



Assalamu'alaikum teman-teman 3IA10 yang lagi mengerjakan PI tentang Web. Pertama, kalau ada kesalahan kata atau penjelasan mohon di maapin yaaa gue juga masih belajar :) Yaudah cukup deh basa-basi basi nya langsung aja ya dimulai...

## Framework, itu apaan sih ?

Nah... Diantara kalian pasti ada yang mikir :

- sebenarnya **Framework** itu apa sih ?
- Sebenarnya kegunaan **Framework** itu apa sih ?
- Sebenarnya **Codeigniter** itu apa sih ?
- Bedanya sama **PHP Native (Murni)** itu apa sih ?

Nah gue coba menjelaskan apa itu **Framework**, menurut pemahaman gue framework itu adalah alat bantunya programmer dalam mengerjakan sebuah aplikasi baik itu proyek iseng-iseng milik pribadi ataupun proyek pesanan client, kenapa disebut alat bantu ?

1. Framework itu memangkas waktu lu dalam mengerjakan aplikasi, ya! Bener-bener kepangkas!
2. banyak hal yang lu butuhin dalam pengerjaan aplikasi sudah tersedia di dalam framework.

Udah mulai paham belum ? Kalo masih belum paham gue coba buat analoginya nih



*"Misal lu lagi membuat rumah, nah rumah kan terdiri dari atap, tembok, pintu, kaca, ubin, kayu. Lu pasti bakal beli itu semua di material kan ? Trus setelah lu beli lu bakal suruh tukang untuk merangkai apa yang udah lu beli semua kan ? **Framework** itu bisa di ibaratin gitu, lu hanya perlu merangkai apa yang sudah ada di dalam **framework**. Ga ada ceritanya kan lu buat rumah tapi lu juga yang buat batu bata nya dari tanah, trus mahat sendiri pintu rumahnya, buat sendiri ubinnya, buat sendiri kaca nya dan itu memakan waktu kalian kan ketimbang beli langsung di material ? Fungsi framework tuh sama seperti itu."*

Sampai segini sudah paham belum apa itu framework ?

Kalo lu udah ngerasa paham gw mau coba test pemahaman lu tentang framework nih.

## 1. Apa sih framework itu ?

Buat jawabnya Jangan baca penjelasan gue di atas ya! Kalo belum

bisa jawab pertanyaan itu coba resapin lagi penjelasan yang gue tulis di atas ok! Semoga cepat paham! :)

Lanjut... **Kegunaan Framework.** Setelah membaca penjelasan gue diatas mungkin masih bingung kan kegunaan si framework itu apaansih... Gue mau mencoba menjelaskan nih apasih kegunaan inti dari sebuah Framework, gue buat sebuah permisalan yah...

*"Contohnya lu lagi ngerjain proyek pribadi untuk latihan didalam aplikasi lu ada fitur **login, paginasi (paginasi** tuh semisal kita buka sebuah halaman nah dibawahnya tuh ada kayak nomor, disetiap nomornya di klik bakal keganti ke halaman selanjutnya) trus setelah aplikasi iseng lu selesai lu dapet orderan nih dari client yang fiturnya sama kayak aplikasi iseng lu yang udah selesai lu buat, lu pasti bakal mikir kan **"apa gue copas aja ya codingan gue untuk login dan paginasi ke aplikasi pesanan client ?"** Disinalah kegunaan dari framework, dia udah menyediakan fungsi yang mempermudah lu untuk membuat fitur-fitur yang udah umum digunakan dalam pembuatan web, jadi lu gausah membuat fitur dua kali untuk aplikasi yang beda dengan fungsi yang sama. Semua fungsi umum itu sudah tersedia di dalam framework."*

Mungkin ada yang mikir juga, yang punya framework itu cuman bahasa pemrograman PHP doang bukan sih ?

Hampir di setiap bahasa pemrograman ada frameworknya tersendiri yang tentunya inti dari fungsi nya sama, mempermudah dan mempercepat kita dalam mem "build" sebuah aplikasi. Nih gue coba list setiap bahasa pemrograman dengan framework yang mendukung :

- **PHP**

- A. Laravel Framework
- B. Codeigniter Framework
- C. Symfony Framework
- D. CakePHP

E. Phalcon Framework

F. Yii Framework

- **Python**

A. Django Framework

- **Visual Basic**

A. .NET Framework, Visual Basic + .NET Framework biasa dikenal VB.NET

- **Ruby**

A. Ruby On Rails

Nah ada pertanyaan lagi nih dari gue untuk memastikan kalo lu pada paham kegunaan dari framework, coba jawab pertanyaan gue tanpa nyontek keatas, jujur sama diri sendiri biar lu tau sampe mana pemahaman lu tentang materi ini

## **1. Apa sih Kegunaan Framework ?**

Semoga bisa cepat paham!

### **Apasih Codeigniter itu ? Spesies apa ? Darimana ?!**

Masuk pada pembahasan selanjutnya yang lebih mengerucut, karena Codeigniter adalah salah satu framework yang cukup populer, jujur saat pertama gue banting stir ke dunia development web yang keras ini wkwkwk framework yang pertama gue coba adalah Codeigniter. Pertama gue make ini gue dihadapkan dengan rasa bingung yang bener-bener bikin bingung dan ngebuat gue berkali-kali

harus ngebuka dokumentasinya Codeigniter yang dapat lu akses melalui link ini :

Codeigniter versi 3.x

[http://www.codeigniter.com/user\\_guide/](http://www.codeigniter.com/user_guide/)

Codeigniter versi 2.x

<https://ellislab.com/codeigniter/user-guide/>

Inget bruh membaca itu penting kalo lu mau terjun dalam Web Development, hilangkan sifat malas membaca, setiap masalah yang umum soal framework ini bisa lu search dan biasanya akan berujung pada website Stackoverflow :)

Ada salah satu quote dari teman sekelas kita, bunyinya gini :

*"Baca Dokumentasi dulu"* - **Arief Hikam**.

Dari Codeigniter ini gue mengenal **Design Pattern** MVC (Menurut pendapat gue pengertian dari **Design Pattern** itu sendiri adalah solusi dalam pemrograman agar setiap class atau modul yang kita bikin lebih tertata, elegan, dan reuseable / dapat digunakan kembali) , nah apalagi itu MVC ? ~~Gue udah muaaaaaaaak belajar pemrograman pengen nikah ajaaaaaaa~~ MVC itu jenis Design Pattern yang banyak digunakan di dunia ini. Konsep dari MVC itu sendiri memisahkan antara tampilan, koneksi ke database, dan pengontrol dari keduanya.

Jadi di MVC ada itu terdapat bagian untuk tampilan, bagian untuk koneksi ke database, bagian untuk mengontrol agar data dari database bisa di munculin ke tampilan web kita. Penjelasan simple nya sih begitu...

Di MVC ada 3 bagian yang harus kita pahami bersama gan / sis.

- **Model**

Yoms... Biar gue tebak pas pertama kali lu baca kata "Model" hal yang terlintang di kepala lu pertama kali itu cewek cakep, bohay, tobron. Yakan yakan ? Jujur dah ! :p hahaha Bukan... Bukan... Model disini tuh bukan yang kayak begitu :D

Model di konsep Design Pattern MVC itu memiliki peranan yang menurut gue paling penting nih, soalnya di bagian inilah data kita

diolah, di bagian ini data kita ditarik dari database, di bagian ini data kita di tambahkan ke database, di bagian ini kita bisa mengupdate data, menghapus data dari database.

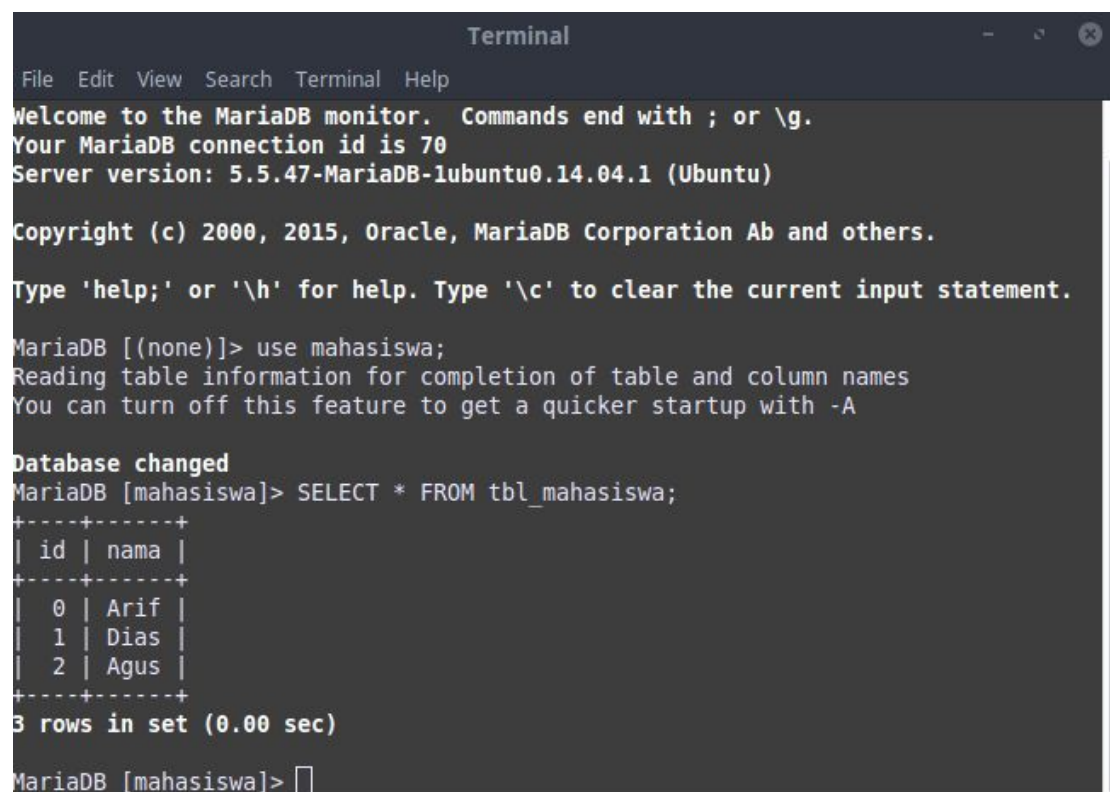
### ***Trus fungsinya framework dalam model ini apaan ?***

Oke mungkin ini pertanyaan nongol di kepala kalian, jadi kalian pasti udah pada bisa dong yah untuk select data dari database mysql, contohnya gini

Di Console Mysql (itu tuh kalo di labti kita praktek yang ngetik di command prompt **mysql -u root -p** itu dinamakan console atau Command Line)

**SELECT \* FROM tbl\_mahasiswa;**

Trus datanya nongol, bener gak ? Gue contohin dalam bentuk screenshot dibawah ini.



```
Terminal
File Edit View Search Terminal Help
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 70
Server version: 5.5.47-MariaDB-1ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use mahasiswa;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [mahasiswa]> SELECT * FROM tbl_mahasiswa;
+-----+-----+
| id | nama |
+-----+-----+
| 0 | Arif |
| 1 | Dias |
| 2 | Agus |
+-----+-----+
3 rows in set (0.00 sec)

MariaDB [mahasiswa]> 
```

Kalo di PHP Native (Tanpa Menggunakan Framework) kurang lebih begini kodingan untuk nongolin data dari database.

Class Database.php

```

<?php
class Database
{
    private static $dbName = 'ganteng' ;
    private static $dbHost = 'localhost' ;
    private static $dbUsername = 'root';
    private static $dbUserPassword = '';

    private static $cont  = null;

    public function __construct() {
        die('Init function is not allowed');
    }

    public static function connect()
    {
        if ( null == self::$cont )
        {
            try
            {
                self::$cont =  new
PDO( "mysql:host=".self::$dbHost.";"."dbname=".self::$dbName,
self::$dbUsername, self::$dbUserPassword);
            }
            catch(PDOException $e)
            {
                die($e->getMessage());
            }
        }
        return self::$cont;
    }

    public static function disconnect()
    {
        self::$cont = null;
    }
}

```

Eh jangan terlalu serius bacanya itu cuman contoh aja, cuman kalo mau tau lebih lanjut sih gpp dibaca-baca tentang koneksi database dari PHP Native.

baru class untuk koneksiinnya udah njelimet banget kan ?

Kalo di framework lu tinggal jalanin perintah :

```
$query = $this->db->get('tbl_mahasiswa');
```

Simple kan ? Ga percaya kalo sesimple itu untuk nampilin data ?  
Cobain sendiri...

Udah tambah lagi belum pemahaman soal kegunaan framework ? Itu salah satunya :)

Pokoknya nih di Model semua transaksi antara aplikasi ke database dilakukan disini...

- **Controller**

Nah... Fungsi controller tuh sebagai 'jembatan' antara Model dan Tampilan (View)

Di controller kita bisa menerima inputan data dari tampilan web kita lalu memanggil fungsi **model** untuk menginput data misalnya...  
Atau kita bisa melakukan manipulasi data misal si user menginputkan kata

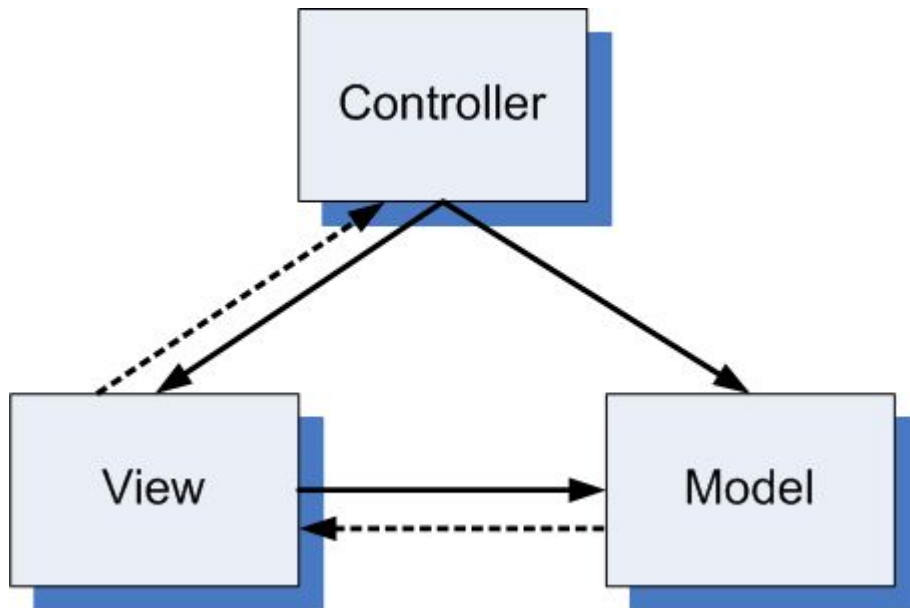
"Ganteng" lalu saat di input kita pengen si kata itu diganti jadi "anteng" lalu di input ke database, nah proses ini terjadi di bagian Controller.

- **View**

Fungsi utama dari View adalah sebagai tampilan untuk web kita, segala yang kita tampilkan tuh ada kaitannya ke View.  
Data yang kita kirim dari controller pun akan di tampilkan lewat View ini.

Biasanya di bagian view ini proses untuk meload data dilakukan, meload data disini dimaksudkan meload data yang di lempar dari bagian controller





Ini gambaran dari arsitektur MVC

Jadi intinya tuh Design Pattern MVC ( Model - View - Controller)  
Membantu kita agar proyek kita tertata dengan rapi sesuai dengan fungsi dan kategorinya, jadi ga tercampur antara bagian database dengan bagian tampilan.

Design Pattern ini sudah diterapkan di Codeigniter :)

## PHP Native vs Framework



Setelah membaca penjelasan gue diatas mengenai framework mungkin ada yang punya pertanyaan **“kenapa ga pakai PHP Native aja untuk development”**

Seperti yang udah dijelaskan sebelumnya Framework itu berfungsi untuk memudahkan kita dalam membangun aplikasi yang kita mau. Jadi kalau mau pakai PHP Native ya sah-sah saja cuman segala sesuatu yang dilakukan disini itu serba “sendiri” maksudnya kita harus buat fungsi sendiri, mengatur tentang keamanannya sendiri, mengatur aplikasi kita agar lebih elegan sendiri pokoknya apa-apa sendiri deh...

### **Kesimpulannya ?**

Menurut gue untuk memahami framework kita harus memahami dasar bahasa pemrogramannya dulu, contoh disini adalah PHP, jangan ujug-ujug langsung lompat ke framework, minimal kita bisa dan paham mengenai OOP (Object Oriented Programming) Karena pada Framework itu sudah Full OOP.

Yaudah mungkin sampe sini dulu pembahasan mengenai Framework, untuk selanjutnya akan dibahas mengenai struktur dari Framework Codeigniter yang akan kita gunakan beserta sample kode nya, setelah itu baru lanjut untuk implementasi ERD yang dibuat kang

Arief.

Terima Kasih telah membaca sampai habis :)

Salam,  
Dias Taufik Rahman

Colek : Mastah Agus, Mastah Arief :)