

Chapter 1

Perkenalan CodeIgniter 2.0

Akhir-akhir ini CodeIgniter menjadi sebuah framework yang hangat dibicarakan di Indonesia. Hampir semua milis dan forum PHP banyak membahas masalah CodeIgniter. Terlebih lagi banyak perusahaan-perusahaan ternama di Indonesia (Kompas.com, okezone.com, urbanesia.com, bejubel.com, dan lain-lain) yang telah menggunakan CodeIgniter dalam produk mereka. Apa sih CodeIgniter? CodeIgniter adalah sebuah framework PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Ada banyak library dan helper yang berguna didalamnya dan tentunya mempermudah proses development. Ibarat ingin membangun rumah maka Anda tidak perlu membuat semen, memotong kayu menjadi papan, mengubah batu menjadi porselen dan lain-lain. Anda cukup memilih komponen-komponen tersebut lalu dikombinasikan menjadi rumah yang indah.



Gambar1. Website-website indonesia yang menggunakan Codeigniter

Jadi keuntungan yang didapat dalam penggunaan framework adalah :

- **Menghemat Waktu Pengembangan** – Dengan struktur dan library yang telah disediakan oleh framework maka tidak perlu lagi memikirkan hal-hal tersebut, jadi Anda hanya fokus ke proses bisnis yang akan dikerjakan.
- **Reuse of code** – Dengan menggunakan framework maka pekerjaan kita akan memiliki struktur yang baku, sehingga kita dapat menggunakannya kembali di proyek-proyek lainnya.
- **Bantuan komunitas** - Ada komunitas-komunitas yang siap membantu jika ada permasalahan, selain itu juga bisa berbagi ilmu sehingga dapat meningkatkan kemampuan pemrograman kita.
- **Kumpulan best practice** – sebuah framework merupakan kumpulan *best practice* yang sudah teruji. Jadi kita dapat meningkatkan kualitas kode kita.

Catatan:

Framework adalah sebuah struktur konseptual dasar yang digunakan untuk memecahkan sebuah permasalahan atau isu-isu kompleks

Sebelum mendalami CodeIgniter lebih jauh, sebaiknya dipahami terlebih dahulu apa itu framework. Framework adalah sebuah struktur konseptual dasar yang digunakan untuk memecahkan sebuah permasalahan, bahkan isu-isu kompleks yang ada. Sebuah framework telah berisi sekumpulan arsitektur/konsep-konsep yang dapat mempermudah dalam pemecahan sebuah permasalahan. Perlu diingat, framework bukanlah peralatan/tools untuk memecahkan sebuah masalah, tetapi sebagai **ALAT BANTU**. Framework hanya menjadi sebuah konstruksi dasar yang menopang sebuah konsep atau sistem yang bersifat “*essential support*” (penting tapi bukan komponen utama).

Kenapa Menggunakan Framework?

Salah satu alasan mengapa orang menggunakan framework terutama dalam membangun sebuah aplikasi adalah kemudahan yang ditawarkan. Didalam sebuah framework biasanya sudah tersedia struktur aplikasi yang baik, ***standard coding*** (1), ***best practice*** (2) dan ***design pattern*** (3), dan ***common function*** (4). Dengan menggunakan framework kita dapat langsung fokus kepada *business process* yang dihadapi tanpa harus berfikir banyak masalah struktur aplikasi, standar coding dan lain-lain.

Dengan memanfaatkan design pattern dan *common function* yang telah ada di dalam framework maka hal tersebut dapat mempercepat proses pengembangan aplikasi. Kita tidak perlu membuat sesuatu fungsionalitas yang bersifat umum. Tanpa disadari ketika kita membangun sebuah aplikasi yang banyak melibatkan banyak fungsionalitas yang telah dibangun itu ternyata sama atau berulang-ulang. Dengan pengelempokkan itulah kita dapat mempercepat pengembangan aplikasi.

Selain kemudahan dan kecepatan dalam membangun sistem, dengan menggunakan framework tertentu kita juga dapat “menyeragamkan” cara kita mengimplementasikan kode program. Dengan framework kita akan “dipaksa” untuk patuh kepada sebuah kesepakatan. Selain itu juga akan memudahkan pengembang lain untuk mempelajari dan mengubah aplikasi yang telah dibuat apabila kode yang dihasilkan konsisten dan patuh pada sebuah aturan tertentu.

¹Standar Coding adalah sebuah standar yang harus diikuti oleh programmer untuk menulis code.

² Best Practice adalah Kumpulan-kumpulan action yang telah teruji oleh para expert

³ Design Pattern adalah Teknik-teknik yang menjadi best practise

⁴ Common Function adalah fungsi-fungsi atau library yang telah umum digunakan dalam pengembangan sebuah sistem

Apa itu Codeigniter?

CodeIgniter adalah sebuah web application framework yang bersifat open source digunakan untuk membangun aplikasi php dinamis. Tujuan utama pengembangan Codeigniter adalah untuk membantu developer untuk mengerjakan aplikasi lebih cepat daripada menulis semua code dari awal. Codeigniter menyediakan berbagai macam library yang dapat mempermudah dalam pengembangan. CodeIgniter diperkenalkan kepada public pada tanggal 28 februari 2006.

CodeIgniter sendiri dibangun menggunakan konsep Model-View-Controller development pattern. CodeIgniter sendiri merupakan salah satu framework tercepat dibandingkan dengan framework lainnya. Pada acara frOSCon (August 2008), pembuat php Rasmus Lerdorf mengatakan dia menyukai codeigniter karena dia lebih ringan dan cepat dibandingkan framework lainnya ("because it is faster, lighter and the least like a framework.")

Apa sih Kelebihan Codeigniter?

CodeIgniter sangat ringan, terstruktur, mudah dipelajari, dokumentasi lengkap dan dukungan yang luar biasa dari forum CodeIgniter. Selain itu CodeIgniter juga memiliki fitur-fitur lainnya yang sangat bermanfaat, antara lain:

- **Menggunakan Pattern MVC.** Dengan menggunakan pattern MVC ini, struktur kode yang dihasilkan menjadi lebih terstruktur dan memiliki standar yang jelas.
- **URL Friendly.** URL yang dihasilkan sangat *url friendly*. Pada CodeIgniter diminimalisasi penggunaan \$_GET dan di gantikan dengan URI.
- **Kemudahan.** Kemudahan dalam mempelajari, membuat library dan helper, memodifikasi serta meng-integrasikan Library dan helper.

Jika kita membandingkan antara CodeIgniter dengan framework-framework lainnya maka beberapa poin yang membuat CodeIgniter unggul adalah:

- **Kecepatan.** Berdasarkan hasil benchmark CodeIgniter merupakan salah satu framework PHP tercepat yang ada saat ini.
- **Mudah dimodifikasi dan beradaptasi.** Sangat mudah memodifikasi *behavior* framework ini. Tidak membutuhkan *server requirement* yang macam-macam serta mudah mengadopsi library lainnya.
- **Dokumentasi lengkap dan jelas.** Bahkan tanpa buku ini pun CodeIgniter sebenarnya telah menyediakan sebuah panduan yang lengkap mengenai CodeIgniter. Semua informasi yang anda butuhkan tentang codeigniter ada disana.
- **Learning Curve Rendah.** CodeIgniter sangat mudah dipelajari. Dalam pemilihan framework hal ini sangat penting diperhatikan karena kita juga harus memperhatikan *skill* dari seluruh

anggota team. Jika sebuah framework sangat sulit dipelajari maka akan beresiko untuk memperlambat team development anda.

CodeIgniter 2.0

CodeIgniter versi 2.0 baru saja diluncurkan pada awal tahun 2011. Pada versi terbaru ini, terdapat banyak perubahan mendasar dari CodeIgniter daripada versi sebelumnya. Perubahan ini menjadikan CodeIgniter jauh lebih kaya dan matang dibandingkan framework lainnya. Perubahan itu diantaranya adalah:

- Menghilangkan dukungan terhadap PHP4. PHP4 sudah tidak didukung oleh tim pengembang PHP, karena memberikan dukungan untuk PHP4 membuat CodeIgniter semakin ketinggalan dari segi fitur.
- Menghilangkan fitur plugin. Plugin mirip dengan helper, bertujuan untuk menghilangkan kerancuan ini maka fitur ini dihilangkan pada CodeIgniter 2.0.
- Menghilangkan fitur scaffolding. Fitur ini nyaris tidak pernah digunakan dan implementasinya masih kurang bagus.
- Penambahan library driver. Ini adalah library khusus dimana kita dapat membuat driver dari library yang telah kita buat.
- Support query string dan command-line execution. Hal ini menjawab kesulitan-kesulitan yang dialami pada versi sebelumnya.
- Penambahan library cache. Untuk meningkatkan kualitas aplikasi maka library cache baik menggunakan apc, memcached maupun file base.
- Penambahan fitur package. Untuk mempermudah distribusi resource dalam sebuah folder.

Apa itu MVC?

MVC adalah konsep dasar yang harus diketahui sebelum mengenal CodeIgniter . MVC adalah singkatan dari Model View Controller. MVC sebenarnya adalah sebuah pattern/teknik pemograman yang memisahkan bisnis logic (alur pikir), data logic (penyimpanan data) dan presentation logic (antarmuka aplikasi) atau secara sederhana adalah memisahkan antara desain, data dan proses. Adapun komponen-komponen MVC antara lain:

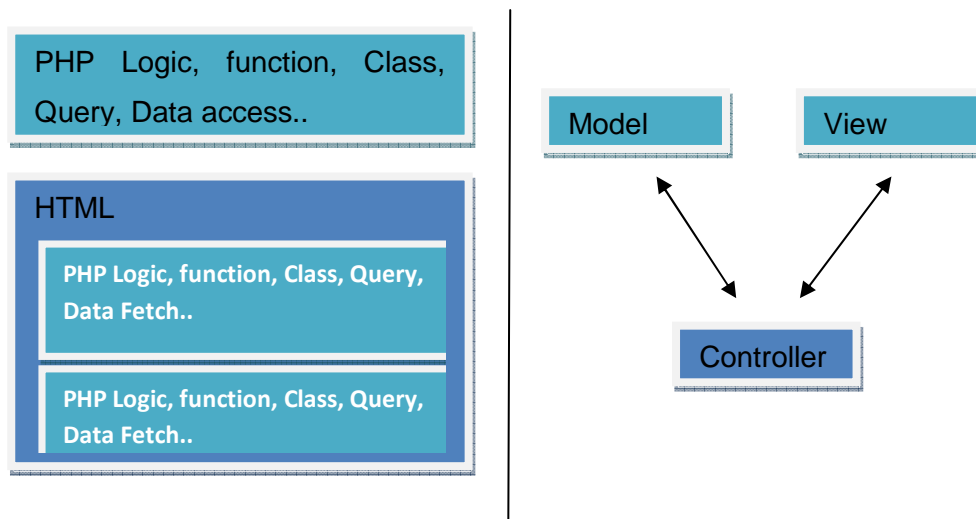
1. Model

Model berhubungan dengan data dan interaksi ke database atau webservice. Model juga merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, file XML maupun webservice. Biasanya di dalam model akan berisi class dan fungsi untuk mengambil, melakukan update dan menghapus data website. Sebuah aplikasi web biasanya menggunakan basis data dalam menyimpan data, maka pada bagian Model biasanya akan berhubungan dengan perintah-perintah query SQL.

2. View

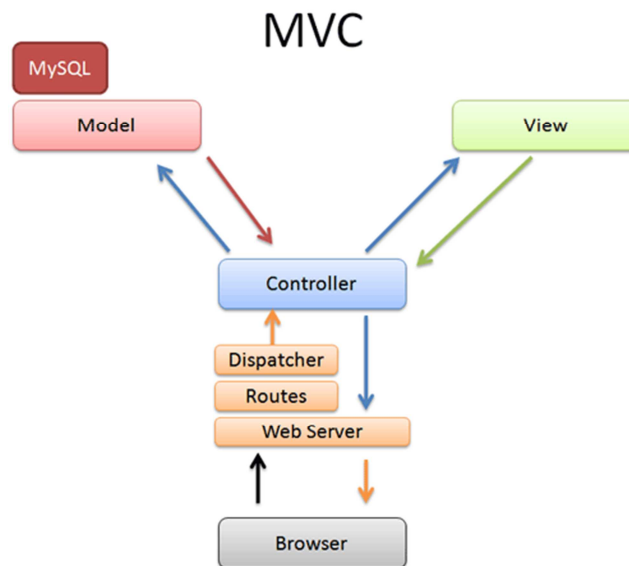
View berhubungan dengan segala sesuatu yang akan ditampilkan ke *end-user*. Bisa berupa halaman web, rss, javascript dan lain-lain. Kita harus menghindari adanya logika atau pemrosesan data di view. Di dalam view hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. View dapat dikatakan sebagai halaman website yang dibuat dengan menggunakan HTML dan bantuan CSS atau JavaScript. Di dalam view jangan pernah ada kode untuk melakukan koneksi ke basisdata. View hanya dikhususkan untuk menampilkan data-data hasil dari model dan controller

3. Controller: Controller bertindak sebagai penghubung data dan view. Di dalam Controller inilah terdapat class-class dan fungsi-fungsi yang memproses permintaan dari View ke dalam struktur data di dalam Model. Controller juga tidak boleh berisi kode untuk mengakses basis data karena tugas mengakses data telah diserahkan kepada model. Tugas controller adalah menyediakan berbagai variabel yang akan ditampilkan di view, memanggil model untuk melakukan akses ke basis data, menyediakan penanganan kesalahan/error, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap input.



Gambar 2. Perbandingan PHP Biasa dan CodeIgniter

Perhatikan gambar 2. Pada eksekusi PHP, biasanya kita akan me-“load” semua library dan fungsi yang dibutuhkan kemudian digabungkan ke dalam HTML untuk di eksekusi oleh PHP. Untuk kasus sederhana cara tersebut masih baik-baik saja, tetapi ketika aplikasi tersebut menjadi kompleks/rumit maka kita akan sulit memeliharanya jika tidak didukung oleh arsitektur software yang bagus. Hal tersebut bisa terjadi disebabkan oleh code yang sama namun dibuat berulang-ulang, kode tidak konsisten dan lain-lain.



Gambar 3. Flow CodeIgniter

Jika dipetakan, alur kerja CodeIgniter akan tampak seperti gambar 3. Browser berinteraksi melalui controller. Controller-lah yang akan menerima dan membalas semua *request* dari browser. Untuk data maka controller akan meminta ke Model dan untuk UI/template akan meminta ke View. Jadi “Otak” dari aplikasi ada di **controller**, “Muka” aplikasi ada di **view** dan “Data” ada di **model**. Ketika

browser meminta sebuah halaman web maka router akan mencari controller mana yang harus menangani *request* tersebut. Setelah itu barulah si controller menggunakan model untuk mengakses data dan View untuk menampilkan data tersebut.

Jangan Belajar CodeIgniter!!

Jangan coba-coba belajar CodeIgniter ketika Anda belum mengenal PHP. Berdasarkan pengalaman di forum dan milis, kebanyakan mereka yang belajar CodeIgniter tanpa memiliki dasar PHP yang baik akan mengalami banyak kesulitan, bukan dalam menguasai konsep CodeIgniter tetapi masih berkutat seputar PHP. Setidaknya Anda telah memahami konsep OOP pada PHP untuk mulai belajar CodeIgniter. Sebaiknya ketika Anda mempelajari CodeIgniter Anda sudah memahami PHP dan Object Oriented Programming. Akan lebih bagus lagi jika Anda sudah familiar dengan design pattern.