

Routing

Konsep dasar CI yang harus dipahami

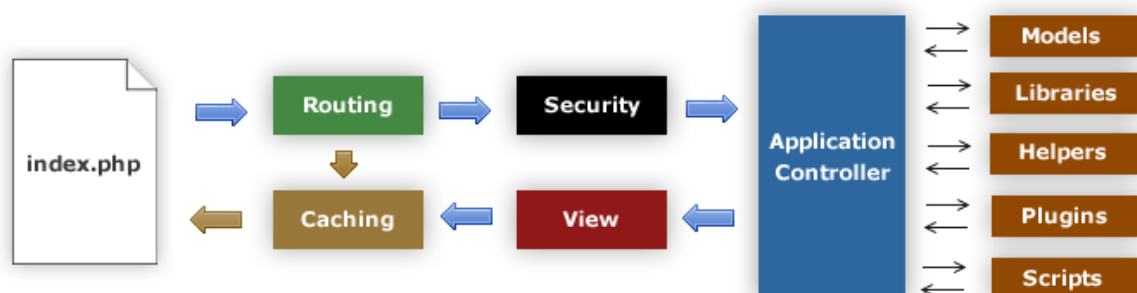
Konsep MVC Pada Codeigniter

MVC (*Model, View, Controller*) adalah sebuah pola desain (*design pattern*) arsitektur pengembangan aplikasi yang memisahkan dan mengelompokkan beberapa kode sesuai dengan fungsinya.

MVC membagi aplikasi ke dalam tiga bagian fungsional: model, view, dan controller.

- **Model** adalah kode-kode untuk model bisnis dan data. biasanya berhubungan langsung dengan database untuk memanipulasi data (insert, update, delete, search), menangani validasi dari bagian controller, namun tidak dapat berhubungan langsung dengan bagian view.
- **View** merupakan bagian yang menangani *presentation logic*. berisi kode-kode untuk tampilan.
- **Controller** merupakan bagian yang mengatur hubungan antara bagian *model* dan bagian *view*, controller berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.

Alur kerja Codeigniter yang berpola MVC ini seperti ini:



1. User mengirim *request* ke web;
2. File yang pertama kali dieksekusi adalah **index.php**;
3. Lalu dari **index.php**, *request* akan diteruskan oleh **routers.php**;
4. **routers.php** akan mencari *cache* di server, jika ada maka *cache* itu yang akan dikirim sebagai balasan (*response*), Jika tidak ada *cache* baru *request* diteruskan ke *Controller*;
5. *Controller* akan bertanggung jawab untuk mengambil data dari *Model* dan merendernya ke dalam *View* dengan menggunakan *library*, *plugin*, dan *helper* yang ada;
6. Hasil *render (view)* dikirim ke pengguna dan disimpan dalam *cache*, apabila fitur *cache* aktif;

Router Pada Codeigniter

Router pada Codeigniter bertugas untuk menentukan controller dan method/fungsi yang akan dieksekusi. Ketika membuka aplikasi di <http://localhost/tokobuah/> maka fungsi yang akan dieksekusi adalah fungsi *index()* yang berada di dalam *Controller Welcome (controller/Welcome.php)*.

```

3
4 class Welcome extends CI_Controller {
5
6     /**
7      * Index Page for this controller.
8      *
9      * Maps to the following URL
10     *     http://example.com/index.php/welcome
11     * - or -
12     *     http://example.com/index.php/welcome/index
13     * - or -
14     * Since this controller is set as the default controller in
15     * config/routes.php, it's displayed at http://example.com/
16     *
17     * So any other public methods not prefixed with an underscore w
18     * map to /index.php/welcome/<method_name>
19     * @see https://codeigniter.com/user\_guide/general/urls.html
20     */
21     public function index()
22     {
23         $this->load->view('welcome_message');
24     }
25 }
26

```

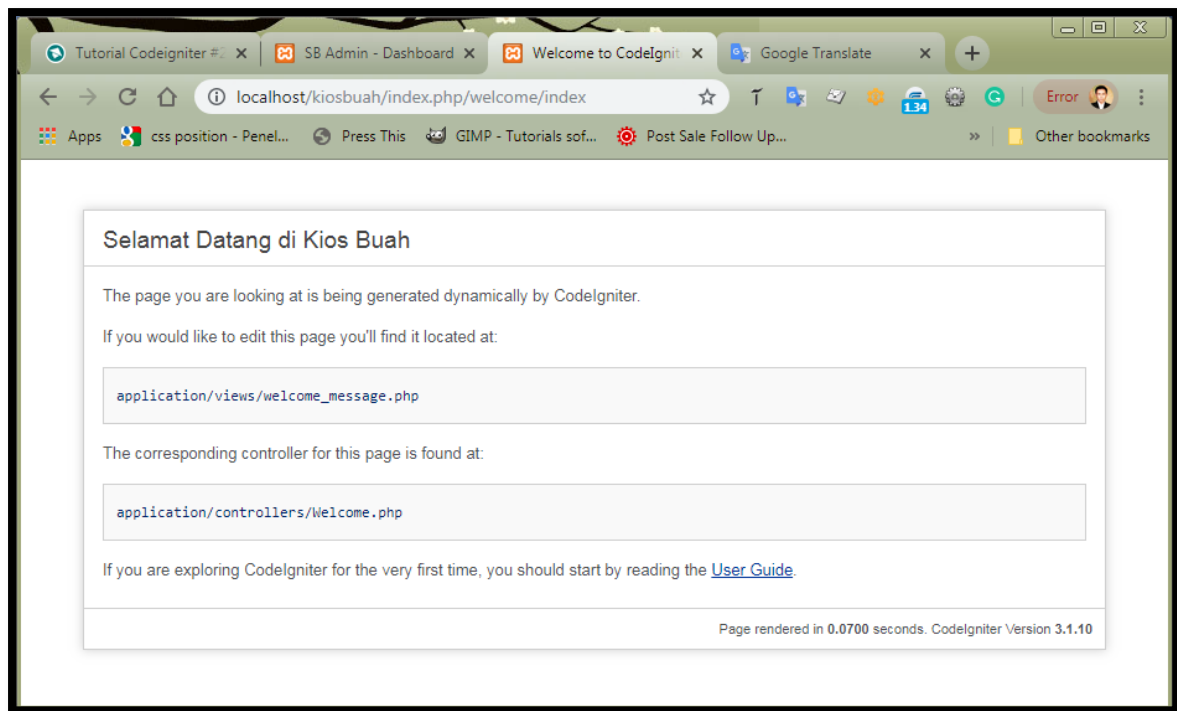
Ini dapat terjadi karena pada konfigurasi router, default controllernya adalah **welcome** seperti sudah ditetapkan pada file **config/routes.php**:

```

49 | Examples: my-controller/index -> my_controller/index
50 |           my-controller/my-method -> my_controller/my_method
51 */
52 $route['default_controller'] = 'welcome';
53 $route['404_override'] = '';
54 $route['translate_uri_dashes'] = FALSE;
55

```

Jika dicoba buka: <http://localhost/tokobuah/index.php/welcome/index>, maka akan didapatkan hal yang sama seperti kalau membuka <http://localhost/tokobuah/>.



Latihan Membuat Router

Tambahkan route **/about/** dan **/contact/** dalam *Controller Welcome* seperti ini:

```
public function about()
{
    // fungsi untuk me-load view about.php
    $this->load->view('about');
}

public function contact()
{
    // fungsi untuk me-load view contact.php
    $this->load->view('contact');
}
```

sehingga *Controller Welcome* menjadi seperti ini:

```
al  Help      • Welcome.php - kiosbuah - Visual Studio Code

welcome_message.php  Welcome.php

1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class Welcome extends CI_Controller {
5
6      /**
7       * Index Page for this controller.
8       *
9       * Since this controller is set as the default controller in
10      * config/routes.php, it's displayed at http://example.com/
11      *
12      */
13      public function index()
14      {
15          $this->load->view('welcome_message');
16      }
17
18      public function about()
19      {
20          // fungsi untuk me-load view about.php
21          $this->load->view('about');
22      }
23
24      public function contact()
25      {
26          // fungsi untuk me-load view contact.php
27          $this->load->view('contact');
28      }
29  }
30
```

Kemudian tambahkan dua buah file view di dalam direktori **views** dengan isi sebagai berikut:

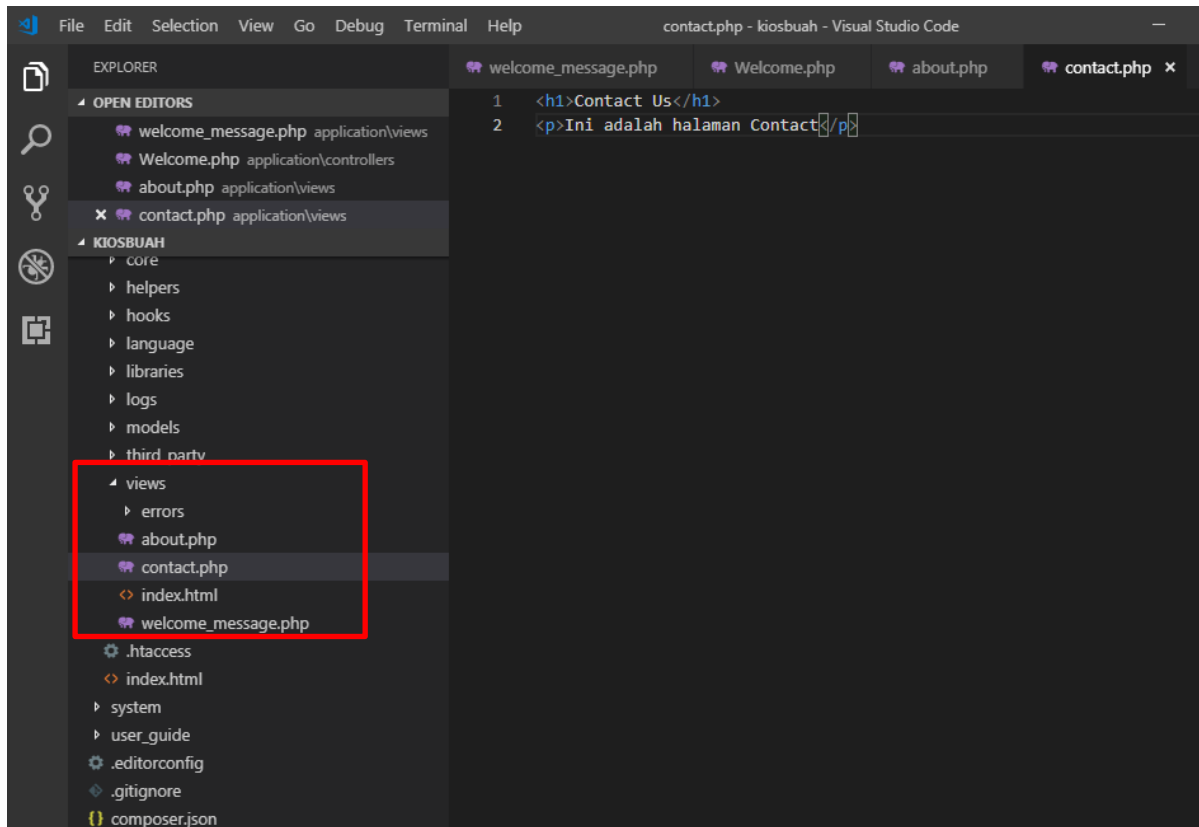
views/about.php

```
<h1>About Us</h1>
<p>Ini adalah halaman about</p>
```

views/contact.php

```
<h1>Contact Us</h1>
<p>Ini adalah halaman Contact</p>
```

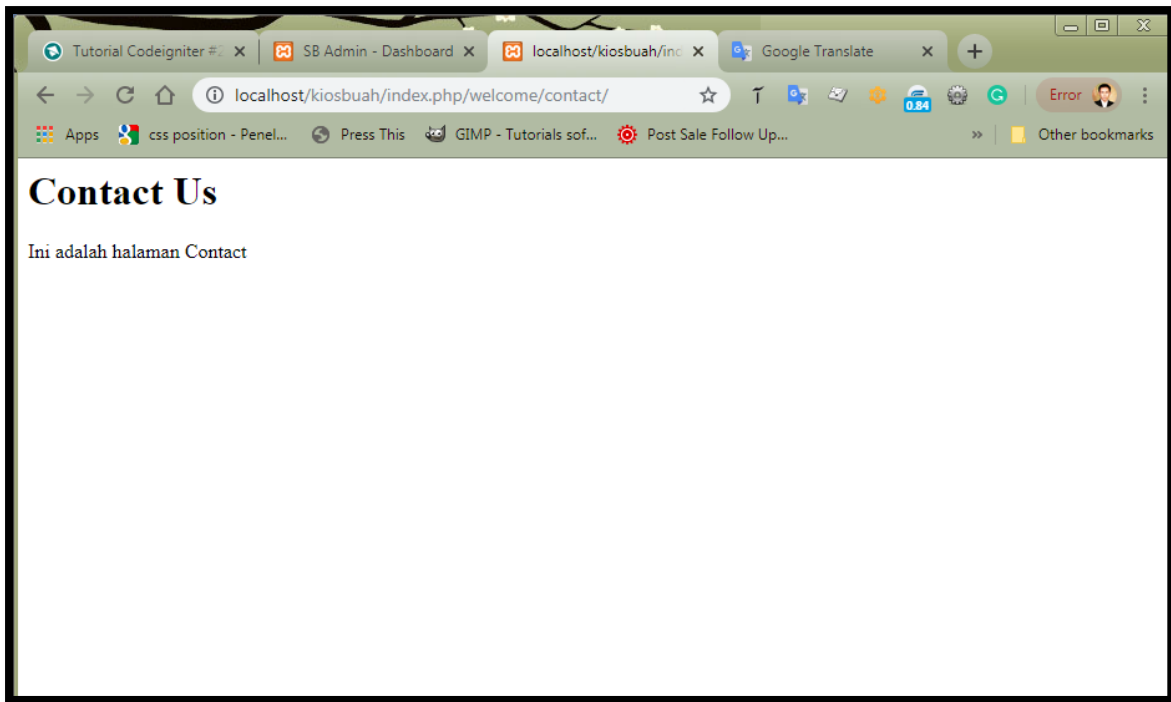
Sehingga direktori **views** menjadi begini:



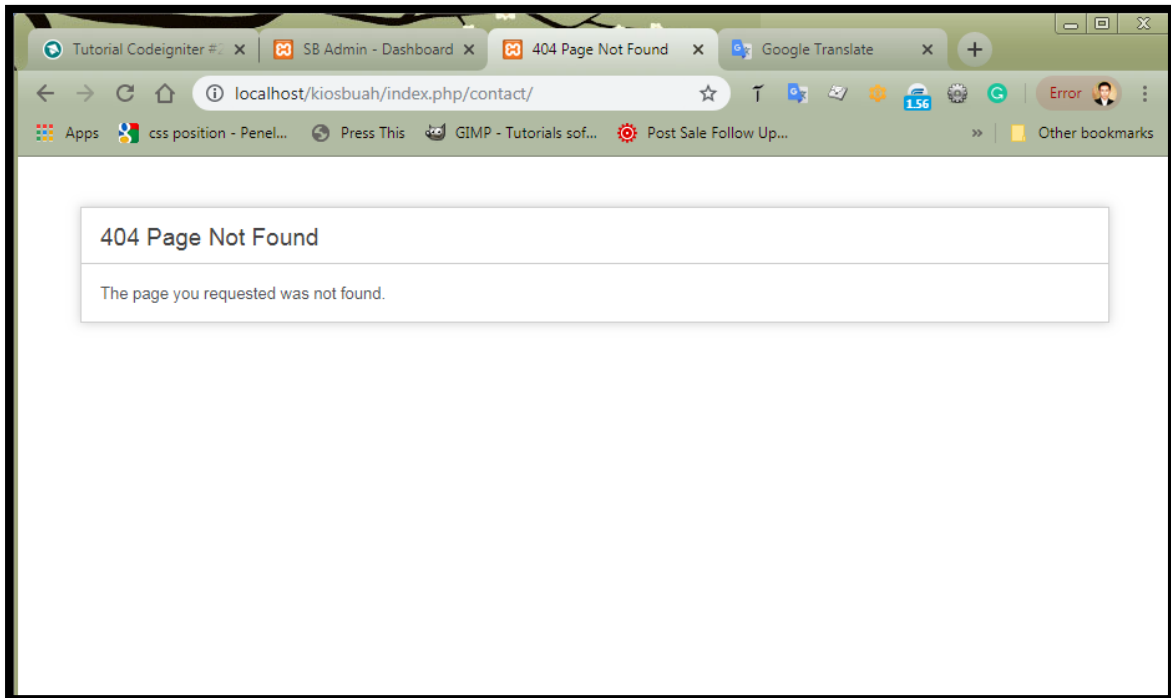
Kemudian coba buka:

- <http://localhost/kiosbuah/index.php/welcome/about/>
- <http://localhost/kiosbuah/index.php/welcome/contact/>

yang menghasilkan:



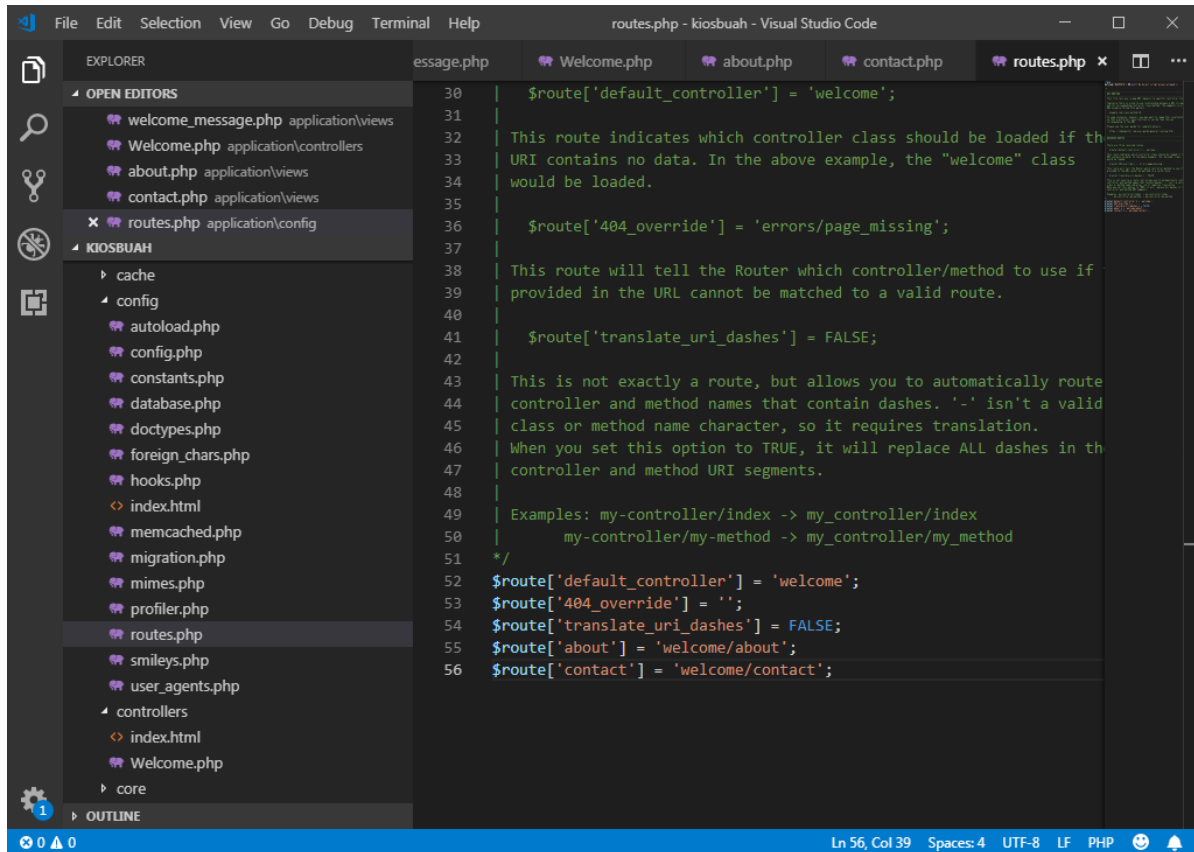
Tetapi jika dijalankan dengan alamat <http://localhost/index.php/contact/> maka akan menghasilkan *error 404 not found*.



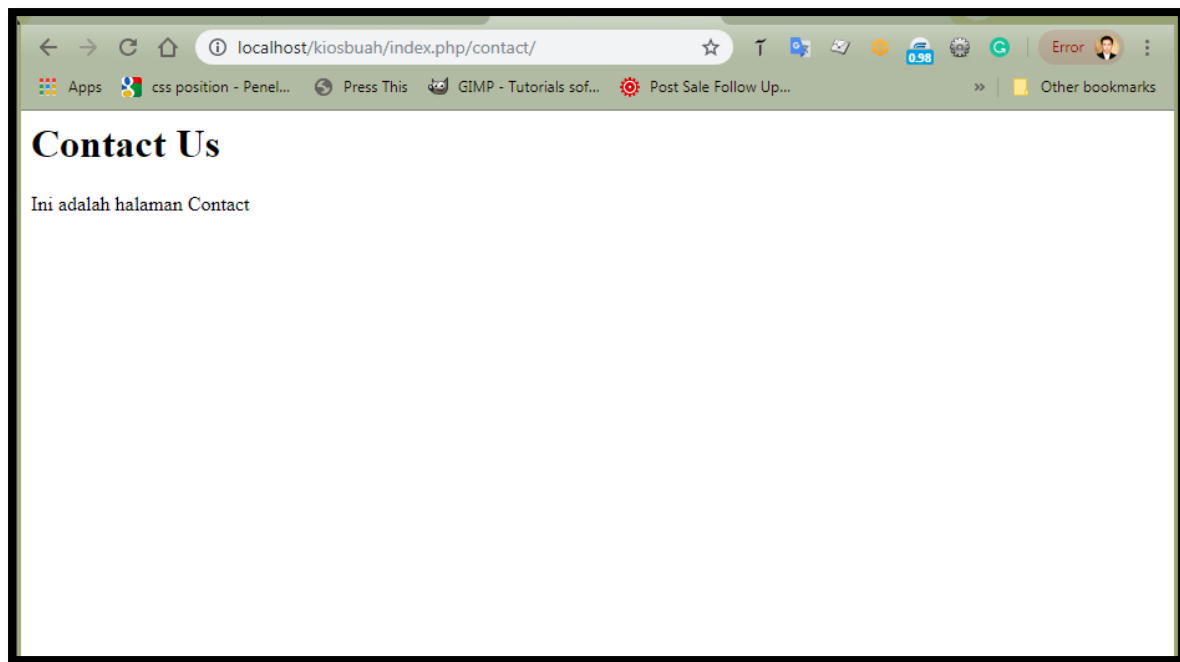
Ini karena route yang baru dibuat belum didaftarkan kedalam file **routes.php**. Tambahkan kode berikut kedalam file **config/routes.php**.

```
$route['about'] = 'welcome/about';  
$route['contact'] = 'welcome/contact';
```

Tambahkan dibawah kode yang sudah ada seperti dibawah ini:

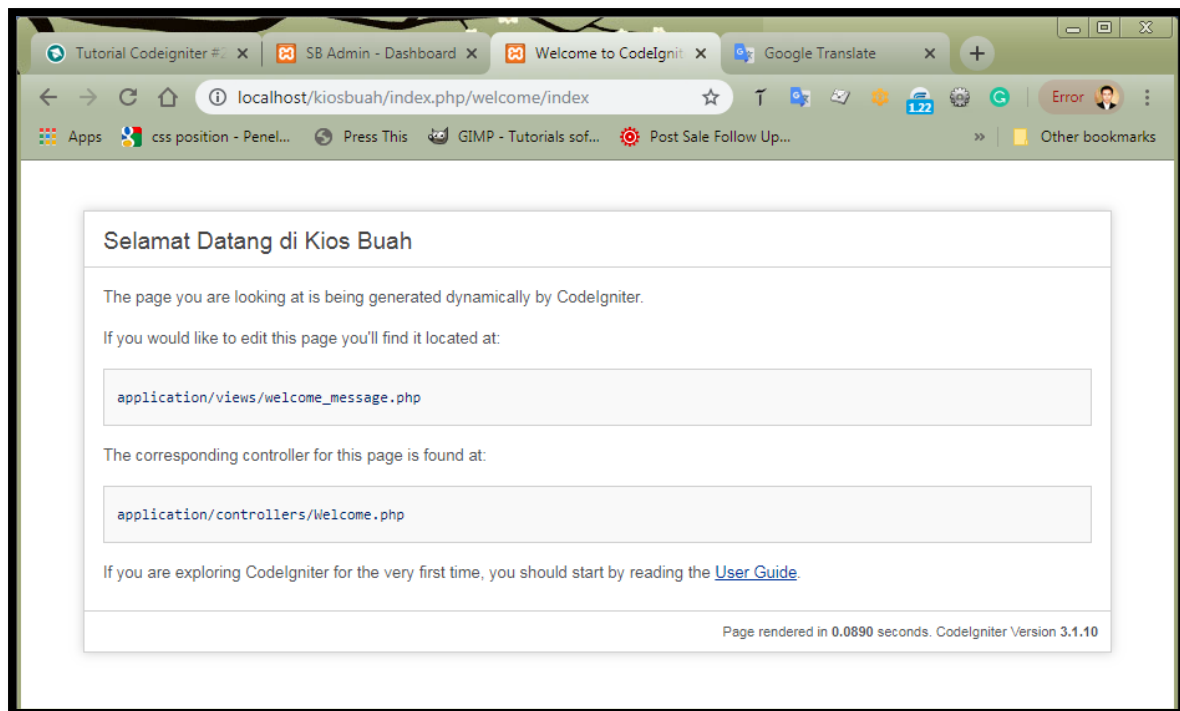


Maka hasilnya:



Penambahan route pada file routes.php ini tidak harus dilakukan karena CI mampu mendeteksi otomatis route berdasarkan nama *controller* dan *method* yang dibuat. Formatnya:

```
http://example.com/[controller-class]/[controller-method]/[arguments]
```



Penambahan route pada routes.php dibutuhkan hanya pada saat ingin dibuat *route kustom* untuk controller tertentu.

Penjelasan selanjutnya dapat dilihat pada file berikutnya.

Referensi

<https://www.petanikode.com/codeigniter-mvc/>