



## **PERTEMUAN 3**

### **DEMONSTRASI TUGAS 2**

**Hasil Pengerjaan contoh1,contoh2,contoh3, dan  
menghilangkan index.php**

## **PERTEMUAN 3**

### **MODEL, VIEW, CONTROLLER**

Pada bab ini diharapkan mahasiswa dapat memahami secara detail tentang pembuatan dan penggunaan komponen model, view, controller yang merupakan kunci utama dalam menggunakan framework codeigniter.

#### **1. Pengertian MVC**

Pada pengertian codeigniter di atas tadi di jelaskan bahwa codeigniter menggunakan metode MVC. Apa itu MVC? kita juga harus mengetahui apa itu MVC sebelum masuk dan lebih jauh dalam belajar codeigniter.

MVC adalah teknik atau konsep yang memisahkan komponen utama menjadi tiga komponen yaitu model, view dan controller.



## MODEL, VIEW, CONTROLLER Lanjutan...

### a. Model

Model adalah kelas yang merepresentasikan atau memodelkan tipe data yang akan digunakan oleh aplikasi.

Sebagai catatan, Semua model harus disimpan di dalam folder **application\models**

### b. View

View merupakan bagian yang menangani halaman user interface atau halaman yang muncul pada user(pada browser). tampilan dari user interface di kumpulkan pada folde

**Application/views**

### c. Controller

Controller merupakan kumpulan intruksi aksi yang menghubungkan model dan view

Untuk bentuk umum dan contoh-contoh penggunaan model, view, controller serta hasilnya bisa dilihat pada modul halaman 9.



## MODEL, VIEW, CONTROLLER Lanjutan...

### 2. Menentukan controller standar (Default)

Controller standar (default) adalah controller yang akan dipanggil secara otomatis ketika user tidak menyertakan nama controller di dalam URI.

Contoh penulisan URI:

`http://localhost/rental_mobil/index.php`

pada contoh permintaan di atas, kita tidak menyertakan nama controller yang akan dipanggil (segmen pertama). Permintaan tersebut akan memanggil controller standar yang sudah didefinisikan oleh codeigniter hasilnya bisa dilihat di modul halaman 13



## MODEL, VIEW, CONTROLLER Lanjutan...

Controller standar dapat ditentukan sendiri sesuai dengan keinginan, yaitu dengan cara melakukan konfigurasi pada file `application\config\routes.php`. cari baris code berikut:

```
$route['default_controller'] = 'welcome';
```

Ubah kata `welcome` dengan nama controller yang diinginkan untuk dijadikan controller standar



## MODEL, VIEW, CONTROLLER Lanjutan...

### 3. Aturan merancang Model, View, dan Controller

#### a. Model

Secara umum perancangan model dapat dibuat menggunakan aturan berikut:

- 1) Model harus memiliki atribut yang dapat mewakili element tertentu.
- 2) Model seharusnya berisi kode kode yang menangani proses bisnis untuk data bersangkutan, misalnya untuk melakukan proses validasi data.
- 3) Model boleh berisi kode kode yang digunakan untuk memanipulasi data, misalnya untuk menambah baris data baru, mengubah baris data, dan menghapus baris data pada sebuah tabel yang terdapat dalam database.

Model seharusnya tidak berisi kode kode yang berkaitan langsung dengan permintaan yang dilakukan oleh user melalui URL (tidak berisi `$_GET` maupun `$_POST`). Pekerjaan seperti ini seharusnya dilakukan oleh controller.

## MODEL, VIEW, CONTROLLER Lanjutan...

Model seharusnya tidak berisi kode kode yang berkaitan dengan presentasi (tampilan) halaman web yang akan disajikan ke hadapan user. Pekerjaan semacam ini seharusnya dilakukan oleh view.

### b. View

Berikut ini aturan yang dapat digunakan untuk merancang view.

- 1) View harus berisi kode kode yang bersifat presentasional, biasanya berupa kode HTML. View juga dapat berisi perintah perintah PHP sederhana yang masih berkaitan dengan proses pembuatan tampilna, misalnya untuk menampilkan data yang dihasilkan oleh model tertentu.
- 2) View seharusnya tidak berisi kode PHP yang melakukan akses data secara langsung ke database. Pekerjaan semacam ini sebaiknya dilakukan oleh model.
- 3) View seharunsnya menghindari kode PHP yang mengakses secara langsung permintaan dari user (tidak berisi `$_GET` maupun `$_POST`). Tugas ini sebaiknya didelegasikan ke controller

## MODEL, VIEW, CONTROLLER Lanjutan...

### c. Controller

Berikut ini aturan yang dapat digunakan untuk merancang controller.

- 1) Controller boleh mengakses `$_GET`, `$_POST`, dan variabel variabel PHP lainnya yang berkaitan dengan permintaan user.
- 2) Controller boleh membuat objek (instance) dari kelas kelas model dan mengarahkan ke view (jika model bersangkutan menghasilkan data yang perlu ditampilkan ke user)
- 3) Controller seharusnya tidak berisi kode kode yang mengakses data secara langsung dari database. Tugas seperti ini sebaiknya dilakukan oleh model. Dan controller tinggal menggunakannya saja.
- 4) Controller seharusnya tidak berisi kode kode HTML yang diperlukan untuk tujuan presentasi. Tugas ini sebaiknya dilakukan oleh view.



## MODEL, VIEW, CONTROLLER Lanjutan...

### 4. Parsing Data Ke View

Untuk memarsing data dari controller ke view, anda dapat memarsingnya dengan menggunakan bantuan array. Jadi data yang akan diparsing kita masukkan ke array.

Perhatikan contoh cara memarsing data ke view codeigniter berikut ini. Buka controller belajar.

**Contoh5** parsing data dari controller ke view

Lihat di modul halaman 16 s/d 20

### Tugas 3

- Mahasiswa mengerjakan dan mengimplementasikan materi yang ada di pertemuan 4 yaitu membuat template sederhana dan dinamis dan setting `base_url()`
- Upload (push) file-file project latihan anda ke repository Github
- Demonstrasikan dan kumpulkan hasilnya di pertemuan selanjutnya