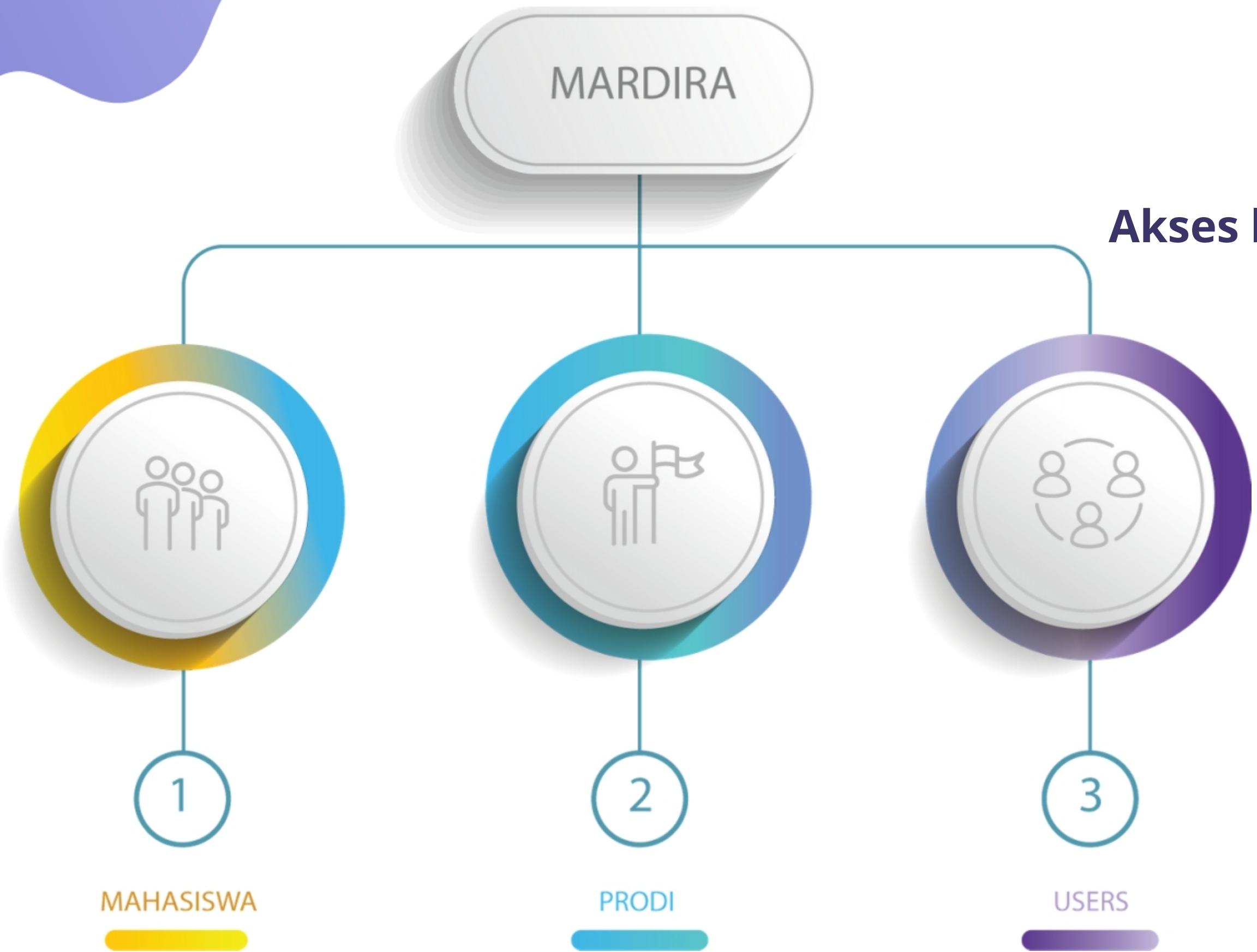
Konfigurasi Database



konfigurasi **database.php** di folder
application/config/ **database.php**

```
● ● ●  
1 $active_group = 'default';  
2 $query_builder = TRUE;  
3  
4 $db['default'] = array(  
5     'dsn'    => '',  
6     'hostname' => 'localhost',  
7     'username' => 'root',  
8     'password' => '',  
9     'database' => 'mardira',  
10    'dbdriver' => 'mysqli',  
11    'dbprefix' => '',  
12    'pconnect' => FALSE,  
13    'db_debug' => (ENVIRONMENT !== 'production'),  
14    'cache_on' => FALSE,  
15    'cachedir' => '',  
16    'char_set' => 'utf8',  
17    'dbcollat' => 'utf8_general_ci',  
18    'swap_pre' => '',  
19    'encrypt' => FALSE,  
20    'compress' => FALSE,  
21    'stricton' => FALSE,  
22    'failover' => array(),  
23    'save_queries' => TRUE  
24 );  
25 
```

Perancangan Database



Akses **localhost/phpmyadmin** di browser
pilih **SQL**, ketik perintah:
CREATE DATABASE mardira ;

Perancangan Table

Mahasiswa

```
CREATE TABLE mahasiswa ( id_mahasiswa INT(3) PRIMARY KEY NOT  
NULL AUTO_INCREMENT, nim INT(10) NOT NULL, nama_mhs  
VARCHAR(30) NOT NULL, id_prodi INT(3) NOT NULL,  
INDEX( id_prodi ));
```

Prodi

```
CREATE TABLE prodi ( id_prodi INT(3) PRIMARY KEY NOT NULL  
AUTO_INCREMENT, nama_prodi INT(10) NOT NULL);
```

Users

```
CREATE TABLE users ( id_users INT(3) PRIMARY KEY NOT NULL  
AUTO_INCREMENT, email VARCHAR(30) NOT NULL, nama_lengkap  
VARCHAR(35) NOT NULL, password VARCHAR(32) NOT NULL);
```

Insert Data

Mahasiswa

```
INSERT INTO mahasiswa ( id_mahasiswa , nim , nama_mhs , id_prodi ) VALUES  
( NULL , '18112321' , 'Mahasiswa1' , '1' ),  
( NULL , '18112323' , 'Mahasiswa2' , '2' ),  
( NULL , '181112392' , 'Mahasiswa3' , '1' );
```

Prodi

```
INSERT INTO prodi ( id_prodi , nama_prodi ) VALUES  
( NULL , 'Teknik Informatika' ),  
( NULL , 'Manajemen Informatika' );
```

Users
Users

```
INSERT INTO users ( id_users , email , nama_lengkap , password ) VALUES  
( NULL , 'admin@gmail.com' , 'administrator' , 'md5(admin'));
```



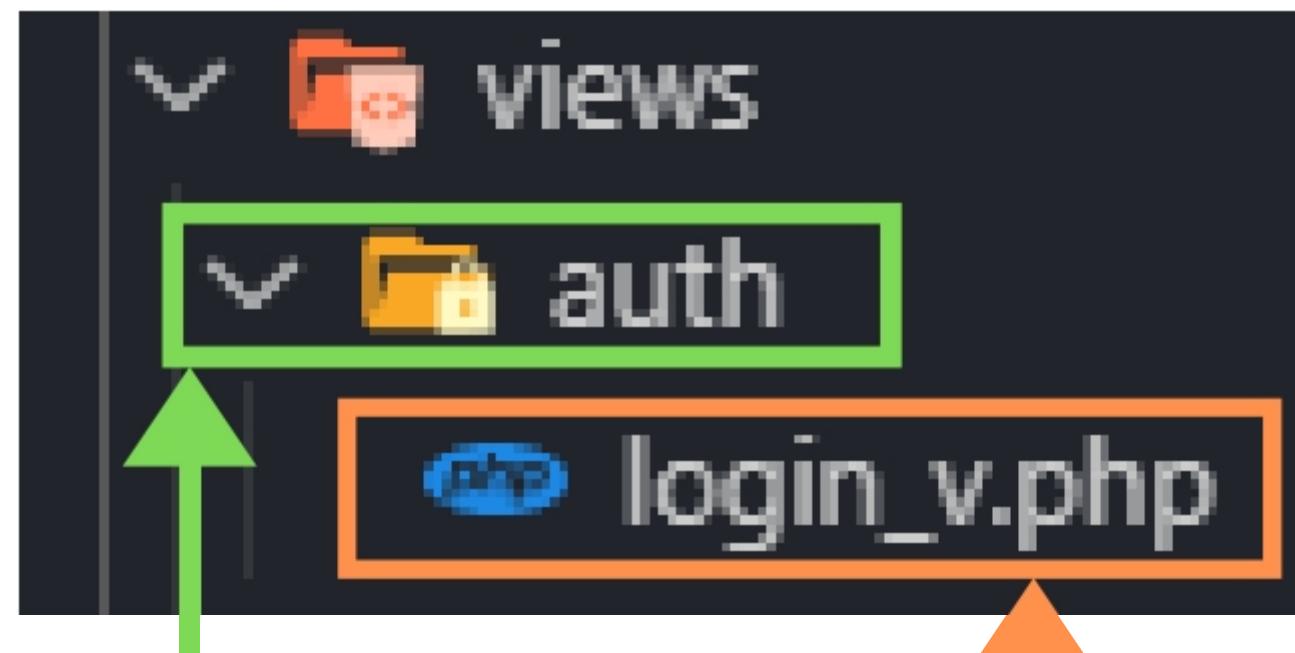
Membuat Controller

Buatlah Controller
dengan nama **Auth.php**

```
1 <?php
2
3 class Auth extends CI_Controller
4 {
5     public function login()
6     {
7         $form_open = form_open('auth/cek_login');
8         $form_close = form_close();
9         $label_email = form_label('Email', 'email');
10        $label_password = form_label('Password', 'password');
11        $input_email = form_input('email');
12        $input_password = form_password('password');
13        $submit = form_submit('login', 'submit');
14        $data = array(
15            'form_open' => $form_open,
16            'form_close' => $form_close,
17            'label_email' => $label_email,
18            'label_password' => $label_password,
19            'input_email' => $input_email,
20            'input_password' => $input_password,
21            'submit' => $submit,
22        );
23        $this->load->view('auth/login', $data);
24    }
25 }
```

Membuat View

Sebelum membuat file view buatlah folder terlebih dahulu dengan nama **auth** setelah itu buatlah view dengan nama **login_v.php**



```
$this->load->view('auth/login_v', $data);
```



```
1 <h3>Halaman Login</h3>
2
3 <?php echo $form_open ?>
4 <p>
5   <?php echo $label_email ?>
6   <?php echo $input_email ?>
7 </p>
8 <p>
9   <?php echo $label_password ?>
10  <?php echo $input_password ?>
11 </p>
12 <?php echo $submit ?>
13 <?php $form_close ?>
14
```

Routing

Konfigurasi `routes.php` di folder `Application/config/` seperti digambar,
kemudian tambahkan baris kode seperti digambar

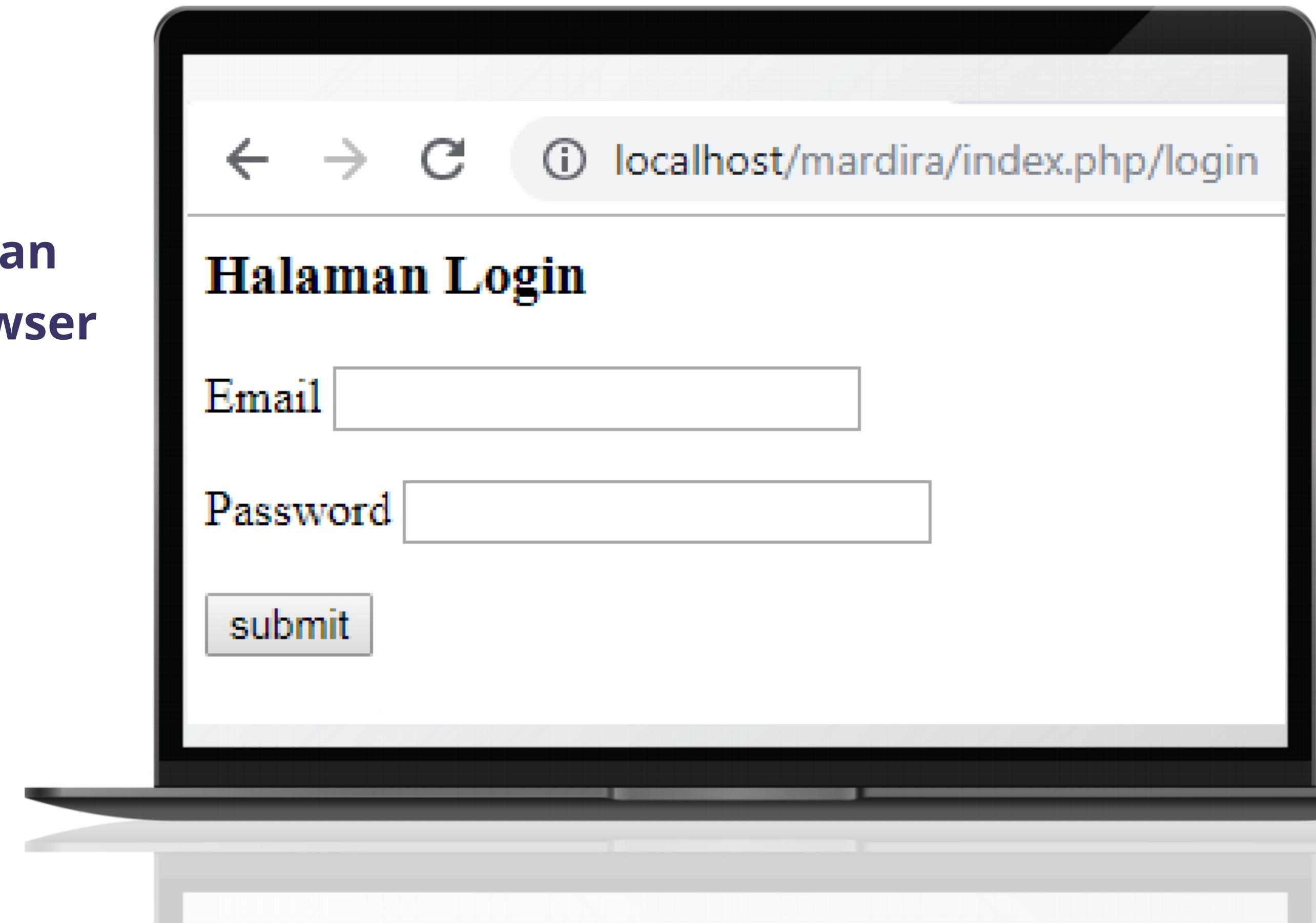
```
1 $route['default_controller'] = 'auth/login';
2 $route['404_override'] = '';
3 $route['translate_uri_dashes'] = FALSE;
4
5 $route['login'] = 'auth/login';
6 $route['logout'] = 'auth/logout';
```



Hasil Tampilan

Hasil tampilan kemudian
diakses melalui URL browser

localhost/mardira



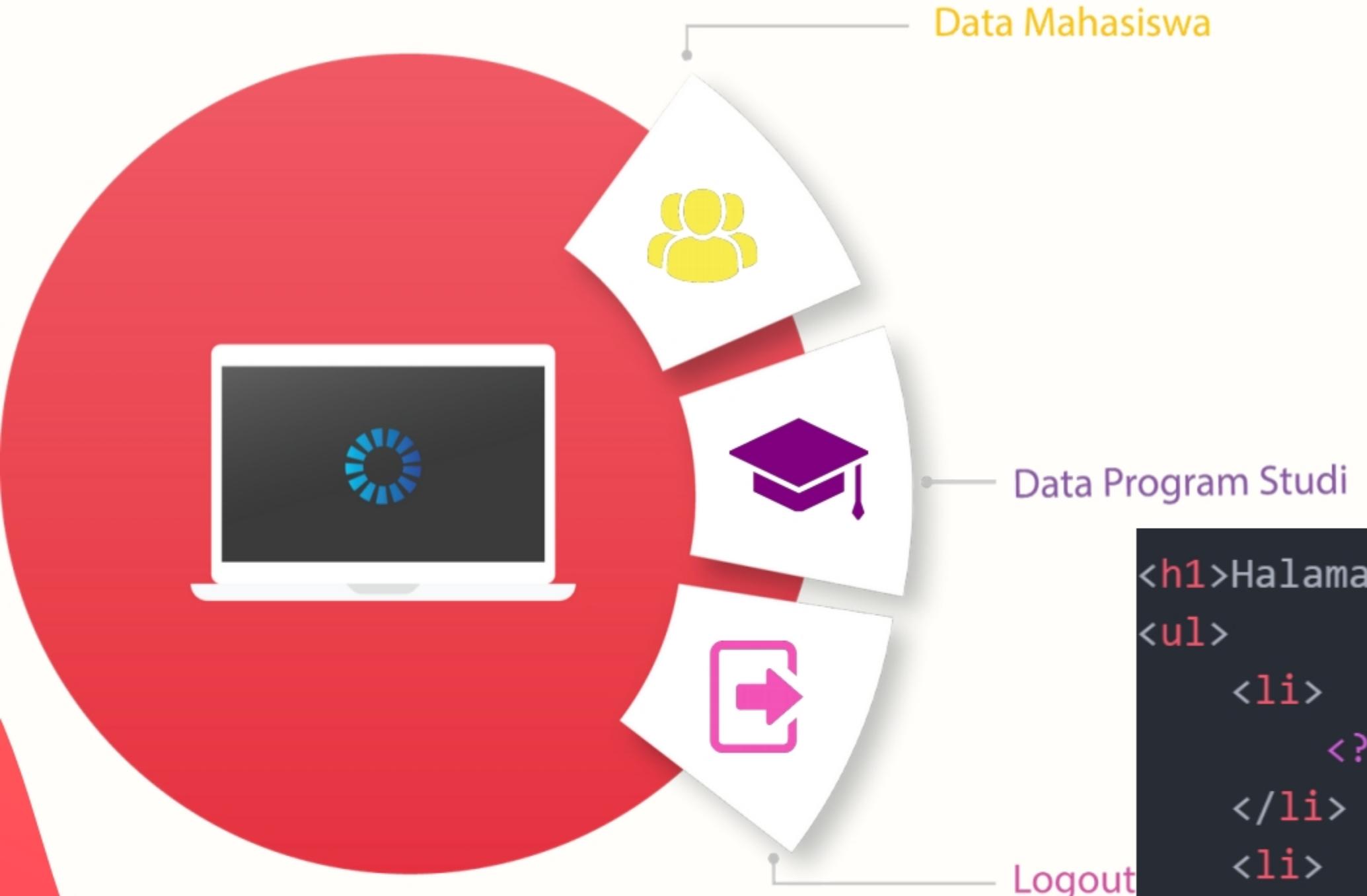
Membuat Menu Utama

Buatlah Controller **Home.php** dengan baris kode seperti di gambar



```
1 <?php
2 class Home extends CI_Controller
3 {
4     public function index()
5     {
6         $this->load->view('home/home_view');
7     }
8 }
```

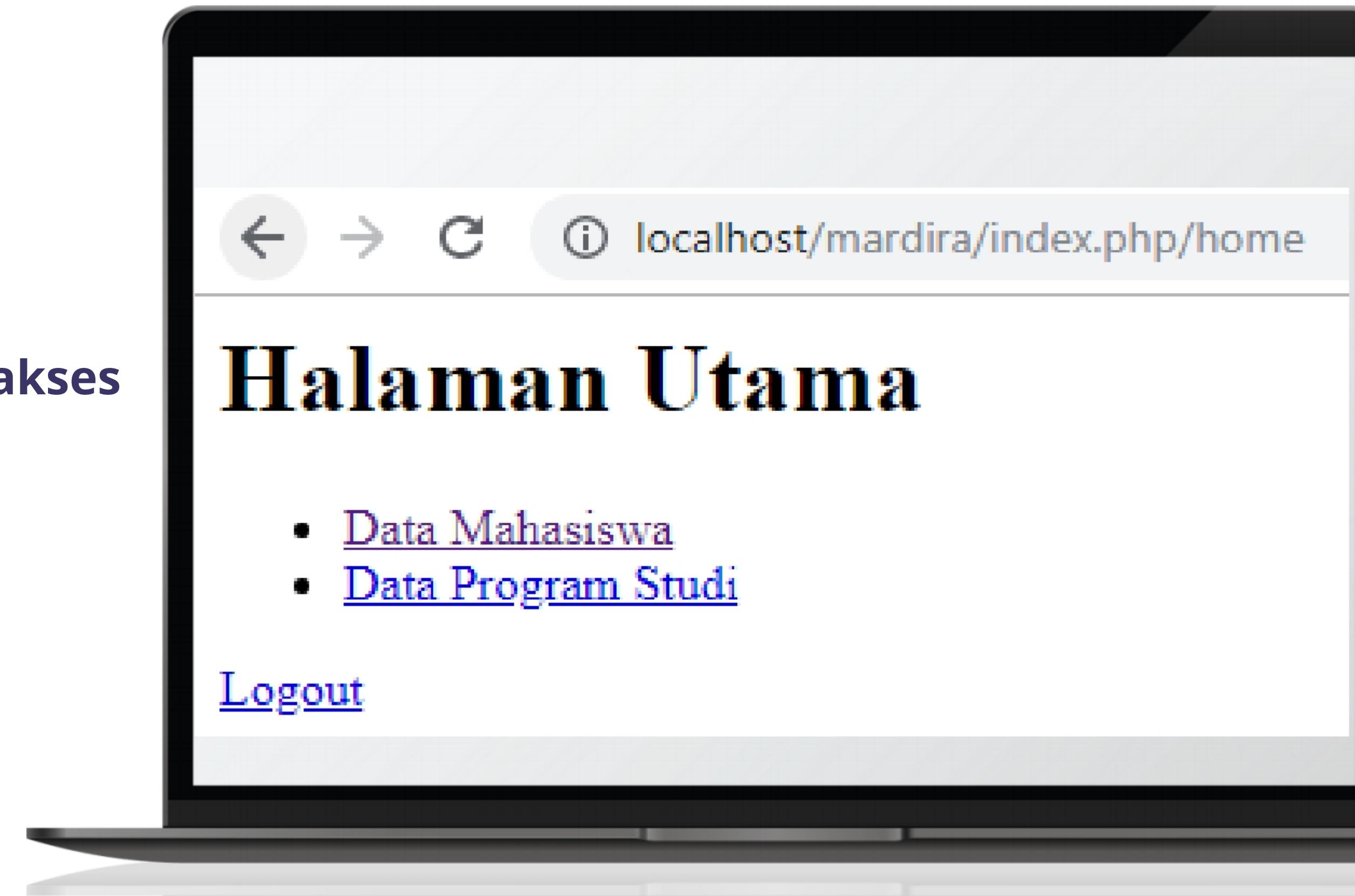
Buatlah View **home_view .php** di dalam folder **home** dengan baris kode seperti di gambar



```
<h1>Halaman Utama</h1>
<ul>
    <li>
        <?php echo anchor('mahasiswa', 'Data Mahasiswa') ?>
    </li>
    <li>
        <?php echo anchor('prodi', 'Data Program Studi') ?>
    </li>
</ul>
<?php echo anchor('logout', 'Logout') ?>
```

Hasil Tampilan

Hasil tampilan kemudian diakses
melalui URL browser

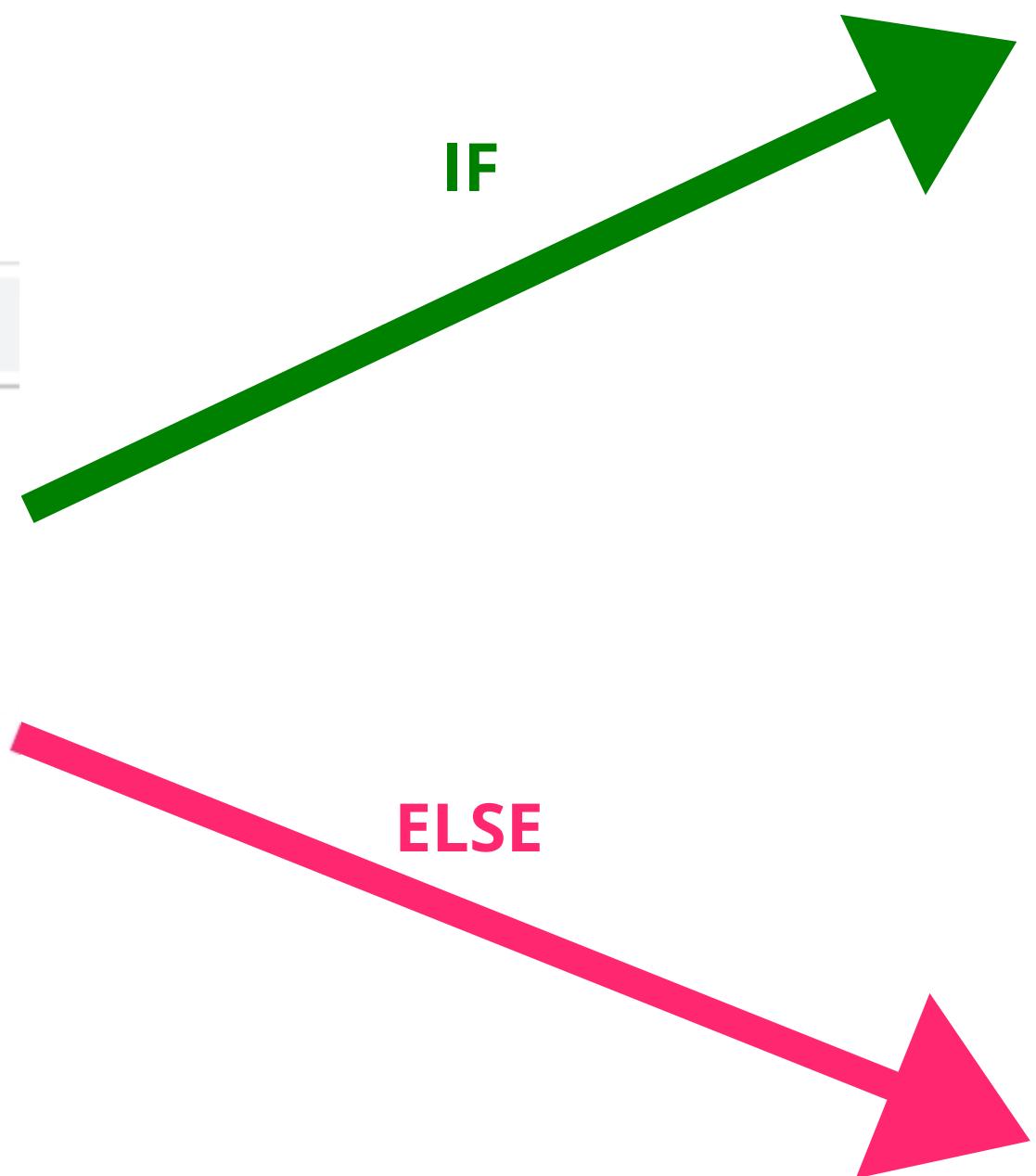


Konsep Login

Halaman Login

Email

Password



Username & Password Benar

localhost/mardira/index.php/home

Halaman Utama

- [Data Mahasiswa](#)
- [Data Program Studi](#)

[Logout](#)

Username & Password Salah

localhost/mardira/index.php/login

Halaman Login

Email

Password

Membuat Aksi Login

Di bagian Controller **Auth.php**
tambahkan baris kode seperti
di gambar.

```
1 class Auth extends CI_Controller
2 {
3
4     public function __construct()
5     {
6         parent::__construct();
7         $this->load->model('Auth_model');
8     }
9
10    public function login()
11    {
12        $form_open = form_open('auth/cek_login');
13        $form_close = form_close();
14        $label_email = form_label('Email', 'email');
15        $label_password = form_label('Password', 'password');
16        $input_email = form_input('email');
```

Membuat Model

Buatlah Model dengan nama **Auth_model.php**

```
1 <?php  
2  
3 class Auth_model extends CI_Model  
4 {  
5     public function cek_user($data)  
6     {  
7         return $this->db->get_where('users', $data);  
8     }  
9 }
```

```
public function cek_login()
{
    $email = $this->input->post('email');
    $password = $this->input->post('password');
    $data = array(
        'email' => $email,
        'password' => md5($password),
    );
    $cek_user = $this->Auth_model->cek_user($data);
    if ($cek_user->num_rows() > 0) {
        $row = $cek_user->row();
        $email = $row->email;
        $nama_lengkap = $row->nama_lengkap;
        $user = array(
            'email' => $email,
            'nama_lengkap' => $nama_lengkap
        );
        $this->session->set_userdata($user);
        redirect('home');
    } else {
        redirect('login');
    }
}
```

Membuat Session Userdata

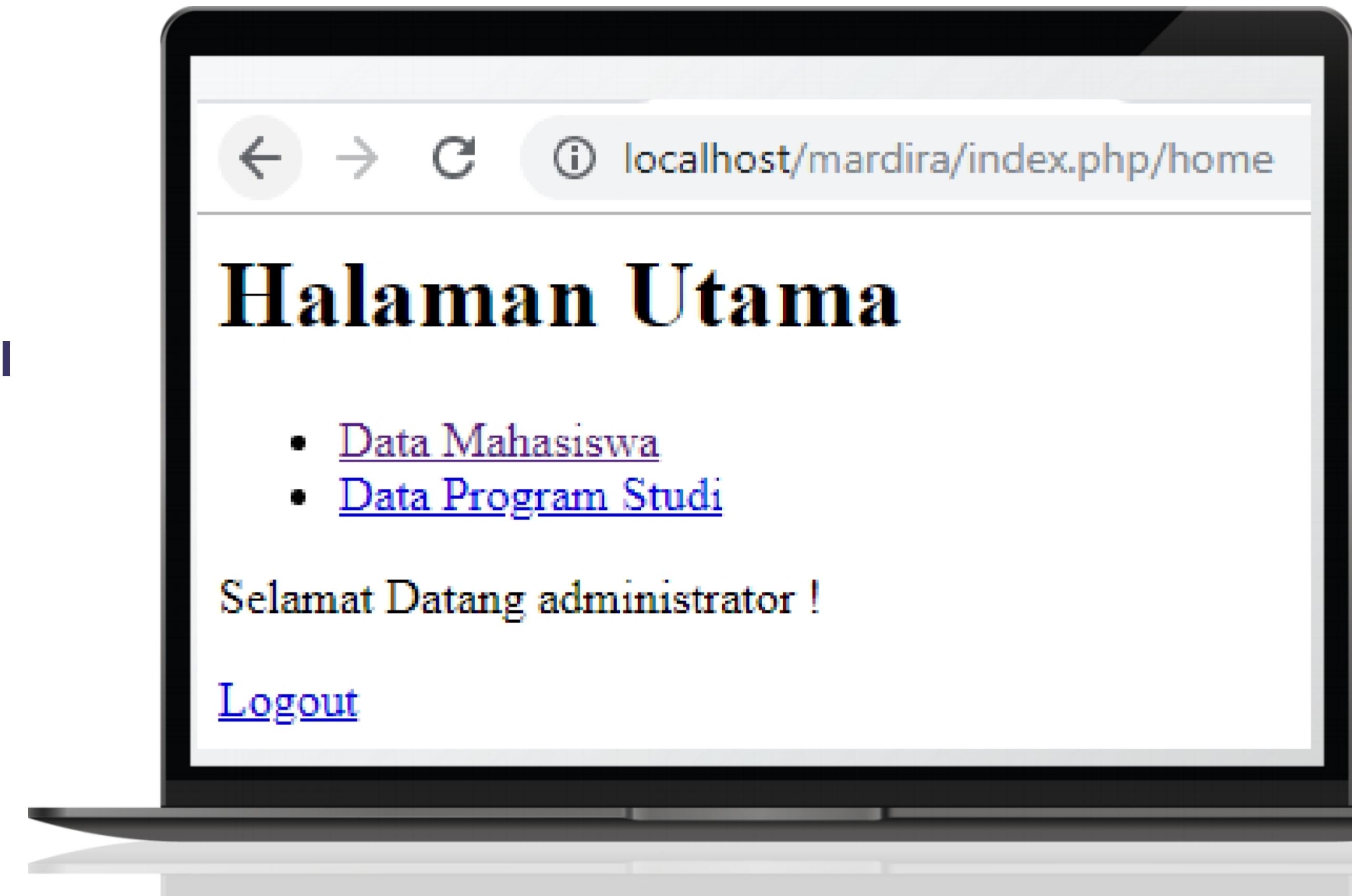
Di bagian **Auth.php** tambahkan fungsi seperti di gambar

Di bagian View **login_v.php** tambahkan baris kode seperti di gambar

```
1 <h1>Halaman Utama</h1>
2 <ul>
3   <li>
4     <?php echo anchor('mahasiswa', 'Data Mahasiswa') ?>
5   </li>
6   <li>
7     <?php echo anchor('prodi', 'Data Program Studi') ?>
8   </li>
9 </ul>
10 <p>
11   <?php
12   $nama_lengkap = $this->session->userdata('nama_lengkap');
13   echo "Selamat Datang {$nama_lengkap} !";
14   ?>
15 </p>
16
17 <?php echo anchor('logout', 'Logout') ?>
```

Hasil Tampilan

Hasil Jika login berhasil



Menghapus Session



Session bisa dihapus tanpa menghancurkan session dengan cara melakukan metode unset session

Di bagian Controller **Auth.php**
tambahkan fungsi seperti di gambar.

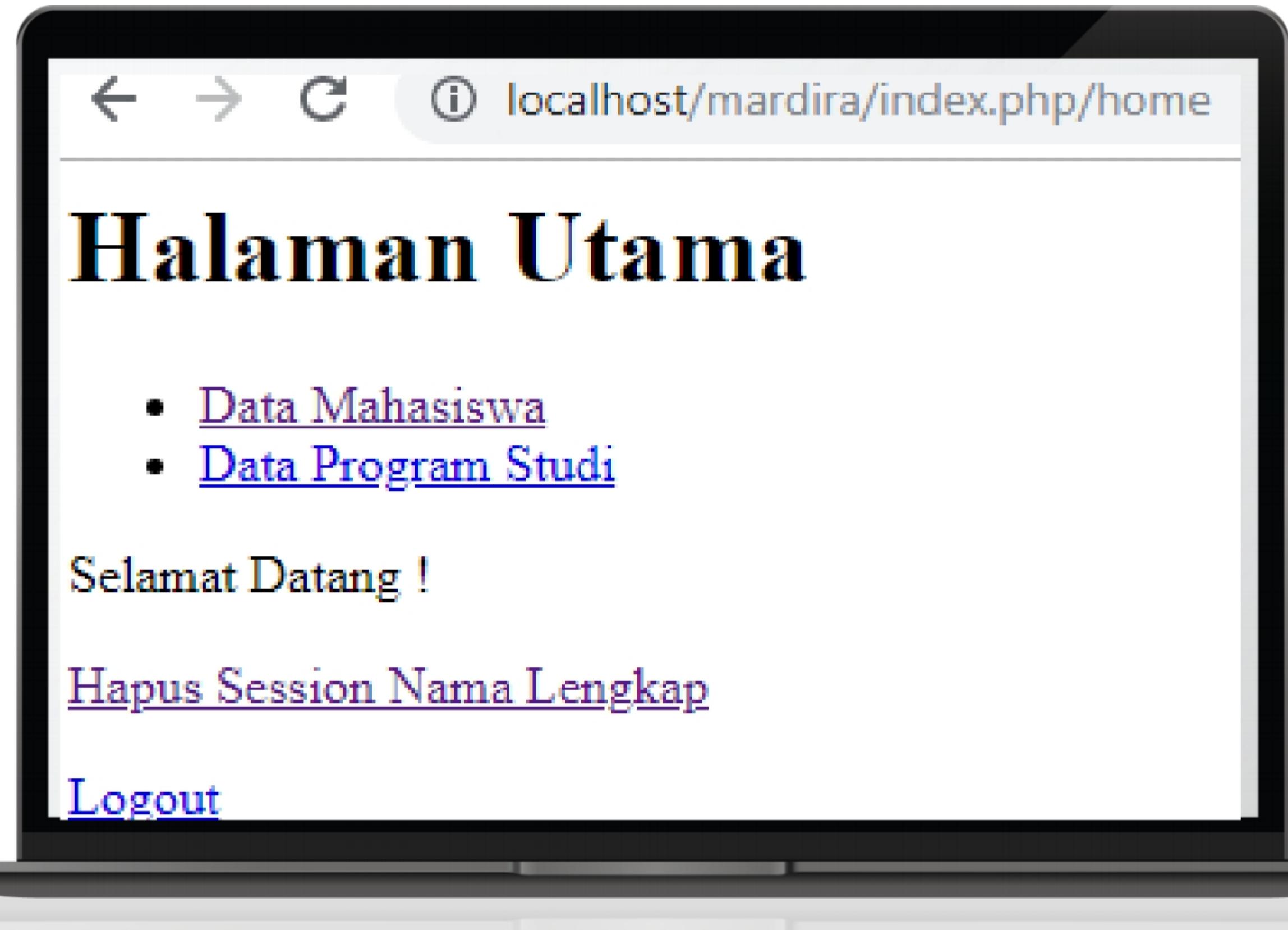
```
1 public function hapus_session()  
2 {  
3     $this->session->unset_userdata('nama_lengkap');  
4     redirect('home');  
5 }
```

Di bagian Controller **login_v.php**
tambahkan baris kode seperti di gambar.

```
1 <p>
2   <?php
3     $nama_lengkap = $this->session->userdata('nama_lengkap');
4     echo "Selamat Datang {$nama_lengkap} !";
5   ?>
6 </p>
7 <p>
8   <?php echo anchor('auth/hapus_session', 'Hapus Session Nama Lengkap') ?>
9 </p>
```

Hasil Tampilan

Hasil ketika session Nama Lengkap
dihapus



Menghancurkan Session



Menghancurkan session digunakan jika ingin melakukan aksi seperti logout

Di bagian Controller **Auth.php**
tambahkan fungsi seperti di gambar.

```
1 public function logout()  
2 {  
3     $this->session->sess_destroy();  
4     redirect('login');  
5 }
```

Membuat Helper



```
1 <?php  
2  
3 function cek_session()  
4 {  
5     $CI = &get_instance();  
6     $session = $CI->session->userdata('email');  
7     if ($session == NULL) {  
8         redirect('login');  
9     }  
10 }
```

Untuk memastikan sudah login atau belumnya, dibantu dengan login Helper.
Buatlah dengan nama Login_helper.php di folder application/helpers/

Di bagian Controller **Home.php**
tambahkan baris kode seperti di gambar.

```
1 <?php
2 class Home extends CI_Controller
3 {
4
5     public function __construct()
6     {
7         parent::__construct();
8         $this->load->helper('Login');
9         cek_session();
10    }
11
12    public function index()
13    {
14        $this->load->view('home/home_view');
15    }
16 }
```

Implementasi Session Flashdata



**Penggunaan session flashdata bisa
disebut sebagai pesan validasi ketika
telah melakukan sebuah aksi**

```
public function cek_login()
{
    $email = $this->input->post('email');
    $password = $this->input->post('password');
    $data = array(
        'email' => $email,
        'password' => md5($password),
    );
    $cek_user = $this->Auth_model->cek_user($data);
    if ($cek_user->num_rows() > 0) {
        $row = $cek_user->row();
        $email = $row->email;
        $nama_lengkap = $row->nama_lengkap;
        $user = array(
            'email' => $email,
            'nama_lengkap' => $nama_lengkap
        );
        $this->session->set_userdata($user);
        redirect('home');
    } else {
        $this->session->set_flashdata('pesan', 'username atau password salah!');
        redirect('login');
    }
}
```

Di bagian function `cek_login` di controller `Auth.php`, tambahkan baris kode seperti di gambar

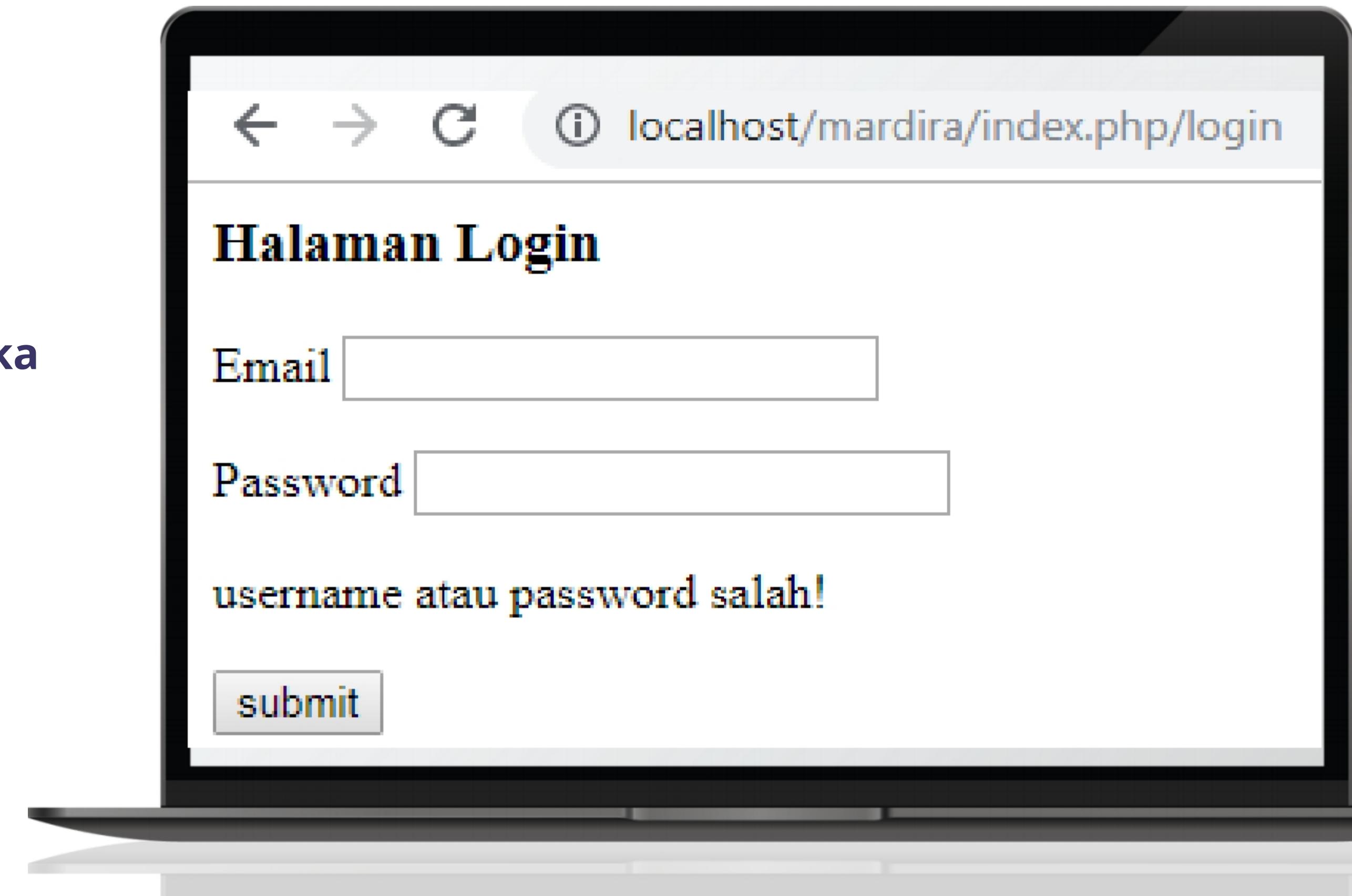


```
1 <h3>Halaman Login</h3>
2
3 <?php echo $form_open ?>
4 <p>
5     <?php echo $label_email ?>
6     <?php echo $input_email ?>
7 </p>
8 <p>
9     <?php echo $label_password ?>
10    <?php echo $input_password ?>
11 </p>
12
13 <p>
14     <?php
15     $pesan = $this->session->flashdata('pesan');
16     echo $pesan;
17     ?>
18 </p>
19 <?php echo $submit ?>
20 <?php $form_close ?>
```

Di bagian **login_v.php**
tambahkan baris kode seperti
digambar

Hasil Tampilan

Hasil session flashdata ketika
login gagal



Kegunaan Lain Session Flashdata



Session flashdata juga bisa
digunakan untuk validasi ketika data
berhasil ditambahkan, diubah dan
dihapus, pengembangan sistem
silahkan mencoba



Next
#7 Form Validation

Referensi

- <http://onestringlab.com/mengelola-session-pada-codeigniter/>
- [https://codeigniter.com/user_guide/libraries/sessions.html?
highlight=session#CI_Session](https://codeigniter.com/user_guide/libraries/sessions.html?highlight=session#CI_Session)