

Pertemuan 8

M(odel) V(iew) C(ontroller)

MVC



MVC: CodeIgniter menggunakan pola pengembangan Model-View-Controller. MVC adalah pendekatan perangkat lunak yang memisahkan logika aplikasi dari presentasi (tampilan).

Peran Setiap MVC:

1. Model: Merepresentasikan struktur data. Class model berisi fungsi yang digunakan untuk select, insert, update informasi di database.
2. View: Informasi yang ditampilkan ke user (Frontend). View biasanya berupa halaman web.
3. Controller: Perantara antara Model dan View, perantara HTTP request, dan men-generate halaman web.

Model

Class PHP yang dirancang untuk bekerja dengan informasi di database.

Contoh, kita memiliki Class Model yang berisi fungsi untuk insert, update, dan select data blog di database.



Model - Rules



Rules:

1. Anatomy Model:

- Class Model disimpan di direktori `application/models`
- Nama file harus memiliki huruf pertama kapital dan sisanya lowercase: `Blog_model.php`
- Nama Class harus sama seperti nama file: `class Blog_Model extends CI_Model{}`

2. Loading Model:

- Model biasanya dipanggil di method Controller. Untuk memanggil model:
`$this->load->model('blog_model')`
- Setelah dipanggil, kita bisa mengakses method menggunakan object yang sama dengan nama class:
`$this->blog_model->method()`

Model - Membuat Model Pasien



```
/*  
Membuat model Pasien.  
Nama file: application/models/Pasien_model.php  
Nama Class seperti nama file: Pasien_model  
*/  
class Pasien_model extends CI_Model {  
    /*  
    Membuat property.  
    Property diberikan modifies public.  
    */  
    public $id, $nama, $kode, $gender;  
    public $tmp_lahir, $tgl_lahir, $email;  
}
```

Model - Membuat Model BMI



```
// Membuat Model BMI
// File: application/models/Bmi_model.php
class Bmi_model extends CI_Model {
    // Membuat property
    public $berat;
    public $tinggi;

    // Membuat Method menghitung nilai BMI
    public function nilaiBmi() {
        $bmi = $this->berat / pow($this->tinggi, 2);
        return $bmi;
    }
}
```

Model - Membuat Model BMI Pasien



```
// Membuat Model BMI Pasien  
// File: application/models/Bmipasien_model.php  
class Bmipasien_model extends CI_Model {  
    // Membuat Property  
    public $id;  
    public $tanggal;  
    public $pasien;  
    public $bmi;  
}
```

Controller

Sebuah file class yang namanya diasosiasikan dengan URL.

URL `localhost/index.php/blog`, akan mengakses controller `Blog.php`.

Controller adalah jantung dari aplikasi, karena menentukan bagaimana HTTP Request harus ditangani.



Controller - Rules



Rules:

1. Controller:
 - a. Class Controller harus disimpan di direktori `application/controllers`
 - b. Nama file harus huruf kapital: `application/controllers/Blog.php`
 - c. Nama Class harus sama seperti nama file: `Class Blog extends CI_Controller {}`
 - d. Kita dapat mengakses URL berdasarkan Controller Blog yang sudah dibuat: `localhost/index.php/blog`
2. Method
 - a. Segmen kedua dari URI menentukan method mana yang akan dipanggil di Controller
 - b. URI `localhost/index.php/blog/index` akan memanggil method index (default)
 - c. Membuat method comment: `public function comment() {}`
 - d. Kita dapat mengakses Method Comment di Controller Blog: `localhost/index.php/blog/comment`

Controller - Membuat Controller Pasien



Next for the Image >

Controller Pasien

```
// Membuat Controller Pasien
class Pasien extends CI_Controller {
    // Membuat Method index
    public function index() {
        // Memanggil model pasien_model, dan memberikan nama object baru: pasien1
        $this->load->model('pasien_model', 'pasien1');
        $this->pasien1->id = 1;
        $this->pasien1->kode = '011021';
        $this->pasien1->nama = 'Aufa';
        $this->pasien1->gender = 'L';

        // Memanggil model pasien_model, dan memberikan nama object baru: pasien2
        $this->load->model('pasien_model', 'pasien2');
        $this->pasien2->id = 2;
        $this->pasien2->kode = '011022';
        $this->pasien2->nama = 'Billah';
        $this->pasien2->gender = 'P';

        // Menyimpan object pasien1 dan pasien2 ke dalam array
        $list_pasien = [$this->pasien1, $this->pasien2];
        $data['list_pasien'] = $list_pasien;

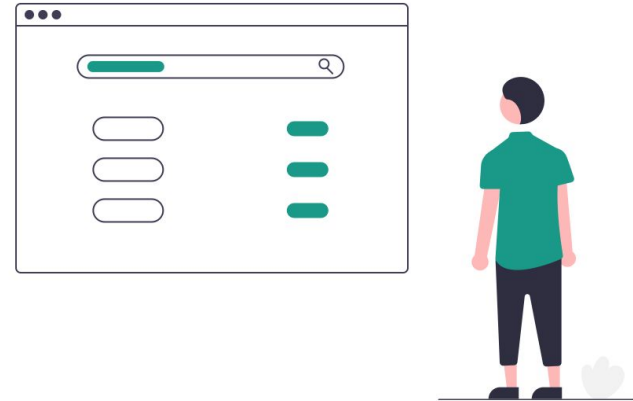
        // Merender view sambil mengirimkan data
        $this->load->view('pasien/index', $data)
    }
}
```

View

View hanyalah tampilan halaman website
(Front-End)

View tidak pernah dipanggil secara langsung,
View harus dipanggil di Controller

Controller bertanggung jawab untuk mengambil
view dan merendernya ke User.



Views - Rules



Rules:

1. Creating View:

- a. File view disimpan di direktori `application/views/`
- b. Membuat file view: `application/views/blogview.php`

2. Loading View

- a. Memanggil view menggunakan method view: `$this->load->view('blogview.php');`

3. Adding Data to View

- a. Membuat data dalam bentuk array atau object: `$data['title'] = 'Halaman Blog';`
- b. Mengirim data di parameter kedua dari method view: `$this->load->view('blogview.php', $data);`
- c. Menampilkan data yang dikirim di view dengan mengakses key dari array: `<h1><?= $title ?></h1>`

View - Membuat View Pasien



```
<div>
  <h3>Daftar Pasien</h3>
  <!--
    menampilkan data patients yang dikirim oleh controller di view.
    data dilooping menggunakan foreach.
    setiap data yang dilooping merupakan sebuah object.
    karena object, untuk menampilkan nilainya harus mengakses property.
  -->
  <ul>
    <?php foreach($patients as $patien): ?>
      <li><?= $patien->kode ?></li>
      <li><?= $patien->nama ?></li>
      <li><?= $patien->gender ?></li>
    <?php endforeach; ?>
  </ul>
</div>
```

Referensi

- Documentation CodeIgniter 3: [Model](#)
- Documentation CodeIgniter 3: [View](#)
- Documentation CodeIgniter 3: [Controller](#)

