

# Tutorial Zend bagian 1 - 3

**Wirawan Prasetyo**

*Bestfriends\_wp@yahoo.com*

*http://www.wirawanprasetyo.web.id*

## **Lisensi Dokumen:**

*Copyright © 2003-2007 IlmuKomputer.Com*

*Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.*

Zend Framework merupakan salah satu framework php yang saat ini banyak digunakan. Pada saat awal pembelajaran akan sedikit sulit karena banyaknya hal yang harus dipahami. Namun, dibalik itu Zend sudah mendukung berbagai API yang disediakan oleh website2 besar.

Zend framework sudah mendukung penggunaan MVC dan php5. Sedangkan database yang sudah didukungnya : MySQL, Oracle, Firebird, SQLServer dll. Sedangkan ser vice yang didukung : Flickr, Amazon, Technorati, Yahoo dan Akismet.

Jadi buat kamu yg tertarik untuk mencoba framework ini atau memang membutuhkan framework ini, silahkan melanjutkan membaca.

Tetapi kalau kamu mau membuat aplikasi dan sudah memiliki requirement tertentu, ada baiknya kamu baca dulu dokumentasi. Apakah sesuai dengan kebutuhan kamu. <http://framework.zend.com/>

Atau baca artikel mengenai framework2 di php

- [wikipedia](#)
- [phpit.net](#)

## 1. Kebutuhan dasar

Disini gua menggunakan XAMPP 1.6.5 sebagai server dengan spek :

- PHP v 5.2.5
  - o Konfigurasi *httpd.conf* ( *root/xampp/apache/conf/httpd.conf* )
    - LoadModule *rewrite\_module* *modules/mod\_rewrite.so* ( **tanpa #** )
    - Allowoverwrite All
- MySQL 5.0.51

- OS XP
- File akan aku letakkan pada "*instalationPath*/xampp/htdocs/Root".  
Jadi "index.php" berada di  
"*instalationPath*/xampp/htdocs/Root/index.php"

File asli dari framework ini, bisa di download di <http://framework.zend.com>. Didalam file tersebut, sudah ada dokumentasinya. Sedangkan di webnya, juga tersedia video tutorial .

Oia, aku menggunakan versi 1.5.2

### **Editor**

Editor yg digunakan bisa bebas, notepad juga bisa digunakan. Namun, perusahaan zend juga membuat Zend Studio. Kelebihan dari editor ini, kita dapat melihat dimana sebuah klass yg kita gunakan dideklarasikan ( tapi bayar loh ).

Kalau di php, sepertinya belum ada aplikasi seperti ini. Namun, di java bisa disamakan dengan NetBeans lah.

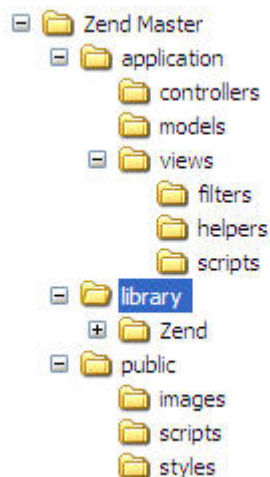
Sedangkankan kemampuan dasar yang diperluan adalah kemampuan menggunakan php secara dasar seperti variabel, perulangan dan array. Kita menggunakan database MySQL dengan GUI phpMyAdmin.

## 2. Struktur Folder

Ketika kita menggunakan suatu framework. Maka kita wajib mengikuti aturan yg dibuat oleh penulis framework.

Setelah kamu meng-unduh frameworknya, kemudian ekstraklah ke suatu tempat. Kemudian kopi filenya ke folder "Root". Didalamnya sudah ada :

- /Application
  - o Controller -> disini tempat melekkan controller ( logika )
  - o Model -> disini tempat melekkan model
  - o View/script -> disini tempat melekkan view( templat )
- /Library -> tempat menyimpan library zend
- /public -> bisa untuk meletakkan css kita



### 3. Hello Word dan Form Sederhana

#### 3.1. Hasil Akhir

Hasil Akhir yang akan kita dapatkan pada akhir pengerjaan tutorial ini adalah bagaimana cara menggunakan framework ini kemudian membuat hello word sampai dengan membuat form sederhana untuk menampilkan hasil inputan user

#### 3.2. Kisi-Kisi

Klas-klas dari zend yang akan digunakan adalah :

- Zend\_Controller
  - Ini merupakan klas utama yang menunjukkan bahwa kita menggunakan MVC yang disediakan oleh framework ini

#### 3.3. Membuat htaccess dan index.php (bootstraping)

Kali ini kita akan membuat suatu file yg fungsinya mencegah user masuk ke dalam folder kita dengan mengetikan nama folder di dalam url. Sehingga, setiap user akan kita alihkan ke halaman index root saja.

Buatah “.htaccess “ file dengan isi, letakan di root aplikasi kamu.  
( missal di folder : Root )

```
# Root\.htaccess
```

```
RewriteEngine On  
RewriteRule .* index.php
```

Disini terlihat, apache akan selalu mengalihkan request kita ke index.php terlebih dahulu.

Kemudian index.php akan memanggil class2 yg dibutuhkan framework dan mengeksekusi perintah yg ada di url. Ini kenapa dinamakan bootstraoping

Sedangkan index.php kita isi dengan :

```
# Root\index.phtml
```

```
<?php

error_reporting(E_ALL|E_STRICT);
ini_set('display_errors', true);
date_default_timezone_set('Asia/Jakarta');

$rootDir = dirname(dirname(__FILE__));

set_include_path('.' . PATH_SEPARATOR . 'library'
    . PATH_SEPARATOR . './application/models/'
    . PATH_SEPARATOR . get_include_path());
include "Zend/Loader.php";
Zend_Loader::loadClass('Zend_Controller_Front');

// setup controller
$frontController = Zend_Controller_Front::getInstance();
$frontController->throwExceptions(true);
$frontController->setControllerDirectory('./application/controllers');

// run!
$frontController->dispatch();
?>
```

Penjelasan :

- Ini\_set ... -> untuk menampirkan pesan kesalahan di browser kita
- Date .... -> jelaslah
- require\_once 'Zend/Controller/Front.php -> inilah file yg menerangkan kita menggunakan framework Zend
- include "Zend/Loader.php"; -> ini class yg mempermudah kita menggunakan class lainnya
- Zend\_Loader::loadClass('Zend\_Controller\_Front'); -> kita menggunakan klas Controller

### 3.4.Kontroller, Aksi ,View dan Variabel

Ketika kita mengetik / membuka suatu url ke aplikasi yg menggunakan framework Zend, sebenarnya kita mengakses sebuah controller dan aksi di dalam controller tersebut. Dan juga akan mengirim variable dan nilai variable ( jika ada ).

Secara default, kalau kita tidak menulis controller yg kita tuju. Maka dianggap kita akan membaca controller *index*. ( **akan disebut index controller** ). Setiap controller diletakkan dalam folder "Root\application\controllers\controllerNameController.php".

Dan jika kita juga tidak menulis nama aksinya, maka secara default itu berarti kita membaca aksi *index*. ( **akan disebut index action di index controller** )

#Root\application\controllers\IndexController.php

```
<?php
class IndexController extends Zend_Controller_Action
{
    function indexAction()
    {}
}
?>
```

Sedangkan jika tidak ada variabel ... yah tidak ada nilai defaultnya.

Sebagai contoh, <http://localhost/Root> sebenarnya kita menuju <http://localhost/Root/index> ( **index** kontroler).

Karena juga tidak ada nama aksinya, jadi kita menuju <http://localhost/Root/index/index> ( **index aksi di index** kontroler). Jika ingin mengirim variable, maka urlnya menjadi : <http://localhost/Root/namaKontroler/namaAction/namaVariabel1/nilaiVariabel1/namaVariabel2/nilaiVariabel2>.

Jika mau mengirim variabel, nama kontroler dan aksi harus ditulis.

Setelah mengetahui nama kontroler, Aksi dan Variabel yg dikirim. Kita memerlukan satu halaman yg berfungsi menampilkan outputnya ke HTML. Ini disebut dengan View. View disimpan di : "Root\application\views\scripts\namaKontroler\namaAksi.phtml".

Jadi setiap kontroler punya satu folder dan setiap aksi memiliki tampilan sendiri (aksi.phtml) yang diletakkan dalam folder kontroler. Ini akan disebut **view nama aksi di nama controller**.

Sebagai contoh, <http://localhost/Root> akan membutuhkan sebuah view index di "Root\application\views\scripts\index\index.phtml".

Mudah lah .....

### 3.5.Hello word

Hehe ... ini kan kata sakti kita setiap kali mempelajari bahasa baru. Ya ndak. Kita akan membuat suatu halaman untuk menginput nama dan mood kita saat ini.

Dan sebagai contoh, buatlah file index controller terlebih dahulu :

#### 3.5.1. Indeks Kontoler

```
#Root\application\controllers\IndexController.php
<?php
class IndexController extends Zend_Controller_Action
{
    function indexAction()
    {}
}
?>
```

### 3.5.2. Indeks View

```
# Root\application\view\script\index\index.phtml
```

```
<?php  
    Echo "Hello Word";  
?>
```

Jalankan / aktifkan web server kamu, kemudian buka browser kesayangan kamu. Di url ketikan : <http://localhost/Root>. Maka browser kamu akan menampilkan " Hello Word" ( pastinya tanpa kutip dua yah ).

Mudah kan .....

### 3.6.Header ,footer dan css

Header dan footer merupakan komponen yg biasanya ada di suatu halaman. Header biasanya berisi logo dan footer berisi copyright dan lain lain.

File header dan footer kita letakkan di  
"Root\application\views\scripts\namafile.phtml"

Untuk menambahkan header dan footer, pada file view kita menambahkan script "<?php echo \$this->render('namafile.phtml'); ?>"

Sedangkan file style, kita letakkan di  
"Root\public\styles\namafilecss.css". Dan di view kita menambahkan  
"<link rel="stylesheet" type="text/css" media="screen" href="<?php echo \$this->baseUri;?>/public/styles/site.css">"

Sebagai contoh, file "Root\application\view\script\index\index.phtml" , kita tambah menjadi : ( [tambahan diberi warna biru](#) )

```
#Root\application\view\script\index\index.phtml
```

```
<?php echo $this->render(header.phtml'); ?>  
Hello Word
```

```
<? echo $this->render("footer.phtml"); ?>
```

Pada file controller index, kita akan membuat fungsi init

#Root\application\controllers\IndexController.php

```
<?php
class IndexController extends Zend_Controller_Action
{
    function init()
    {
        $this->initView();
        $this->view->baseUrl = $this->_request-
>getBaseUrl();
    }
    function indexAction()
    {}
}
?>
```

### Penjelasan :

- Fungsi init pada (Root\application\controllers\IndexController.php) merupakan fungsi pertama yg dijalankan ketika kita membuat class baru.
- `$this->view->baseUrl = $this->_request->getBaseUrl()` menerangkan root dari url kita
- Kita membuat file header.phtml dan footer.phtml yg kemudian di gunakan di halaman index.phtml ( `echo $this->render ("header.phtml")` ). Ini mirip seperti `includes("namafile.php")`;
- Membuat file style sheet. "`<?php echo $this->baseUrl;?>/` " akan menampilkan url kita yg sebenarnya.

Kemudian, kita membuat file header.phtml

#Root\application\view\script\header.phtml



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
lang="en">

<head>

<meta http-equiv="Content-Type" content="text/html;charset=utf-
8" />

<title>My root Title</title>

<link rel="stylesheet" type="text/css" media="screen"
href="<?php echo $this->baseUrl;?>/public/styles/site.css" />

</head>

<body>

<div id="headerRoot">Header root

<hr>

</div>

<div id="isiRoot">
```

Membuat file footer.phtml

#Root\application\view\script\footer.phtml

```
</div>

<div id="footerRoot">

<hr>

My footer copyright

</div>

</body>

</html>
```

Membuat file style

# Root\public\styles\site.css

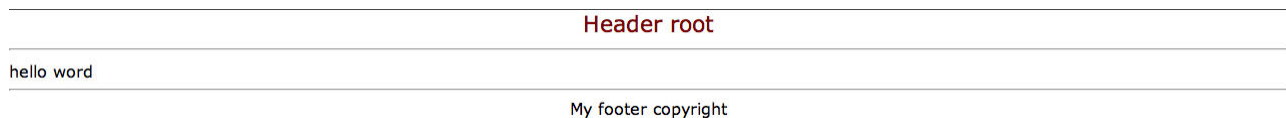
```
#headerRoot

{
```

```
text-align : center;
font-size:1.4em;
color: #800000;
}

#footerRoot
{
text-align : center;
}
```

Refresh browser kamu, tampilan hello word akan berubah



### 3.7. Menampilkan nilai suatu variabel di controller ke dalam view

Suatu halaman dinamis, biasanya diisi dengan tampilan yg berubah. Kali ini saya akan menampilkan "title" web yg dikirim dari controller ke view

```
# Root\application\controllers\IndexController.php
```

```
<?php
class IndexController extends Zend_Controller_Action
{
    function init()
    {
        $this->initView();
        $this->view->baseurl = $this->_request-
>getBaseUrl();
    }

    function indexAction()
```

```
{      $this->view->title = "Root saya";  
      $this->view->isi = "Ini isi halaman saya";  
      $this->render();  
    }  
  }  
?>
```

Pada file, file header.phtml  
(Root\application\view\script\header.phtml). "`<title>My root Title</title>`" diganti menjadi "`<title><?php echo $this->title; ?></title>`".

Pada file index.phtml

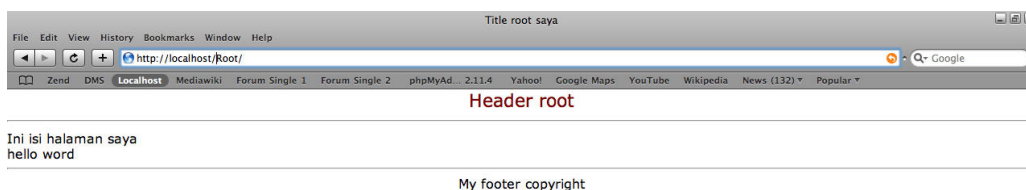
# Root\application\view\script\index\index.phtml

```
<?php echo $this->render('header.phtml'); ?>  
  
<?php echo $this->isi; ?><br>  
  
hello word  
  
<?php echo $this->render('footer.phtml'); ?>
```

### Penjelasan :

- "`$this->view->isi = "Ini isi halaman saya";`" pada `indexcontroller.php` menerangkan kita mengirim sebuah variabel bernama "`isi`" dengan isi "`Ini isi halaman saya`" ke tampilan(view)
- Kita juga dapat menggunakan "`$this->view->assign('isi',' Ini isi halaman saya ');`"
- "`<?php echo $this->isi; ?>`" pada `index.phtml` menerangkan kita akan menampilkan variabel dengan nama "`isi`"

Refresh kembali browser kamu ... berubah



### 3.8. Form Sederhana

Sebagai contoh, kita akan membuat sebuah form sederhana di index.phtml yang meminta kita memasukkan nama kita kemudian akan ditampilkan di read.phtml.

Karena itu, kita juga memerlukan sebuah aksi baru , bernama read.

Pada file index.phtml

#Root\application\view\script\index\index.phtml )

```
<?php echo $this->render('header.phtml'); ?>
<?php echo $this->isi; ?><br>
hello word
<form action="<?php echo $this->baseUrl ?>/index/read"
method="post">
Masukkan nama kamu : <input type="text"
name="txtNama"><br>
Masukkan mood kamu : <input type="text" name="txtMood">
<input type="submit" value="Masuk">
</form>
<?php echo $this->render('footer.phtml'); ?>
```

Membuat file baru, read.phtml

#Root\application\view\script\index\read.phtml

```
<?php echo $this->render('header.phtml'); ?>
<?php echo $this->isi; ?><br>
Selamat datang <?php echo $this->nama; ?>, saat ini mood
kamu <?php echo $this->mood; ?>
<?php echo $this->render('footer.phtml'); ?>
```

Pada Index Controller (Root\application\controllers\IndexController.php), membuat aksi baru (function readAction()). " *function indexAction() dan function init()*" sengaja tidak ditampilkan untuk menghemat baris tapi di dalam aplikasi tetap ada.

```
<?php
class IndexController extends Zend_Controller_Action
{
    .....
    function readAction()
    {
        $request = $this->getRequest();
        $this->view->assign('nama', $request-
>getParam('txtNama'));
        $this->view->assign('mood', $request-
>getParam('txtMood'));
        $this->view->isi = "Mood kamu hari ini";
        $this->render();
    }
?>
```

Kemudian masukkan data kamu ke form di index(<http://localhost/Root>), kemudian tekan tombol masuk. Munculkan ...

### Penjelasan :

#### a) Pada index view

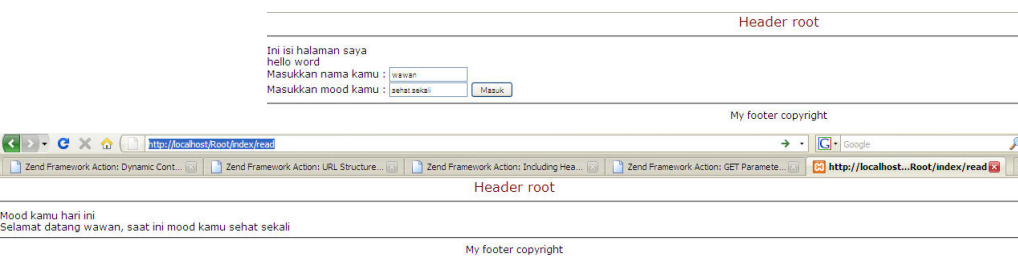
- Kita membuat form sederhana dengan field nama dan mood.
- Action kita arahkan ke controller index dengan function **read** (`<form action="<?php echo $this->baseUrl ?>/index/read" method="post" >`)

#### b) Pada function read action index controller

- Disini kita membuat aksi read sesuai dengan link form diatas.
- `$request = $this->getRequest();` => kita menampung semua variable yg dikirim ke \$request
- `$this->view->assign('nama', $request->getParam('txtNama'));` Kiat mengambil nilai variable satu persatu (`txtNama`) untuk ditampilkan di web kita

c) Pada read view

- Menampilan view yg dikirim



### 3.9. Kesimpulan

Mungkin sampai saat ini masih terlihat sulit ketika akan mengembangkan suatu aplikasi web. Tapi mari di telaah kembali, file **controller** (Root\application\controllers\IndexController.php) dan file **view** (Root\application\view\script\index\index.phtml).

Pada file controller terlihat rumit, namun di file view terlihat mudah. Hanya memerlukan beberapa baris saja. Efeknya, ketika seorang web designer akan melakukan perubahan, dia tidak akan kesulitan. Karena tidak perlu berurusan dengan file php yg begitu banyak.

Begitu juga ketika programmer melakukan coding, dia tidak perlu terlalu memikirkan tampilannya.

#### Source Code:

File yang diikut sertakan hanya berupa hasil akhir. Kamu perlu terlebih dahulu mendownload frameworknya secara online kemudian ekstrak ke htdocs. Kemudian letakkan file source code dengan menimpa folder app dan public.

### Biografi Penulis



**Wirawan Prasetyo** (Wawan) lahir pada bulan Desember tahun 1984. Menyelesaikan S1 di Universitas Bina Nusantara, Jakarta jurusan Teknik Informatika pada tahun 2007. Saat ini berkerja di sebuah media hukum online ( [www.hukumonline.com](http://www.hukumonline.com) ) sebagai programmer.

<http://www.wirawanprasetyo.web.id>