

Framework MVC (Model View Controller)

(Ultra Framework)

By : Yudi Purwanto

Kali ini saya akan membagikan sebuah framework ala yudi purwanto yang saya berinama Ultra Framework. Ya gara-gara iseng aja sebenarnya.

Aplikasi yang dibuat sangat sederhana namun memungkinkan kita untuk memahami cara kerjanya.

Ingat framework ini masih cukup sederhana dan sangat sederhana menurut saya sendiri karena iseng, lagi lagi iseng alasannya :D . tapi tidak ada salahnya saya berbagi dan bisa memudahkan pekerjaan anda, hobby anda dalam hal coding dan lain sebagainya. Yang pastinya ini Open Source jadi kembangkan sesuai kemampuan dan skill anda.

MVC pada awalnya banyak digunakan dalam aplikasi GUI dekstop, tetapi saat ini dengan perkembangan dan kematangan bahasa pemrograman web, bermunculanlah beberapa framework yang menggunakan arsitektur ini.

MVC, singkatnya adalah cara yang efisien, bersih dan cepat untuk memisahkan kode program dari kode HTML, atau antar muka lain yang kita pilih.

MVC berarti, ***Model – View – Controller***.

Model: bertanggungjawab terhadap pengelolaan data seperti SQL databases dan operasi SELECT, UPDATE, DELETE, dst, dll

View: menampilkan antarmuka (UI), html, css, javascript dan sebagainya

Controller: Jantung dan otak dari arsitektur MVC, tempat semua request masuk, **controller** adalah yang menjembatani antara **view** dan **model**.

Ultra Framework ini menggunakan Default Driver Database Mysql

Disini akan saya berikan Dokumentasi dan cara penggunaan Framework ini.

1. Langkah pertama download file yang ada di github.
2. langkah kedua konfigurasi di Database.php yang ada dalam root dir/base/Database.php
Setting nama semua yang ada di Database.php sesuaikan kebutuhan.
Seperti Host, Database, User, Password dan url

Penjelasan

- Host : nama host (eg: localhost)
- Database : nama database (eg: ultra)
- User : Nama user (eg: root)
- Password : Password (eg: root)
- url : Nama url root site (eg: http://domain.com / <http://localhost/folder>)

Oke kita masuk ketahap penggunaannya.
Disini saya memberikan contoh penggunaan dengan database.

Secara default sudah bisa berjalan dengan lancar jika diakses <http://domain.com> atau <http://localhost/folder> dan dicontrollers sudah ada file indexController.php dan testController.php.

Buat sebuah database dengan apa saja.
Lalu buat sebuah tabel dengan nama books

```
CREATE TABLE IF NOT EXISTS `books` (  
  `id` int(11) NOT NULL,  
  `author` int(11) NOT NULL,  
  `title` varchar(70) NOT NULL,  
  `isbn` varchar(20) DEFAULT NULL,  
  `publisher` varchar(30) NOT NULL,  
  `year` int(4) NOT NULL,  
  `summary` text,  
  `coverMime` varchar(20) DEFAULT NULL,  
  `coverImage` blob,  
  `copies` tinyint(4) NOT NULL DEFAULT '1'  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;
```

```
INSERT INTO `books` (`id`, `author`, `title`, `isbn`, `publisher`, `year`, `summary`, `coverMime`,  
  `coverImage`, `copies`) VALUES  
(1, 1, 'Creating your MySQL Database: Practical Design Tips and\r\nTechniques', '1904811302', 'Packt  
Publishing Ltd', 2006, 'A short guide for everyone on how to structure your data and\r\nset-up your MySQL  
database tables efficiently and easily.', NULL, NULL, 1),  
(2, 2, 'ImageMagick Tricks', '1904811868', 'Packt Publishing\r\nLtd', 2006, 'Unleash the power of  
ImageMagick with this fast, friendly\r\ntutorial, and tips guide', NULL, NULL, 1),  
(3, 3, 'Building Websites with Plone', '1904811027', 'Packt\r\nPublishing Ltd', 2004, 'An in-depth and  
comprehensive guide to the Plone content\r\nmanagement system', NULL, NULL, 1);
```

Model

Pembuatan model sangat mudah sebagai contoh seperti dibawah ini
Buat sebuah file didalam folder models dengan nama SiteModel.php dengan nama tabel books didatabase, terlebih dahulu setting Database.php di base/Database.php

```
<?php
/**
 * *****
 * @author : Yudi Purwanto
 * @link  : http://yudi-purwanto.com
 * @since : 14 May 2014
 * File   : SiteModel.php
 *
 *
 *
 *
 *
 *
 */
class SiteModel extends Model
{
    public function getBooks()
    {
        $list = $this->db->query("SELECT * FROM books");
        return $list->fetchall( PDO::FETCH_OBJ );
    }
}

/* End of file SiteModel.php */
/* Location: ./models/SiteModel.php */
```

Note : Untuk load sebuah model didalam sebuah controller dengan cara :
\$this->loadModel('NamaModel');

Contoh : \$this->loadModel('Site');

Controller

Kita bahas membuat sebuah controller disini saya contohkan testController.php yang mana sudah ada didalam folder controllers, nanti anda bisa mengakses <http://localhost/folder/test>

Membuat sebuah controller, sebagai contoh :

```
<?php
/**
 * *****
 * @author : Yudi Purwanto
 * @link   : http://yudi-purwanto.com
 * @since  : 14 May 2014
 * File    : testController.php
 *
 *
 * *****/
class testController extends Controller
{
    //<-- * Index Error * --->
    public function index() {
        header('HTTP/1.0 404 Not Found');
        include 'public/error/404.php'; // Show Error 404
    }

    public function test()
    {
        $output = $this->loadModel('Site');
        $this->view->title = "Testing controller";
        $this->view->list = $output->getBooks();
        /* Show Views */
        $this->view->render('index', null);
    }
}

/* End of file indexController.php */
/* Location: ./controllers/indexController.php */
```

View

View, seperti halnya membuat sebuah Model dan Controller, langsung saja kita ke langkah pembuatannya dari pembuatan Model dan Controller kita akan menampilkan dengan view ini. Simak contoh berikut.

Exstensi file view adalah .phtml jadi setiap membuat view dengan exstensi phtml

```
<!DOCTYPE html>
<html>
<head>
    <title><?php echo $this->title; ?></title>
    <link rel="stylesheet" type="text/css" href="<?php echo $layout['css']; ?>bootstrap.min.css">
</head>
<body>
<div class="container">

<nav class="navbar navbar-inverse">
    <div class="navbar-header">
        <a class="navbar-brand" href="<?php echo url; ?>">Crud</a>
    </div>
</nav>
<h1>All Data</h1>

<table class="table table-striped table-bordered">
    <thead>
        <tr>
            <td>Author and Title</td>
            <td>ISBN</td>
            <td>Publisher</td>
            <td>Year</td>
            <td>Summary</td>
            <td>Copies</td>
        </tr>
    </thead>
    <tbody>
        <?php
        $data = $this->list;
        foreach($data as $value):
        ?>
        <tr>
            <td><?php echo $value->title; ?></td>
            <td><?php echo $value->isbn; ?></td>
            <td><?php echo $value->publisher; ?></td>
            <td><?php echo $value->year; ?></td>
            <td><?php echo $value->summary; ?></td>
            <td><?php echo $value->copies; ?></td>
        </tr>
        <?php endforeach ?>
    </tbody>
</table>
</div>
</body>
</html>
```

Router

Pada dasarnya jika anda membuat sebuah Controller, anda harus mendaftarkan ke router.php di base/Router.php, dengan cara sebagai berikut.

Perhatikan baik baik, bisa dibilang susah susah gampang disaat mendaftarkan controller ke router.

```
<?php
/**
 * *****
 * @author : Yudi Purwanto
 * @link : http://yudi-purwanto.com
 * @since : 14 May 2014
 * File : Router.php
 *
 *
 *
 *
 */
class Router
{
    private $_controller;
    private $_method;

    public function __construct() {
        $page = $_GET['test']; //Test eg: http:website.com/test dan jangan lupa setting htaccess

        /* function method controller */
        $staticPages = array(
            'test'
        );

        //<-- * Pages Default * -->
        if( !in_array( $page, $staticPages )
            && isset( $page )
        ) {
            $this->_controller = strtolower( $page );
            $this->_method = strtolower( $page );
        }
        //<--- ***** Pages ***** --->
        else if( isset( $page )
            && in_array( $page, $staticPages )
        ) {
            //load nama controller adalah test = load nama folder test di dalam folder views, jadi harus sama
            $this->_controller = 'test';
            $this->_method = strtolower( $page );
        }

        //<-- * if no have variables defined $ _GET showed index * -->
        if( !$this->_controller ) {
            $this->_controller = default_controller;
        }
    }
}
```

```

        if( !$this->_method ) {
            $this->_method = 'index';
        }

    } //<--- * End * -->

    public function getController() {
        return $this->_controller;
    }

    public function getMethod() {
        return $this->_method;
    }

} //<-- * End Class * -->

/* End of file Router.php */
/* Location: ./base/Router.php */

```

Untuk file htaccessnya anda bisa menambahkan seperti ini :

```

# Test
RewriteRule ^([a-zA-Z0-9-_]+)$ index.php?test=$1 [QSA,L]

```

Selamat mencoba...!!!