

|  |
| --- |
| **Web Programming II** |
|  |
| **Modul 2** |
| **Model, View, Controller**  **#Program Penjumlahan** |

**Disusun oleh:**

**Dwi Intan Af’idah, S.T., M.Kom**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**POLITEKNIK HARAPAN BERSAMA**

**TAHUN AJARAN 2020/2021**

Daftar Isi

[Daftar Isi ii](#_Toc53438718)

[1 MVC 3](#_Toc53438719)

[2 Aturan Merancang Model, View, dan Controller 5](#_Toc53438720)

[3 Program Penjumlahan 7](#_Toc53438721)

[4 Menentukan Controller Standar (Default) 10](#_Toc53438722)

[5 Menghilangkan index.php pada Codeigniter 11](#_Toc53438723)

1. MVC

MVC adalah teknik atau konsep yang memisahkan komponen utama menjadi tiga komponen yaitu model, view dan controller.

1. Model

Model adalah kelas yang merepresentasikan atau memodelkan tipe data yang akan digunakan oleh aplikasi. Model juga dapat didefinisakn sebagai bagian penanganan yang berhubungan dengan pengolahan atau manipulasi database. seperti misalnya mengambil data dari database, menginput dan pengolahan database lainnya. semua intruksi atau fungsi yang berhubung dengan pengolahan database di letakkan di dalam model. Sebagai contoh, jika ingin membuat aplikasi untuk menghitung luas dan keliling lingkaran, maka dapat memodelkan objek lingkaran sebagai kelas model.

Sebagai catatan, Semua model harus disimpan di dalam folder **application\models**

1. View

View merupakan bagian yang menangani halaman user interface atau halaman yang muncul pada user(pada browser). tampilan dari user interface di kumpulkan pada view untuk memisahkannya dengan controller dan model sehingga memudahkan web designer dalam melakukan pengembangan tampilan halaman website.

9

1. Controller

Controller merupakan kumpulan intruksi aksi yang menghubungkan model dan view, jadi user tidak akan berhubungan dengan model secara langsung, intinya data yang tersimpan di database (model) di ambil oleh controller dan kemudian controller pula yang menampilkan nya ke view. Jadi controller lah yang mengolah intruksi.

Dari penjelasan tentang model view dan controller di atas dapat di simpulkan bahwa controller sebagai penghubung view dan model. misalnya pada aplikasi yang menampilkan data dengan menggunakan metode konsep mvc, controller memanggil intruksi pada model yang mengambil data pada database, kemudian controller yang meneruskannya pada view untuk di tampilkan. jadi jelas sudah dan sangat mudah dalam pengembangan aplikasi dengan cara mvc ini karena web designer atau front-end developer tidak perlu lagi berhubungan dengan controller, dia hanya perlu berhubungan dengan view untuk mendesign tampilann aplikasi, karena back-end developer yang menangani bagian controller dan modelnya. Jadi pembagian tugas pun menjadi mudah dan pengembangan aplikasi dapat di lakukan dengan cepat dan terstruktur.

Bentuk umum model seperti berikut:

Class Nama\_model extends CI\_model {

// badan kelas

}

jika ingin memanggil konstruktor kelas CI\_model dari konstruktor kelas model yang didefinisikan, maka bentuk umumnya seperti berikut:

class Nama\_model extends CI\_model {

//konstruktor kelas model

Function \_\_construct( ) {

//memanggil konstruktor kelas CI\_model

$parent: : \_\_construct( );

//...

}

// ...

}

1. Aturan Merancang Model, View, dan Controller
2. Model

Secara umum perancangan model dapat dibuat menggunakan aturan berikut:

* 1. Model harus memiliki atribut yang dapat mewakili element tertentu.
  2. Model seharusnya berisi kode kode yang menangani proses bisnis untuk data bersangkutan, misalnya untuk melakukan proses validasi data.
  3. Model boleh berisi kode kode yang dugunakan untuk memanipulasi data, misalnya untuk menambah baris data baru, mengubah baris data, dan menghapus baris data pada sebuah tabel yang terdapat dalam database.
  4. Model seharusnya tidak berisi kode kode yang berkaitan langsung dengan permintaan yang dilakukan oleh user melalu URL (tidak berisi $\_GET maupun $\_POST). Pekerjaan seperti ini seharusnya dilakukan oleh controller.
  5. Model seharusnya tidak berisi kode kode yang berkaitan dengan presentasi (tampilan) halaman web yang akan disajikan ke hadapan user. Pekerjaan semacam ini seharusnya dilakukan oleh view.

1. View

Berikut ini aturan yang dapat digunakan untuk merancang view.

* 1. View harus berisi kode kode yang bersifat presentasional, biasanya berupa kode HTML. View juga dapat berisi perintah perintah PHP sederhana yang masih berkaitan dengan proses pembuatan tampilna, misalnya untuk menampilkan data yang dihasilkan oleh model tertentu.
  2. View seharusnya tidak berisi kode PHP yang melakukan akses data secara langsung ke database. Pekerjaan semacam ini sebaiknya dilakukan oleh model.
  3. View seharunsnya menghindari kode PHP yang mengakses secara langsung permintaan dari user (tidak berisi $\_GET maupun $\_POST). Tugas ini sebaiknya didelegasikan ke controller

1. Controller

Berikut ini aturan yang dapat digunakan untuk merancang controller.

* 1. Controller boleh mengakses $\_GET, $\_POST, dan variabel variabel PHP lainnya yang berkaitan dengan permintaan user.
  2. Controller boleh membuat objek (instance) dari kelas kelas model dan mengarahkan ke view (jika model bersangkutan menghasilkan data yang perlu ditampilkan ke user)
  3. Controller seharusnya tidak berisi kode kode yang mengakses data secara langsung dari database. Tugas seperti ini sebaiknya dilakukan oleh model. Dan controller tinggal menggunakannya saja.
  4. Controller seharusnya tidak berisi kode kode HTML yang diperlukan untuk tujuan presentasi. Tugas ini sebaiknya dilakukan oleh view.

1. Program Penjumlahan

**Controller Latihan1** simpan dalam folder **Application/controllers/**

<?php

class Latihan1 extends CI\_Controller

{

public function index()

{

echo "Selamat Datang.. selamat belajar Web Programming";

}

}

Untuk melihat hasilnya ketikkan url **localhost/pustaka-booking/latihan1**

**Contoh menggunakan Controller dan Model**

Model **Model\_latihan1.php** simpan di **application\models\**

<?php

class Model\_latihan1 extends CI\_Model

{

//membuat variable untuk menampung nilai public $nilai1, $nilai2, $hasil;

//method penjumlahan

public function jumlah($nil1 = null, $nil2 = null)

{

$this->nilai1 = $nil1;

$this->nilai2 = $nil2;

$this->hasil = $this->nilai1 + $this->nilai2; return $this->hasil;

}

}

Ubah controller Latihan1 yang sebelumnya sudah dibuat sehingga menjadi seperti berikut:

<?php

class Latihan1 extends CI\_Controller

{

public function index()

{

echo "Selamat Datang.. selamat belajar Web Programming"; //$this->load->view('view-latihan1');

}

public function penjumlahan($n1, $n2)

{

$this->load->model('Model\_latihan1');

$hasil = $this->Model\_latihan1->jumlah($n1, $n2); echo "Hasil Penjumlahan dari". $n1 ." + ". $n2 ." = "

.$hasil;

}

}

Untuk melihat hasilnya ketikkan url

**localhost/pustaka-booking/index.php/latihan1/penjumlahan/2/6**

**Contoh3 menggunakan Controller dan View dan Model**

View **view-latihan.php** Simpan di **application\views\**

<html>

<head>

<title>Latihan 1</title>

</head>

<body>

Halo Kawan.. Yuk kita belajar web programming..!!!<br>

Nilai 1 = <?= $nilai1; ?>

Nilai 2 = <?= $nilai2; ?>

ini hasil dari pemodelan dengan methode penjumlahan yaitu <?= $nilai1 . " + " . $nilai2 . " = " . $hasil; ?>

</body>

</html>

Ubah controller Latihan1 yang sebelumnya sudah dibuat sehingga menjadi seperti berikut:

<?php

class Latihan1 extends CI\_Controller

{

public function index()

{

echo "Selamat Datang.. selamat belajar Web Programming"; //$this->load->view('view-latihan1');

}

public function penjumlahan($n1, $n2)

{

$this->load->model('Model\_latihan1');

$data['nilai1'] = $n1;

$data['nilai2'] = $n2;

$data['hasil'] = $this->Model\_latihan1->jumlah($n1, $n2);

$this->load->view('view-latihan1', $data);

}

}

Untuk melihat hasilnya ketikkan url

**localhost/pustaka-booking/index.php/latihan1/penjumlahan/2/6**

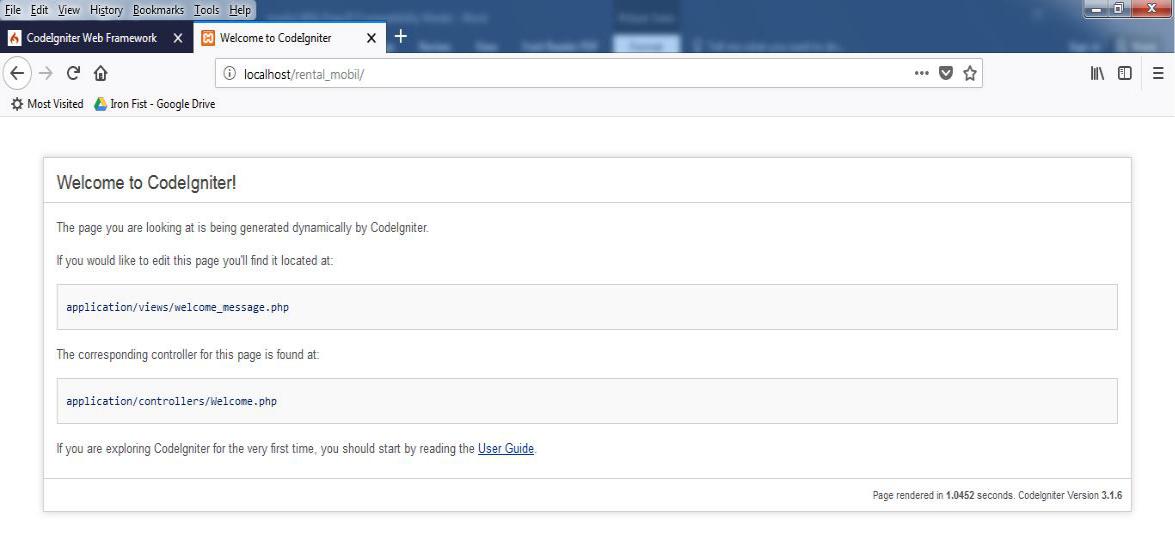
1. Menentukan Controller Standar (Default)

Controller standar (default) adalah controller yang akan dipanggil secara otomatis ketika user tidak menyertakan nama controller di dalam URI.

Contoh penulisan URI:

http://localhost/pustaka-booking/index.php

pada contoh permintaan di atas, kita tidak menyertakan nama controller yang akan dipanggil (segmen pertama). Permintaan tersebut akan memanggil controller standar yang sudah didefinisikan oleh codeigniter, dan akan memberikan hasil seperti berikut:



Controller standar dapat ditentukan sendiri sesuai dengan keinginan, yaitu dengan cara melakukan konfigurasi pada file **routes.php** sesaui dengan keinginan, yaitu dengan cara

cari baris code berikut:

$route['default\_controller'] = 'welcome';

Ubah kata welcome dengan nama controller yang diinginkan untuk dijadikan controller standar. Misal dengan Controller Latihan1

Akses URL ke http://localhost/pustaka-booking/Latihan1

1. Menghilangkan index.php pada Codeigniter

Dalam rangka pengamanan web site, biasanya seorang web developer akan melakukan berbagai trik untuk dapat mengamankan web miliknya. Ada yang dibuatkan sistem sandi yang diubah secara berkala, ada yang menggunakan sistem sandi dengan teknik kriptografi yaitu mengenkrip sandi sedemikian rupa agar tidak mudah dipecahkan oleh pihak yang tidak bertanggung jawab, ada juga yang menggunakan trik kamuplase pengaksesan file, dan lain lain. Berikut ini adalah salah satu trik untuk mengelabuhi pihak yang tidak bertanggung jawab agar tidak mudah untuk melakukan pencurian atau penggunaan data yang merugikan yaitu dengan cara menghilangkan index.php pada url codeigniter. cara nya buka **config.php** yang terletak pada folder **application/config/***,* kemudian cari pengaturan

**$config[„index\_page]** seperti di bawah ini.

$config['index\_page'] = 'index.php';

Kemudian hapus index.php pada pengaturan sehingga menjadi

$config['index\_page'] = '';

Setelah menghapus index, selanjutnya adalah membuat file **.htaccess** pada direktori root codeigniter. Caranya buat file baru dengan nama **.htaccess** tanpa diberi extensi, karena .htaccess merupakan file yang tidak berekstensi. Diawali dengan tanda titik ( . ).

File **pustaka-booking/.htaccess**

RewriteEngine On

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteCond %{REQUEST\_FILENAME} !-d

RewriteRule ^(.\*)$ index.php/$1 [L]

Simpan dan sekarang coba akses di web. Disini untuk contoh dalam rangka mencoba apakah sudah berhasil atau belum menghilangkan index pada codeigniter, kita akan coba mengakses method penjumlahan yang sudah ada pada controller **Latihan1.php** tanpa menyertakan kata index.php pada URL.

Jika biasanya untuk mengakses method dalam kontroler Latihan1 diatas harus dengan mengetikkan

http://localhost/pustaka-booking/index.php/latihan1/penjumlahan/4/6

maka sekarang tidak perlu, melainkan bisa langsung mengakses alamat url di atas tanpa menyertakan kata index.php

http://localhost/pustaka-booking/ latihan1/penjumlahan/4/6

Kemudian hapus index.php pada pengaturan sehingga menjadi

$config['index\_page'] = '';

Setelah menghapus index, selanjutnya adalah membuat file **.htaccess** pada direktori root codeigniter. Caranya buat file baru dengan nama **.htaccess** tanpa diberi extensi, karena .htaccess merupakan file yang tidak berekstensi. Diawali dengan tanda titik ( . ).

File **pustaka-booking/.htaccess**

RewriteEngine On

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteCond %{REQUEST\_FILENAME} !-d

RewriteRule ^(.\*)$ index.php/$1 [L]

Simpan dan sekarang coba akses di web. Disini untuk contoh dalam rangka mencoba apakah sudah berhasil atau belum menghilangkan index pada codeigniter, kita akan coba mengakses method penjumlahan yang sudah ada pada controller **Latihan1.php** tanpa menyertakan kata index.php pada URL.

Jika biasanya untuk mengakses method dalam kontroler Latihan1 diatas harus dengan mengetikkan

http://localhost/pustaka-booking/index.php/latihan1/penjumlahan/4/6

maka sekarang tidak perlu, melainkan bisa langsung mengakses alamat url di atas tanpa menyertakan kata index.php

http://localhost/pustaka-booking/ latihan1/penjumlahan/4/6

berikut hasil dalam browser

