

# **LAPORAN TUGAS WEB FRAMEWORK**

*Minggu 6*

**Konsep MVC dan CodeIgniter**



Disusun oleh

Nama : Aldo Diva Saputra

NIM/Golongan : E31192079 / D

Jember , 16 Maret 2021

Disetujui

---

**Program Studi Manajemen Informatika  
Jurusan Teknologi Informasi  
Politeknik Negeri Jember  
2021**

# BAB 1. PENDAHULUAN

## 1.1 Latar Belakang

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.

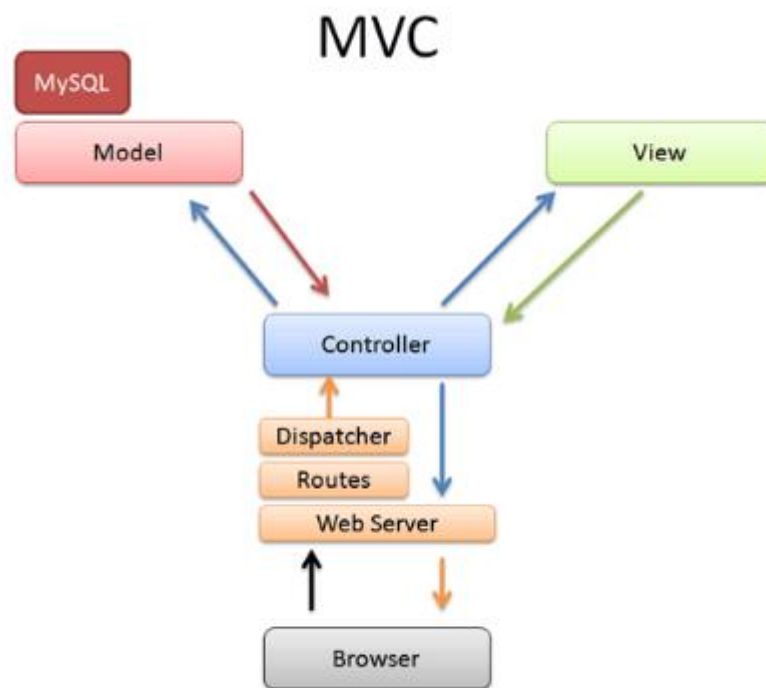
## Tujuan

1. Mahasiswa mampu memahami konsep dasar MVC
2. Mahasiswa mampu menerapkan konsep MVC pada CI

## BAB 2 DASAR TEORI

### Dasar Teori

Konsep MVC Konsep MVC memisahkan antara proses pengolahan data ke database (Model), Tempat Pengolahan Request atau bussines logic aplikasinya di Controller, dan Penampilan datanya hanya di View. skemanya seperti Gambar 1.1. Ketika kita megetikan sebuah alamat semisal <http://ci.com>, Proses itu langsung di ambil alih oleh controller, setelah itu di controller jika ada data yang ingin di tampilkan dari database, dia akan memanggil model dan menanyakan data, jika datanya ada, maka akan di kembalikan lagi di controller kemudian diolah di controller setelah itu baru data tersebut di kirim ke View dan siap di tampilkan



**Gambar 1.1. Konsep MVC**

## CodeIgniter

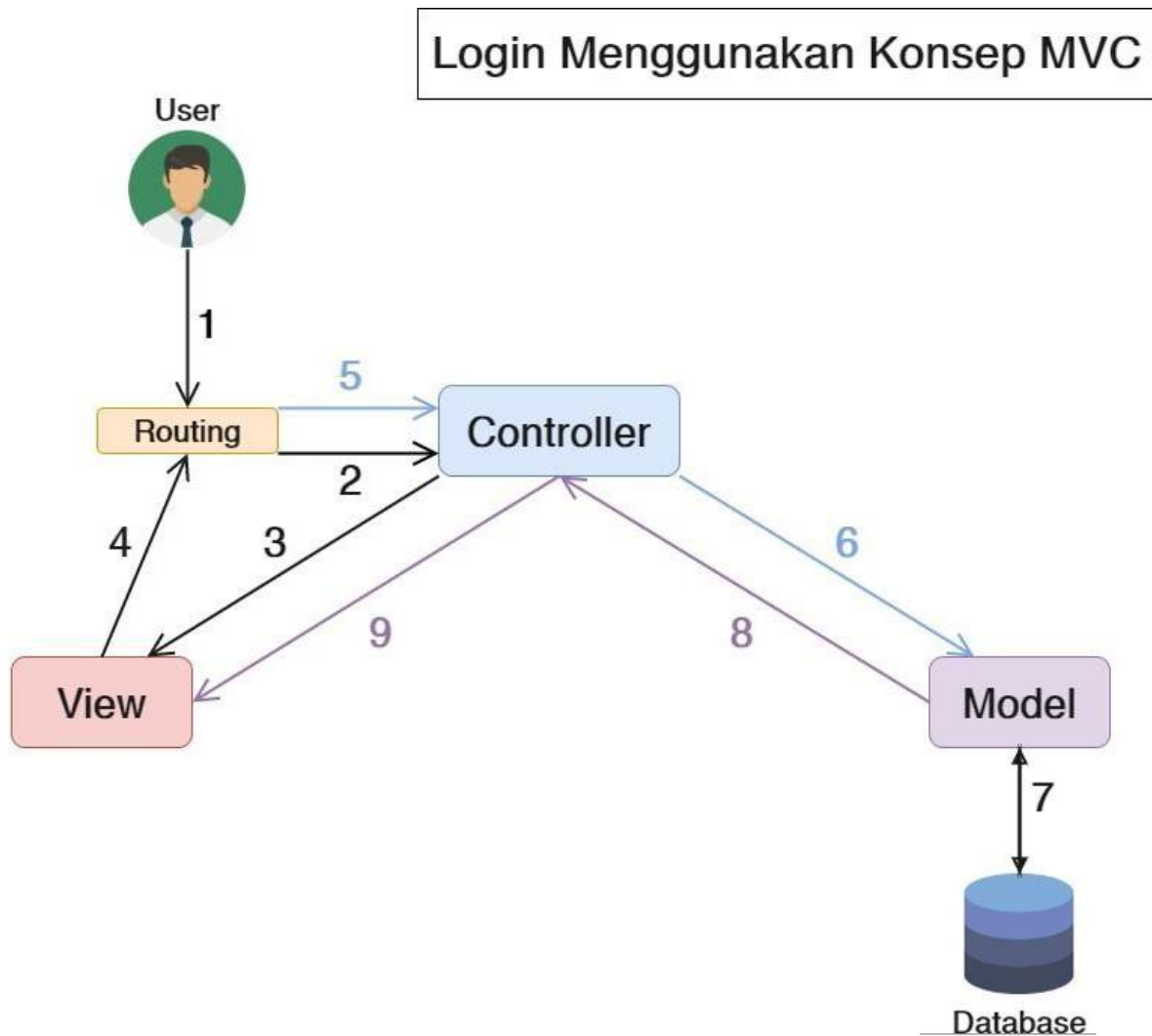
Codeigniter adalah salah satu framework php, saat ini banyak sekali framework php yang bisa digunakan, salah satunya adalah Codeigniter.

Kelebihan Codeigniter :

- Ukuran dari framework cukup kecil hanya sekitar 2.6 MB, itupun didalamnya sudah terdapat dokumentasi dari penggunaan framework.
  - Codeigniter sudah mendukung database –database yang paling umum digunakan seperti MySQL
  - Codeigniter menjadi salah satu framework php yang paling cepat. Karena tidak menggunakan template engine dan ORM yang dapat memperlambat proses.
  - Komunitas CI di Indonesia sangat banyak, sehingga tutorial penggunaan CI sangat mudah dicari
  - Bagi Pemula, CI sangat mudah dipelajari daripada framework lainnya, karena Codeigniter tidak bergantung pada tool tambahan seperti composer, Template, Engine, dll.
  - Sedikit konfigurasi yang dibutuhkan untuk dapat digunakan, Codeigniter tidak membutuhkan konfigurasi yang rumit, bahkan anda dapat menjalankan tanpa melakukan konfigurasi apapun.
- Kekurangan Codeigniter
- Codeigniter tidak ditujukan untuk pembuatan aplikasi web dengan skala besar, codeigniter ditujukan untuk aplikasi dengan skala kecil ataupun menengah.
  - Library yang disediakan cukup terbatas, karena anda sulit mencari plugin tambahan yang terverifikasi secara resmi, karena pada situs codeigniter tidak menyediakan plugin –plugin tambahan yang digunakan untuk mendukung pengembangan aplikasi yang dibuat dengan codeigniter
  - Codeigniter dikembangkan oleh Ellislab dan bukan oleh sebuah komunitas, sehingga update core engine tidak secepat framework lainnya.
  - Masih terdapat banyak kelonggaran pada saat coding, contohnya masih bebas menambahkan file –file tertentu.

## BAB 3 PEMBAHASAN

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



**Penjelasan :**

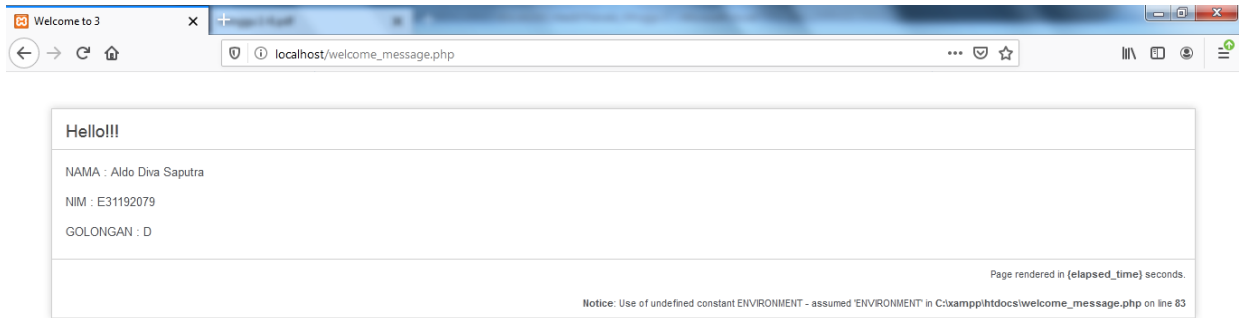
1. User melakukan request ke halaman login dan request akan dilakukan pada proses Routing.
2. Kemudian periksa perekrutannya, jika permintaan tersebut valid, teruskan permintaan tersebut ke controller.
3. Pada Controller akan menintruksikan untuk menampilkan View form login.
4. Ketika tampilan View dilihat dari login, pengguna memasukan nama pengguna dan kata sandi. Kemudian lanjutkan ke proses Routing.
5. Routing akan mengarahkan ke Controller yang sesuai.
6. Controller akan mengirimkan username dan password ke bagian Model.
7. Model melakukan pengambilan data untuk dilakukan pengecekan username dan password di Database.
8. Model mengirimkan hasil pengecekan username dan password ke Controller.
9. Controller akan memanggil View sesuai dengan hasil pemeriksaannya yaitu :
  - a. View form Home, jika hasil login True/Berhasil.
  - b. View form Login dan menampilkan pesan gagal, jika hasil login False/Gagal.

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



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <title>Welcome to CI 3</title>
6
7 <style type="text/css">
8
9 ::selection { background-color: #E13300; color: white; }
10 ::moz-selection { background-color: #E13300; color: white; }
11
12 body {
13     background-color: #fff;
14     margin: 40px;
15     font: 13px/20px normal Helvetica, Arial, sans-serif;
16     color: #4F5155;
17 }
18
19 a {
20     color: #003399;
21     background-color: transparent;
22     font-weight: normal;
23 }
24
25 h1 {
26     color: #444;
27     background-color: transparent;
28     border-bottom: 1px solid #D0D0D0;
29     font-size: 19px;
30     font-weight: normal;
31     margin: 0 0 14px 0;
32     padding: 14px 15px 10px 15px;
33 }
34
35 code {
36     font-family: Consolas, Monaco, Courier New, Courier, monospace;
37     font-size: 12px;
38     background-color: #f9f9f9;
39     border: 1px solid #D0D0D0;
40     color: #002166;
41     display: block;
42     margin: 14px 0 14px 0;
43     padding: 12px 10px 12px 10px;
44 }
45
46 #body {
47     margin: 0 15px 0 15px;
48 }
49
50 p.footer {
51     text-align: right;
52     font-size: 11px;
53     border-top: 1px solid #D0D0D0;
54     line-height: 32px;
55     padding: 0 10px 0 10px;
56     margin: 20px 0 0 0;
57 }
58
59 #container {
60     margin: 10px;
61     border: 1px solid #D0D0D0;
62     box-shadow: 0 0 8px #D0D0D0;
63 }
64
65 </style>
66 </head>
67 <body>
68 <div id="container">
69 <h1>Hello!!!</h1>
70
71 <div id="body">
72 <p>NAMA : Aldo Diva Saputra</p>
73
74 <p>NIM : E31192079</p>
75
76 <p>GOLONGAN : D</p>
77
78 </div>
79
80 </div>
81
82 <p class="footer">Page rendered in <strong>{elapsed_time}</strong> seconds. <?php echo (ENVIRONMENT === '
83 development') ? 'CodeIgniter Version <strong>'. CI_VERSION . '</strong>' : '' ?></p>
84
85 </div>
86 </body>
87 </html>
```

Output:

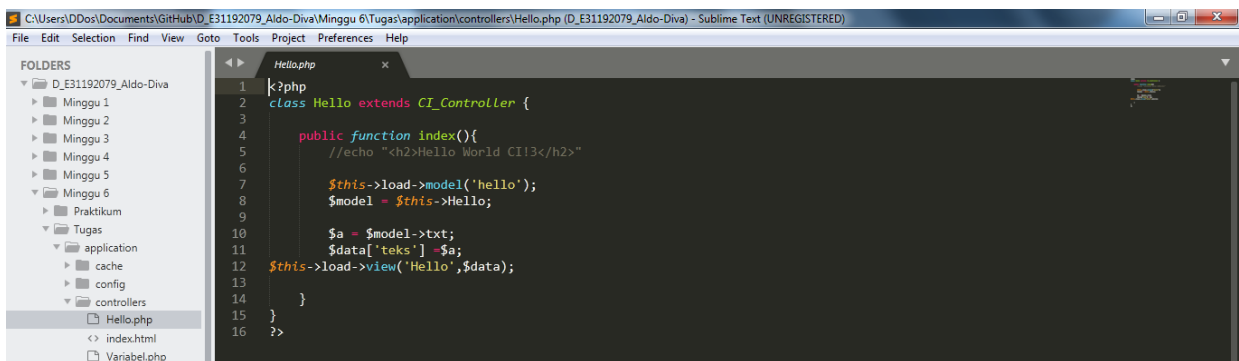


3.3 Buatlah sebuah halamanyang menampilkan pesan “Hello Wolrd dari CI Model” dimana pesan tersebut terdapat di dalam model dan dipanggil dari controller

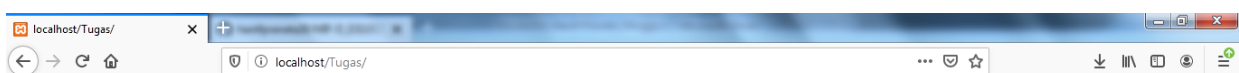
### 3.3.1 Code di dalam model



### 3.3.2 Code pada Controller



Output:



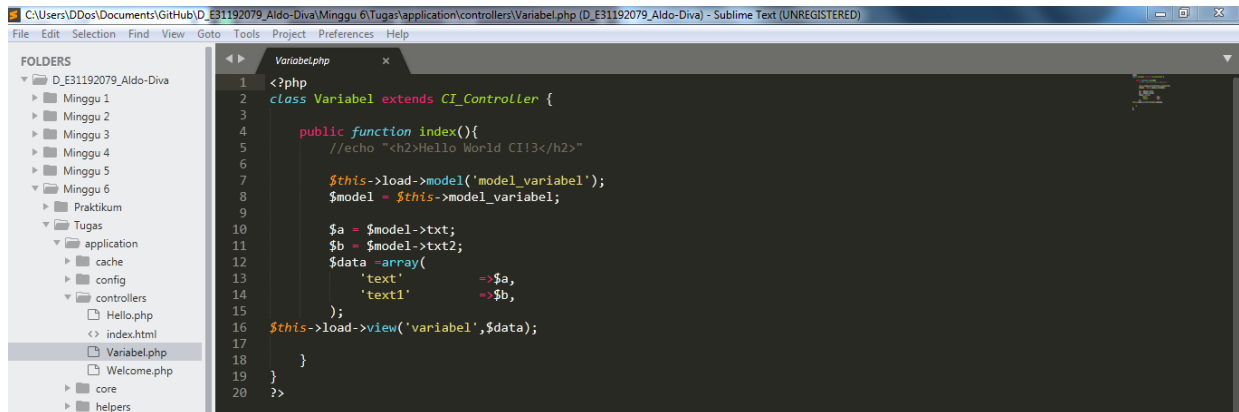
**Hello World!**

**Hello World From CI Model!**



3.4 Buatlah sebuah tampilan yang mengirim lebih dari 1 variabel menggunakan Controller, Model dan View.

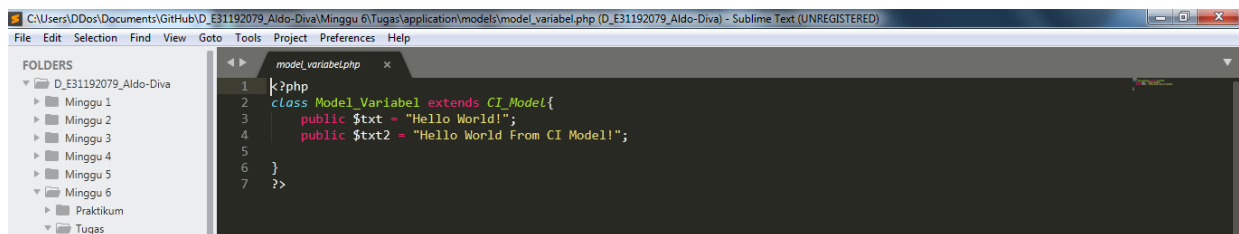
### 3.4.1 Variabel Controller



The screenshot shows the Sublime Text editor with the file 'Variabel.php' open. The file is located in the path 'C:\Users\DDos\Documents\GitHub\D\_E31192079\_Aldo-Diva\Minggu 6\Tugas\application\controllers\'. The code defines a class 'Variabel' extending 'CI\_Controller' with an 'index()' method. The method loads a model 'model\_variabel', assigns data to an array, and renders the 'variabel' view.

```
1 <?php
2 class Variabel extends CI_Controller {
3
4     public function index(){
5         //echo "<h2>Hello World CI!3</h2>"
6
7         $this->load->model('model_variabel');
8         $model = $this->model_variabel;
9
10        $a = $model->txt;
11        $b = $model->txt2;
12        $data = array(
13            'text' => $a,
14            'text1' => $b,
15        );
16        $this->load->view('variabel', $data);
17    }
18 }
19
20 ?>
```

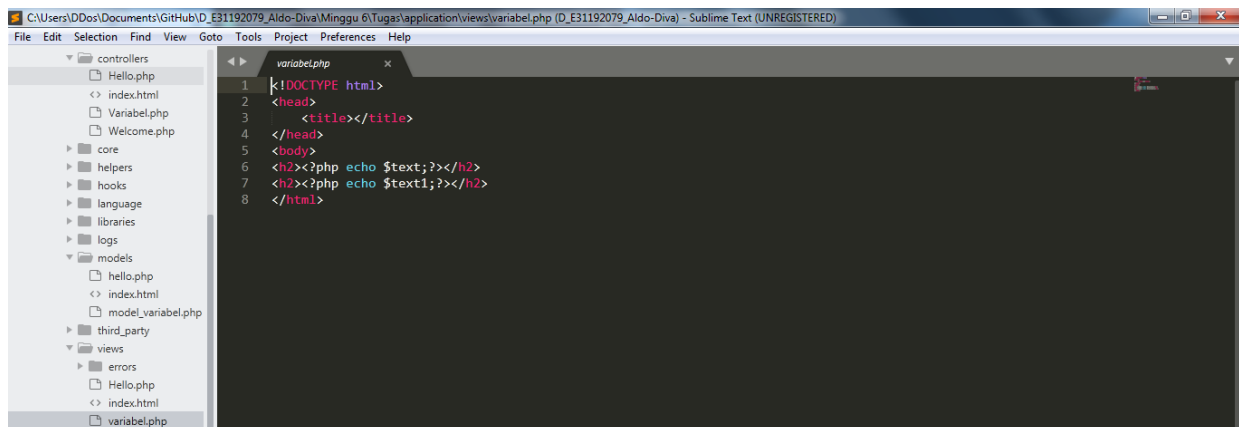
### 3.4.2 Model Variabel



The screenshot shows the Sublime Text editor with the file 'model\_variabel.php' open. The file is located in the path 'C:\Users\DDos\Documents\GitHub\D\_E31192079\_Aldo-Diva\Minggu 6\Tugas\application\models\'. The code defines a class 'Model\_Variabel' extending 'CI\_Model' with two public attributes: '\$txt' and '\$txt2'.

```
1 <?php
2 class Model_Variabel extends CI_Model{
3     public $txt = "Hello World!";
4     public $txt2 = "Hello World From CI Model!";
5
6 }
7 ?>
```

### 3.4.3 View Variabel



The screenshot shows the Sublime Text editor with the file 'variabel.php' open. The file is located in the path 'C:\Users\DDos\Documents\GitHub\D\_E31192079\_Aldo-Diva\Minggu 6\Tugas\application\views\'. The code is an HTML template that uses PHP to echo the values of '\$text' and '\$text1' into two separate h2 tags.

```
1 <!DOCTYPE html>
2 <html>
3     <head>
4         <title></title>
5     </head>
6     <body>
7         <h2><?php echo $text;?></h2>
8         <h2><?php echo $text1;?></h2>
9     </body>
10 </html>
```