

# PHP Framework

## Framework

- ▶ Suatu aplikasi parsial yang tersusun dari banyak komponen berbentuk kelas yg dikhususkan untuk pembangunan aplikasi tertentu atau aplikasi domain spesifik

## Karakteristik Framework

- ▶ Mendukung mekanisme extension (penambahan / perbaikan fitur)
- ▶ Dibuat sesimpel mungkin, tapi tidak lebih simpel
- ▶ Memiliki dokumentasi how-to
- ▶ Dapat diaplikasikan secara luas dan reusable (dipakai berulang)
- ▶ Berstruktur “skeletal” atau “assembly”

## Framework Trade-off

### ▶ Kelebihan:

- Reusability: mempercepat waktu pengembangan
- Extensibility: fleksibel (kompleksitas pengembangan berkurang), one for all (bila bekerja secara tim, cukup satu dibuat, dapat dipakai seluruh tim)

### ▶ Kekurangan:

- Learning curve (developer harus mempelajari framework)
- Kebanyakan open source (bila ada bug dalam framework tidak langsung dapat diperbaiki oleh pengembang framework)
- Flow control dasar sistem tidak diketahui oleh pengguna framework

## Model-View-Controller (MVC)

- ▶ MVC adalah pola arsitektur pada perangkat lunak yang memisahkan interaksi antara user dengan aplikasi ke dalam tiga role:
  - Model (business logic)
  - View (user interface)
  - Controller (process logic / user input)
- ▶ Pemisahan “urusan” ini memungkinkan pengembangan, pengujian, dan pemeliharaan perangkat lunak pada tiap role menjadi lebih fleksibel dan independen

## Model

- ▶ Mengelola “behavior” dan data dari lingkup aplikasi
- ▶ Merespon request / informasi state dari role View
- ▶ Merespon request untuk melakukan suatu instruksi dari role Controller
- ▶ Contoh: kelas representasi basis data, active record, dll

## View

- ▶ Membuat Model menjadi bentuk yg sesuai untuk interaksi atau user interface
- ▶ Satu Model / Controller dapat mempunyai beberapa View untuk kebutuhan yg berbeda
- ▶ Contoh: halaman login, halaman lihat data, dll

## Controller

- ▶ Menerima input user
- ▶ Merespon dengan memanggil objek Model / View
- ▶ Memberi instruksi pada Model / View untuk melakukan aksi berdasarkan input tadi
- ▶ Contoh: kelas pemrosesan untuk CRUD, kelas koneksi, dll



## Alur siklus MVC

- ▶ User berinteraksi dengan View (UI) dengan cara tertentu (misal: klik tombol)
- ▶ Controller menangani input dari View, menerjemahkannya menjadi instruksi aksi yang dikenali oleh Model
- ▶ Controller memanggil Model dan memberikan instruksi aksi tadi ke Model
- ▶ Model menjalankan instruksi
- ▶ Model memberi notifikasi kepada View (biasanya melalui Controller dulu) dan tampilan harus diperbarui
- ▶ View mengubah tampilan berdasarkan notifikasi tadi

## CodeIgniter

- Framework untuk bahasa scripting PHP
- Mudah diimplementasi
- Size framework tidak besar
- Library dan API cukup lengkap
- Dokumentasi mudah dimengerti

## CI instalasi

- ▶ Ekstrak framework CI yang sudah didownload ke folder instalasi web server (....xampp/htdocs)
- ▶ Buka framework CI di browser, jika instalasi benar maka akan muncul halaman dengan heading “Welcome to CodeIgniter”
- ▶ File konfigurasi di dalam framework CI berada di dalam folder config (...xampp/htdocs/CodeIgniterXXX/application/config). Ada banyak jenis konfigurasi disini, tapi cukup hanya empat file konfigurasi yang biasanya sering digunakan yang akan dibahas

## CI konfigurasi – config.php

- Buka file konfigurasi di dalam framework CI (.../xampp/htdocs/CodeIgniterXXX/application/config/config.php)
- Banyak pengaturan disini, yang paling penting adalah 'base\_url'. Pengaturan lain silakan dieksplorasi mandiri
- Isi 'base\_url' dengan alamat URL framework CI. Pengaturan ini berguna agar path URL aplikasi menjadi absolut, yang akan digunakan dalam penulisan link, submit form, dan load file css & js

```
$config['base_url'] = 'http://localhost/CodeIgniterXXX/';
```

## CI konfigurasi – autoload.php

- ▶ Buka file autoload.php di dalam framework CI (.../xampp/htdocs/CodeIgniterXXX/application/config/autoload.php)
- ▶ File ini berguna untuk memudahkan kita menggunakan suatu config, kelas model, library, atau helper (kumpulan fungsi) secara otomatis, sehingga kita tidak perlu secara berulang menulis kode untuk me-load config/model/library/helper di tiap file yang menggunakan config/model/library/helper tersebut (masih ingat ketika di PHP non-framework yang telah dibahas sebelumnya, jika kita ingin melakukan query ke DB, kita harus selalu menuliskan kode "include koneksi.php", agar kita bisa menggunakan library database)
- ▶ Helper yang akan sering digunakan adalah url, sedangkan library yang akan sering digunakan adalah database dan session (untuk autentikasi dan notifikasi pesan)

```
$autoload['libraries'] = array('database', 'session');  
$autoload['helper'] = array('url');
```

## CI konfigurasi – routes.php

- File routes.php di dalam framework CI (.../xampp/htdocs/CodeIgniterXXX/application/config/routes.php) berguna untuk hal routing (redirecting/forwarding URL)
- 'default controller': menspesifikkan sebuah controller yang akan diakses jika di dalam URL tidak terdapat controller mana yang diakses
- '404\_override': meng-override halaman 404 default milik CI (halaman 404 ini muncul jika URL yang kita akses menuju ke controller atau function yang tidak ada), dan mengarahkan ke suatu controller yang halaman 404-nya kita buat sendiri
- Untuk saat ini tidak perlu mengubah apapun dalam file ini

## CI konfigurasi – database.php

- ▶ File database.php di dalam framework CI (.../xampp/htdocs/CodeIgniterXXX/application/config/database.php) berguna untuk koneksi database
- ▶ File ini berlaku layaknya file 'koneksi.php' di PHP non-framework
- ▶ Isi config sesuai dengan isi 'koneksi.php' (bagian 'hostname', 'username', 'password', dan 'database')

## Dasar URL di CI

- ▶ Ketika menggunakan CI, kita tidak bisa lagi mengakses suatu URL dengan suatu nama halaman berekstensi .php, tetapi kita harus mengaksesnya melalui controller. Format CI yaitu URL bersegment:

`http://localhost/CodeIgniterXXX/index.php/[namacontroller]/[namafunction]/[parameter1]/[parameter2]/...`

- ▶ Jika [namacontroller] tidak dituliskan, maka URL akan mengakses controller yang ditulis dalam config routes.php
- ▶ [namafunction] adalah suatu function di dalam controller [namacontroller]. Jika [namafunction] tidak dituliskan, maka URL akan mengakses function dengan nama 'index' di dalam controller [namacontroller]
- ▶ [Parameter...] bersifat opsional, tergantung [namafunction] yang didefinisikan memiliki parameter atau tidak



## Dasar Controller & View di CI

- ▶ Controller terdapat di dalam folder ../application/controllers
- ▶ Kelas controller harus meng-extends dari kelas bawaan CI\_Controller, dan tutup tag php (?>) tidak perlu ditulis di akhir file
- ▶ Halaman (UI) di-load dari dalam function di controller dengan menggunakan:

```
$this->load->view(' [namahalaman] ', $data);
```

- ▶ [namahalaman] mengacu ke suatu halaman web yang ada di dalam folder ../application/views. [namahalaman] ditulis tanpa ekstensinya (jika nama halaman 'test.php', cukup tulis 'test')
- ▶ \$data bersifat opsional. \$data disertakan jika ingin mem-passing variabel dari controller agar bisa digunakan di dalam halaman view

## Dasar Controller & View di CI

- ▶ \$data harus berbentuk array (agar variabel yang disimpan dan dilempar bisa banyak)
- ▶ Cara mengakses variabel di dalam halaman view adalah dengan mengakses variabel mapping asosiatifnya:

```
//controller
```

```
$data['rebus'] = 'mie';
```

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

```
//view test.php
```

```
<?php echo $rebus; ?> <!-- output: mie -->
```

## Dasar Model di CI

- ▶ Model terdapat di dalam folder `.../application/models`
- ▶ Kelas model harus meng-extends dari kelas bawaan `CI_Model`, dan tutup tag php (`?>`) tidak perlu ditulis di akhir file
- ▶ Semua yang berhubungan dengan query DB disarankan dilakukan disini
- ▶ Model dapat di-load secara otomatis dari config `autoload.php` atau manual dari dalam function di controller dengan menggunakan:

```
$this->load->model(' [namamodel] ');
```

- ▶ `[namamodel]` mengacu ke suatu kelas model yang ada di dalam folder `.../application/models`. `[namamodel]` ditulis tanpa ekstensinya (jika nama model `'test.php'`, cukup tulis `'test'`)
- ▶ Setelah di-load, model dan function yang dimilikinya dapat diakses dengan:

```
$this->test->namafunction();
```

## Latihan

- ▶ Buatlah aplikasi PHP non-framework yang sudah dibuat sebelumnya menjadi berbasis framework CI!