



Apa itu codeigniter?

Codeigniter adalah salah satu framework untuk membuat website dengan bahasa pemrograman PHP. Codeigniter terkenal dengan konsep MVC-nya. MVC merupakan singkatan dari Model–View–Controller.



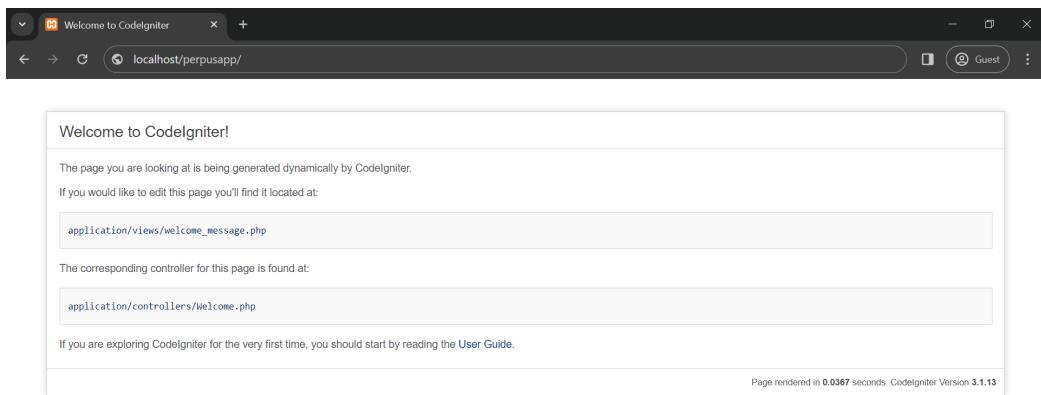
Persiapan sebelum membuat aplikasi dengan codeigniter

- ↗ Teks Editor (Sublime)
- ↗ Web Browser (Chrome)
- ↗ Web Server: PHP, MySQL, Phpmyadmin (XAMPP)
- ↗ File Project Codeigniter (Codeigniter 3 - Zip file)
- ↗ File Asset Pendukung (Bootstrap 5 - Zip file)



Persiapan membuat project baru untuk aplikasi codeigniter 3

- ↗ Download Codeigniter 3
- ↗ Ekstrak File Codeigniter-3.zip ke htdocs, dan ubah namanya menjadi perpusapp
- ↗ Penamaan “perpusapp” dipilih karena kita akan membuat aplikasi sederhana untuk mendata anggota, dan buku yang dipinjam maupun dikembalikan. Diharapkan aplikasi sederhana ini akan dapat dikembangkan ke aplikasi yang lebih profesional.
- ↗ Aktifkan Apache dan MySQL di XAMPP masing-masing, kemudian silahkan buka: <http://localhost/perpusapp/>

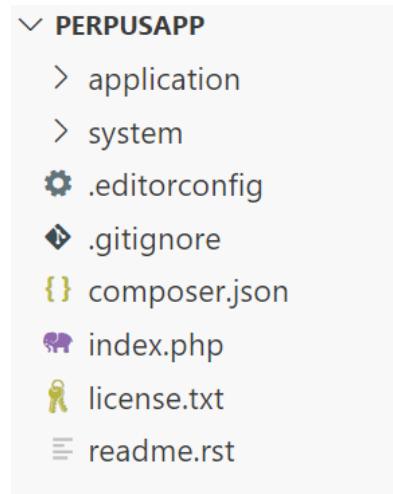


- ↗ Selamat kamu sudah membuat project codeigniter baru

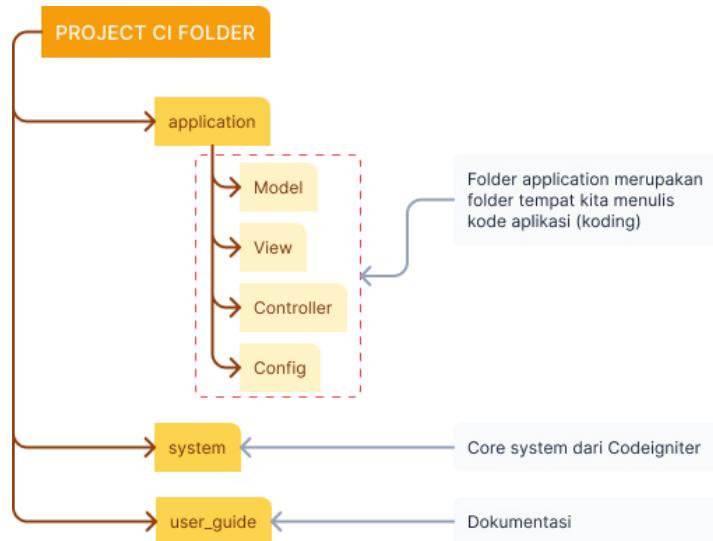


Struktur direktori codeigniter 3

- ↗ Silahkan buka Sublime, lalu pilih File→Open Folder. Cari folder perpusapp di htdocs.



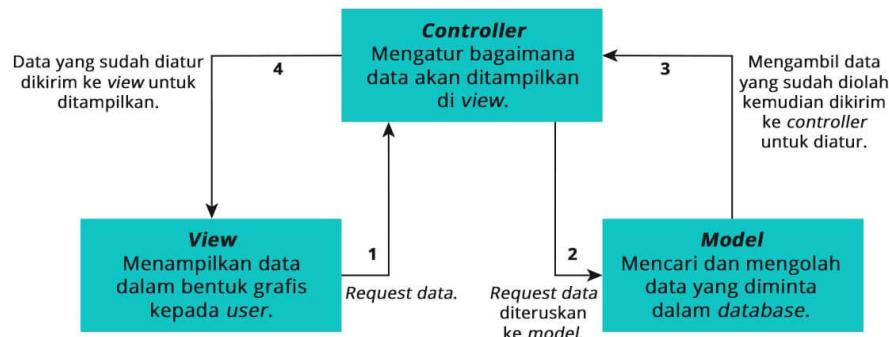
- ↗ Tedapat dua direktori penting di dalam CI yang harus kamu perhatikan, yakni: application dan system.





Apa itu konsep MVC pada codeigniter

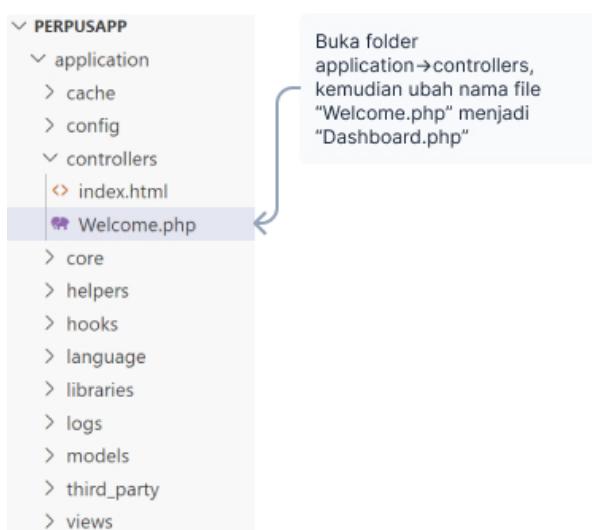
MVC adalah singkatan dari Model–View–Controller. MVC merupakan sebuah pola desain arsitektur yang umum digunakan dalam pengembangan aplikasi. MVC ini adalah pola penulisan kode yang umum dipakai, dimana kode untuk Model di taruh dalam folder yang sama, begitu juga dengan kode untuk View dan Controller.



- ↗ Model adalah bagian yang bertugas untuk menyiapkan, mengatur, memanipulasi, dan mengorganisasikan data yang ada di database.
- ↗ View adalah bagian yang bertugas untuk menampilkan informasi dalam bentuk Graphical User Interface (GUI).
- ↗ Controller adalah bagian yang bertugas untuk menghubungkan serta mengatur model dan view agar dapat saling terhubung.



Mengubah controller dan view pertama



- ↗ Dengan memanfaatkan file yang sudah ada, kita ubah menjadi file yang akan digunakan untuk halaman awal aplikasi.

- ↗ Dilanjutkan mengubah isi file controller yang sudah kita ubah namanya seperti berikut :

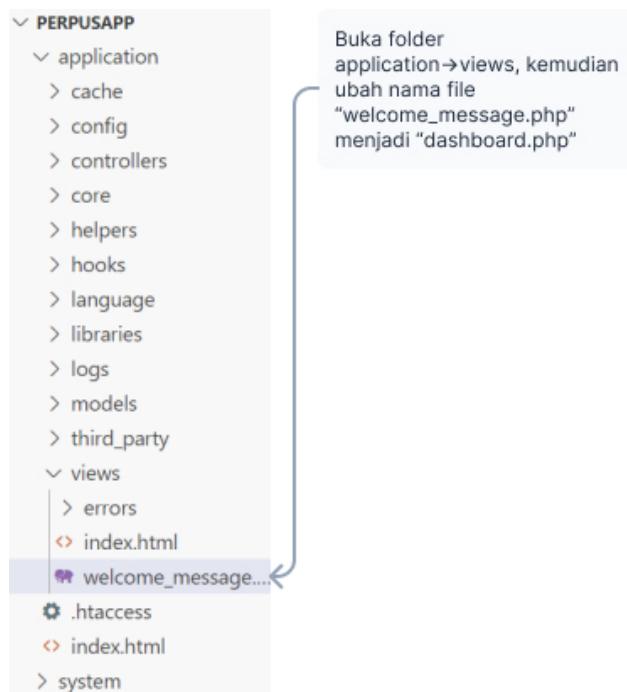
Buka file "Dashboard.php",
kamudian ubah kode menjadi
berikut

```

Dashboard.php X
application > controllers > Dashboard.php > ...
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class Dashboard extends CI_Controller {
5      public function index()
6      {
7          $this->load->view('dashboard');
8      }
9  }

```

- ↗ Setelah ini kita akan mengubah bagian view, dimana kalian bisa perhatikan ada kode yang ditandai dengan garis merah. Itu merupakan kode yang menunjukan lokasi view kita disimpan "\$this→load→view('dashboard')."
- ↗ Yang artinya controller "Dashboard" akan memanggil view yang bernama "dashboard" di dalam folder application→views→dashboard.php



- ↗ Seperti sebelumnya, dengan memanfaatkan file yang sudah ada, kita ubah menjadi file yang akan digunakan untuk halaman awal aplikasi.
- ↗ Dilanjutkan mengubah isi file view yang sudah kita ubah namanya seperti berikut :

Buka file "dashboard.php"
kemudian ubah kode menjadi
berikut



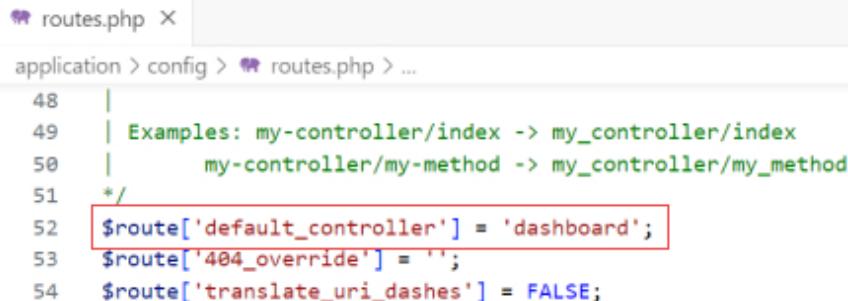
```

dashboard.php X
application > views > dashboard.php > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9      <h1>Selamat datang di Project <strong>PERPUSAPP</strong></h1>
10 </body>
11 </html>

```

- ↗ Terakhir kita akan mengubah file "routes.php" agar saat pertamakali aplikasi kita jalankan akan mengarah ke halaman dashboard
- ↗ Buka folder application→config, kemudian buka file "routes.php". Setelah itu ubah kode kira-kira dibagian akhir menjadi seperti berikut :

Buka file "routes.php"
kemudian ubah kode menjadi
berikut



```

routes.php X
application > config > routes.php > ...
48
49 | Examples: my-controller/index -> my_controller/index
50 |         my-controller/my-method -> my_controller/my_method
51 */
52 $route['default_controller'] = 'dashboard';
53 $route['404_override'] = '';
54 $route['translate_uri_dashes'] = FALSE;

```

- ↗ Silahkan reload kembali halaman aplikasi pada browser kalian, maka akan tampil halaman seperti berikut :



Selamat datang di Project PERPUSAPP

- ↗ Dengan ini kalian sudah berhasil membuat aplikasi dengan codeigniter dan mampu memahami controller dan view pada codeigniter 🎉