



MODUL CODEIGNITER PART I

Jefri Maruli H



PENDAHULUAN

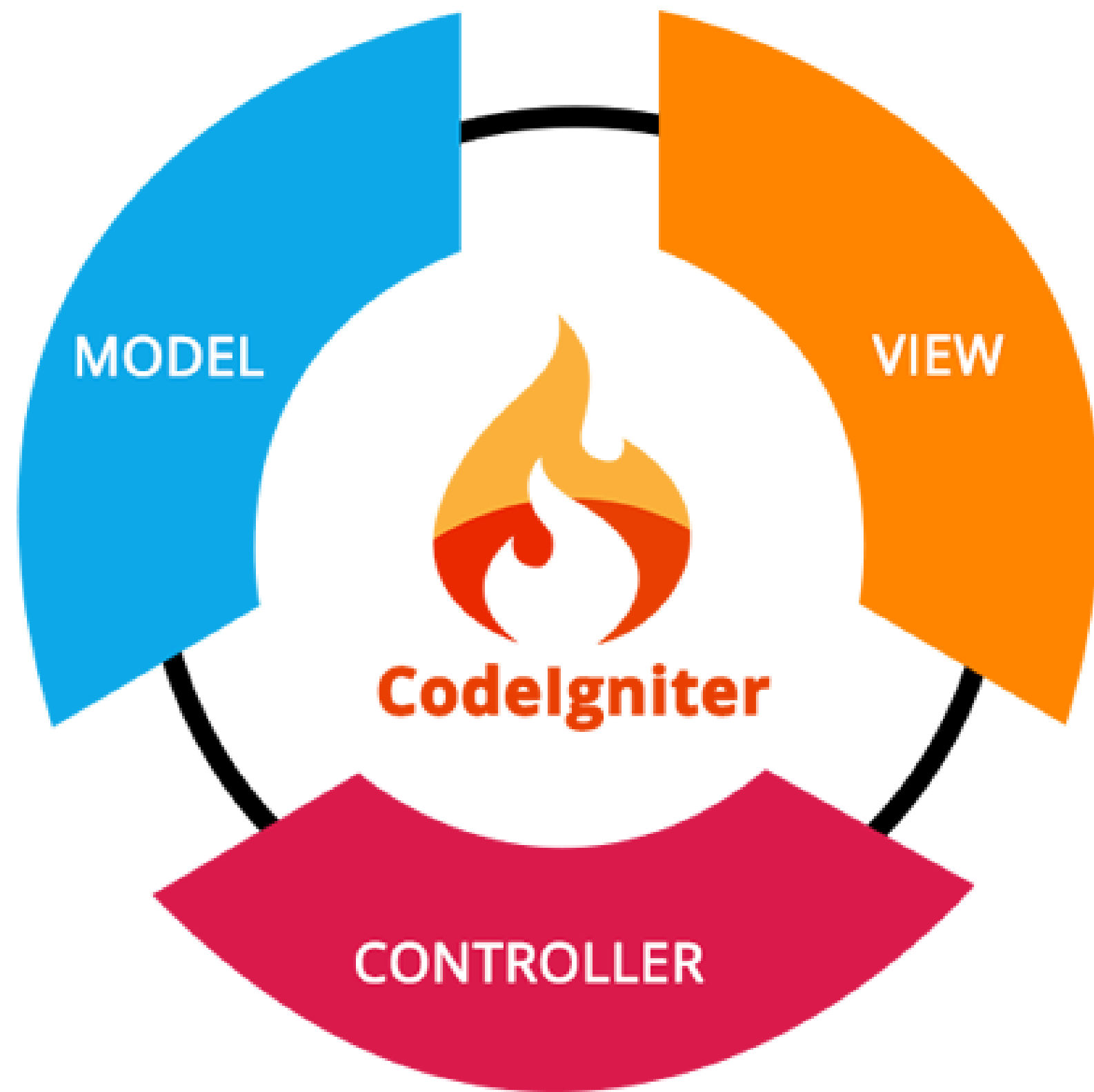
Sebelum pengenalan tentang CodeIgniter, penulis akan menjelaskan terlebih dahulu apa itu framework, karena CodeIgniter merupakan bagian dari framework itu sendiri

FRAMEWORK

Framework dalam bahasa Indonesia artinya kerangka kerja atau kumpulan intruksi yang dikumpulkan dalam satu class, dan beberapa function dengan fungsi masing-masing untuk memudahkan developer/programmer tanpa harus menulis syntax program yang sama berulang ulang.

sumber referensi : malasngoding.com





Apa itu CodeIgniter?

CodeIgniter merupakan sebuah framework PHP yang menggunakan pola desain MVC (Model View Controller), yang dibuat untuk memudahkan developer atau programmer dalam membangun aplikasi berbasis web tanpa harus membuatnya dari awal.

KELEBIHAN CODEIGNITER

- **Performa Cepat**
- **Konfigurasi yang minim**
- **Cocok dipelajari bagi pemula**
- **Tidak bergantung dengan aturan dari CI**
- **Dokumentasi yang lengkap**
- **Komunitas yang cukup ramai**



HAL YANG HARUS DIPAHAMI SEBELUM MENGGUNAKAN CODEIGNITER

BAHASA PEMROGRAMAN PHP

Minimal memahami atau menguasai bahasa pemrograman PHP secara prosedural.

KONSEP MVC

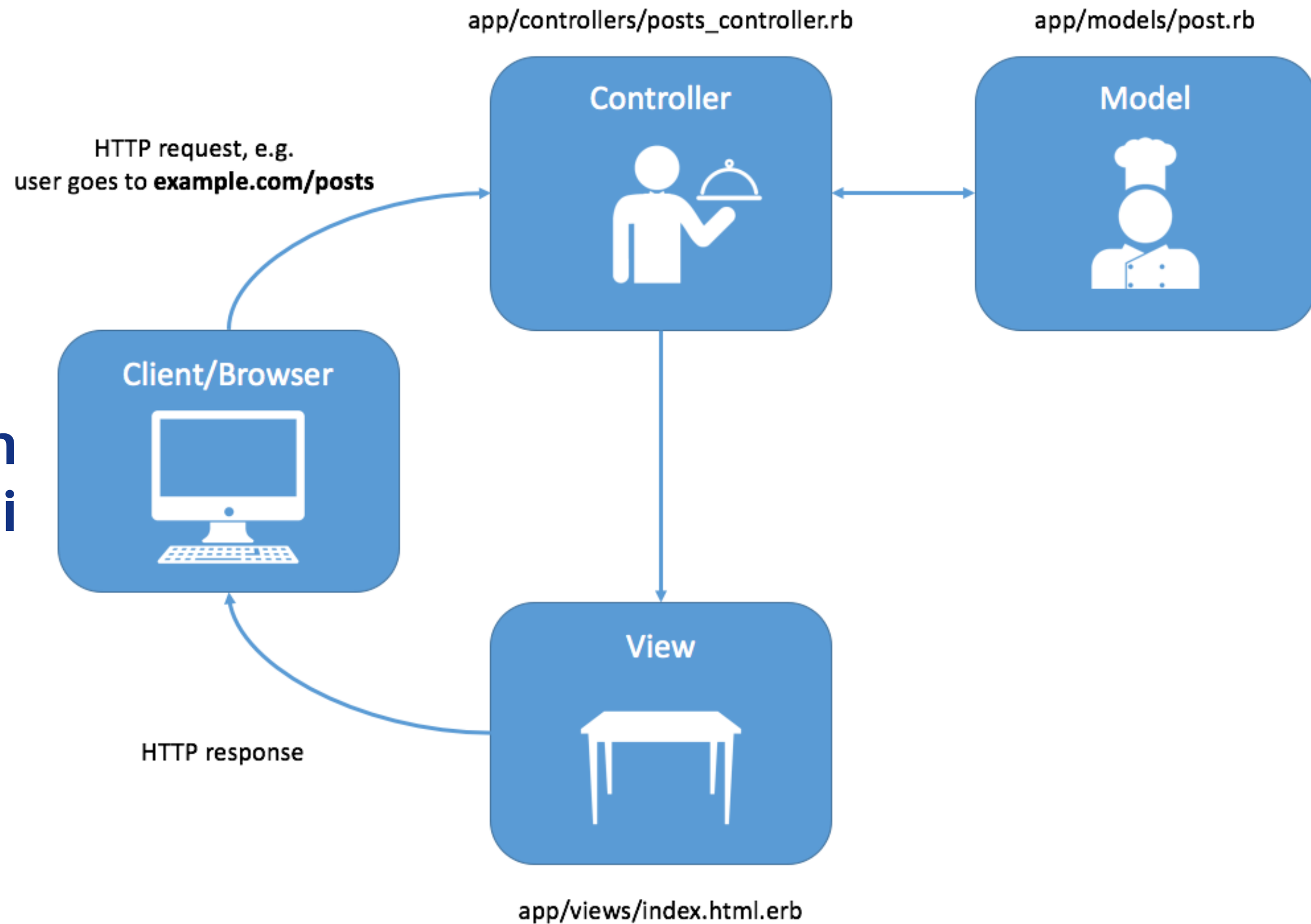
Mengenal cara kerja MVC (Model View Controller).

METODE OOP

Memahami metode pemrograman berbasis Objek atau Object Oriented Programming (OOP) .

KONSEP MVC

MVC adalah teknik atau konsep yang memisahkan komponen utama menjadi tiga bagian yaitu Model, View, dan Controller.

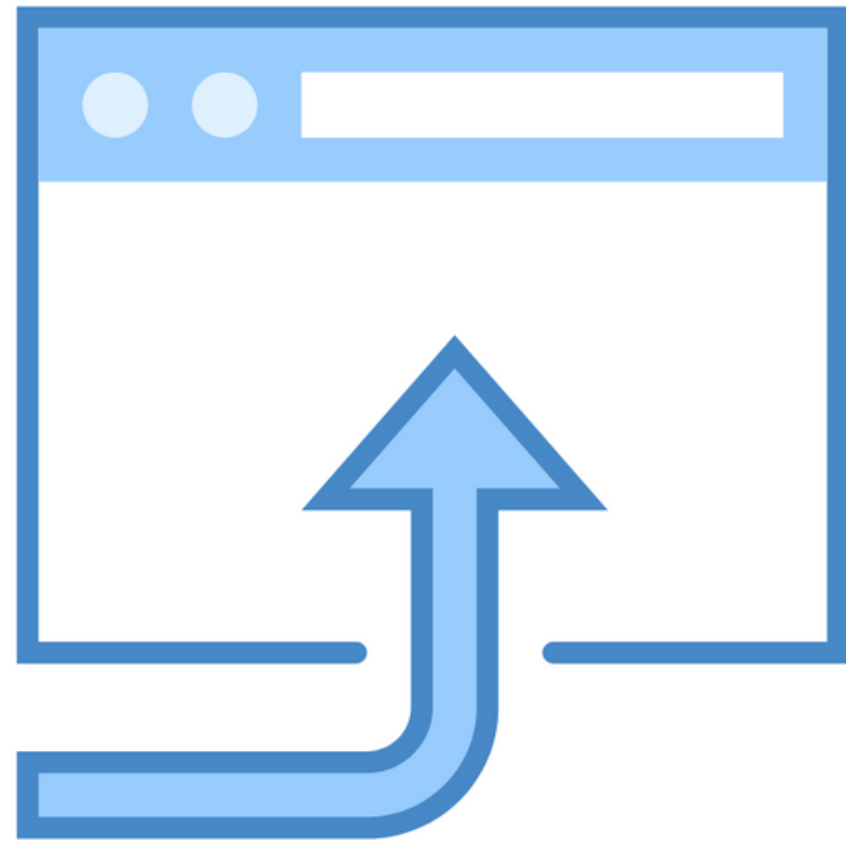


sumber gambar: https://s3.amazonaws.com/flatiron-bucket/readme-lessons/mvc_flow_updated.png



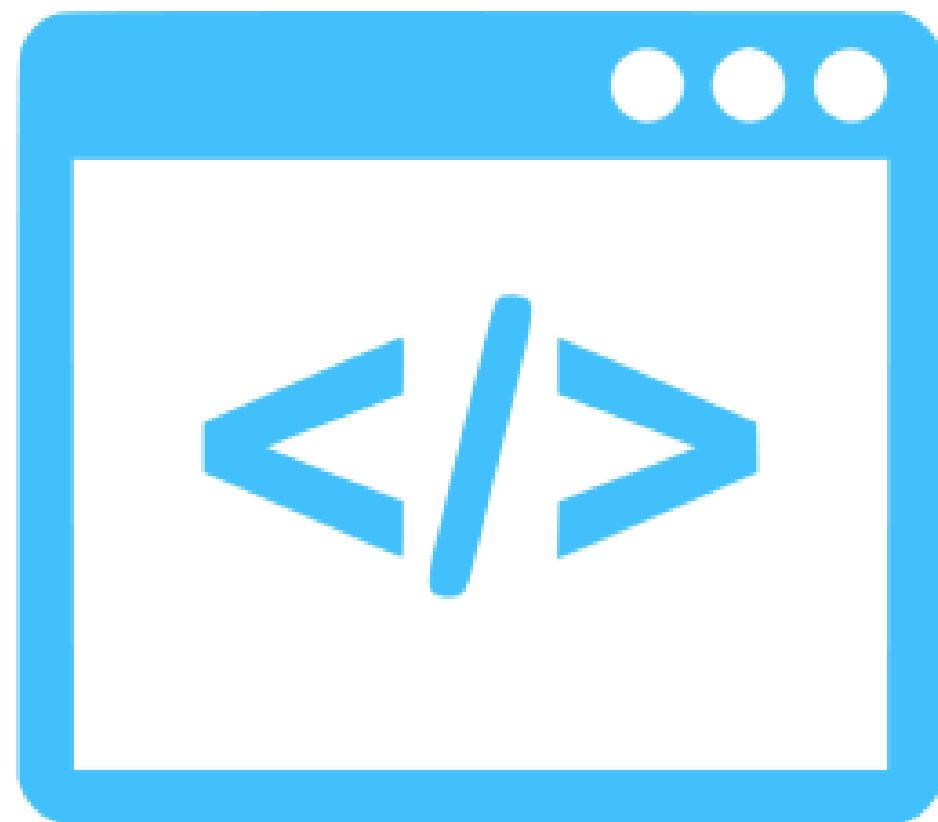
MODEL

Model merupakan bagian penanganan yang berhubungan dengan pengolahan atau manipulasi database seperti mengambil data dari database, menambahkan, mengubah, dan menghapus data.



VIEW

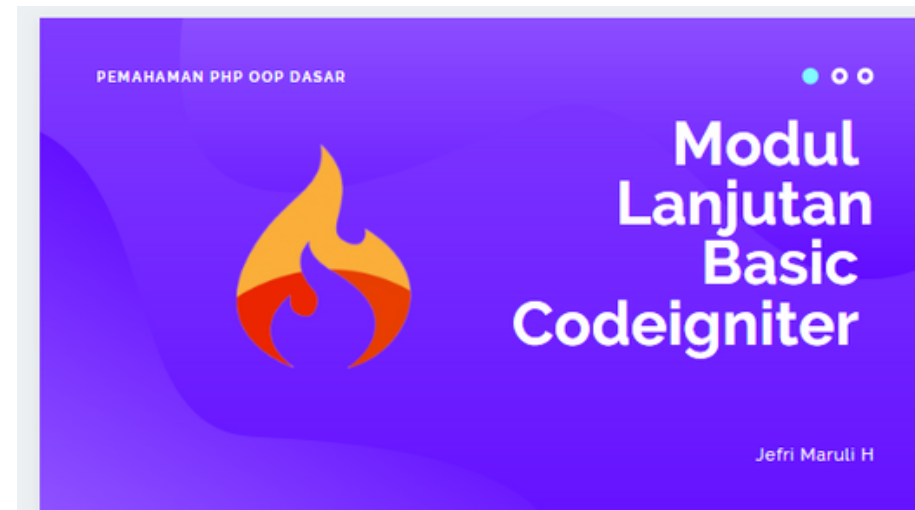
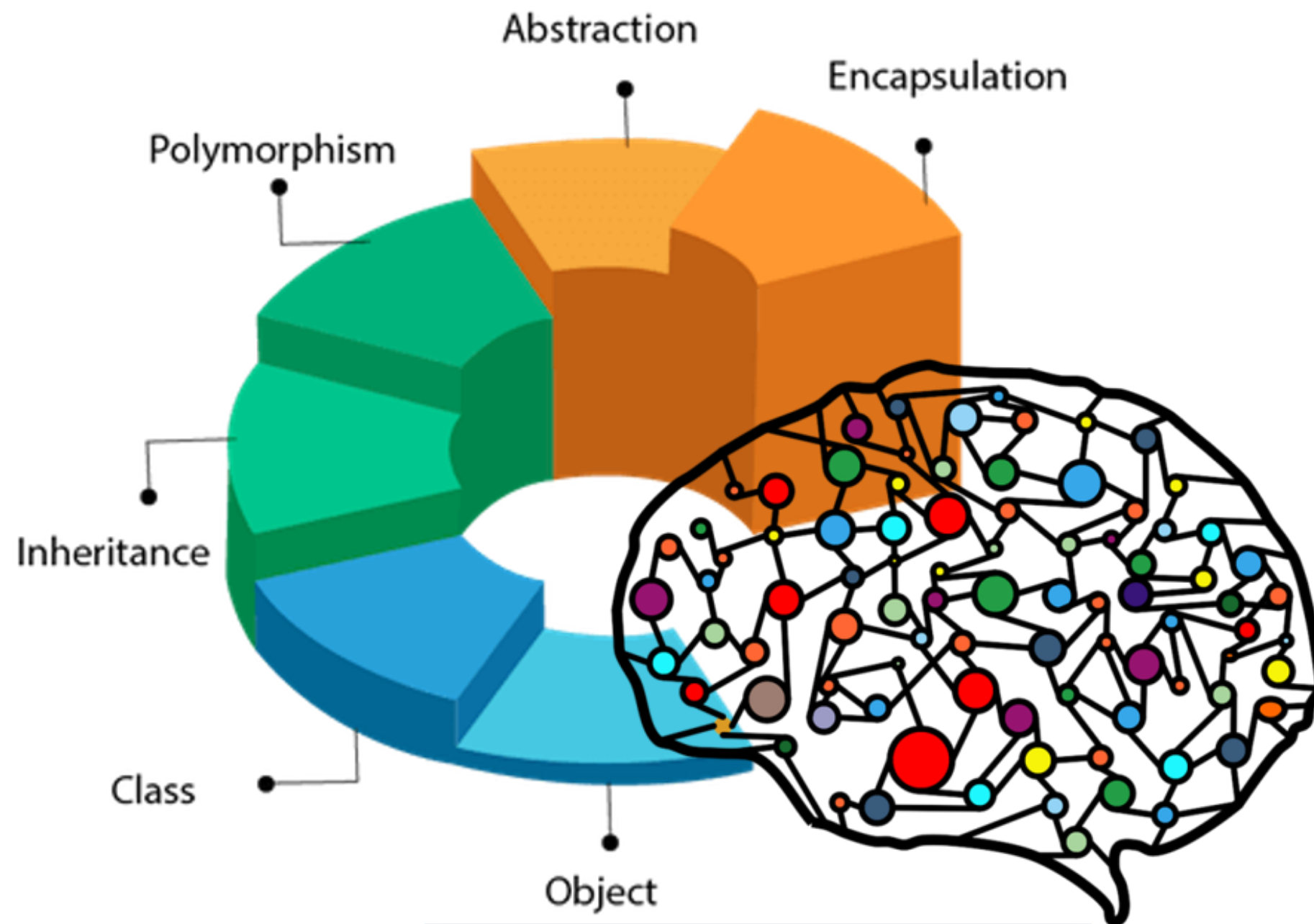
View merupakan bagian yang menangani halaman User Interface (UI) atau halaman yang muncul pada user, sehingga memudahkan Web Designer dalam melakukan pengembangan tampilan pada website.



CONTROLLER

Controller merupakan kumpulan intruksi aksi yang menghubungkan Model dan View atau sebagai jembatan antara Model dan View.

OOPs (Object-Oriented Programming System)

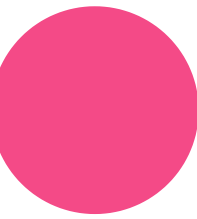


METODE OOP

CodeIgniter menggunakan metode pemrograman OOP (Object Oriented Programming).




Untuk pembahasan tentang OOP silahkan baca modul serial pemahaman OOP dasar.

PERSIAPAN MENGGUNAKAN CODEIGNITER



Perangkat lunak yang harus dipersiapkan untuk menggunakan CodeIgniter, kalian bisa menggunakan salah satu dari kategori dibawah ini:

Keterangan

-  Direkomendasikan buat pemula
-  Untuk pengguna tahap lanjut
-  Opsional

IDE / TEXT EDITOR

Sublime Text, Atom, Brackets, VSCode



WEB SERVER

Xampp, Wampp, Apache, Nginx



DATABASE SERVER

Mysql, MongoDB, NoSQL

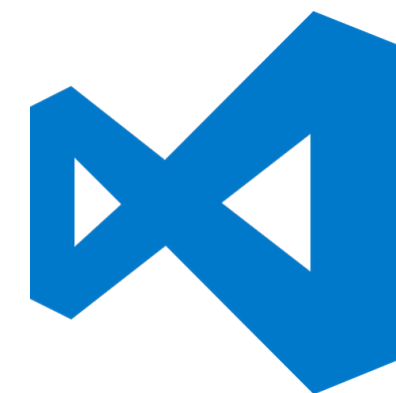
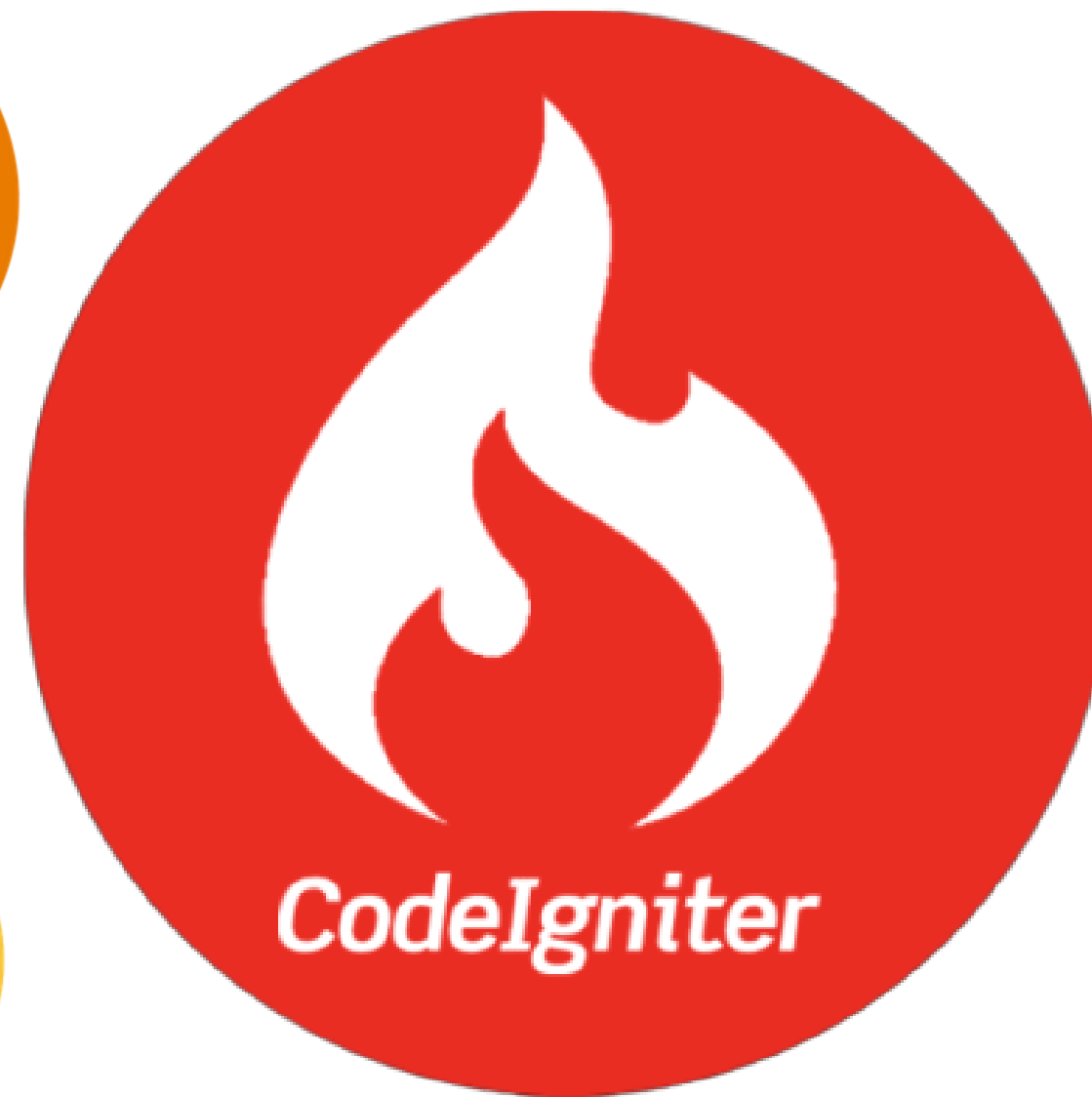


BROWSER

Google Chrome, Mozilla Firefox



PERANGKAT LUNAK YANG PENULIS GUNAKAN





VERSI CODEIGNITER



Jika versi PHP dibawah 5.6 gunakan CodeIgniter versi 2, dan gunakan CodeIgniter versi 3 jika versi PHP diatas 5.6.

CodeIgniter 2

CodeIgniter 2.2.6 is the legacy version of the framework. The 2.x branch was originally released January 2011, and the last version (2.2.6) came out in October, 2015. CodeIgniter 2 has reached its end-of-life for support and updates, as of October 31, 2015. No further updates are planned.

[Download](#)

CodeIgniter 3

CodeIgniter 3.1.11 is the current version of the framework, intended for use with PHP 5.6+. There have been a number of refinements since version 2, notably with the database, session handling and encryption. Development of this version is ongoing.

[Download](#)

[Translations](#)

[Sources](#)

Web server

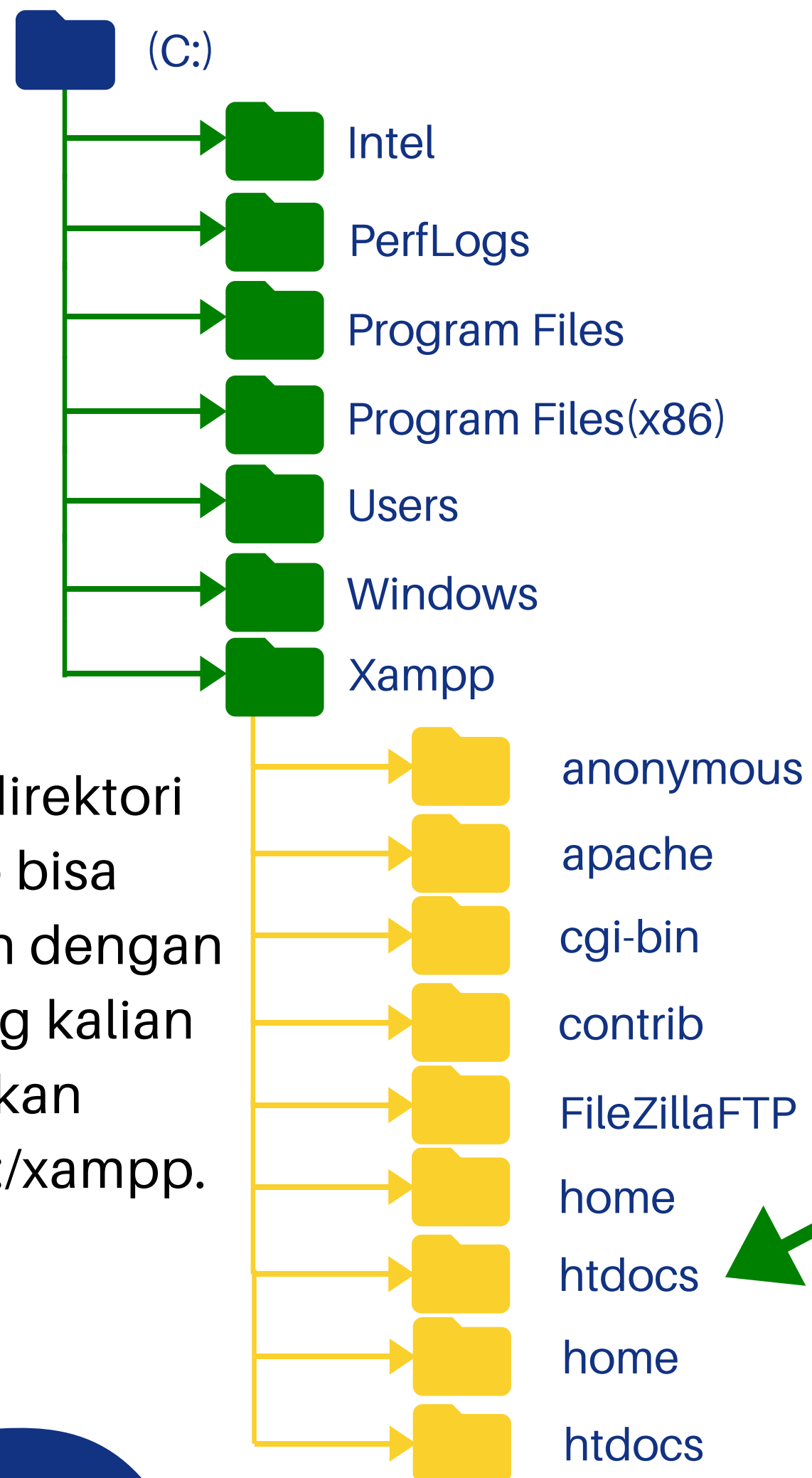
- Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.3.10
- Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - \$Id: 7cc7cc96e675f6d72e5cf0f267f48e167c2abb23 \$
- PHP extension: mysqli [?](#) curl [?](#) mbstring [?](#)
- PHP version: 7.3.10

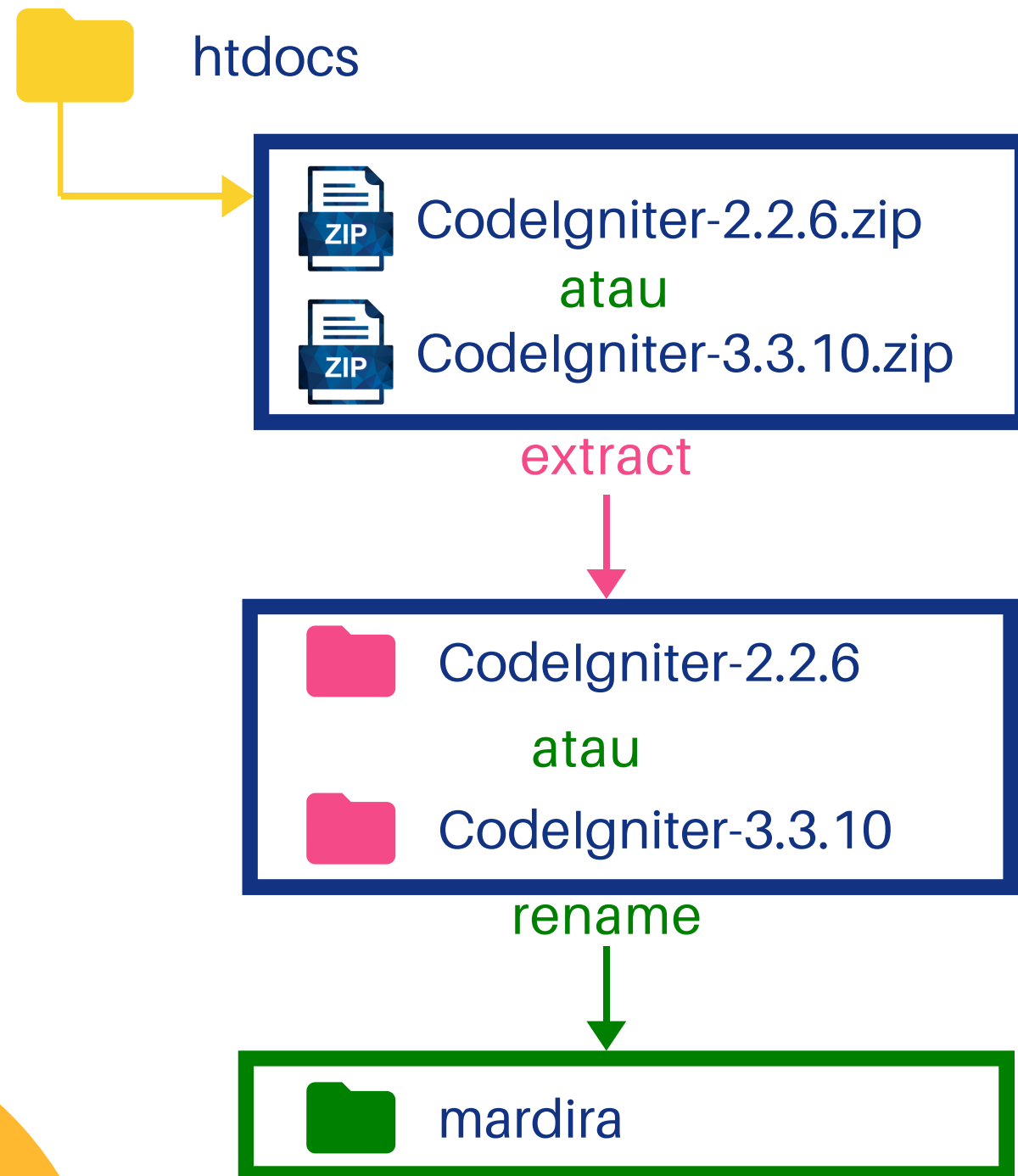
cek versi PHP
akses melalui URL
<localhost/phpmyadmin>

PEMASANGAN CODEIGNITER

Setelah download CodeIgniter sesuai dengan versi PHP , kemudian letakkan file berbentuk zip di folder direktori `c:/xampp/htdocs` seperti petunjuk dibawah ini:

instalasi direktori
Xampp bisa
disesuaikan dengan
folder yang kalian
tentukan
default : c:/xampp.

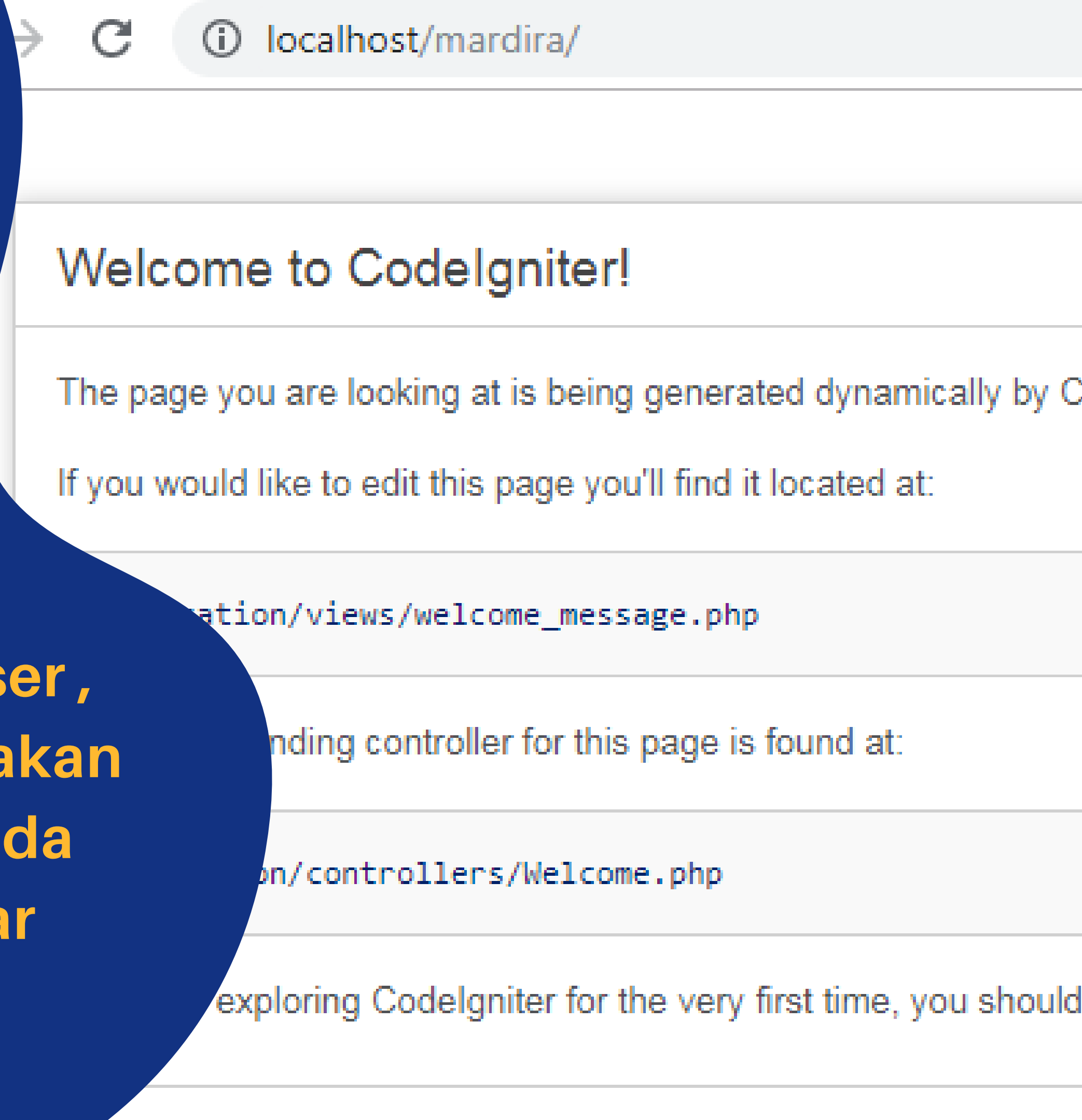




Langkah berikutnya extract file CodeIgniter dan rename Folder menjadi "mardira".

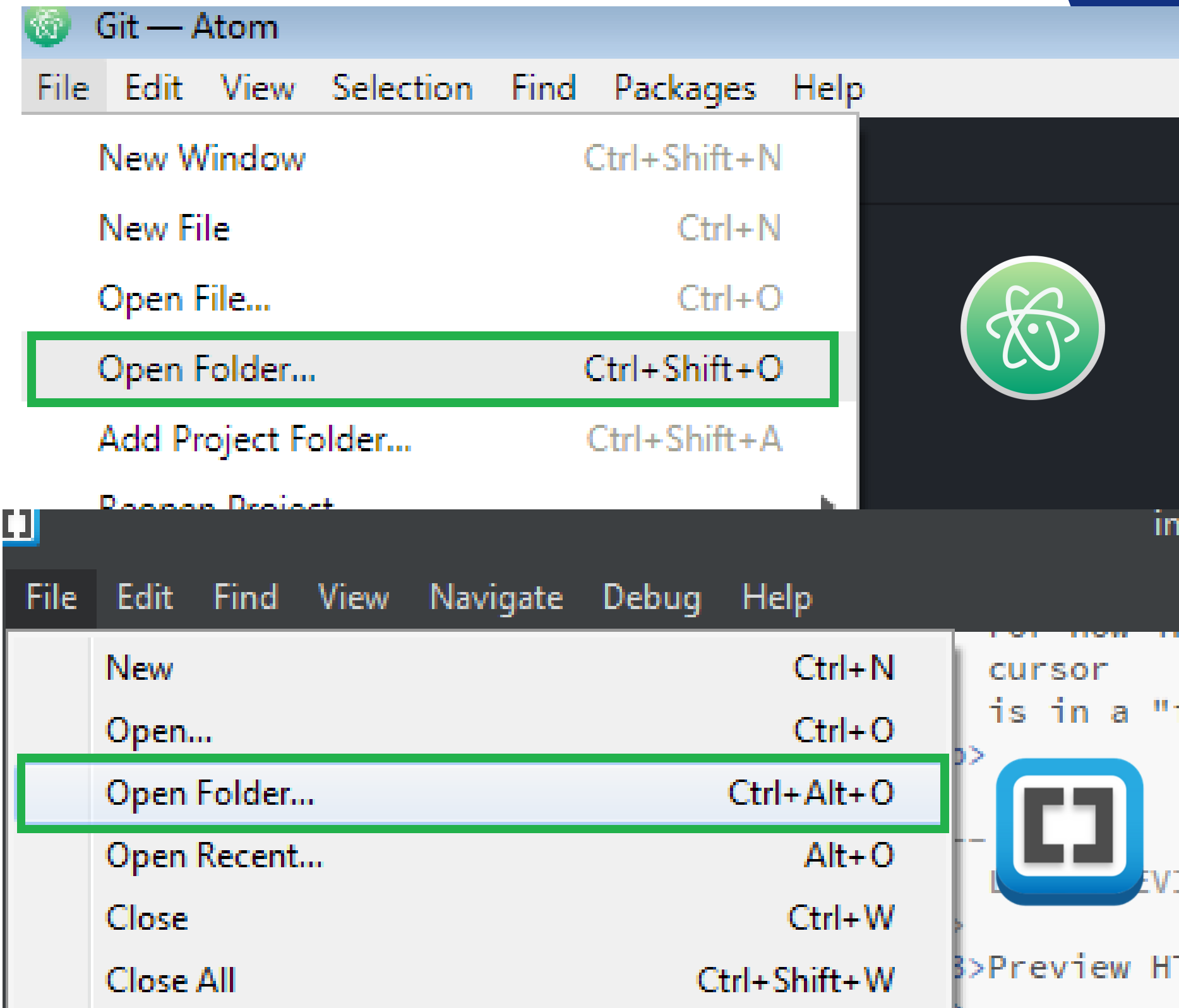
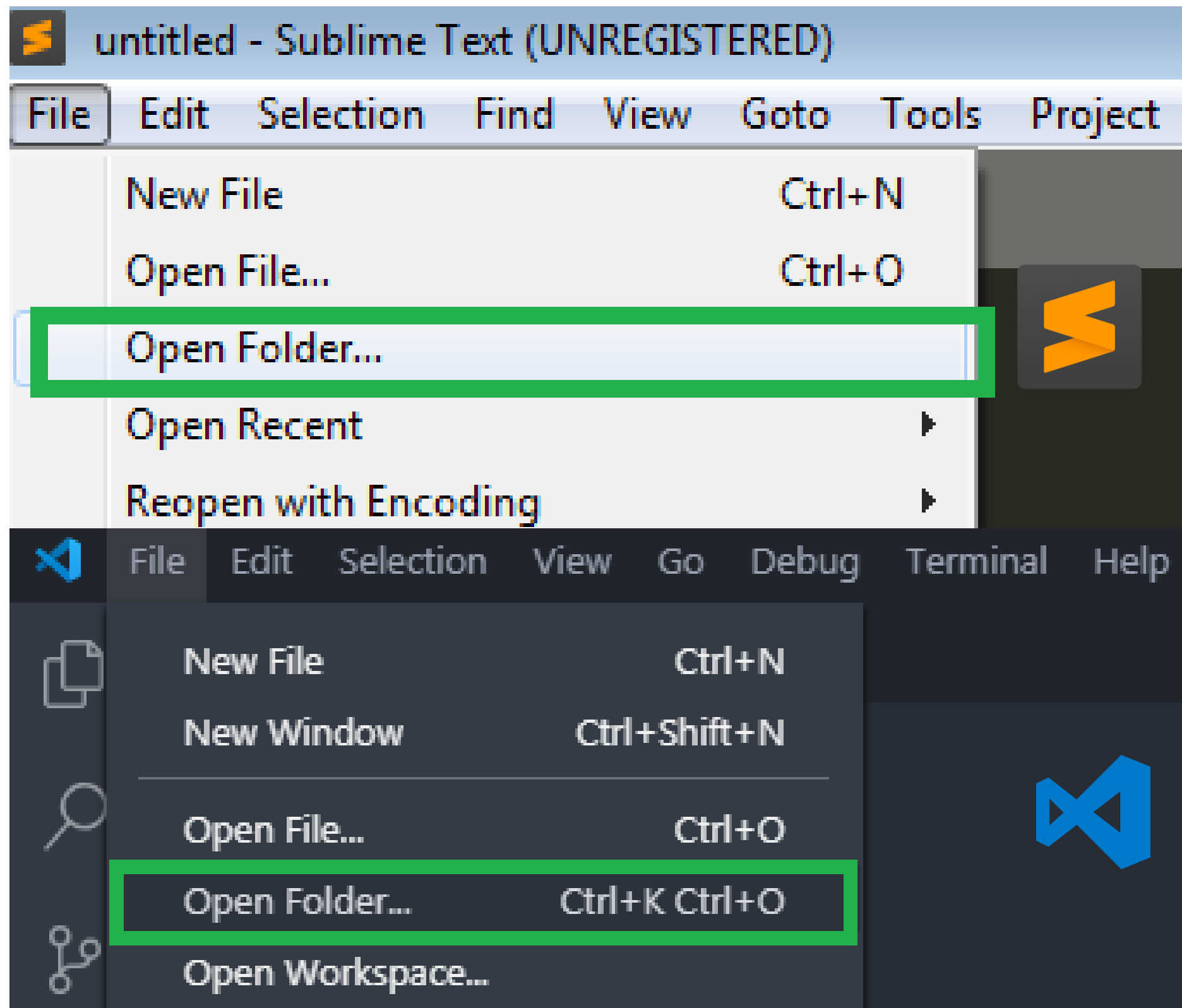
MEMULAI CODEIGNITER

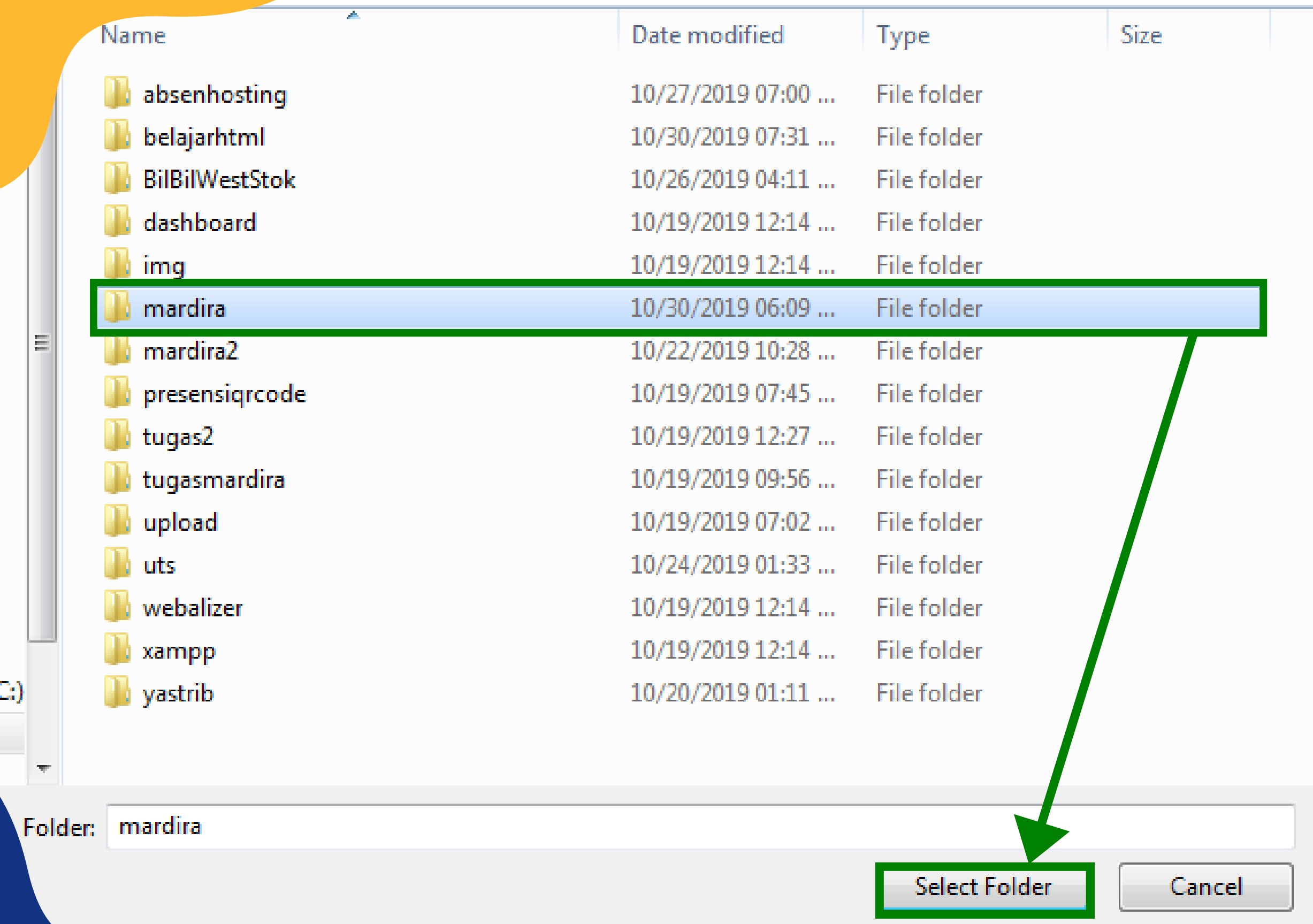
Kemudian akses project
CodeIgniter melalui URL Browser,
ketik "localhost/mardira" maka akan
muncul hasil tampilan awal pada
CodeIgniter seperti di gambar



MENGGUNAKAN IDE / TEXT EDITOR

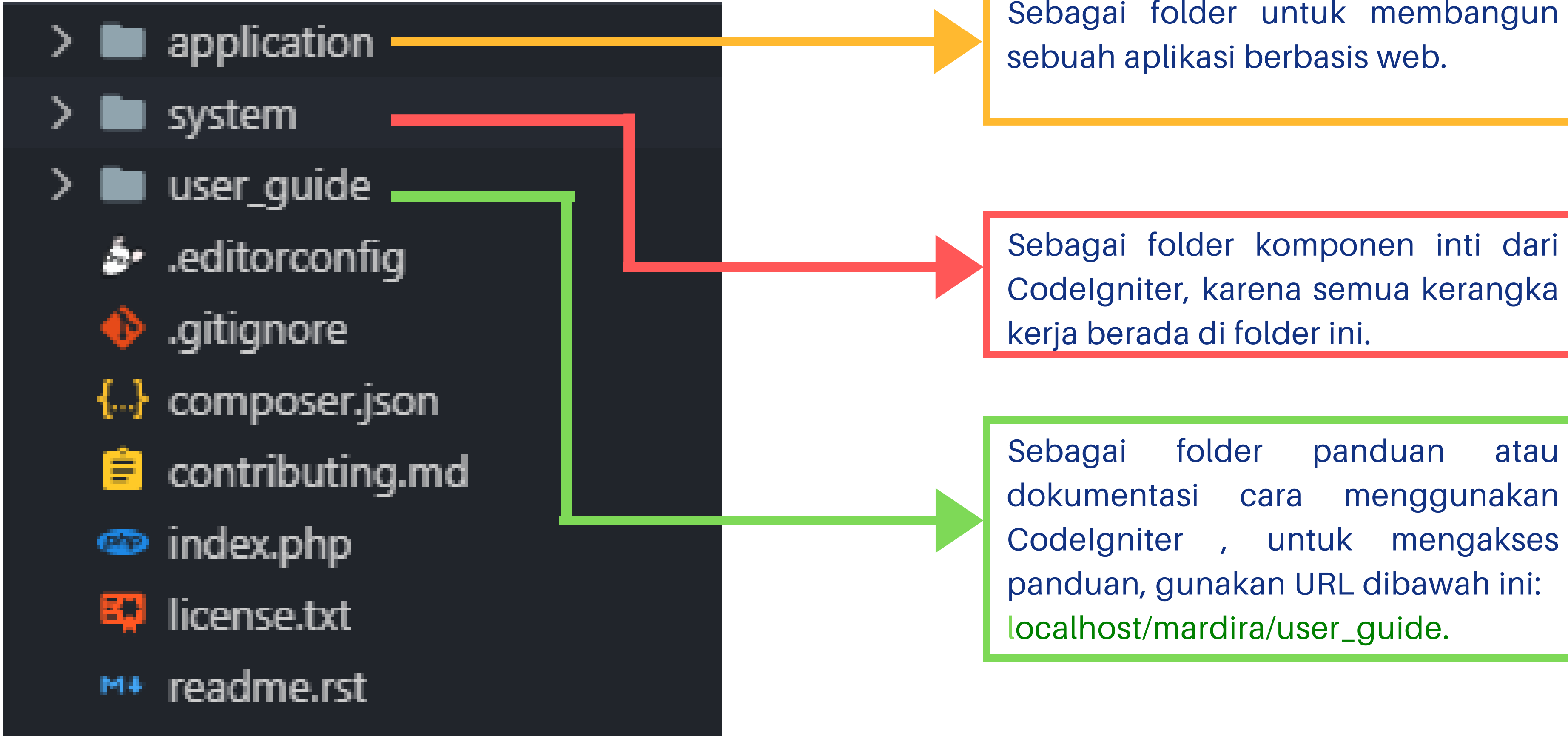
Buka aplikasi text editor sesuai dengan pilihan masing-masing, kemudian pilih "File" -> "Open Folder".





Pilih folder project CodeIgniter "**mardira**", kemudian pilih tombol **Select Folder**.

PENGENALAN STRUKTUR FOLDER CODEIGNITER



The diagram shows a file explorer view of the CodeIgniter project structure. Three folders are highlighted with colored lines and arrows pointing to descriptive text boxes:

- application** (orange line and arrow)
- system** (red line and arrow)
- user_guide** (green line and arrow)

Other files visible in the explorer include:

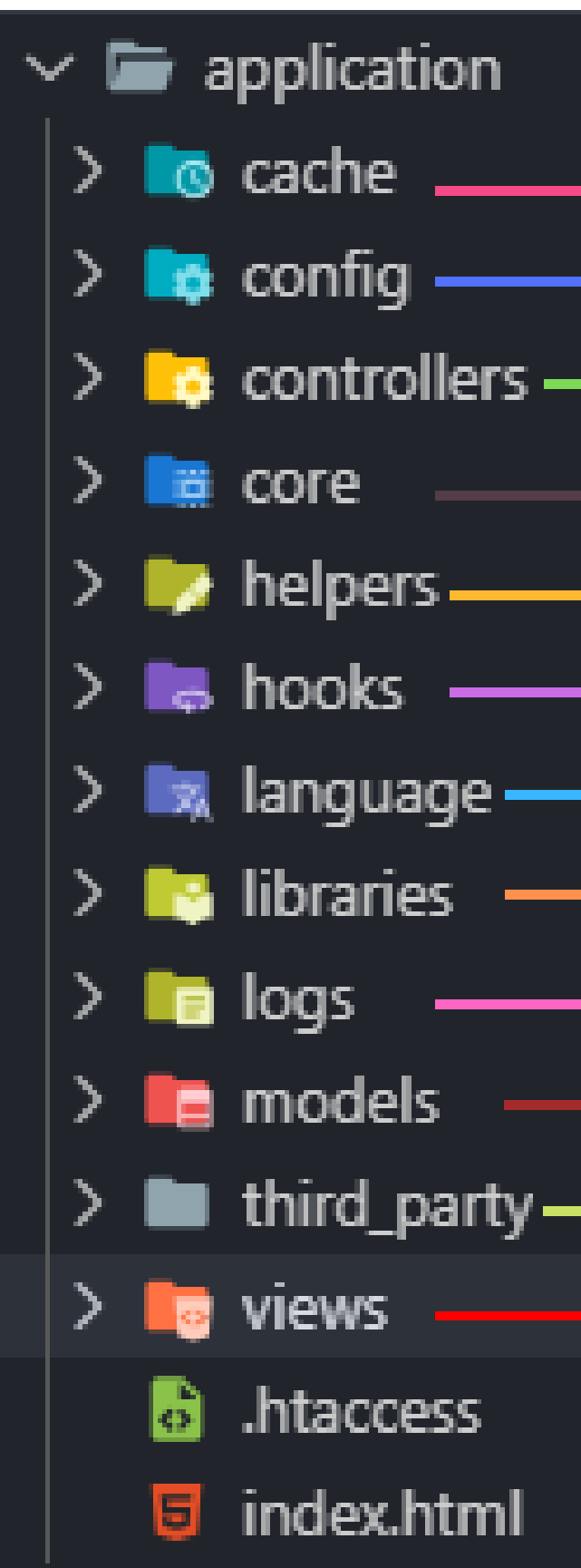
- `.editorconfig`
- `.gitignore`
- `composer.json`
- `contributing.md`
- `index.php`
- `license.txt`
- `readme.rst`

Sebagai folder untuk membangun sebuah aplikasi berbasis web.

Sebagai folder komponen inti dari CodeIgniter, karena semua kerangka kerja berada di folder ini.

Sebagai folder panduan atau dokumentasi cara menggunakan CodeIgniter , untuk mengakses panduan, gunakan URL dibawah ini:
localhost/mardira/user_guide.

PENGENALAN STRUKTUR FOLDER APPLICATION



Tempat menyimpan file cache untuk melakukan Caching atau meningkatkan kecepatan pemuatan halaman jika metode ini dipakai.

Tempat menyimpan semua file konfigurasi yang ada di dalam aplikasi, mulai dari database, router, dan autoload aplikasi.

Tempat menyimpan semua file Controller.

Tempat menyimpan file yang bersifat inti dari CodeIgniter.

Tempat menyimpan file helper, dimana file tersebut berisi beberapa fungsi yang dapat mengatur logika pemrograman untuk membantu Controller.

Tempat menyimpan file hook, yang berguna untuk mengubah alur dari inti atau core CodeIgniter.

Tempat menyimpan file bahasa yang akan digunakan.

Tempat menyimpan file custom library, yang berguna agar developer memungkinkan untuk menggunakan kembali baris programnya.

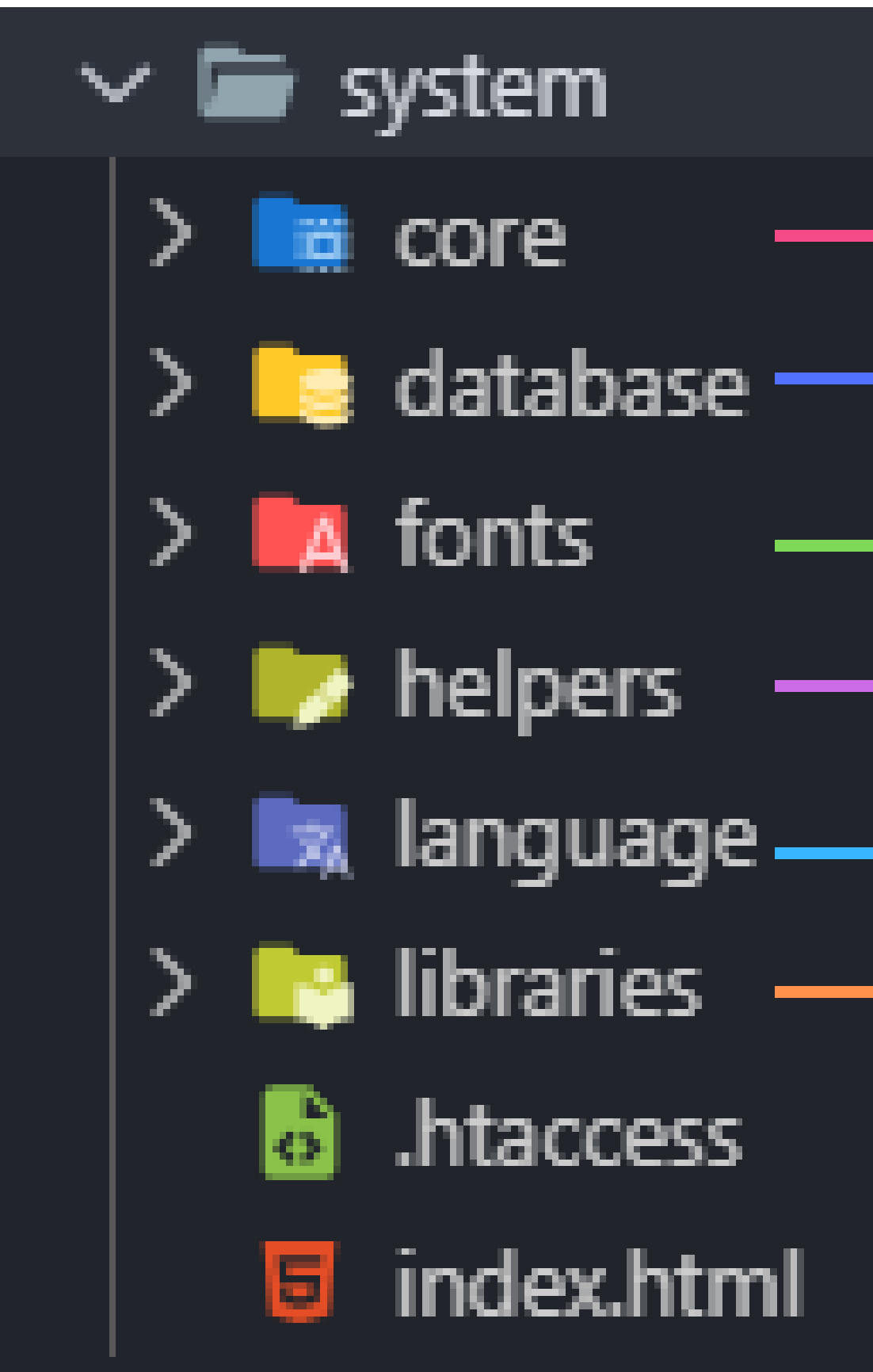
Tempat menyimpan file log (catatan tentang peristiwa users saat beraktivitas pada sistem).

Berisi file yang berfungsi untuk menangani semua fungsi yang berhubungan dengan manipulasi database.

Berisi folder atau file yang berasal dari pihak ketiga, yang berfungsi untuk menambah fungsionalitas yang lebih besar pada sistem.

Berisi folder atau file yang berhubungan dengan tampilan website yang akan ditujukan kepada user.

PENGENALAN STRUKTUR FOLDER SYSTEM



Folder yang merupakan wadah untuk menyimpan kumpulan file-file inti, jadi karena di system merupakan bawaan dari CodeIgniternya

Berisi folder dan file database drivers, berfungsi untuk mengintegrasikan sistem dengan jenis database.

Tempat menyimpan semua font yang digunakan image manipulation library.

Berisi kumpulan data helper yang mungkin akan dipanggil fungsinya oleh user.

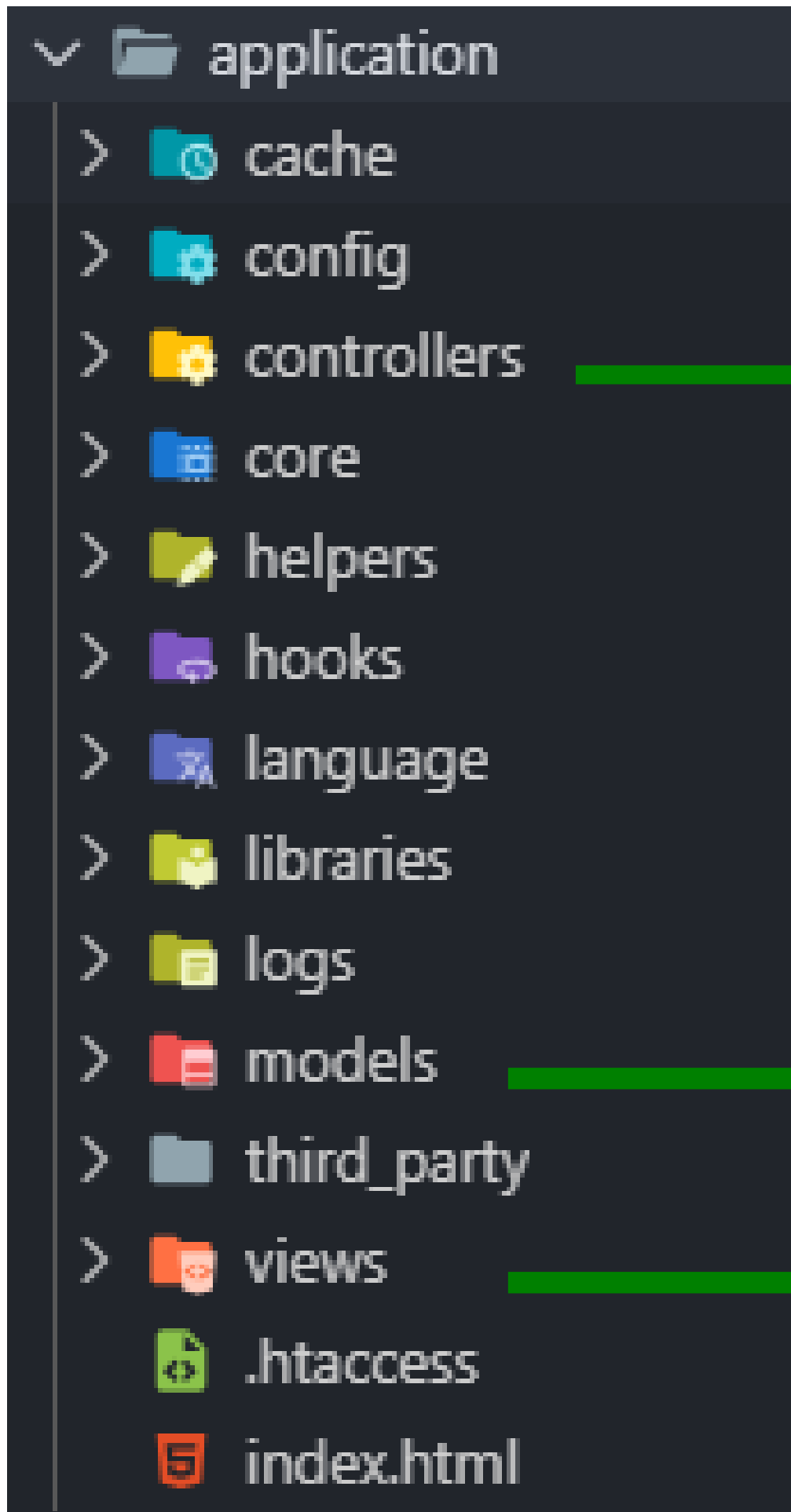
Berisi folder english sebagai penggunaan bahasanya, dimana merupakan bawaan dari CI tersebut.

Berisi kumpulan folder dan file yang berguna jika sewaktu-waktu pihak developer akan menggunakan library bawaan tersebut.

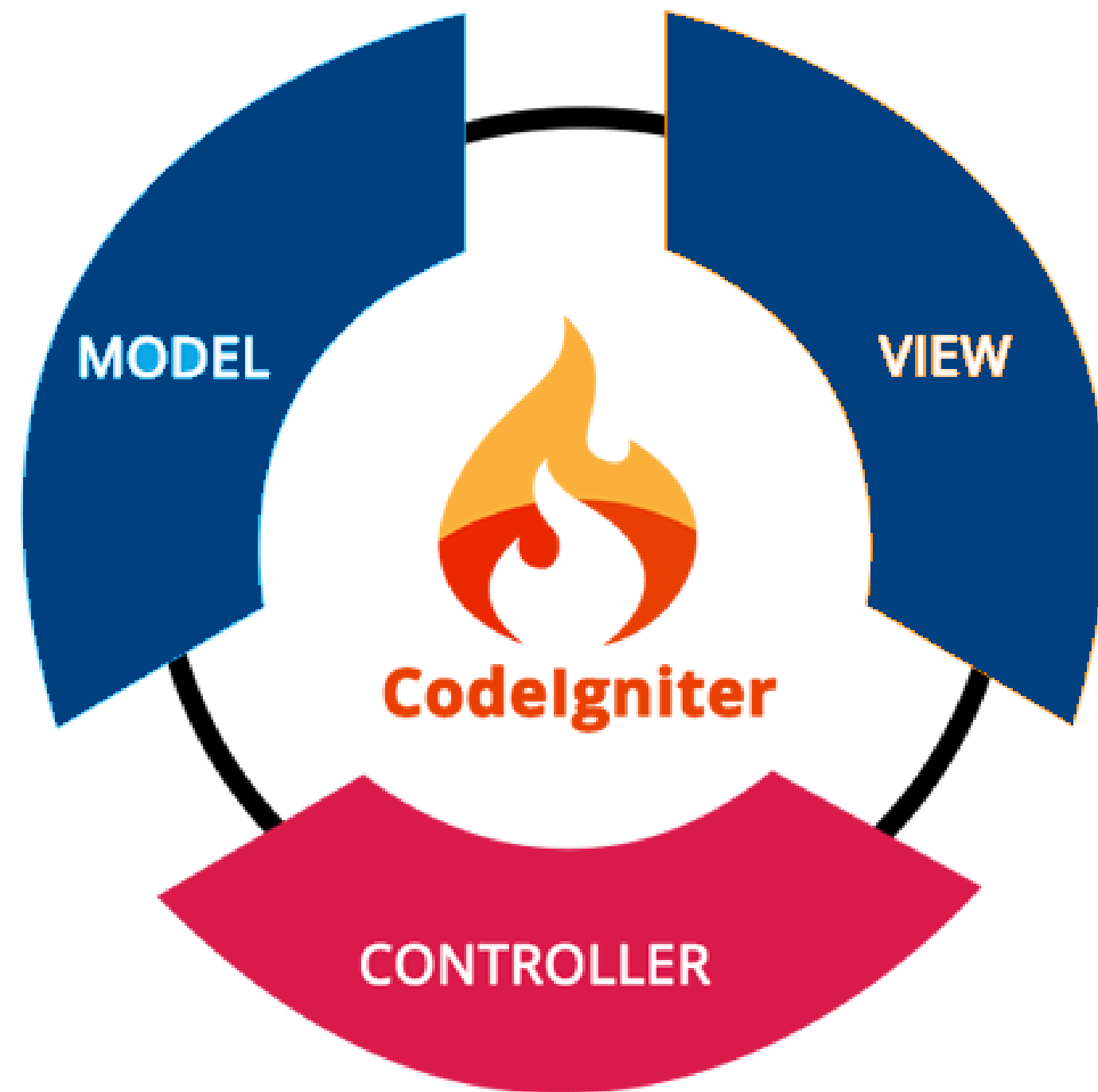
MEMULAI PERANCANGAN APLIKASI

Setelah mengenal konsep dasar dari pola desain MVC , hal yang diutamakan dalam membuat sebuah aplikasi yaitu :

- **Membuat Controller**
- **Membuat Model**
- **Membuat View**



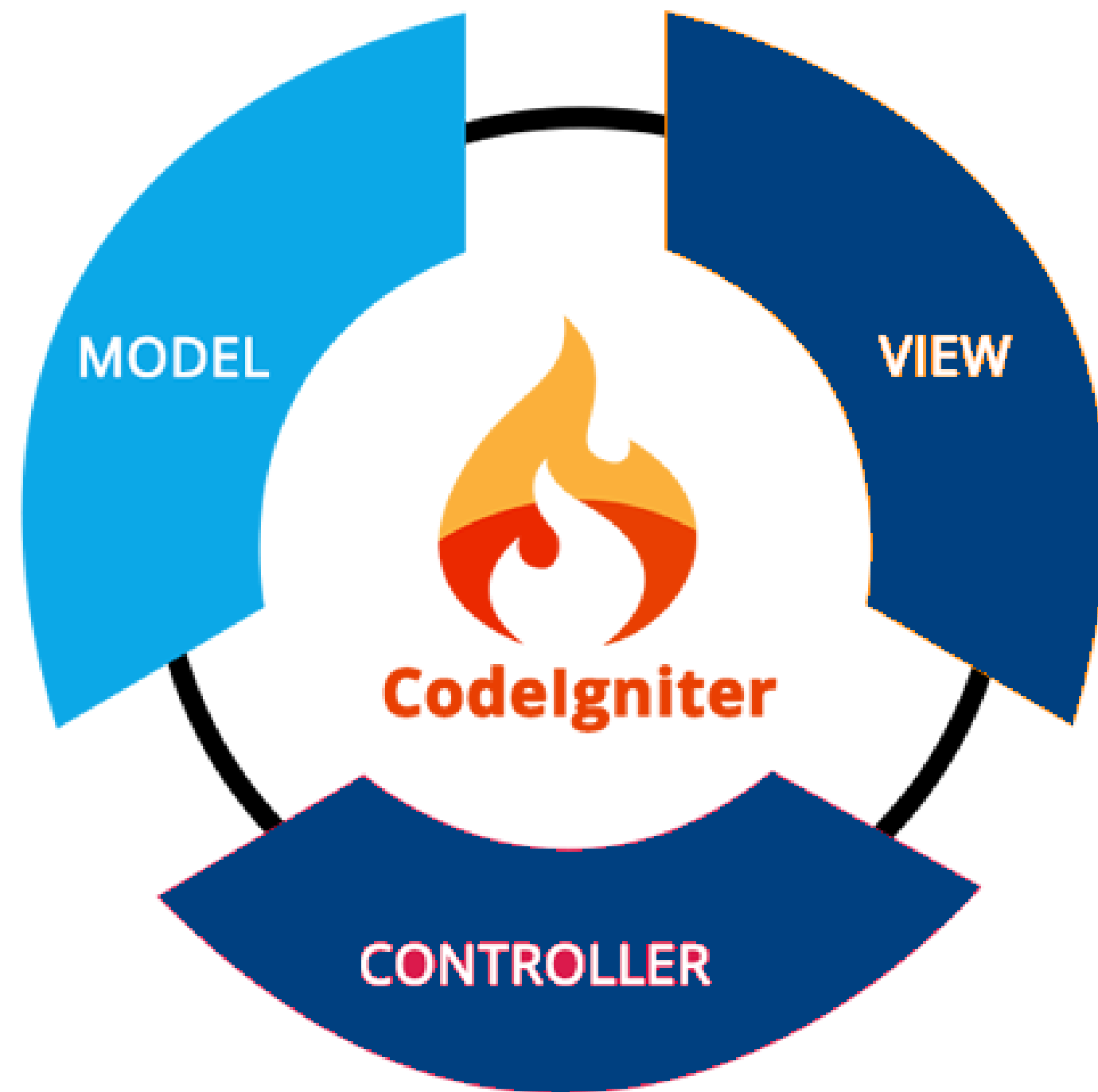
MEMBUAT CONTROLLER



Hal yang harus diperhatikan untuk membuat sebuah Controller

- Semua penyimpanan file Controller berada di direktori `Application/controllers`
- Huruf awalan sebuah file Controller harus menggunakan huruf kapital
- Mendeklarasi sebuah Class juga harus menggunakan huruf kapital dan sesuai dengan nama file Controller yang telah dibuat

MEMBUAT MODEL

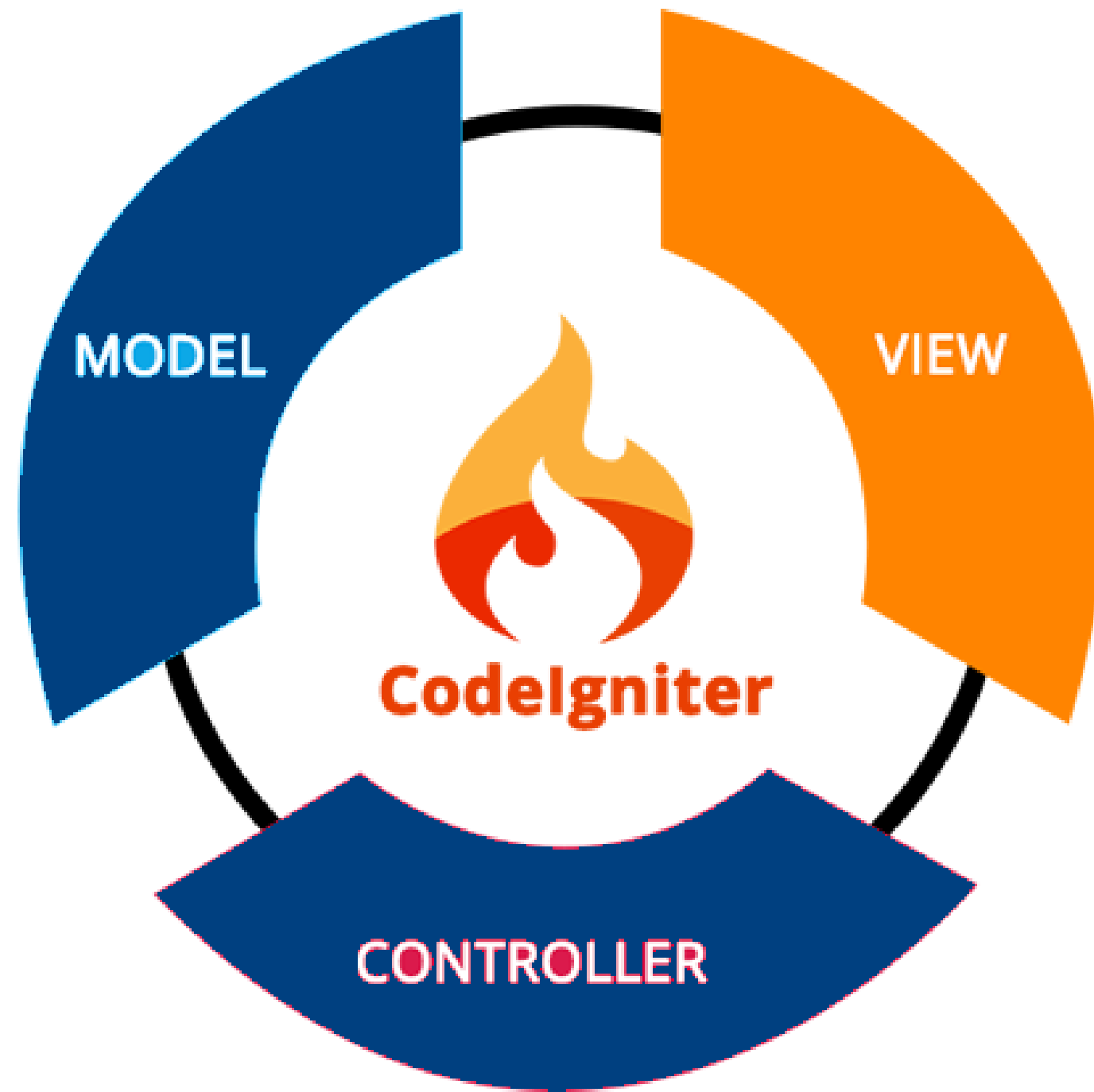


Hal yang harus diperhatikan untuk membuat sebuah Model

- Semua penyimpanan file Model berada di direktori Application/models
- Huruf awalan sebuah file Model harus menggunakan huruf kapital
- Sama seperti membuat Controller, mendeklasi sebuah Class menggunakan huruf awalan kapital dan diikuti dengan kata '_model'

contoh : Hello_model

MEMBUAT VIEW



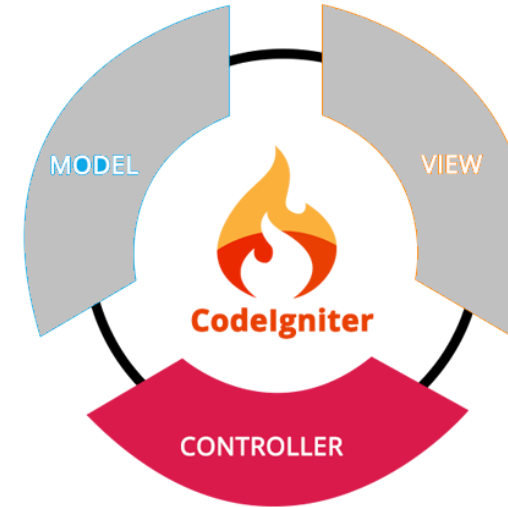
Hal yang harus diperhatikan untuk membuat sebuah View

- Semua penyimpanan file View berada di direktori `Application/views`
- View tidak dapat dipanggil langsung melalui browser tanpa melalui Controller, intinya Controller harus dibuat terlebih dahulu jika ingin menggunakan sebuah View
- Pembuatan file view diawal atau diakhir dengan kata 'view' atau 'v' menggunakan huruf kecil

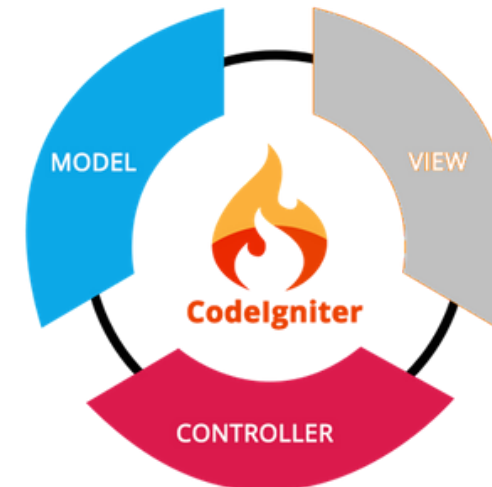
Contoh: `hello_view` , `v_hello`

IMPLEMENTASI APLIKASI "HELLO MARDIRA"

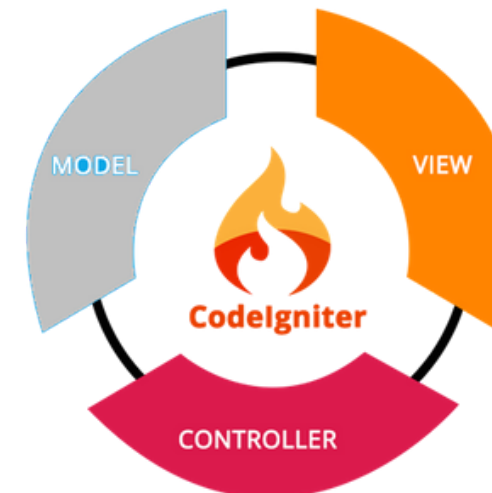
Untuk memahami peran dari setiap bagian di dalam pola desain MVC, aplikasi akan dibuat dalam empat versi yaitu:



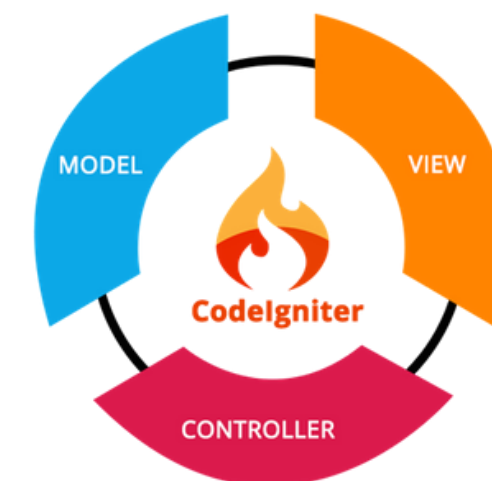
Aplikasi yang hanya menggunakan Controller



Aplikasi yang menggunakan Controller dan Model



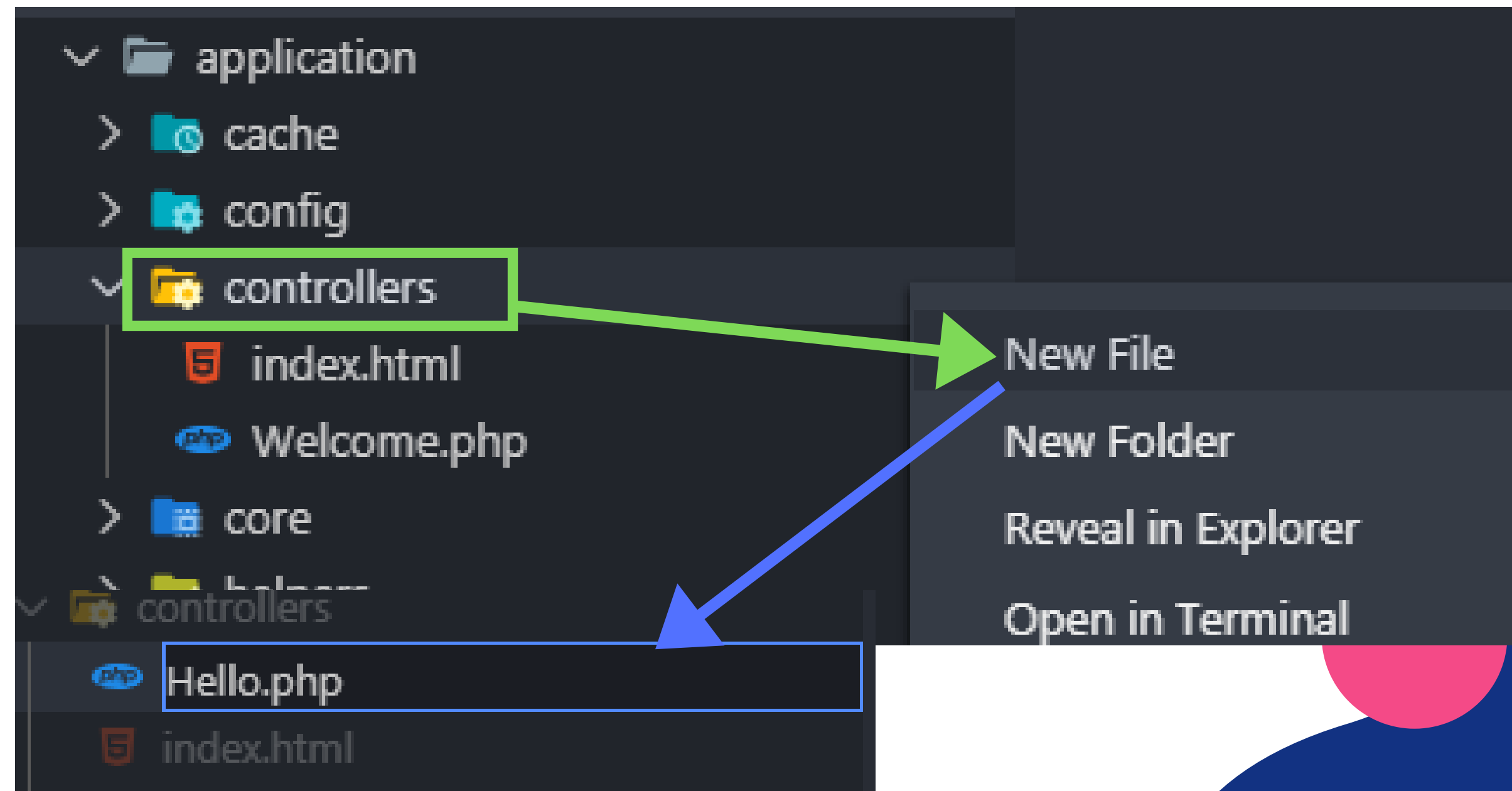
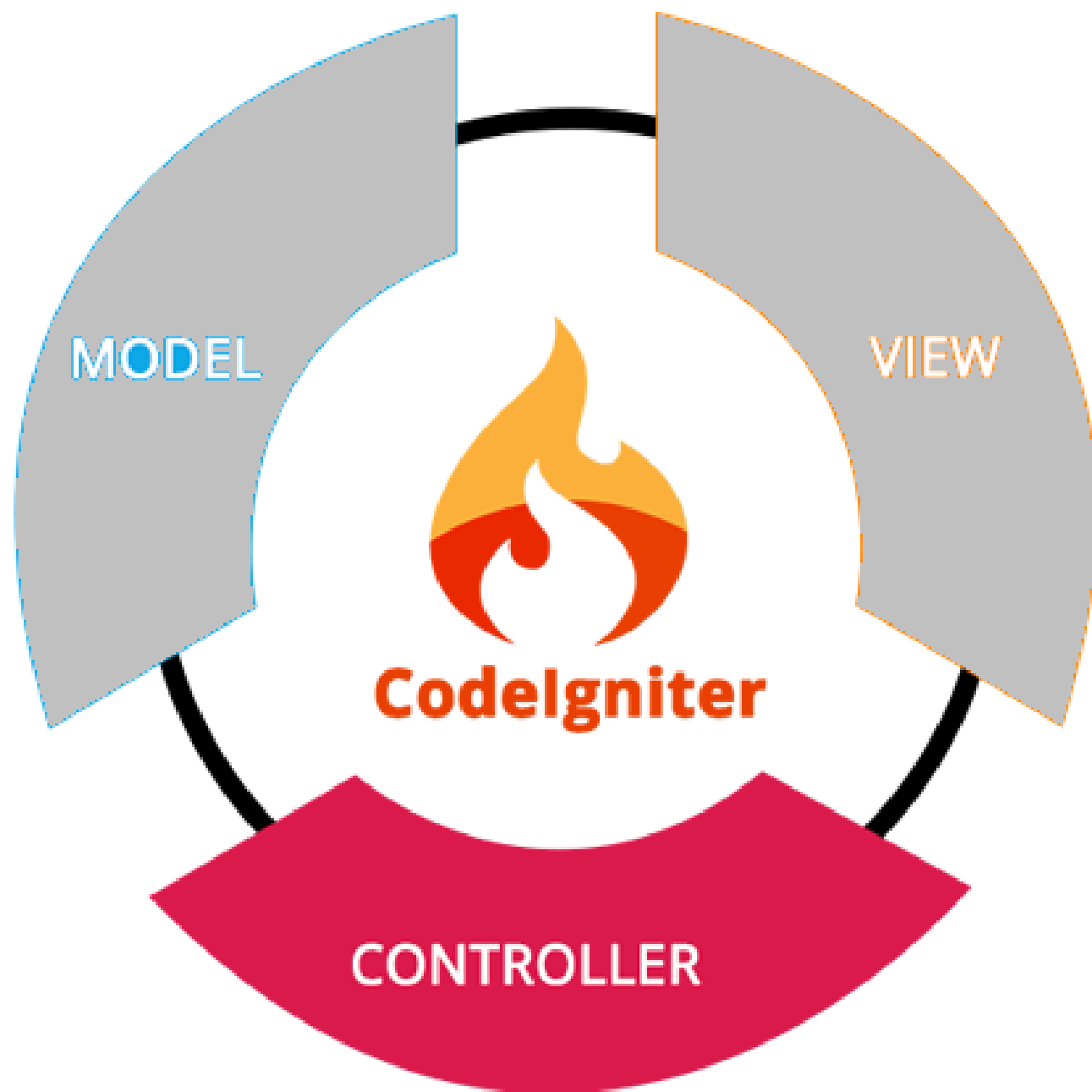
Aplikasi yang menggunakan Controller dan View



Aplikasi yang menggunakan Controller, Model, dan View

APLIKASI YANG HANYA MENGGUNAKAN CONTROLLER

Untuk membuat Controller baru, klik kanan pada folders controllers kemudian pilih 'New File' setelah itu membuat file dengan nama "**Hello.php**".



Lalu tuliskan kode berikut :



```
1  <?php
2
3  class Hello extends CI_Controller
4  {
5      public function index()
6      {
7          echo "Hello Mardira";
8      }
9  }
10
```



CATATAN

```
<?php
class Hello extends CI_Controller
{
    public function index()
    {
        echo "Hello Mardira";
    }
}
```

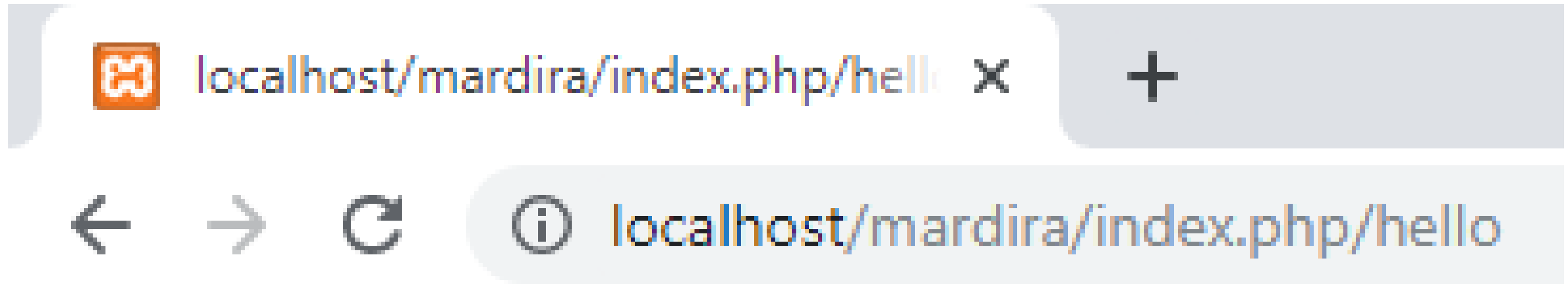
Metode/Function **index()** di dalam Class **Hello** merepresentasikan aksi default yang akan dieksekusi ketika controller **Hello** dipanggil tanpa menyertakan nama metode .

Kemudian jalankan Controller dengan menuliskan URL berikut:

`http://localhost/mardira/index.php/hello`

atau

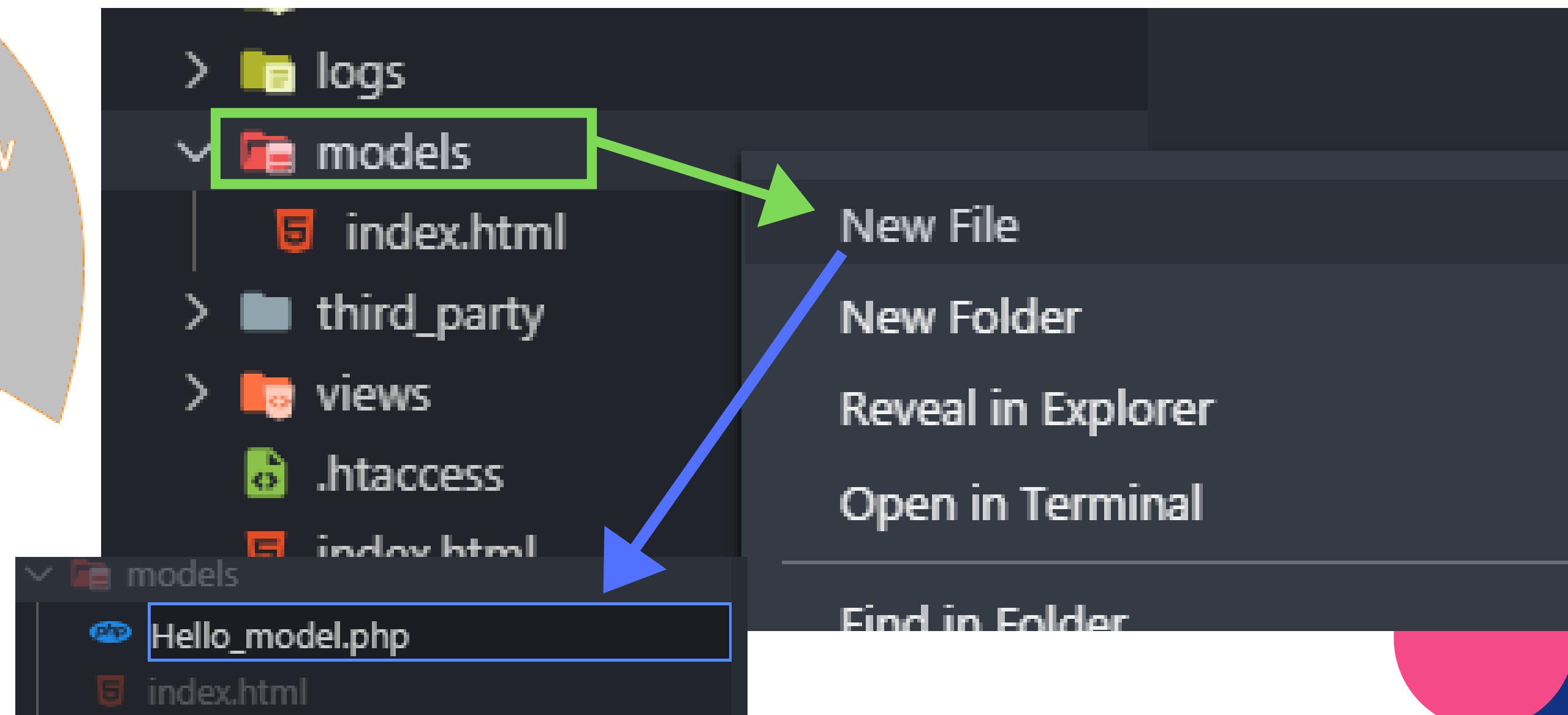
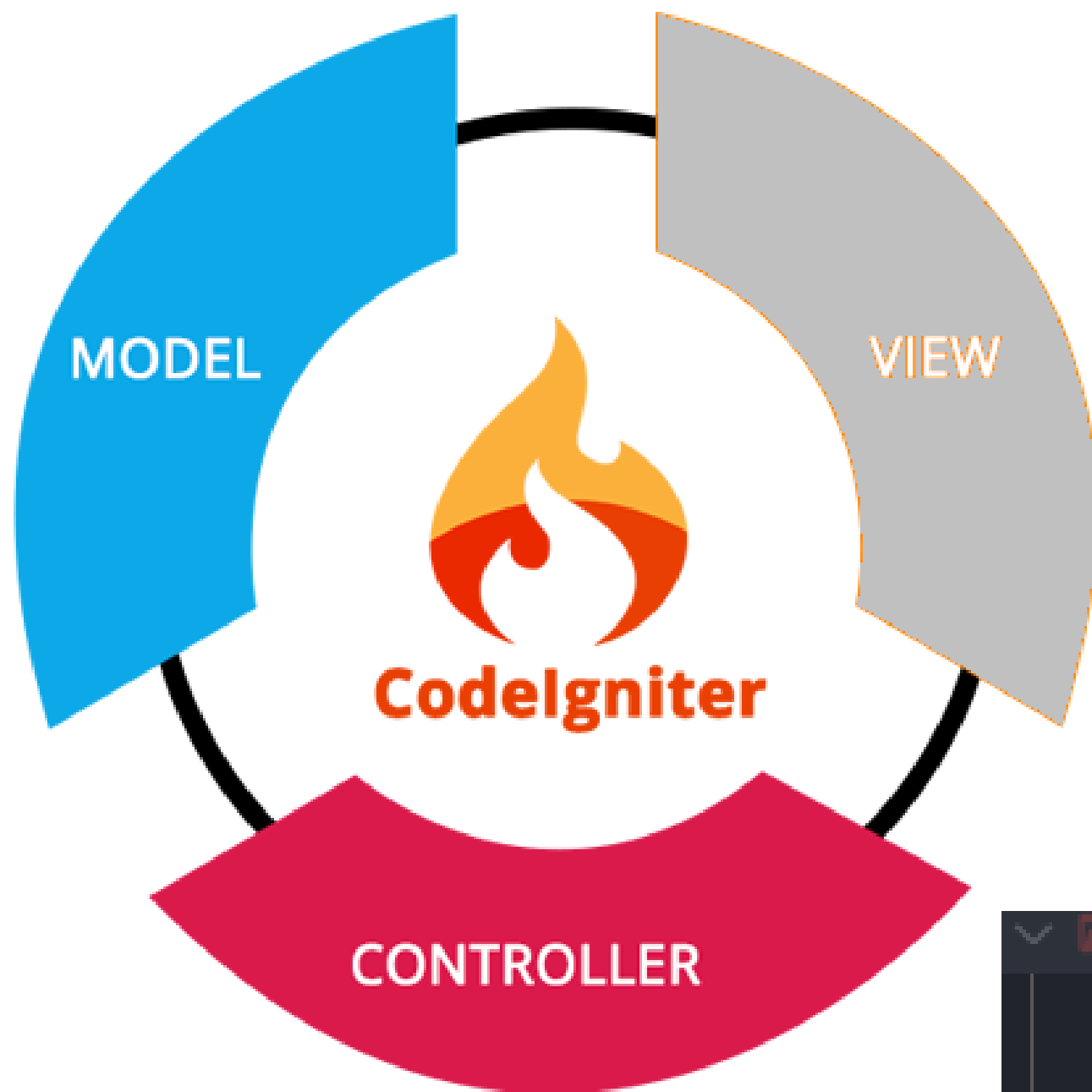
`http://localhost/mardira/index.php/hello/index`



Hello Mardira

APLIKASI YANG MENGGUNAKAN CONTROLLER DAN MODEL

Langkah selanjutnya yaitu membuat Model baru, dengan cara klik kanan pada folders models kemudian pilih 'New File' setelah itu membuat file dengan nama "**Hello_model.php**".



kemudian tuliskan kode berikut:



```
1  <?php
2
3  class Hello_model extends CI_Model
4  {
5
6      public function hello_mardira()
7      {
8          return "Hello Mardira";
9      }
10 }
11
```

Kemudian di file Controller tambahkan kode berikut

```
<?php
```

```
class Hello extends CI_Controller
{
    public function index()
    {
        echo "Hello Mardira";
    }
}
```

```
public function hello_model()
{
    $this->load->model('Hello_model');
    $data = $this->Hello_model->hello_mardira();
    echo $data;
}
```

```
}
```



CATATAN

```
<?php
```

```
class Hello extends CI_Controller  
{  
    public function index()  
    {  
        echo "Hello Mardira";  
    }  
}
```

```
public function hello_model()  
{  
    $this->load->model('Hello_model');  
    $data = $this->Hello_model->hello_mardira();  
    echo $data;  
}
```

Untuk menggunakan sebuah Model di Controller, Model tersebut harus dimuat terlebih dahulu melalui Controller dengan baris kode:

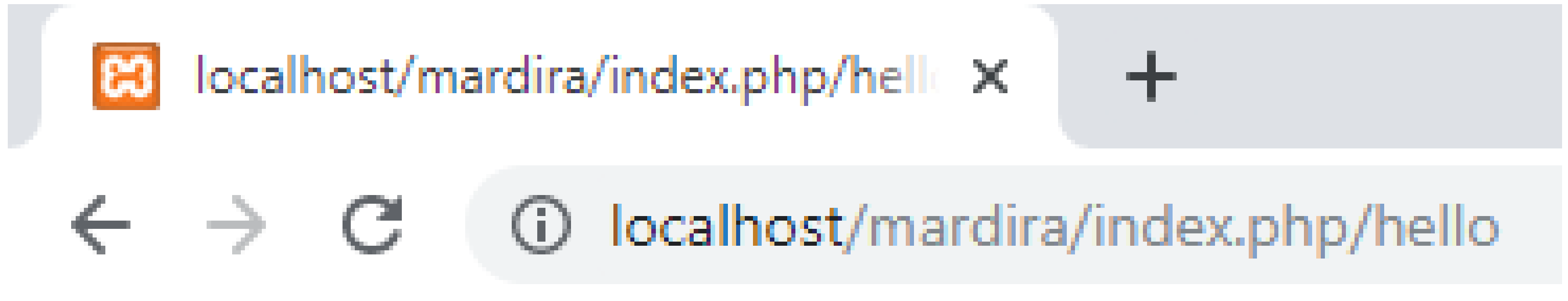
```
$this->load->model('Nama_model');
```

setelah pemuatan Model, mengakses function/metode di Model akan dipanggil melalui Controller dengan baris kode:

```
$this->Nama_model->nama_fungsi();
```

Kemudian jalankan Aplikasi dengan menuliskan URL sebagai berikut:

http://localhost/mardira/index.php/hello/hello_model



Hello Mardira

```
<?php

class Hello extends CI_Controller
{
    public function index()
    {
        echo "Hello Mardira";
    }

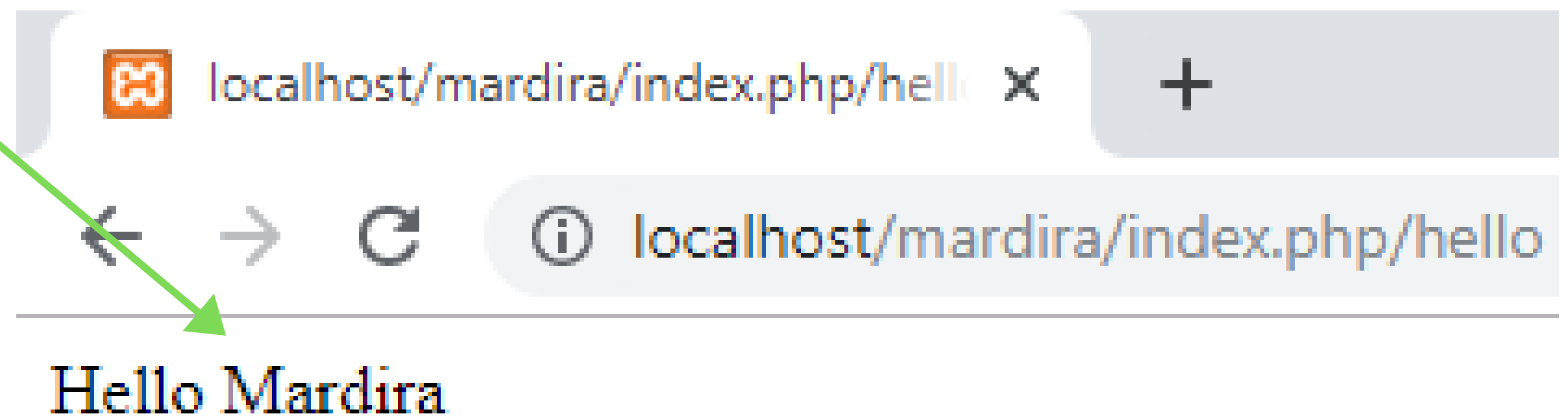
    public function hello_model()
    {
        $this->load->model('Hello_model');

        $data = $this->Hello_model->hello_mardira();
        echo $data;
    }
}
```

```
<?php

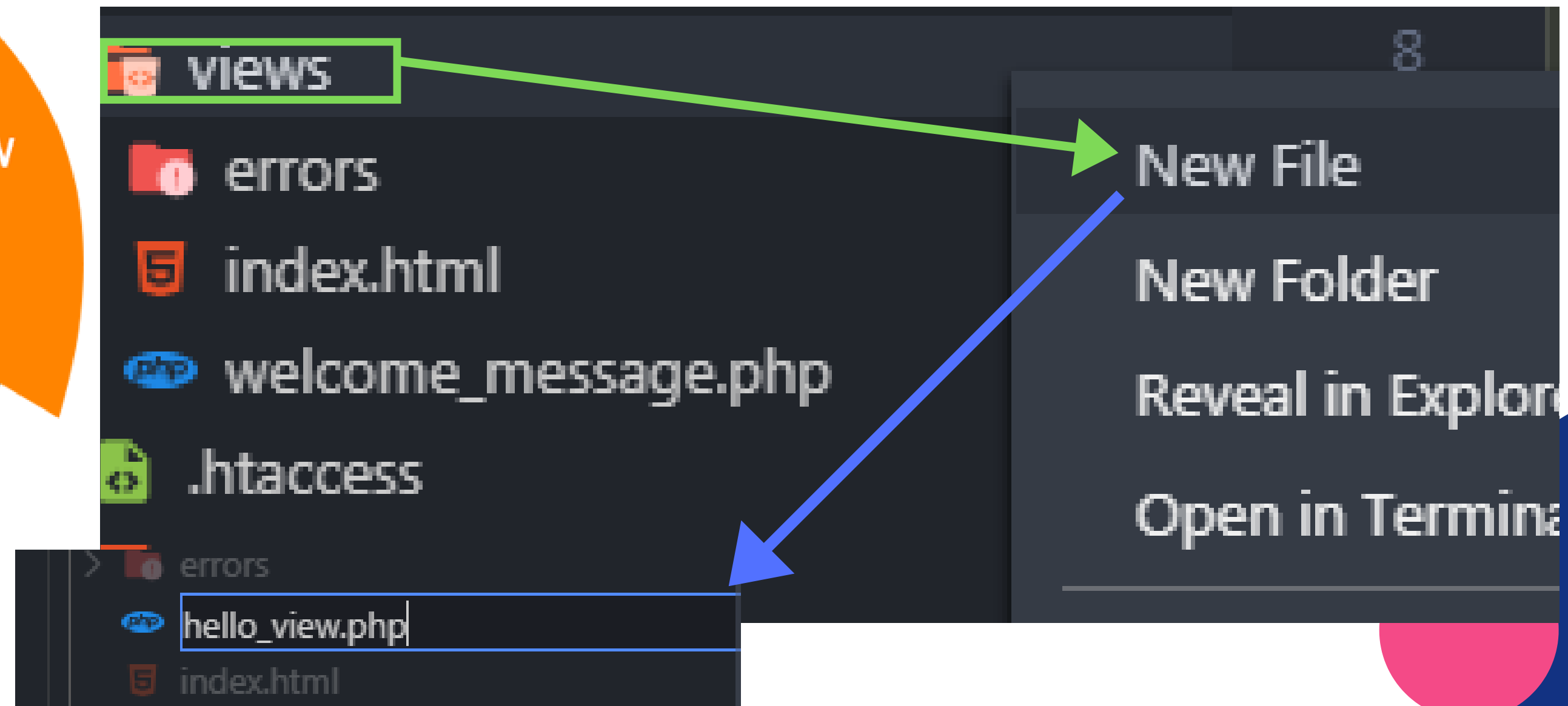
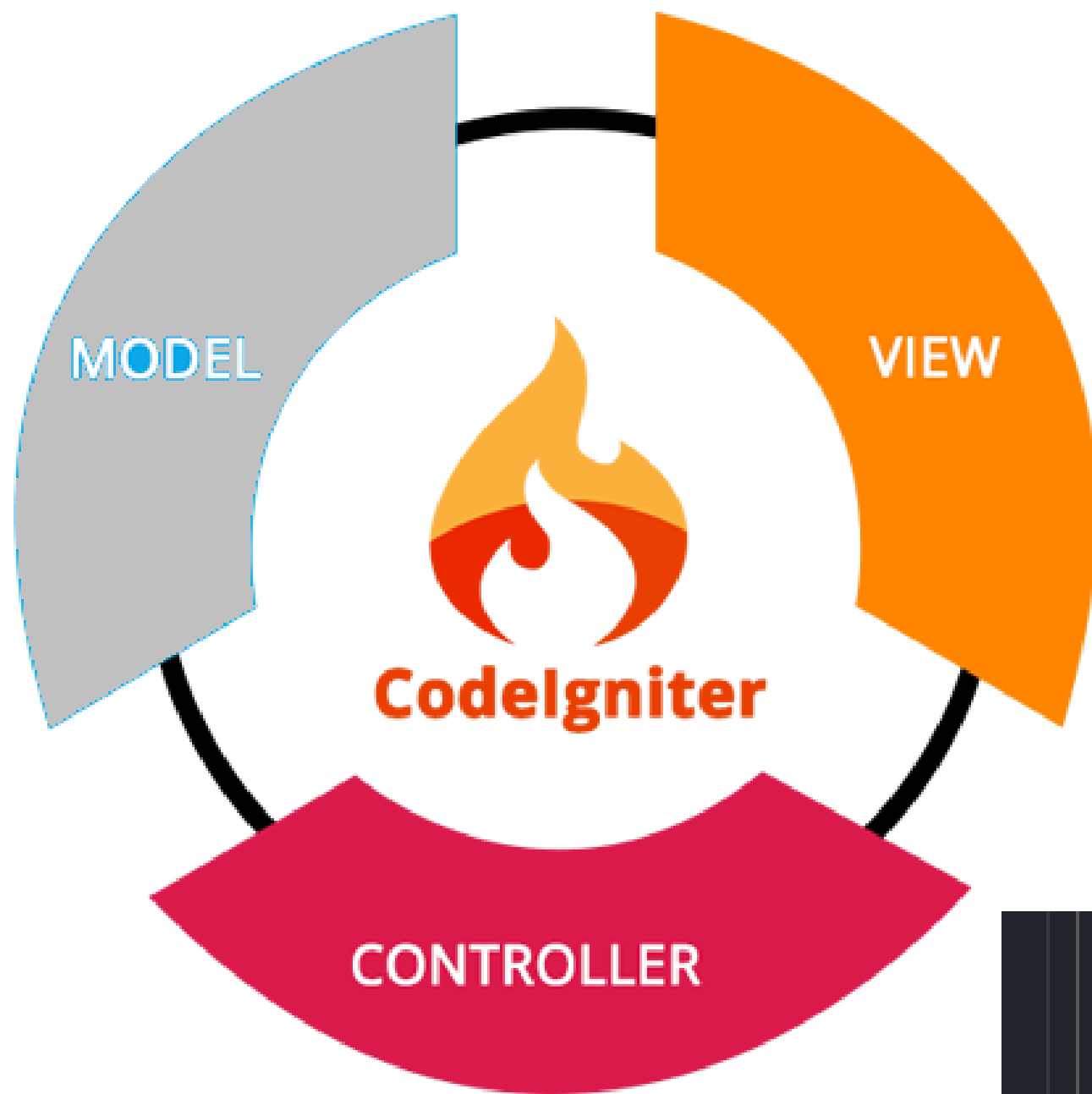
class Hello_model extends CI_Model
{
    public function hello_mardira()
    {
        return "Hello Mardira";
    }
}
```

**Konsep pemanggilan
data dari function Model
ke Controller**



APLIKASI YANG MENGGUNAKAN CONTROLLER DAN VIEW

Untuk membuat sebuah View baru, dengan cara klik kanan pada folders views kemudian pilih 'New File' setelah itu membuat file dengan nama "**hello_view.php**" atau "**v_hello.php**" dengan huruf kecil.



Lalu tuliskan kode berikut di file hello_view



```
1  <html>
2  <head>
3      <title>Hello Mardira</title>
4  </head>
5  <body>
6      <h3>Hello Mardira</h3>
7  </body>
8  </html>
```

Kemudian di file Controller tambahkan kode berikut

```
1  <?php
2
3  class Hello extends CI_Controller
4  {
5      public function index()
6      {
7          echo "Hello Mardira";
8      }
9
10     public function hello_model()
11     {
12         $this->load->model('Hello_model');
13         $data = $this->Hello_model->hello_mardira();
14         echo $data;
15     }
16
17     public function hello_view()
18     {
19         $this->load->view('hello_view');
20     }
21 }
```




CATATAN

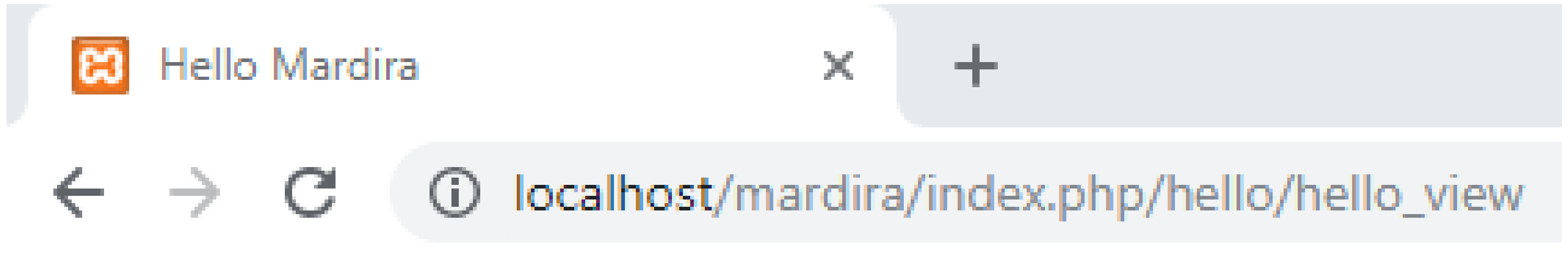
```
1  <?php
2
3  class Hello extends CI_Controller
4  {
5      public function index()
6      {
7          echo "Hello Mardira";
8      }
9
10     public function hello_model()
11     {
12         $this->load->model('Hello_model');
13         $data = $this->Hello_model->hello_mardira();
14         echo $data;
15     }
16
17     public function hello_view()
18     {
19         $this->load->view('hello_view');
20     }
21 }
```

Untuk menggunakan sebuah View di Controller, maka View tersebut harus dimuat terlebih dahulu melalui Controller dengan baris kode:

```
$this->load->view('nama_view');
```

Kemudian jalankan Aplikasi dengan menuliskan URL sebagai berikut:

`http://localhost/mardira/index.php/hello/hello_view`

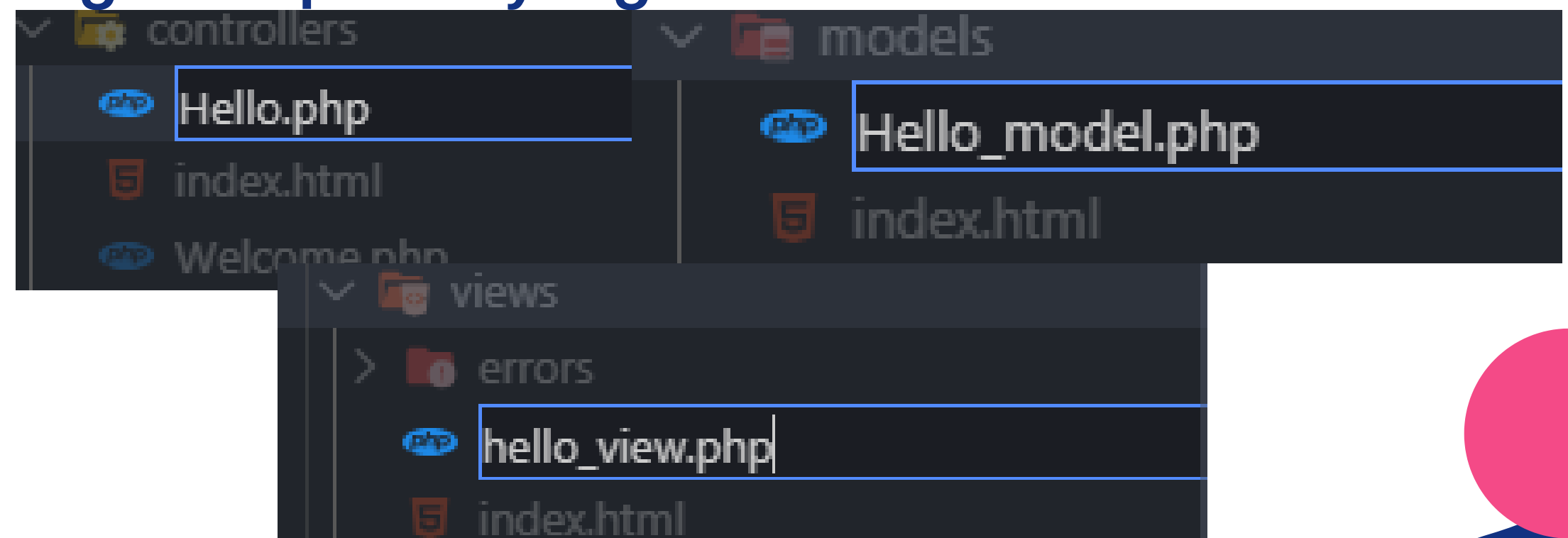
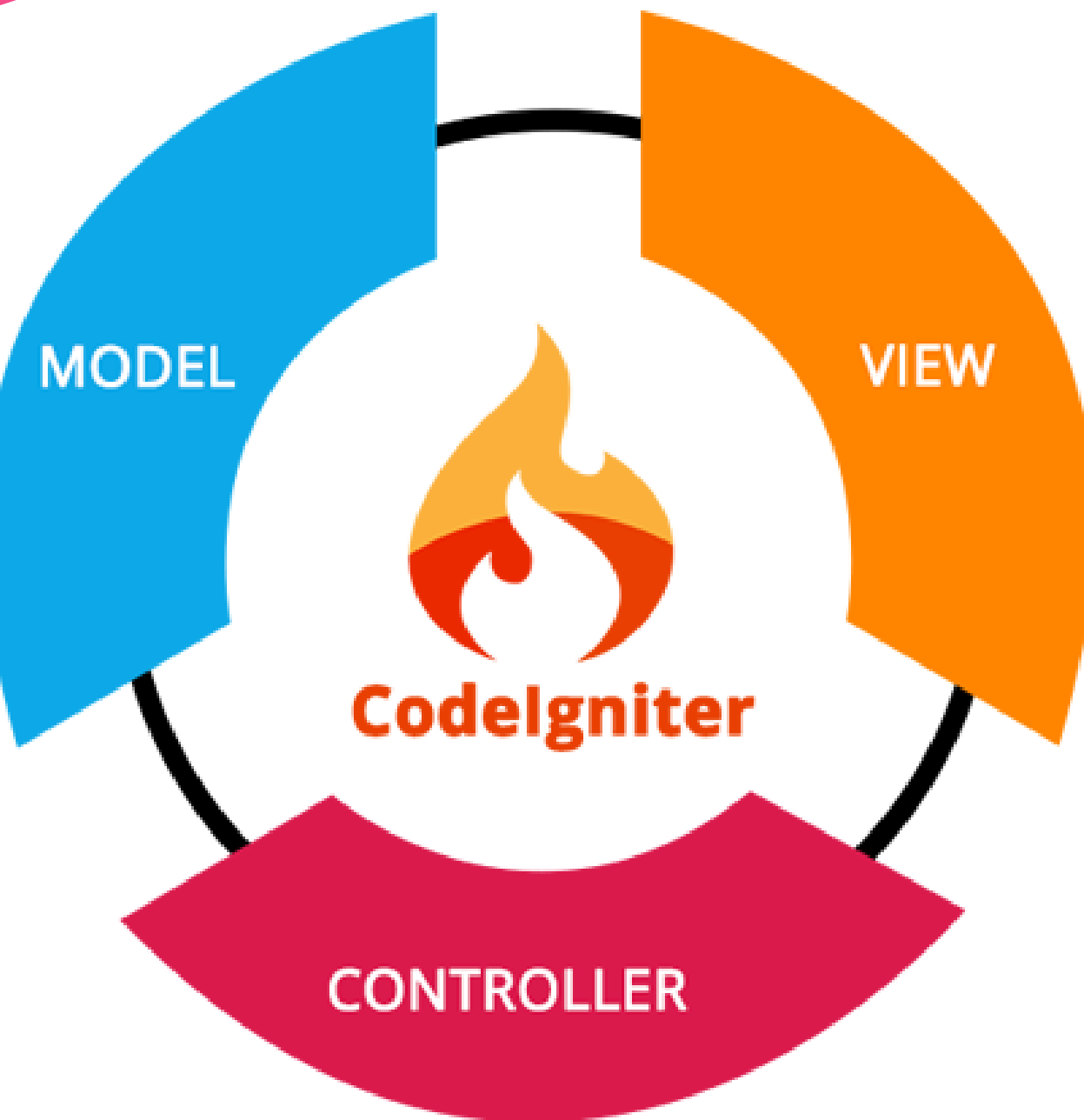


Hello Mardira

APLIKASI YANG MENGGUNAKAN CONTROLLER, MODEL, DAN VIEW

Aplikasi dengan komponen Controller, Model, dan View merupakan aplikasi yang lengkap, teknik inilah yang seharusnya digunakan oleh setiap aplikasi yang dikembangkan menggunakan CodeIgniter.

Untuk implementasi aplikasi masih menggunakan ketiga komponen yang sudah dibuat.



Kemudian di file Controller tambahkan kode berikut

```
public function hello_view()
{
    $this->load->view('hello_view');
}
```

```
public function hello_mvc()
{
    $this->load->model('Hello_model');
    $data['mvc'] = $this->Hello_model->mardira_mvc();
    $this->load->view('hello_view', $data);
}
```

Lalu di file Model tambahkan kode berikut

```
1  <?php
2
3  class Hello_model extends CI_Model
4  {
5      public function hello_mardira()
6      {
7          return "Hello Mardira";
8      }
9
10     public function mardira_mvc()
11     {
12         return "Ini menggunakan MVC";
13     }
14 }
```

Di bagian View tambahkan kode di bawah ini



```
1  <html>
2  <head>
3      <title>Hello Mardira</title>
4  </head>
5  <body>
6      <h3>Hello Mardira</h3>
7      <?php
8          if (isset($mvc)) {
9              echo $mvc;
10         } ?>
11  </body>
12  </html>
```



CATATAN

```
public function hello_view()
{
    $this->load->view('hello_view');
}
```

```
public function hello_mvc()
{
    $this->load->model('Hello_model');
    $data['mvc'] = $this->Hello_model->mardira_mvc();
    $this->load->view('hello_view', $data);
}
```

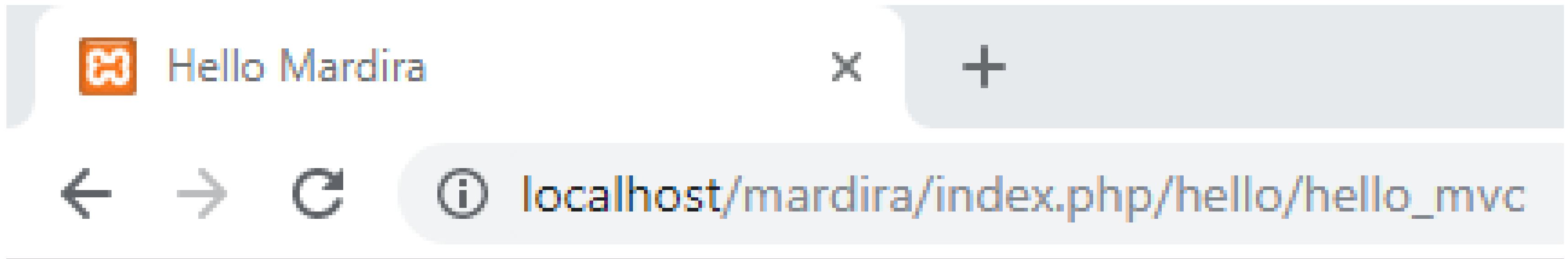
```
<html>
<head>
    <title>Hello Mardira</title>
</head>
<body>
    <h3>Hello Mardira</h3>
    <?php
        if (isset($mvc)) {
            echo $mvc;
        } ?>
</body>
</html>
```

Untuk menggunakan pola desain MVC Model di Controller, langkah pertama load Model terlebih dahulu melalui baris kode `$this->load->model('Nama_model');` setelah itu mengakses function di Model melalui Controller dengan baris kode `$this->Nama_model->nama_fungsi();`.

Baris kode tersebut ditampung melalui variable `$data` dengan alias `['mvc']` yang akan dilempar ke View untuk pemanggilan variable dari Controller, View cukup memanggil alias dari variable tersebut yaitu `$mvc`, view sengaja menggunakan statement `if isset($mvc)` supaya tidak terjadinya error karena masih menggunakan view yang sama.

Kemudian jalankan Aplikasi dengan menuliskan URL sebagai berikut:

`http://localhost/mardira/index.php/hello/hello_mvc`



Hello Mardira

Ini menggunakan MVC


```
public function hello_mvc()
```

```
{
```

```
$this->load->model('Hello_model');
```

```
$data['mvc'] = $this->Hello_model->mardira_mvc();
```

```
$this->load->view('hello_view', $data);
```

```
}
```

Controller

```
class Hello_model extends CI_Model
{
    public function hello_mardira()
    {
        return "Hello Mardira";
    }

    public function mardira_mvc()
    {
        return "Ini menggunakan MVC";
    }
}
```

Model

View

```
<html>
```

```
<head>
```

```
<title>Hello Mardira</title>
```

```
</head>
```

```
<body>
```

```
<h3>Hello Mardira</h3>
```

```
<?php
```

```
if (isset($mvc)) {
```

```
    echo $mvc;
```

```
} ?>
```

```
</body>
```

```
</html>
```

Konsep
pemanggilan data
dari Controller,
Model dan View

Hello Mardira

Ini menggunakan MVC

TERIMA KASIH