



Pemrograman Web

# Materi

- What is Frameworks ?
- MVC Pattern
- Instalasi dan Konfigurasi CodeIgniter
- Creating Simple URL Shortener

# What is Frameworks ?



**matakuliah.php**



**login.php**

```
<?php
    $conn = mysql_connect('localhost', 'root', 'rootpassword');
    $db = mysql_select_db('perwalian', $conn);
    $sql = 'select * from matakuliah order by kodematakuliah';
    $query = mysql_query($sql);
    while ($row = mysql_fetch_assoc($query)) {
        //display data in a table
    }
?>
```

# What is Frameworks ?



matakuliah.php



login.php

```
<?php
$username = $_POST['user']; $password = $_POST['pass'];
$conn = mysql_connect('localhost', 'root', 'rootpassword');
$db = mysql_select_db('perwalian', $conn);
$sql = 'select * from user where username = $user' and password = $pass;
$query = mysql_query($sql);
if(mysql_num_rows($query) == 1)
    echo 'Login OK';
?>
```

# What is Frameworks ?



matakuliah.php



login.php



dbconnect.php

```
<?php
$dbhostname = 'localhost';
$dbusername = 'root';
$dbpassword = 'rootpassword';
$dbname = 'perwalian';
$conn = mysql_connect($dbhostname, $dbusername, $dbpassword);
$db = mysql_select_db($dbname, $conn);
?>
```

# What is Frameworks ?



matakuliah.php



login.php



dbconnect.php

```
<?php
include("dbconfig.php");
$sql = 'select * from matakuliah order by kodematakuliah';
$query = mysql_query($sql);
while ($row = mysql_fetch_assoc($query)) {
    echo $row['kodematakuliah'] . ' | ' . $row['namamatakuliah'];
}
?>
```

# What is Frameworks ?



```
<?php
include("dbconfig.php");
$username = $_POST['user']; $password = $_POST['pass'];
$sql = 'select * from user where username = $user and password = $pass';
$query = mysql_query($sql);
if(mysql_num_rows($query) == 1)
    echo 'Login OK';
?>
```

# Frameworks

- Lots of chunks of code, stored in separate files, which simplify the coding of **repetitive operations**
- Repetitive operations:
  - Database (insert, retrieve, update, delete data)
  - Input handling (input checking, reading input)
  - Session Handling (create session, check session)
  - Many more...



# PHP Frameworks



# Frameworks

- Save Time (less code)

```
$connection = mysql_connect("localhost","progweb","12345");
mysql_select_db("progweb_a-1", $connection);
$result = mysql_query ("SELECT * FROM mahasiswa",
    $connection);
while ($row = mysql_fetch_array($result, MYSQL_NUM))
{
    foreach ($row as $attribute)
        print "$attribute['nama']";
}
```

```
$this->load->database('progweb_a-1');
$query = $this->db->get('mahasiswa');
foreach ($query->result() as $row)
{
    print $row->nama
}
```

# Frameworks

- Separation between logic, view and data
- Coding standard, consistent across files
- Attractive URL
- All simple (tedious and boring) things are already handled
  - Input validation
  - Managing database
  - more...

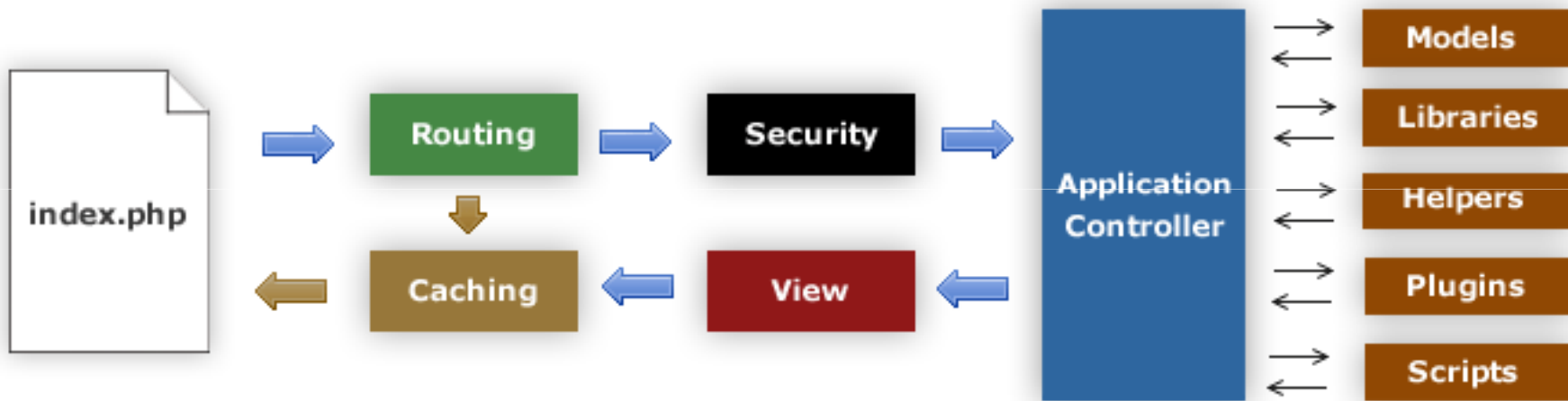
# CodeIgniter

## CodeIgniter is right for you if...

- You want a framework with a small footprint.
- You need exceptional performance.
- You need broad compatibility with standard hosting accounts that run a variety of PHP versions and configurations.
- You want a framework that requires nearly zero configuration.
- You want a framework that does not require you to use the command line.
- You want a framework that does not require you to adhere to restrictive coding rules.
- You are not interested in large-scale monolithic libraries like PEAR.
- You do not want to be forced to learn a templating language (although a template parser is optionally available if you desire one).
- You eschew complexity, favoring simple solutions.
- You need clear, thorough documentation.

<http://www.codeigniter.com>

# CodeIgniter

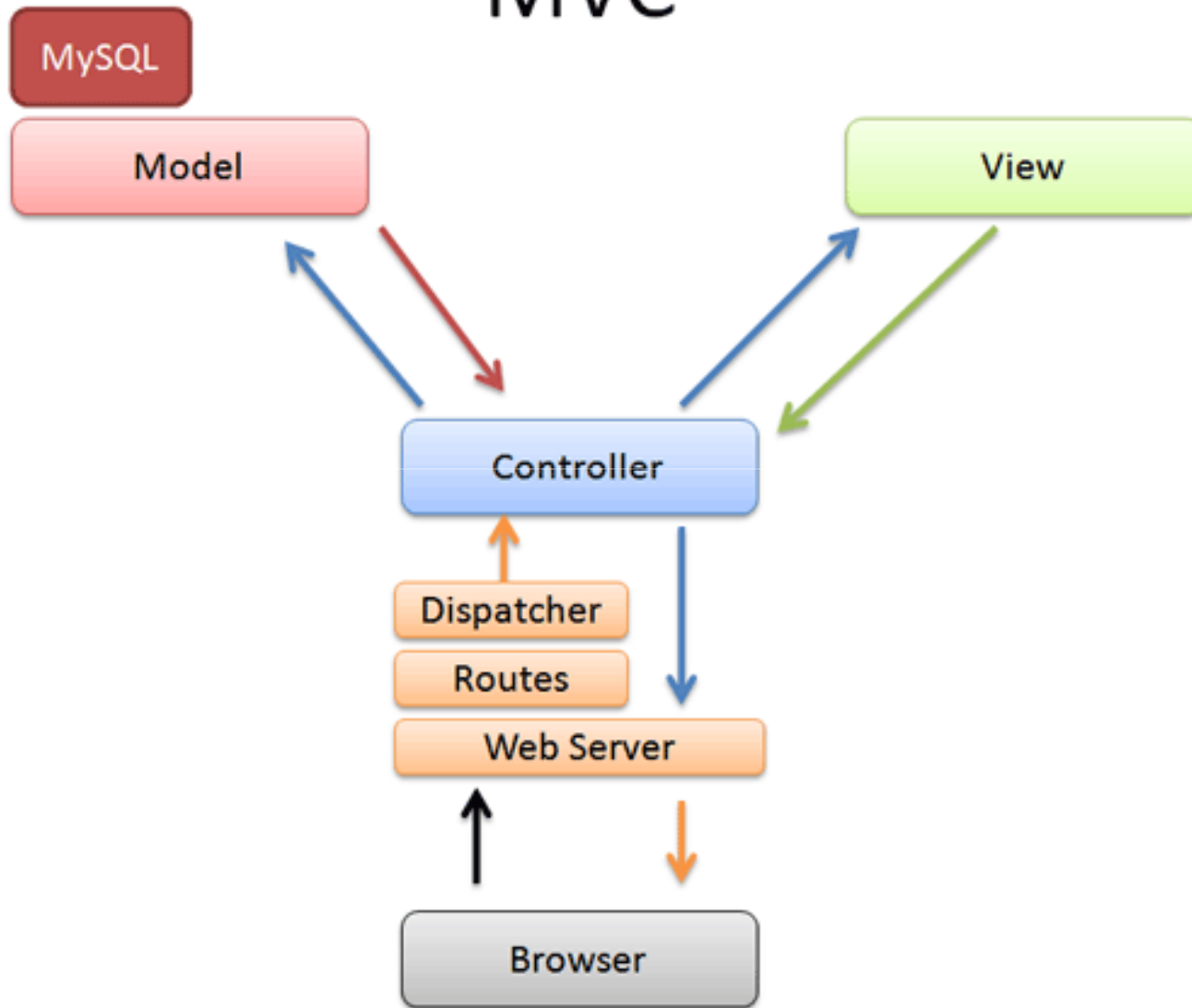


# **MVC PATTERN**

# M + V + C

- MVC = Model View Controller
- Aplikasi dibagi menjadi 3 bagian dengan fungsi berbeda
  - Model
  - View
  - Controller

# MVC





# Model

- Data structures
- **Data**
  - Database
  - Internal Resources (eg: files, rules, logs)
  - External Resources (eg: tweets, rss feeds, XML)

# View

- **User Interface**
- Displaying data (How data is displayed)
- **Interacting with user**
- Collect input

# Controller

- Bussines Logic / Application Logic
- **Process** input from View
- Model (Data) apa yang harus diambil dan harus ditampilkan pada view yang mana

# MVC ?

- Proses Login
- Menampilkan daftar mata kuliah yang bisa diambil
- Mengubah password
- Menghapus beberapa mata kuliah yang sudah diambil sebelumnya

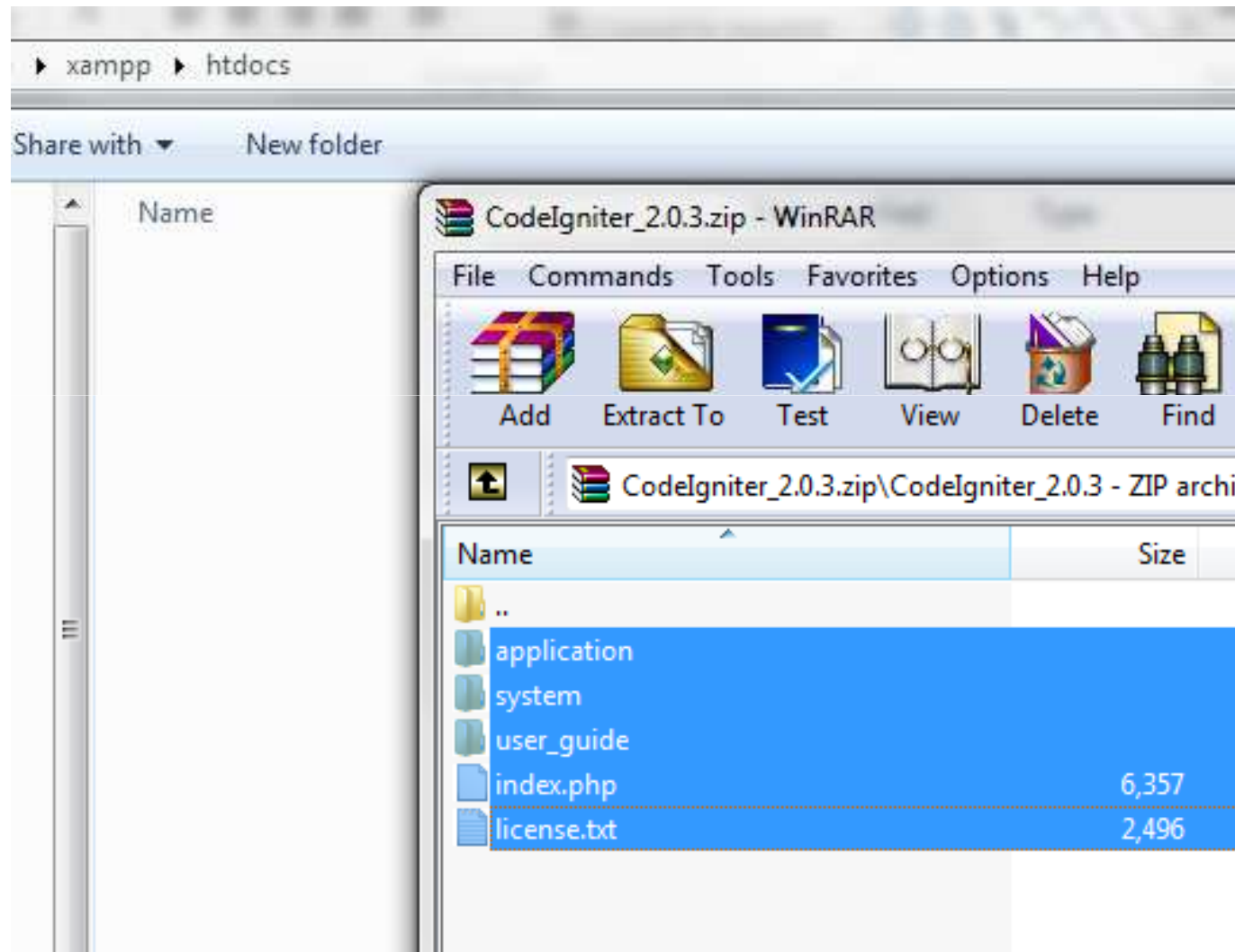
# **INSTALASI DAN KONFIGURASI CODEIGNITER**

# Download

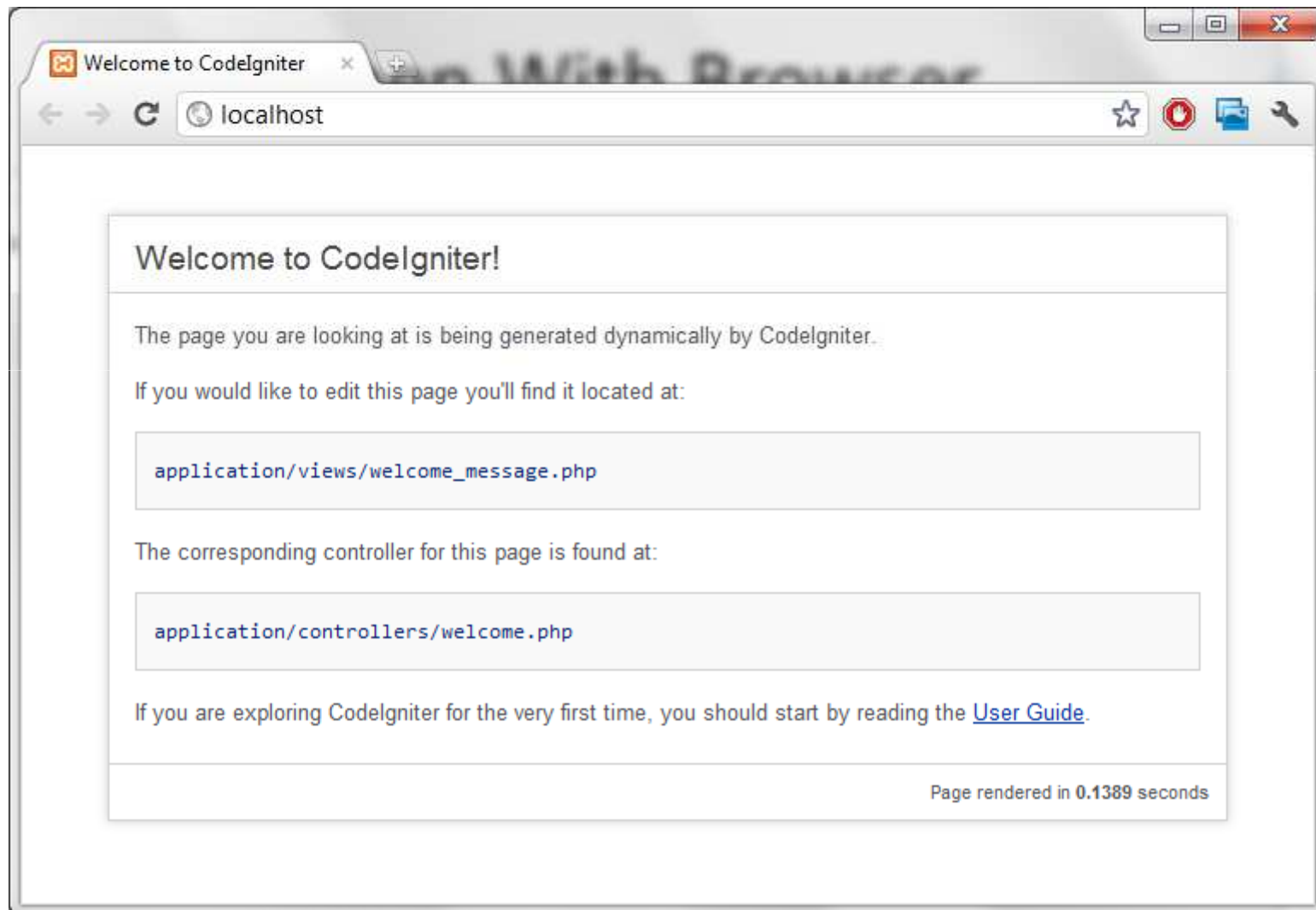


The screenshot shows the CodeIgniter website homepage. At the top left is the CodeIgniter logo, a stylized flame icon. To its right is the text "CodeIgniter®". In the top right corner, there are links for "Login" and "Register". Below the logo is a navigation menu with the following items: "Home", "Downloads", "News", "VideoTutorials", "User Guide", "Forums", "IRC", "Wiki", and "Bug Tracker". The main content area features a large banner with a flame background. On the left side of the banner is the CodeIgniter logo. To its right, the text reads: "CodeIgniter is an Open Source Web Application Framework that helps you write incredible PHP programs". On the right side of the banner, there are four links with icons: "Download CodeIgniter Version 2.6.3" (with a green download arrow icon), "Read the User Guide CodeIgniter is clearly documented" (with a document icon), "View the Video Tutorials Build a weblog in under 20 minutes!" (with a video icon), and "Talk About it in the Forums Join the growing community" (with a microphone icon). Below the banner, there is a section titled "Welcome to CodeIgniter!" followed by a paragraph of text: "CodeIgniter is a powerful PHP framework with a very small footprint, built for PHP coders who need a simple and elegant toolkit to create full-featured web applications. If you're a developer who lives in the real world of shared hosting accounts and clients with deadlines, and if you're tired of ponderously large". To the right of this section is a "CODEIGNITER NEWS" section with the heading "New User Guide in Development" and a link "Read More...".

# Extract

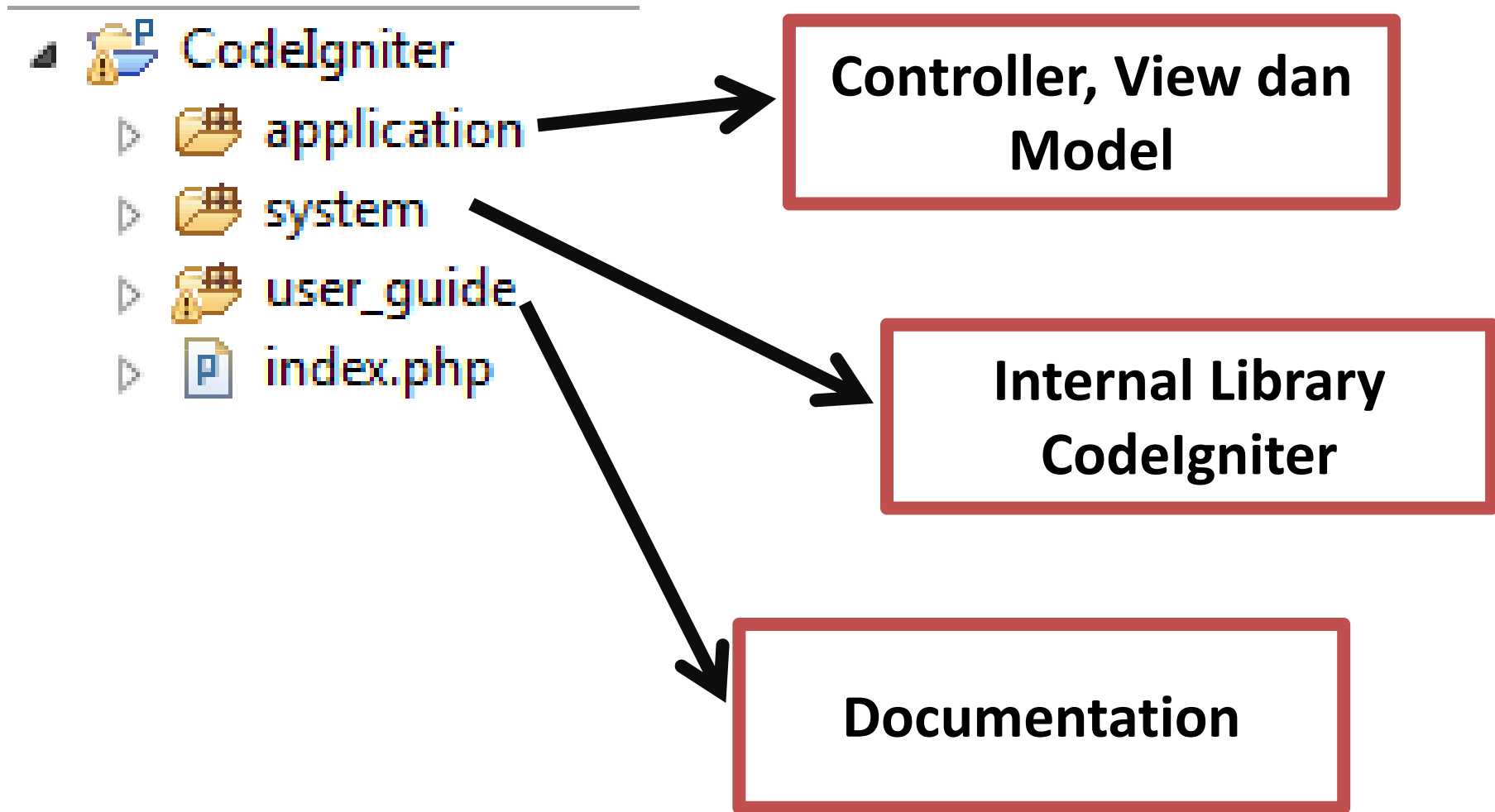


# Test With Browser

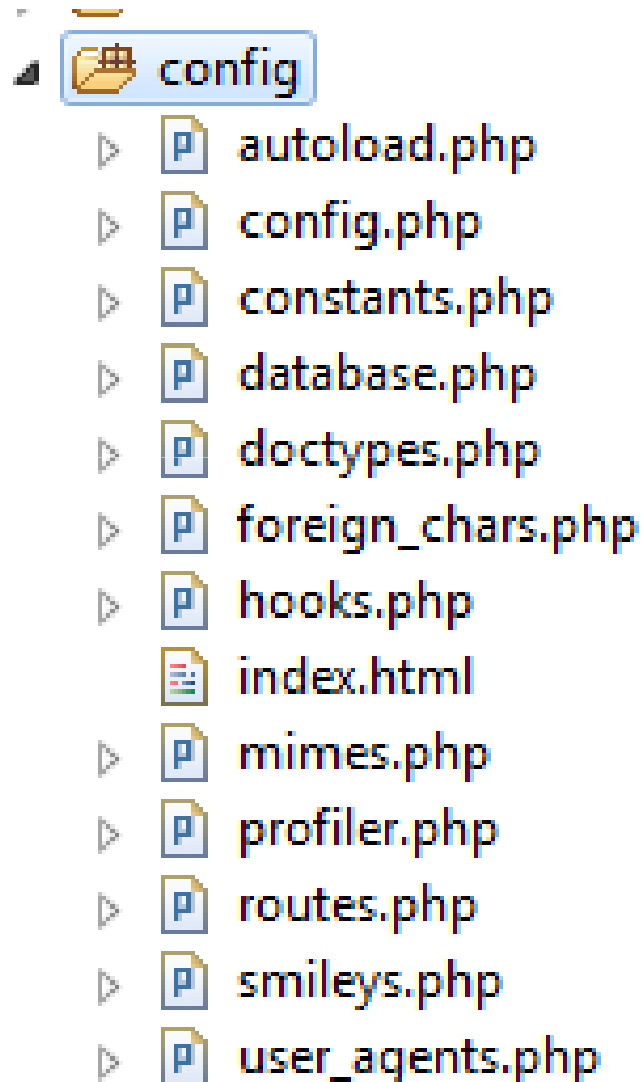




# Structure



# Configuration



# Configuration: config.php

- General Settings
  - Base URL
  - Index page
  - URL suffix
  - Language
  - Character set
  - Log
  - Session
  - etc...

Letak di folder: ***application/config/config.php***

# Configuration: autoload.php

- Konfigurasi modul-modul yang akan **diload secara otomatis**
  - Packages
  - Libraries
  - Helper files
  - Custom config files
  - Language files
  - Models

# Configuration: database.php

Letak di folder: *application/config/database.php*

- Database settings

```
$active_group = 'default';
$active_record = TRUE;

$db['default']['hostname'] = 'localhost';
$db['default']['username'] = '';
$db['default']['password'] = '';
$db['default']['database'] = '';
$db['default']['dbdriver'] = 'mysql';
$db['default']['dbprefix'] = '';
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = '';
$db['default']['char_set'] = 'utf8';
$db['default']['dbcollat'] = 'utf8_general_ci';
$db['default']['swap_pre'] = '';
$db['default']['autoinit'] = TRUE;
$db['default']['stricton'] = FALSE;
```

# CodeIgniter Frameworks : Helper

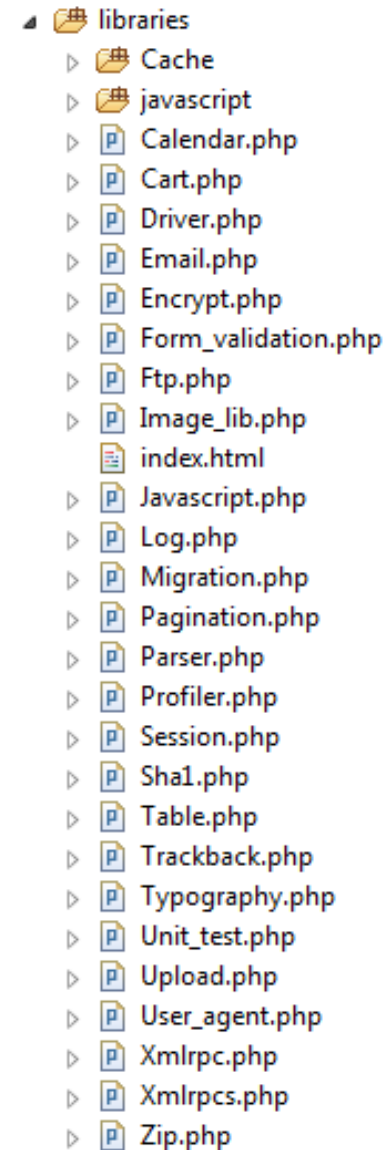
- Collection of functions
- Tidak berupa Class, hanya kumpulan fungsi
- Helper diletakkan pada **`/system/helpers`**

## Helper Reference

- [Array Helper](#)
- [CAPTCHA Helper](#)
- [Cookie Helper](#)
- [Date Helper](#)
- [Directory Helper](#)
- [Download Helper](#)
- [Email Helper](#)
- [File Helper](#)
- [Form Helper](#)
- [HTML Helper](#)
- [Inflector Helper](#)
- [Language Helper](#)
- [Number Helper](#)
- [Path Helper](#)
- [Security Helper](#)
- [Smiley Helper](#)
- [String Helper](#)
- [Text Helper](#)
- [Typography Helper](#)
- [URL Helper](#)
- [XML Helper](#)

# CodeIgniter Frameworks: Libraries

- Class-class yang disediakan oleh CodeIgniter
- Digunakan untuk melakukan pekerjaan-pekerjaan yang sering dilakukan saat membuat aplikasi web
- Diletakkan pada **`/system/libraries`**



# Controller

**/application/controllers/url.php**

```
class Url extends CI_Controller {  
    ...  
}
```

**http://localhost/index.php/url**



# Controller

<http://localhost/index.php/url/index>

```
public function index() {  
    ...  
}
```

```
public function browse($page) {  
    ...  
}
```

<http://localhost/index.php/url/browse/3>

# Controller: private function

```
<?php
```

```
class Url extends CI_Controller {
```

```
...
```

```
private function _generateurl() {
```

```
...
```

```
}
```

[http://localhost/index.php/url/\\_generateurl](http://localhost/index.php/url/_generateurl)

```
?>
```

# Model

```
class Url_model extends CI_Model {  
    ...  
}
```

**`/application/models/url_model.php`**

# Model

```
<?php
    class Url_model extends CI_Model {

        public function add_url($url1, $url2) {
            ...
        }

        public function geturl() {
            ...
        }
    }
?>
```

# Loading Model

- Menggunakan `$this->load->model(tabel)`

```
class Url extends CI_Controller {  
    public function index() {  
        $this->load->model('Urls');  
        $list = $this->Url_model->geturl();  
        ...  
    }  
}
```

`$this->load->model('Urls');`

`$list = $this->Url_model->geturl();`

# Querying Database

```
$this->load->database();  
$sql = 'SELECT * FROM urls;  
$query = $this->db->query($sql);  
  
foreach($query->result as $row) {  
    $id= $row->id;  
    $url_origin = $row->url_origin;  
    ...  
    $clicks= $row->clicks;  
}
```

# Active Record dari Database

- Mengambil seluruh url yang ada di tabel **urls**

```
$query = $this->db->query('SELECT * FROM urls');
```

- Dengan **active record**

```
$query = $this->db->get('urls');
```

Contoh limit query:

```
$query = $this->db->get('urls', 10, 20);
```

# Active Record dari Database

- Mengambil id,url\_origin dari **urls**

```
$this->db->select('id, url_origin');
```

```
$this->db->get('urls');
```

- Mengambil id url no 5

```
$id= 5;
```

```
$this->db->select('id,url_origin,url_short');
```

```
$this->db->where('id', $id);
```

```
$query = $this->db->get('urls');
```



# Active Record

- Dengan Query Bindings

```
$id = 100; $url_origin = 'http://www.ukdw.ac.id';
```

```
$url_short = 'zxy'; $clicks = 2;
```

```
$sql = 'insert into urls(id, url_origin,url_short,clicks)  
values(?, ?, ?, ?)';
```

```
$query = $this->db->query($sql, array($id,  
$url_origin, $url_short, $clicks);
```

# Pemanggilan View

- Pada Controller, dengan fungsi

```
$this->load->view(namaview)
```

Contoh:

```
$this->load->view('url_view');
```

- Pengiriman data ke View

```
$data= $this->Url_model->getURL();
```

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

**DEMO URL SHORTENER**

# Database Configuration

```
$db['default']['hostname'] = 'localhost';  
$db['default']['username'] = 'root';  
$db['default']['password'] = '';  
$db['default']['database'] = 'shorty';  
$db['default']['dbdriver'] = 'mysql';  
$db['default']['dbprefix'] = '';  
$db['default']['pconnect'] = TRUE;  
$db['default']['db_debug'] = TRUE;  
$db['default']['cache_on'] = FALSE;  
$db['default']['cachedir'] = '';  
$db['default']['char_set'] = 'utf8';  
$db['default']['dbcollat'] = 'utf8_general_ci';  
$db['default']['swap_pre'] = '';  
$db['default']['autoinit'] = TRUE;  
$db['default']['stricton'] = FALSE;
```

## Table : urls

```
CREATE TABLE `urls` (  
  `id` INT(10) NOT NULL AUTO_INCREMENT,  
  `url_origin` VARCHAR(500) NULL DEFAULT NULL,  
  `url_short` VARCHAR(500) NULL DEFAULT NULL,  
  `created` TIMESTAMP NULL DEFAULT  
CURRENT_TIMESTAMP,  
  `clicks` INT(10) NULL DEFAULT '1',  
  PRIMARY KEY (`id`) ) COLLATE='latin1_swedish_ci'  
ENGINE=InnoDB;
```

# Step 1 : Creating Views

- Displaying data from Controller
- Collect user input
- Create wireframe for designing

# Wireframes (index\_view)

**ShortMe**

Long URL

[List All](#)

ShortMe (C) 2012

# Wireframes (Result)

The wireframe shows a central area with the title "ShortMe" in bold. Below it is a text input field labeled "Long URL". Underneath the input field, the text "Short URL:" is followed by a blue, underlined link "http://localhost/shortme/abc". Below the link is another blue, underlined link "List All". At the bottom center of the wireframe, the text "ShortMe (C) 2012" is displayed.

**ShortMe**

Long URL

Short URL: <http://localhost/shortme/abc>

[List All](#)

ShortMe (C) 2012



# Wireframes (List URL)

**ShortMe**

[List All](#)

URL	Short URL	Click	Delete
http://www.ukdw.ac.id	http://localhost /shortme/abc	3	delete
http://www.ukdw.ac.id	http://localhost /shortme/abc	3	delete
http://www.ukdw.ac.id	http://localhost /shortme/abc	3	delete

ShortMe (C) 2012

# Step 2 : Creating Models

- URLModel
  - Kumpulan dari data-data URL yang sudah dipendekkan

# Step 3 : Creating Controllers

- URLController
  - Handling URL operation

# Creating View (ShortMe)

Tampilan awal

**ShortMe!**

[List All](#)

Tampilan setelah ada url yang dipendekkan

**ShortMe!**

<http://localhost/shortme/vyps>

[List All](#)

# View List URL

## ShortMe!

Paste your long URL here

ShortMe

[Hide](#)

URL	Short	Created	Click	Delete
<a href="http://mediafiremoviez.com/brrip/gangs-of-wasseypur-2012-dvdrip-720p-800mb/?utm_source=rss&amp;utm_medium=rss&amp;utm_campaign=gangs-of-wasseypur-2012-dvdrip-720p-800mb">http://mediafiremoviez.com/brrip/gangs-of-wasseypur-2012-dvdrip-720p-800mb/?utm_source=rss&amp;utm_medium=rss&amp;utm_campaign=gangs-of-wasseypur-2012-dvdrip-720p-800mb</a>	<a href="http://localhost/shortme/pg5r">http://localhost/shortme/pg5r</a>	2012-08-12 01:07:29	1	<a href="#">Delete</a>
<a href="http://ganool.com/kyou-no-go-no-ni-ova">http://ganool.com/kyou-no-go-no-ni-ova</a>	<a href="http://localhost/shortme/hpg1">http://localhost/shortme/hpg1</a>	2012-08-12 00:51:10	1	<a href="#">Delete</a>
<a href="http://ganool.com/war-of-the-dead-2011-bluray-720p-500mb-ganool">http://ganool.com/war-of-the-dead-2011-bluray-720p-500mb-ganool</a>	<a href="http://localhost/shortme/h7t0">http://localhost/shortme/h7t0</a>	2012-08-12 00:47:10	1	<a href="#">Delete</a>
<a href="http://ganool.com/maji-de-watashi-ni-koi-shinasai">http://ganool.com/maji-de-watashi-ni-koi-shinasai</a>	<a href="http://localhost/shortme/51j">http://localhost/shortme/51j</a>	2012-08-12 00:45:43	1	<a href="#">Delete</a>
<a href="http://codeigniter.com/user_guide/database/results.html">http://codeigniter.com/user_guide/database/results.html</a>	<a href="http://localhost/shortme/oj3s">http://localhost/shortme/oj3s</a>	2012-08-12 00:40:05	2	<a href="#">Delete</a>
<a href="http://lecturer.ukdw.ac.id/anton/v2/pbk.php">http://lecturer.ukdw.ac.id/anton/v2/pbk.php</a>	<a href="http://localhost/shortme/o8vz">http://localhost/shortme/o8vz</a>	2012-08-11 23:53:28	4	<a href="#">Delete</a>
<a href="http://lecturer.ukdw.ac.id/anton/v2/progweb.php">http://lecturer.ukdw.ac.id/anton/v2/progweb.php</a>	<a href="http://localhost/shortme/77cb">http://localhost/shortme/77cb</a>	2012-08-11 23:39:07	2	<a href="#">Delete</a>

# Routes and Config

Jika url yang diinputkan oleh user tidak ada nama controllernya maka lempar ke url/decode dan kemudian redirect ke halaman aslinya dan update clicks +1

```
$route['default_controller'] = "url";  
$route['url/add'] = "url/add";  
$route['url/del'] = "url/del";  
$route['404_override'] = 'url/decode';
```

```
*/  
$config['base_url'] = 'http://localhost/shortme/';  
/*  
|-----  
-----
```

Misalnya: <http://localhost/shortme/xyz>

# Kode Controller

```
class URL extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->model('url_model');
    }

    public function index($x='')
    {
        $data['title'] = "ShortMe! - A URL Shortener";
        $this->load->view('index_view', $data);
    }
}
```

```
public function getUrl () {
    $data['title'] = 'ShortMe! - View All URLs';
    $data['all'] = $this->url_model->getUrl();
    $this->load->view('index_view', $data);
}

public function decode(){
    $shorty=$this->uri->segment(1);
    redirect($this->url_model->getUrlOri($shorty));
}

public function del($id){
    if((int)$id > 0){
        $this->url_model->delete_url($id);
        redirect("url/getURL");
    }
}
```

# Controller add url baru

```
public function add() {
    $this->load->helper('form');
    $this->load->library('form_validation');
    $this->form_validation->set_rules('url_origin', 'URL cannot
be empty', 'required');
    |
    if ($this->form_validation->run() === FALSE)
    {
        $this->load->view('index_view');
    }
    else
    {
        $urlinput=mysql_real_escape_string($_POST['url_origin']
]);
        $id=rand(10000,99999);
        $shorturl=base_convert($id,20,36);
        $this->url_model->add_url($urlinput,$shorturl);
        $data['title'] = "ShortMe! - Add URL";
        $data['urlhasil'] = $shorturl;
        $this->load->view('index_view',$data);
        //redirect("url/getURL");
    }
}
```



# Model

```
class Url_model extends CI_Model {

    public function __construct()
    {
        $this->load->database();
    }

    public function add_url($url1, $url2)
    {
        $sql = "insert into urls(url_origin,url_short) values
(? , ?)";
        $query = $this->db->query($sql, array($url1, $url2));
    }

    public function getURL(){
        $hasil = $this->db->query("select * from urls order by
created desc");
        return $hasil;
    }
}
```

# Model

```
public function getURLOri($url) {
    $query = $this->db->query("select * from urls where
url_short='$url'");
    if($query->num_rows()==1)
    {
        $row = $query->row();
        $query2 = $this->db->query("update urls set
clicks=clicks+1 where id=".$row->id);
        return $row->url_origin;
    }
    else return "#";
}

public function delete_url($id)
{
    $sql = "delete from urls where id=?";
    $query = $this->db->query($sql,array($id));
}
}
```

**DEMO**

**LAST LECTURE:  
ARSITEKTUR INFORMASI**