**Pertemuan 4  
pengenalan CodeIgniter**

1. **Pengertian MVC (Model View Controller)**

MVC adalah Model, View, Controller. MVC adalah pola pemrograman yang digunakan dalam pembangunan aplikasi web. Pola ini memisahkan business logic, user interface dan database. Implementasi yang bagus dari pola ini adalah apabila developer dapat memodifikasi interface atau business logic tanpa mempengaruhi lainnya. MVC juga meningkatkan fleksibilitas aplikasi dengan me-reuse function di model atau view pada controller lainnya.

**Model**  interaksi ke database dan akan berisi function untuk melakukan aktivitas CRUD pada database.

**View**  tampilan data dan interface

**Controller** Controller berisi business logic yang berjalan di aplikasi, dan berada ditengah antara View dan Model. Dan mengatur apa yang dikirim ke view dari model

Contoh dari pendekatan MVC.

1. User berinteraksi dengan view pada saat melihat halaman web, kemudian memilih berita
2. Controller menerima request dari user untuk menampilkan berita, dan kemudian controller mengirimkan perintah untuk menjalankan function yang ada di model.
3. Model mengirimkan hasil query ke controller
4. Hasilnya akan ditampilkan melalui view dan ditampilkan ke user

Meskipun sepertinya pendekatan framework dengan menggunakan konsep MVC akan menambah pekerjaan, tetapi sebetulnya tidak seperti itu. Misalnya :

1. Apabila akan ada perubahan pada jumlah data yang diambil, yang diupdate adalah file model
2. Apabila ada perubahan pada tampilan, yang diupdate adalah file view
3. Apabila ada perubahan logika, yang diupdate adalah controller
4. **Pengertian Framework**

Frameworks membantu kita dalam mengembangkan aplikasi dengan menyediakan class yang dapat digunakan kembali dan fungsi yang dapat mengurangi waktu pengembangan secara signifikan. Beberapa kelemahan untuk framework, adalah bahwa mereka menyediakan kelas yang tidak diperlukan untuk aplikasi kita, dan menambahkan kode yang membuat aplikasi sulit untuk dinavigasi.

1. **Pengantar CodeIgniter**

CodeIgniter adalah sebuah aplikasi web framework untuk PHP yang dikembangkan oleh EllisLab . Hal ini memungkinkan pengembang untuk membangun aplikasi web lebih cepat, dan library yang terdapat di CI akan banyak membantu dan mempercepat pembangunan aplikasi di PHP. CodeIgniter didasarkan pada desain modular, yang berarti bahwa Anda dapat menerapkan library khusus sesuai dengan seleraAnda. Tutorial ini akan berusaha untuk menunjukkan anda dasar-dasar pembentukan framework CI yang menggunakan pendekatan MVC.

CodeIgniter adalah framework yang sangat ringan dengan performa yang sangat baik. Selain itu sangat cocok untuk pemula karena

1. learning curve rendah
2. user guide yang sangat bagus, berurutan dan sangat mudah untuk dipahami
3. komunitas yang sangat bagus (indonesia maupun internasional).

Berikut adalah daftar alasan apa yang membuat CodeIgniter kerangka cerdas untuk digunakan?

1. Small footprint dengan performa yang bagus ( Footprint : It does refer to resources. Particularly memory. It requires a smaller amount of memory when running.)
2. Pendekatan MVC
3. PHP 4 (4.3.2+) dan 5 (CI 1.7.x) , PHP 5.x (CI(2.0)
4. Support databases MySQL (4.1+), MySQLi, MS SQL, Postgres, Oracle, SQLite, dan ODBC.
5. Application security
6. Easy caching operations
7. Banyak libraries dan helpers untuk membantu developer dalam coding yang kompleks seperti email, image manipulation, form validation, file uploading, sessions, multilingual apps.
8. **Memasang CodeIgniter ke web page**

Setelah melakukan download, anda harus melakukan unzip. Di dalam folder CodeIgniter\_2.x akan terdiri dari 3 folder : application, system, dan user\_guide, dan 2 file : index.php dan license.txt

system : menyimpan file dan kode yang dibutuhkan untuk menjalankan CI

application : menyimpan file yang berisi aplikasi kita dan apa saja yang menjadi bagian dari aplikasi yang berhubungan dengan CI, 99% file yang akan kita kerjakan ada disini

user\_guide : dokumentasi dari CI

index.php : file yang berisi konfigurasi awal dari CI

Folder Application berisi :

1. config menyimpan semua file konfigurasi
2. controllers menyimpan semua controller yang merupakan bisnis proses dari aplikasi
3. errors menyimpan template dari halaman error
4. helpers menyimpan helper untuk membantu aplikasi
5. hooks menyimpan file yang digunakan untuk memodifikasi fungsi dari core files CI
6. language menyimpan text yang dapat digunakan untuk membuat multilingual sites.
7. libraries menyimpan library untuk aplikasi
8. Models menyimpan semua model yang berarti semua yang berhubungan dengan database
9. views menyimpan file untuk menampilkan halaman web ke user
10. cache menyimpan semua cache yang digenerate library caching
11. **Konfigurasi CodeIgniter** 
    1. autoload.php : semua yang akan diload secara otomatis oleh aplikasi
    2. config.php : berisi konfigurasi CI

beberapa yang penting :

$config['base\_url'] = '';

base\_url yang akan menjadi base url dari aplikasi, misal :

$config['base\_url'] = "http://localhost/";

$config['index\_page'] = ''; : jembatan untuk menjalankan controller

$config['log\_threshold'] = “”; : log dari aplikasi

* 1. database.php : konfigurasi database

['hostname'] hostname dari database server

['username'] username untuk connect ke database

['password'] password untuk connect ke database

['database'] nama database

['dbdriver'] driver database, contoh: mysql

['dbprefix'] prefix, yang akan DITAMBAHKAN ke nama table pada saat menggunakan Active Record

* 1. routes.php : digunakan untuk melakukan remap URL request ke controller yang kita inginkan, Contoh Segment dari URL :

http://www.detanto.net/index.php/(controller|folder)/method/parameter/

ada 2 yang sudah direserve :

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

$route['404\_override'] = 'errors/page\_missing' : apabila controller yang diremap atau di akses tidak ada Wildcards

(:any) : semua karakter akan dimasukkan

(:num) : hanya numerik

Regular Expression (Regex)

$route['hrdlaporan/([\w\-]+)?$'] = "hrd/laporan/list\_form/$1"; : w : regex untuk word boundaries

1. **Controller**

Controller adalah class yang dinamakan sedemikian rupa, sehingga dapat dipanggil di URL, disimpan di folder application/controller

Contoh :

http://www.bansal.ac.id/index.php/tampil

CI akan mencari file controller yang dibuat, misalnya tampil.php

Sintak umum dari Controller

class Namacontroller extends CI\_Controller {

function index()

{

=== code';

}

function namafunction()

{

=== code';

}

}

**Function dalam Controller**

Function dalam CI adalah function yang bisa dipanggil dari URL. Default yang pertama kali diload oleh CI apabila URL tidak menyebutkan nama function, adalah function bernama index().

<?php

class Tampil extends CI\_Controller {

function index() {

echo 'Selamat Malaam!';

}

function testing(){

echo “Testing”;

}

}?>

Catatan

1. Apabila kita memanggil http://www.banisaleh.ac.id/index.php/tampil, yang akan ditampilkan adalah function index.

2. apabila kita memanggil http://www. banisaleh.ac.id /index.php/tampil/testing yang akan ditampilkan adalah function testing.

Jadi apabila misalnya kita mempunyai halaman:

1. Home

2. Content

3. About us

Dan kita mempunyai 1 controller untuk semua halaman tersebut, kita harus mempunyai 3 fungsi di dalam 1 controller, yaitu untuk menampilkan home, content dan about us.

<?php

class Tampil extends CI\_Controller {

function index() {

echo 'Selamat Malaam!';

}

function home($parameter){

echo “ini adalah home”;

}

function content($parameter){

echo “ini adalah content”;

}

function aboutus($parameter){

echo “ini adalah about us”;

}

}?>

**Function Paramater**

Parameter bisa direquest dari URL

<?php

class Tampil extends CI\_Controller {

function index() {

echo 'Selamat Malaam!';

}

function testing($parameter){

echo $parameter;

}

}?>

Apabila kita memanggil http://www. banisaleh.ac.id /index.php/tampil/testing/123, akan ditampilkan 123 di halaman web.

**Private Functions**

Dengan menambahkan \_ (underscore) pada nama function

<?php

class Tampil extends CI\_Controller {

function index(){

echo 'Selamat Malaam!';

}

function \_testing($parameter){

echo $parameter;

}

}

?>

1. **View**

View adalah halaman web yang akan ditampilkan kepada user, disimpan di folder application/view

file : v\_tampil.php

<html>

<head>

<title>Testing 123</title>

</head>

<body>

<h1>Selamat Datang Boss!</h1>

</body>

</html>

**View dan Controller**

Penggabungan View dan Controller:

<?php

class Tampil extends CI\_Controller {

function index() {

echo 'Selamat Malaam!';

}

function testing($parameter){

echo $parameter;

$this->load->view(‘v\_tampil’);

}

}?>

**Menampilkan data dari Controller ke View**

Data disisipkan ke view dari controller menggunakan array dari object tampil.php

<?php

class Tampil extends CI\_Controller {

function index() {

echo 'Selamat Malaam!';

}

function testing($parameter){

$data[‘variabel’] = $parameter;

$this->load->view(‘tampil’, $data);

}

}?>

file : v\_tampil.php

<html>

<head>

<title>Testing 123</title>

</head>

<body>

<h1>Selamat Datang <?=$variabel?>!</h1>

</body>

</html>

1. **Model**

Model digunakan untuk melakukan aktivitas di database

Anatomi dari Model

class Model\_name extends CI\_Model {

function \_\_construct()

{

parent::\_\_construct();

}

}

**Load model ke aplikasi**

Bisa dilakukan di autoload atau di controller

$this->load->model('model\_name');

Contoh model:

class Tampil\_model extends CI\_Model {

function \_\_construct()

{

parent::\_\_construct();

}

function getDatabase(){

$this->db->select(‘\*’);

$this->db->from(‘Nama\_table);

$query = $this->db->get();

}

}

**Mempercantik URL Codeigniter**

Untuk menghilangkan kata index.php di url maka perlu dikonfigurasi file .htaccess yang ada di application dengan script sebagai berikut:

RewriteEngine on

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteCond %{REQUEST\_FILENAME} !-d

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

Kemudian Mengubah konfigurasi application/configs/config.php. Membuang "index.php" pada $config['index\_page'] = "index.php"; menjadi $config['index\_page'] = ""; di file

application/config/config.php

Dengan menyelesaikan tahap kedua anda telah bisa menghilangkan index.php pada url

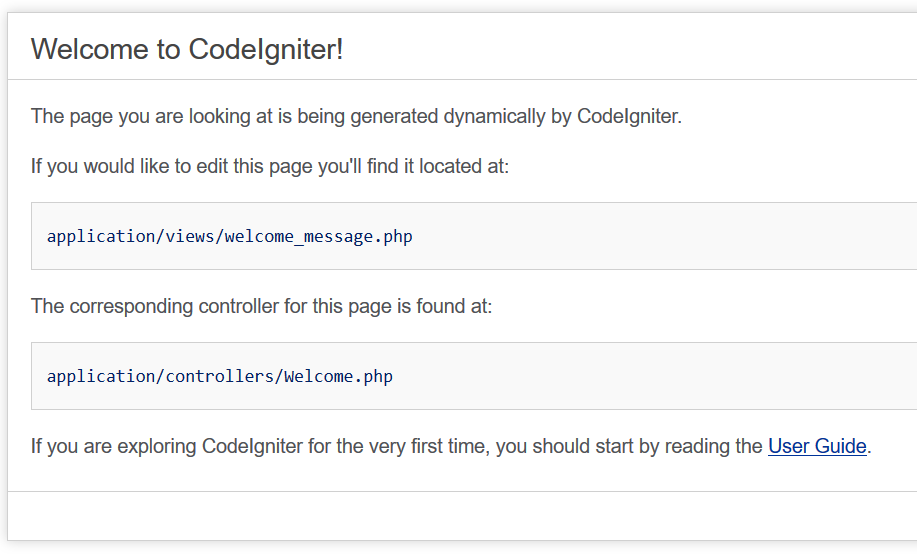
misalnya url http://localhost/rahmadi/index.php/latsatu/ sudah dapat diakses dengan url http://localhost/rahmadi/latsatu/ tanpa index.php

1. **Latihan**
   1. Buat website dengan menampilkan nama anda sendiri dengan CI!

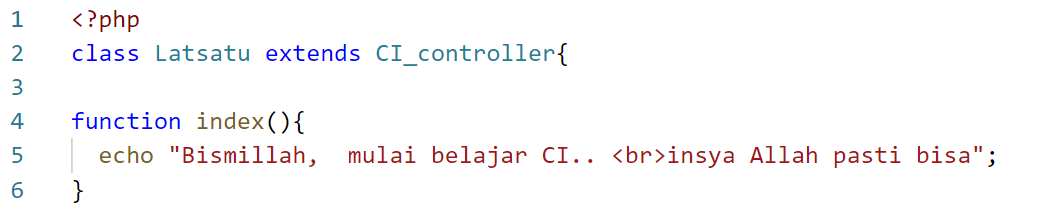
Pastikan instalasi framework CI anda berjalan dengan baik dengan perintah:

<http://localhost/rahmadi> --> contoh nama anda

Tampilan sebagai berikut:

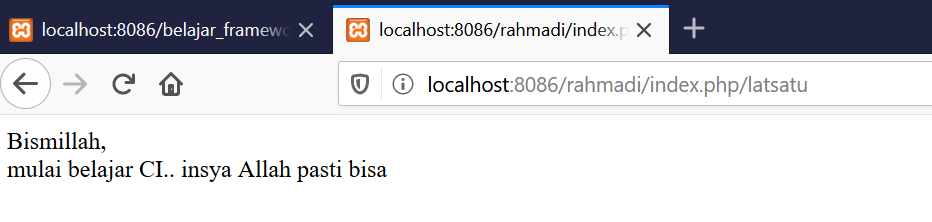


* 1. Buatlah file Latsatu.php di controller dengan script dibawah ini:

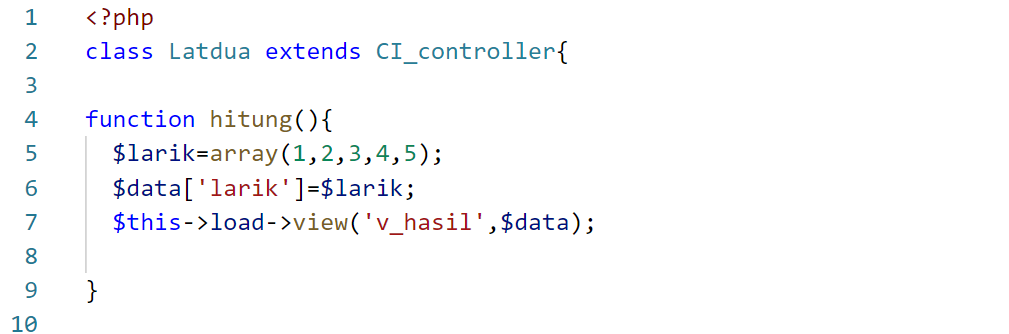


Dari url ketik perintah http://localhost/rahmadi/index.php/latsatu

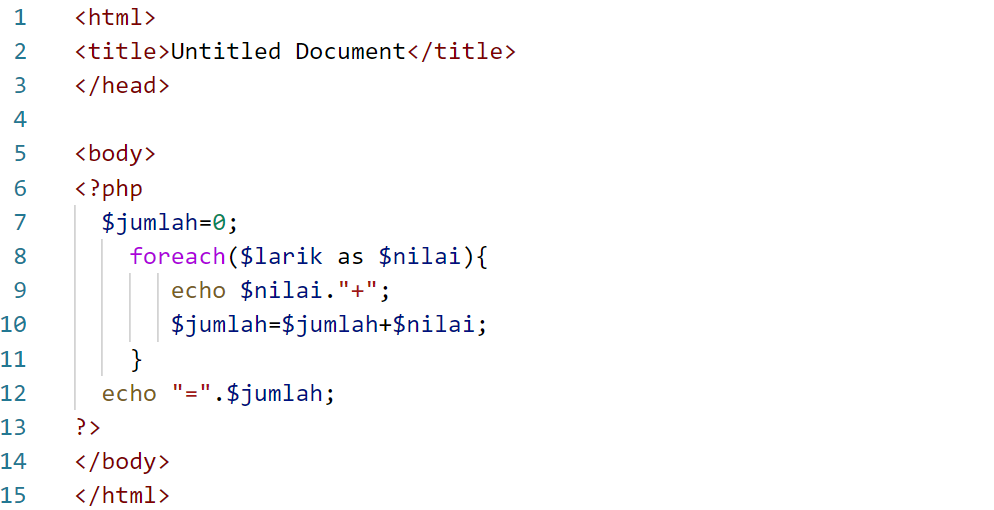
Hasilnya sebagai berikut:



* 1. Buatlah file **Latdua.php** di controller dengan script di bawah ini:

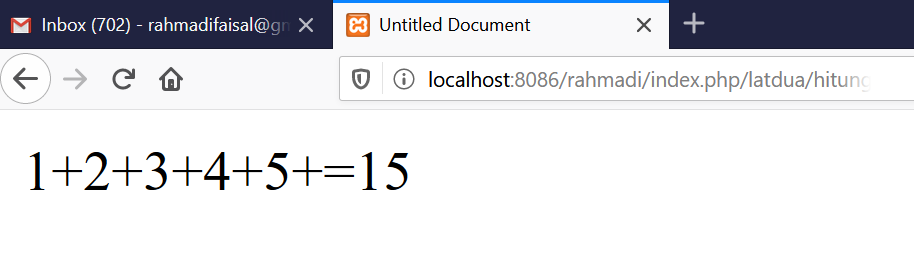


Buatlah file **v\_hasil.php** di views dengan script di bawah ini:



Dari url ketik perintah http://localhost/rahmadi/index.php/latdua/hitung

Hasilnya sebagai berikut:



1. **Soal Latihan**
   1. Buatlah program untuk menampilkan data array bertipe string yang diinisialisasi nilainya di class controller dengan tampilan melalui view.

Data array sebagai berikut:

|  |  |  |  |
| --- | --- | --- | --- |
| Jujur | Amanah | Adil | Rendah hati |

Ouputnya menjadi list sebagai berikut:

**Sikap** yang harus dimiliki oleh seorang **mahasiswa**:

1. Jujur
2. Amanah
3. Adil
4. Rendah Hati
   1. Buatlah program untuk menampilkan data array dua dimensi yang diinisialisasi nilainya di class controller dengan tampilan melalui view.

Data array sebagai berikut (baris pertama adalah index buat kolom):

|  |  |  |
| --- | --- | --- |
| **Pengarang** | **Judul** | **Harga** |
| Imam Adz Dzahabi | Siyar Alaam An – Nubalaa | 2000000 |
| Ibnul Qayyim | Zaadul Maad | 40000 |
| Ibnul Jauzy | Talbis Iblis | 85000 |

Outputnya sebagai berikut:

**Kitab-kitab Ulama Salaf yang perlu dibaca**:

1. Siyar Alaam An – Nubalaa, Imam Adz Dzahabi, Rp. 2.000.000,-
2. Zaadul Maad, Ibnul Qayyim, Rp. 40.000,-
3. Talbis Iblis, Ibnul Jauzy, Rp. 85.000,-