

Pengaturcaraan Web dengan PHP

Advance PHP

Kandungan Bengkel

1. Hari pertama
 - a. Session Handling
 - b. Submit form to DB
2. Hari kedua
 - a. Output from DB and reporting
 - b. PHP Framework with CodeIgniter

Hari 2 Sessi Pagi

Data dari DB ke PHP

tarik data dari DB ke php

Kita perlukan

- sambungan kepada DB
- fail php yang menghantar request menggunakan SQL
- menerima data dari DB dan diproses untuk dipaparkan
- struktur html untuk memaparkan data

SQL dan mysql_query

- gunakan pembolehubah untuk menyimpan SQL statement, biasanya \$query.
- simpan hasil daripada mysql_query(\$query) dalam satu pembolehubah, biasanya \$result.
- uji dahulu samada \$result tidak error
- uji dahulu samada \$result tidak kosong dengan mysql_num_rows();

Manipulasi \$result

- manipulasi \$result dengan menggunakan
 - `mysql_fetch_array($result)`
 - `mysql_fetch_assoc($result)`
 - `mysql_result($result, 0, 1)`

mysql_* dan mysqli_*

Fungsi mysql_* seperti mysql_query, mysql_fetch_array dan yang sama dengannya akan menjadi usang (deprecated) bagi PHP versi 5.5 ke atas.

Fungsi tersebut digantikan dengan mysqli_* namun masih lagi memegang fungsi yang sama.

Perlukan Data

- Cari data dari table
- Papar data berdasarkan input
- Cari data untuk update
- Jana laporan
- Statistik

Hari 2 Sessi Petang

Php Framework

CodeIgniter

Framework?

- Kejututeraan: Struktur yang menyokong bangunan
- Dalam perisian; Struktur hierarki yang merangkumkan sumber-sumber yang berkongsi kegunaan dalam satu pakej.
- Contoh-contoh sumber adalah;
 - library
 - header
 - ini, conf, function...
 - references
- Platform untuk membina aplikasi

Framework

- Framework mengandung fungsi dan class yang tersedia untuk melakukan fungsi-fungsi asas seperti sambungan ke DB, proses input, proses output, susunan fail library..
- Framework membantu aturcara menjadi lebih tersusun, teratur dan 'agile'

PHP Framework : CodeIgniter (CI)

“CodeIgniter is a powerful PHP framework with a very small footprint, built for developers who need a simple and elegant toolkit to create full-featured web applications”

<http://www.codeigniter.com/>

CodeIgniter Download

- Open Source, percuma, bebas, boleh ubah.
- update terbaru pada : Januari 2015 v2.2.1
- CI versi3.x akan datang!
- Dev : <https://github.com/bcit-ci/CodeIgniter/>
- 317 contributors

- Download : (2.2MB)

<https://github.com/bcit-ci/CodeIgniter/archive/2.2-stable.zip>

Install CI

1. Unzip
2. pindah ke folder webroot
cth ; C:\wamp\www\CodeIgniter2.12
3. tukar nama folder CodeIgniter.. kepada nama web anda
4. set base URL dalam `application/config/config.php`
5. tukar DB setting dalam `application/config/database.php`

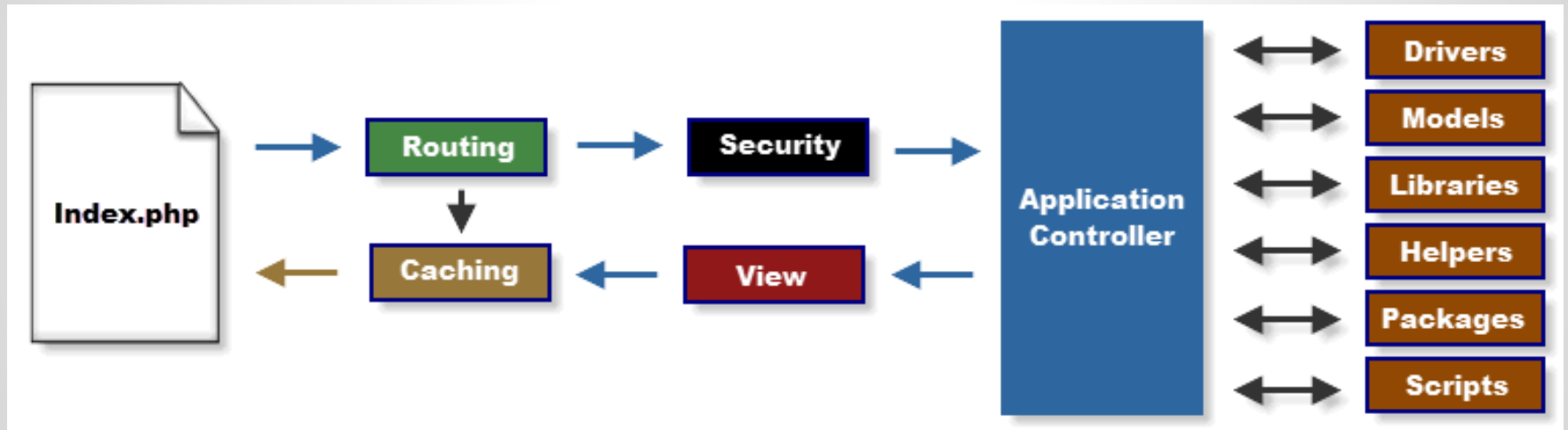
Pengenalan CI

- Application framework
- Free
- Ringan
- Laju
- Guna **M-V-C**
- URL yang cantik: www.nama.com/blog/entry/23
- Semua ada - akses db, hantar emel, data validation, session, image, dll
- Boleh dikembangkan

Features

- Model-View-Controller Based System
- Extremely Light Weight
- Full Featured database classes with support for several platforms.
- Active Record Database Support
- Form and Data Validation
- Security and XSS Filtering
- Session Management
- Email Sending Class. Supports Attachments, HTML/Text email, multiple protocols (sendmail, SMTP, and Mail) and more.
- Image Manipulation Library (cropping, resizing, rotating, etc.). Supports GD, ImageMagick, and NetPBM
- File Uploading Class
- FTP Class
- Localization
- Pagination
- Data Encryption
- Benchmarking
- Full Page Caching
- Error Logging
- Application Profiling
- Calendaring Class
- User Agent Class
- Zip Encoding Class
- Template Engine Class
- Trackback Class
- XML-RPC Library
- Unit Testing Class
- Search-engine Friendly URLs
- Flexible URI Routing
- Support for Hooks and Class Extensions
- Large library of "helper" functions

Flow Chart Aplikasi CI



Model-View-Controller

- **Model** mewakili struktur data. Biasanya Model menjalankan fungsi ambil, tulis, update dan buang data daripada DB
- **View** adalah maklumat yang dipersembahkan kepada pengguna. Biasanya fail web.
- **Controller** berfungsi sebagai orang-tengah antara model dan view, juga sumber-sumber lain yang diperlukan untuk menjana satu laman web.

Tutorial : Pengenalan kepada CI

- Asas MVC
- Asas Routing
- Form Validation
- Kueri DB dengan “Active Record”

Laman Statik - Asas Routing

1. Setkan controller dahulu

- a. controller adalah pemudahcara aplikasi anda. Ia menggambarkan apa yang diminta dan model yang berkenaan

contohnya;

<http://example.com/news/latest/10>

- controller : news
- method : latest
- params : 10

biasanya dalam MVC, paten URL adalah seperti berikut

`http://example.com/[controller-class]/[controller-method]/[arguments]`

Asas Controller

namakan Controller untuk
menguruskan laman statik : pages

Cipta fail

`application/controllers/pages.php`

Code di sebelah menunjukkan
function view dalam class Pages
boleh menerima argument `$page`
(default : 'home')

```
<?php
```

```
class Pages extends CI_Controller {  
    public function view($page = 'home')  
    {  
    }  
}  
  
?>
```

Header & Footer

application/views/templates/header.php

```
<html>
<head>
    <title><?php echo $title ?> -
    CodeIgniter 2 Tutorial</title>
</head>
<body>
    <h1>CodeIgniter 2 Tutorial</h1>
```

application/views/templates/footer.php

```
<strong>&copy; 2011</strong>
</body>
</html>
```

home.php & about.php

```
application/views/pages/home.php
```

```
<h1>Assalamualaikum!</h1>  
<p>Ini adalah home.php</p>
```

```
application/views/pages/about.php
```

```
<h1>Salam Dunia!</h1>  
<p>Ini adalah about.php</p>
```

Tambah Logik kepada Controller

```
public function view($page = 'home')
{
    if ( ! file_exists(APPPATH.'/views/pages/'.$page.'.php'))
    {
        // fail tak wujud!
        show_404();
    }

    $data['title'] = ucfirst($page); // huruf besar: huruf pertama
    $this->load->view('templates/header', $data);
    $this->load->view('pages/'.$page, $data);
    $this->load->view('templates/footer', $data);
}
```


Cuba!

Guna browser anda, tujukan ke

1. <localhost/index.php/pages/view>
2. <localhost/index.php/pages/view/about>

Routing

- routing adalah untuk petakan akses dalam laman web.
- edit fail [application/config/routes.php](#) dan tambah kod di bawah. Buang semua kod lain dalam array \$route

```
$route['default_controller'] = 'pages/view';  
$route['(:any)'] = 'pages/view/$1';
```

- akses ke laman web anda dengan [index.php/about](#).
Adakah anda tiba di laman yang betul?
- anda patut berada di [index.php/pages/view/about](#)

Tutorial CI : Model

- tempat semua operasi dengan database
- ambil, tulis dan kemaskini data dalam database.
- untuk tutorial, bina database table nama news

```
CREATE TABLE news (  
    id int(11) NOT NULL  
    AUTO_INCREMENT,  
    title varchar(128) NOT NULL,  
    slug varchar(128) NOT NULL,  
    text text NOT NULL,  
    PRIMARY KEY (id),  
    KEY slug (slug)  
);
```

Cipta Model

- cipta satu fail dalam folder [application/models](#)
- nama fail news_model.php
- setting database berada di [application/config/database.php](#)

```
application/models/news_model.php

<?php

class News_model extends CI_Model {

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

Senaraikan semua post

- kueri untuk senaraikan semua post dari table news dalam db
- kueri menggunakan ActiveRecord
- tambahan kod model news

```
public function get_news($slug = FALSE)
{
    if ($slug === FALSE)
    {
        $query = $this->db->get('news');
        return $query->result_array();
    }

    $query = $this->db->get_where('news',
        array('slug' => $slug));
    return $query->row_array();
}
```

Controller untuk news

- cipta controller untuk uruskan data kueri dari model news
- `application/controllers/news.php`
- controller akan menerima data dari model, hantarkan kepada views menerusi pembolehubahat `$data['news']`
- controller akan load model dalam kelasnya sendiri.

```
<?php
```

```
class News extends CI_Controller {
```

```
    public function __construct()
```

```
    {
```

```
        parent::__construct();
```

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

```
    }
```

```
    public function index()
```

```
    {
```

```
        $data['news'] = $this->news_model->get_news();
```

```
    }
```

```
    public function view($slug)
```

```
    {
```

```
        $data['news'] = $this->news_model->get_news($slug);
```

```
    }
```

```
}
```

Amalan penamaan laman web

- index : senarai kesemua item
- view/\$id : papar 1 item dengan params id
- new : borang untuk cipta 1 item
- create : cipta 1 row dalam table
- delete/\$id : padam 1 item

Cipta method dalam controller

- tambah fungsi index dalam controller
- umpukkan dalam array \$data
- header perlukan \$data ['title']

```
public function index()
{
    $data['news'] = $this->news_model->get_news();
    $data['title'] = 'News archive';

    $this->load->view('templates/header', $data);
    $this->load->view('news/index', $data);
    $this->load->view('templates/footer');
}
```

Views untuk news index

Cipta fail di

`application/views/news/index.php`

array `$data['news']`

akan menjadi

`$news` dalam view

```
<?php foreach ($news as $news_item): ?>
<h2><?php echo $news_item['title'] ?></h2>
<div class="main">
    <?php echo $news_item['text'] ?>
</div>
<p>
    <a href="news/<?php echo $news_item['slug'] ?>">
        View article
    </a>
</p>
<?php endforeach ?>
```

View untuk satu item (tunggal)

```
public function view($slug)
{
    $data['news_item'] = $this->news_model->get_news($slug);

    if (empty($data['news_item']))
    {
        show_404();
    }
    $data['title'] = $data['news_item']['title'];

    $this->load->view('templates/header', $data);
    $this->load->view('news/view', $data);
    $this->load->view('templates/footer');
}
```

view satu item

```
<?php
//application/views/news/view.php
echo '<h2>'.$news_item['title'].'</h2>';
echo $news_item['text'];
```

Form

```
application/views/news/create.php
```

```
<h2>Create a news item</h2>
```

```
<?php echo validation_errors(); ?>
```

```
<?php echo form_open('news/create') ?>
```

```
    <label for="title">Title</label>
```

```
    <input type="input" name="title" /><br />
```

```
    <label for="text">Text</label>
```

```
    <textarea name="text"></textarea><br />
```

```
    <input type="submit" name="submit" value="Create news item" />
```

```
</form>
```

news controller

```
public function create()
{
    $this->load->helper('form');
    $this->load->library('form_validation');
    $data['title'] = 'Create a news item';
    $this->form_validation->set_rules('title', 'Title', 'required');
    $this->form_validation->set_rules('text', 'text', 'required');
    if ($this->form_validation->run() === FALSE){
        $this->load->view('templates/header', $data);
        $this->load->view('news/create');
        $this->load->view('templates/footer');
    }
}
```

```
else{
    $this->news_model->set_news();
    $this->load->view('news/success');
}
}
```

model news

```
public function set_news(){
    $this->load->helper('url');
    $slug = url_title($this->input->post('title'), 'dash', TRUE);

    $data = array(
        'title' => $this->input->post('title'),
        'slug' => $slug,
        'text' => $this->input->post('text')
    );

    return $this->db->insert('news', $data);
}
```

Bengkel PHP Jan 2015

terima kasih