

Pengertian MVC Framework PHP

Sering kita mendengar tentang **MVC Framework PHP** tetapi kurang mengerti belum arti dari MVC itu sendiri. Salah satu mvc framework yang populer digunakan saat ini adalah Code Igniter Framework yang dikembangkan oleh Eliss Lab. Framework Codeigniter atau sering disebut dengan singkatan CI ini adalah PHP framework yang terkenal di dunia dan paling banyak digunakan para developer untuk membuat aplikasi website php. Hal itu dibuktikan dengan aktifnya forum di website Code Igniter dan banyaknya masyarakat dunia yang membahasnya di forum-forum nasional mereka. Dan tentu saja standar framework yang bagus itu dilihat dari banyak audience yang memakai framework tersebut dan mempunyai dokumentasi lengkap tentang framework itu sendiri.

Sejarah Singkat MVC PHP Framework

Setelah dahulu saya menjelaskan tentang sejarah PHP, PHP sendiri telah berkembang dengan kuat sebagai salah satu jenis pemrograman website. OOP (Object Oriented Programming) atau Pemrograman Berbasis Objek telah berkembang pesat selaras dengan perkembangan PHP di versi 5. Sekarang ini, banyak aplikasi perusahaan swasta maupun perusahaan pemerintah lebih memilih menggunakan aplikasi website PHP dikarenakan memang aplikasi website sekarang ini sangat kompleks.

Hal itu juga didukung dengan script pemrograman lainnya seperti ajax, jquery, css, html yang digabungkan menjadi satu aplikasi website dengan PHP sebagai script utama websitenya. Mulai

dari aplikasi yang menggunakan API berbentuk SOAP, API WDSL, maupun aplikasi website yang membutuhkan formulir yang kompleks.

Dari situlah perusahaan besar yang fokus di PHP development, yaitu ElissLab mulai mengembangkan framework PHP dengan nama Code Igniter (Pembakar Kode) yang merupakan opensource. Eliss Lab mempunyai produk berbayar yang bernama Expression Engine dan Mojo Motor. Mereka telah berdiri sejak 2001, jadi bukan hal yang mengejutkan jika produk mereka yang gratis maupun berbayar saat ini telah dikenal di dunia.

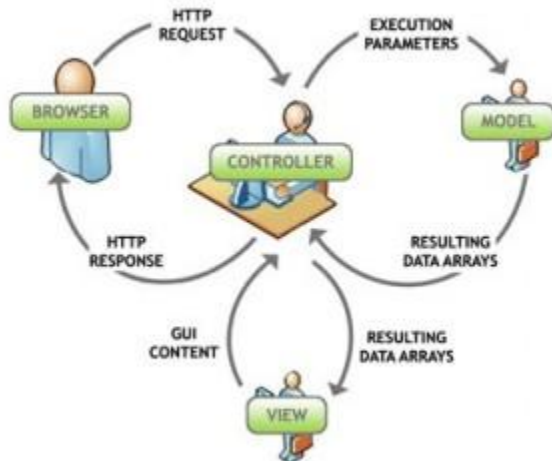
Konsep MVC Framework PHP

Codeigniter Framework adalah salah satu dari sekian framework yang menggunakan sistem MVC (Model-View-Controller). Jangan bingung dulu di awal, banyak programmer PHP di forum-forum luar mengatakan **KISS (Keep It Simple Stupid!) with MVC**.

Pengertian Framework merupakan suatu perangkat lunak (software) yang bersifat penggunaan ulang suatu library atau classes yang ada di dalam sistem tersebut untuk menjadi suatu aplikasi. Di dalam framework PHP, sudah terdapat beberapa *classes*, *objects* dan *library*. Dengan begitu ketika kita membuat aplikasi menggunakan framework, kita bisa menggunakan fitur yang sudah ada di dalam software tersebut.

Sebagai contoh, kita ingin mengirimkan email dan di dalam framework sudah ada library email, jadi tidak usah membuat code pengirim email, cukup kita panggil function library email. Di dalam Framework Codeigniter sendiri telah terdapat beberapa library di dalamnya, seperti library email, cart (keranjang belanja web ecommerce), validasi form, pagination, session, dan lain-lain.

MVC itu sendiri adalah suatu konsep **Model – View – Controller**. Pada dasarnya dalam konsep MVC ini, **Model** merupakan pengolah data dan berkomunikasi dengan database, **View** adalah tampilan yang ditampilkan ke user, sedangkan **Controller** merupakan pengolah request user dan mengembalikan request user dalam bentuk View.



Lain kali akan saya jelaskan dalam implementasinya di Framework Code Igniter kalau sempat. Masalahnya, selain saya sedang sibuk membuat php mvc framework versi saya sendiri, juga sudah begitu banyak tutorial yang ada di luar sana tentang Code Igniter. Oke, inilah singkat penjelasan tentang **MVC Framework PHP**.

Pengenalan Framework PHP CodeIgniter, Bab Konfigurasi CodeIgniter

CodeIgniter tidak terlalu sulit, apalagi bagi anda yang sudah menguasai PHP dan lebih mudah lagi apabila anda memahami PHP yang berbasis OOP(Object Oriented Programming) maupun konsep MVC(Model View Controller). Framework Codeigniter berbasiskan konsep MVC yang memudahkan programmer dalam mengolah data.

Sebelum memulai membangun aplikasi web, alangkah baiknya mengetahui konfigurasi awal yang harus dilakukan sebelum menggunakan framework Codeigniter ini.

Pertama kita lihat file **config.php** di folder `%CI_Root%\system\application\config`

terdapat kode

```
$config['base_url'] = "";
```

ini adalah konfigurasi untuk mengakses alamat dari aplikasi yang akan kita bangun. Contoh

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

atau

```
$config['base_url'] = "http://192.168.0.114/xxis/";
```

xxis disini adalah nama folder root dari aplikasi CI yang kita bangun. Jadi apabila nama folder rootnya

adalah **Contoh**, maka kodenya harus

```
$config['base_url'] = "http://192.168.0.114/Contoh/";
```

Selanjutnya kita lihat file **database.php** yang masih se folder dengan file config.php

terdapat kode

```
$db['default']['hostname'] = "localhost";
```

kode ini untuk mendefinisikan host dari database yg akan kita pakai, secara default hostnya

adalah localhost

```
$db['default']['username'] = "postgres";
```

kode ini untuk mendefinisikan username dari pengguna database (secara default untuk MySQL : root, Postgre : postgre dan oracle : system).

```
$db['default']['password'] = "xxis";
```

merupakan password yang kita pakai untuk mengakses database

```
$db['default']['database'] = "xxis";
```

merupakan database mana yang akan kita pilih

```
$db['default']['dbdriver'] = "postgre";
```

merupakan driver untuk mengakses database (untuk MySQL : mysql, postgresql : postgre, sedang oracle : oci8).

```
$db['default']['port'] = 5432;
```

port default CI khusus untuk mengakses database PostgreSQL. Jika selain PostgreSQL,

```
$db['default']['port'] = 5432;
```

 bisa diabaikan/dihilangkan.

untuk kode-kode lain yang ada di file database.php bisa diabaikan sesuai dengan nilai defaultnya.

Selanjutnya mari kita menjelajah lagi ke file **autoload.php** yang juga masih se folder dengan file config.php dan database.php. Disini adalah file untuk menyimpan konfigurasi mengenai file yang akan di load pertama kali saat aplikasi dijalankan.

contoh

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

autoload file database.php, session.php dan site_sentry.php yang ada di folder libraries

```
$autoload['helper'] = array('url', 'form', 'html');
```

autoload file url_helper.php, form_helper.php dan html_helper.php yang ada di folder helper file ini bisa kita abaikan jika kita tidak menginginkan autoload.

dan yang terakhir dari folder config yang ingin kita bahas adalah file **router.php** terdapat kode

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

kode di atas adalah untuk memanggil controller welcome saat pertama kali dijalankan. Jadi saat aplikasi pertama kali dijalankan, controller welcome.php akan pertama kali dieksekusi.

contoh apabila kita klik address

```
http://localhost/xxis/
```

maka secara otomatis akan menjadi

```
http://localhost/xxis/index.php/welcome
```

apabila kode kita ubah menjadi

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

maka untuk mengakses controller welcome.php harus dengan

```
http://localhost/xxis/index.php/welcome
```

karena apabila kita mengakses hanya dengan "http://localhost/xxis/" atau "http://localhost/xxis/index.php/"

framework Codeigniter tidak dapat mendefinisikan controller mana yang harus diakses.

Instalasi/Konfigurasi

- Instalasi maupun konfigurasi framework CodeIgniter terletak pada direktori

system/application/config

- Secara umum ada 4 file yang perlu dilakukan perubahan untuk konfigurasi/instalasi, yaitu :

- Config.php

- Autoload.php

- Database.php

- Routes.php

Pada file config.php

- Ubahlah baris program berikut :

```
$config['base_url'] = "http://example.com/";
```

- Menjadi URL project, misalnya:

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

- Base URL merupakan alamat default dari website/aplikasi yg dibuat.

- Secara default URL tersebut akan digunakan sebagai awal link dalam halaman website/aplikasi.

Pada file database.php

- Ubahlah baris program berikut :

```
$db['default']['hostname'] = "localhost";
```

```
$db['default']['username'] = "";
```

```
$db['default']['password'] = "";
```

```
$db['default']['database'] = "";
```

–Menjadi sesuai konfigurasi koneksi database server, misalnya:

```
$db['default']['hostname'] = "localhost";
```

```
$db['default']['username'] = "root";
```

```
$db['default']['password'] = "";
```

```
$db['default']['database'] = "latihandb";
```

•Untuk kebutuhan konfigurasi di atas, nama database harus dibuat terlebih dahulu

Pada file autoload.php

–Ubahlah baris program berikut :

```
$autoload['libraries'] = array();
```

–Menjadi :

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

•Kumpulan library di atas merupakan library yang akan diaktifkan secara otomatis diawal program.

•Beberapa library yang umum digunakan adalah database, session, validation, form_validation, dll.

Pada file autoload.php

–Ubahlah baris program berikut :


```
$autoload['helper'] = array();
```

–Menjadi :

```
$autoload['helper'] = array('url', 'file', 'form', 'table');
```

- Helper merupakan kumpulan fungsi-fungsi dalam berbagai kategori untuk memudahkan pengembangan aplikasi.

- Kumpulan helper di atas merupakan helper yang akan diaktifkan secara otomatis diawal program.

- Beberapa helper yang umum digunakan adalah url, file, form, table, dll.

Pada file routes.php

–Ubahlah baris program berikut :

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

–Menjadi default controller kita, misalnya:

```
$route['default_controller'] = "utama";
```

- Default controller ini dimaksudkan sebagai controller yang akan dijalankan pertama kali oleh user.

- Contoh :

<http://localhost/latihan/index.php/utama/index>

- Maka hanya perlu ditulis:

<http://localhost/latihan>