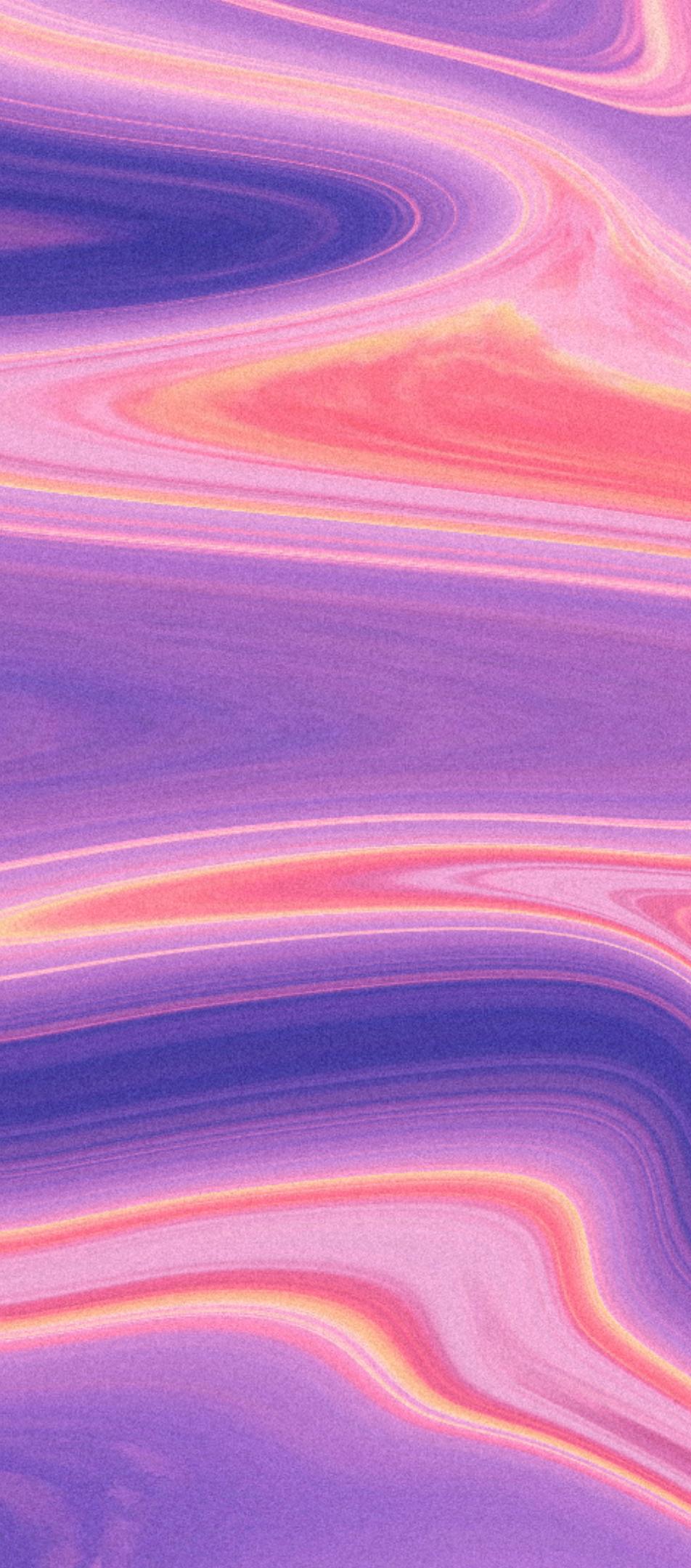


JEFRI MARULI H

MODUL CODEIGNITER PART III





URL ROUTING (ROUTE)

Adalah salah satu metode yang digunakan untuk memetakan URL ke dalam sumber daya tertentu dengan memberikan nama lain dari alamat sumber daya yang dimaksud

KEGUNAAN ROUTING

L

URL MENJADI MUDAH DIBACA

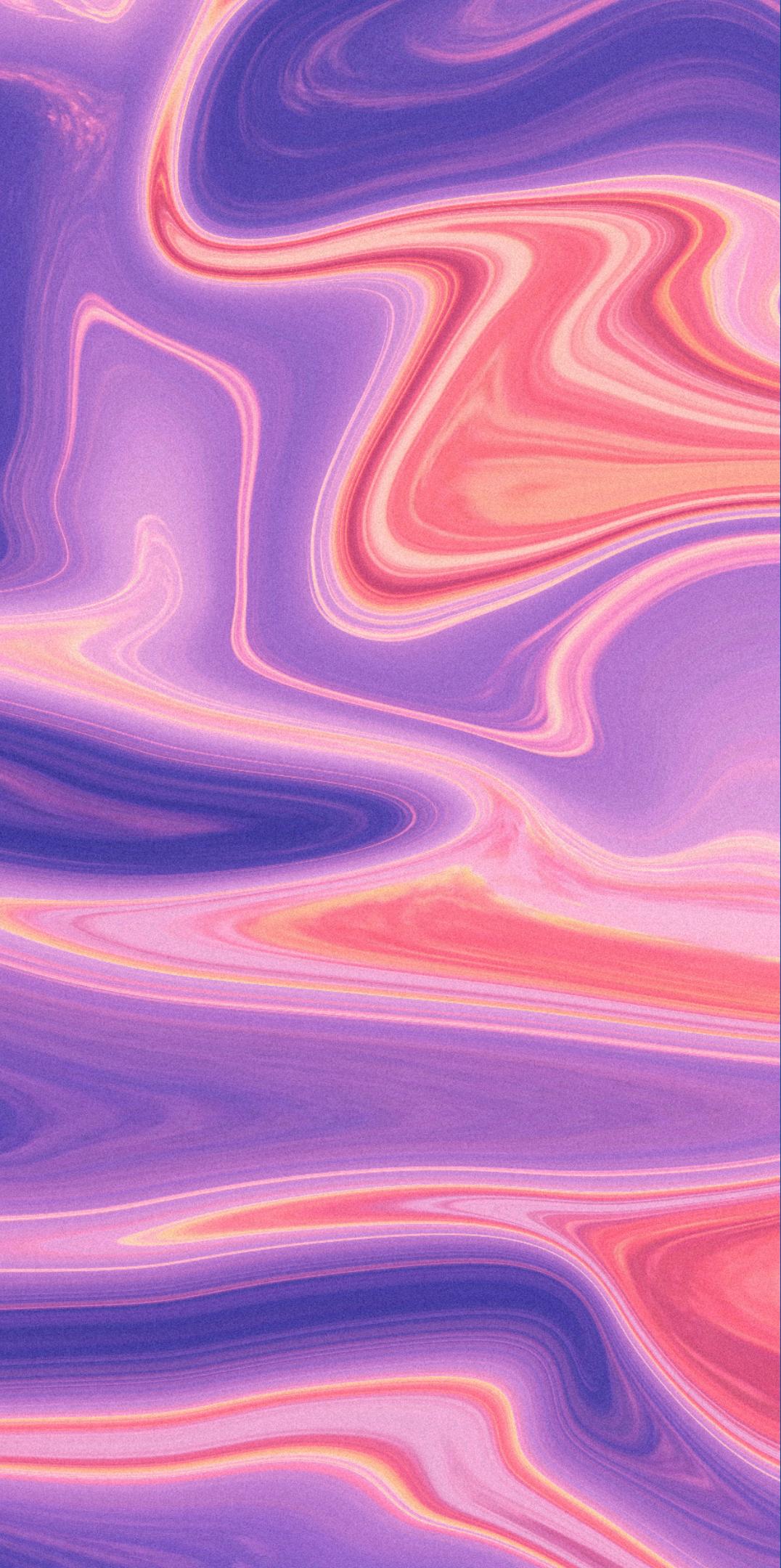
Menjadikan URL sumber daya yang sulit dibaca manusia dengan membuat pemetaan baru ke URL alias dari route yang lebih mudah dibaca manusia.

URL MENJADI LEBIH PENDEK

Membuat URL sumber daya menjadi lebih pendek dengan memberikan penamaan routing yang lebih pendek.

URL SESUAI FORMAT YANG DIINGINKAN

Memantau agar URL sesuai dengan format yang diinginkan dengan memanfaatkan fungsi regular expression (regex)



PENGGUNAAN ROUTE



```
1 $route['default_controller'] = 'welcome';
2 $route['404_override'] = '';
3 $route['translate_uri_dashes'] = FALSE;
4
```

Digunakan untuk melakukan otomatis replace karakter dashes (-) dengan underscores (_) pada controller dan uri segment, fungsi ini dapat diaktifkan dengan mengganti FALSE menjadi TRUE jika membutuhkan fungsi tersebut.

Digunakan untuk mengatur Controller mana yang akan digunakan atau diload pertama kali sebagai default controller.

Digunakan untuk mengatur tampilan jika halaman ini tidak ditemukan (error404) dan secara standar bagian ini akan dikosongkan.

File routes.php terdapat di folder Application/config/routes.php



IMPLEMENTASI ROUTING

HAL YANG DIPERSIAPKAN

- Konfigurasi autoload.php
- Konfigurasi database.php dan perancangan database
- Konfigurasi Route
- Pembuatan Model View Controller (MVC)

KONFIGURASI AUTOLOAD

file autoload.php terdapat di folder application/config/autoload.php

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

kemudian bagian
\$autoload['libraries'] = array();
isi dengan 'database' seperti di
gambar.

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

dan bagian
\$autoload['libraries'] = array();
isi dengan 'url' seperti di gambar.

KONFIGURASI DATABASE

file `database.php` terdapat di folder `application/config/database.php`

di bagian `'hostname'`, `'username'`, dan `'database'` isi data konfigurasi sesuai dengan di gambar

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

PERANCANGAN DATABASE

MEMBUAT DATABASE MARDIRA

```
CREATE DATABASE mardira;
```

MEMBUAT 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));
```

	#	Name	Type
	1	<code>id_mahasiswa</code> 	<code>int(3)</code>
	2	<code>nim</code>	<code>int(10)</code>
	3	<code>nama_mhs</code>	<code>varchar(30)</code>
	4	<code>id_prodi</code> 	<code>int(3)</code>

MEMBUAT TABLE PRODI

```
CREATE TABLE prodi  
(id_prodi INT(3) PRIMARY KEY NOT NULL  
AUTO_INCREMENT, nama_prodi VARCHAR(25)  
NOT NULL);
```

#	Name	Type
1	<code>id_prodi</code> 	<code>int(3)</code>
2	<code>nama_prodi</code>	<code>varchar(25)</code>

INSERT TABLE PRODI

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

	<code>id_prodi</code>	<code>nama_prodi</code>
1	1	Teknik Informatika
2	2	Manajemen Informatika

INSERT TABLE MAHASISWA

```
INSERT INTO mahasiswa (id_mahasiswa,nim,nama_mhs,id_prodi) VALUES  
(NULL,'171234567','nama kalian','1'),  
(NULL,'1711234568','nama kalian','1'),  
(NULL,'1711234569','nama kalian','2');
```

id_mahasiswa	nim	nama_mhs	id_prodi
1	18130146	Asep Mulyono	1
2	18310415	Dani Septiono	1
3	18112342	Haryono Hasan	2

KONFIGURASI ROUTE



```
1 $route['default_controller'] = 'halaman_utama';
2 $route['404_override'] = '';
3 $route['translate_uri_dashes'] = FALSE;
4
5 $route['home'] = 'halaman_utama';
6 $route['mhs'] = 'mahasiswa';
7 $route['prodi'] = 'program_studi';
```

Pada file routes.php
tambahkan konfigurasi
route seperti di gambar ini

PEMBUATAN MVC

Membuat Controller

Setelah konfigurasi routes
kemudian buat Controller
dengan nama
"Halaman_utama.php" ,
dan ikuti script di gambar

```
1 <?php  
2  
3 class Halaman_utama extends CI_Controller  
4 {  
5  
6     public function index()  
7     {  
8         $this->load->view('halaman_utama_v');  
9  
10    }  
11 }
```

Membuat View

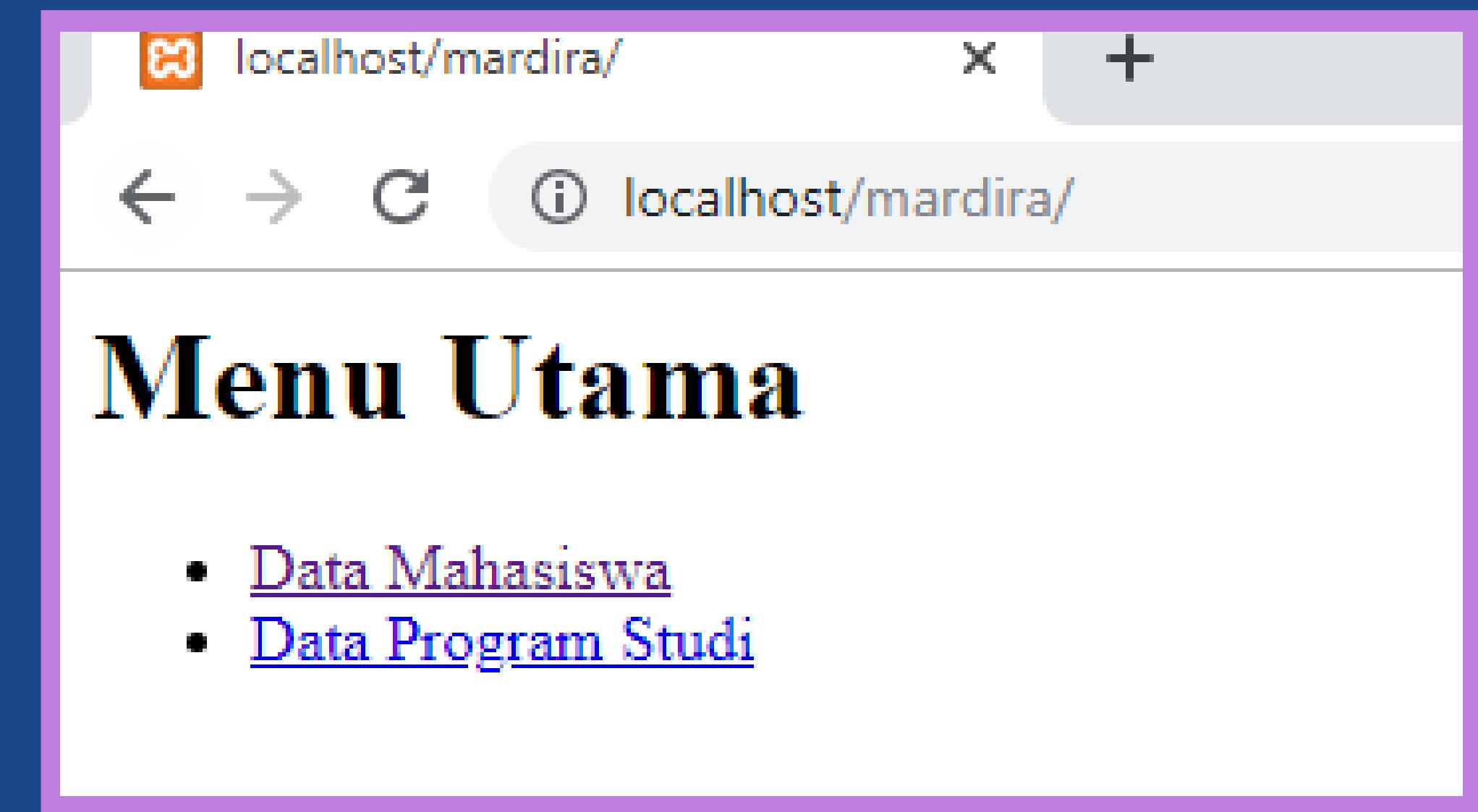
Langkah selanjut yaitu membuat view dengan nama "Halaman_utama_v.php" dan ikuti script seperti di gambar.



```
1 <h1>Menu Utama</h1>
2 <ul>
3   <li><?php echo anchor('mhs', 'Data Mahasiswa') ?></li>
4   <li><?php echo anchor('prodi', 'Data Program Studi') ?></li>
5 </ul>
```

Hasil

Kemudian jalankan melalui web browser ,
lalu ketik di URL "localhost/mardira"



PEMBUATAN MVC MAHASISWA



```
1 <?php
2
3 class Mahasiswa extends CI_Controller
4 {
5     public function __construct()
6     {
7         parent::__construct();
8         $this->load->model('Mahasiswa_model');
9     }
10
11    public function index()
12    {
13        $data['mahasiswa'] = $this->Mahasiswa_model->lihat_data();
14        $this->load->view('mahasiswa_v', $data);
15    }
16 }
```

Membuat Controller

Setelah membuat halaman utama, kemudian buatlah Controller dengan nama Mahasiswa.php

Membuat Model

Langkah selanjut yaitu membuat Model dengan nama "Mahasiswa_model.php" dan ikuti script seperti di gambar.



```
1 <?php
2 class Mahasiswa_model extends CI_Model
3 {
4     public function lihat_data()
5     {
6         return $this->db->join('prodi', 'prodi.id_prodi = mahasiswa.id_prodi', 'left')
7             ->get('mahasiswa')->result();
8     }
9 }
```



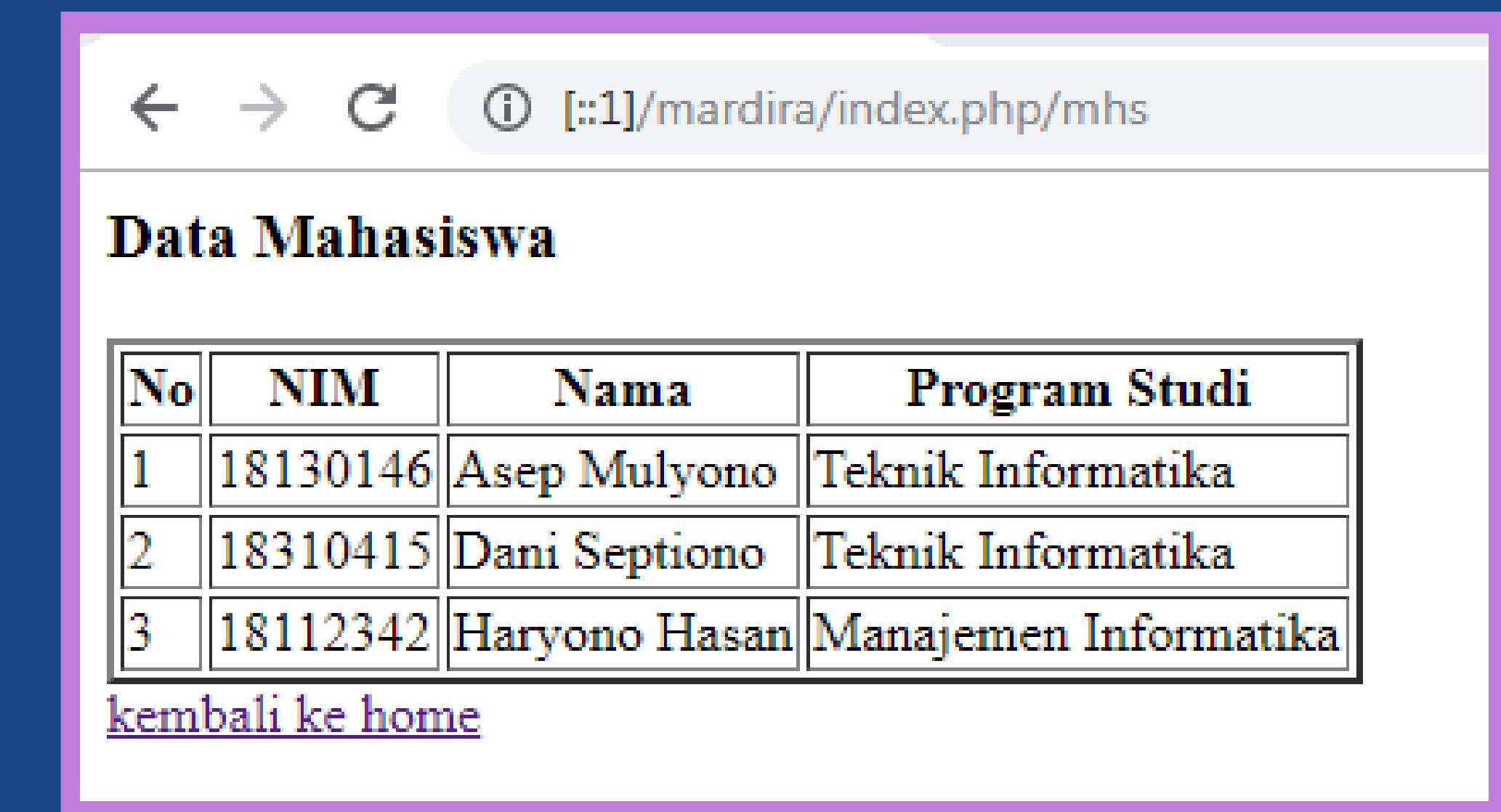
Membuat View

Lalu membuat view dengan nama mahasiswa_v.php

```
1 <h3>Data Mahasiswa</h3>
2 <table border="2">
3   <thead>
4     <tr>
5       <th>No</th>
6       <th>NIM</th>
7       <th>Nama</th>
8       <th>Program Studi</th>
9     </tr>
10    </thead>
11    <tbody>
12      <?php
13      $no = 1;
14      foreach ($mahasiswa as $row) { ?>
15        <tr>
16          <td><?php echo $no++ ?></td>
17          <td><?php echo $row->nim ?></td>
18          <td><?php echo $row->nama_mhs ?></td>
19          <td><?php echo $row->nama_prodi ?></td>
20        </tr>
21      <?php } ?>
22    </tbody>
23  </table>
24  <?php echo anchor('home', 'kembali ke home') ?>
```

Hasil MVC Mahasiswa

Kemudian jalankan melalui web browser ,
lalu ketik di URL
"localhost/mardira/index.php/mhs"



The screenshot shows a web browser window with a title bar containing navigation icons and the URL [\[:1\]/mardira/index.php/mhs](http://[:1]/mardira/index.php/mhs). The main content area displays a table titled "Data Mahasiswa". The table has four columns: No, NIM, Nama, and Program Studi. It contains three rows of data. Below the table is a link labeled "kembali ke home".

No	NIM	Nama	Program Studi
1	18130146	Asep Mulyono	Teknik Informatika
2	18310415	Dani Septiono	Teknik Informatika
3	18112342	Haryono Hasan	Manajemen Informatika

[kembali ke home](#)

PEMBUATAN MVC PROGRAM STUDI

```
1 <?php  
2  
3 class Program_studi extends CI_Controller  
4 {  
5     public function __construct()  
6     {  
7         parent::__construct();  
8         $this->load->model('Prodi_model');  
9     }  
10  
11    public function index()  
12    {  
13        $data['prodi'] = $this->Prodi_model->lihat_data();  
14        $this->load->view('prodi_v', $data);  
15    }  
16 }
```

Membuat Controller
kemudian buatlah
Controller dengan nama
Program_studi.php

Membuat Model

Lalu membuat Model dengan nama "Prodi_model.php"

```
1 <?php  
2  
3 class Prodi_model extends CI_Model  
4 {  
5     public function lihat_data()  
6     {  
7         return $this->db->get('prodi')->result();  
8     }  
9 }  
10
```

Membuat View

Langkah terakhir yaitu membuat view dengan nama prodi_v.php

```
<h3>Data Program Studi</h3>
<table border="2">
    <thead>
        <tr>
            <th>No</th>
            <th>Nama Prodi</th>
        </tr>
    </thead>
    <tbody>
        <?php
        $no = 1;
        foreach ($prodi as $row) { ?>
            <tr>
                <td><?php echo $no++ ?></td>
                <td><?php echo $row->nama_prodi ?></td>
            </tr>
        <?php } ?>
    </tbody>
</table>
<?php echo anchor('home', 'kembali ke home') ?>
```

Hasil MVC Program Studi

Kemudian jalankan melalui web browser ,
lalu ketik di URL
"localhost/mardira/index.php/prodi"

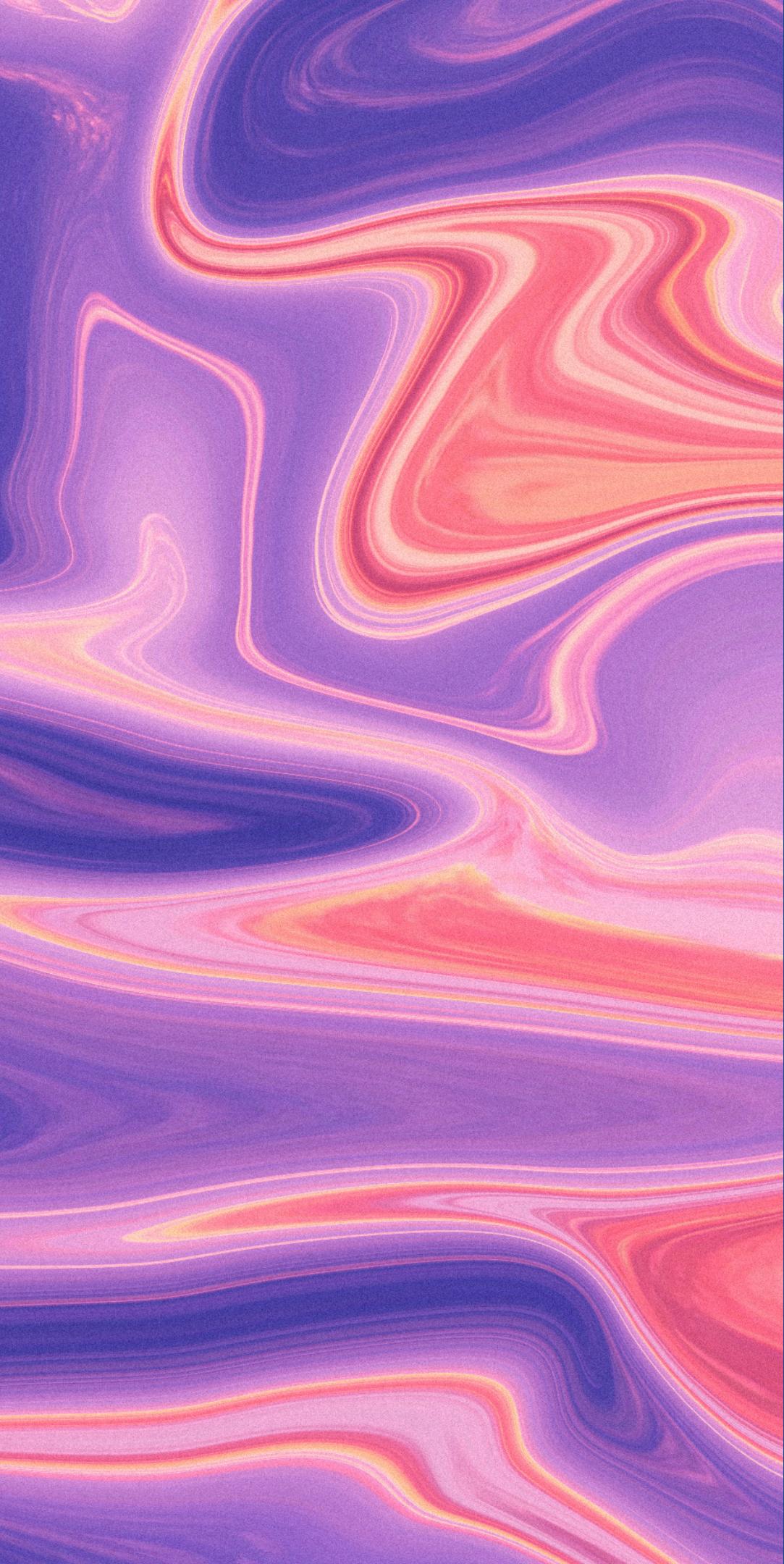
The screenshot shows a web browser window with the following details:

- Address Bar:** Shows the URL `[::1]/mardira/index.php/prodi`.
- Title Bar:** Displays the title "Data Program Studi".
- Table:** A data table with two columns: "No" and "Nama Prodi". It contains two rows of data:
 - Row 1: No 1, Nama Prodi Teknik Informatika
 - Row 2: No 2, Nama Prodi Manajemen Informatika
- Text:** Below the table is a link "kembali ke home".

MENGHILANGKAN INDEX PAGE

Sering kali melihat tulisan index.php di URL setiap pemanggilan hasil pembuatan aplikasi berbasis MVC, disini akan membahas bagaimana cara menghilangkan index.php tsb.

localhost/mardira/index.php/mahasiswa



KONFIGURASI INDEX



```
1 $config['index_page'] = 'index.php';
```

menjadi



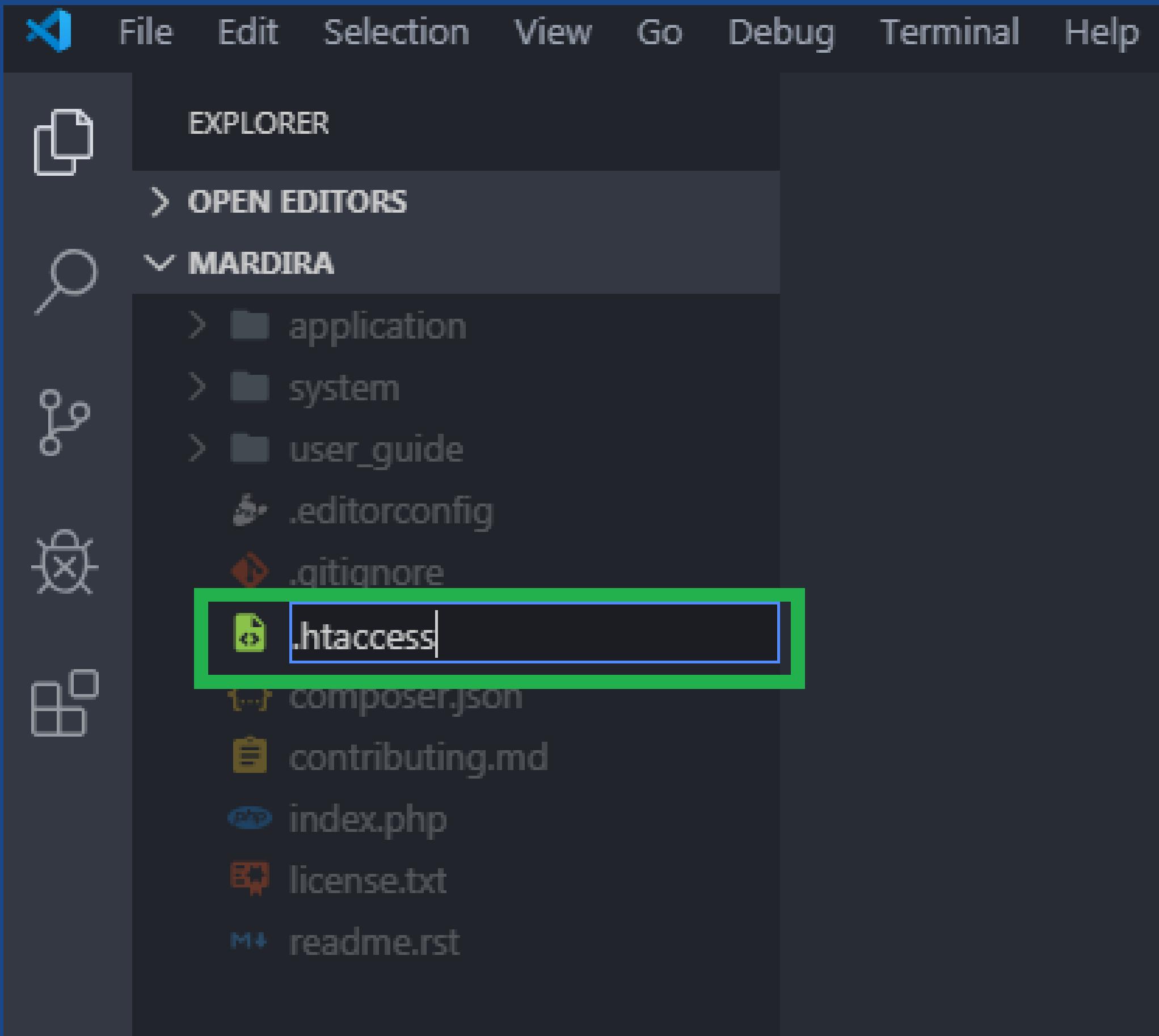
```
1 $config['index_page'] = '';
```

file config.php terdapat di folder
application/config/config.php

kemudian bagian
\$config['index_page'] = 'index.php';
hapus bagian 'index.php' atau
kosongkan seperti di gambar.

Membuat .htaccess

htaccess harus dibuat terlebih dahulu jika ingin menghilangkan index.php pada URL yaitu membuat file baru dengan nama .htaccess (menggunakan titik didepannya) di Folder paling luar yaitu root folder Mardira



Pada file .htaccess

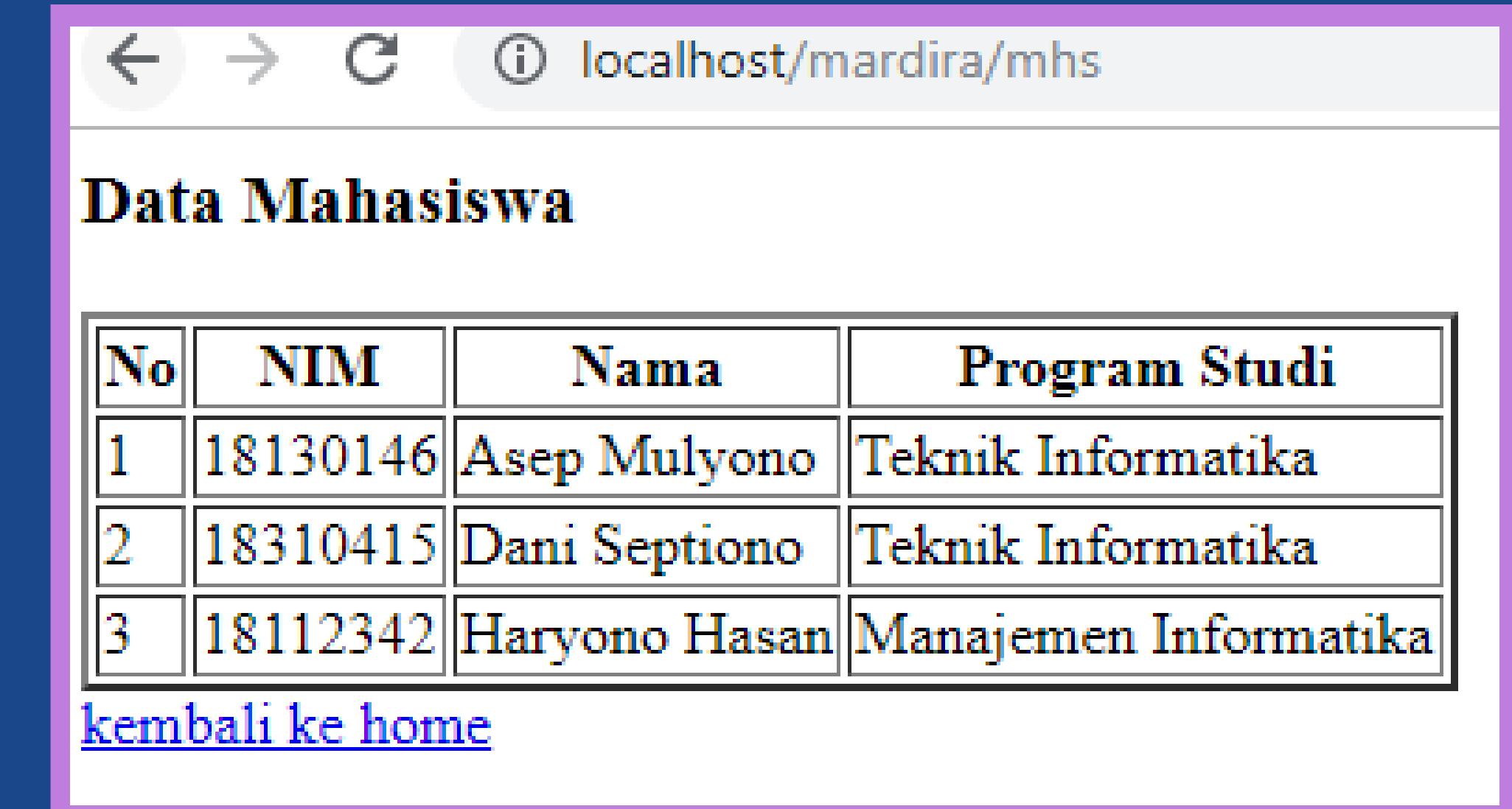
Kemudian ikutilah script di bawah ini



```
1 RewriteEngine On  
2 RewriteCond %{REQUEST_FILENAME} !-f  
3 RewriteCond %{REQUEST_FILENAME} !-d  
4 RewriteRule ^(.*)$ index.php/$1 [L]
```

Hasil

Setelah itu ketik URL tanpa menggunakan index.php seperti di gambar



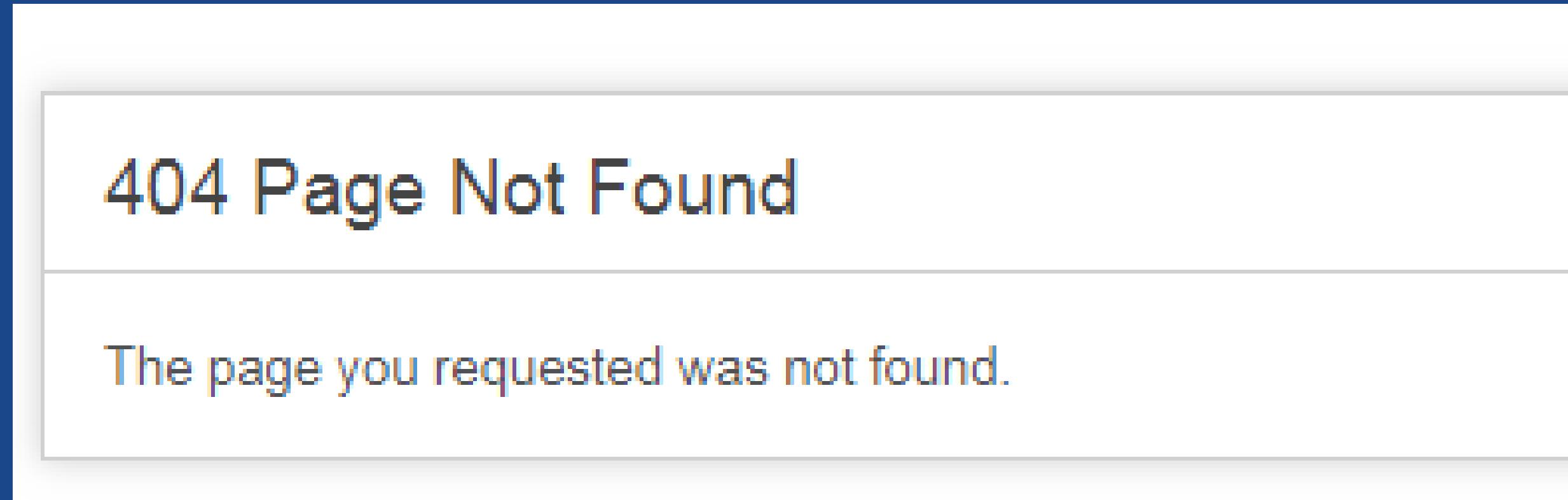
A screenshot of a web browser window titled "localhost/mardira/mhs". The main content is a table titled "Data Mahasiswa" with four columns: No, NIM, Nama, and Program Studi. The table contains three rows of data. Below the table is a blue link labeled "kembali ke home".

No	NIM	Nama	Program Studi
1	18130146	Asep Mulyono	Teknik Informatika
2	18310415	Dani Septiono	Teknik Informatika
3	18112342	Haryono Hasan	Manajemen Informatika

[kembali ke home](#)

MENGKUSTOMISASI ERROR 404 NOT FOUND PADA CI

Buat yang sering melihat halaman error bertulisan 404 Page Not Found, ingin merubah tampilan error tsb sesuai dengan keinginan, disini akan dibahas cara merubah atau mengkustom tampilan Error Not Found 404



KONFIGURASI ROUTE



```
1 $route['default_controller'] = 'halaman_utama';
2 $route['404_override'] = 'myerror';
3 $route['translate_uri_dashes'] = FALSE;
4
```

Kembali lagi ke file routes.php di bagian
\$route['404_override'] = '' ;
isi dengan myerror seperti
di gambar

Membuat Controller

Setelah konfigurasi file routes.php kemudian buat Controller dengan nama Myerror.php

```
1 <?php  
2  
3 class Myerror extends CI_Controller  
4 {  
5     public function index()  
6     {  
7         $this->load->view('myerror_v');  
8     }  
9 }
```

Membuat View

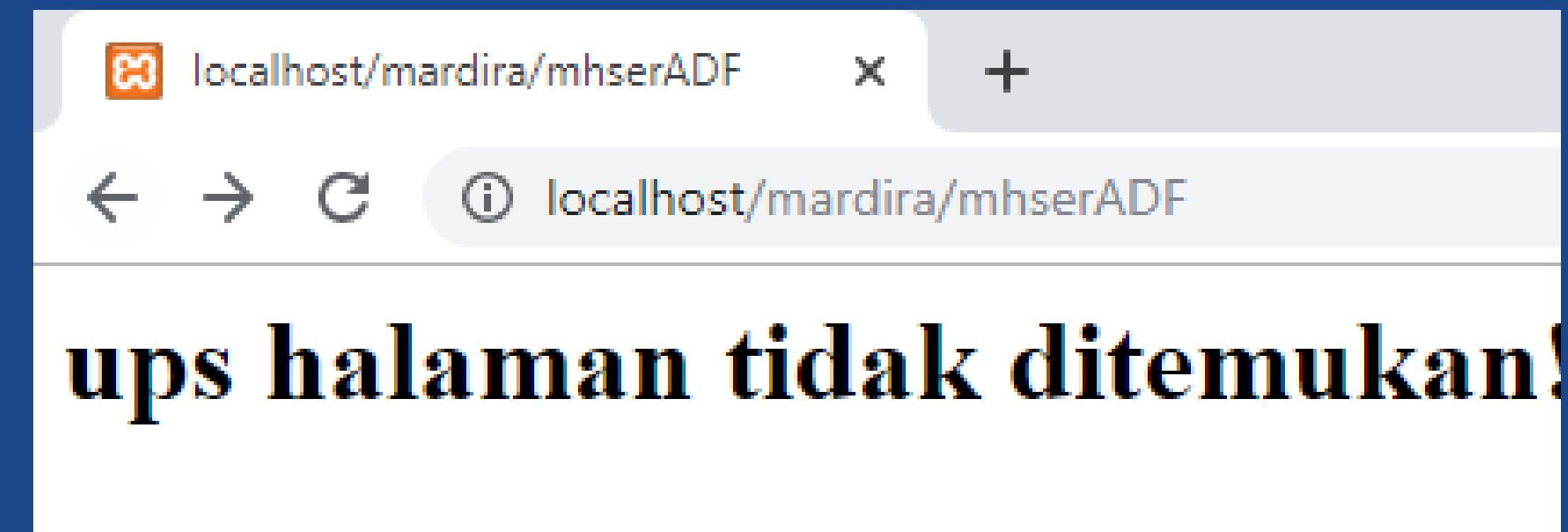
Kemudian buatlah view dengan nama `myerror_v.php`



```
1 <h1>ups halaman tidak ditemukan!</h1>
```

Hasil

Kemudian tampilan error 404 Not Found seperti di gambar, tampilan error bisa dirias lebih menarik lagi sesuai dengan keinginan masing-masing.



TERIMA KASIH