

Modul 5 Framework

Lab.Assignment : Instalasi codeigniter, konfigurasi codeigniter, MVC dasar

A. Tujuan

1. Memahami tentang framework khususnya Codeigniter
2. Memahami instalasi Codeigniter
3. Memahami konfigurasi Codeigniter
4. Memahami Konsep MVC dasar pada Codeigniter

B. Dasar Teori

- Pengertian framework secara umum

Framework adalah kerangka kerja. Framework juga dapat diartikan sebagai kumpulan script (terutama class dan function) yang dapat membantu developer/programmer dalam menangani berbagai masalah-masalah dalam pemrograman seperti koneksi ke database, pemanggilan variabel, file,dll sehingga developer lebih fokus dan lebih cepat membangunaplikasi.Bisa juga dikatakan Framework adalah komponen pemrograman yang siap re-use kapan saja, sehingga programmer tidak harus membuat skrip yang sama untuk tugas yang sama.

- Jenis – jenis framework

- a. Codeigniter
- b. Laravel
- c. Yii

- Manfaat framework

- Manfaatnya dapat membantu kerja developer dalam membangun aplikasi sehingga aplikasi bisa selesai dalam waktu yang singkat.
- Penerapan Design Patterns memudahkan dalam rancangan, pengembangan dan pemeliharaan sistem.
- Stability dan Reliability aplikasi yang kita bangun lebih stabil dan handal karena berbasis pada framework yang sudah teruji stabilitas dan keandalannya.
- Coding Style konsisten, memudahkan dalam membaca kode dan dalam menemukan bugs.
- Security Concern framework mengantisipasi dan memasang perisai terhadap adanya berbagai masalah keamanan yang mungkin timbul.
- Dokumentasi, framework dapat mendisiplinkan kita untuk menulis dokumentasi untuk apa yang kita tulis

- Framework Codeigniter

CodeIgniter adalah sebuah web application framework yang bersifat open source digunakan untuk membangun aplikasi php dinamis. Tujuan utama pengembangan Codeigniter adalah untuk membantu developer untuk mengerjakan aplikasi lebih cepat daripada menulis semua code dari awal. Codeigniter menyediakan berbagai macam library yang dapat mempermudah dalam pengembangan.

Manfaat framework codeigniter ialah sangat ringan, terstruktur, mudah dipelajari, dokumentasi lengkap dan dukungan yang luar biasa dari forum CodeIgniter.

Fitur codeigniter :

- a. **Menggunakan Pattern MVC.** Dengan menggunakan pattern MVC ini, struktur kode yang dihasilkan menjadi lebih terstruktur dan memiliki standar yang jelas.
- b. **URL Friendly.** URL yang dihasilkan sangat *url friendly*. Pada CodeIgniter diminimalisasi penggunaan \$_GET dan di gantikan dengan URI.
- c. **Kemudahan.** Kemudahan dalam mempelajari, membuat library dan helper, memodifikasi serta meng-integrasikan Library dan helper.

Susunan folder Codeigniter secara default :

- Folder **application**: disinilah aplikasi yang akan kita bangun diletakkan.
 - ✓ Folder **config** - tempat menyimpan semua file konfigurasi yang ada di dalam aplikasi, mulai dari database, router dan autoload aplikasi.
 - ✓ Folder **controllers** - tempat menyimpan semua file controller.
 - ✓ Folder **errors** - tempat menyimpan semua template error aplikasi.
 - ✓ Folder **helpers** - tempat menyimpan helper-helper yang bukan berasal dari CI.
 - ✓ Folder **hooks** - tempat menyimpan hook yang digunakan untuk mengubah alur fungsi dari core CI.
 - ✓ Folder **language** - tempat menyimpan bahasa-bahasa yang akan di gunakan.
 - ✓ Folder **libraries** - tempat menyimpan semua library buatan kita sendiri.
 - ✓ Folder **models** - tempat menyimpan semua model.
 - ✓ Folder **views** - tempat menyimpan semua file view aplikasi.
- Folder **system** menyimpan semua file baik itu file aplikasi yang dibuat maupun core framework- nya.
 - ✓ Folder **cache** - tempat menyimpan semua cache yang dibuat caching library.
 - ✓ Folder **codeigniter** - tempat menyimpan semua semua file

internals CI.

- ✓ Folder **database** - tempat menyimpan semua driver database drivers dan class yang akan digunakan.
- ✓ Folder **fonts** - tempat menyimpan semua font yang digunakan image manipulation library.
- ✓ Folder **helpers** - tempat menyimpan semua helper core CI.
- ✓ Folder **language** - tempat menyimpan semua language core CI.
- ✓ Folder **libraries** - tempat menyimpan semua library core CI.
- ✓ Folder **logs** - tempat menyimpan semua logs generated oleh CI.
- ✓ Folder **plugin** - tempat menyimpan semua plugin core CI.
- ✓ Folder **scaffolding** - tempat menyimpan semua file yang berfungsi sebagai scaffolding.
- Folder **user_guide** berisi userguide/manual penggunaan CI.
- File **index.php** file yang akan handle semua request yang dilakukan oleh client.

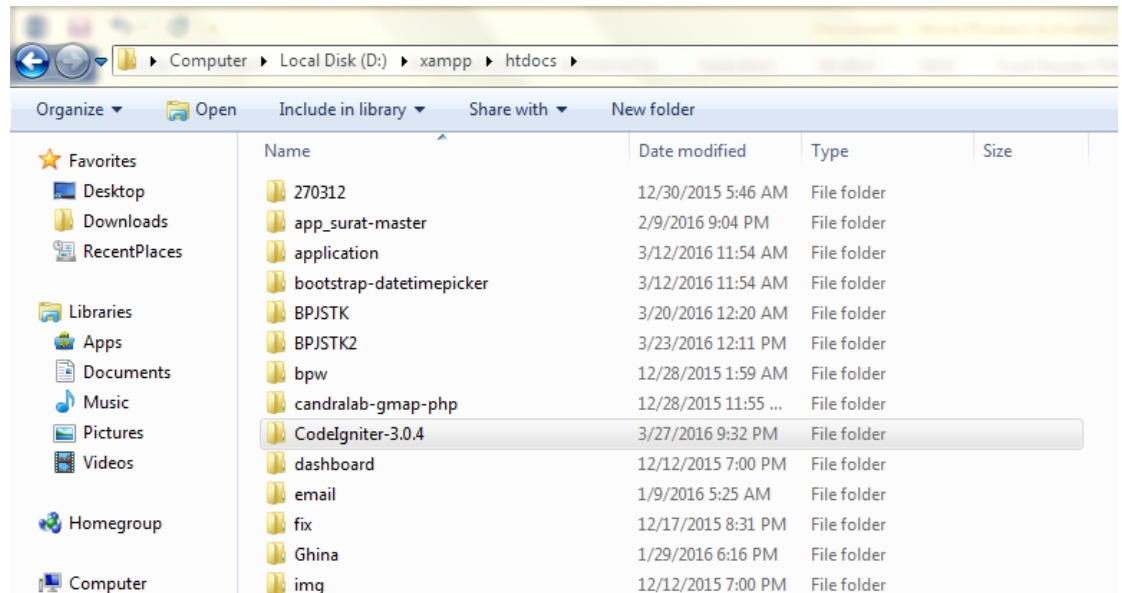
C. Peralatan

1. Editor Sublime
2. Browser
3. Webserver
4. Database server (mysql)
5. Codeigniter

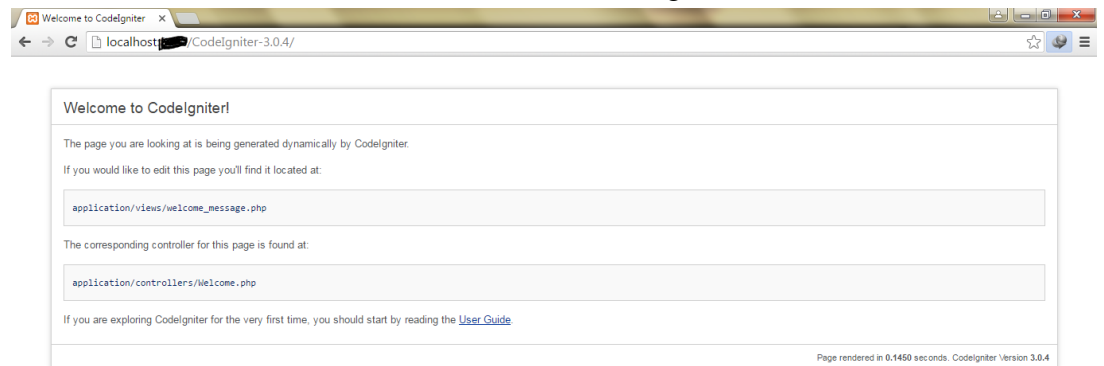
D. Latihan

1. Instalasi Codeigniter

- Aktifkan apache serta mysql
- Copy folder codeigniter yang telah di download di <http://www.Codeigniter.com/> pada htdocs



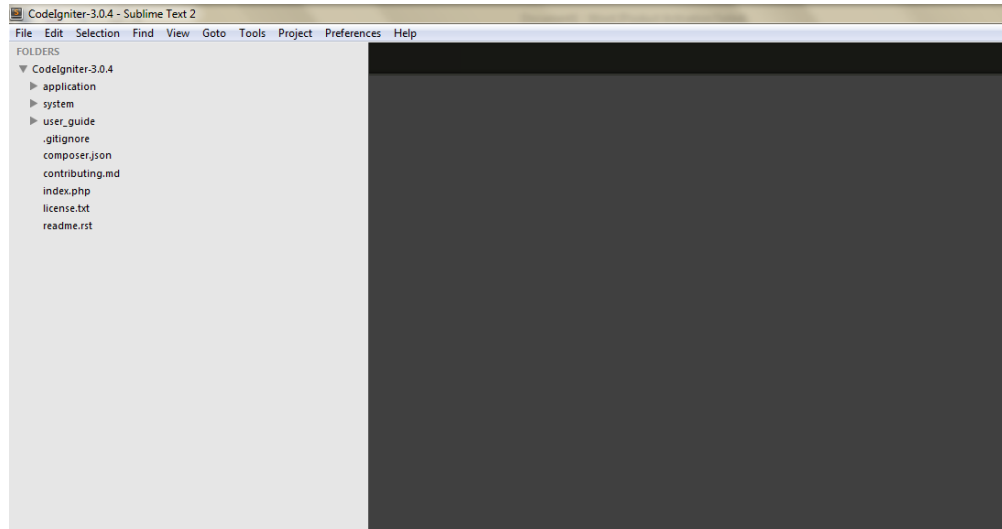
- Setelah itu, buka browser dan ketik localhost/Codeigniter-3.0.4



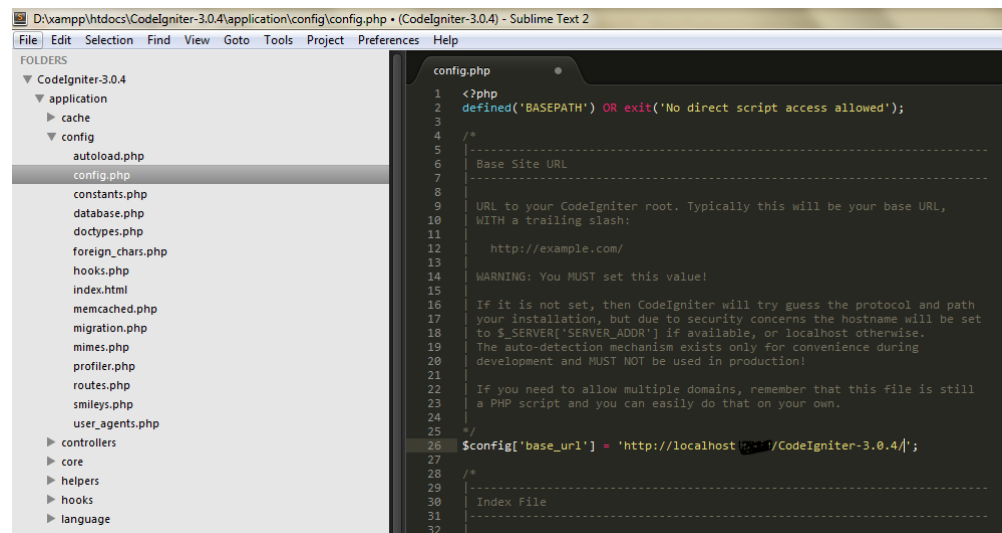
- Instalasi selesai

2. Konfigurasi Codeigniter

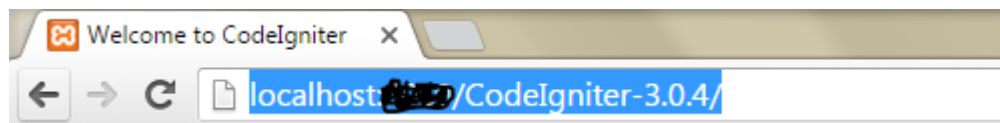
- Untuk memulai konfigurasi, buka codeigniter dengan open folder di sublime



- Pertama, atur alamat url dengan cara

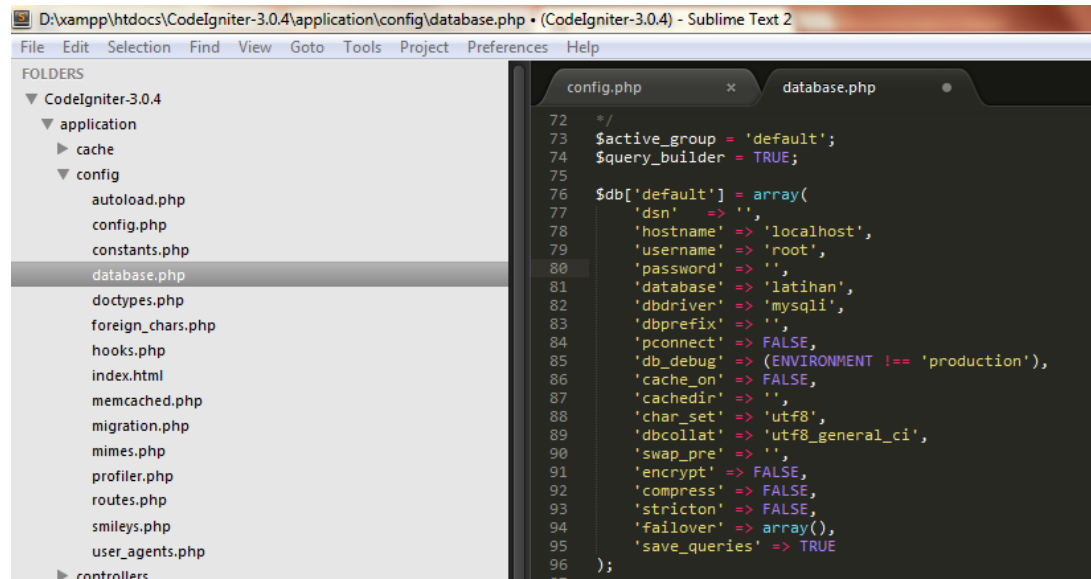


<http://localhost/Codeigniter-3.0.4/> di dapat dari alamat web yang di buka di browser



konfigurasi `$config['base_url'] = "http://localhost/Codeigniter-3.0.4"` ini berfungsi untuk alamat website

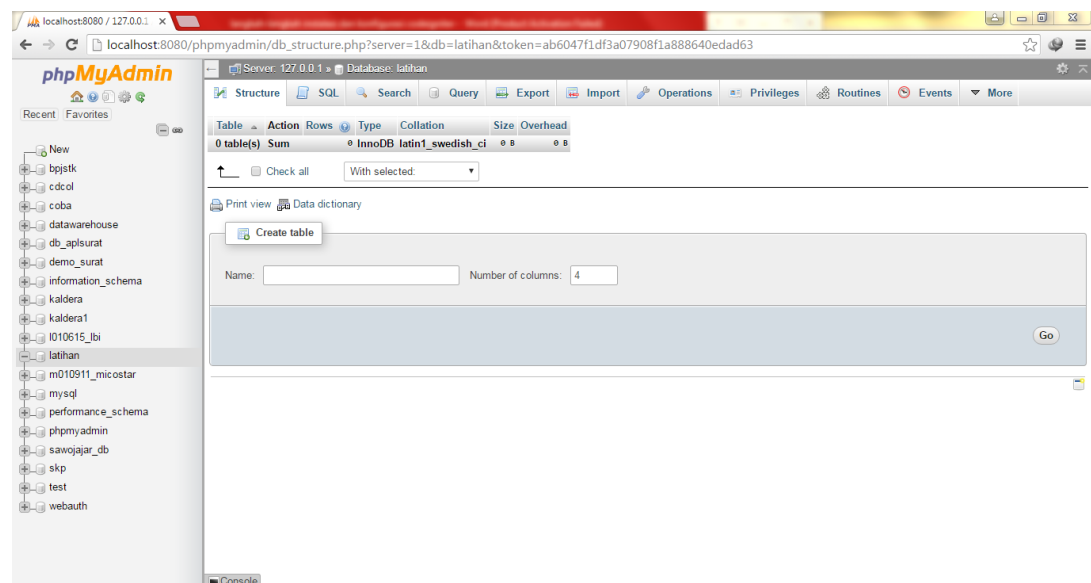
- Kedua, konfigurasi database



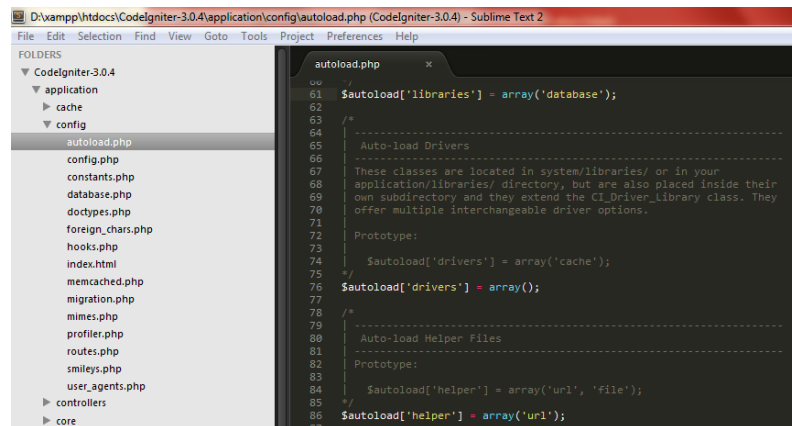
```

72  */
73  $active_group = 'default';
74  $query_builder = TRUE;
75
76  $db['default'] = array(
77      'dsn' => '',
78      'hostname' => 'localhost',
79      'username' => 'root',
80      'password' => '',
81      'database' => 'latihan',
82      'dbdriver' => 'mysqli',
83      'dbprefix' => '',
84      'pconnect' => FALSE,
85      'db_debug' => (ENVIRONMENT !== 'production'),
86      'cache_on' => FALSE,
87      'cachedir' => '',
88      'char_set' => 'utf8',
89      'dbcollat' => 'utf8_general_ci',
90      'swap_pre' => '',
91      'encrypt' => FALSE,
92      'compress' => FALSE,
93      'stricton' => FALSE,
94      'failover' => array(),
95      'save_queries' => TRUE
96  );
  
```

'hostname' => 'localhost' : itu merupakan nama dari host database kalian
 'username' => 'root' : itu merupakan username mysql
 'password' => '' : itu merupakan password dari sql
 'database' => 'database' : itu merupakan nama dari database yang telah di buat



- Ketiga, konfigurasi autoload



```

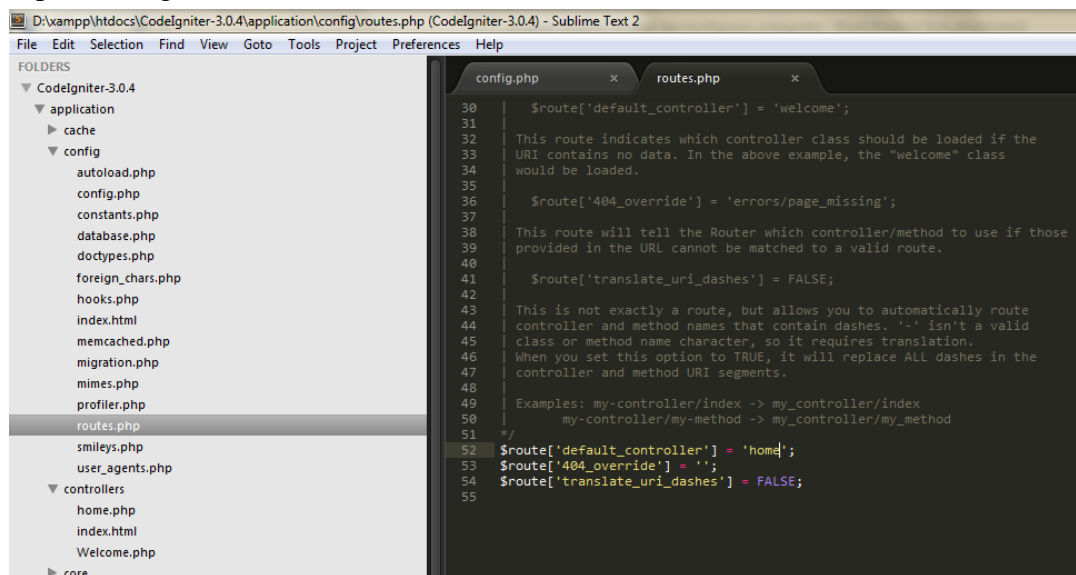
61 $autoload['libraries'] = array('database');
62
63 /*
64 -----
65 Auto-load Drivers
66 -----
67 These classes are located in system/libraries/ or in your
68 application/libraries/ directory, but are also placed inside their
69 own subdirectory and they extend the CI_Driver_Library class. They
70 offer multiple interchangeable driver options.
71
72 Prototype:
73
74 $autoload['drivers'] = array('cache');
75 */
76 $autoload['drivers'] = array();
77
78 /*
79 -----
80 Auto-load Helper Files
81 -----
82 Prototype:
83
84 $autoload['helper'] = array('url', 'file');
85 */
86 $autoload['helper'] = array('url');
87

```

`$autoload['libraries'] = array('database')` berfungsi untuk mengaktifkan library database.

`$autoload['helper'] = array('url')` berfungsi untuk menggunakan mengaktifkan url

- Keempat, konfigurasi default controller



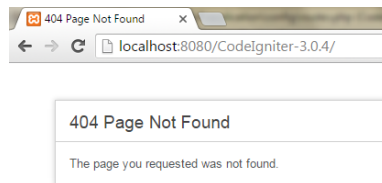
```

30 $route['default_controller'] = 'welcome';
31
32 This route indicates which controller class should be loaded if the
33 URI contains no data. In the above example, the "welcome" class
34 would be loaded.
35
36 $route['404_override'] = 'errors/page_missing';
37
38 This route will tell the Router which controller/method to use if those
39 provided in the URL cannot be matched to a valid route.
40
41 $route['translate_uri_dashes'] = FALSE;
42
43 This is not exactly a route, but allows you to automatically route
44 controller and method names that contain dashes. '-' isn't a valid
45 class or method name character, so it requires translation.
46 When you set this option to TRUE, it will replace ALL dashes in the
47 controller and method URI segments.
48
49 Examples: my-controller/index -> my_controller/index
50             my-controller/my-method -> my_controller/my_method
51
52 $route['default_controller'] = 'home';
53 $route['404_override'] = '';
54 $route['translate_uri_dashes'] = FALSE;
55

```

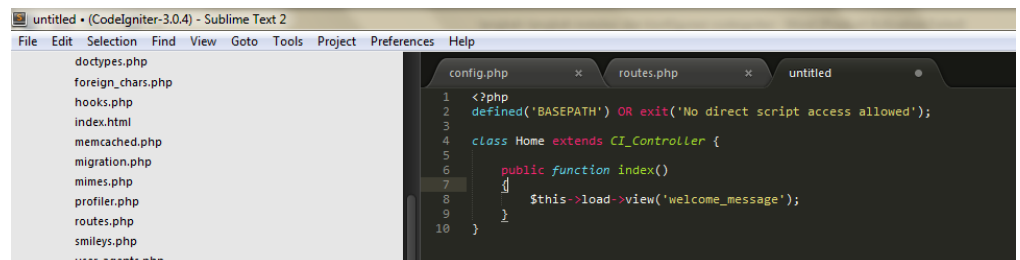
konfigurasi `$route['default_controller'] = "home"` ini berfungsi untuk merubah frontcontroller (sebuah kelas controller yang akan di eksekusi pertama kali ketika

kita mengakses alamat web). Sekarang silahkan refresh kembali website yang telah di konfigurasi

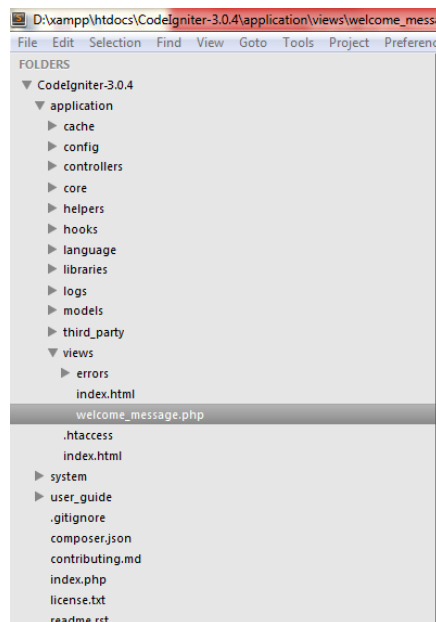


Kalau hasilnya seperti ini dan pasti seperti ini, maka hal yang harus dilakukan adalah membuat controller, dengan cara :

- Tekan ctrl + N atau klik file dan pilih new file
- Ketikkan seperti contoh di bawah ini

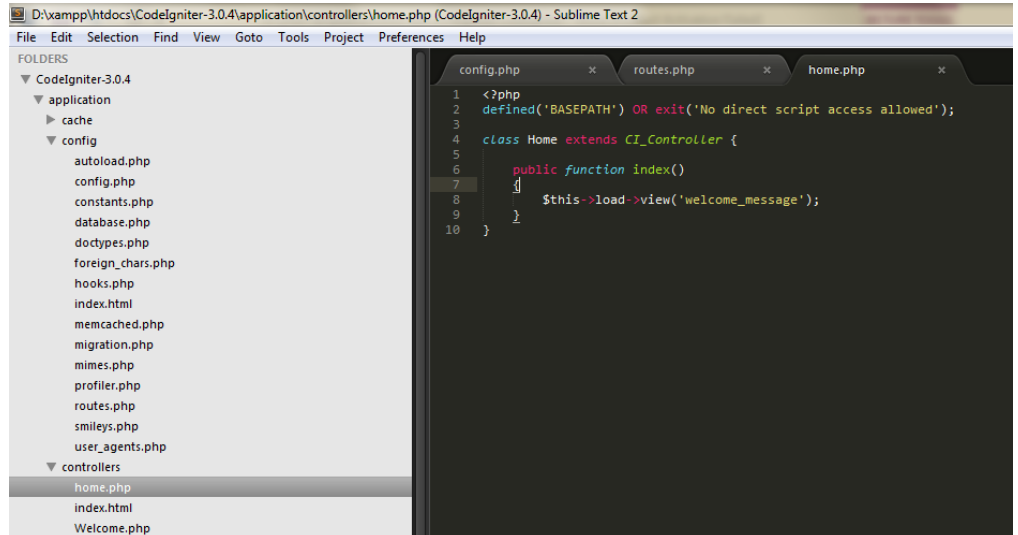


Home adalah sebuah nama kelas dan juga nama file nantinya
Public function index(){} : merupakan sebuah fungsi yang dikerjakan pertama kali ketika kita memanggil kelas ini. Dan disini kita tetap memanggil atau meload view welcome_message pada folder view

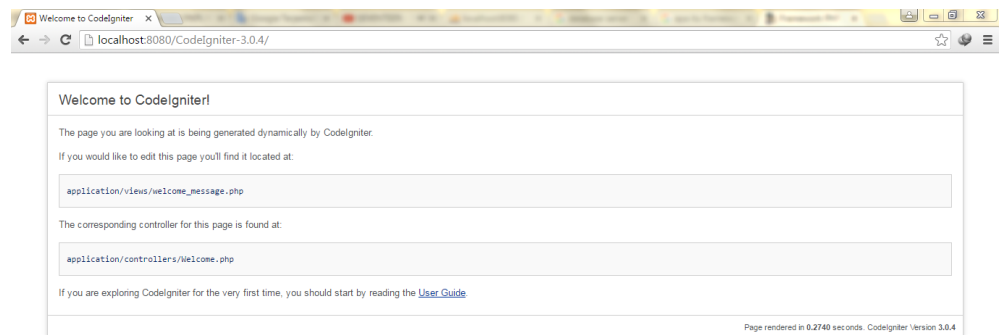


Gambar file welcome_message.php di folder views

- Simpan file diatas dengan nama home.php di folder application -> Controllers

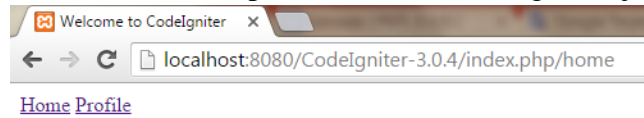


- Maka hasilnya sebagai berikut

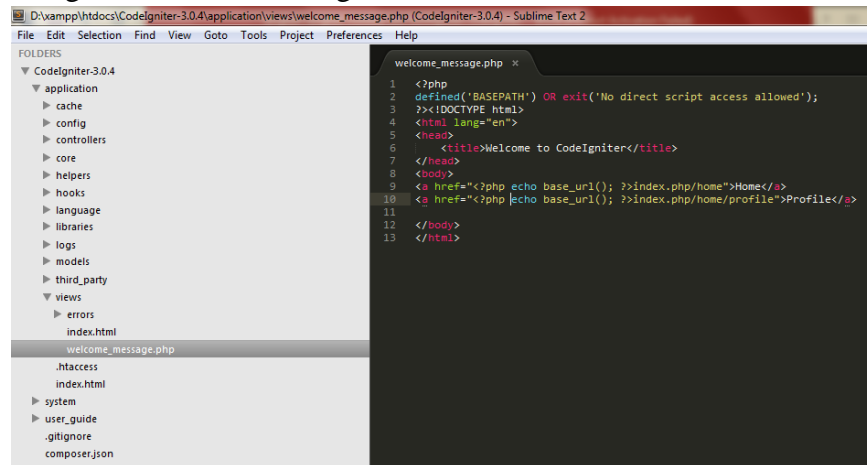


3. MVC dasar

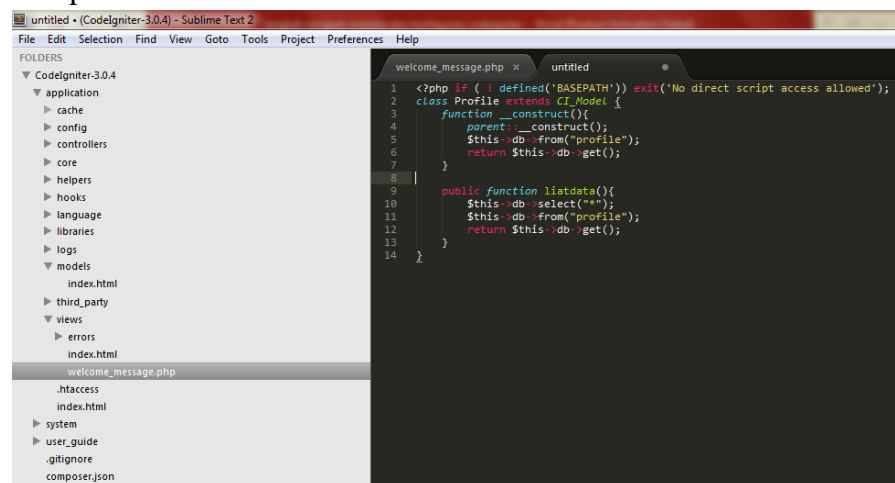
- Pertama, ubah tampilan welcome_message menjadi



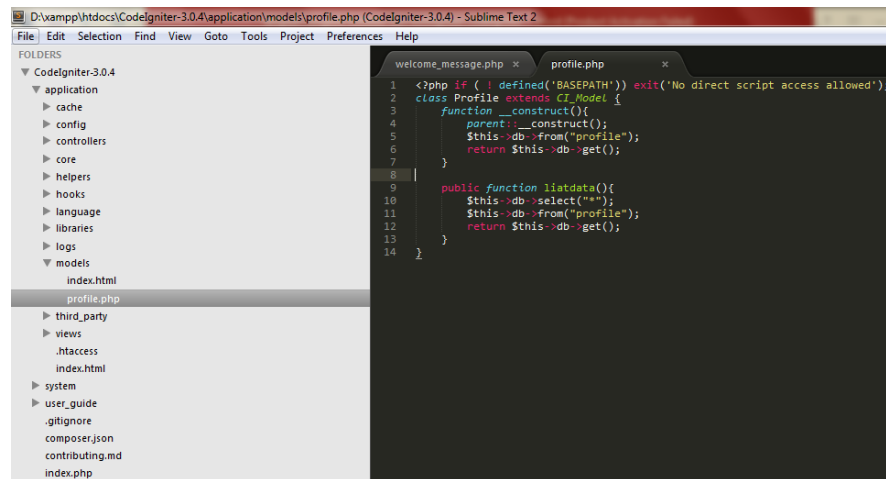
Dengan source code sebagai berikut :



- Kedua, buat kelas profile untuk mengambil data dari database dengan cara klik File pilih new file atau tekan ctrl + n



Simpan dengan nama “profile.php” di folder model



Keterangan :

`$this->db->from("profile");`

`return $this->db->get();`

Merupakan pendefinisian dari tabel database yang akan selalu diakses dalam kelas profile

`$this->db->select("*");`

`$this->db->from("profile");`

`return $this->db->get();`

yang ada pada function lihatdata merupakan perintah sql untuk menampilkan data dari database yang fungsinya sama seperti “SELECT * FROM `profile`”

#	Name	Type	Collation	Attributes
<input type="checkbox"/> 1	NIM 🔑	varchar(20)		
<input type="checkbox"/> 2	Nama	varchar(30)		
<input type="checkbox"/> 3	Angkatan	varchar(20)		
<input type="checkbox"/> 4	Alamat	varchar(255)		

Gambar structure tabel profile

- Ketiga, konfigurasi controller home.php

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Home extends CI_Controller {
5     public function __construct() {#
6         parent::__construct();
7         $this->load->model('profile');
8     }
9
10    public function index()
11    {
12        $this->load->view('welcome_message');
13    }
14
15    public function profile(){
16        $data['identitas'] = $this->profile->lihatdata();
17        $this->load->view('profile', $data);
18    }
19 }
  
```

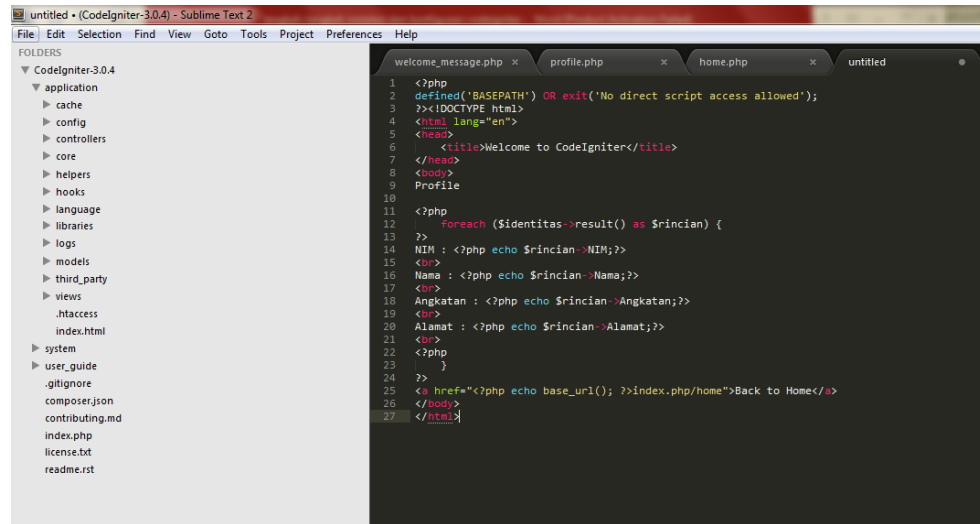
Keterangan :

`$this->load->model('profile');` merupakan sintax untuk memanggil model yang telah kita buat tadi.

`$data['identitas'] = $this->profile->lihatdata();` merupakan sintax untuk membuat objek data yang didapat dari proses select data dari database yang berasal dari class profile dalam fungsi lihatdata yang nantinya untuk diproses di view.

`$this->load->view('profile', $data);` merupakan sintax untuk memanggil file profile di view serta mengirim data dari proses diatas.

- Keempat, create file view dengan cara klik File pilih new file atau tekan ctrl + n

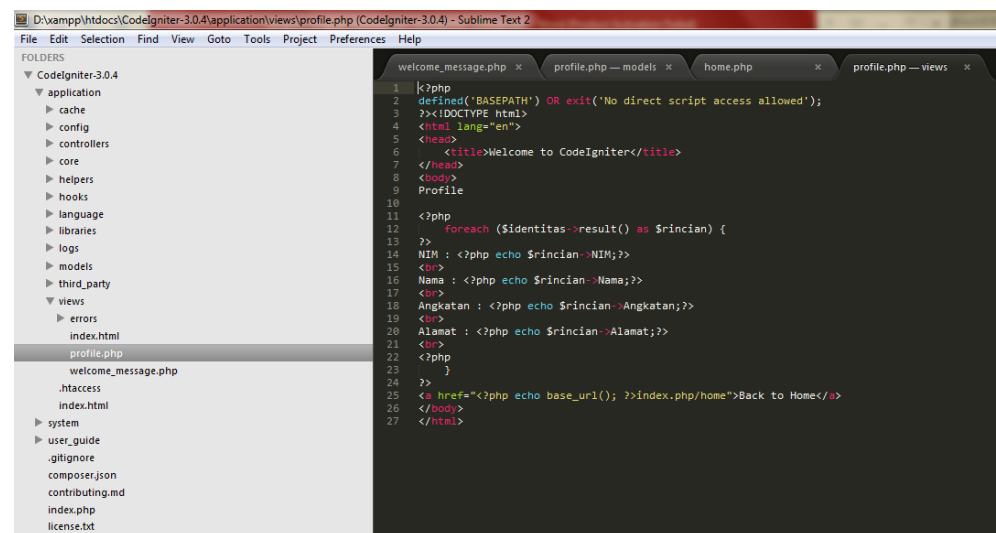


```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 >><!DOCTYPE html>
4 <html lang="en">
5 <head>
6 <title>Welcome to CodeIgniter</title>
7 </head>
8 <body>
9 Profile
10
11 <?php
12 foreach ($identitas->result() as $rincian) {
13 }
14 NIM : <?php echo $rincian->NIM;?>
15 <br>
16 Nama : <?php echo $rincian->Nama;?>
17 <br>
18 Angkatan : <?php echo $rincian->Angkatan;?>
19 <br>
20 Alamat : <?php echo $rincian->Alamat;?>
21 <br>
22 <?php
23 }
24 }
25 <a href="<?php echo base_url(); ?>index.php/home">Back to Home</a>
26 </body>
27 </html>

```

Simpan di folder view dengan nama “profile.php”

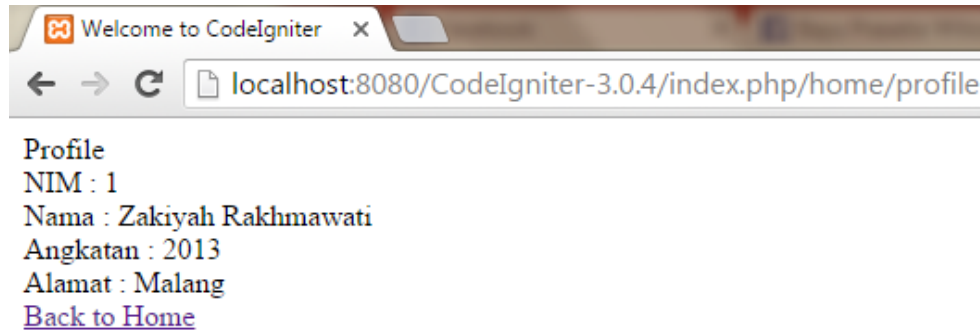


```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 >><!DOCTYPE html>
4 <html lang="en">
5 <head>
6 <title>Welcome to CodeIgniter</title>
7 </head>
8 <body>
9 Profile
10
11 <?php
12 foreach ($identitas->result() as $rincian) {
13 }
14 NIM : <?php echo $rincian->NIM;?>
15 <br>
16 Nama : <?php echo $rincian->Nama;?>
17 <br>
18 Angkatan : <?php echo $rincian->Angkatan;?>
19 <br>
20 Alamat : <?php echo $rincian->Alamat;?>
21 <br>
22 <?php
23 }
24 }
25 <a href="<?php echo base_url(); ?>index.php/home">Back to Home</a>
26 </body>
27 </html>

```

- Kelima, sekarang kembali ke browser dan klik profile



Note : isi database di isi manual

E. Tugas

Transformasikan UI **Tugas Besar** kalian ke dalam Framework Codeigniter (Minimal UI, Sistem Login)