



# Pemrograman Web dan Perangkat Bergerak

Modul Praktikum CodeIgniter 3 – Project Aplikasi Pengelolaan Laundry

# DAFTAR ISI

|  |    |
|--|----|
| DAFTAR ISI.....                              | i  |
| BAB I PENDAHULUAN .....                      | 1  |
| I. Framework .....                           | 1  |
| II. Pengenalan Codeigniter .....             | 1  |
| III. Keunggulan Codeigniter .....            | 1  |
| IV. Model-View-Controller (MVC) .....        | 2  |
| A. Model .....                               | 2  |
| B. View .....                                | 3  |
| C. Controller .....                          | 3  |
| V. Instalasi Codeigniter.....                | 3  |
| VI. Konfigurasi dasar Codeigniter .....      | 4  |
| 1) Autoload.php .....                        | 5  |
| 2) Config.php.....                           | 6  |
| 3) Database.php .....                        | 6  |
| VII. Hello World Codeigniter .....           | 7  |
| VIII. Menghilangkan index.php pada URL ..... | 9  |
| IX. Controller dan View .....                | 10 |
| X. Codeigniter dan Bootstrap.....            | 12 |
| XI. Study Kasus .....                        | 14 |
| BABII Membuat Tabel .....                    | 16 |
| I. Tujuan .....                              | 16 |
| II. Teori .....                              | 16 |
| III. Praktikum .....                         | 16 |
| Membuat Database/Basis Data .....            | 18 |
| Membuat Tabel.....                           | 19 |
| Membuat Relasi Antar Tabel .....             | 20 |
| BAB III Membuat Fitur Login .....            | 23 |
| I. Tujuan .....                              | 23 |
| II. Teori .....                              | 23 |
| III. Praktikum .....                         | 23 |
| I. Persiapan.....                            | 23 |
| II. Proses Coding.....                       | 26 |

|  |    |
|--|----|
| III.    Pengujian.....                               | 31 |
| BAB III Create Read Update Delete.....               | 32 |
| I.    Tujuan .....                                   | 32 |
| II.   Teori .....                                    | 32 |
| III.   Praktikum .....                               | 32 |
| IV.   Pengujian .....                                | 44 |
| BAB IV Membuat Report/Laporan .....                  | 45 |
| I.    Tujuan .....                                   | 45 |
| II.   Teori .....                                    | 45 |
| III.   Praktikum.....                                | 45 |
| Download Dompdf .....                                | 45 |
| Extract Dompdf di folder libraries CodeIgniter ..... | 45 |
| Membuat Library Pdfgenerator.php.....                | 48 |
| Membuat function pdfview .....                       | 49 |
| Membuat Views laporan member .....                   | 49 |
| V.    Pengujian .....                                | 51 |

# BAB I

## PENDAHULUAN

### I. Framework

Framework adalah kumpulan intruksi-intruksi yang di kumpulkan dalam class dan function-function dengan fungsi masing-masing untuk memudahkan developer dalam pemanggilannya tanpa harus menuliskan syntax program yang sama berulang-ulang. hal ini memiliki kegunaan untuk menghemat waktu dan mencegah penulisan syntax secara berulang-ulang agar tercipta nya source code yang bersih dan terstruktur.

### II. Pengenalan Codeigniter

Codeigniter merupakan suatu Web Application Framework (WAF) yang dirancang khusus untuk mempermudah developer web dalam mengembangkan aplikasi berbasis web. Codeigniter berisi kumpulan kode berupa pustaka (library) dan alat (tools) yang dipadukan sedemikian rupa menjadi suatu kerangka kerja (framework). Codeigniter adalah framework web untuk bahasa pemrograman PHP yang rancang oleh Rick Ellis pada tahun 2006, penemu dan pendiri EllisLab ([www.ellislab.com](http://www.ellislab.com)).

Codeigniter menganut pola desain atau arsitektur Model-View-Controller (MVC) yang memisahkan bagian kode untuk penanganan proses bisnis dengan bagian kode untuk keperluan presentasi (tampilan). Dengan menggunakan pola desain ini, memungkinkan para pengembangan web untuk mengerjakan aplikasi berbasis web secara bersama (teamwork). Dengan begitu para pengembang web lebih berfokus pada bagiannya masing-masing tanpa mengganggu bagian yang lain. Sehingga aplikasi yang dibangun akan selesai lebih cepat.

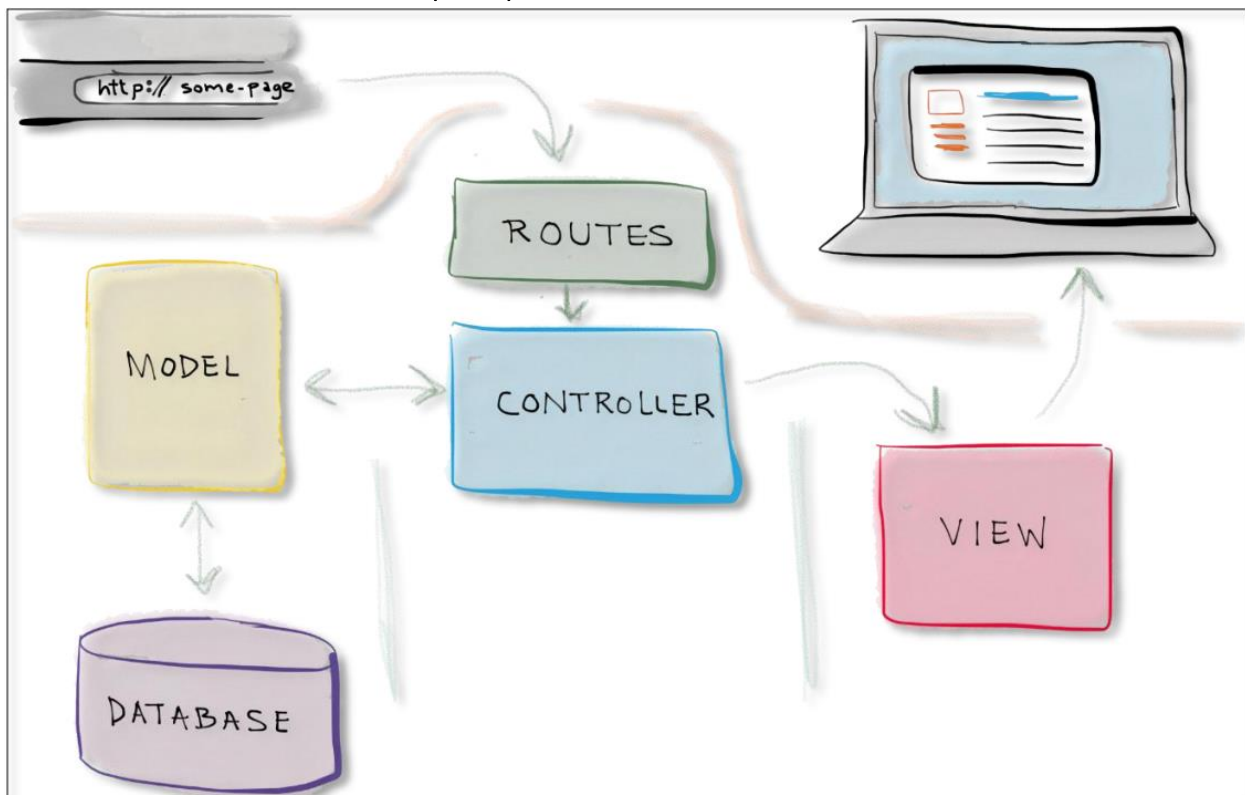
### III. Keunggulan Codeigniter

Codeigniter merupakan sebuah toolkit yang ditujukan untuk Anda yang ingin membangun aplikasi berbasis web dalam bahasa pemrograman PHP. Adapun beberapa keunggulan yang ditawarkan oleh codeigniter adalah sebagai berikut:

1. Codeigniter adalah framework PHP yang bersifat open-source.
2. Codeigniter memiliki ukuran yang kecil dibandingkan dengan framework lain. Setelah proses instalasi, framework Codeigniter hanya berukuran kurang lebih 2 MB (tanpa dokumentasi atau jika user\_guide dihapus).
3. Aplikasi yang dibuat menggunakan codeigniter bisa bejalan cepat.
4. Codeigniter menggunakan pola desain Model-View-Controller (MVC) sehingga satu file tidak terlalu berisi banyak kode. Hal ini menjadikan kode lebih mudah dibaca, dipahami, dan dipelihara di kemudian hari.
5. Codeigniter dapat diperluas sesuai dengan kebutuhan.
6. Codeigniter terdokumentasi dengan baik informasi tentang pustaka (Library) dan fungsi yang disediakan oleh codeigniter dapat diperoleh melalui dokumentasi yang disertakan di dalam paket distribusinya.
7. Codeigniter memiliki library dan helper yang lengkap.
8. Codeigniter memiliki security yang handal seperti xss filtering, session encryption, dan lain-lain.

9. Codeigniter mengizinkan pengembang web menggunakan library atau helper yang tidak disediakan oleh codeigniter seperti: Google Map API, Facebook API, fpdf, dan lain sebagainya.
10. Codeigniter bersifat tidak kaku. sehingga memberikan kebebasan kepada developer web untuk mengembangkan aplikasi berbasis web bahkan tanpa framework.
11. Codeigniter memiliki komunitas yang besar dan tersebar di seluruh dunia, sehingga memudahkan para pengembang web untuk memecahkan permasalahan (problem solving) yang dihadapi para pengembang web di saat mengembangkan aplikasi berbasis web.
12. Codeigniter mendukung banyak RDBMS (Relational Database Management System) seperti MySQL, MySQLi, SQL Server, Oracle, Maria DB, PostgreSQL, SQLite, dan lain sebagainya.
13. Codeigniter pada dasarnya menganut Clean URL dan mendukung SEO (Search Engine Optimazation).

#### IV. Model-View-Controller (MVC)



Gambar 1 Skema MVC

MVC adalah sebuah pendekatan yang ditempuh untuk memisahkan aplikasi menjadi tiga bagian, yaitu Model, View, dan Controller. MVC memberikan struktur kepada aplikasi, sehingga dapat di capai “code reusability”. Berikut penjabaran dari komponen-komponen MVC:

##### A. Model

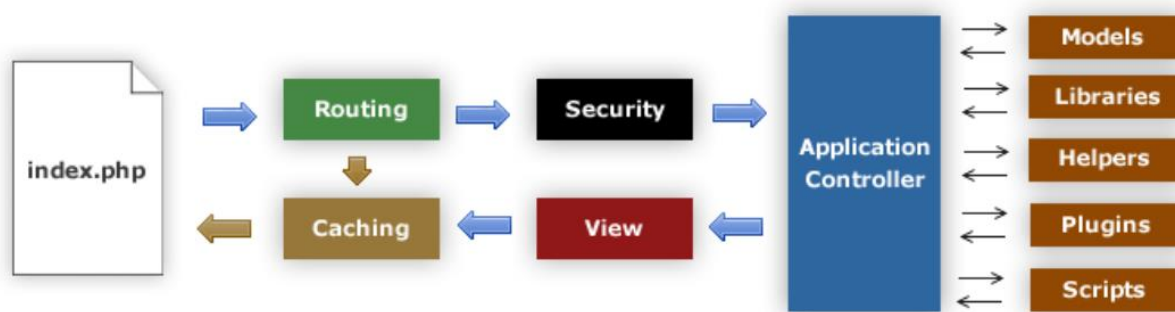
Model merepresentasikan data yang digunakan aplikasi, seperti database, RSS, atau data yang diperoleh dari pemanggilan API, dan aksi yang melibatkan operasi Create, Read, Update, dan Delete (CRUD) data.

### B. View

View adalah informasi yang ditampilkan kepada user melalui browser. Biasanya berupa file HTML atau kode PHP yang menyusun template untuk sebuah website. Pada codeigniter, view dapat berupa bagian-bagian sebuah halaman, template, atau jenis lain dari halaman atau template

### C. Controller

Controller adalah “business logic” yang bertugas sebagai jembatan antara model dan view. Controller akan merespon HTTP request yang datang dari user (melalui browser), dari request ini controller akan menentukan apa yang harus dilakukan. Didalam codeigniter, secara detail MVC digambarkan sebagai berikut:



Gambar 2 Detail MVC

Pada gambar diatas, file “index.php” berperan sebagai controller utama yang memanggil fungsi-fungsi dasar yang digunakan untuk menjalankan controller. Router memeriksa HTTP request kemudian memutuskan Controller mana yang akan digunakan untuk menangani request tersebut. Apabila file cache tersedia, alur aplikasi akan dilewati dan file cache tersebut yang akan dikirimkan ke browser pengguna. Sebelum controller dipanggil, HTTP request dan data yang dikirimkan pengguna akan di sortir terlebih dahulu untuk alasan keamanan. Controller memanggil file Model, Library, Helper, dan file lain yang dibutuhkan untuk menangani HTTP request. Hasil akhirnya akan ditampilkan ole file View kemudian dikirimkan ke browser pengguna untuk ditampilkan. Jika modus caching diaktifkan, hasil view akan di-cache terlebih dahulu. Sehingga jika nanti ada request yang sama, bisa langsung digunakan.

## V. Instalasi Codeigniter

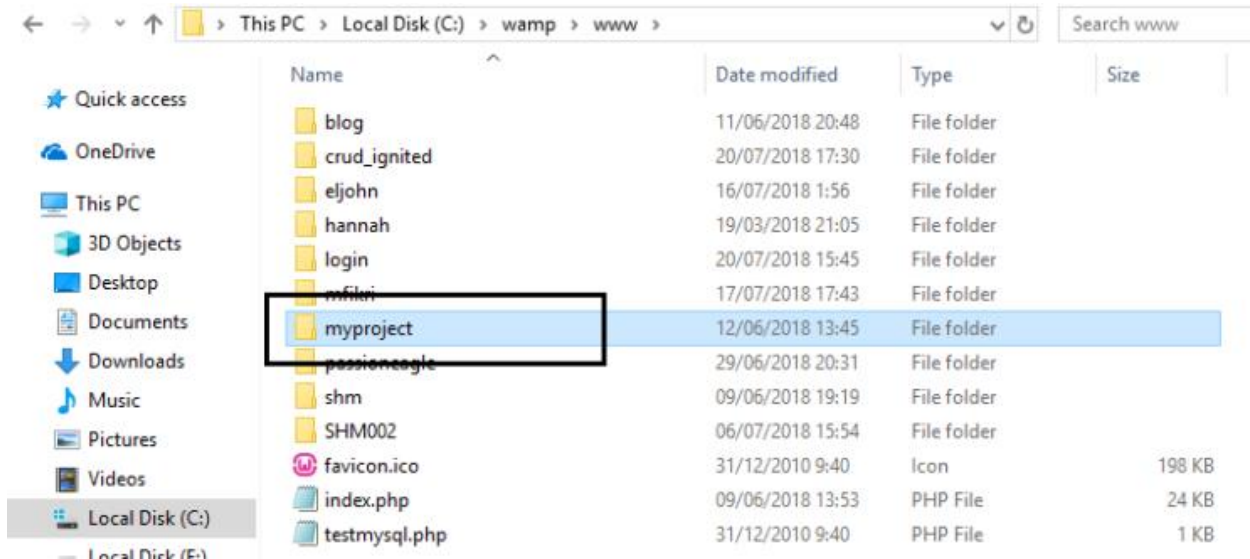
Codeigniter merupakan framework PHP yang cara installnya paling sederhana dibandingkan framework PHP lainnya. Anda hanya perlu mengextractnya ke web server Anda dan langsung terinstall. Mudah bukan? Jika Anda ingin menginstall codeigniter di server lokal atau localhost, ada beberapa software yang bisa Anda gunakan. Diantaranya yang paling populer yaitu WAMP SERVER, MAMP, atau XAMPP.

Untuk melakukan instalasi pada codeigniter ikuti langkah berikut:

1. Pastikan Web Server telah terinstall dan berjalan (running) di komputer Anda.
2. Download file codeigniter di situs resminya: [www.codeigniter.com](http://www.codeigniter.com)
3. Extract file Codeigniter.zip ke direktori C:/wamp/www/ (jika Anda menggunakan wampserver). Tetapi, jika Anda menggunakan XAMPP. Extract file Codeigniter.zip ke direktori C:/xampp/htdocs/.

4. Pergi ke folder `c:/wamp/www/` (jika Anda menggunakan WAMP) dan rename (ganti nama) file folder codeigniter yang baru di extraxt tadi menjadi nama project Anda.

Misalnya, disini saya ganti menjadi “myproject”. Sehingga terlihat seperti gambar berikut:

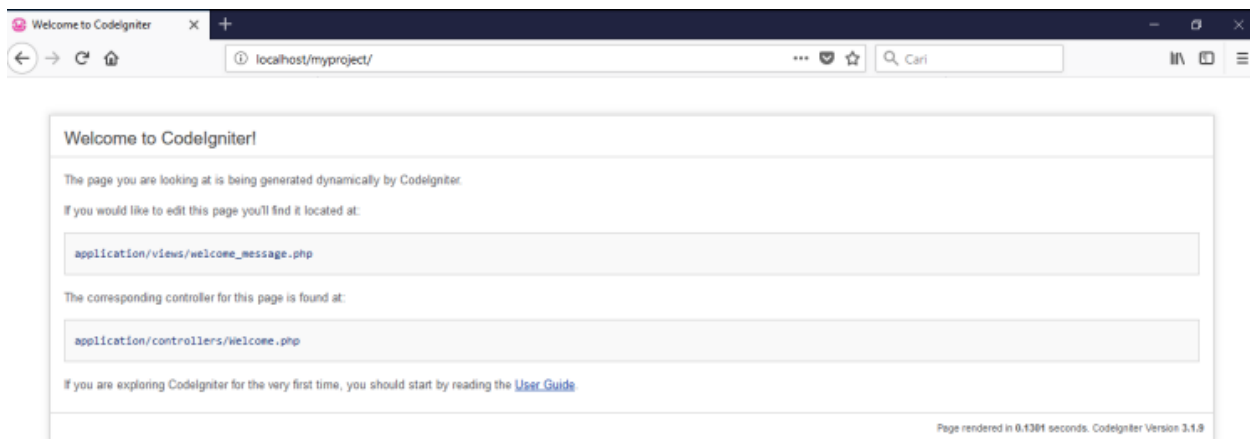


Gambar 3 Contoh File Project

5. Selanjutnya, buka browser Anda. disini saya menggunakan Mozilla Firefox. Kemudian kunjungi URL berikut:

**`http://localhost/myproject/`**

Jika installasi berhasil maka, akan terlihat seperti gambar berikut:

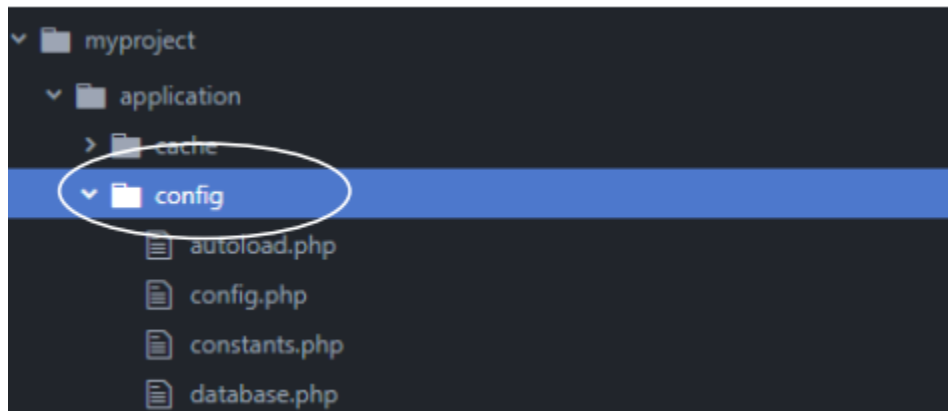


Gambar 4 Tampilan Awal CodeIgniter

## VI. Konfigurasi dasar Codeigniter

Dalam memulai codeigniter, ada beberapa konfigurasi dasar yang perlu Anda ketahui. Yaitu **autoload.php**, **config.php**, dan **database.php**.

Semua konfigurasi pada codeigniter, terletak pada satu tempat yaitu di dalam folder **application/config**.



Gambar 5 Directoty Konfigurasi Awal (*application/config*)

### 1) Autoload.php

Autoload.php, file ini digunakan untuk mengatur fungsi-fungsi yang akan dimuat otomatis di awal ketika program dijalankan. Untuk melakukan konfigurasi pada file autoload.php, silahkan buka folder:

**application/config/autoload.php**

Ada beberapa hal yang bisa diload secara otomatis diantaranya: packages, libraries, drivers, helper files, custom config files, language files, dan models. Untuk konfigurasi dasar yang perlu Anda ketahui adalah **libraries** dan **helper files**. Hal ini bertujuan agar beberapa library dan helper tertentu berjalan secara otomatis. Untuk melakukan konfigurasi pada libraries, buka file **autoload.php** dengan text editor seperti notepad++, sublime text, atau lainnya.

kemudian temukan kode berikut:

```
$autoload['libraries'] = array();
```

Atur menjadi seperti berikut:

```
$autoload['libraries'] = array('database');
```

Pada kode diatas, artinya kita meload library “database” secara otomatis. Dengan demikian Anda dapat menggunakan fungsi-fungsi database pada codeigniter. Seperti fungsi: **Query Builder Class**

Selanjutnya, untuk melakukan konfigurasi pada helper files, buka file **autoload.php** dengan text editor.

kemudian temukan kode berikut:

```
$autoload['helper'] = array();
```

Atur menjadi seperti berikut:

```
$autoload['helper'] = array('url');
```



Pada kode diatas, artinya kita meload helper “url” secara otomatis. Dengan demikian Anda dapat menggunakan fungsi-fungsi url pada codeigniter. Seperti fungsi: `base_url()`, `site_url()`, `URI Segment`, dan sebagainya.

## 2) *Config.php*

Pada file ini terdapat beberapa konfigurasi yang secara standar sudah terkonfigurasi. Namun terdapat beberapa konfigurasi yang perlu diperhatikan yaitu:

```
$config['base_url']  
$config['index_page']  
$config['encryption_key']
```

Untuk konfigurasi dasar, Anda cukup mengetahui konfigurasi **base\_url**. Base\_url merupakan url dasar dari project Anda. Untuk mengkonfigurasi base\_url, buka file **config.php** dengan text editor.

kemudian temukan kode berikut:

```
$config['base_url'] = '';
```

Atur menjadi seperti berikut:

```
$config['base_url'] = 'http://localhost/myproject/';
```

## 3) *Database.php*

Dilihat dari nama filenya maka Anda sudah dapat menangkap apa fungsi dari file ini. File database.php digunakan untuk melakukan konfigurasi yang berkaitan dengan konfigurasi database dari website yang akan dibuat. Adapun konfigurasi yang perlu diperhatikan yaitu: hostname, username, password, dan database.

Untuk melakukan konfigurasi pada database.php. Buka file database.php dengan text editor. Kemudian temukan kode berikut:

```
$active_group = 'default';  
$query_builder = TRUE;  
  
$db['default'] = array(  
    'dsn' => '',  
    'hostname' => 'localhost',  
    'username' => '',  
    'password' => '',
```

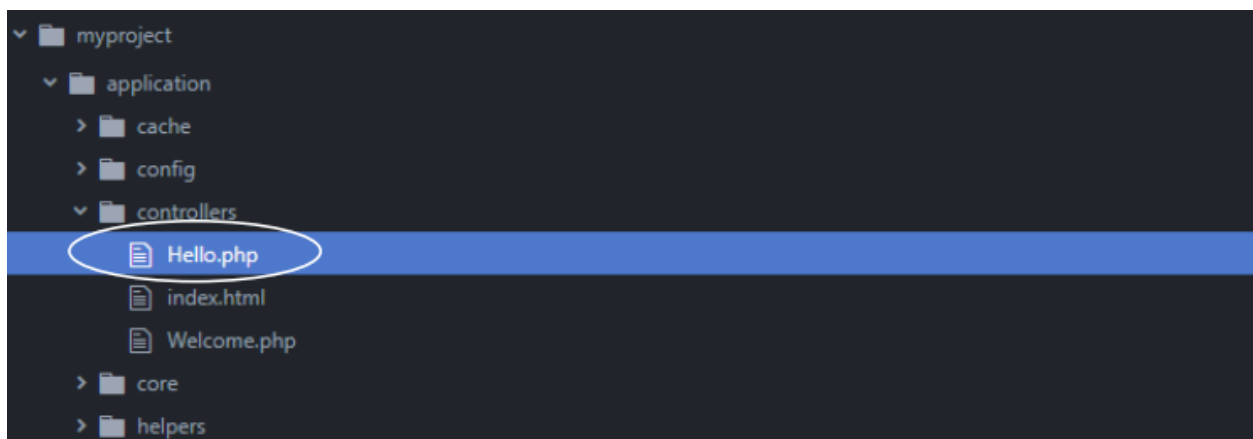
Atur menjadi seperti berikut:

```
$active_group = 'default';  
$query_builder = TRUE;  
  
$db['default'] = array(  
    'dsn' => '',  
    'hostname' => 'localhost', // Hostname  
    'username' => 'root',    // Username  
    'password' => '',        // password  
    'database' => 'database_name', //database name  
    'dbdriver' => 'mysqli',  
    'dbprefix' => '',  
    'pconnect' => FALSE,  
    'db_debug' => (ENVIRONMENT !== 'production'),  
    'cache_on' => FALSE,
```

## VII. Hello World Codeigniter

Jika serius dengan codeigniter, Anda harus mengerti bagaimana sebuah controller bekerja. Untuk lebih jelasnya, saya akan sharing kasus sederhana agar Anda dapat memahami bagaimana controller bekerja. Disini saya mengangkat kasus yaitu bagaimana menampilkan text “Hello World” pada browser menggunakan controller.

Buat sebuah controller dengan nama Hello.php seperti gambar berikut:



Kemudian ketikkan kode berikut:

```
<?php
class Hello extends CI_Controller{

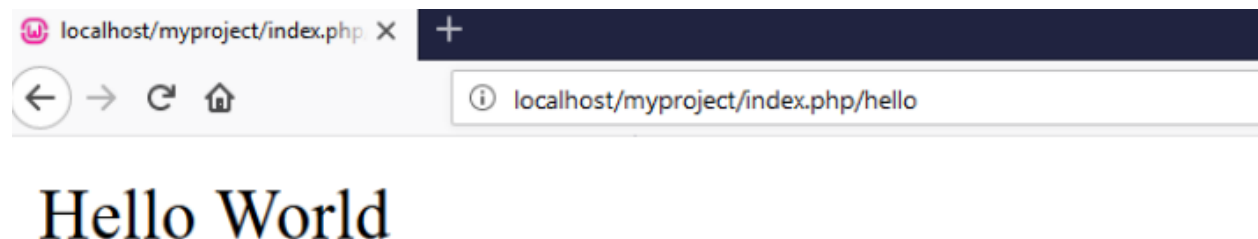
    function index(){
        echo "Hello World";
    }

}
```

NB: Setiap penulisan nama file dan nama class selalu di dahului dengan huruf Capital. Setelah itu save dan buka browser Anda, lalu kunjungi url berikut:

**<http://localhost/myproject/index.php/hello>**

Maka akan akan terlihat text “Hello World” pada browser Anda seperti berikut:



*Gambar 6 Hasil Output Text “Hello World”*

Jika Anda perhatikan dengan seksama, pada dasarnya url pada codeigniter terlihat seperti gambar berikut:



*Gambar 7 Struktur dasar url pada CodeIgniter*

Dimana, terdapat protocol, primary domain, index.php, class name, dan function name. Mungkin terdengar rumit, tapi sebenarnya tidak. Untuk lebih jelasnya silahkan tambahkan satu function lagi pada Controller Hello.php. disini saya beri nama "show". Sehingga controller Hello.php menjadi seperti berikut:

```
<?php
class Hello extends CI_Controller{

    function index(){
        echo "Hello World";
    }

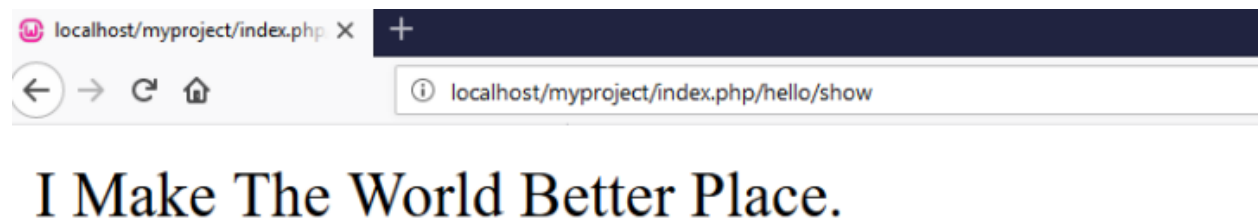
    function show(){
        echo "I Make The World Better Place.";
    }

}
```

Jika Anda jalankan dengan mengunjungi URL berikut:

**<http://localhost/myproject/index.php/hello/show>**

Maka, akan tampil hasilnya seperti berikut:



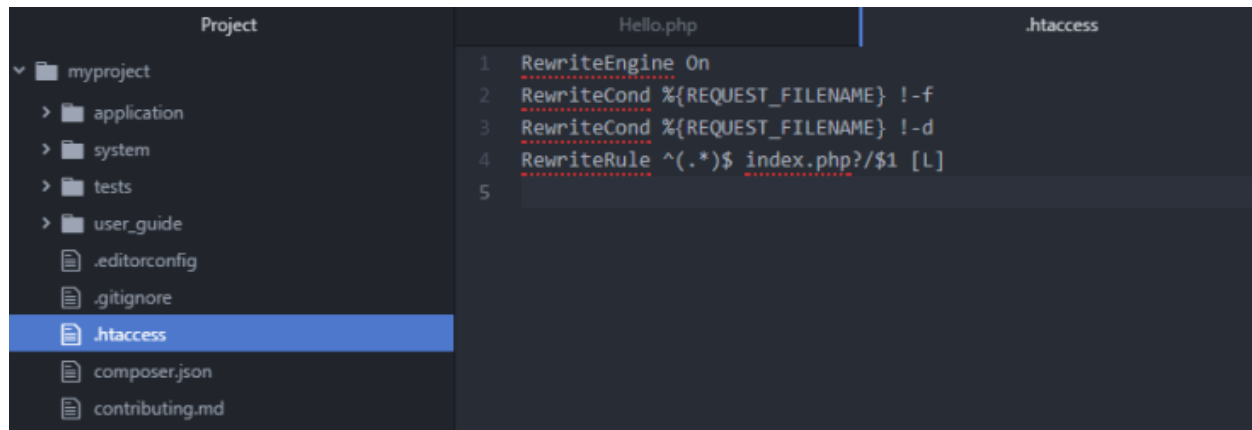
*Gambar 8 Output memanggil Function secara langsung*

## VIII. Menghilangkan index.php pada URL

Codeigniter merupakan framework php yang mendukung clean URL. Dengan demikian Anda dapat membuat URL yang mudah dibaca dan sekaligus SEO Friendly. Pada URL aplikasi "Hello World" diatas, dapat dilihat bahwa adanya index.php pada url yang terlihat mengganggu. Adakah cara untuk menghilangkan index.php dari URL? Tentu saja, Anda dapat menggunakan file **.htaccess** untuk menghilangkannya. Bagaimana membuat file .htaccess? Mari kita mulai. Buat sebuah file dengan nama .htaccess pada web root Anda dan ketikkan kode berikut:

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php?/$1 [L]
```

Seperti gambar berikut:



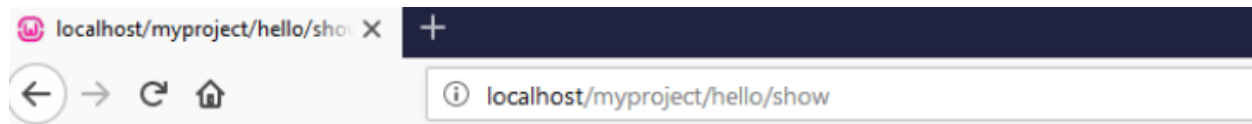
Kemudian buka folder application/config/config.php dengan text editor.

Kemudian temukan kode berikut: `$config['index_page'] = 'index.php';`

Atur menjadi seperti berikut: `$config['index_page'] = '';`

Sekarang silahkan kunjungi url berikut untuk uji coba: <http://localhost/myproject/hello/show>

Maka akan terlihat hasilnya seperti berikut:



## I Make The World Better Place.

Pada gambar diatas, dapat dilihat bahwa URL menjadi lebih rapi dan SEO friendly dengan menghilangkan index.php pada URL.

### IX. Controller dan View

Pada kasus sebelumnya, Anda telah mengetahui bagaimana menampilkan text “Hello World” langsung dari controller. Namun, hal tersebut sebaiknya dilakukan di view. Sekarang saya akan menunjukkan bagaimana menampilkan view melalui controller. Mari kita mulai. Pertama, buat sebuah file pada **application/controller** dengan nama **Blog.php**. Kemudian ketikkan kode berikut:

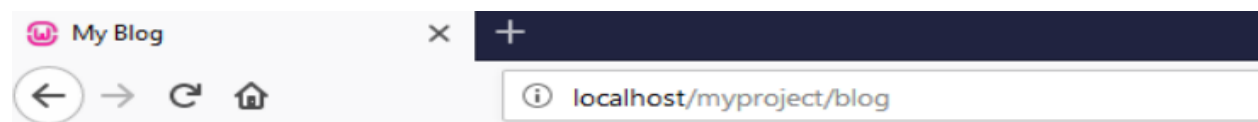
```
<?php
class Blog extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
    }

    function index(){
        $this->load->view('blog_view');
    }
}
```

Kedua, buat sebuah file di **application/views** dengan nama **blog\_view.php**. Kemudian ketikkan kode berikut:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>My Blog</title>
    </head>
    <body>
        <h1>Welcome To My Blog.</h1>
    </body>
</html>
```

Kemudian, buka browser Anda dan akses controller blog. Maka akan terlihat hasilnya seperti berikut:



# Welcome To My Blog.

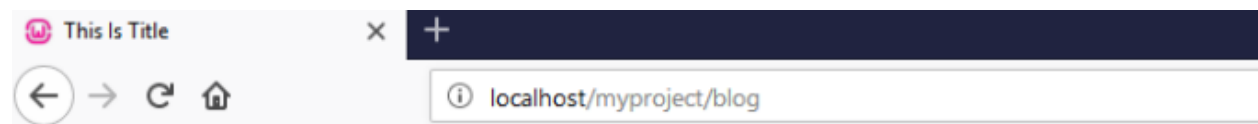
Anda juga dapat mengirimkan **parameter** ke view melalui controller. Sebagai contoh, silahkan ubah controller Blog.php menjadi seperti berikut:

```
function index(){
    $data['title']    = "This Is Title";
    $data['content'] = "This Is The Contents";
    $this->load->view('blog_view',$data);
}
```

Kemudian ubah view blog\_view.php menjadi seperti berikut:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title><?php echo $title;?></title>
  </head>
  <body>
    <h1><?php echo $content;?></h1>
  </body>
</html>
```

Kemudian, buka browser Anda dan akses kembali controller blog. Maka akan terlihat hasilnya seperti berikut:



## This Is The Contents

Saya harap Anda dapat memahami perbedaannya.

### X. Codeigniter dan Bootstrap

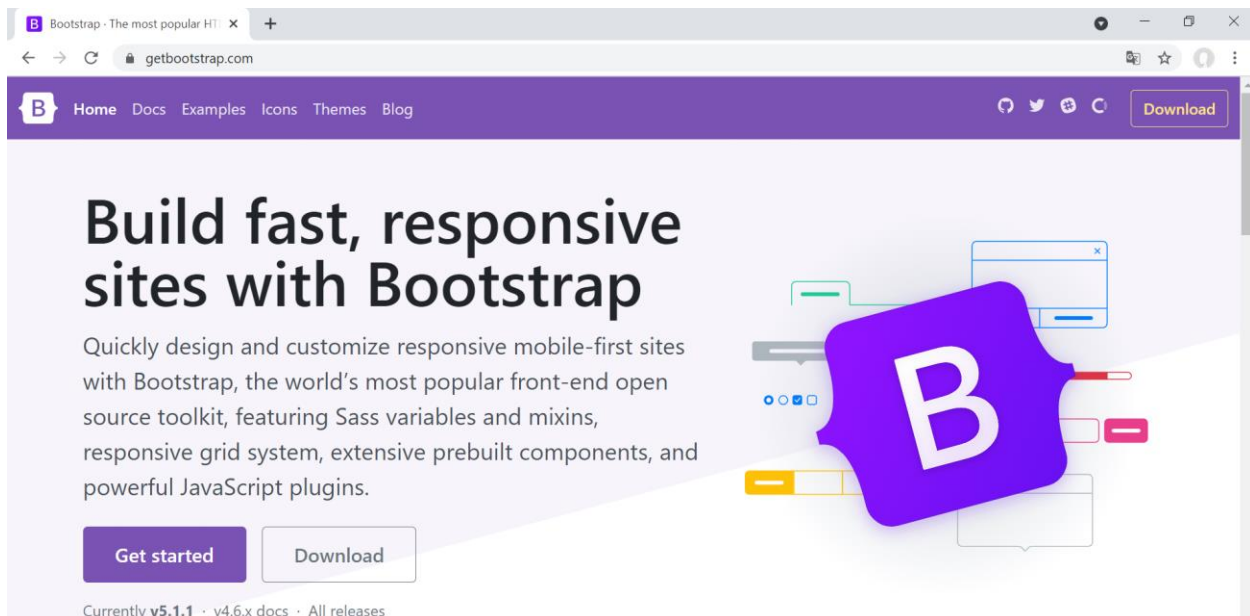
Pada kasus sebelumnya, Anda telah memahami bagaimana memanggil view melalui controller. Sekarang, ada hal yang sangat penting untuk Anda ketahui, yaitu mengkombinasikan codeigniter dengan bootstrap.

Apa itu **BOOTSTRAP**?

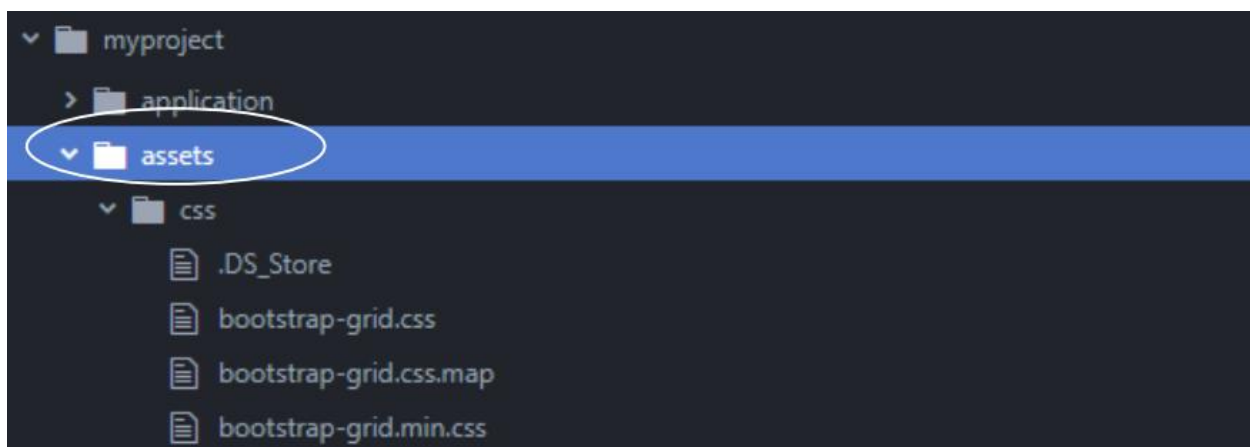
Bootstrap adalah toolkit open source untuk dikembangkan dengan HTML, CSS, dan JS. Dengan kata lain, Bootstrap merupakan framework untuk mempercantik user interface (UI). Bootstrap bersifat responsive. Dengan kata lain, merender dengan baik di berbagai macam perangkat (platform) seperti tablet maupun mobile phone.

Bagaimana mengkombinasikan codeigniter dan bootstrap? **Let's begin.**

Pertama-tama, silahkan download bootstrap di situs resminya [getbootstrap.com](https://getbootstrap.com).



Kedua, buat sebuah folder baru pada project (webroot) Anda. Disini saya beri nama folder “assets”. Kemudian extract file bootstrap yang telah di download tadi kedalam folder assets. Seperti berikut:



Agar Anda dapat memahami seperti apa bootstrap, silahkan edit file view `blog_view.php` menjadi seperti berikut:



```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title><?php echo $title;?></title>
    <!-- load bootstrap css file -->
    <link href="<?php echo base_url('assets/css/bootstrap.min.css');?>"
rel="stylesheet">
  </head>

  <!-- load bootstrap js file -->
  <script src="<?php echo
base_url('assets/js/bootstrap.min.js');?>"></script>
</body>
</html>

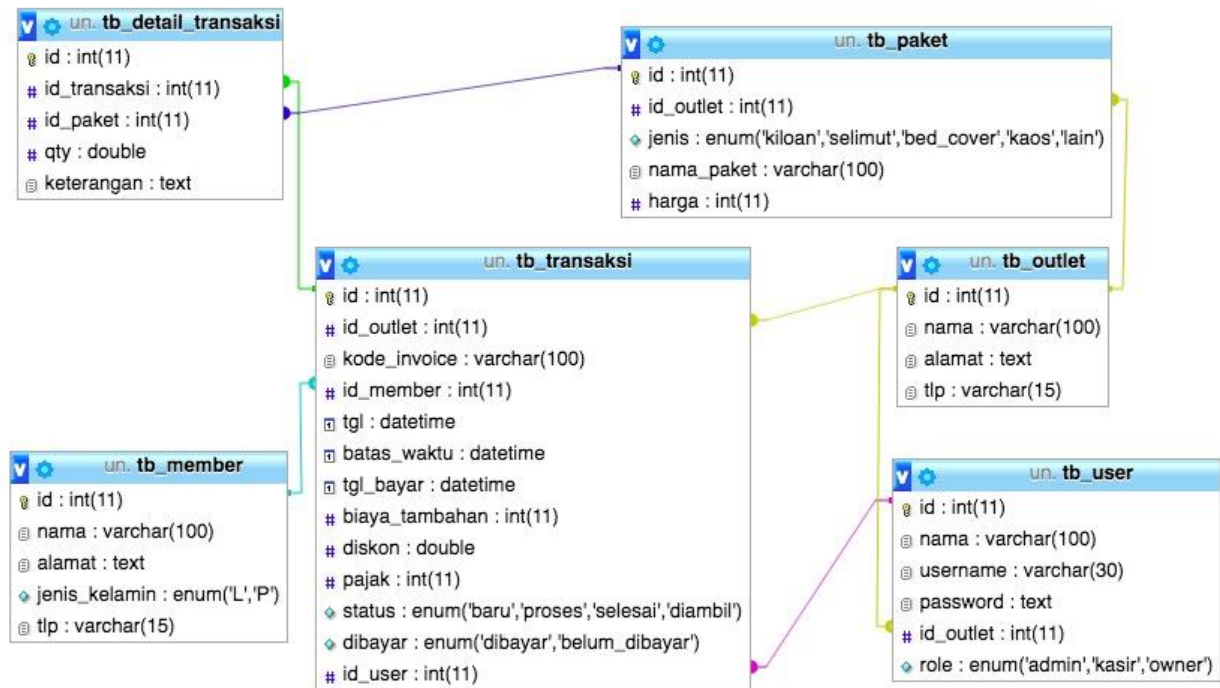
```

## XI. Study Kasus

Pada pembelajaran pemrograman web dan perangkat bergerak akan menggunakan study kasus aplikasi pengelolaan laundry dan akan menggunakan CodeIgniter dan Bootstrap5 dengan fitur yang terdapat pada aplikasi adalah sebagai berikut :

| <i><b>Fitur</b></i>             | <i><b>admin</b></i> | <i><b>kasir</b></i> | <i><b>owner</b></i> |
|---------------------------------|---------------------|---------------------|---------------------|
| <i>Login</i>                    | ✓                   | ✓                   | ✓                   |
| <i>Logout</i>                   | ✓                   | ✓                   | ✓                   |
| Registrasi <u>Pelanggan</u>     | ✓                   | ✓                   |                     |
| CRUD Outlet                     | ✓                   |                     |                     |
| CRUD <u>produk/paket cucian</u> | ✓                   |                     |                     |
| CRUD <u>pengguna</u>            | ✓                   |                     |                     |
| Entri transaksi                 | ✓                   | ✓                   |                     |
| <i>Generate laporan</i>         | ✓                   | ✓                   | ✓                   |

Dengan PDM sebagai berikut :



## BABII

### Membuat Tabel

#### I. Tujuan

1. Siswa dapat membaca, Mengimplementasikan PDM
2. Siswa dapat Membuat Tabel
3. Siswa dapat membuat relasi antar tabel

#### II. Teori

**Physical Data Model (PDM)** menjelaskan/menjabarkan bagaimana data disimpan di komputer dengan menyajikan informasi seperti record formats, record orderings, dan access path.

**Field** merujuk pada kumpulan berbagai karakter didalam database yang mempunyai arti didalamnya. Misal didalam field ada info seputar No. Barang atau sebagainya, maka paparan field harus ada kaitannya dengan nomor barang dan tidak boleh melenceng dari pembahasan tersebut.

**Record** adalah kumpulan field yang sudah lengkap didalam basis data. Kumpulan tersebut biasanya dihitung di satuan baris yang telah tersedia di database.

**Tabel** disebut sebagai kumpulan record dan field yang sudah lengkap di sistem database.

**File** menunjukkan kumpulan beberapa record yang bisa menggambarkan informasi data tertentu dengan baik dalam sebuah database. Contoh file yang ada didalam database adalah informasi berisi data yang berisikan nama barang tertentu didalamnya.

**Data** merupakan kumpulan kejadian dan fakta yang bisa dipakai untuk penyelesaian masalah berbentuk informasi khusus didalam database. Data bisa tercantum dalam bentuk bunyi, gambar, teks, simbol, angka, huruf atau kombinasi beberapa didalamnya.

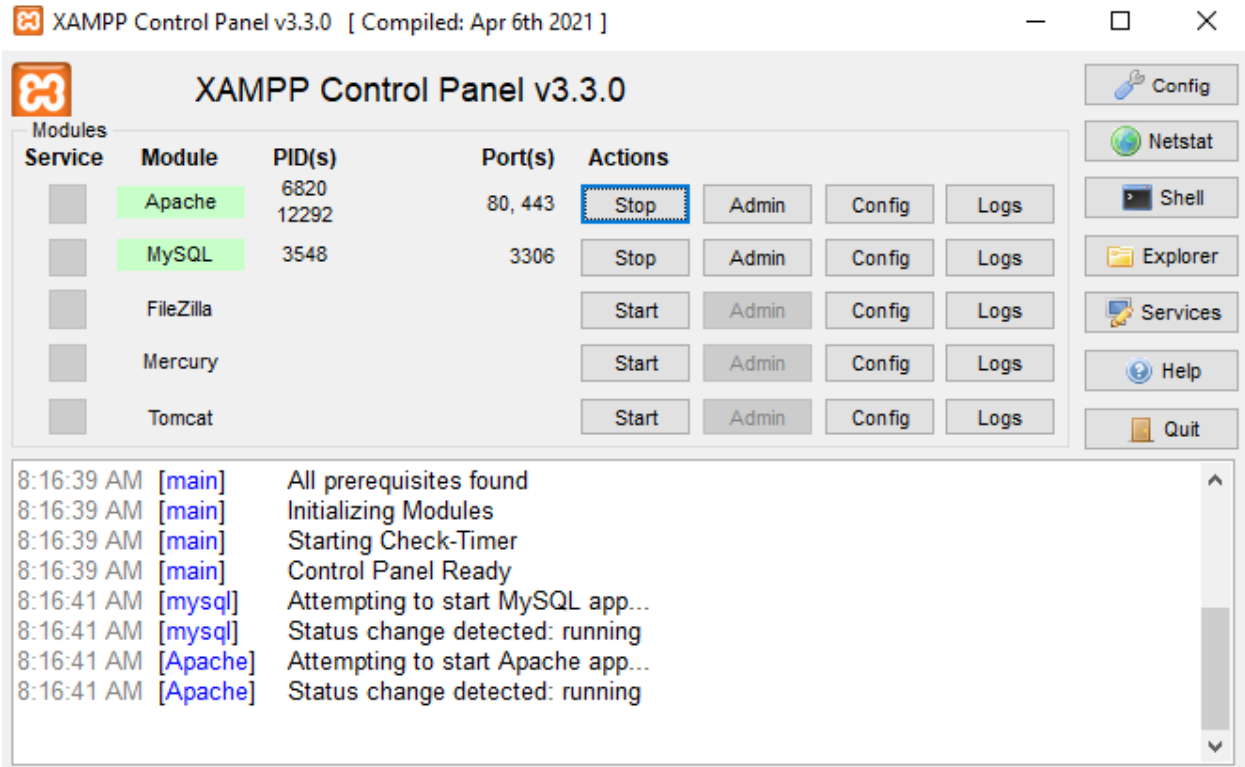
**Database** atau sering juga disebut **basis data** adalah istilah yang terbentuk dari dua kata berbeda. **Basis** adalah gudang atau pusat berkumpulnya sesuatu. Sedangkan **data** adalah kumpulan fakta yang mewakili objek tertentu seperti barang, manusia, benda dan masih banyak lainnya. **Basis data** adalah sekumpulan data yang telah terorganisir dengan rapi dan baik oleh sistem. Semua data tersebut bisa disimpan, dimanipulasi dan bisa dipanggil kapan saja oleh penggunaanya.

**Query:** Sarana untuk mengatur record data yang terdapat dalam tabel serta mengelompokkan field dan record yang ditentukan dari suatu tabel

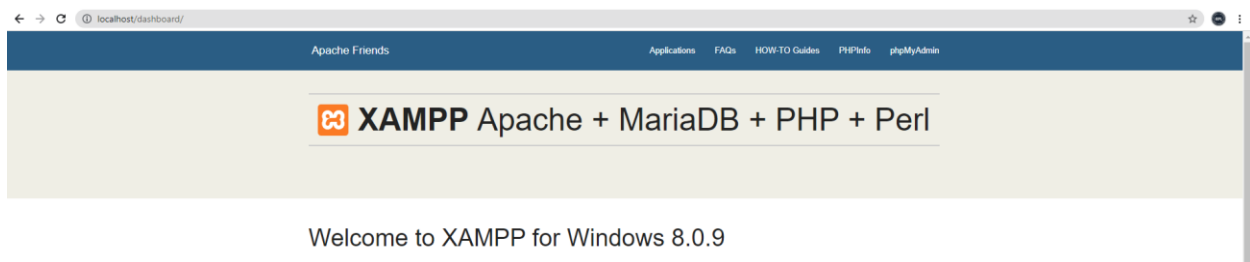
#### III. Praktikum

Pastikan kalian sudah menginstal aplikasi xampp pada komputer masing-masing,

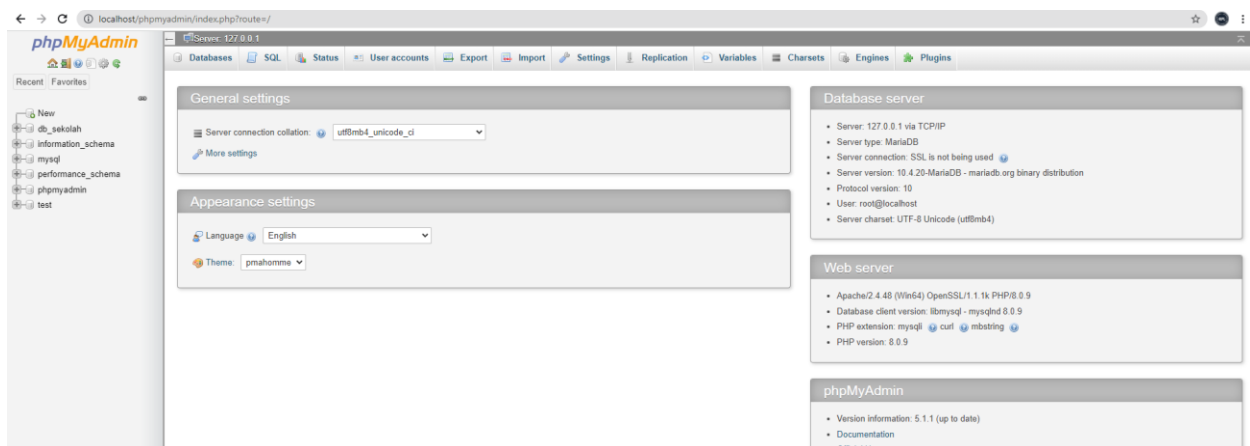
Aktifkan modul Apache dan mysql pada xampp control panel seperti gambar berikut:



Kemudian buka browser (dianjurkan menggunakan google chrome) lalu ketikkan di address bar localhost/ kemudian tekan enter sehingga tampilan pada browser seperti berikut:

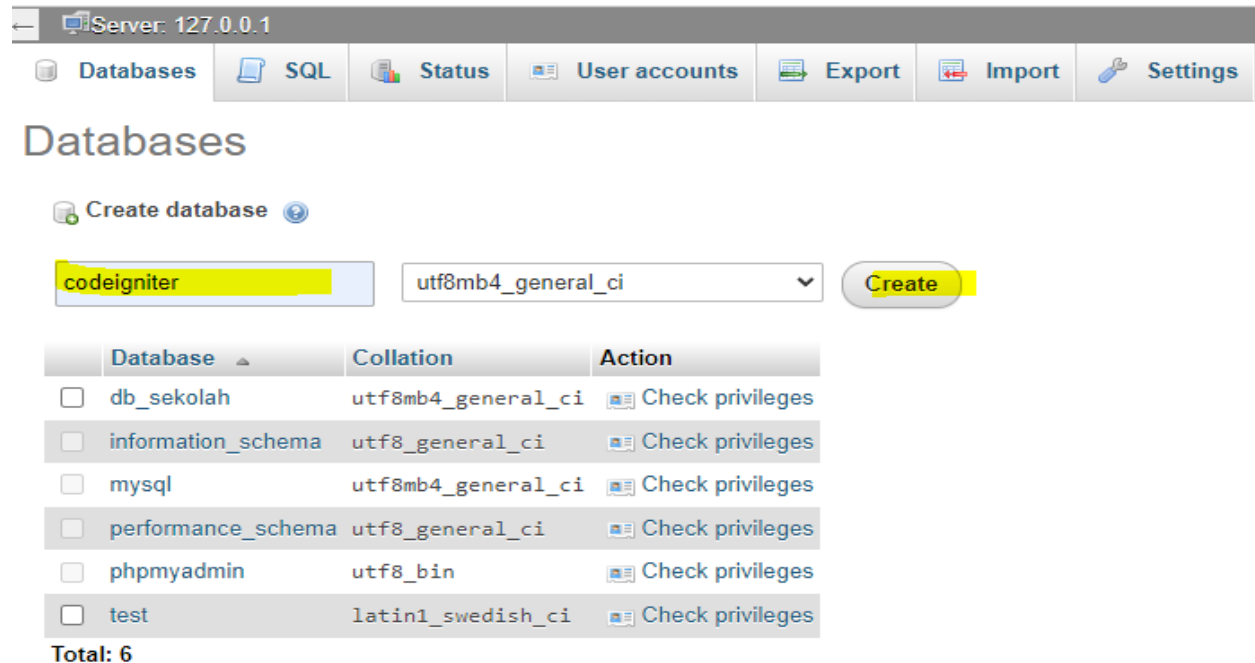


Kemudian pilih menu phpMyAdmin sehingga tampilan pada browser seperti berikut:

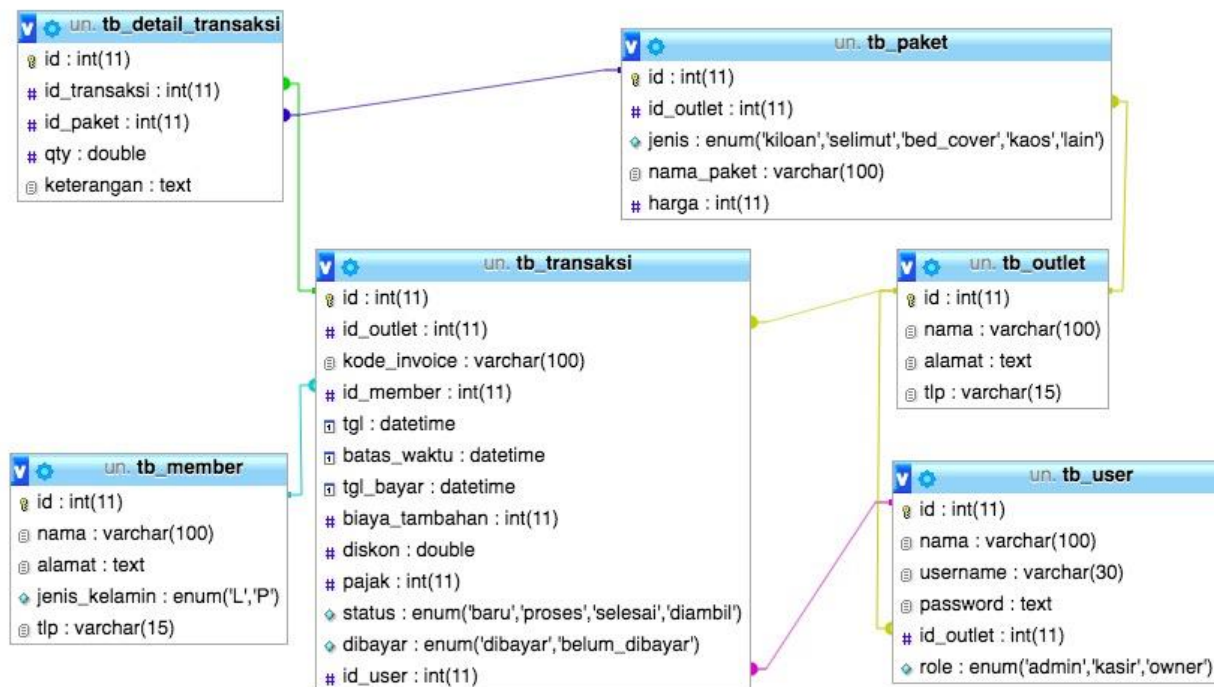


## Membuat Database/Basis Data

Pada tampilan sebelumnya adalah tampilan awal dari phpMyAdmin selanjutnya kita akan membuat database dengancara memilih menu databases kemudian mengisikan nama database lalu tekan tombol create. Contoh tampilan nya seperti berikut:



Setelah dibuat databasenya maka akan muncul tampilan untuk membuat tabel, pada Langkah ini kita akan membuat tabel sesuai dengan PDM pada studi kasus aplikasi pengelolaan laundry sebagai berikut:

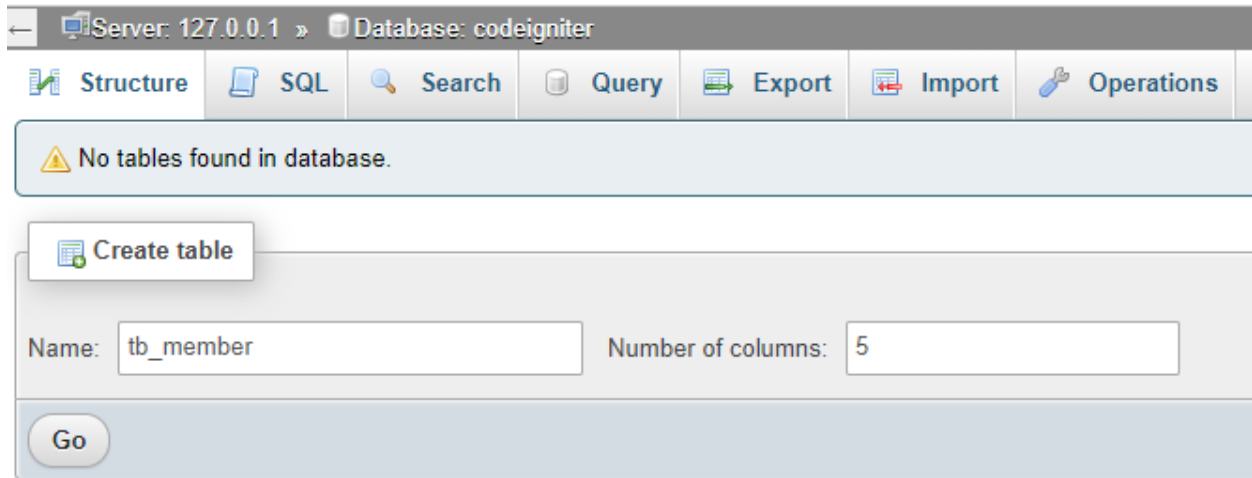


Pada PDM diatas terdapat 6 tabel yang semuanya saling berelasi satu sama lainnya, kita akan mengidentifikasi PDM diatas

setiap relasi antar tabel ditunjukan pada setiap garis yang menghubungkan setiap tabel. Setiap tabel mempunyai kardinalitas bisa one to many, one to one atau many to many. Pada PDM diatas terlihat untuk kardinalitasnya adalah one to many atau many to one yang ditandai dengan primary key pada kardinalitas one akan di masukan pada tabel yang mempunyai kardinalitas many.

## Membuat Tabel

Yang pertama kita akan membuat tabel tb\_meber terlebih dahulu. Contoh sebagai berikut :



Server: 127.0.0.1 » Database: codeigniter

Structure SQL Search Query Export Import Operations

⚠ No tables found in database.

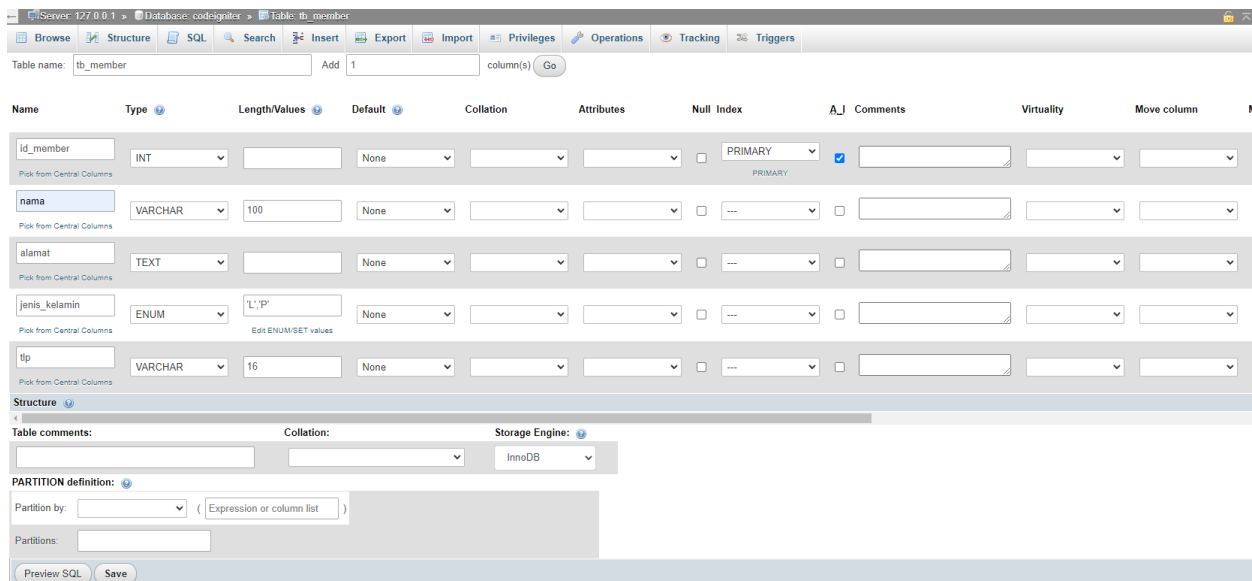
Create table

Name:  Number of columns:

Go

|                   |   |
|-------------------|---|
| Name              | : menandakan nama tabel yang akan dibuat      |
| Number of columns | : menandakan berapa field yang akan kita buat |

Setelah selesai mengisi name dan number of columns kita pilih menu go dan akan diarahkan pada tampilan untuk membuat struktur tabelnya kemudian kita isi sesuai pada PDM dengan nama tabel tb\_member sehingga tampilan nya sebagai berikut:



Server: 127.0.0.1 » Database: codeigniter » Table: tb\_member

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Table name:  Add  column(s) Go

| Name          | Type    | Length/Values | Default | Collation | Attributes | Null                     | Index   | A_I                                 | Comments | Virtuality | Move column | M |
|---------------|---------|---------------|---------|-----------|------------|--------------------------|---------|-------------------------------------|----------|------------|-------------|---|
| id_member     | INT     |               | None    |           |            | <input type="checkbox"/> | PRIMARY | <input checked="" type="checkbox"/> |          |            |             |   |
| nama          | VARCHAR | 100           | None    |           |            | <input type="checkbox"/> |         | <input type="checkbox"/>            |          |            |             |   |
| alamat        | TEXT    |               | None    |           |            | <input type="checkbox"/> |         | <input type="checkbox"/>            |          |            |             |   |
| jenis_kelamin | ENUM    | 'L','P'       | None    |           |            | <input type="checkbox"/> |         | <input type="checkbox"/>            |          |            |             |   |
| tlp           | VARCHAR | 16            | None    |           |            | <input type="checkbox"/> |         | <input type="checkbox"/>            |          |            |             |   |

Structure

Table comments:  Collation:  Storage Engine:

PARTITION definition:

Partition by:  ( Expression or column list )

Partitions:

Preview SQL Save

|                 |  |
|-----------------|--|
| Name            | : menandakan nama field  |
| Type            | Menandakan tipe data yang di gunakan   |
| Length/value    | Menandakan Panjang karakter maksimal pada field  |
| Index {PRIMARY} | Untuk menandakan primary key pada tabel tsb  |
| A_I             | Digit unik saat data disimpan dan apabila tidak disi maka defaultnya akan terisi otomatis dari mulai angka 1 sampai batas maksimum karakter pada field |

Kemudian tekan save dan hasilnya seperti berikut :

The screenshot shows the phpMyAdmin interface for a database named 'codeigniter'. The 'Table: tb\_member' is selected. The 'Table structure' tab is active, displaying a table with 5 columns:

| # | Name          | Type          | Collation          | Attributes | Null | Default | Comments | Extra          | Action           |
|---|---------------|---------------|--------------------|------------|------|---------|----------|----------------|------------------|
| 1 | id_member     | int(11)       |                    |            | No   | None    |          | AUTO_INCREMENT | Change Drop More |
| 2 | nama          | varchar(100)  | utf8mb4_general_ci |            | No   | None    |          |                | Change Drop More |
| 3 | alamat        | text          | utf8mb4_general_ci |            | No   | None    |          |                | Change Drop More |
| 4 | jenis_kelamin | enum('L','P') | utf8mb4_general_ci |            | No   | None    |          |                | Change Drop More |
| 5 | tlp           | varchar(16)   | utf8mb4_general_ci |            | No   | None    |          |                | Change Drop More |

Below the table structure, there are various tools like 'Check all', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Spatial', 'Fulltext', and 'Add to'. At the bottom, there is a 'Propose table structure' section with a 'Go' button.

Lakukan hal yang sama pada tabel tabel yang lain.

### Membuat Relasi Antar Tabel

Sebagai contoh saya akan mengambil relasi antara tabel tb\_member dengan tb\_transaksi, pastikan untuk tb\_transaksi sudah dibuat sesuai dengan PDM yang ada :



Kemudian pada tabel transaksi kita pilih menu relation view seperti gambar berikut :

Server: 127.0.0.1 » Database: codeigniter » Table: tb\_transaksi

Table structure Relation view

| #                           | Name           | Type   | Collation          | Attributes | Null | Default | Comments | Extra          | Action             |
|-----------------------------|----------------|--|--------------------|------------|------|---------|----------|----------------|--------------------|
| <input type="checkbox"/> 1  | id_transaksi   | int(11)                                      |                    |            | No   | None    |          | AUTO_INCREMENT | Change  Drop  More |
| <input type="checkbox"/> 2  | id_outlet      | int(11)                                      |                    |            | No   | None    |          |                | Change  Drop  More |
| <input type="checkbox"/> 3  | kode_invoice   | varchar(100)                                 | utf8mb4_general_ci |            | No   | None    |          |                | Change  Drop  More |
| <input type="checkbox"/> 4  | id_member      | int(11)                                      |                    |            | No   | None    |          |                | Change  Drop  More |
| <input type="checkbox"/> 5  | tgl            | datetime                                     |                    |            | No   | None    |          |                | Change  Drop  More |
| <input type="checkbox"/> 6  | batas_waktu    | datetime                                     |                    |            | No   | None    |          |                | Change  Drop  More |
| <input type="checkbox"/> 7  | tgl_bayar      | datetime                                     |                    |            | No   | None    |          |                | Change  Drop  More |
| <input type="checkbox"/> 8  | biaya_tambahan | int(11)                                      |                    |            | No   | None    |          |                | Change  Drop  More |
| <input type="checkbox"/> 9  | diskon         | double                                       |                    |            | No   | None    |          |                | Change  Drop  More |
| <input type="checkbox"/> 10 | pajak          | int(11)                                      |                    |            | No   | None    |          |                | Change  Drop  More |
| <input type="checkbox"/> 11 | status         | enum('baru', 'proses', 'selesai', 'diambil') | utf8mb4_general_ci |            | No   | None    |          |                | Change  Drop  More |
| <input type="checkbox"/> 12 | dibayar        | enum('dibayar', 'belum_dibayar')             | utf8mb4_general_ci |            | No   | None    |          |                | Change  Drop  More |
| <input type="checkbox"/> 13 | id_user        | int(11)                                      |                    |            | No   | None    |          |                | Change  Drop  More |

Kemudian isi Sehingga tampilan browser seperti berikut:

Server: 127.0.0.1 » Database: codeigniter » Table: tb\_transaksi

Table structure Relation view

Foreign key constraints

| Actions | Constraint properties      | Column                                | Foreign key constraint (InnoDB) |           |           |
|---------|----------------------------|---------------------------------------|---------------------------------|-----------|-----------|
|         |                            |                                       | Database                        | Table     | Column    |
|         | relasi member dengan trans | ON DELETE CASCADE ON UPDATE CASCADE   | codeigniter                     | tb_member | id_member |
|         | Constraint name            | ON DELETE RESTRICT ON UPDATE RESTRICT | codeigniter                     |           |           |

+ Add constraint

+ Internal relationships

Choose column to display: kode\_invoice

Preview SQL Save

Untuk melihat apakah sudah terelasi atau belum maka pada tampilan awal database kita pilih menu designer seperti gambar berikut:

Server: 127.0.0.1 » Database: codeigniter

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Designer Central columns

Filters

Containing the word:

| Table                                 | Action   | Rows     | Type          | Collation                 | Size            | Overhead   |
|---------------------------------------|--|----------|---------------|---------------------------|-----------------|------------|
| <input type="checkbox"/> tb_member    | Browse  Structure  Search  Insert  Empty  Drop | 0        | InnoDB        | utf8mb4_general_ci        | 16.0 KiB        | -          |
| <input type="checkbox"/> tb_transaksi | Browse  Structure  Search  Insert  Empty  Drop | 0        | InnoDB        | utf8mb4_general_ci        | 32.0 KiB        | -          |
| <b>2 tables</b>                       | <b>Sum</b>                                     | <b>0</b> | <b>InnoDB</b> | <b>utf8mb4_general_ci</b> | <b>48.0 KiB</b> | <b>0 B</b> |

↑ ☐ Check all With selected:

Print Data dictionary

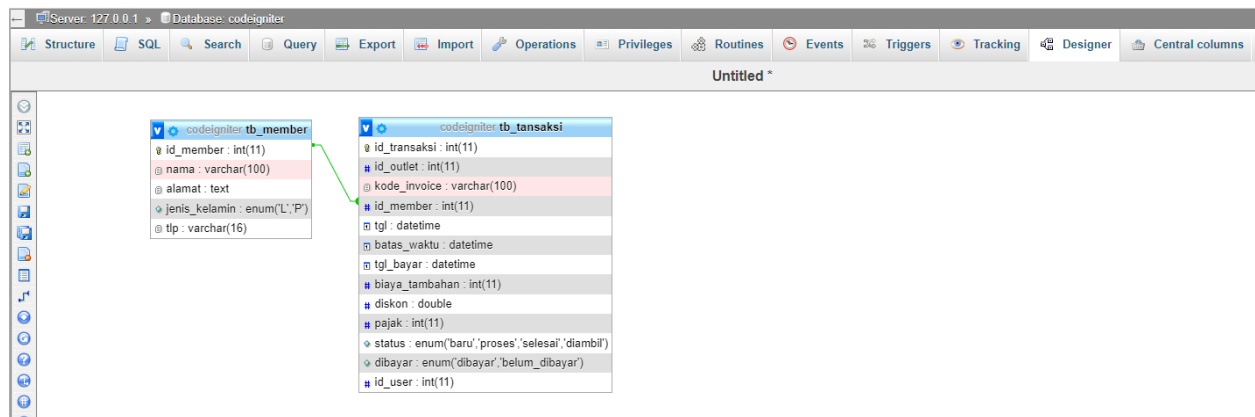
Create table

Name: Number of columns: 4

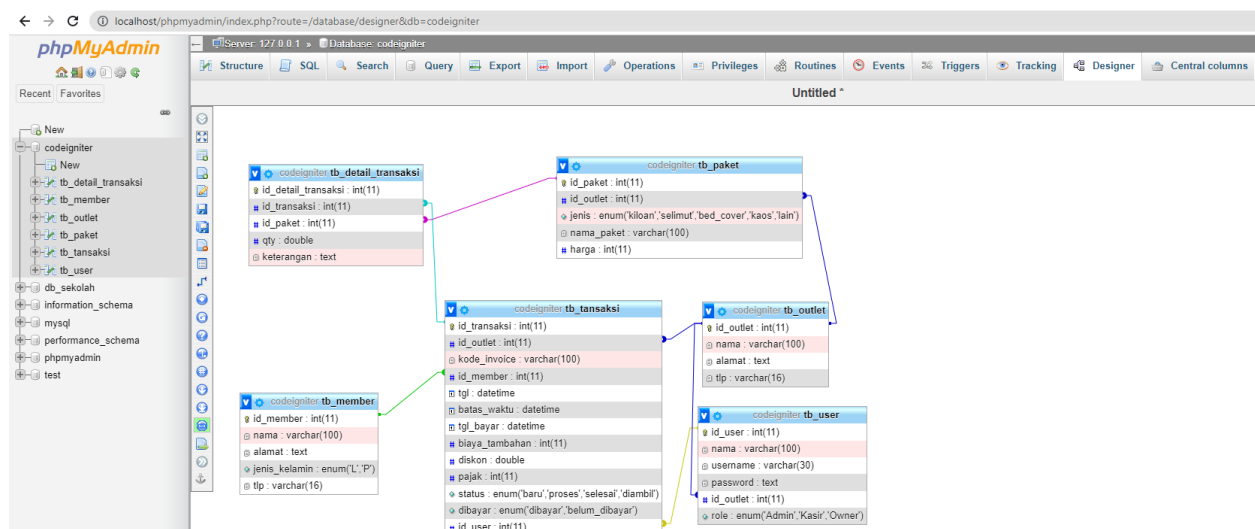
Go



Maka output yang akan di tampilkan pada layar seperti berikut :



Lakukan hal yang sama pada setiap relasi yang ada sesuai dengan PDM. Sehingga desain database yang kita buat sudah sama dengan PDM, berikut contoh nya:



## BAB III

### Membuat Fitur Login

#### I. Tujuan

1. Mengetahui dan menerapkan fitur login multi user pada framework CodeIgniter
2. Mengetahui dan menerapkan session pada framework CodeIgniter

#### II. Teori

**Login** disebut juga “logon” atau “sign in” adalah istilah dalam hal keamanan komputer, yakni berupa proses pintu masuk bagi pengguna untuk mengakses sistem komputer. Login dimaksudkan untuk mengatur proses identifikasi. Proses Login minimal terdiri dari username/akun pengguna dan password untuk mendapatkan hak akses.

**Login Multi User Level** adalah proses login yang memiliki validasi level dari seorang user saat login karena setiap user atau pengguna sistem memiliki hak akses yang berbeda-beda.

**session** merupakan data yang disimpan dalam suatu server yang dapat digunakan secara global di server tersebut, dimana data tersebut spesifik merujuk ke user/client tertentu, contoh penggunaan session adalah ketika user telah login di halaman tertentu, maka ketika membuka halaman lain, php akan mengingat bahwa user tersebut telah login, contoh ketika kita telah login pada Google, maka setiap kali kita membuka layanan Google seperti GMail, Google Drive, dll di tab berbeda, kita akan selalu dalam keadaan login, kecuali kita buka dengan browser yang berbeda. Contoh lain ketika web ecommerce merekam keranjang belanja user, maka ketika pindah ke halaman pembayaran, daftar belanja masih terekam. Enkripsi adalah teknik untuk mengamankan data-data tersebut agar isinya tidak diketahui orang lain.

**Enkripsi** adalah proses teknis yang mengonversikan informasi menjadi kode rahasia, sehingga mengaburkan data yang Anda kirim, terima, atau simpan. Pada dasarnya, sebuah algoritme digunakan untuk mengacak data, sebelum pihak penerima menyatukan kembali data yang diacak tersebut menggunakan kunci dekripsi. Pesan yang ada di dalam file terenkripsi ini disebut sebagai "plaintext" atau teks polos, sementara dalam bentuk terenkripsi pesan itu disebut sebagai "ciphertext" atau teks yang disandikan.

#### III. Praktikum

Untuk contoh login multi user yang akan kita buat adalah login multi user pada **Aplikasi Pengelolaan Laundry**. Jadi level user yang disediakan adalah admin, kasir dan owner.

##### I. Persiapan

Dalam bab ini kita akan membuat coding untuk fitur login pada **Aplikasi Pengelolaan Laundry** untuk persiapan maka kita akan melakukan pengisian data pada tabel `tb_outlet` dan `tb_user`. contoh pada gambar berikut:

Server: 127.0.0.1 » Database: codeigniter » Table: tb\_outlet

Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

`SELECT * FROM `tb_outlet``

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

|   | id_outlet | nama                                 | alamat  | tlp          |
|---|-----------|--------------------------------------|---|--------------|
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete 1 |           | Loundy Bersih Outlet Cabang Banjaran | Jl. Raya Banjaran, Banjaran, Kec. Banjaran, Bandun... | 089654664597 |

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table

Gambar 9 Contoh data tabel tb\_outlet

Server: 127.0.0.1 » Database: codeigniter » Table: tb\_user

Showing rows 0 - 2 (3 total, Query took 0.0004 seconds.)

`SELECT * FROM `tb_user``

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

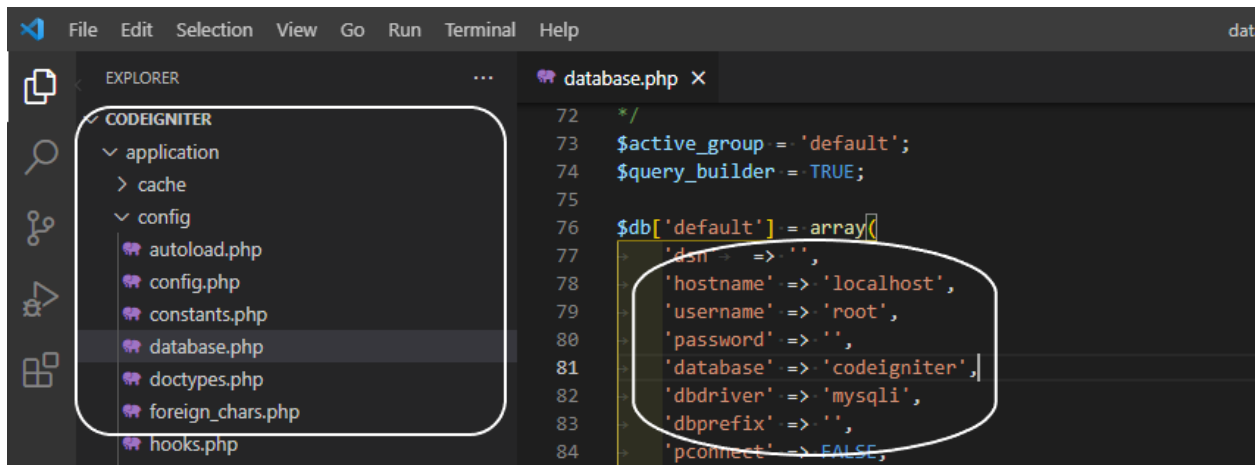
|   | id_user | nama                         | username                | password                         | id_outlet | role  |
|---|---------|------------------------------|-------------------------|----------------------------------|-----------|-------|
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete 2 |         | Bagus Admin Outlet Banjaran  | adminloundry@gmail.com  | 21232f297a57a5a743894a0e4a801fc3 | 1         | Admin |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete 3 |         | Kasir Outlet Banjaran        | kasirbanjaran@gmail.com | d43b33ab1d799cd5fce043c5a826a3ca | 1         | Kasir |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete 4 |         | Owner Outlet Cabang Banjaran | ownerbanjaran@gmail.com | 1055a4fbec70e05f004ed9eea86c7b7  | 1         | Owner |

Check all | With selected: Edit Copy Delete Export

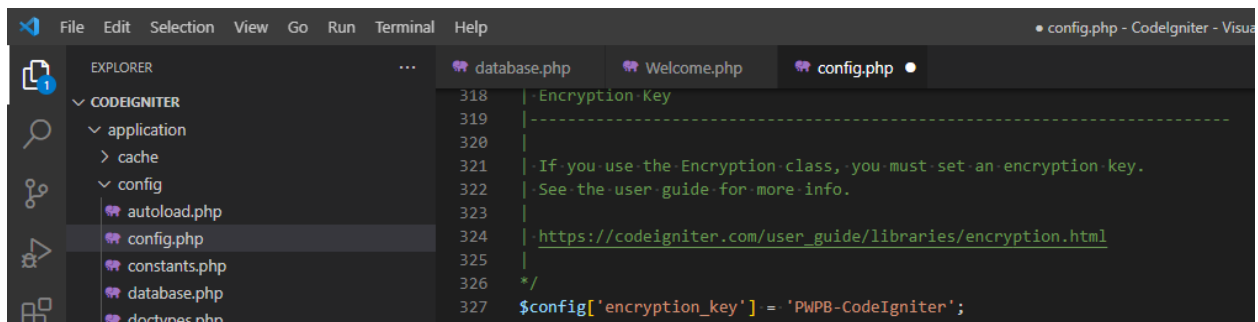
Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Gambar 10 Contoh data tabel tb\_user

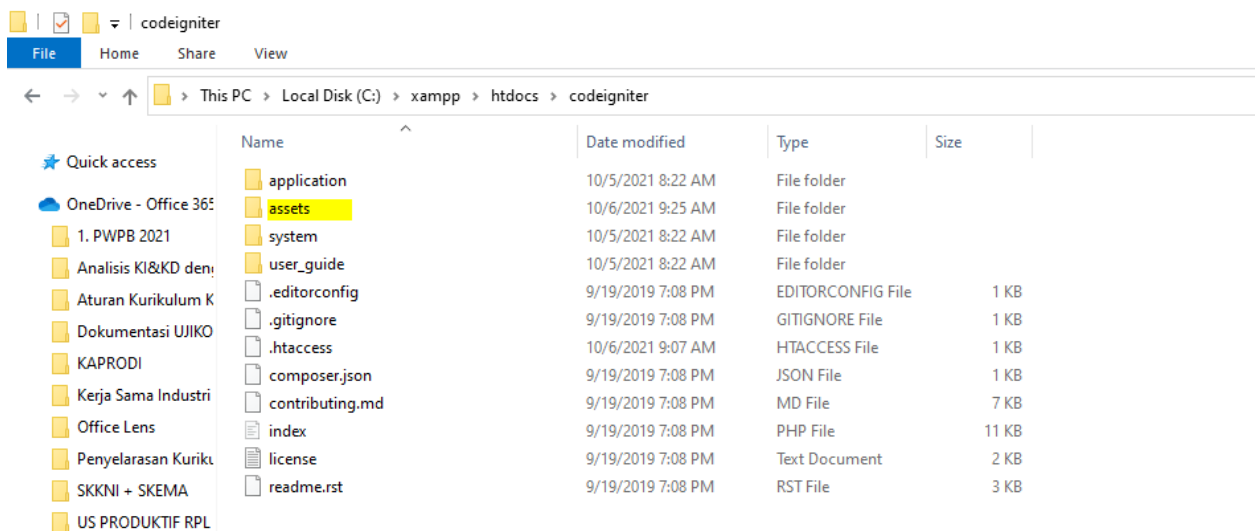
Hubungkan project kita dengan database yang kita buat dengan cara melakukan pengeditan pada file **database.php** folder **application/config/**

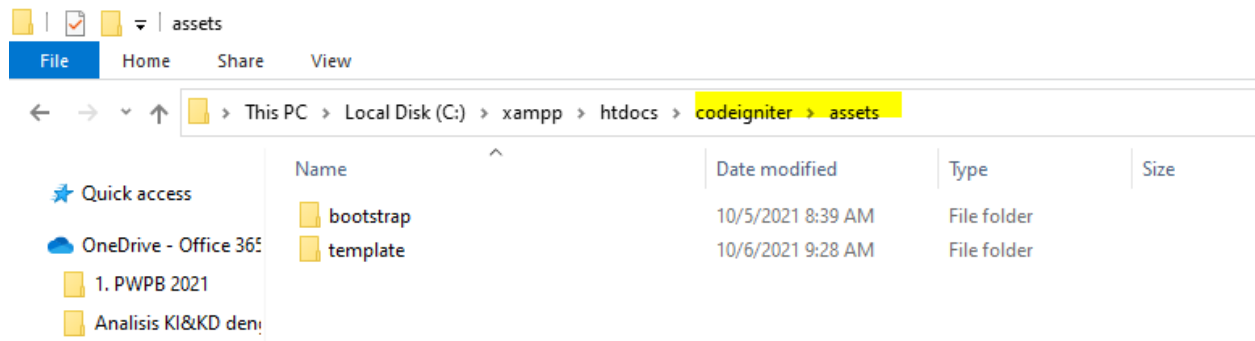


Karena kita akan menggunakan session di codeigniter maka kita harus mengisikan `encryption_key` yang terletak pada `application/config/config.php`.



Untuk UI pada aplikasi yang akan kita buat kita akan menggunakan Bootstraps 5, pastikan anda sudah punya file bootstraps 5 atau bisa didownload di <https://getbootstrap.com/>, setelah di download kemudian buat sebuah folder assets pada root project kalian, extract bootstrap pada folder tersebut. Contohnya sebagai berikut:





Dalam folder asset terdapat 2 folder yakni bootstrap dan template, folder bootstraps untuk pembuatan Form login. Sedangkan untuk folder template adalah untuk peruntukan untuk halaman backend (admin,kasir,owner). Untuk template yang dipergunakan silahkan kalian mencari sendiri template untuk bagian backend.

## II. Proses Coding

### Membuat Controller

Buat controller baru dengan nama Login.php dengan script sebagai berikut:

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Login extends CI_Controller {

    public function index()
    {
        $this->load->view('login/Form_login');
    }

    public function ceklogin()
    {
        $this->form_validation->set_rules('email', 'Email', 'required');
        $this->form_validation->set_rules('password', 'Password', 'required');

        if($this->form_validation->run() != false){

            // menangkap data username dan password dari halaman login
            $username = $this->input->post('email');
            $password = $this->input->post('password');

            $where = array(
                'username' => $username,
                'password' => md5($password)
            );
        }
    }
}
```

```

$this->load->model('m_data');

// cek kesesuaian login pada table pengguna
$cek = $this->m_data->cek_login('tb_user',$where)->num_rows();

// cek jika login benar
if($cek > 0){

    // ambil data pengguna yang melakukan login
    $data = $this->m_data->cek_login('tb_user',$where)->row();
    $role = $data->role;
    if ($role=='Admin') {
        // buat session untuk pengguna yang berhasil login
        $data_session = array(
            'id' => $data->id_user,
            'nama' => $data->nama,
            'username' => $data->username,
            'role' => $data->role,
            'status' => 'telah_login'
        );
        $this->session->set_userdata($data_session);

        // alihkan halaman ke halaman dashboard pengguna(admin)

        redirect(base_url(). 'admin');
    }elseif ($role=='Kasir') {
        // buat session untuk pengguna yang berhasil login
        $data_session = array(
            'id' => $data->id_user,
            'nama' => $data->nama,
            'username' => $data->username,
            'role' => $data->role,
            'status' => 'telah_login'
        );
        $this->session->set_userdata($data_session);

        // alihkan halaman ke halaman dashboard pengguna(kasir)
        redirect(base_url(). 'kasir');
    }elseif ($role=='Owner') {
        // buat session untuk pengguna yang berhasil login
        $data_session = array(
            'id' => $data->id_user,
            'nama' => $data->nama,
            'username' => $data->username,
            'role' => $data->role,

```

```

        'status' => 'telah_login'
    );
    $this->session->set_userdata($data_session);

    // alihkan halaman ke halaman dashboard pengguna (owner)
    redirect(base_url().'owner');
}
}else{
    redirect(base_url().'login?alert=gagal');
}
}else{
    $this->load->view('login/Form_login');
}
}
}

```

Selanjutnya kita membuat controller admin, owner dan kasir dengan contoh membuat controller Admin.php scripting sebagai berikut:

```

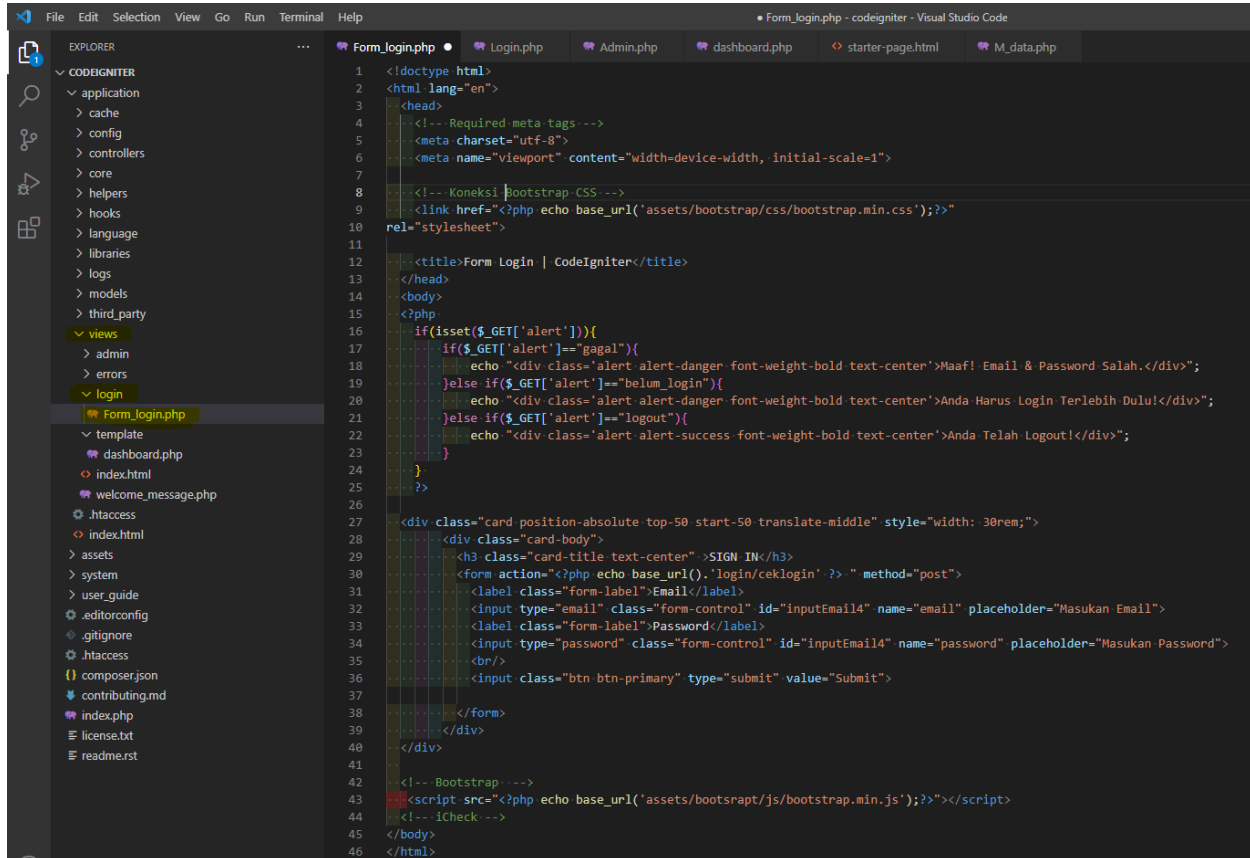
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class Admin extends CI_Controller {
5
6      function __construct()
7      {
8          parent::__construct();
9
10         date_default_timezone_set('Asia/Jakarta');
11
12         $this->load->model('m_data');
13
14         // cek session yang login,
15         // jika session status tidak sama dengan session telah_login, berarti pengguna belum login
16         // maka halaman akan di alihkan kembali ke halaman login.
17         if($this->session->userdata('status')!="telah_login"){
18             redirect(base_url().'login?alert=belum_login');
19         }
20     }
21
22     public function index()
23     {
24         $this->load->view('admin/dashboard');
25         // $this->load->view('admin/v_index');
26         // $this->load->view('admin/v_footer');
27     }
28
29     public function keluar()
30     {
31         $this->session->sess_destroy();
32         redirect('login?alert=logout');
33     }
34 }

```

Lakukan hal yang sama pada setiap hak akses yang lainnya

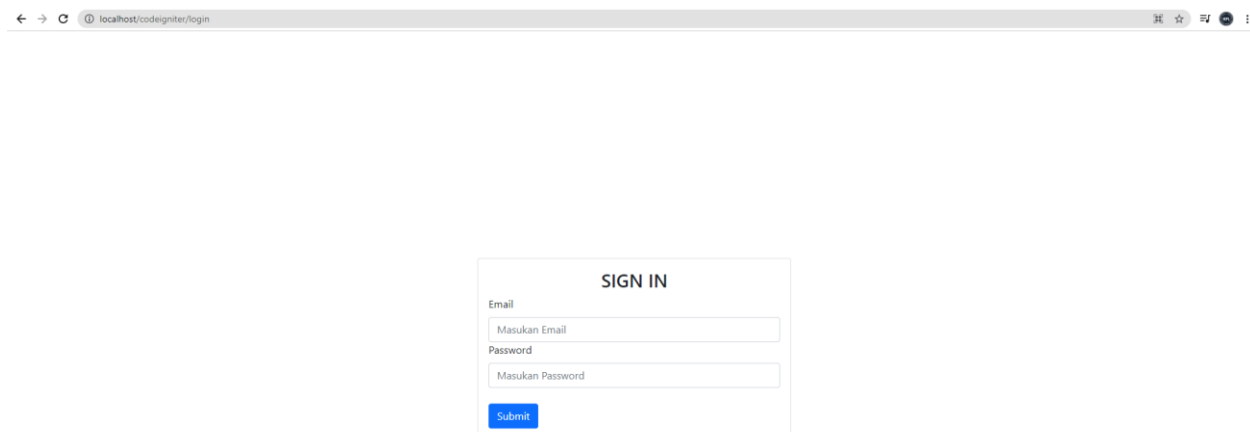
## Membuat Form login

Untuk membuat sebuah tampilan maka folder yang kita gunakan adalah view, pada folder view semua tampilan aplikasi disimpan disana, yang terletak pada **application/view**. Selanjutnya kita membuat Form login dengan nama file Form\_login.php yang terdapat pada folder baru dengan nama login. Contohnya seperti berikut:



```
1 <!doctype html>
2 <html lang="en">
3 <head>
4     <!-- Required meta tags -->
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7
8     <!-- Koneksi Bootstrap CSS -->
9     <link href="<?php echo base_url('assets/bootstrap/css/bootstrap.min.css');?>"
10     rel="stylesheet">
11
12     <title>Form Login | CodeIgniter</title>
13 </head>
14 <body>
15 <?php
16     if(isset($_GET['alert'])){
17         if($_GET['alert']=="gagal"){
18             echo "<div class='alert alert-danger font-weight-bold text-center'>Maaf! Email & Password Salah.</div>";
19         }else if($_GET['alert']=="belum_login"){
20             echo "<div class='alert alert-danger font-weight-bold text-center'>Anda Harus Login Terlebih Dulu!</div>";
21         }else if($_GET['alert']=="logout"){
22             echo "<div class='alert alert-success font-weight-bold text-center'>Anda Telah Logout!</div>";
23         }
24     }
25 >
26
27 <div class="card position-absolute top-50 start-50 translate-middle" style="width: 30rem;">
28     <div class="card-body">
29         <h3 class="card-title text-center">SIGN IN</h3>
30         <form action="<?php echo base_url().<?php echo base_url('login/ceklogin')?>" method="post">
31             <label class="form-label">Email</label>
32             <input type="email" class="form-control" id="inputEmail4" name="email" placeholder="Masukan Email">
33             <label class="form-label">Password</label>
34             <input type="password" class="form-control" id="inputPassword4" name="password" placeholder="Masukan Password">
35             <br/>
36             <input class="btn btn-primary" type="submit" value="Submit">
37         </form>
38     </div>
39 </div>
40
41 <!-- Bootstrap JS -->
42 <script src="<?php echo base_url('assets/bootstrap/js/bootstrap.min.js');?>"></script>
43 <!-- jQuery -->
44 </body>
45 </html>
```

Kemudian simpan, untuk melihat hasilnya kalian ketikan <http://localhost/codeigniter/login> dan tampilan nya akan seperti berikut:

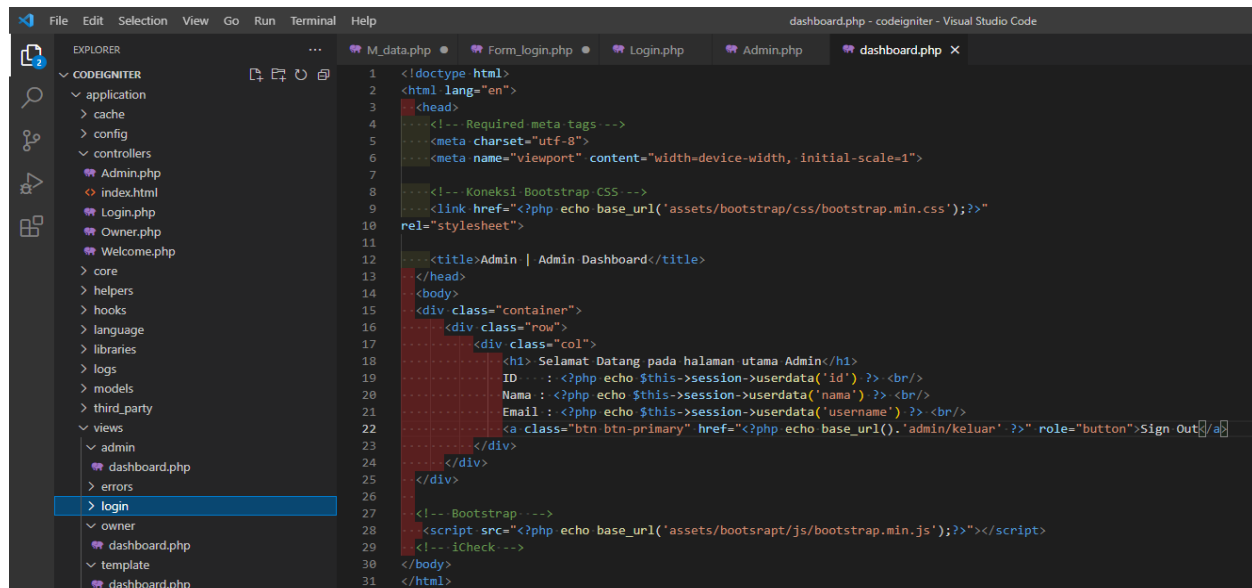


Gambar 11 Tampilan Form Login



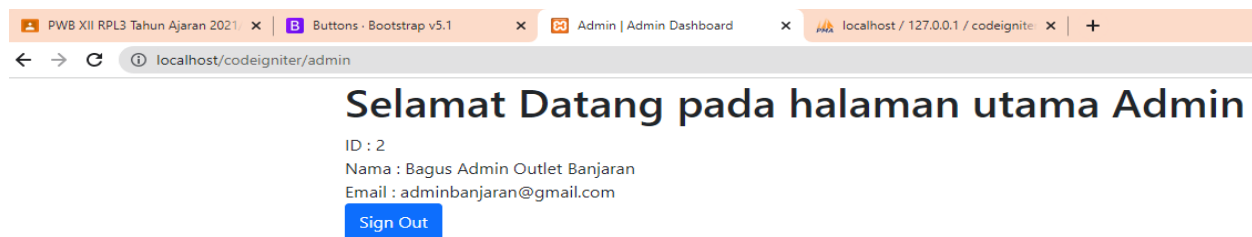
### Membuat halaman Admin

Selanjutnya saya akan mencontohkan membuat halaman utama untuk hak akses admin dengan contoh scripting berikut:



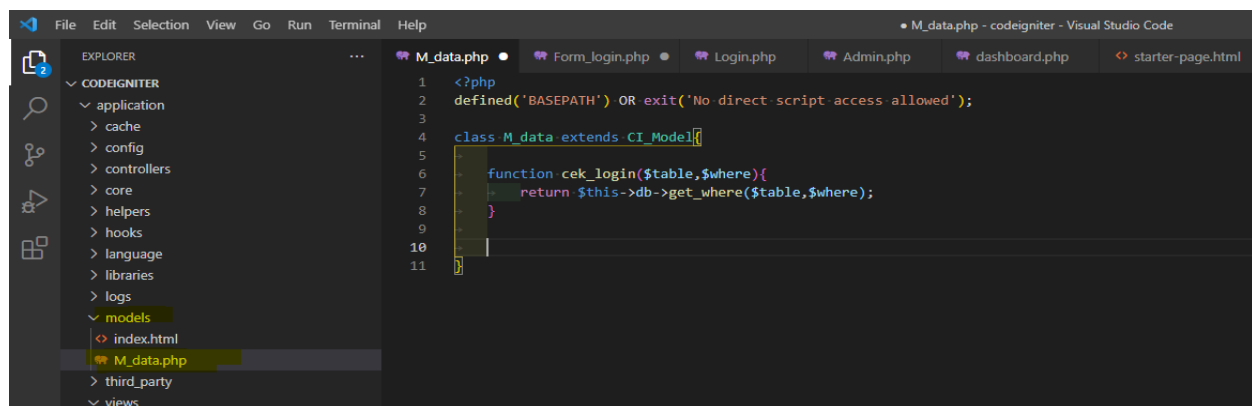
```
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <!-- Required meta tags -->
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7
8 <!-- Koneksi Bootstrap CSS -->
9 <link href="<?php echo base_url('assets/bootstrap/css/bootstrap.min.css');?>"
10 rel="stylesheet">
11
12 <title>Admin | Admin Dashboard</title>
13 </head>
14 <body>
15 <div class="container">
16 <div class="row">
17 <div class="col">
18 <h1> Selamat Datang pada halaman utama Admin</h1>
19 ID : <?php echo $this->session->userdata('id') ?> <br/>
20 Nama : <?php echo $this->session->userdata('nama') ?> <br/>
21 Email : <?php echo $this->session->userdata('username') ?> <br/>
22 <a class="btn btn-primary" href="<?php echo base_url(). 'admin/keluar' ?>" role="button">Sign Out</a>
23 </div>
24 </div>
25 </div>
26
27 <!-- Bootstrap JS -->
28 <script src="<?php echo base_url('assets/bootstrap/js/bootstrap.min.js');?>"></script>
29 <!-- iCheck -->
30 </body>
31 </html>
```

Jika kita berhasil masuk dengan hak akses admin maka akan muncul tampilan sebagai berikut :



### Membuat model data

Kemudian untuk akses ke database kita harus berfokus pada folder **application/model/** , buat sebuah file dengan nama **M\_data.php** dan scripting seperti berikut:



```
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class M_data extends CI_Model
5 {
6     function cek_login($table,$where){
7         return $this->db->get_where($table,$where);
8     }
9 }
10
11
```

Kemudian Simpan setiap pekerjaan kalian dan mulai untuk uji coba.

### ***III. Pengujian***

Setelah semua Langkah, scripting sudah dilaksanakan Langkah selanjutnya kita lakukan pengujian pada fitur yang telah kita buat.

#### *Tugas*

1. Buatlah dokumen pengujian pada office word kemudian beri nama [“Pengujian Fitur Login – {nama} {kelas}”].
2. Capture setiap tampilan User Interface yang telah dibuat dan simpan pada dokumen pengujian. {Form Login, Halaman Admin, Halaman Kasir, Halaman Owner}
3. Capture setiap error yang terjadi (bila terjadi error) dan simpan pada dokumen pengujian .
4. Lakukan perbaikan pada kesalahan/error yang terjadi , serta catat dan simpan pada dokumen pengujian.
5. Dokumen pengujian diserahkan kepada guru yang bersangkutan.

## BAB III

### Create Read Update Delete

#### I. Tujuan

1. Mengetahui CRUD dasar framework codeigniter
2. Penerapan MVC dengan Teknik OOP

#### II. Teori

CRUD adalah singkatan dari Create, Read, Update, dan Delete. Proses ini sangat berkaitan dengan pengambilan atau transaksi data dari atau ke database. Hal ini menjadi krusial apabila berhubungan dengan sistem informasi perusahaan karena data yang diproses biasanya merupakan data transaksi.

Bagi PHP Developer, operasi CRUD biasanya menjadi pilar untuk mempelajari proses pengelolaan data menggunakan PHP dan tentu saja MySQL database. Nah CRUD ini bisa diibaratkan sebagai jalur atau koneksi yang menghubungkan antara bahasa pemrograman PHP dengan MySQL.

Lebih jelasnya, berikut ini ulasan operasi CRUD beserta contohnya:

Create (C) merupakan proses pembuatan data baru. Proses ini biasanya dilakukan ketika Anda mendaftar pada sebuah halaman website. Data yang Anda masukkan akan disimpan di dalam database menggunakan operasi Create.

Read (R) merupakan proses pengambilan data dari database. Proses ini biasanya terjadi ketika Anda ingin melakukan proses login di halaman website tertentu. Saat Anda klik "Login" maka website akan menjalankan proses "Read" untuk melakukan verifikasi akun yang Anda gunakan.

Contoh operasi yang lebih sederhana yaitu ketika Anda mengakses sebuah halaman website, saat itu pula proses Read berjalan. Hasil dari operasi ini adalah tampilan halaman website yang Anda lihat di browser.

Update (U) adalah proses mengubah data yang berada di dalam database. Contoh proses ini ketika Anda mengubah profil di dalam akun sosial media. Saat Anda klik "Ubah", website akan mengirimkan proses Update ke dalam database. Kemudian database meresponnya dengan mengubah data lama menjadi data baru yang Anda tambahkan melalui halaman profil.

Delete (D) adalah proses untuk menghapus data yang ada di database. Proses ini mirip dengan Update, bedanya, proses Delete akan mengubah data yang ada di database menjadi 'tidak ada'.

#### III. Praktikum

Pada kegiatan praktikum saya akan mengambil contoh untuk pembuatan CRUD pada data member dengan struktur tabel seperti berikut:



| un.tb_member  |               |
|---------------|---------------|
| id            | int(11)       |
| nama          | varchar(100)  |
| alamat        | text          |
| jenis_kelamin | enum('L','P') |
| tlp           | varchar(15)   |

*Menambahkan Fungsi pada Controller admin*

Menambahkan fungsi member, member\_tambah, member\_aksi, member\_edit, member\_update dan member\_hapus pada controller admin. Dengan listing program seperti berikut:

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Admin extends CI_Controller {

    function __construct()
    {
        parent::__construct();

        date_default_timezone_set('Asia/Jakarta');

        $this->load->model('m_data');

        // cek session yang login,
        // jika session status tidak sama dengan session telah_login, berarti pen
        gguna belum login
        // maka halaman akan di alihkan kembali ke halaman login.
        if($this->session->userdata('status')!="telah_login"){
            redirect(base_url().'login?alert=belum_login');
        }
    }

    public function index()
    {
        $this->load->view('admin/dashboard');
        // $this->load->view('admin/v_index');
        // $this->load->view('admin/v_footer');
    }

    public function keluar()
    {
        $this->session->sess_destroy();
        redirect('login?alert=logout');
    }

    // CRUD Member
    public function member()
    {
        $data['member'] = $this->m_data->get_data('tb_member')->result();
        $this->load->view('admin/member',$data);
    }
}
```

```

public function member_tambah()
{
    $this->load->view('admin/f_tambah_member');
}

public function member_aksi()
{
    //validasi input
    $this->form_validation->set_rules('id_member','id_member','required');
    $this->form_validation->set_rules('nama','nama','required');
    $this->form_validation->set_rules('alamat','alamat','required');
    $this->form_validation->
>set_rules('jenis_kelamin','jenis_kelamin','required');
    $this->form_validation->set_rules('tlp','tlp','required');
    //chek kondisi validasi
    if($this->form_validation->run() != false){
        //ambil input dari form
        $id_member = $this->input->post('id_member');
        $nama = $this->input->post('nama');
        $alamat = $this->input->post('alamat');
        $jenis_kelamin = $this->input->post('jenis_kelamin');
        $tlp = $this->input->post('tlp');
        // data yang di simpan ke DB
        $data = array(
            'id_member' => $id_member,
            'nama' => $nama,
            'alamat' => $alamat,
            'jenis_kelamin' => $jenis_kelamin,
            'tlp' => $tlp
        );
        // perintah untuk menambahkan data ke DB melalui model m_data
        $this->m_data->insert_data($data,'tb_member');
        // halaman di arahkan ke halaman admin member
        redirect(base_url().'admin/member');

    }else{
        // jika proses input tidak berhasil akan di arahkan ke halaman tambah
member
        $this->load->view('admin/f_tambah_member');
    }
}

public function member_edit($id_member) // mengambil data dari button edit

```

```

{
    // kondisi data yang akan di ambil dari database
    $where = array(
        'id_member' => $id_member
    );

    $data['member'] = $this->m_data->edit_data($where,'tb_member')-
>result(); // perintah ambil data dari tabel member
    $this->load-
>view('admin/f_member_edit',$data); // ambil data UI form edit member
}

public function member_update()
{
    //validasi update
    $this->form_validation->set_rules('id_member','id_member','required');
    $this->form_validation->set_rules('nama','nama','required');
    $this->form_validation->set_rules('alamat','alamat','required');
    $this->form_validation-
>set_rules('jenis_kelamin','jenis_kelamin','required');
    $this->form_validation->set_rules('tlp','tlp','required');
    // cek kondisi validasi
    if($this->form_validation->run() != false){
        // ambil data dari form edit member
        $id_member = $this->input->post('id_member');
        $nama = $this->input->post('nama');
        $alamat = $this->input->post('alamat');
        $jenis_kelamin = $this->input->post('jenis_kelamin');
        $tlp = $this->input->post('tlp');
        // untuk kondisi data yang akan di update
        $where = array(
            'id_member' => $id_member
        );
        // data yang akan di update
        $data = array(
            'nama' => $nama,
            'alamat' => $alamat,
            'jenis_kelamin' => $jenis_kelamin,
            'tlp' => $tlp
        );
        // perintah update data ke database
        $this->m_data->update_data($where, $data,'tb_member');
        // di arahkan ke halaman admin member
        redirect(base_url(). 'admin/member');
    }
}

```

```

    }else{
        // jika validasi update tidak berhasil
        $id_member = $this->input->post('id_member');
        // kondisi data yang akan di ambil
        $where = array(
            'id_member' => $id_member
        );
        // perintah untuk mengambil data dari database
        $data['member'] = $this->m_data->edit_data($where,'tb_member')-
>result();
        // halaman di alihkan ke form edit member
        $this->load->view('admin/f_member_edit',$data);
    }
}

public function member_hapus($id_member)
{
    $where = array(
        'id_member' => $id_member
    );

    $this->m_data->delete_data($where,'tb_member');

    redirect(base_url(). 'admin/member');
}
// END CRUD Member
}

```

*Menambahkan perintah CRUD pada model data*

Selanjutnya kita akan menambahkan scripting untuk perintah input, update dan delete data pada database. Dengan scripting sebagai berikut :

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class M_data extends CI_Model{

    function cek_login($table,$where){
        return $this->db->get_where($table,$where);
    }

    // FUNGSI CRUD
    // fungsi untuk mengambil data dari database
    function get_data($table){

```

```

        return $this->db->get($table);
    }

    // fungsi untuk menginput data ke database
    function insert_data($data,$table){
        $this->db->insert($table,$data);
    }

    // fungsi untuk mengedit data
    function edit_data($where,$table){
        return $this->db->get_where($table,$where);
    }

    // fungsi untuk mengupdate atau mengubah data di database
    function update_data($where,$data,$table){
        $this->db->where($where);
        $this->db->update($table,$data);
    }

    // fungsi untuk menghapus data dari database
    function delete_data($where,$table){
        $this->db->delete($table,$where);
    }
    // AKHIR FUNGSI CRUD
}

```

Langkah selanjutnya kita akan membuat tampilan member, tambah member, update member pada **view/admin/**. Dengan scripting seperti berikut:

*Membuat UI member*

```

<!doctype html>
<html lang="en">
    <head>
        <!-- Required meta tags -->
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">

        <!-- Koneksi Bootstrap CSS -->
        <link href="<?php echo base_url('assets/bootstrap/css/bootstrap.min.css');?>"
rel="stylesheet">

        <title>Admin | Data Member</title>
    </head>
    <body>

```



```

<div class="container">
    <div class="row">
        <div class="col">
            <h1> Data Member</h1><br/>
            <a class="btn btn-
primary" href="<?php echo base_url(). 'admin' ?>" role="button">Kembali</a>
            <a class="btn btn-
primary" href="<?php echo base_url(). 'admin/member_tambah' ?>" role="button">Tamb
ah Data Member</a><br/>
            <table class="table table-striped">
                <thead class="text-center">
                    <tr>
                        <th Scope="col">No </th>
                        <th scope="col">Id Member</th>
                        <th scope="col">Nama</th>
                        <th scope="col">Alamat</th>
                        <th scope="col">Jenis Kelamin</th>
                        <th scope="col">Nomor Telepon</th>
                        <th scope="col">Action</th>
                    </tr>
                </thead>
                <tbody class="text-center">
                    <?php
                        $no = 1;
                        foreach($member as $m){
                            ?>
                            <tr>
                                <th scope="row"><?php echo $no++; ?></th>
                                <td><?php echo $m->id_member; ?></td>
                                <td><?php echo $m->nama; ?></td>
                                <td><?php echo $m->alamat; ?></td>
                                <td><?php echo $m->jenis_kelamin; ?></td>
                                <td><?php echo $m->tlp; ?></td>
                                <td class="text-center">
                                    <a href="<?php echo base_url(). 'admin/member_edit/' . $m-
>id_member; ?>" class="btn btn-warning btn-sm"> Edit Data Member </a>
                                    <a href="<?php echo base_url(). 'admin/member_hapus/' . $m-
>id_member; ?>" class="btn btn-danger btn-sm"> Hapus Data Member</a>
                                </td>
                            </tr>
                        <?php } ?>
                    </tbody>
                </table>

                <hr/>

```

```

        </div>
    </div>
</div>

<!-- Bootstrap -->
<script src="<?php echo base_url('assets/bootsrapt/js/bootstrap.min.js');?>"><
/script>
<!-- iCheck -->
</body>
</html>

```

Hasil tampilan :

| No | Id Member | Nama | Alamat | Jenis Kelamin | Nomor Telepon | Action |
|----|-----------|------|--------|---------------|---------------|--------|
|    |           |      |        |               |               |        |

Membuat UI tambah member

```

<!doctype html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Koneksi Bootstrap CSS -->
    <link href="<?php echo base_url('assets/bootstrap/css/bootstrap.min.css');?>"
rel="stylesheet">

    <title>Admin | Tambah Member</title>
</head>
<body>
<div class="container">
    <div class="row">
        <div class="col">
            <h1 class="text-center"> Form Tambah Data Member</h1><br/>
            <a class="btn btn-
primary" href="<?php echo base_url().'admin/member' ?>" role="button">kembali</a>
        </div>
    </div>
<br/>

```

```

        <form method="post" action="<?php echo base_url('admin/member_aksi') ?>" en
ctype="multipart/form-data">
        <div class="row">
            <div class="col">
                <div class="card">
                    <div class="card-body">
                        <div class="mb-3 row">
                            <label for="staticEmail" class="col-sm-2 col-form-
label">Id Member</label>
                            <div class="col-sm-10">
                                <input type="text" name="id_member" class="form-control" >
                                </div>
                                <?php echo form_error('id_member'); ?>
                            </div>
                        <div class="mb-3 row">
                            <label for="inputPassword" class="col-sm-2 col-form-
label">Nama</label>
                            <div class="col-sm-10">
                                <input type="text" class="form-
control" id="inputPassword" name="nama" >
                                </div>
                            </div>
                        <div class="mb-3 row">
                            <label for="inputPassword" class="col-sm-2 col-form-
label">Alamat</label>
                            <div class="col-sm-10">
                                <textarea type="text" class="form-
control" id="inputPassword" name="alamat"> </textarea>
                                </div>
                            </div>
                        <div class="mb-3 row">
                            <label for="inputPassword" class="col-sm-2 col-form-
label">Jenis Kelamin</label>
                            <div class="col-sm-10">
                                <select class="form-select" aria-
label="Default select example" name="jenis_kelamin">
                                    <option selected>Pilih Jenis Kelamin</option>
                                    <option value="L">Laki-Laki</option>
                                    <option value="P">Perempuan</option>
                                </select>
                            </div>
                        </div>
                        <div class="mb-3 row">
                            <label for="inputPassword" class="col-sm-2 col-form-
label">Nomor Telepon</label>

```

Hasil tampilan :

## Form Tambah Data Member

Membuat UI edit member

41 | Page

```

    <!-- Koneksi Bootstrap CSS -->
    <link href="<?php echo base_url('assets/bootstrap/css/bootstrap.min.css');?>"
rel="stylesheet">

    <title>Admin | Edit Member</title>
</head>
<body>
<div class="container">
    <div class="row">
        <div class="col">
            <h1 class="text-center"> Form Update Data Member</h1><br/>
            <a class="btn btn-
primary" href="<?php echo base_url().'admin/member' ?>" role="button">kembali</a>
        </div>
    </div>
    <br/>
    <form method="post" action="<?php echo base_url('admin/member_update') ?>"
enctype="multipart/form-data">
        <div class="row">
            <div class="col">
                <div class="card">
                    <div class="card-body">
                        <?php foreach($member as $m){ ?>
                            <div class="mb-3 row">
                                <label for="staticEmail" class="col-sm-2 col-form-
label">Id Member</label>
                                <div class="col-sm-10">
                                    <input type="text" name="id_member" class="form-
control" value="<?php echo $m->id_member; ?>" readonly >
                                </div>
                                <?php echo form_error('id_member'); ?>
                            </div>
                            <div class="mb-3 row">
                                <label for="inputPassword" class="col-sm-2 col-form-
label">Nama</label>
                                <div class="col-sm-10">
                                    <input type="text" class="form-
control" id="inputPassword" name="nama" value="<?php echo $m->nama; ?>" >
                                </div>
                            </div>
                            <div class="mb-3 row">
                                <label for="inputPassword" class="col-sm-2 col-form-
label">Alamat</label>
                                <div class="col-sm-10">

```

```

        <textarea type="text" class="form-
control" id="inputPassword" name="alamat"> <?php echo $m->alamat; ?></textarea>
    </div>
</div>
<div class="mb-3 row">
    <label for="inputPassword" class="col-sm-2 col-form-
label">Jenis Kelamin</label>
    <div class="col-sm-10">
        <select class="form-select" aria-
label="Default select example" name="jenis_kelamin">
            <option selected><?php echo $m->jenis_kelamin; ?></option>
            <option value="L">Laki-Laki</option>
            <option value="P">Perempuan</option>
        </select>
    </div>
</div>
<div class="mb-3 row">
    <label for="inputPassword" class="col-sm-2 col-form-
label">Nomor Telepon</label>
    <div class="col-sm-10">
        <input type="text" class="form-
control" id="inputPassword" name="tlp" value="<?php echo $m->tlp; ?>" >
    </div>
    <?php } ?>
</div>
    <input type="submit" class="btn btn-
success " value="Simpan Data Member">
    </div>
</div>
</div>

<!-- Bootstrap -->
<script src="<?php echo base_url('assets/bootsrapt/js/bootstrap.min.js');?>"><
/script>
<!-- iCheck -->
</body>
</html>

```

Hasil tampilan :

Form controls - Bootstrap v5.1 x Admin | Edit Member x localhost / 127.0.0.1 / codeignite x +

iter/admin/member\_edit/2021010001

## Form Update Data Member

[kembali](#)

|               |   |
|---------------|---|
| Id Member     | 2021010001  |
| Nama          | bagus abdul karim                                     |
| Alamat        | Kp. Ciapus 05/09, Ciapus, Banjaran, Kabupaten Bandung |
| Jenis Kelamin | L   |
| Nomor Telepon | 089654664597  |

[Simpan Data Member](#)

#### IV. Pengujian

Setelah semua Langkah, scripting sudah dilaksanakan Langkah selanjutnya kila lakukan pengujian pada fitur yang telah kita buat.

##### Tugas

1. Buatlah dokumen pengujian pada office word kemudian beri nama ["Pengujian Fitur CRUD member – {nama} {kelas}].
2. Capture setiap tampilan User Interface yang telah dibuat dan simpan pada dokumen pengujian. {halaman member, Halaman tambah member, Halaman edit member}
3. Capture setiap error yang terjadi (bila terjadi error) dan simpan pada dokumen pengujian .
4. Lakukan perbaikan pada kesalahan/error yang terjadi , serta catat dan simpan pada dokumen pengujian.
5. Dokumen pengujian diserahkan kepada guru yang bersangkutan

## BAB IV

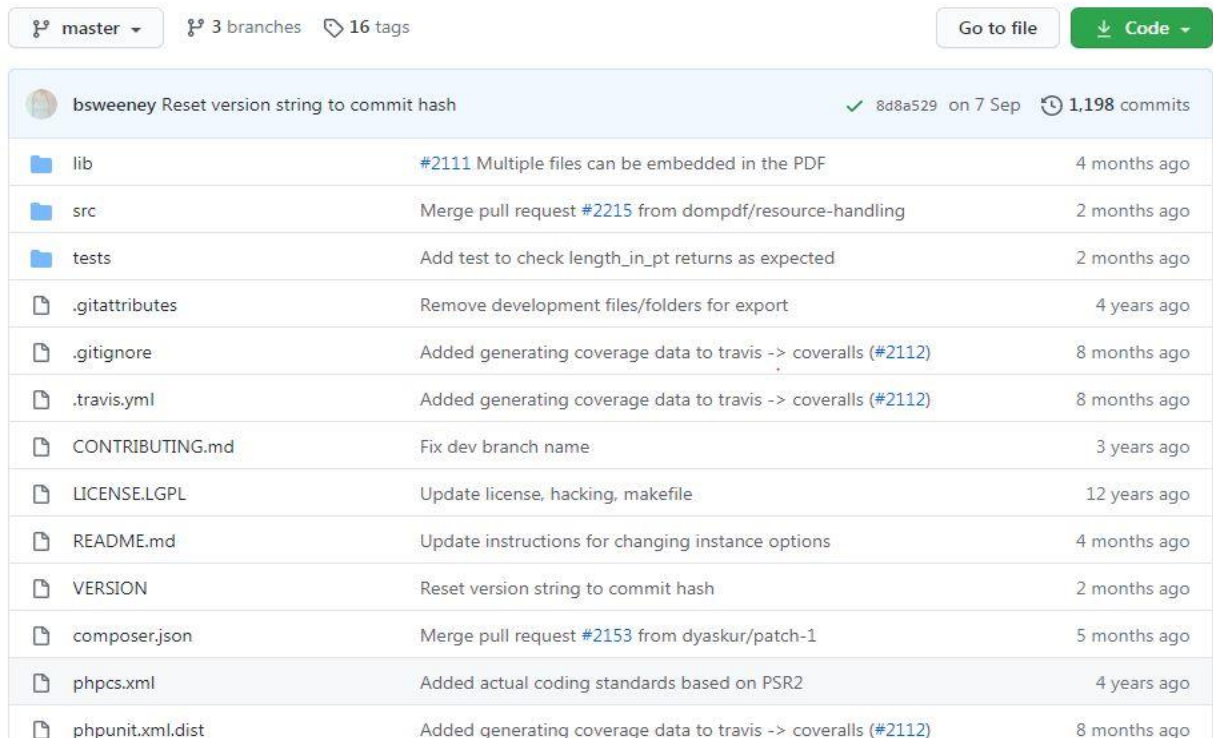
### Membuat Report/Laporan

- I. Tujuan
- II. Teori
- III. Praktikum

#### *Download Dompdf*

pada langkah ini kita akan download dompdfnya di link github yakni

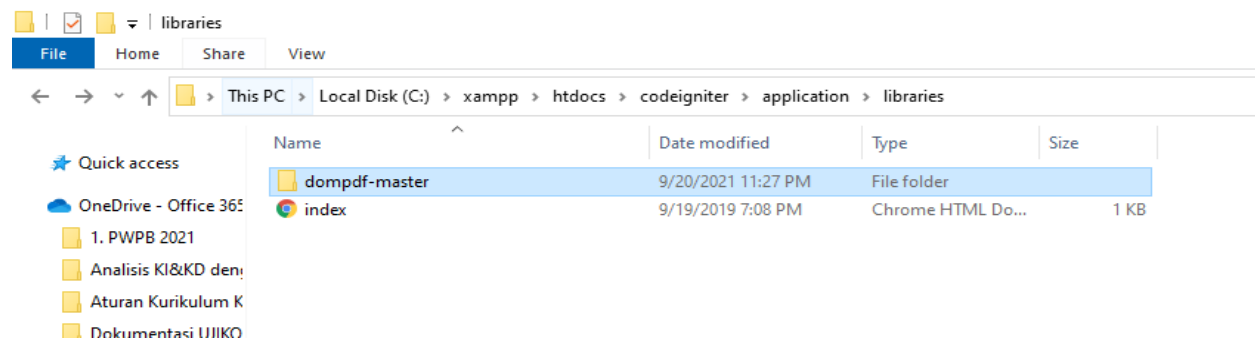
<https://github.com/dompdf/dompdf/>



| File             | Commit Message  | Time Ago     |
|------------------|---|--------------|
| lib              | #2111 Multiple files can be embedded in the PDF               | 4 months ago |
| src              | Merge pull request #2215 from dompdf/resource-handling        | 2 months ago |
| tests            | Add test to check length_in_pt returns as expected            | 2 months ago |
| .gitattributes   | Remove development files/folders for export                   | 4 years ago  |
| .gitignore       | Added generating coverage data to travis -> coveralls (#2112) | 8 months ago |
| .travis.yml      | Added generating coverage data to travis -> coveralls (#2112) | 8 months ago |
| CONTRIBUTING.md  | Fix dev branch name   | 3 years ago  |
| LICENSE.LGPL     | Update license, hacking, makefile                             | 12 years ago |
| README.md        | Update instructions for changing instance options             | 4 months ago |
| VERSION          | Reset version string to commit hash                           | 2 months ago |
| composer.json    | Merge pull request #2153 from dyaskur/patch-1                 | 5 months ago |
| phpcs.xml        | Added actual coding standards based on PSR2                   | 4 years ago  |
| phpunit.xml.dist | Added generating coverage data to travis -> coveralls (#2112) | 8 months ago |


#### *Extract Dompdf di folder libraries CodeIgniter*


setelah itu file dompdf yang kita download, extrack ke libraries CodeIgniter yang terletak di folder **application/libraries**







kita akan download autoload.inc.php dan load\_font.php untuk menjalankan lib dompdfnya. link source: <https://github.com/dompdf/utls>


 master ▾







 1 branch

 0 tags

Go to file

 Code ▾


 **jeremy-seipelt** and **bsweeney** my linter threw some errors and I thought I would ... 47ddd6d on 28 Aug 2019 9 commits


|  |  |               |
|--|--|---------------|
|  .gitattributes   | Add initial command-line font loader                                     | 5 years ago   |
|  .gitignore       | Remove unused ignore references  | 5 years ago   |
|  LICENSE          | Initial commit   | 5 years ago   |
|  README.md        | Initial commit   | 5 years ago   |
|  autoload.inc.php | Add autoloader used with packaged versions                               | 2 years ago   |
|  load_font.php    | my linter threw some errors and I thought I would update the master w... | 14 months ago |


README.md

lalu siapkan php-font-lib dan php-svg-lib nya tanpa ini autoload.inc.php tidak akan berjalan karena autoload memanggil lib ini,


1. php-font-lib, link source : <https://github.com/PhenX/php-font-lib>


 master ▾















 1 branch

 11 tags

Go to file

 Code ▾

 **mondrake** TravisCI tests up to PHP 7.4 (#74) ... ✓ ca6ad46 on 8 Mar 143 commits

|  |  |               |
|--|--|---------------|
|  maps             | Added support for custome encoding to build Adobe Font Metrics files ... | 9 years ago   |
|  sample-fonts     | Add support for format 12 cmap. Issue #40                                | 3 years ago   |
|  src/FontLib      | Fixes a problem with php 7.4   | 10 months ago |
|  tests/FontLib    | Update FontTest to use PHPUnit\Framework namespace.                      | 3 years ago   |
|  .gitattributes   | Fix tabs to spaces   | 8 years ago   |
|  .gitignore       | Remove .idea folder  | 8 years ago   |
|  .htaccess        | Changed the way WOFF fonts are loaded (uncompressed once and for all)    | 9 years ago   |
|  .travis.yml      | TravisCI tests up to PHP 7.4 (#74)                                       | 8 months ago  |
|  LICENSE          | Add license file (Fixes #13)   | 7 years ago   |
|  README.md        | Added example  | 4 years ago   |
|  bower.json       | Fix to Bower config to use many selector rather than a specific versi... | 7 years ago   |
|  composer.json    | TravisCI tests up to PHP 7.4 (#74)                                       | 8 months ago  |
|  index.php        | Add index.php to redirect to www/  | 8 years ago   |
|  phpunit.xml.dist | Add phpunit.xml.dist file.   | 5 years ago   |

2. php-svg-lib, link : <https://github.com/PhenX/php-svg-lib>

The screenshot shows a GitHub repository page for a pull request. At the top, it says 'bsweeney Merge pull request #38 from UCIS/patch-1' with a commit hash '16fd4d7' and the date 'on 10 Feb'. There are 78 commits. Below this, a list of files and their commit messages is shown:

| File           | Commit Message                           | Time Ago     |
|----------------|--|--------------|
| src            | Merge pull request #38 from UCIS/patch-1 | 9 months ago |
| tests/Svg      | Update the PHPUnit namespace             | 3 years ago  |
| .gitattributes | Initial commit                           | 7 years ago  |
| .gitignore     | Removal of the "gui" directory           | 6 years ago  |
| .travis.yml    | Test enhancement                         | 3 years ago  |
| COPYING        | Clarify project license                  | 3 years ago  |
| COPYING.GPL    | Clarify project license                  | 3 years ago  |
| README.md      | Add Travis and Coveralls badges          | 6 years ago  |
| composer.json  | Merge branch 'master' into patch-3       | 9 months ago |
| phpunit.xml    | Test enhancement                         | 3 years ago  |

Below the list, there is a section for 'README.md'.

setelah semua sudah disiapkan, kita akan extrack file tersebut didalam folder dompdf-master

Extract autoload.inc.php dan load\_font.php didalam folder dompdf-master

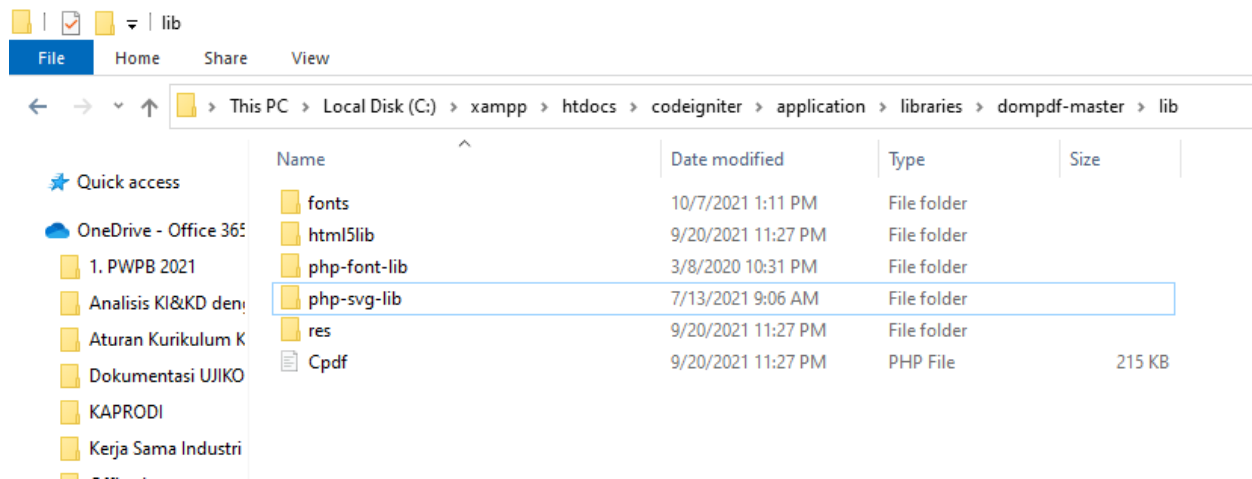
langkah ini kita akan extrack utils-master.zip, dan keluarkan isi nya yaitu autoload.inc.php dan load\_font.php, pindahkan ke dalam folder dompdf-master, atau lihat struktur pada folder berikut :

The screenshot shows a Windows File Explorer window with the address bar displaying the path: 'This PC > Local Disk (C:) > xampp > htdocs > codeigniter > application > libraries > dompdf-master'. The left sidebar shows 'Quick access' and 'OneDrive - Office 365'. The main pane shows a list of files and folders:

| Name            | Date modified      | Type         | Size  |
|-----------------|--------------------|--------------|-------|
| lib             | 9/20/2021 11:27 PM | File folder  |       |
| src             | 9/20/2021 11:27 PM | File folder  |       |
| autoload.inc    | 11/19/2020 8:34 PM | PHP File     | 2 KB  |
| composer.json   | 9/20/2021 11:27 PM | JSON File    | 2 KB  |
| CONTRIBUTING.md | 9/20/2021 11:27 PM | MD File      | 3 KB  |
| LICENSE.LGPL    | 9/20/2021 11:27 PM | LGPL File    | 24 KB |
| load_font       | 11/19/2020 8:34 PM | PHP File     | 7 KB  |
| phpcs           | 9/20/2021 11:27 PM | XML Document | 5 KB  |
| README.md       | 9/20/2021 11:27 PM | MD File      | 8 KB  |
| SECURITY.md     | 9/20/2021 11:27 PM | MD File      | 1 KB  |
| VERSION         | 9/20/2021 11:27 PM | File         | 1 KB  |

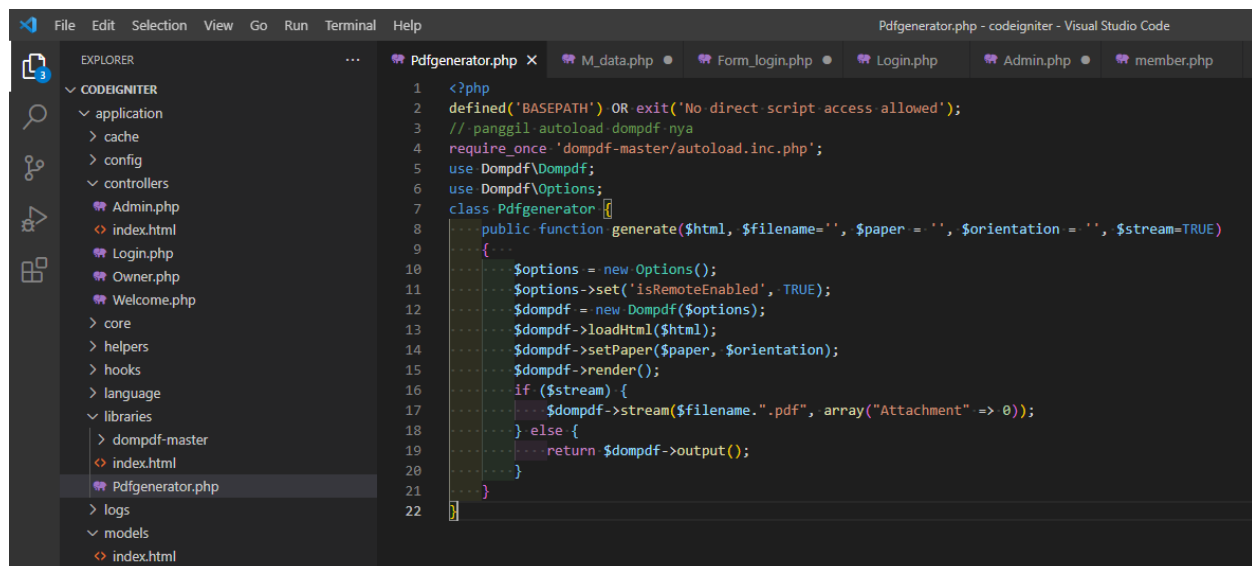
Extract php-font-lib dan php-svg-lib di dalam folder dompdf-master/lib

langkah ini kita akan extrack php-font-lib-master.zip dan php-svg-lib-master.zip, dan keluarkan isi nya dan rename folder tersebut menjadi php-font-lib dan php-svg-lib, pindahkan ke dalam folder dompdf-master/lib, atau lihat struktur pada folder berikut:



### Membuat Library Pdfgenerator.php

sekarang kita membuat library Pdfgenerator.php, untuk menjalankan dan memanggil dompdf yang terletak pada folder yang sama yaitu libraries, setelah itu letakan script di application/libraries/Pdfgenerator.php.



setelah kita membuat library untuk memanggil dompdfnya, kita akan membuat controller untuk memanggil library yang kita buat.

### Membuat function pdfview

Pada Langkah ini kita akan membuat fungsi pdfview pada controller admin untuk memanggil librarynya, dengan menambahkan script sebagai berikut:

```
// fungsi PDFView
public function pdfview_member()
{
    // panggil library yang kita buat sebelumnya yang bernama pdfgenerator
    $this->load->library('pdfgenerator');

    // title dari pdf
    $this->data['title_pdf'] = 'Laporan Penjualan Toko Kita';

    // filename dari pdf ketika didownload
    $file_pdf = 'laporan_member';
    // setting paper
    $paper = 'A4';
    //orientasi paper potrait / landscape
    $orientation = "portrait";

    $data['member'] = $this->m_data->get_data('tb_member')->result();
    $html = $this->load->view('admin/laporan_member',$data, true);

    // run dompdf
    $this->pdfgenerator->generate($html, $file_pdf,$paper,$orientation);
}
```

Setelah membuat controller, kita akan membuat file views.

### Membuat Views laporan member

pada langkah ini kita akan mencoba membuat file laporan\_member.php untuk mengisi data pdfnya dengan menggunakan html dan css, pada file views ini saya akan membuat laporan penjualan toko kita seperti pada tutorial dompdf yang kemarin, yang akan di generate ke pdf dengan dompdf, buat nama file laporan\_member.php , Letakan script ini pada folder application/views/admin/laporan\_member.php. contoh script seperti berikut:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title><?= $title_pdf;?></title>
        <style>
            #table {
                font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
                border-collapse: collapse;
                width: 100%;
```

```

    }

    #table td, #table th {
        border: 1px solid #ddd;
        padding: 8px;
    }

    #table tr:nth-child(even){background-color: #f2f2f2;}

    #table tr:hover {background-color: #ddd;}

    #table th {
        padding-top: 10px;
        padding-bottom: 10px;
        text-align: left;
        background-color: #4CAF50;
        color: white;
    }
</style>
</head>
<body>
    <div style="text-align:center">
        <h3> Report Data Member/Pengguna</h3>
    </div>
    <table id="table">
        <thead>
            <tr>
                <th>No.</th>
                <th>Id Member</th>
                <th>Nama</th>
                <th>Alamat</th>
                <th>Jenis Kelamin</th>
                <th>Telepon</th>
            </tr>
        </thead>
        <tbody>
            <?php
                $no = 1;
                foreach($member as $m){
            ?>
                <tr>
                    <td scope="row"><?php echo $no++; ?></td>
                    <td><?php echo $m->id_member; ?></td>
                    <td><?php echo $m->nama; ?></td>
                    <td><?php echo $m->alamat; ?></td>

```

```

        <td><?php echo $m->jenis_kelamin; ?></td>
        <td><?php echo $m->tlp; ?></td>
    </tr>
<?php } ?>
</tbody>
</table>
</body>
</html>

```

## V. Pengujian

Setelah semua Langkah, scripting sudah dilaksanakan Langkah selanjutnya kila lakukan pengujian pada fitur yang telah kita buat.

### Tugas

1. Buatlah dokumen pengujian pada office word kemudian beri nama [“Pengujian Fitur laporan – {nama} {kelas}”].
2. Capture setiap tampilan User Interface yang telah dibuat dan simpan pada dokumen pengujian. {halaman laporan member}
3. Capture setiap error yang terjadi (bila terjadi error) dan simpan pada dokumen pengujian .
4. Lakukan perbaikan pada kesalahan/error yang terjadi , serta catat dan simpan pada dokumen pengujian.
5. Dokumen pengujian diserahkan kepada guru yang bersangkutan