# LAPORAN PRAKTIKUM
# WEB FRAMEWORK

***Bagian 2***
## CODEIGNITER
## ( Konsep MCV dalam CodeIgniter)


Disusun oleh

Nama : Rizqa Alfiani

NIM/Golongan : E31191919/ C

Kelompok :

Dosen Pengampu :


Jember , 15-Maret - 2020
Disetujui


**Laboratorium Rekayasa Sistem Informasi**
**Jurusan Teknologi Informasi**
**Politeknik Negeri Jember**

**2020**

# I. Pendahuluan

## 1.1. Latar Belakang

Framework atau dalam bahasa indonesia dapat diartikan sebagai “kerangka kerja” merupakan kumpulan dari fungsi-fungsi/prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programer, tanpa harus membuat fungsi atau class dari awal.

Banyak framework yang dapat digunakan unutk pembanguan website agar lebi atraktif seperti pada javascript framework yang dapat digunakan vue.js, react.js, angular.js. pada PHP framework yang dapat digunakan laravel, codeIgniter, symfony dl. Pada materi ini akan di bahas tentang salah satu framework dari PHP yaitu codeIgnier yang menggunakan metode MVC ( model-view-controller ).

## 1.2. Masalah

1. Bagaimana Konsep dasar MVC
2. Bagaimana menerapkan MVC dalam CodeIgniter?

## 1.3. Tujuan

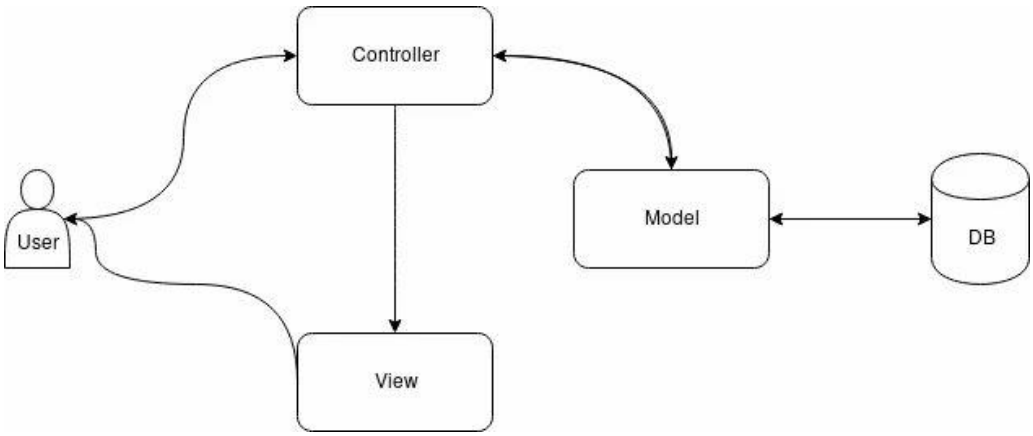
1. Memahami konsep dasar MVC
2. Memahami penerapan MVC dalam CodeIgniter.

# II. Teori

## 2.1. Konsep MVC

MVC adalah konsep arsitektur dalam pembangunan aplikasi berbasis web yang membagi aplikasi web menjadi 3 bagian besar. Yang mana setiap bagian memiliki tugas-tugas serta tanggung jawab masing-masing. Tiga bagian tersebut adalah: model, view dan controller.

- **Model**: Bertugas untuk mengatur, menyiapkan, memanipulasi dan mengorganisasikan data (dari database) sesuai dengan instruksi dari controller.
- **View**: Bertugas untuk menyajikan informasi (yang mudah dimengerti) kepada user sesuai dengan instruksi dari controller.
- **Controller**: Bertugas untuk mengatur apa yang harus dilakukan model, dan view mana yang harus ditampilkan berdasarkan permintaan dari user. Namun, terkadang permintaan dari user tidak selalu memerlukan aksi dari model. Misalnya seperti menampilkan halaman form untuk registrasi user.



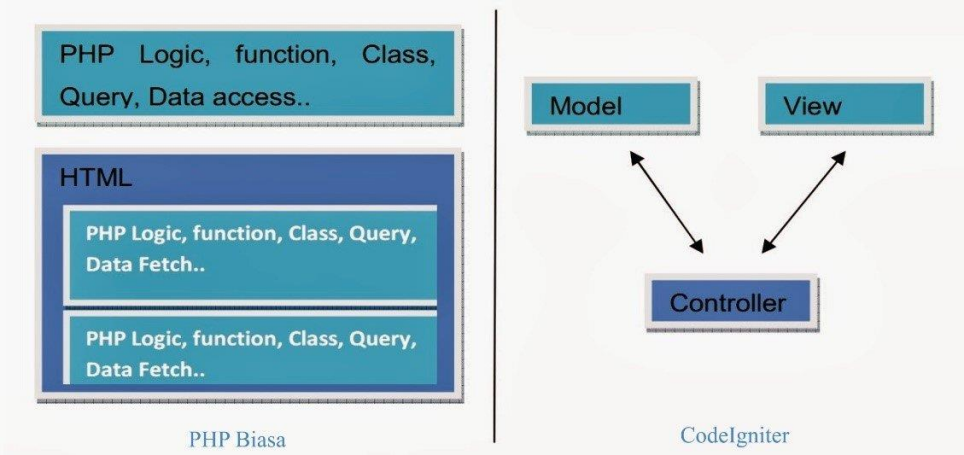

## 2.2. MVC dalam codeIgniter

CodeIgniter adalah sebuah web application network yang bersifat open source yang digunakan untuk membangun aplikasi php dinamis.

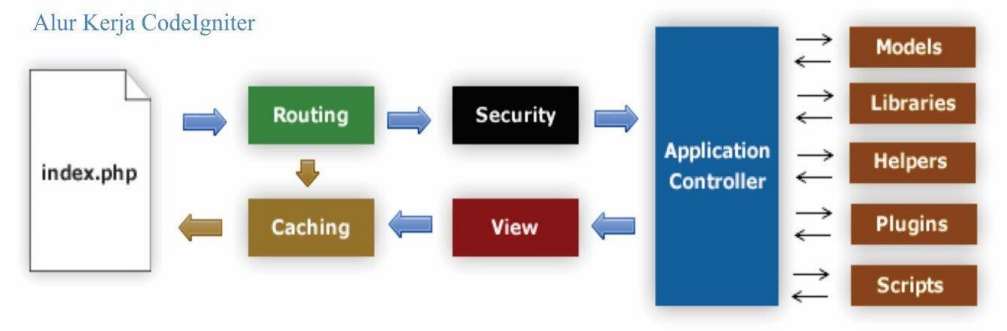
CodeIgniter menjadi sebuah framework PHP dengan model MVC untuk membangun website dinamis dengan menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Selain ringan dan cepat, CodeIgniter juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih CodeIgniter sebagai framework pilihannya. Karena kelebihan-kelebihan yang dimiliki oleh CodeIgniter, pembuat PHP Rasmus Lerdorf memuji CodeIgniter di frOSCon (Agustus 2008) dengan mengatakan bahwa dia menyukai CodeIgniter karena "it is faster, lighter and the least like a framework."

CodeIgniter pertamakali dikembangkan pada tahun 2006 oleh Rick Ellis . Dengan logo api yang menyala, CodeIgniter dengan cepat "membakar" semangat para web developer untuk mengembangkan web dinamis dengan cepat dan mudah menggunakan framework PHP yang satu ini.

**Perbandingan PHP Biasa dengan CodeIgniter**




**Alur Kerja Framework CodeIgniter**



- **Index.php**: Index.php disini berfungsi sebagai file pertama dalam program yang akan dibaca oleh program.
- **The Router**: Router akan memeriksa HTTP request untuk menentukan hal apa yang harus dilakukan oleh program.
- **Cache File**: Apabila dalam program sudah terdapat "cache file" maka file tersebut akan langsung dikirim ke browser. File cache inilah yang dapat membuat sebuah website dapat di buka dengan lebih cepat. Cache file dapat melewati proses yang sebenarnya harus dilakukan oleh program codeigniter.
- **Security**: Sebelum file controller di load keseluruhan, HTTP request dan data yang disubmit oleh user akan disaring terlebih dahulu melalui fasilitas security yang dimiliki oleh codeigniter.
- **Controller**: Controller akan membuka file model, core libraries, helper dan semua resources yang dibutuhkan dalam program tersebut.
- **View**: Hal yang terakhir akan dilakukan adalah membaca semua program yang ada dalam view file dan mengirimkannya ke browser supaya dapat dilihat. Apabila file view sudah ada yang di "cache" maka file view baru yang belum ter-cache akan mengupdate file view yang sudah ada.

# III. Pembahasan

## 2.1. Alat dan bahan

a. BKPM
b. Komputer/laptop
c. IDE Android Studio
d. Browser

## 2.2. Praktik BKPM

### 1.2.1 Praktikum

#### 1. Kode 1

routes.php

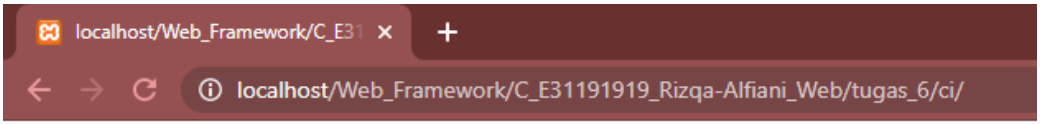
tugas_6 > ci > application > config > routes.php > ...

```
| Examples: my-controller/index -> my_controller/index
|   my-controller/my-method -> my_controller/my_method
*/
$route['default_controller'] = 'hello';
$route['404_override'] = '';
$route['translate_uri_dashes'] = FALSE;
```

hello.php

tugas_6 > ci > application > controllers > hello.php > ...

```
<?php
class Hello extends CI_Controller
{
    public function index()
    {
        echo "<h2>Hello World CI!</h2>";
    }
}
```




**Hello World CI!**

## 2. Kode 2

helloview.php

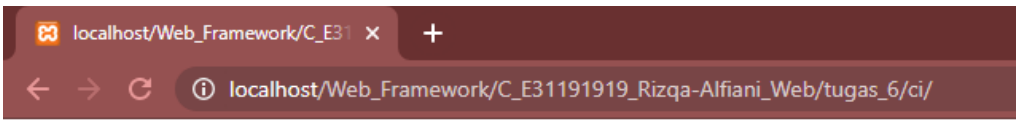
tugas_6 > ci > application > views > helloview.php > html > body

```
<html>

<head></head>

<body>

  <h2>Hello World CI !</h2>
  <h3>Menggunakan Controller dan view!</h3>

</body>

</html>
```

Hello.php

tugas_6 > ci > application > controllers > Hello.php > ...

```
<?php
class Hello extends CI_Controller
{
  public function index()
  {
    //echo "<h2>Hello World CI!</h2>";

    $this->load->view('helloview'); //memanggil file helloview
  }
}

```




# Hello World CI !

**Menggunakan Controller dan view!**

### 3. Kode 3

Hello.php

tugas_6 > ci > application > controllers > Hello.php > PHP Intelephense > Hello > index

```php
<?php

class Hello extends CI_Controller
{
    public function index()
    {

        //memuat model hello_model
        $this->load->model('Hello_model');

        //pengambilan object dari kelas Hello_model dan dimuat di var $model
        $model = $this->Hello_model;

        //mengambil data dari model
        $a = $model->txt;

        //membuat data yang akan dikirim ke view
        $data['text'] = $a;

        //memanggil file view
        $this->load->view('helloview', $data);
    }
}

```

Hello_model.php

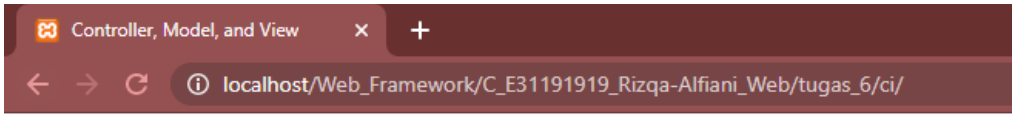
tugas_6 > ci > application > models > Hello_model.php > ...

```php
<?php

class Hello_model extends CI_Model{
    //membuat properti dengan nama var $txt
    public $txt = 'Hello World';
}

```

helloview.php

tugas_6 > ci > application > views > helloview.php > html

```
1   <html>
2
3   <head>
4     <title>Controller, Model, and View</title>
5   </head>
6
7   <body>
8     <h2><?php echo $text; ?></h2>
9     <h3>Menggunakan controller, model dan view</h3>
10  </body>
11
12  </html>
```




**Hello World**

**Menggunakan controller, model dan view**

### 4. Kode 4

variabel_view.php

tugas_6 > ci > application > views > variabel_view.php > html

```
1   <html>
2
3   <head>
4     <title>Controller dan view lebih dari 1 variabel</title>
5   </head>
6
7   <body>
8     <h2>mengirim data dari Controller ke view</h2>
9     <!--memanggil variable-->
10    variabel1: <?php echo $variabel1; ?><br>
11    variabel2: <?php echo $variabel2;
12                ?>
13  </body>
14
15  </html>
```

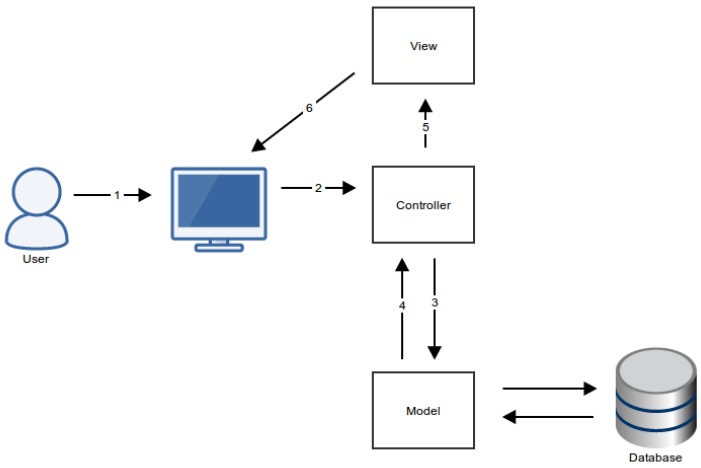
Variabel.php

tugas_6 > ci > application > controllers > Variabel.php > PHP Intelephense > Variabel > index > $data

```php
<?php

class Variabel extends CI_Controller
{
    public function index()
    {
        $data = ['variabel1' => 'Data variabel ke 1', 'variabel2' => 'Data variabel ke 2'];

        $this->load->view('variabel_view', $data);
    }
}

```


### 1.2.2 Tugas

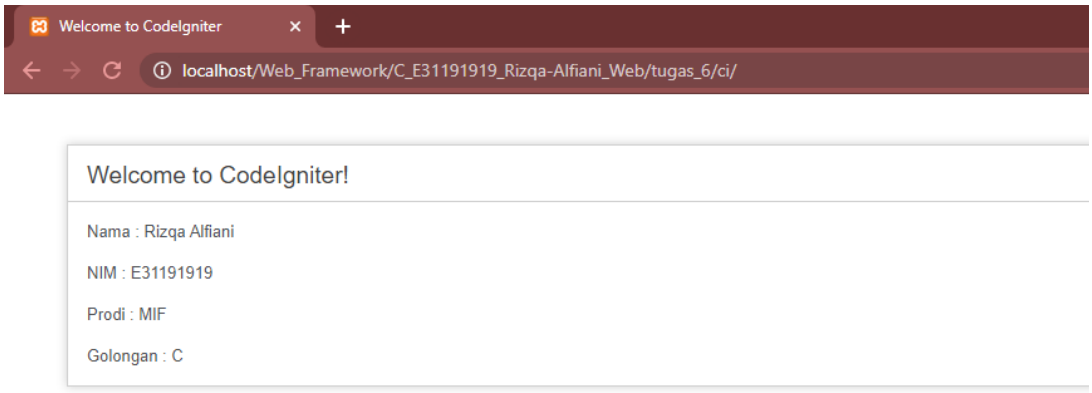
1. **Buatlah penjelasan bagaimana sebuah proses login pada sebuah web bekerja menggunakan konsep MVC (disertai gambar alur untuk penjelasan).**



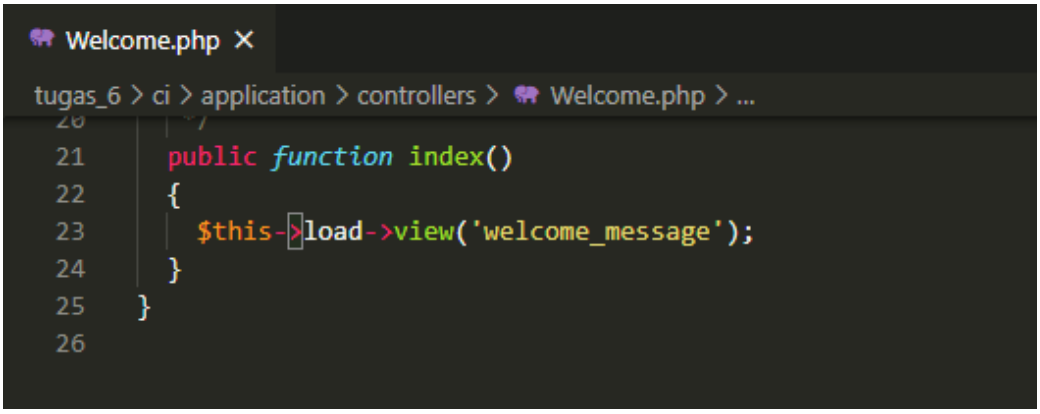

1. User menginpukan email dan password ke komputer
2. Komputer mengirimkan pada controller

3. Controller mengakses model dan akan diperiksa di database. Apakah datanya ada atau tidak.
4. Model mengirim hasil ke controller
5. Controller mengirimnya pada view
6. View menampilkan hasilnya di Komputer user.

2. **Rubahlah Tampilan welcome message pada CI sehingga menampilkan hello <nama anda><NIM><golongan>.**





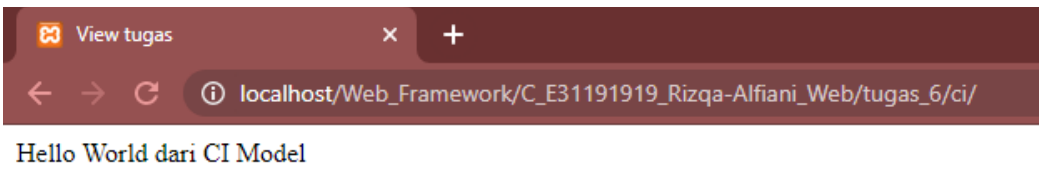


welcome_message.php

tugas_6 > ci > application > views > welcome_message.php > html > body > div#container

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');
?>
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>Welcome to CodeIgniter</title>

  <style type="text/css">···
  </style>
</head>

<body>

  <div id="container">
    <h1>Welcome to CodeIgniter!</h1>

    <div id="body">
      <p>Nama : Rizqa Alfiani</p>
      <p>NIM : E31191919</p>
      <p>Prodi : MIF</p>
      <p>Golongan : C</p>
      <!--  <p>The page you are looking at is being generated dynamicall

      <p>If you would like to edit this page you'll find it located at:</p
```

3. **Buatlah sebuah halaman yang menampilkan pesan "Hello Wolrd dari CI Model" dimana pesan tersebut terdapat di dalam model dan dipanggil dari controller.**




Hello World dari CI Model

hello_control.php | routes.php | view.php | hello.php

tugas_6 > ci > application > config > routes.php > ...

```
| Examples: my-controller/index -> my_controller/index
|   my-controller/my-method -> my_controller/my_method
*/
$route['default_controller'] = 'Hello_control';
$route['404_override'] = '';
$route['translate_uri_dashes'] = FALSE;
```

hello_control.php | routes.php | view.php | hello.php

tugas_6 > ci > application > controllers > hello_control.php > ...

```php
<?php
class Hello_control extends CI_Controller
{
  public function index()
  {
    // memuat hello model
    $this->load->model('hello');

    //pengambilan object dari hello
    $x = $this->hello;

    //ambil data dari model
    $ambil = $x->txt;

    //membuat data yg akan dikirim ke view
    $data['text'] = $ambil;

    //memanggil file view
    $this->load->view('view', $data);
  }
}
```

hello_control.php | routes.php | view.php | hello.php

tugas_6 > ci > application > views > view.php > html

```php
<html>

<head>
  <title>View tugas</title>
</head>

<body>
  <p><?php echo $text; ?></p>

</body>

</html>
```

hello_control.php | routes.php | view.php | hello.php

tugas_6 > ci > application > models > hello.php > PHP Intelephense > Hello > $txt

```php
<?php
class Hello extends CI_Model
{
  public $txt = "Hello World dari CI Model";
}
```

4. **Buatlah sebuah tampilan yang mengirim lebih dari 1 variabel menggunakan Controller,Model danView.**


routes.php

tugas_6 > ci > application > config > routes.php > ...

```
50  |   my-controller/my-method -> my_controller/my_method
51  */
52  $route['default_controller'] = 'tgs4_control';
53  $route['404_override'] = '';
54  $route['translate_uri_dashes'] = FALSE;
55
```

tgs4_control.php | tgs4_model.php | tgs4_view.php

tugas_6 > ci > application > controllers > tgs4_control.php > PHP Intelephense > Tgs4_control > index

```
1  <?php
2
3  class Tgs4_control extends CI_Controller
4  {
5
6    public function index()
7    {
8
9
10     $this->load->model('tgs4_model');
11
12     //ambil object
13     $x = $this->tgs4_model;
14
15     //mengambil data dari model
16     $y = $x->list;
17
18
19
20     //memanggil file view
21     $this->load->view('tgs4_view', $y);
22   }
23 }
24
```

tgs4_model.php | tgs4_view.php

tugas_6 > ci > application > models > tgs4_model.php > PHP Intelephense > Tgs4_model

```php
<?php

class Tgs4_model extends CI_Model
{

  public $list =
  [
    'list1' => 'ini variable 1',
    'list2' => 'ini variable 2'
  ];
}

```

tgs4_view.php

tugas_6 > ci > application > views > tgs4_view.php > html

```php
<html>

<head>
  <title>Controller dan view lebih dari 1 variabel</title>
</head>

<body>
  <h2>mengirim data dari Controller ke view</h2>
  <!--memanggil variable-->
  variabel1: <?php echo $list1; ?><br>
  variabel2: <?php echo $list2; ?><br>

</body>

</html>
```

# IV. Penutup

## 4.1. Kesimpulan

CodeIgniter adalah sebuah framework PHP yang menggunakan metode MVC (model-view-controller) dimana data, tampilan dan control di letakkan dalam file terpisah. Sehingga mempermudah para developer untuk berkolaborasi mengedit dan merubah tampilan atau data.

# Daftar Pustaka

BKPM Web Framework

idcouldhost.com(2017, 04  Agustus). Mengenal apa itu codeIgniter . Diakses pada 13 Maret 2021, dari https://idcloudhost.com/panduan/mengenal-apa-itu-framework-codeigniter/

jagongoding.com(2017, 27  Oktober). Apa itu MVC?. Diakses pada 13 Maret 2021, dari https://jagongoding.com/web/memahami-konsep-mvc/

Yusuf Supriatna(2018, 07 November). Mengenal Framewor CodeIgniter dan Memecahkan Masalah serta Mencari Solusi CodeIgniter. Diakses pada 13 Maret 2021, dari https://medium.com/@supriatna1607/mengenal-framework-codeigniter-dan-memecahkan-masalah-serta-mencari-solusi-codeigniter-3205826b4733