

**MODUL 10**  
**FRAMEWORK CODEIGNITER**

**1. Pengertian CodeIgniter**

CodeIgniter adalah sebuah web framework yang dikembangkan oleh Rick Ellis dari Ellis Lab. CodeIgniter dirancang untuk menjadi sebuah web framework yang ringan dan mudah untuk digunakan. Bahkan pengakuan dari Rasmus Lerdorf, pencipta bahasa pemrograman PHP, mengatakan bahwa CodeIgniter merupakan web framework mudah dan handal.

Sebelum mencoba CodeIgniter, perlu diketahui istilah web framework itu sendiri. Menurut Microsoft Computer Dictionary, web adalah sekumpulan dokumen yang saling terhubung dalam sistem hypertext yang penggunaannya akan menjelajahi web melalui halaman beranda. Sedangkan framework adalah desain struktur dasar yang dapat digandakan kembali (reuseable) yang terdiri dari abstract class dan concrete class di pemrograman berorientasi objek.

Menurut dokumentasi CodeIgniter, CodeIgniter merupakan toolkit bagi orang yang ingin membangun aplikasi web menggunakan PHP. Tujuannya adalah membuat pengembangan proyek menjadi lebih cepat dibandingkan dengan menulis kode dari awal (scratch). CodeIgniter menyediakan kumpulan library untuk tugas – tugas yang sering dilakukan (commonly needed task) dan sangat mudah untuk mengakses library yang tersedia di CodeIgniter. Dengan menggunakan CodeIgniter, kita cukup fokus pada pengembangan proyek dan meminimalisir jumlah kode yang akan ditulis.

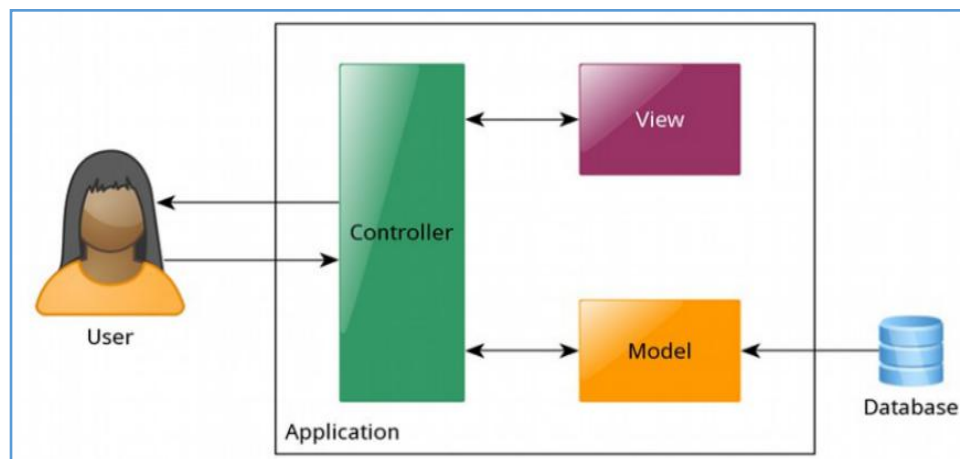
Sebagai web framework yang populer yang menggunakan bahasa pemrograman PHP, CodeIgniter mempunyai berbagai keunggulan seperti yang disebutkan di dokumentasinya:

- a) Free, karena berada dibawah lisensi open source mirip Apache/BSD, kita dapat melakukan apapun dengan CodeIgniter. Lisensi lengkapnya dapat dilihat di halaman dokumentasi
- b) Light Weight, sistem inti CodeIgniter memerlukan library yang sedikit. Berbeda sekali dengan framework lainnya yang membutuhkan banyak sumber daya tambahan. Library tambahan akan digunakan ketika request secara dinamis, membuat sistem yang dibangun menjadi efisien dan cukup cepat
- c) Fast, menurut dokumentasi, performa yang dimiliki CodeIgniter terbukti cepat setelah dibandingkan dengan web framework lainnya
- d) Menggunakan kaidah M-V-C, Dengan menggunakan Model-View-Controller, kita dapat memisahkan bagian logic dan presentation dari aplikasi yang kita bangun. Hal ini tentu sangat cocok dan bagus untuk proyek yang memfokuskan desainer fokus pada template file dan programmer fokus pada pembangunan logic dari aplikasi yang dibangun
- e) Menghasilkan URL yang bersih, URL yang dihasilkan oleh CodeIgniter bersih dan ramah terhadap mesin pencari. CodeIgniter menggunakan pendekatan segment-based dibandingkan dengan query string yang biasa digunakan oleh programmer yang tidak menggunakan web framework.
- f) Packs a Punch, CodeIgniter hadir dengan berbagai library yang akan membantu tugas – tugas di pengembangan web yang sudah umum dan sering dilakukan seperti mengakses database, mengirim email, validasi data dari form, mengelola session, manipulasi gambar,

bekerja dengan XML-RPC dan masih banyak lagi.

- g) Extensible, kita dapat menambahkan library atau helper yang kita ciptakan sendiri ke dalam CodeIgniter. Selain itu kita dapat juga menambahkan fitur lewat class extension atau system hooks.
- h) Thoroughly Documented, hampir semua fitur, library, dan helper yang ada di CodeIgniter telah terdokumentasi dengan lengkap dan tersusun dengan baik. Ketika mendapatkan unduhan CodeIgniter, dokumentasinya sudah tersedia dan siap digunakan
- i) Mempunyai komunitas yang ramah, komunitas CodeIgniter sangat ramah dan siap membantu pengguna CodeIgniter pemula atau yang sudah mahir. Komunitasnya dapat ditemui di : <http://codeigniter.com/forums>

## **2. Cara Kerja MVC**



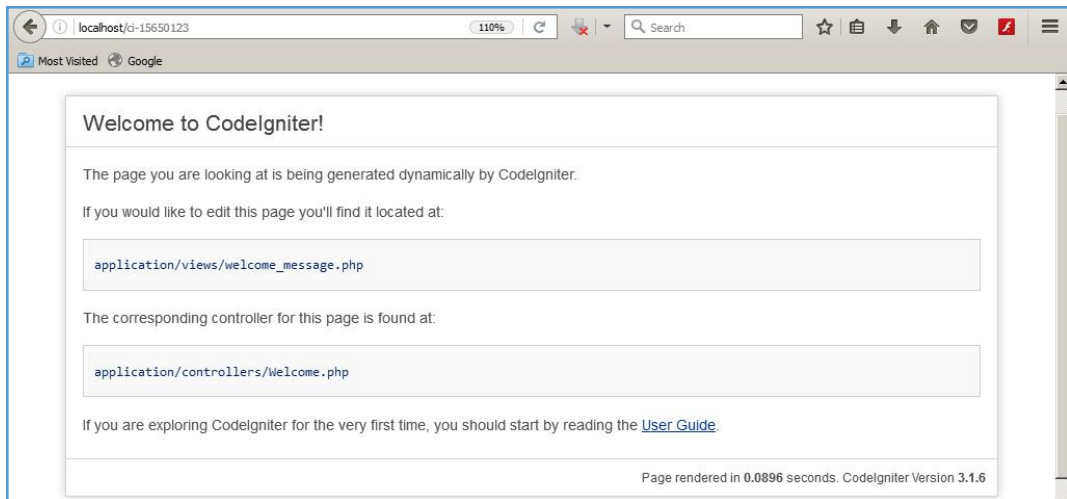
CodeIgniter menggunakan pendekatan Model-View-Controller, yang bertujuan untuk memisahkan logika dan presentasi. Konsep ini mempunyai keunggulan dimana desainer dapat bekerja pada template file, sehingga redundansi kode presentasi dapat diperkecil. Berikut adalah konsep Model- View-Controller yang diterapkan di CodeIgniter:

- a) Model menggambarkan struktur data. Biasanya kelas model akan berisi fungsi yang digunakan untuk mengambil, menambah, dan memperbaharui informasi yang ada di database.
- b) View adalah informasi yang diperlihatkan kepada user. View adalah halaman web yang terdiri dari HTML, CSS dan Javascript, tapi pada Code Igniter, view dapat juga sebagai potongan halaman seperti header atau footer. Bahkan dapat juga halaman RSS atau tipe halaman lainnya.
- c) Controller adalah perantara Model, View, dan resource lainya yang dibutuhkan untuk menangani HTTP request dan menghasilkan halaman web.

Tapi pada CodeIgniter, developer juga dapat mengabaikan pemakaian Model dan cukup menggunakan Controller dan View.

### 3. Langkah - langkah Codeigniter

- a) Download versi Codeigniter 3.x terbaru di <https://codeigniter.com>
- b) Ekstrak file “CodeIgniter-3.x” ke folder “htdocs”
- c) Ubah Nama folder “CodeIgniter-3.x” menjadi ci+nim Anda, misalnya ci-15650123
- d) Akses web browser <http://localhost/ci-15650123>



- e) Buka “/application/config/config.php”. Ubah  
`$config['base_url'] = '';`  
menjadi  
`$config['base_url'] = 'http://localhost/ci-15650123';`  
  
`$config['index_page'] = 'index.php';`  
menjadi  
`$config['index_page'] = '';`
- f) Di folder ‘/’ (letaknya sejajar dengan folder application dan system),  
tambahkan .htaccess yang berisi:  
`RewriteEngine On`  
`RewriteCond %{REQUEST_FILENAME} !-f`  
`RewriteCond %{REQUEST_FILENAME} !-d`  
`RewriteRule ^(.*)$ index.php/$1 [L]`
- g) Pada “/application/config/autoload.php”, tambahkan url di helper sebagai berikut:  
`$autoload['helper'] = array('url');`
- h) Buka “/application/config/routes.php”. Ubah  
`$route['default_controller'] = 'welcome';`  
menjadi  
`$route['default_controller'] = 'aplikasi';`
- i) Di folder ‘/’ (letaknya sejajar dengan folder application dan system), buat folder  
“assets”. Masukkan bootstrap dan jquery ke dalamnya.

- j) Buat file Aplikasi.php pada folder “/application/controllers/” dengan source code:

```
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Aplikasi extends CI_Controller {
5
6     public function penjumlahan($angka1,$angka2)
7     {
8         $data['angka1'] = $angka1;
9         $data['angka2'] = $angka2;
10        $data['hasil'] = $angka1 + $angka2;
11
12        $this->load->view('v_penjumlahan', $data);
13    }
14 }
```

- k) Buat file v\_penjumlahan.php pada folder “/application/views/” dengan source code:

```
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 ?>
4 <!DOCTYPE html>
5 <html lang="en">
6 <head>
7     <meta charset="utf-8">
8     <meta name="viewport" content="width=device-width, initial-scale=1.0">
9     <link href="<?=site_url('assets/css/bootstrap.css')?>" rel="stylesheet">
10    <link href="<?=base_url('assets/css/bootstrap-responsive.css')?>" rel="stylesheet">
11 </head>
12 <body>
13 <div class="container">
14     <fieldset>
15         <legend>Penjumlahan - CodeIgniter</legend>
16         <? if (isset($angka1) and isset($angka2) and isset($hasil))
17             {
18                 echo $angka1." + ".$angka2." = ".$hasil;
19             }
20         ?>
21     </fieldset>
22 </div>
23 </body>
24
25 <script src="<?=base_url('assets/js/jquery.js')?>"></script>
26 <script src="<?=base_url('assets/js/bootstrap.js')?>"></script>
27 </html>
```

- l) Akses aplikasi penjumlahan melalui url di web browser

<http://localhost/ci-15650123/aplikasi/penjumlahan/2/3>

Dalam Codeigniter, url tersebut memiliki arti sebagai berikut:

aplikasi = controller

penjumlahan = function

Angka 2 = parameter pertama

Angka 3 = parameter kedua

Sehingga tampil seperti gambar di bawah ini



m) Buat database ci+nim, misalnya ci15650123 dan tabel kalkulator

```
1 CREATE TABLE `kalkulator` (  
2   `id_kalkulator` int(5) NOT NULL auto_increment,  
3   `angka1` int(7) default NULL,  
4   `angka2` int(7) default NULL,  
5   `operator` char(1) default NULL,  
6   `hasil` int(7) default NULL,  
7   PRIMARY KEY (`id_kalkulator`)  
8 ) ENGINE=MyISAM AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;  
9  
10 insert into `kalkulator` (`id_kalkulator`,`angka1`,`angka2`,`operator`,`hasil`) values (1,1,2,'+',3);
```

- n) Setting koneksi database di “/application/config/database.php”.
- o) Pada “/application/config/autoload.php”, tambahkan database di libraries sebagai berikut:  
\$autoload['libraries'] = array('database');
- p) Pada file “/application/controllers/Aplikasi.php” tambahkan source code berikut:

```
1 <?php  
2 defined('BASEPATH') OR exit('No direct script access allowed');  
3  
4 class Aplikasi extends CI_Controller {  
5  
6     public function penjumlahan($angka1,$angka2)  
7     {  
8         $data['angka1'] = $angka1;  
9         $data['angka2'] = $angka2;  
10        $data['hasil'] = $angka1 + $angka2;  
11  
12        $this->load->view('v_penjumlahan', $data);  
13    }  
14    public function __construct()  
15    {  
16        parent::__construct();  
17        $this->load->model('m_aplikasi');  
18    }  
19  
20    public function index()  
21    {  
22        $this->load->view('v_aplikasi');  
23    }  
24  
25    public function kalkulator()  
26    {  
27        // mengecek masukan dari form  
28        $angka1 = $this->input->post('angka1');  
29        $angka2 = $this->input->post('angka2');  
30        $pilih_hitung = $this->input->post('pilih-hitung');  
31        $hasil_hitung = 0;  
32  
33        // mengecek proses perhitungan yang diminta  
34        if ($pilih_hitung == "+")  
35        {  
36            $hasil_hitung = $angka1 + $angka2;  
37        } else if ($pilih_hitung == "-")  
38        {  
39            $hasil_hitung = $angka1 - $angka2;  
40        } else if ($pilih_hitung == "*")  
41        {  
42            $hasil_hitung = $angka1 * $angka2;  
43        } else if ($pilih_hitung == "/")  
44        {  
45            $hasil_hitung = $angka1 / $angka2;  
46        }  
47  
48        // membungkus semua data perhitungan untuk ditampilkan di view  
49        $data['angka1'] = $angka1;  
50        $data['angka2'] = $angka2;  
51        $data['pilih_hitung'] = $pilih_hitung;  
52        $data['hasil_hitung'] = $hasil_hitung;  
53  
54        // simpan ke database  
55        $this->m_aplikasi->simpan($data);  
56  
57        // menampilkan hasil  
58        $this->load->view('v_aplikasi', $data);  
59    }  
60 }  
61 }
```



q) Buat file M\_aplikasi.php pada folder “/application/models/” dengan source code:

```
1 <?
2 class M_aplikasi extends CI_Model
3 { function __construct()
4 { parent::__construct();
5 }
6
7 public function simpan($data)
8 { // query binding utk security
9 $this->db->query("INSERT INTO kalkulator (angka1,
10 angka2,operator,hasil) VALUES (?, ?, ?, ?)",
11 array($data['angka1'],$data['angka2'],
12 $data['pilih_hitung'],$data['hasil_hitung']));
13
14 // menghapus variabel dari memory
15 // $data = null;
16 unset($data);
17 }
18 }
19 ?>
```

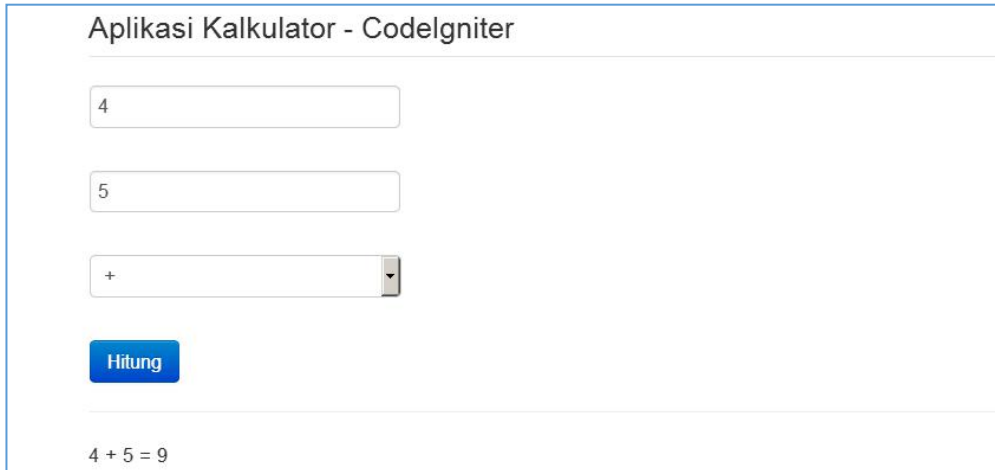
r) Buat file v\_aplikasi.php pada folder “/application/views/” dengan source code:

```
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 ?>
4 <!DOCTYPE html>
5 <html lang="en">
6 <head>
7 <meta charset="utf-8">
8 <meta name="viewport" content="width=device-width, initial-scale=1.0">
9 <link href="<?=site_url('assets/css/bootstrap.css')?>" rel="stylesheet">
10 <link href="<?=base_url('assets/css/bootstrap-responsive.css')?>" rel="stylesheet">
11 </head>
12 <body>
13 <div class="container">
14 <fieldset>
15 <legend>Aplikasi Kalkulator - CodeIgniter</legend>
16 <form method="post" action="<?=base_url('aplikasi/kalkulator')?>" >
17 <input type="text" name="angka1" rel="tooltip" data-placement="right" title="Angka 1"/> <br /><br />
18 <input type="text" name="angka2" rel="tooltip" data-placement="right" title="Angka 2"/> <br /><br />
19 <select name="pilih_hitung">
20 <option value="+">+</option>
21 <option value="-">-</option>
22 <option value="*">*</option>
23 <option value="/">/</option>
24 </select><br/><br/>
25 <button type="submit" name="hitung" class="btn btn-primary">Hitung</button>
26 </form>
27
28 <? if (isset($angka1) and isset($angka2))
29 { echo "<hr/>".$angka1." ".$pilih_hitung." ".$angka2." = ".$hasil_hitung;
30 }
31 ?>
32 </fieldset>
33 <br/><br/><br/>
34 </div>
35 </body>
36
37 <script src="<?=base_url('assets/js/jquery.js')?>"></script>
38 <script src="<?=base_url('assets/js/bootstrap.js')?>"></script>
39 </html>
```

- s) Akses aplikasi kalkulator melalui web browser

<http://localhost/ci-15650123>

Sehingga tampil seperti gambar di bawah ini



Aplikasi Kalkulator - CodeIgniter

4

5

+

Hitung

4 + 5 = 9

Angka dan operator yang kita masukkan akan tersimpan dalam database sebagai log aplikasi kalkulator.

#### **4. Latihan**

Buat aplikasi sederhana menggunakan Codeigniter untuk menyimpan ke dalam database dengan item sebagai berikut:

NIM, Nama Mahasiswa, Tempat Lahir, Tanggal Lahir, Jenis Kelamin, Alamat.

Selamat mengerjakan !!!