

# **IMPLEMENTASI GENETIK ALGORITMA MENGUNAKAN CODEIGNITER**



---

# IMPLEMENTASI GENETIK ALGOR MENGUNAKAN CODEIGNITER Dalam 24 Jam

---

**Rolly M. Awangga**  
Informatics Research Center



**Kreatif Industri Nusantara**

***Penulis:***

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

***Editor:***

M. Yusril Helmi Setyawan

***Penyunting:***

Syafrial Fachrie Pane

Khaera Tunnisa

Diana Asri Wijayanti

***Desain sampul dan Tata letak:***

Deza Martha Akbar

***Penerbit:***

Kreatif Industri Nusantara

***Redaksi:***

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

***Distributor:***

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara  
apapun tanpa ijin tertulis dari penerbit

*‘Jika Kamu tidak dapat  
menahan lelahnya  
belajar, Maka kamu harus  
sanggup menahan  
perihnya Kebodohan.’  
Imam Syafi’i*

# CONTRIBUTORS

---

ROLLY MAULANA AWANGGA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia



# CONTENTS IN BRIEF

---

<b>1</b>	<b>XAMPP</b>	<b>1</b>
<b>2</b>	<b>CODEIGNITER</b>	<b>9</b>
<b>3</b>	<b>ALGORITMA GENETIKA</b>	<b>19</b>





# DAFTAR ISI

---

Daftar Gambar	xi
Daftar Tabel	xiii
Foreword	xvii
Kata Pengantar	xix
Acknowledgments	xxi
Acronyms	xxiii
Glossary	xxv
List of Symbols	xxvii
Introduction	xxix
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
<b>1 XAMPP</b>	<b>1</b>
1.1 Tutorial Install Xampp	1
1.2 Mengatasi Error Pada Xampp	4
<b>2 CODEIGNITER</b>	<b>9</b>
	<b>ix</b>

2.1	Tutorial Install CodeIgniter 3	9
2.2	Struktur CodeIgniter	10
2.3	Konfigurasi CodeIgniter 3	11
2.4	Konfigurasi Bootstrap dan Template CodeIgniter 3	13
2.5	MVC dan CRUD CodeIgniter 3	14
<b>3</b>	<b>ALGORITMA GENETIKA</b>	<b>19</b>
3.1	Implementasi Penjadwalan Algoritma Genetika di CodeIgniter	19
Daftar Pustaka		21

# DAFTAR GAMBAR

---

1.1	Control Panel Xampp	4
2.1	Tampilan Awal Template / Bootstrap CodeIgniter 3	14
2.2	Tampilan Hasil MVC dan CRUD	18



# DAFTAR TABEL

---



# Listings

---





# FOREWORD

---

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa



# KATA PENGANTAR

---

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat  
Februari, 2019*



# ACKNOWLEDGMENTS

---

Terima kasih atas semua masukan dari para mahasiswa agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk team IRC yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Intership.

R. M. A.



# ACRONYMS

---

ACGIH	American Conference of Governmental Industrial Hygienists
AEC	Atomic Energy Commission
OSHA	Occupational Health and Safety Commission
SAMA	Scientific Apparatus Makers Association





# GLOSSARY

---

git	Merupakan manajemen sumber kode yang dibuat oleh linus torvald.
bash	Merupakan bahasa sistem operasi berbasiskan *NIX.
linux	Sistem operasi berbasis sumber kode terbuka yang dibuat oleh Linus Torvald



# SYMBOLS

---

- $A$  Amplitude
- $\&$  Propositional logic symbol
- $a$  Filter Coefficient
  
- $\mathcal{B}$  Number of Beats



# INTRODUCTION

---

ROLLY MAULANA AWANGGA, S.T., M.T.

Informatics Research Center  
Bandung, Jawa Barat, Indonesia

Pada era disruptif saat ini. git merupakan sebuah kebutuhan dalam sebuah organisasi pengembangan perangkat lunak. Buku ini diharapkan bisa menjadi penghantar para programmer, analis, IT Operation dan Project Manajer. Dalam melakukan implementasi git pada diri dan organisasinya.

Rumusnya cuman sebagai contoh aja biar keren[1].

$$ABCDEF\alpha\beta\Gamma\Delta\sum_{def}^{abc} \tag{I.1}$$



# BAB 1

---

## XAMPP

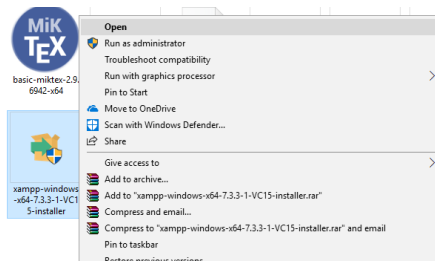
---

Xampp merupakan singkatan dari X (empat sistem operasi apapun), Apache, Mysql, PHP, dan Perl. Xampp adalah tool yang menyediakan paket perangkat lunak dalam satu buah paket. Dalam paket Xampp sudah terdapat Apache (web server), Mysql (database), PHP (server side scripting), Perl ,FTP server, PhpMyAdmin dan berbagai pustaka bantu lainnya. XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program [2].

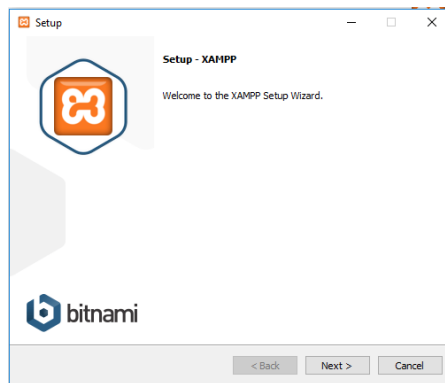
### 1.1 Tutorial Install Xampp

1. Download terlebih dahulu aplikasi Xampp di <https://www.apachefriends.org/index.html>, download sesuai sistem operasi yang anda gunakan, pada tutorial kali ini saya akan melakukan instalasi XAMPP di Windows 10.
2. Setelah download aplikasi, lakukan instalasi XAMPP, dengan cara klik kanan pada file instalasi kemudian pilih Open.

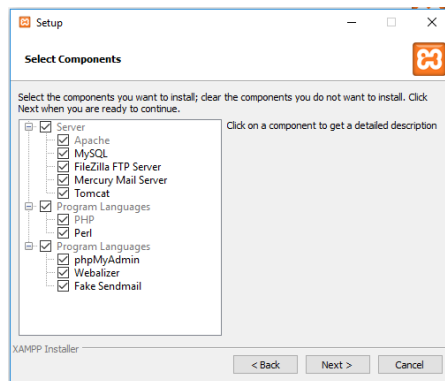




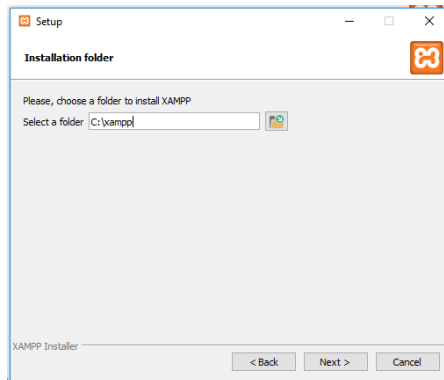
3. Jika pada saat melakukan instalasi muncul peringatan yang bertujuan untuk memastikan apakah Anda akan menginstal aplikasi ini, Silakan klik Ok/Yes untuk melanjutkan instalasi.



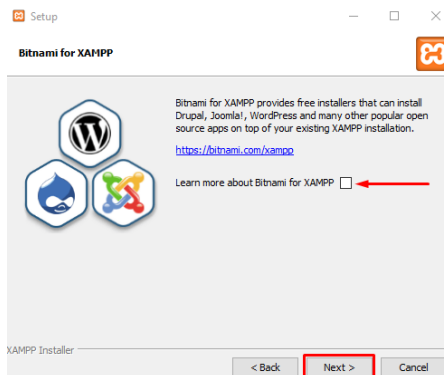
4. Klik next untuk melanjutkan, kemudian akan tampil pilihan aplikasi apa yang akan Anda install dan tidak ingin Anda install.



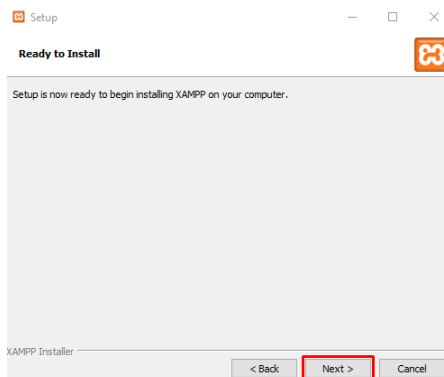
5. Tahap selanjutnya adalah memilih folder dimana lokasi instalasi xampp akan disimpan.



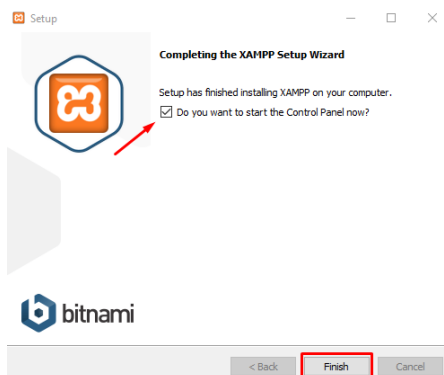
6. Silakan hilangkan centang pada “Learn more about Bitnami for XAMPP”, kemudian klik Next.



7. Klik next untuk malanjutkan ke proses instalasi xampp.

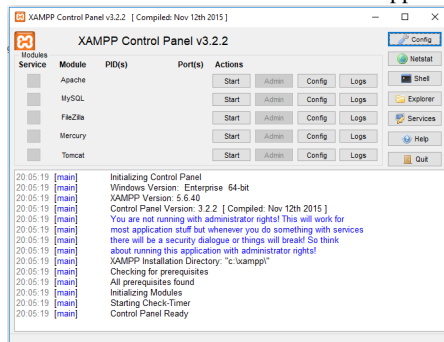


8. Apabila aplikasi sudah terinstal maka akan tampil pertanyaan mengenai apakah Anda ingin langsung menjalankan control panel. Pastikan pilihan tersebut sudah tercentang, kemudian klik tombol Finish.



9. Control panel akan muncul otomatis, tapi jika Anda tidak mencentang pilihan di halaman sebelumnya, maka Anda perlu membuka langsung control panel melalui start menu atau folder XAMPP di komputer Anda.
10. Apabila control panel sudah muncul dan terlihat seperti gambar 1.1, maka proses instalasi Xampp berhasil.

**Gambar 1.1** Control Panel Xampp



## 1.2 Mengatasi Error Pada Xampp

Hal yang menjadi penyebab utama kenapa tampil error pada XAMPP biasanya disebabkan aplikasi lain pada komputer Anda menggunakan port 80 atau 443, dimana port tersebut digunakan oleh Apache dan MySQL. Berikut cara mengatasi error pada xampp:

1. Klik Start, kemudian ketikkan “services.msc” pilih Services yang ada di bagian Best match.
2. Scrol ke bawah, pada bagian World Wide Web Publishing Service klik kanan dan pilih Stop.
3. Silakan close XAMPP, kemudian buka kembali dan jalankan Apache dan MySQL pada XAMPP.

Jika langkah yang Anda lakukan tidak berhasil mengatasi masalah yang dihadapi atau tidak menemukan World Wide Web Publishing, silakan lakukan langkah di bawah ini:

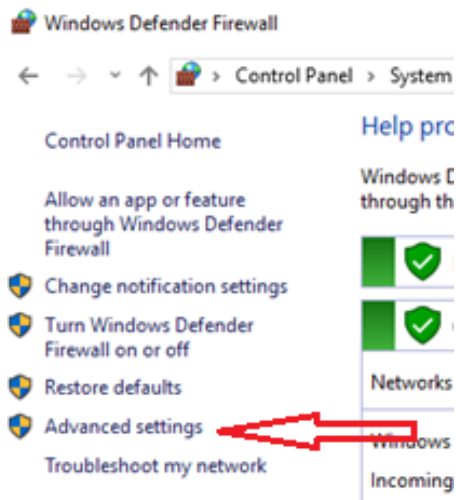
1. Buka control panel melalui tombol start yang ada pada pojok kiri bawah
2. Kemudian pilih system and security



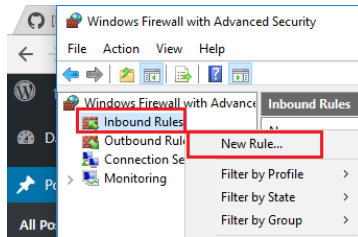
3. Pilih windows defender firewall



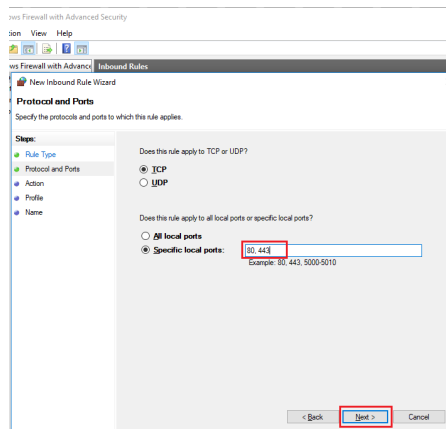
4. Pilih advanced settings



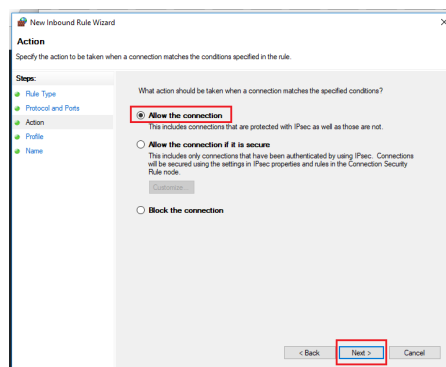
5. Klik Inbound dan klik kanan kemudian pilih New Rule, dapat dilihat seperti pada gambar dibawah



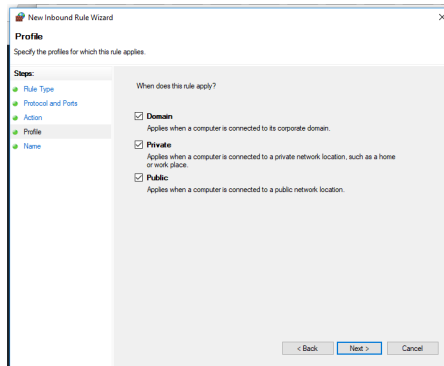
6. Pilih Port dan tekan tombol Next, kemudian pada kolom Specific Ports isi dengan 80, 443 kemudian klik Next.



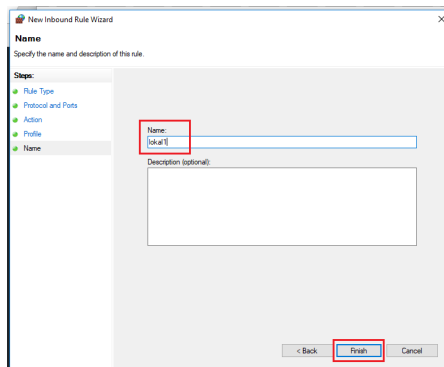
7. Centang Allow the Connection kemudian klik Next



8. Pastikan semua pilihan dicentang seperti pada gambar dibawah, kemudian klik Next



9. Masukkan lokal1 pada kolom name, kemudian klik Finish



10. Ulangi kembali langkah 1 sampai 6, untuk langkah 6 isi dengan lokal2, kemudian klik Finish
11. Restart komputer Anda



## BAB 2

---

# CODEIGNITER

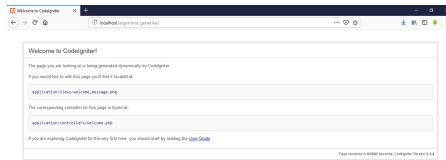
---

Codeigniter adalah sebuah framework untuk web yang dibuat dalam format PHP. Format yang dibuat ini selanjutnya dapat digunakan untuk membuat sistem aplikasi web yang kompleks. Codeigniter dapat mempercepat proses pembuatan web, karena semua class dan modul yang dibutuhkan sudah ada dan programmer hanya tinggal menggunakannya kembali pada aplikasi web yang akan dibuat [3].

### 2.1 Tutorial Install CodeIgniter 3

1. Pertama download Framework CodeIgniter di <https://www.codeigniter.com/>
2. Setelah mengunduh file CodeIgniter 3, ekstrak file tersebut menggunakan WinRAR atau 7Zip kedalam folder htdocs jika kamu menggunakan XAMPP atau `/var/www/html`. jika kamu menggunakan Apache2 Standalone, setelah itu ubahlah nama foldernya menjadi namaaplikasi.
3. Sekarang silahkan Kamu coba akses URL <http://localhost/namaaplikasi/> melalui browser Kamu, akan langsung ditampilkan halaman awal Codeigniter yang berarti Instalasi telah berhasil.





## 2.2 Struktur CodeIgniter

1. Folder **Application**, merupakan folder yang pada dasarnya menyimpan aplikasi yang sedang kita buat
2. Folder **Cache**, merupakan folder yang menyimpan semua cache yang dibuat oleh cache library
3. Folder **Config**, merupakan folder yang menyimpan informasi mengenai konfigurasi aplikasi seperti autoload, database, routes dan lainnya.
4. Folder **Controller**, merupakan folder menyimpan controller - controller aplikasi yang dapat digunakan untuk menyusun aktivitas program .
5. Folder **Core**, adalah folder untuk memperluas class class inti codeigniter.
6. Folder **Helpers**, merupakan folder untuk menyimpan helpers.
7. Folder **Hooks**, merupakan folder untuk menyimpan hooks untuk mengubah alur fungsi dari core Codeigniter
8. Folder **Language**, merupakan folder untuk menyimpan bahasa - bahasa yang akan digunakan.
9. Folder **Libraries**, merupakan folder untuk menyimpan library.
10. Folder **Logs**, merupakan folder untuk menyimpan semua error log apabila error log diaktifkan.
11. Folder **Models**, merupakan folder untuk menyimpan models yang akan mendefinisikan tabel dari database yang dapat kita gunakan oleh Controller yang kita buat untuk mengakses database.
12. Folder `third_party`, merupakan folder untuk menyimpan fungsi fungsi tambahan dalam cara kerja codeigniter.
13. Folder **Views**, merupakan folder untuk menyimpan tampilan dari aplikasi yang kita buat.
14. Folder **System**, merupakan folder untuk menyimpan sistem inti dari Codeigniter.

## 2.3 Konfigurasi CodeIgniter 3

Di dalam folder `application/config/` terdapat berbagai macam file konfigurasi yang dapat kita atur sendiri nantinya.

1. **autoload.php**, digunakan untuk menambahkan package, libraries, drivers, helper, atau custom config lainnya agar secara otomatis diload oleh codeigniter.
2. **config.php**, digunakan untuk membuat pengaturan dasar untuk web app codeigniter anda, seperti `base_url`, index page, cookie, proxy dan lain lain.
3. **constants.php**, digunakan untuk kita dapat membuat constant baru.
4. **database.php**, digunakan untuk mengatur koneksi web app kita ke database.
5. **doctypes.php**, sebagai tempat penyimpanan deklarasi dokumen Doctype.
6. `foreign_chars.php`, sebagai tempat penyimpanan karakter karakter asing.
7. **hooks.php**, digunakan untuk mendefine "hooks" untuk meng extends CI
8. **memcached.php**, config yang memungkinkan kita memcache database, driver dan lain lain sehingga lebih efektif.
9. **migration.php**, config yang memungkinkan kita melakukan database migration. Secara default dijadikan False.
10. **mimes.php**, menyimpan array yang berisi tipe file untuk fungsi upload.
11. **profiler.php**, digunakan untuk mengatur profiler yang berguna pada saat debugging.
12. **routes.php**, digunakan untuk mengatur default controller dan override 404
13. **smileys.php**, menyimpan array yang berisi smiley yang membantu helper emoticon.
14. `user_agents.php`, menyimpan data user agent, yang membantu class User Agen untuk mengidentifikasi browser, platform, robot dan data mobile device

Pada konfigurasi yang saya lakukan hanya melakukan konfigurasi pada file `autoload.php`, `config.php`, `database.php` dan `routes.php`. Berikut cara konfigurasinya:

### 1. Autoload.php



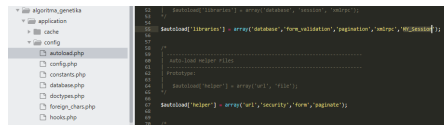
```

1  * @author Albert R. Sanjurjo
2
3  * See the documentation for details:
4  * http://codeigniter.com/user_guide/general/autoloader.html
5
6  * To use the autoloader, you must set the following configuration
7  * items in your config/autoload.php file:
8  * - $autoload: An array of class names to load.
9  * - $config: An array of configuration options.
10
11  * If this is not set then CodeIgniter will guess the protocol, domain and
12  * path to your installation.
13
14  * The following array lists the class names to load. You can add your own
15  * classes here. Note: Only class names with trailing slashes are
16  * supported.
17
18  * The following array lists the configuration options. You can add your own
19  * configuration options here.

```

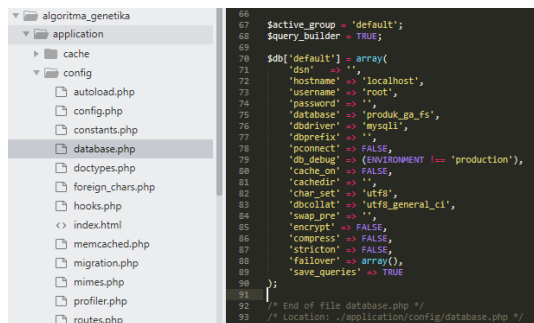
Pada file ini saya meng-input libraries untuk support framework CodeIgniter ini terhadap database, form\_validation yang akan dibuat nantinya, pagination dan Session untuk mengaktifkan session pada CodeIgniter. Pada variable autoload helper saya meng-input url dan form semua di inputkan sesuai dengan kebutuhan pembuat aplikasi. Array tersebut akan di eksekusi secara otomatis oleh CodeIgniter.

## 2. Config.php



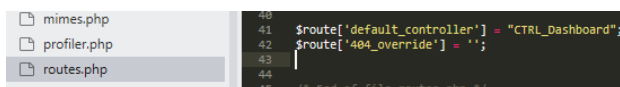
Pada config.php inputkan url utama aplikasi pada variable config `base_url` seperti pada gambar diatas.

## 3. Database.php



Pada database.php konfigurasi yang dilakukan untuk mengkoneksikan database yaitu MySQL dengan aplikasi web berbasis framework CodeIgniter. Dapat dilihat pada line 72, hostname yang diinputkan sesuai dengan hostname yang dipakai, disini saya menginputkan localhost dengan username default yaitu root password dikosongkan karena pada Xampp saya tidak menggunakan password. Pada database line 75 inputkan nama database sesuai dengan nama database yang ada pada MySQL.

## 4. Routes.php

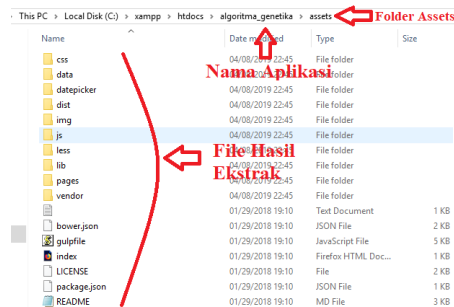


Pada file ini dilakukan konfigurasi dimana controller mana yang akan pertama di eksekusi ketika url dijalankan.

## 2.4 Konfigurasi Bootstrap dan Template CodeIgniter 3

Ada berbagai macam konfigurasi bootstrap dan template terhadap CodeIgniter, baik secara install maupun dengan cara konfigurasi sendiri. Pada tutorial kali ini saya ingin menerapkan bootstrap dan template di CodeIgniter dengan cara cepat. Untuk yang ingin menggunakan cara instan, bisa dengan cara mengunjungi website [w3layout.com](http://w3layout.com) dan website yang menyediakan assets template dan bootstrap siap pakai. Berikut cara konfigurasi template dan bootstrap pada CodeIgniter:

1. Siapkan template bootstrap yang sudah didownload
2. Ekstrak file tersebut jika dalam bentuk .rar atau .zip
3. Buat folder baru dengan nama assets terhadap aplikasi yang ingin di konfigurasi kemudian copy file hasil ekstrak tadi ke dalam folder tersebut.



4. Setelah menkopi file kedalam folder assets, langkah selanjutnya adalah memanggil config tersebut. Jangan lupa untuk membuat header dan footer ketika membuat website guna untuk mempermudah apabila terjadi perubahan terhadap beberapa menu.
5. Copy isi dari index.html yang ada dalam assets kemudian buat file di dalam application/views/namafilename.php dengan format .php dan pastekan dalam file tersebut.
6. Kemudian pisahkan antara header dan footer aplikasi anda.
7. Pertama lakukan konfigurasi terhadap header dengan cara memanggil link dan script yang sudah di copy di dalam assets, berikut contoh pemanggilannya:

```
<link href="php echo base_url()'/assets/vendor/bootstrap/css/bootstrap.min.css'"&gt;?&gt;"rel="stylesheet"&gt;
&lt;link href="<?php echo base_url()'/assets/vendor/datatables/jquery.dataTables.min.css'"&gt;?&gt;"rel="stylesheet"&gt;
&lt;link href="<?php echo base_url()'/assets/dist/css/bs-admin-2.css'"&gt;?&gt;"rel="stylesheet"&gt;
&lt;link href="<?php echo base_url()'/assets/vendor/font-awesome/css/font-awesome.min.css'"&gt;?&gt;"rel="stylesheet"&gt;
&lt;script src="<?php echo base_url()'/assets/vendor/jquery/jquery.min.js'"&gt;?&gt;"rel="script"&gt;
&lt;/script&gt;
&lt;script src="<?php echo base_url()'/assets/vendor/datatables/jquery.dataTables.min.js'"&gt;?&gt;"rel="script"&gt;
&lt;script src="<?php echo base_url()'/assets/vendor/datatables-responsive/datatables-responsive.min.js'"&gt;?&gt;"rel="script"&gt;
&lt;script src="<?php echo base_url()'/assets/datepicker/datepicker.js'"&gt;?&gt;"rel="script"&gt;
&lt;script src="<?php echo base_url()'/assets/datepicker/datepicker.js'"&gt;?&gt;"rel="script"&gt;</pre

```

Lakukan pemanggilan terhadap semua code yang berbau href dan src dengan mengisikan kodingan seperti `echo base_url()/assets/linkygdituj`, terlihat

seperti gambar diatas. Lakukan hal yang sama terhadap footer.php. Setelah itu simpan.

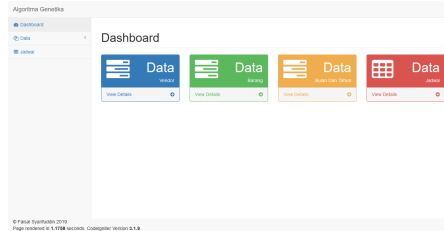
8. Selanjutnya membuat file untuk tampilan awal yaitu index.php atau dashboard.php
9. Edit beberapa codingan yang sudah dipastekan tadi dari assets/index.html ke application/views/index.php sesuai dengan tampilan yang diinginkan. Apabila memisahkan header dan footer jangan lupa untuk memanggil header footer tersebut dengan cara berikut:

```
1 <?php $this->load->view('page/header') ?>
2
3 <div id="page-wrapper">
4   <div class="row">
5     <div class="col-lg-12">
6       <h1 class="page-header">Dashboard</h1>
7     </div>
8   </div>
```

Load view berarti meload file yang ada di dalam folder view dalam artian codin-gan diatas akan meload file header yang ada di folder application/views/page/-header. Simpan semua konfigurasi dan coba jalankan.

10. Berikut tampilan yang saya modifikasi dari index.html menjadi dashboard.php

**Gambar 2.1** Tampilan Awal Template / Bootstrap CodeIgniter 3



## 2.5 MVC dan CRUD CodeIgniter 3

### 1. Konfigurasi Model

- (a) Buat table sesuai dengan kebutuhan aplikasi
- (b) Kemudian buat file baru dengan format .php pada application/models
- (c) Pastekan codingan dibawah ini

```
<?php
class MDL_Barang extends CI_Model{
function __construct(){
parent::__construct();
}
```

```
public function get_barang() {
    $hasil=$this->db->get('barang');
    if($hasil->num_rows() > 0){
        return $hasil->result();
    }else{
        return false;
    }
}

public function insert_barang($barang_data)
{
    $this->db->insert('barang', $barang_data);
}

public function find_barang($kode_barang)
{
    $hasil = $this->db->where('kode_barang', $kode_barang)->li
    if($hasil->num_rows() > 0){
        return $hasil->row();
    }else{
        return array();
    }
}

public function update_barang($kode_barang, $barang_data)
{
    $this->db->where('kode_barang', $kode_barang)
    ->update('barang', $barang_data);
}

public function delete_barang($kode_barang)
{
    $this->db->where('kode_barang', $kode_barang)
    ->delete('barang');
}

public function detail_barang($kode_barang)
{
    $hasil = $this->db->where('kode_barang', $kode_barang)->li
    if($hasil->num_rows() > 0){
        return $hasil->result();
    }else{
        return array();
    }
}
```

```

}
}

```

Penjelasan:

- i. function `get_barang` berfungsi untuk meload semua data yang ada di database barang
  - ii. function `insert_barang` berfungsi untuk melakukan execute insert data ke dalam table barang
  - iii. function `find_barang` berfungsi untuk mencari kode barang yang akan di edit
  - iv. function `update_barang` berfungsi untuk melakukan execute update terhadap table barang
  - v. function `delete_barang` berfungsi untuk melakukan execute delete data pada table barang
  - vi. function `detail_barang` berfungsi untuk melihat data lengkap barang sesuai dengan id yang dipanggil.
- Function-function di atas merupakan function dasar untuk melakukan proses CRUD di model CodeIgniter 3.

## 2. Konfigurasi Controller

- (a) Buat file baru dengan format `.php` di folder `application/controller`, kemudian buat beberapa function crud
- (b) Function pertama adalah function `index_barang`, untuk menampilkan seluruh data yang ada pada table barang

```

public function index_barang(){
    $data['barang'] = $this->MDL_Barang->get_barang();
    $this->load->view('web/barang/index_barang', $data);
}

```

- (c) Function kedua yaitu function `add_barang`, untuk melakukan proses input data dalam bentuk variable sesuai dengan field yang ada pada table barang yang kemudian variable tersebut akan di lempar ke model untuk di inputkan kedalam table

```

public function add_barang()
{
    $this->form_validation->set_rules('nama_barang','Nama barang','required');
    $this->form_validation->set_rules('kategori_barang','Kategori barang','required');
    $this->form_validation->set_rules('tingkat_kebutuhan','Tingkat kebutuhan','required');
    $this->form_validation->set_rules('jumlah_dalam_kategori','Jumlah dalam kategori','required');

    if($this->form_validation->run() == FALSE){
        $this->load->view('web/barang/form_add_barang');
    }else{
        $barang_data = array (
            'nama_barang' => $this->MDL_Barang->get_nama_barang(),
            'kategori_barang' => set_value('nama_barang'),
            'tingkat_kebutuhan' => set_value('kategori_barang'),
            'jumlah_dalam_kategori' => set_value('tingkat_kebutuhan'),
            'jumlah_dalam_kategori' => set_value('jumlah_dalam_kategori')
        );
        $this->MDL_Barang->insert_barang($barang_data);
        $this->session->set_flashdata('notif', 'Data berhasil Di Simpan');
        redirect('web/index_barang');
    }
}

```

- (d) Selanjutnya function `edit_barang`, untuk melakukan proses edit data barang yang kemudian variable yang di tampung akan di execute pada model function `update_barang`





```

<?php
$kode_barang = $barang->kode_barang;
if($this->input->post('is_submitted')){
    $nama_barang = set_value('nama_barang');
    $kategori_barang = set_value('kategori_barang');
    $tingkat_kebutuhan = set_value('tingkat_kebutuhan');
    $jumlah_dalam_kategori = set_value('jumlah_dalam_kategori');
}else{
    $nama_barang = $barang->nama_barang;
    $kategori_barang = $barang->kategori_barang;
    $tingkat_kebutuhan = $barang->tingkat_kebutuhan;
    $jumlah_dalam_kategori = $barang->jumlah_dalam_kategori;
}
}
?>

```

Setelah memanggil dibawah codingan tersebut buat form untuk view edit data barang

```

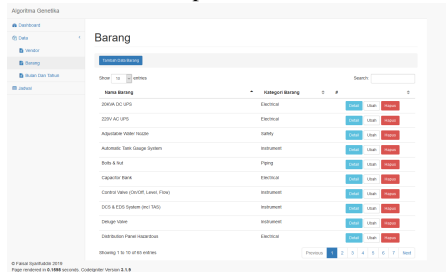
<div class="row">
<div class="col-lg-11">
<div form_open_multipart('web/edit_barang/' . $kode_barang, ['class' => 'form-horizontal'])>
<div class="form-group">
<div class="col-sm-12">
<div class="form-control" name="nama_barang" value="{$nama_barang}">
</div>
</div>
</div>
<div echo $error; >

```

Pastikan value terisi agar saat melakukan edit data yang diedit dapat terlihat dan pastikan pada name sesuai dengan field yang ada pada table dan function `edit_barang`

- (d) Simpan semua file tersebut dan jalankan aplikasi. Berikut tampilan CRUD dari codingan yang di buat

**Gambar 2.2** Tampilan Hasil MVC dan CRUD



## BAB 3

---

# ALGORITMA GENETIKA

---

Algoritma Genetika (GA) merupakan salah satu metode heuristic yang merupakan cabang dari evolutionary algorithm, yaitu suatu teknik untuk memecahkan masalah-masalah optimasi yang rumit dengan menirukan proses evolusi makhluk hidup. GA terbukti sesuai digunakan untuk menyelesaikan masalah multi objektif. GA berkembang seiring dengan perkembangan teknologi informasi yang sangat pesat [4], [5].

Algoritma ini banyak digunakan dalam bidang fisika, biologi, ekonomi, sosiologi dan lain-lain yang sering menghadapi masalah optimasi dengan model matematika yang kompleks atau bahkan sulit dibangun [4].

### 3.1 Implementasi Penjadwalan Algoritma Genetika di CodeIgniter



# DAFTAR PUSTAKA

---

1. R. Awangga, “Sampeu: Servicing web map tile service over web map service to increase computation performance,” in *IOP Conference Series: Earth and Environmental Science*, vol. 145, no. 1. IOP Publishing, 2018, p. 012057.
2. M. I. Sugiarto, A. Linarta, and A. Sofiyan, “Aplikasi layanan informasi absen dan nilai berbasis sms gateway menggunakan php pada smk taruna persada dumai,” *INFORMATIKA*, vol. 9, no. 2, pp. 60–69, 2019.
3. D. Prabowo, “Website e-commerce menggunakan model view controller (mvc) dengan framework codeigniter studi kasus: Toko miniatur,” *Data Manajemen dan Teknologi Informasi (DASI)*, vol. 16, no. 1, p. 23, 2015.
4. J. Liu, K. Wang, and B. C. Fung, “Mining high utility patterns in one phase without generating candidates,” *IEEE Transactions on knowledge and data engineering*, vol. 28, no. 5, pp. 1245–1257, 2016.
5. H. Wei and X.-S. Tang, “A genetic-algorithm-based explicit description of object contour and its ability to facilitate recognition,” *IEEE transactions on cybernetics*, vol. 45, no. 11, pp. 2558–2571, 2015.

