

**PANDUAN MEMBANGUN SEBUAH  
*WEBSITE* BESERTA STUDI KASUS  
MENGUNAKAN BAHASA PEMOGRAMAN  
PHP DENGAN *FRAMEWORK*  
*CODEIGNITER* DAN *BOOTSTRAP*  
SERTA TERINTEGRASI OLEH GOOGLE  
MAPS API**

## **Apa yang anda harus ketahui**

Sebelum anda memulai mengikuti panduan/tutorial ini anda harus sudah mempunyai setidaknya pengetahuan dasar tentang pemograman web seperti, HTML,CSS,JavaScript,dsb. Karena dalam panduan ini penulis tidak memuat detail semuanya.

*Output* dari buku ini penulis akan membangun sebuah aplikasi berbasis *web* dengan menggunakan bahasa pemograman PHP dan *Framework* Codeigniter pada sisi *back-end* dan Bootstrap pada sisi *front-end* serta metode pengembangan menggunakan R&D (*Research And Development*) dan di integrasikan dengan menggunakan Google Map API.

# BAB 1

## Pengenalan Dasar

---

Dalam buku panduan/tutorial ini penulis akan membangun sebuah aplikasi website dinamis dan responsif dengan studi kasus pelelangan yang di kembangkan dengan metode R&D (*Research And Development*) akan di bangun menggunakan bahasa pemograman PHP pada *Back-end* dan dibantu *framework Codeigniter* serta untuk *front-end* menggunakan *Bootstrap* dengan sedikit tambahan *jQuery*. dan *Database MYSQL* serta *library GoogleMaps API*.

Sebelum kita ke tahap pemangunan sebuah *web*, ada beberapa hal yang perlu diketahui dan dilakukan, seperti pengetahuan dasar apa yang akan kita gunakan, penginstallan aplikasi-aplikasi yang dibutuhkan dan aplikasi-aplikasi pendukung dalam membangun sebuah *project web*.

## 1.1 Pengenalan PHP



Gambar 1.1 PHP

Sejarah awal mula PHP kala itu kepanjangan dari *Personal Home Page* (Situs Personal) ditemukan pertama kali oleh Rasmus Lerdorf pada tahun 1995. Saat itu PHP masih berwujud berupa sekumpulan kode/skrip yang digunakan untuk mengolah data formulir dari *web*.

Ditahun berikutnya Rasmus merilis skripnya secara umum dan menamainya PHP/FI, dengan perilsan tersebut skrip nya menjadi sumber terbuka untuk semua orang khususnya para programmer/pemrogram sehingga banyak yang tertarik untuk ikut mengembangkannya. Tahun ke tahun skripnya banyak digunakn dan dikembangkan sampai pada tahun 1997 ada sebuah perusahaan bernama Zend menulis ulang skrip PHP Rasmus menjadi lebih baik dan lebih cepat daripada sebelumnya. Kemudia pada tahun 1998

pada bulan Juni perusahaan tersebut merilis skrip PHP dengan gaya baru dan diresmikan menjadi PHP 3.0 dan merubah kepanjangannya menjadi *PHP : Hypertext Preprocessor*.

Pada pertengahan tahun 1999 Zend meng *upgrade* lagi PHP dan merilisnya dengan nama PHP 4.0 pada versi ini banyak sekali programmer web menggunakannya karena lebih kompleks, cepat dan memiliki stabilitas yang tinggi kala itu. Sampai pada Juni 2004 Zend kembali rilis dengan PHP 5.0 dalam versi ini banyak sekali perubahan besar, karena dalam versi ini Zend memasukan model pemrograman berorientasi objek atau yang lebih dikenal dengan OOP, jadi semua data dan fungsi di dalam paradigma ini dibungkus kedalam objek-objek atau kelas. Dan server web bawaan ditambahkan pada versi PHP 5.4 untuk mempermudah *Developer* menjalankan skrip PHP tanpa harus menginstall *software server*. Hingga saat ini versi terbaru dari php yaitu PHP 7.4.

### **1.1.1 Penjelasan PHP**

Pada intinya PHP merupakan bahasa pemrograman untuk membuat web yang bersifat *server-side scripting*. PHP akan memudahkan anda dalam membuat halaman web yang sifatnya dinamis. PHP dapat digunakan dalam berbagai macam jenis sistem operasi, Hingga kini, PHP sudah mendukung penuh *Object Oriented*

*Programming (OOP)*, integrasi XML, mendukung semua ekstensi terbaru *MySQL*, pengembangan *web service* dengan *SOAP & REST*, serta ratusan peningkatan lainnya dibandingkan dengan versi terdahulunya. *PHP* juga bersifat *open source* sehingga setiap orang dapat menggunakannya secara gratis.

PHP tentunya berbeda dengan HTML dan CSS perbedaanya adalah proses dari PHP itu sendiri. HTML adalah bahasa statis yang jika kita ingin merubah isinya maka yang harus dilakukan adalah, membuka file-nya terlebih dahulu, lalu menambahkan isi kedalam file tersebut. berbeda dengan PHP. Bagi anda yang pernah menggunakan wordpress atau joomla yang dibangun dengan PHP tentunya, saat akan menambahkan isi/konten kedalam website, anda tinggal masuk kedalam halaman admin, kemudian pilih artikel untuk membuat halaman baru. Artinya hal ini, seorang *user* tidak berhubungan langsung dengan scriptnya. Sehingga seorang pemula sekalipun dapat menggunakan aplikasi seperti itu.

### **1.1.2 Keunggulan PHP**

Kenapa penulis lebih memilih PHP? Kenapa tidak bahasa pemrograman yang lain? Selain karena cocok untuk pemula, PHP juga memiliki kelebihan dari bahasa pemrograman yang lain diantaranya adalah :

1. Tentunya yang pertama ialah gratis,
2. PHP juga merupakan *Cross Platform* yang artinya dapat digunakan di berbagai sistem operasi seperti *Linux, Windows* dan *Mac OS*.
3. Dalam sisi pemahaman PHP merupakan bahasa skrip yang paling mudah untuk pemula karena memiliki referensi yang banyak.
4. PHP telah banyak mendukung *database* seperti, dBase dBase, Empress Empress, FilePro (read-only) FilePro (read-only) Hyperwave, IBM DB2, Informix, Ingres, InterBase, FrontBase mSQL, Direct MS-SQL, MySQL MySQL, ODBC, Oracle (OCI7 and OCI8), Ovrimos, PostgreSQL SQLite, Solid, Sybase, Velocis, Unix dbm, dll.
5. Dalam PHP juga sudah dapat membuat dokumen-dokumen text seperti, word, excel, PDF, zip, xml, dll. Serta dapat menciptakan gambar dan flash serta banyak lagi.
6. PHP juga memiliki banyak *framework* yang dapat mempermudah dan mempersingkat penulisan skrip dari PHP seperti contohnya, *Framework Cake PHP, Codeigniter, Laravel, PRADO, Yii, ZooP, QPHP, Symfony, Zend Framework*, dan banyak lagi.

### 1.1.3 Contoh Sintaks PHP

Berikut penulis berikan contoh skrip/sintaks sederhana dari PHP yang disisipkan di HTML,

```
<html>

    <head>

        <title>test PHP </title>

    </head>

    <body>

        <?php

            For ($r = 1; $r <= 100; $r++)

            {

                Echo "<li>Halo saya orang ke - $r</li> ";

            }

        ?>

    </body>

</html>
```



Pada sintaks diatas akan menapilakan *output* perulangan berupa tulisan Halo saya orang ke - 1 sampai dengan Halo saya orang ke – 100. Seperti gambar berikut :

- Halo saya orang ke - 1
- Halo saya orang ke - 2
- Halo saya orang ke - 3
- Halo saya orang ke - 4
- Halo saya orang ke - 5
- Halo saya orang ke - 6
- Halo saya orang ke - 7
- Halo saya orang ke - 8
- Halo saya orang ke - 9
- Halo saya orang ke - 10
- Halo saya orang ke - 11
- Halo saya orang ke - 12
- Halo saya orang ke - 13
- Halo saya orang ke - 14
- Halo saya orang ke - 15
- Halo saya orang ke - 16
- Halo saya orang ke - 17
- Halo saya orang ke - 18
- Halo saya orang ke - 19
- Halo saya orang ke - 20
- Halo saya orang ke - 21
- Halo saya orang ke - 22
- Halo saya orang ke - 23
- Halo saya orang ke - 24
- Halo saya orang ke - 25
- Halo saya orang ke - 26
- Halo saya orang ke - 27
- Halo saya orang ke - 28
- Halo saya orang ke - 29
- Halo saya orang ke - 30
- Halo saya orang ke - 31
- Halo saya orang ke - 32
- Halo saya orang ke - 33
- Halo saya orang ke - 34
- Halo saya orang ke - 35
- Halo saya orang ke - 36

Gambar 1.2 Contoh Sintaks PHP

## 1.2 Pengenalan jQuery

Penemu pertamanya adalah John Resig dalang dari pembuatan jQuery, hasil karyanya ini pertama kali di umumkan di NYC BarCamp di awal tahun 2006. Di websitenya sendiri dia menulis skrip, dia membuat jQuery karena ketidak puasanya dengan *library* yang saat itu tersedia dia merasa *framework-framework* saat itu tidak membantunya dengan cepat.

Kemudian para *developer* datang kepadanya untuk membantu menyempurnakan *library* yang dibuat oleh John, dan saat itu berhasil dan rilis pertama dari jQuery versi 1.0 di tahun 2006. Sejak itu, jQuery telah banyak berkembang. dan terus di *upgrade* kemampuannya dengan rilis versi 1.7.1 dan telah mempunyai *plug-in* / ekstensi yang begitu banyak.

### 1.2.1 Penjelasan jQuery



Gambar 1.3 jQuery

jQuery adalah *library* dari sebuah Javascript, sebelumnya jika anda kurang memahami apa itu Javascript anda dapat mencari tau lebih detailnya di internet. Begitu banyak para *developer web* menggunakan *library* dari javascript ini, karena penulisan skrip

codingnya yang pendek dan mudah dimengerti itu sebabnya jQuery memiliki slogan “*write less, do more*”.

jQuery dibuat untuk memperingkas skrip-skrip dari javascript. JQuery merupakan *library* yang cepat dan ringan untuk mengatasi dokumen HTML, Event, membuat sebuah *motion* atau animasi dan interaksi ke ajax, ajax disini merupakan *asynchronous javascript and XML* memproses mengirim dan menerima data dari server tanpa harus mereload keseluruhan halaman web kita, ini juga yang nanti akan kita implementasikan pada web yang akan kita bangun. jQuery adalah *library* yang membuat sebuah aplikasi web dari sisi *client*, tidak terlihat sebagai skrip JavaScript biasa, yang harus secara eksplisit disisipkan pada sebuah dokumen web.

### ***1.2.2 Keunggulan jQuery***

Kenapa penulis menggunakan jQuery? Karena, jQuery memiliki keunggulanya sendiri diantaranya adalah :

1. Menyederhanakan penulisan Javascript biasa.
2. AJAX : Mengambil informasi dari server tanpa me-refresh seluruh halaman.
3. Mengatasi masalah kompatibilitas antar-browser.
4. Kemudahan mengakses elemen-elemen HTML.

5. Memanipulasi elemen HTML,CSS dan Penanganan event HTML
6. Efek-efek javascript dan animasi.

### ***1.2.3 Contoh Sintaks jQuery***

Berikut penulis berikan contoh skrip/sintaks *motion* sederhana dari jQuery yang disisipkan di HTML tapi sebelumnya anda harus mendownload terlebih dahulu jquery.js di website resminya <https://jquery.com/download/> , copy semua file dan paste di *text editor* anda, lalu *save* dan beri nama jquery.js. Jadi ketika anda ingin menggunakan file javascript jquery.js ini anda harus menyisipkannya ke skrip javascript anda,

```
<script type="text/javascript" src="jquery.js"></script>
```

Berikut contohnya.

```
<html>

<head>

<script type="text/javascript" src="jquery.js">

</script>
```

```
<script type="text/javascript"> $(document).ready(function(){  
  
    $(".btn1").click(function(){  
  
        $(".p").hide(1000);  
  
    });  
  
    $(".btn2").click(function(){  
  
        $(".p").show(1000);  
  
    });  
  
});  
  
</script>  
  
</head>  
  
<body>  
  
<p>Halo bro!</p> <button class="btn1">Sembunyikan</button>  
  
<button class="btn2">Tampilkan</button>  
  
</body>  
  
</html>
```

Pada sintaks diatas akan menampilkan duabuah *button*, jika anda mengklik sembunyikan maka tulisa hello bro akan disembuyikan dengan *effect motion* jika mengklik tampilkan maka sebaliknya,

Halo bro!

Gambar 1.4 Contoh Sintaks jQuery

## 1.3 Pengenalan Framework

Jika ada yang bertanya-tanya kenapa harus menggunakan *framework*? Lalu jika menggunakan *framework* kenapa harus *codeigniter* (CI), salah satu alasanya ialah begitu banyak kemudahan dalam membangun sebuah *web* menggunakan *framework* membuat skrip menjadi singkat dan praktis. Dalam *framework* juga sudah tersedia struktur pengaplikasian yang baik seperti,

***standard coding***, standar penulisan skrip/codingan yang harus diikuti programmer.

***best practice***, kumpulan aksi yang telah teruji oleh para programmer *expert*.

***design pattern***, teknik-teknik yang ada dalam *best practice*.

dan ***common function***, fungsi/library (pustaka) yang umum yang sering dikembangkan dalam membangun sebuah sistem/aplikasi.

Dengan bantuan *framework* ini kita dapat langsung fokus pada bisnis proses / latar belakang masalah yang dihadapi dalam membangun sebuah *web* tanpa memikirkan strukturnya, standard penulisan skrip/coding,dsb. Intinya *framework* ini dapat diartikan kumpulan dari fungsi/prosedur dan kelas-kelas untuk tujuan mempercepat proses pengembangan aplikasi, menyeragamkan cara kita dalam mengimplementasikan sebuah skrip/codingan.

### ***1.3.1 Codeigniter***



Gambar 1.5 *Codeigniter*

*Codeigniter* merupakan *framework* dari *Back-end*. Ditemukan pertama kali oleh seseorang yang memiliki banyak bakat bukan hanya menjadi programmer saja tetapi ia juga seorang yang menyukai music dan photographer, ia adalah Rick Ellis pada tanggal 28 Februari 2006, saat itu ia sedang membuat penelitian kecil dan berhasil serta mampu membuat sebuah *framework* PHP yang memiliki ukuran kecil, ringan dan memenuhi banyak fitur untuk pengaplikasian pada bahasa pemograman PHP. Karena sifatnya yang *opensource* pada tahun 2014 *codeigniter* telah banyak



dikembangkan, dan pengembangan itu dilakukan oleh British Columbia Institute of Technology (BCIT) hingga sampai saat ini versi stabil yang di kembangkan adalah versi 3.1.11.

Mengapa memilih *codeigniter* sebagai *framework* dalam pembangunan aplikasi? Jawabanya salah satunya adalah karena merupakan aplikasi yang *opensource* dengan berupa model MVC (Model,View,Controller) dapat membangun sebuah *web* menjadi dinamis dengan bahasa pemograman PHP. Referensi yang begitu banyak menjadi alasanya juga.

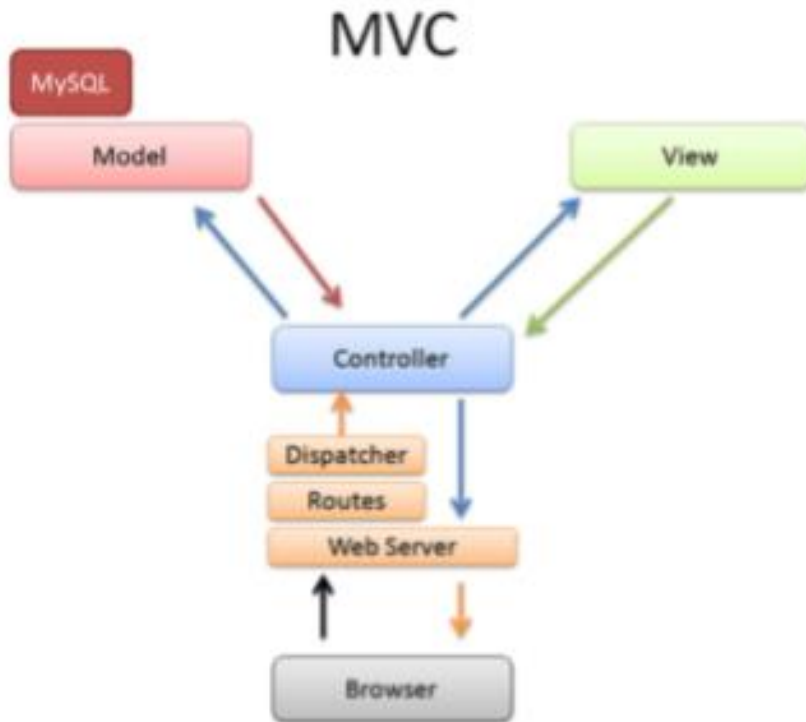
*Codeigniter* membantu para *developer* untuk mengerjakan aplikasi menjadi cepat daripada menulis skrip/codingan dari awal, *framework* ini memiliki library yang dapat mempermudah dalam pengembangan memuat file-file yang berekstensi PHP yang menyediakan *class* seperti *libraries*, *helpers*, *plugins*, dan lainnya ini sendiri lebih banyak orang dengan model MVC.

### **1.3.2 Model MVC**

MVC (Model,View,Controller) merupakan konsep dasar yang harus diketahui. MVC ini merupakan sebuah *pattern* atau teknik pemograman yang memisahkan bagian alur logika (proses), penyimpanan data (data), dan antarmuka aplikasi (desain). Berikut adalah penjelasan dari masing-masing komponen model :

1. **Model**, ini merupakan yang berhubungan langsung dengan data atau yang interaksi langsung dengan *database/webservice*. Disini tempat merepresentasikan struktur data. Biasanya dalam model berisi codingan fungsi seperti tambah data, update data, menampilkan data dan menghapus data. berhubungan langsung dengan *query-query* SQL yang nantinya akan masuk ke *database*.
2. **View**, ini merupakan yang berhubungan langsung dengan tampilan antarmuka/*user interface* isinya dapat berupa skrip HTML,CSS,Javascript,dsb. Dibagian ini berisi variabel-variabel yang berisi data yang siap untuk ditampilkan ke *user*. Di dalam *view* ini hanya di khususkan untuk menampilkan data yang telah di proses oleh controller dan model.
3. **Controller**, ini merupakan yang menjembatani antara data-data dengan *view*. Bisa dikatakan *controller* merupakan otak dari berjalanya aplikasi. Bagian ini lah kelas-kelas dan fungsi-fungsi di proses dari permintaan *view* kedalam struktur data yang ada di dalam model. Controller mengerjakan proses logika, menerima input dari pengguna dan menginstruksikan model dan *view* untuk melakukan aksi.

Berikut adalah ilustrasi dari proses model MVC :



Gambar 1.6 Ilustrasi MVC

### 1.3.3 Keunggulan *Codeigniter*

Adapun keunggulan *codeigniter* daripada *framework* yang lain diantaranya adalah :

1. Performanya yang cepat serta ringan dan mudah di implementasikan.
2. Dokumentasi yang lengkap dan bagus setiap instalasi terdapat *user guide* serta sedikit/minimnya konfigurasi. Tidak perlu banyak konfigurasi hanya seperti sedikit konfigurasi pada folder config.
3. *Compatible* dengan semua hosting serta mendukung banyak *database*.
4. Banyaknya referensi.

Untuk installasi *Codeigniter* anda dapat langsung mengunjungi *website* resminya di, <https://codeigniter.com/> download, setelah di download ekstrak file *codeigniter* dan pindahkan atau salin file ke tempat server anda, misal ( C:\xampp\htdocs\lelang-pgd ).

## 1.4 Pengenalan Bootstrap



Gambar 1.7 Bootstrap

Bootstrap merupakan *framework* dari *front-end*, Perancangnya adalah dua orang programmer Twitter yaitu Mark Otto dan Jacob Thornton. Ketika itu mereka membuatnya dengan alasan untuk mempermudah pekerjaan mereka sehingga mereka membuat sebuah *library front-end* yang berisi *source HTML, CSS, plugin-plugin Javascript, icon*, dan semua yang mereka sukai. Ketika itu mereka memberi nama *framework* itu Twitter Blueprint.

Setelah beberapa bulan pengembangan, banyak *developer* Twitter yang lain ikut berkontribusi sehingga rilislah pertama kali pada tanggal 19 Agustus 2011. Tahun ke tahun bulan ke bulan mereka semua mulai mengembangkan sedikit demi sedikit pada 31 Januari 2012 rilis bootstrap versi 2 dengan penambahan *system layout* dua belas kolom grid sehingga tampilanya menjadi responsif. 19 Agustus 2013 mereka kembali rilis dengan bootstrap versi 3 menggunakan gaya desain lebih flat dan pendekatan pada mobile. Dan pada 18 Januari 2018 rilislah bootstrap versi 4 versi paling stabil daripada sebelumnya, lebih banyak menyediakan kontrol-kontrol.

#### **1.4.1 Penjelasan Bootstrap**

Jadi pada intinya Bootstrap ini merupakan *library* (pustaka) yang berisi kumpulan fungsi-fungsi dari bahasa Cascading Style Sheet (CSS) yang dibangun khusus untuk pembuatan atau pengembangan pada sisi *front-end* dari sebuah *website*. Didalam *library* tersebut terdapat berbagai jenis file yang diantaranya HTML, CSS, dan Javascript. Sehingga jika anda tidak begitu banyak memahami CSS/Javascript Bootstrap ini adalah jawabanya karena di dalamnya telah dirangkas menjadi sebuah kerangka yang siap digunakan.

### 1.4.2 Keunggulan Bootstrap

Kenapa penulis memilih Bootstrap? Kenapa tidak membangun *front-end* nya sendiri? Selain karena cocok untuk pemula, Bootstrap ini juga memiliki kelebihan diantaranya adalah :

1. Untuk efisiensi dan menghemat waktu, karena didalam bootstrap sudah menyediakan banyak *library* yang membuat sebuah *website* menjadi enak di pandang tampilanya *clean*, modern dan responsif.
2. Fleksibel, dapat digunakan sesuka hati. Seperti anda memodifikasi menyesuaikan dengan apa yang anda butuhkan.
3. Dilengkapi dengan *Javascript* sehingga antarmuka lebih *powerful* serta sangat ringan dan terstruktur.
4. Dan masih banyak keuntungan lainnya pada bootstrap ini dalam membangun sebuah *website*.

Untuk cara penggunaan bootstrap, penulis akan memberikan metode secara *offline* dengan cara mengunjungi terlebih dahulu *website* resminya bootstrap untuk mendownload kumpulan *library* nya melalui link berikut, <https://getbootstrap.com/> setelah terdownload, ekstrak file lalu pindahkan semua filenya kedalam folder kerja anda. Jika anda bingung dengan metode yang

saya berikan anda dapat mencarinya lebih lengkap di internet, atau

<https://getbootstrap.com/docs/4.3/getting-started/download/>



## 1.5 Pengenalan MySQL



Gambar 1.8 MySQL

Pada awalnya *MySQL* diciptakan oleh Michael Monty Widenius yang biasa dikenal dengan Monty seorang programmer asal swedia. Ketika itu monty bekerja pada perusahaan bernama TcX, pada 1994 TcX mengembangkan aplikasi berbasis web dengan menggunakan sistem *database* UNIREG, tapi ketika itu *database* UNIREG dianggap tidak cocok untuk aplikasi web yang dinamis. Beralihlah ketika itu TcX menggunakan mSQL (miniSQL), namun versi ini masih memiliki banyak sekali kekurangan. Dengan mencoba memperbaiki kekurangan dari mSQL Monty menghubungi David Hughes (programmer yang mengembangkan mSQL).

Pada Mei 1995 Montys, David, dan TcX berhasil merancang dan membuat konsep sistem *database* baru RDBMS (*Relation Database Management System*) dengan menggabungkan UNIREG dan mSQL ketika itu dua *database* sistem tersebut *opensource* sehingga mereka dapat mengembangkannya dan berhasil merilisnya secara gratis. Mereka menamai *database* itu dengan nama MySQL. Pada tahun itu juga perusahaan TcX merubah namanya menjadi MySQL AB (Aktiebolag) dan terus mengembangkannya hingga sampai saat ini.

### **1.5.1 Penjelasan MySQL**

Pada initinya *MySQL* merupakan sistem manajemen basis data *RDBMS multithread, multi-user*, dengan jutaan pengguna instalasi di seluruh dunia. Semua orang bebas menggunakan *MySQL*, namun tidak boleh di jadikan produk turunan salah satu konsep utama dalam database sejak lama yaitu *Structured Query Lenguage SQL, Database* ini dibuat untuk keperluan sistem *database* yang cepat, handal dan mudah digunakan.

MySQL juga memiliki banyak fungsi seperti diibaratkan sebagai wadah yang menyimpan data *string* (*text based*) data. Gambar, media, file, file audio, dll.

### 1.5.2 Keunggulan MySQL

Kenapa penulis memilih MySQL untuk sistem *database* nya? Kenapa tidak menggunakan *database* yang lain? Selain karena cocok untuk pemula, MySQL ini juga memiliki kelebihan diantaranya adalah :

1. Sudah pasti Gratis berlisensi GPL (*General Public License*) serta *multiplatform*.
2. *Multiuser* dapat digunakan beberapa pengguna dalam waktu bersamaan tanpa mengalami konflik, serta memiliki kecepatan dalam menangani query.
3. Memiliki tipe data yang beragam, operator dan fungsi – fungsi yang lengkap, seperti *select, where*, dll.
4. Memiliki struktur tabel yang fleksibel dalam menangani perintah-perintah seperti *alter table*, dll. Serta dapat di integrasikan dengan Hosting.
5. Dapat mendeteksi pesan kesalahan, serta spesifikasi *hardware* yang rendah karena hemat *resource memory* dibandingkan dengan *database* lain.

6. Banyak terintegrasi oleh bahasa pemrograman seperti, PHP,Python,Net,Java,dll.

Untuk penginstallan MySQL penulis sudah mempersiapkannya dengan bantuan menggunakan server yang berdiri sendiri yaitu dengan menggunakan Aplikasi XAMPP, yang nanti akan di jelaskan di bab selanjutnya.

## **1.6 Pengenalan Google Maps API**



Gambar 1.9 Google Maps API

Google Maps sendiri merupakan produk dari perusahaan raksasa google yang di rilis 8 februari 2005, memungkinkan kita semua dapat melihat peta, gambar, lokasi, bangunan, jalan. dll. secara visual diseluruh belahan dunia ini.

Nah sekarang apa itu Google Maps API?

Google Maps API merupakan *library* khusus yang disediakan Google Maps sendiri untuk para *developer* atau programmer dalam mengembangkan sistem/aplikasinya. API disini adalah kepanjangan dari *Application Programming Interface* dimana semua aplikasi yang menggunakan data secara *realtime* selalu menggunakan API. Google Maps sendiri memiliki 5 macam API yang memang ditujukan untuk para *developer* dengan Bahasa program dan platform yang berbeda-beda diantaranya yaitu :

1. **Google Map Android API**, merupakan API yang dikhususkan untuk platform yang mempunyai OS Android, api ini banyak digunakan seperti pada aplikasi ojek *online* yang sedang marak saat ini, memiliki lokasi yang *realtime*. Untuk lebih jelasnya anda dapat melihatnya di link berikut, <https://developers.google.com/maps/documentation/android-sdk/intro>
2. **Google Map JavaScript API**, API yang sering kita temui banyak diterapkan di *website-website* yang menggunakan fitur lokasi atau tempat. Untuk lebih jelasnya anda dapat melihat di link

berikut,

<https://developers.google.com/maps/documentation/javascript/tutorial>

3. **Google Map Geocoding API**, API ini digunakan untuk mengkonversi tempat atau lokasi menjadi data berupa array yang digunakan untuk kebutuhan tertentu. Untuk lebih jelasnya anda dapat melihat di link berikut, <https://developers.google.com/maps/documentation/geocoding/start>
4. **Google Map Direction API**, merupakan API yang dibuat untuk memberikan navigasi arah terutama pada transportasi darat. Untuk lebih jelasnya anda dapat melihat di link berikut, <https://developers.google.com/maps/documentation/directions/start>
5. **Google Map Place API WebService**, merupakan API yang sama yang digunakan oleh google maps & google plus sendiri dengan menampilkan 100juta bisnis dan tempat rekomendasi serta menarik yang sering diperbarui oleh pemilik akun google plus yang terferivikasi. Untuk lebih jelasnya anda dapat melihat di link berikut, <https://developers.google.com/places/web-service/intro>

Tetapi kabarnya saat ini google maps api sudah tidak gratis artinya sudah berbayar lebih jelasnya anda dapat melihat di link berikut, <https://cloud.google.com/maps-platform/pricing>

Tetapi tenang untuk pembangunan aplikasi yang akan kita bangun saya sudah menyediakan *library* Google Map nya secara gratis walaupun tidak memiliki fasilitas yang lebih, kalian dapat mendownloadnya di github penulis berikut link nya, <https://github.com/Rojasqifadilla/googleMapAPI> untuk penerapan nya nanti akan penulis jelaskan apa bab selanjutnya.

## BAB 2

### Persiapan Tools

---

Setelah anda mengetahui materi pengenalan dasar yang akan kita gunakan dalam pembangun *website* ini, anda juga harus mengetahui rekomendasi *tools* apa saja yang akan kita gunakan dalam membangun *website* ini.

#### 2.1 XAMPP



Gambar 2.1

XAMPP merupakan *software*/perangkat lunak komputer yang mendukung banyak sistem operasi dengan kompilasi beberapa program, yang di jadikan penamaan XAMPP itu sendiri yaitu,



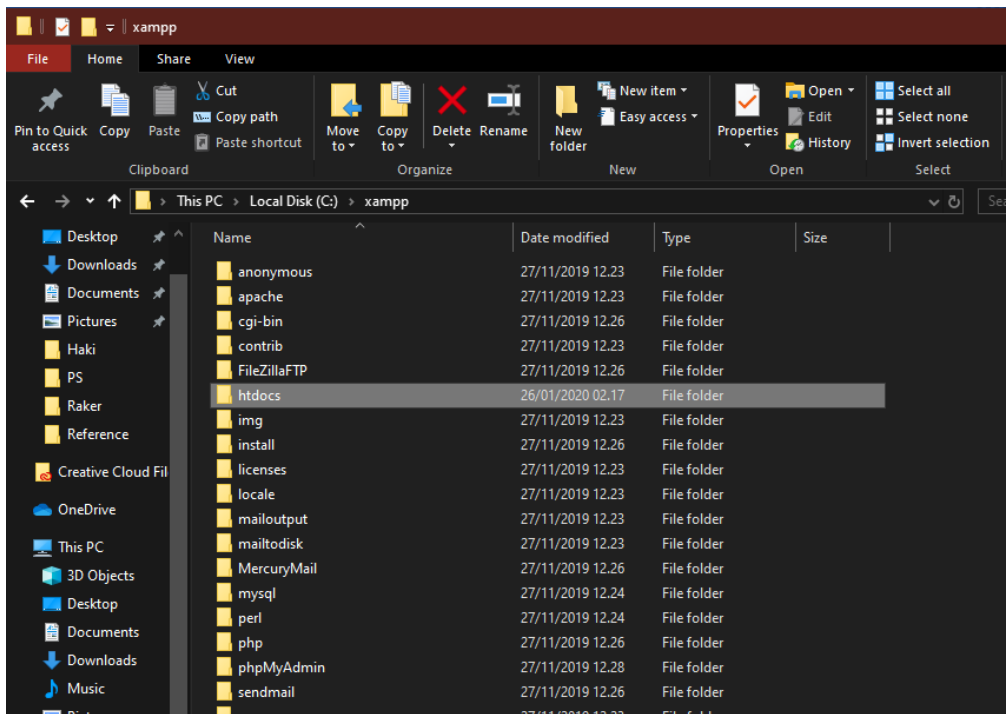
1. **X**, dapat diartikan sebagai *cross platform* yang artinya banyak sistem operasi yang dapat menggunakannya.
2. **A**, dapat diartikan sebagai Apache merupakan program web server yang bersifat gratis dan *opensource*, apache disini bertugas untuk menghasilkan halaman web yang benar ke user berdasarkan skrip PHP yang telah dibangun sebelumnya oleh programmer.
3. **M**, dapat diartikan sebagai MySQL program database server, digunakan untuk membuat dan mengelola *database* beserta isinya didalamnya seperti, menambahkan, mengubah, menghapus data yang berada pada *database*.
4. **P**, dapat diartikan sebagai bahasa pemrograman PHP, yang bersifat *server-side scripting*, memungkinkan anda untuk membuat *website* dinamis.
5. **P**, dapat diartikan sebagai bahasa pemrograman Perl yang sering digunakan untuk memenuhi berbagai macam kebutuhan.

Jadi pada intinya XAMPP ini berfungsi sebagai server lokal yang berdiri sendiri. Yang didalamnya terdiri atas Apache HTTP server, *database* MySQL, serta penerjemah dari bahasa pemrograman PHP dan Perl.

### 2.1.1 Bagian Penting pada XAMPP

Mengenal bagian penting yang sering digunakan pada umumnya, diantaranya adalah :

#### 1. Htdocs

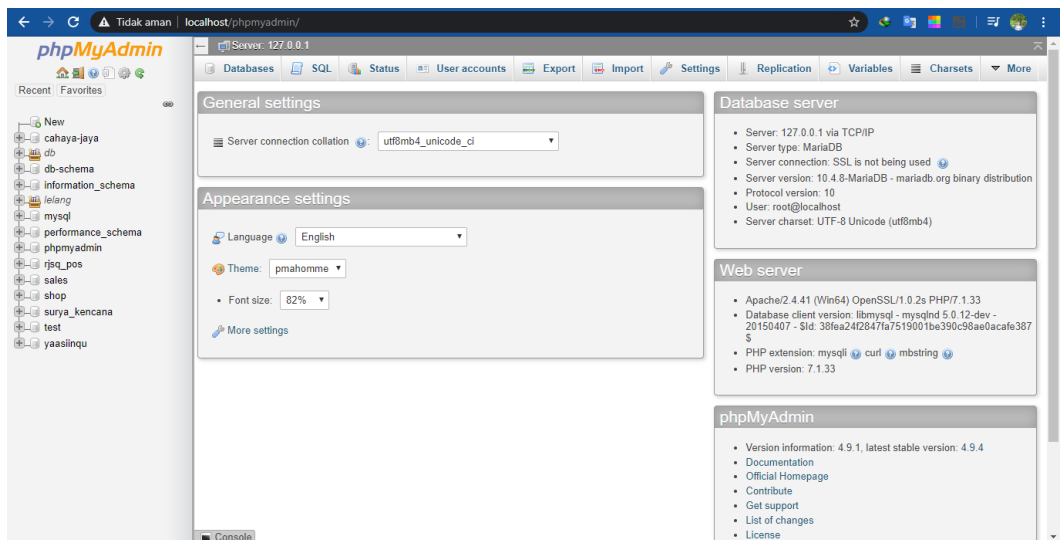


Gambar 2.2 Htdocs

Merupakan sebuah folder yang digunakan untuk penyimpanan file-file atau berkas kerja kita seperti file PHP,HTML,CSS,dsb yang digunakan untuk sebuah halaman *web*.

Kapasitas yang diberikan untuk mengisi file di folder htdocs ini tergantung dari *space harddisk* yang anda miliki. Sedangkan bila menggunakan hosting, tergantung dari pembelian & penyedia hosting tersebut.

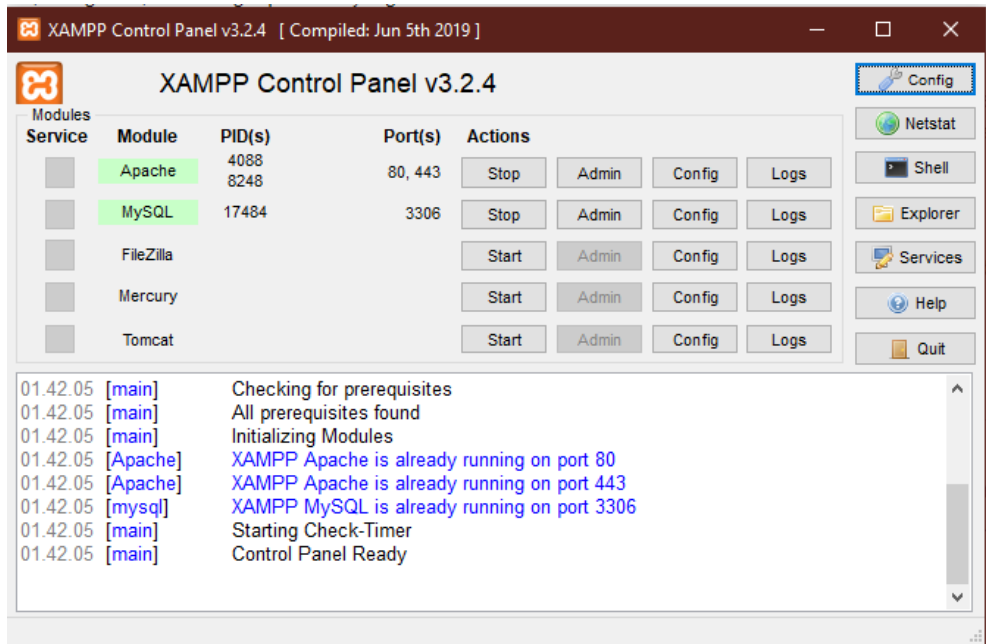
## 2. phpMyAdmin,



Gambar 2.3 phpMyAdmin

Merupakan halaman tempat mengelola dimana *database* atau MySQL itu berada, untuk menjalankannya anda dapat membuka di browser anda dengan mengetikan alamat, <https://localhost/phpmyadmin/> tapi sebelumnya anda harus mengaktifkan terlebih dahulu MySQL di control panel XAMPP.

### 3. Control Panel,



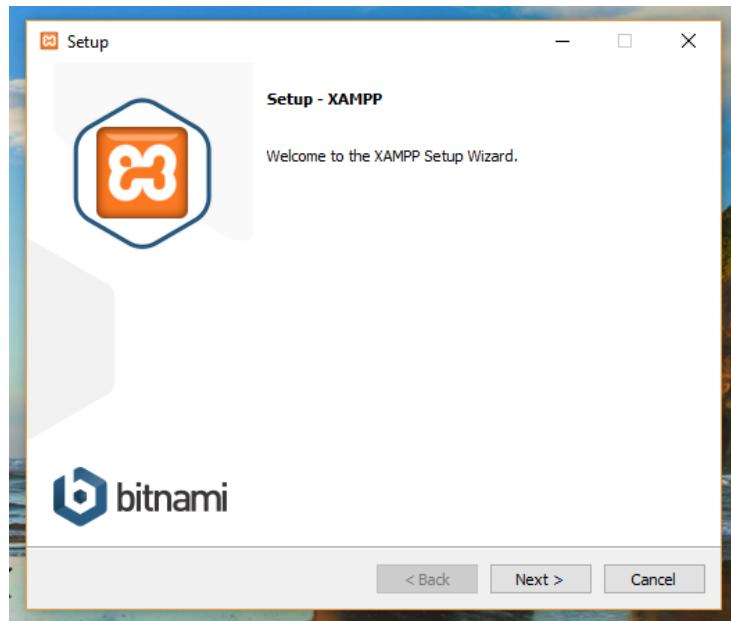
Gambar 2.4 Control Panel

Merupakan tempat untuk mengaktifkan atau mengelola layanan XAMPP, seperti *start* untuk memulai layanan dan *stop* untuk memberhentikan layanan. Sama halnya seperti pada hosting yaitu cPanel.

### 2.1.2 Instalasi XAMPP

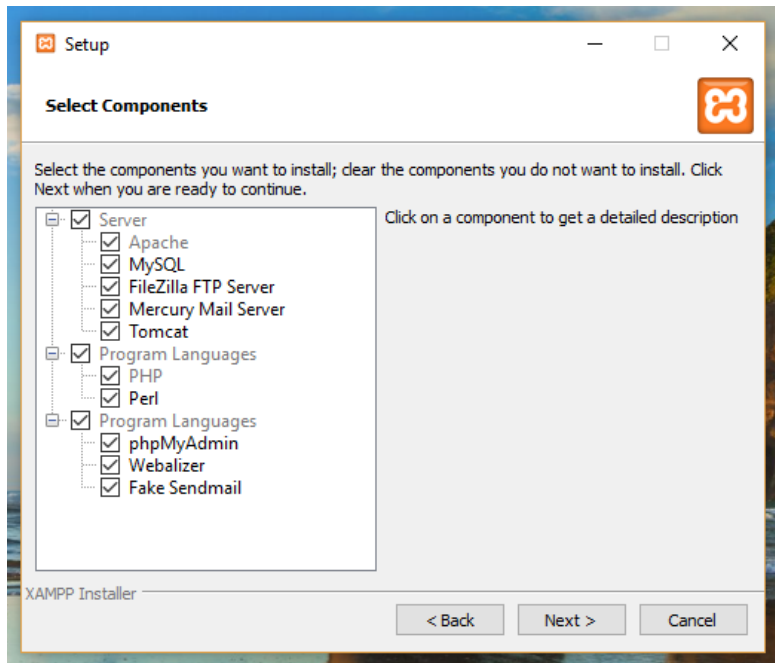
Setelah mengenal jauh program XAMPP seperti apa maka penulis berikan cara penginstallan xampp pada perangkat anda berikut cara installasinya,

1. Siapkan terlebih dahulu file XAMPP nya bisa kalian download dan dapat secara gratis pada link berikut, <https://www.apachefriends.org/download.html> sesuaikan dengan OS perangkat anda masing-masing.
2. Setelah kalian berhasil mendownloadnya, selanjutnya pastikan *space* dari ruang penyimpanan anda cukup, lalu jalankan file **xampp-windows-x64-7.3.9-0-VC15-installer** atau versi yang lain sesuai perangkat anda.
3. Akan tampil jendela baru, untuk memulai proses instalasi, pilih bahasa yang anda inginkan, lalu klik next.



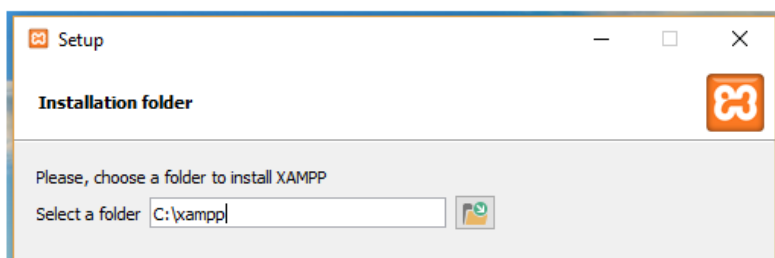
Gambar 2.5 Instalasi Jendela Xampp

4. Selanjutnya anda diminta untuk memilih komponen mana yang ingin anda install atau gunakan, umumnya untuk server web berbasis CMS seperti WordPress menggunakan MySQL, FileZilla FTP server, dan Apache, sedangkan dari bahasa pemrograman nya anda bisa centang pilihan phpMyAdmin, PHP, dan Perl. Kemudian klik next.



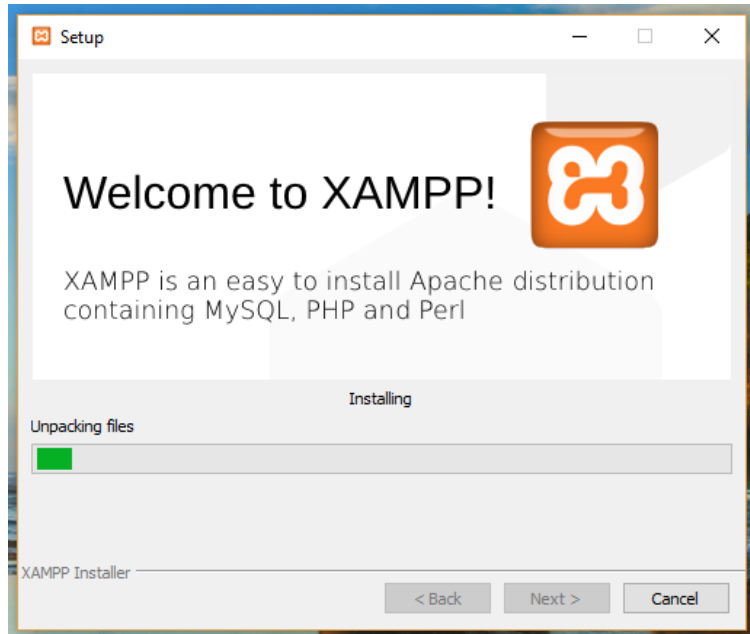
Gambar 2.6 Pilih komponen Instalasi XAMPP

5. Setelah itu anda, diminta untuk memilih lokasi, dimana program XAMPP anda akan di install, saya merekomendasikan di *Local Disk C :* , jika sudah next saja.



Gambar 2.6 Lokasi Instalasi XAMPP

6. Tunggu beberapa menit hingga progress instalasi selesai dan berhasil.



Gambar 2.7 Progress Instalasi XAMPP

7. Setelah selesai anda sudah bisa menggunakannya, dan klik *start* pada control panel untuk menjalankannya, seperti yang sudah di jelaskan pada halaman sebelumnya.



### 2.1.3 Keunggulan XAMPP

Kenapa penulis memilih XAMPP untuk server nya? Kenapa tidak menggunakan program/aplikasi yang lain? Selain karena cocok untuk pemula, XAMPP ini juga memiliki kelebihan diantaranya adalah :

1. *Database* server ini banyak digunakan programmer atau *developer* untuk ujicoba sebelum dihosting dan yang paling penting adalah gratis.
2. Kemampuan dan kapasitasnya yang luar biasa dapat menampung/membuat 60.000 tabel dengan jumlah isinya bisa mencapai 5.000.000.000 bahkan versi terbarunya saat ini bisa lebih.
3. Keamanan datanya cukup diandalkan.
4. *Multiplatform* dapat di aplikasikan pada berbagai sistem operasi.
5. Sangat cepat mengeksekusi perintah khususnya pada SQL.
6. Mempunyai *user privillage system* yang mudah, fleksibel dan efisien.

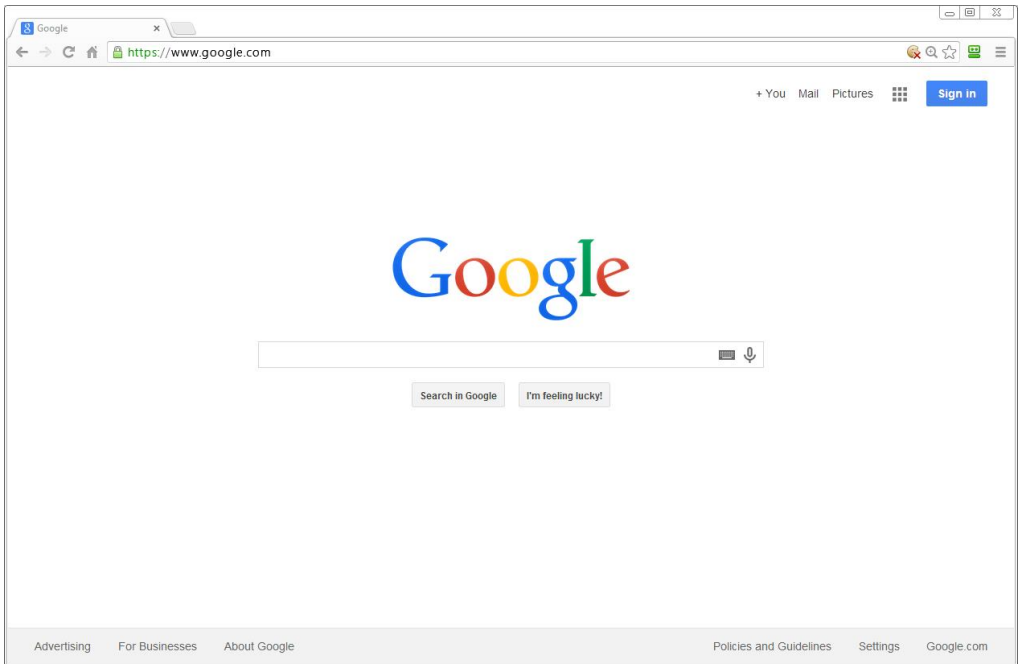
## 2.2 Web Browser

*Web browser* bisa di ibaratkan sebagai penerjemah halaman web yang dikirim menggunakan HTTP/HTTPS (*Hypertext Transfer Protocol*) atau juga FTP (*File Transfer Protocol*) yang diolah dan nantinya menjadi berupa konten yang dapat dilihat. *Web Browser* merupakan *software* penting pertama yang akan digunakan dalam membangun sebuah *project website* karena berfungsi sebagai *output* atau hasil dari halaman *web* yang kita bangun dan *web browser* ini pada dasarnya dibuat dari bahasa pemograman HTML dan CSS.

Tetapi tenang kita disini tidak akan membuat sendiri halaman *web* kita dengan HTML dan CSS, karena disini kita menggunakan *Framework Front-end* terkenal yang sudah menyediakan semua *source code* yang kita butuhkan dalam membuat *project website* ini.

Berikut penulis tampilkan beberapa macam rekomendasi *web browser* yang dapat digunakan sebagai pendukung pembuatan *project website* :

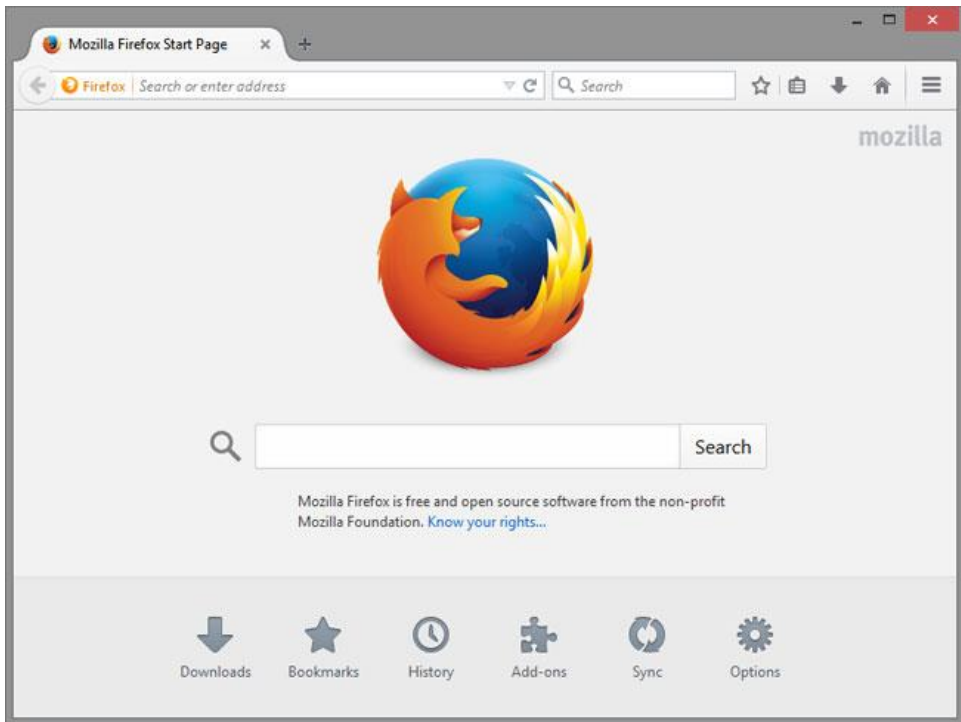
## 2.2.1 Google Chrome



Gambar 2.8 Google Chrome

Google Chrome merupakan browser yang di keluarkan oleh Google dan sudah banyak sekali yang menggunakannya khususnya negara-negara maju dan berkembang. dan hampir semua penggunanya mengatakan bahwa browser ini browser *fast and frendly*. Browser ini bisa anda dapat dengan mendownload di website resminya, [https://www.google.com/intl/id\\_id/chrome/](https://www.google.com/intl/id_id/chrome/)

## 2.2.2 Mozilla Firefox



Gambar 2.9 Mozilla Firefox

Mozilla Firefox ini dikatakan lebih baik dalam halam privasi dibandingkan browser lainnya, browser ini juga tak kalah tenarnya dengan Chrome karena browser ini paling banyak digunakan pengguna internet dunia khususnya negara kita indonesia. Browser ini bisa anda dapat dengan mendownload di website resminya, <https://www.mozilla.org/id/firefox/new/>

### 2.2.3 Safari



Gambar 2.9 Safaris

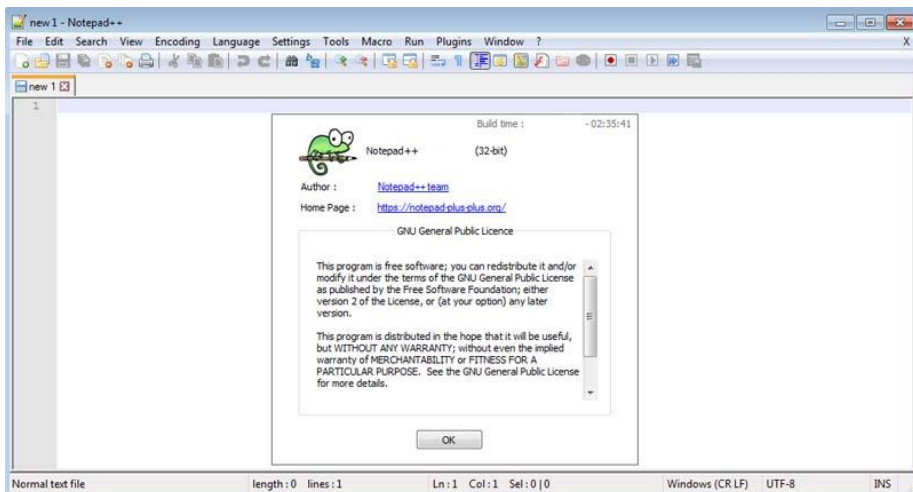
Safari merupakan browser besutan dari perusahaan raksasa *Apple*, termasuk yang paling populer dikarenakan kalangan dari pengguna *Apple* hampir semuanya menggunakan browser ini selain bentuknya yang elegan browser ini juga memiliki kecepatan *load* sebuah halaman website. Browser ini bisa anda dapat dengan mendownload di website resminya, <https://support.apple.com/downloads/safari>

## 2.3 Text Editor

*Text Editor* merupakan *software* penting selanjutnya, *software* atau aplikasi ini memang di khususkan untuk para programmer dalam berkreasi dalam bahasa pemograman apapun. Aplikasi ini merupakan *tools* pendukung dari pembuatan *website* bertujuan memudahkan aktivitas pemograman untuk membuat dan mengubah suatu codingan atau *source code*.

Beberapa *Text Editor* Gratis yang di rekomendasi oleh penulis diantaranya adalah :

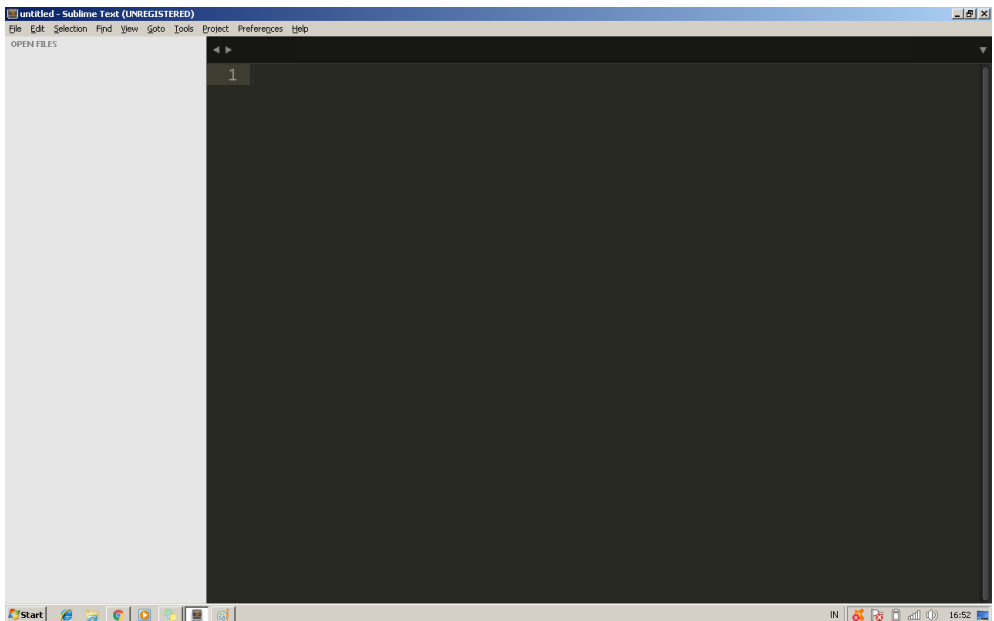
### 2.3.1 Notepad ++



Gambar 2.10 Notepad ++

Merupakan *Text Editor* yang tersedia untuk sistem operasi *Windows*. Dengan fitur-fitur yang menarik dan mendukung para programmer dalam menulis *source code*. Merupakan *software* yang banyak diminati programmer karena di distribusikan secara gratis. *Text Editor* ini bisa anda dapat dengan mendownload di website resminya, <https://notepad-plus-plus.org/downloads/>

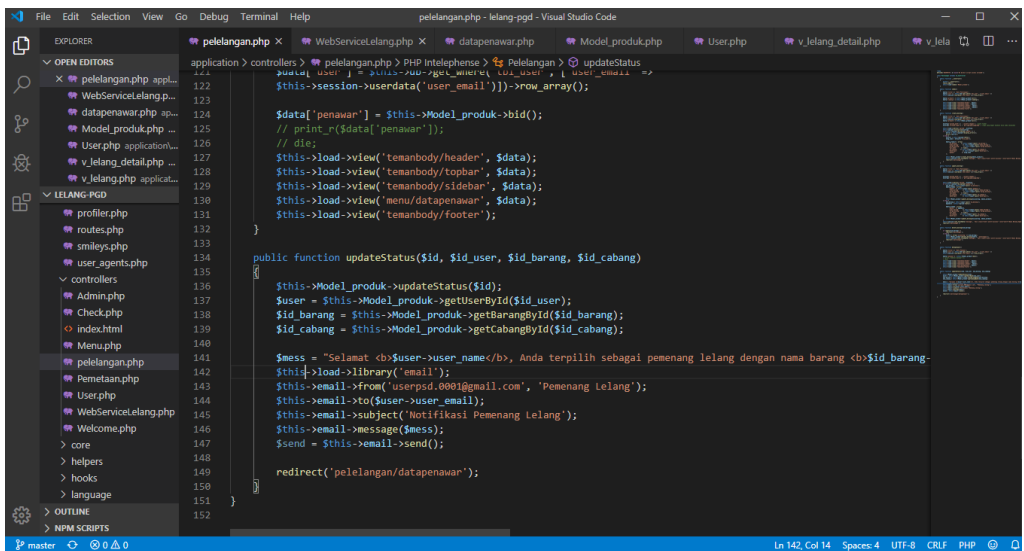
### 2.3.2 Sublime Text



Gambar 2.11 Sublime Text

Sublime merupakan *text editor* yang populer di kalangan programmer, tampilanya yang simpel namun enak di lihat serta fleksibel dan *powerfull, Software* atau aplikasi ini tersedia untuk sitem operasi *Windows, Linux, dan Mac*. Aplikasi ini juga bisa didapatkan secara gratis, *Text Editor* ini bisa anda dapat dengan mendownload di website resminya, <https://www.sublimetext.com/download>

### 2.3.3 Visual Studio Code (VS Code)



Gambar 2.12 VS Code



VS Code merupakan *text editor* yang populer saat ini di kalangan programmer dikarenakan ringan dan handal, *software* besutan *Microsoft* ini tersedia di berbagai sistem operasi seperti *Windows*, *Linux*, dan *Mac*. *Text editor* ini memiliki kertertarikan sendiri untuk para programmer dikarenakan memiliki fitur-fitur yang banyak sekali seperti *Git*, *Intellisense*, *Debugging* dan fitur ekstensinya yang menambah kemampuan *text editor* ini. Selain itu *software* ini juga bersifat *open source* dan gratis. *Text Editor* ini bisa anda dapat dengan mendownload di website resminya, <https://code.visualstudio.com/download>

## BAB 3

### Metode Pengembangan

---

Pada pengamplikasian pembuatan *web* yang akan kita bangun dengan studi kasus sebuah pelelangan, kita akan membangun aplikasi pelelangan berbasis *web* berdasarkan pelelangan yang masih dilakukan dengan penawaran harga secara tatap muka. Kita akan mengubah metode tersebut dengan pengaplikasian nya menjadi sistem berbasis web, dengan bantuan metode pengembangan *Research And Development* atau biasa dikenal R&D. Metode ini digunakan penulis untuk membangun dan menghasilkan produk/sistem baru yang berupaya melakukan pengembangan dari sistem atau proses bisnis yang sudah ada sebelumnya.

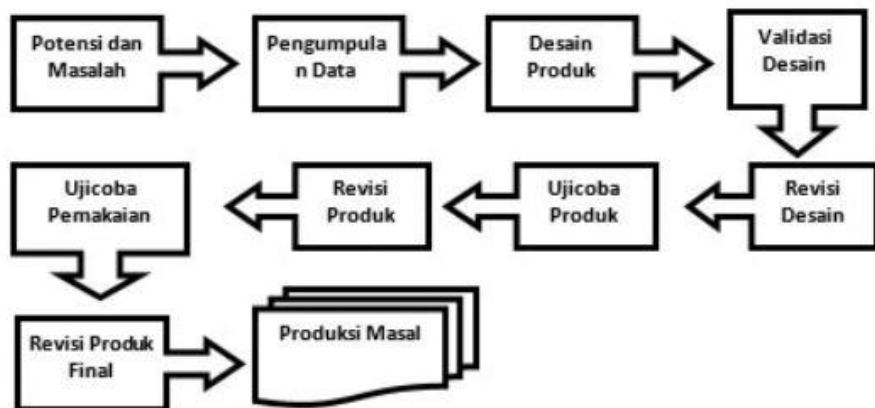
### 3.1 Penjelasan R&D



Gambar 3.1 R&D

R&D Merupakan rancangan pengembangan yang mempunyai tujuan untuk mengembangkan dan memvalidasi produk atau aplikasi yang akan di rancang dan di kembangkan. Model R&D ini memiliki langkah-langkah antara lain Penelitian menemukan potensi masalah, mengumpulkan informasi, riset skala kecil, *planning*, pengemangan desain, uji coba tahap awal, revisi hasil ujicoba, ujicoba kembali, revisi hasil, uji kelayakan, revisi produk terakhir, implementasi produk, Metode ini mengacu pada penelitian dan pengembangan.

### 3.2 Tahapan-tahapan Metode R&D



Gambar 3.2 Tahapan-tahapan R&D

Pada umumnya penelitian pembangunan ini terdapat sepuluh alur tahapan seperti gambar berikut,

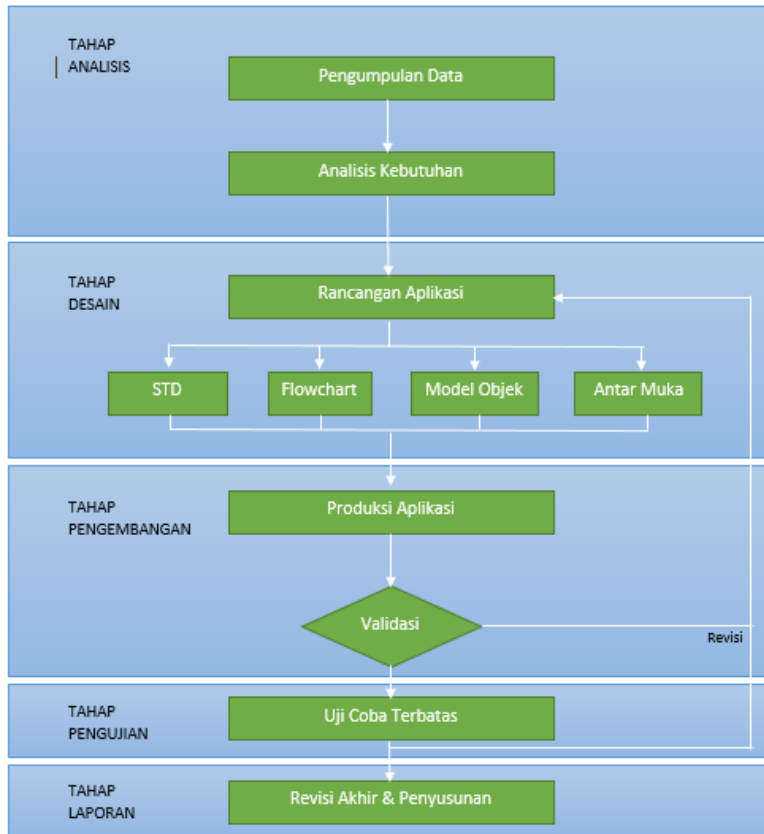
1. Potensi dan masalah. Penelitian ini didapatkan berdasarkan latar belakang masalah yang ada. Potensi merupakan sesuatu yang memiliki nilai tambah. Masalah adalah merupakan penyimpangan antara sesuatu yang diharapkan dengan apa yang sudah terjadi.
2. Mengumpulkan informasi. Setelah potensi permasalahan telah ditemukan, maka selanjutnya perlu dikumpulkan informasi berguna sebagai bahan untuk perencanaan

produk tertentu dengan tujuan mengatasi dari potensi masalah yang ada.

3. Desain produk. Produk di desain dengan ketentuan *user experience* agar mudah digunakan.
4. Validasi desain. Validasi desain adalah kegiatan penilaian yang telah di rancang dan telah di validasi produk oleh ahli yang berkompeten pada bidangnya sesuai studi kasus yang diangkat.
5. Perbaikan desain. Setelah dirasa desain produk sudah selesai, maka akan dilakukan penilaian melalui diskusi dengan orang-orang yang paham dengan alur proses bisnis yang kita buat, sehingga akan dapat diketahui kelemahannya. Setelah itu dapat diperbaiki hingga menjadi sempurna.
6. Uji Coba produk. Produk akan di ujicoba oleh orang yang memahami proses bisnis dari apa yang kita buat.
7. Revisi produk. Setelah di ujicoba akan ada pencarian atas kelemahan-kelemahan produk yang kita buat yang selanjutnya diperbaiki agar mendapatkan produk yang lebih sempurna.
8. Uji coba pemakaian. Produk akan di uji coba kembali oleh orang yang menegerti alur proses bisnis sistem yang akan kita bangun.

9. Revisi produk. Akan ada perbaikan kembali pada sistem/produk apabila ditemukan *bug* kembali.
10. Pembuatan produk masal. jika dirasa produk/sistem yang kita bangun telah optimal maka akan dilakukan pembuatan secara masal.

Tetapi untuk pengembangan aplikasi yang akan penulis bangun, penulis merubah nya dengan hanya menggunakan lima tahapan saja diantaranya ialah, tahap analisis, desain, pengembangan, pengujian dan terakhir menggunakan tahapan laporan.



Gambar 3.3 Tahapan Ringkas R&D

Berikut merupakan penjelasan tahapan alur diagram metode R&D yang penulis bangun :

### 1. Tahapan Analisis

Meliputi analisa kebutuhan dan pengumpulan data. Pengumpulan data dilakukan dengan cara observasi melihat proses pelelangan saat ini yang masih dilakukan dengan

cara tatap muka dan akan kita ubah dengan cara membuat sebuah aplikasi/sistem pelelangan.

## 2. Tahapan Desain

Perancangan yang dilakukan adalah diawali dengan analisis yang akan dibangun/dikembangkan, perancangan proses, dan perancangan tampilan antar muka, pada tahap ini *framework front-end* atau Bootstrap kita akan di gunakan.

## 3. Tahapan Pengembangan

Merupakan tahap dimana hasil rancangan dari tahap / proses bisnis sebelumnya dan diimplementasikan ke pembuatan aplikasi. Proses pengembangan ini bertujuan untuk menghasilkan produk awal. Pada tahap ini kita akan mengembangkan proses pelelangan yang awalnya masih secara manual akan kita ubah menjadi terkomputerisasi dengan membangun aplikasi lelang berbasis *web*.

## 4. Tahapan Pengujian

Dilakukan dengan *software tester* yang bernama selenium. Pengujian ini dilakukan untuk memastikan apakah hasil aplikasi sudah sesuai dengan tujuan yang diharapkan sebelumnya. Pada tahap ini kita akan menguji ketika aplikasi telah selesai dengan menggunakan program selenium.

## 5. Tahapan Laporan

Adalah penulisan laporan yang disusun sesuai dengan program aplikasi yang telah di buat dengan studi kasus yang ada.



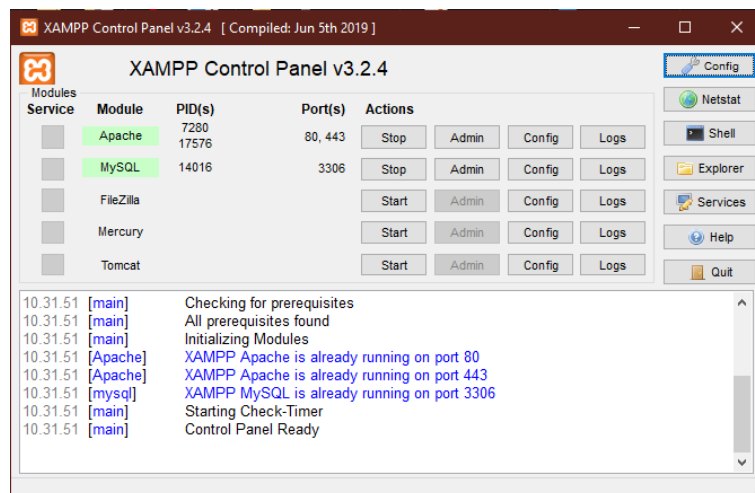
# BAB 4

## Pembuatan Database Aplikasi

Setelah pada bab selanjutnya telah di singgung tentang *database* MySQL serta *Tools* yang akan digunakan, maka pada bab ini kita akan langsung membuat sebuah *database* dengan beberapa tabel untuk aplikasi pelelangan yang akan kita buat.

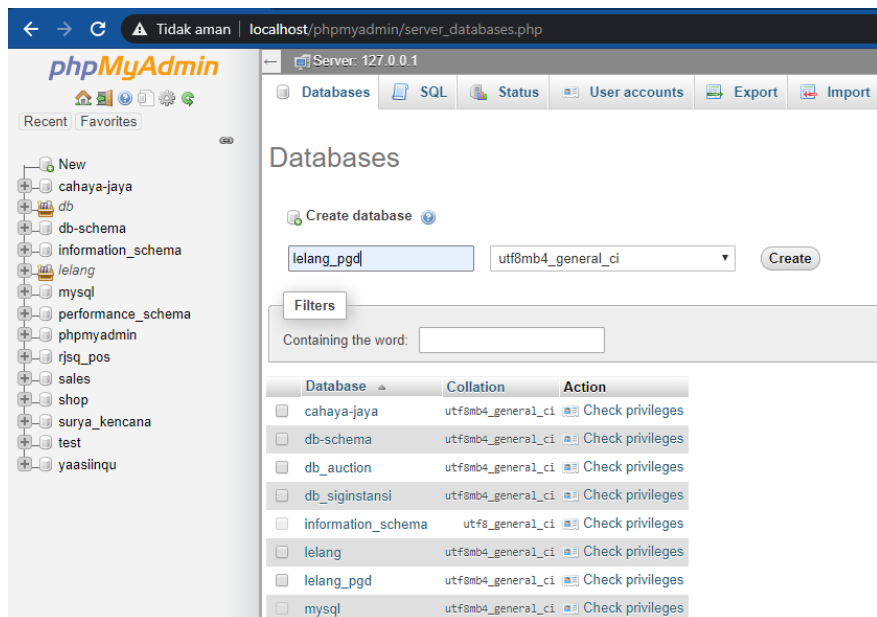
### 4.1 Langkah-langkah Membuat Database Pelelangan

1. Nyalakan terlebih dahulu Apache dan MySQL anda melalui Control Panel XAMPP,



Gambar 3.4 Buka Control Panel

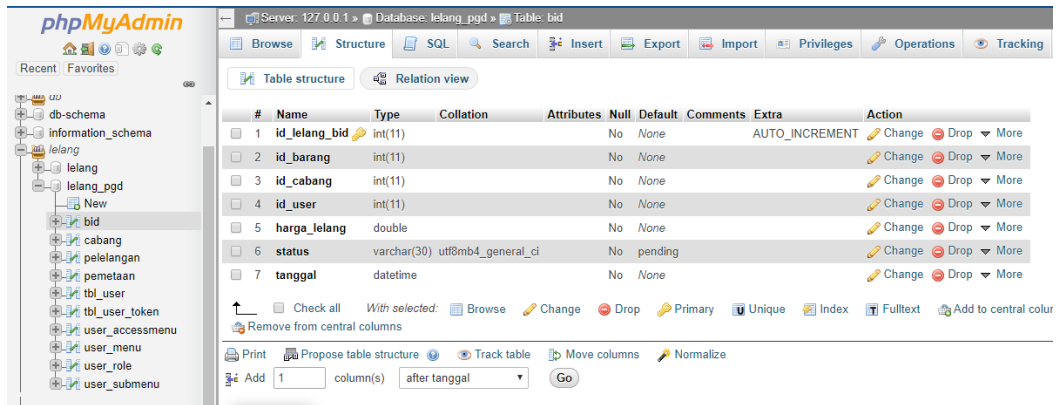
2. Buka browser anda, lalu masuk kedalam phpMyAdmin dengan url seperti berikut, <https://localhost/phpmyadmin/> setelah terbuka, Buat terlebih dahulu sebuah *database* dengan nama terserah anda sesuai keinginan anda. Disini saya membuatnya dengan nama **lelang\_pgd**.



Gambar 3.5 Pembuatan *Database*

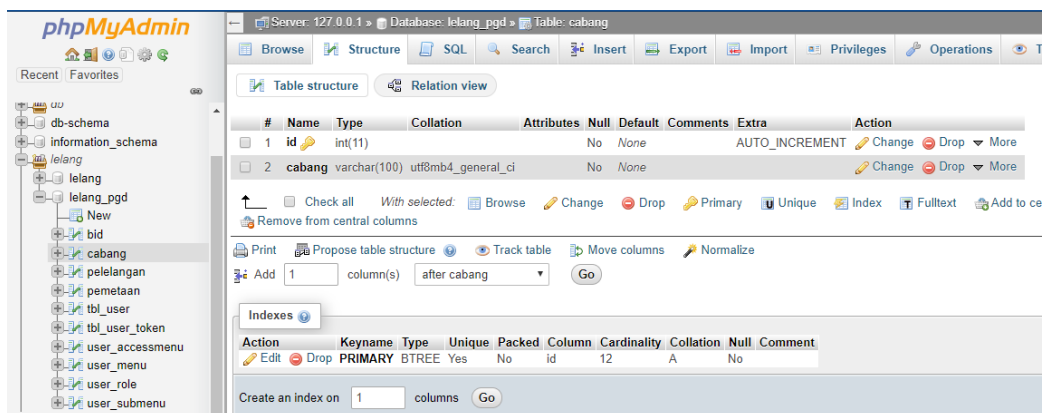
3. Buat 10 tabel dengan struktur dan tipe data seperti berikut,

## Pembuatan tabel **bid**,



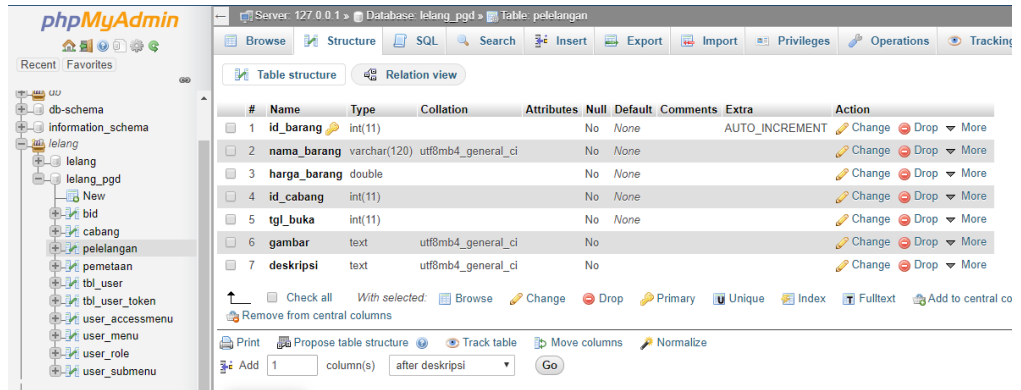
Gambar 3.6 Pembuatan Tabel bid

## Pembuatan tabel **cabang**,



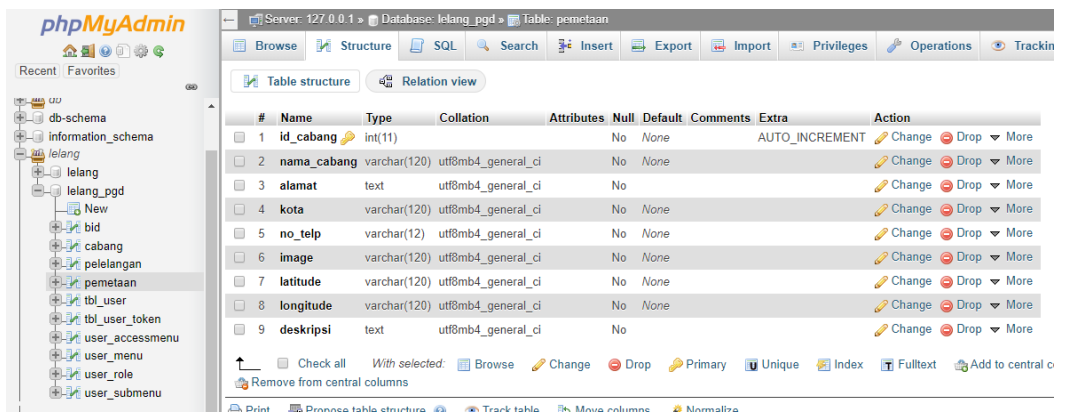
Gambar 3.7 Pembuatan Tabel cabang

## Pembuatan tabel **pelelangan**,



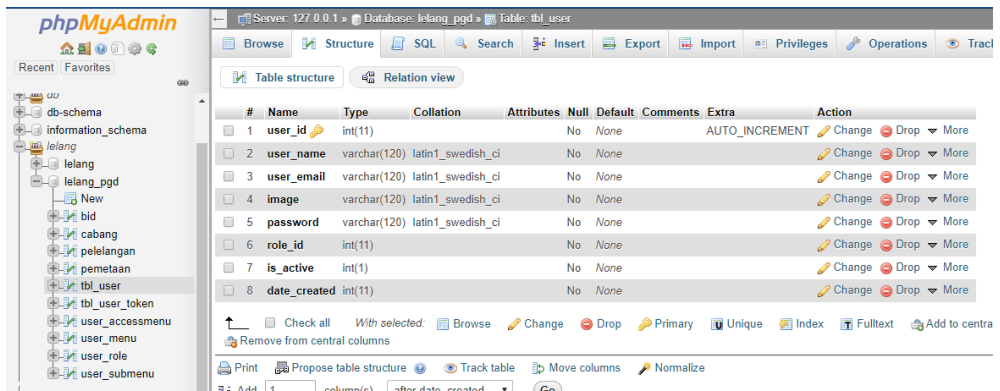
Gambar 3.8 Pembuatan Tabel pelelangan

## Pembuatan tabel **pemetaan**,



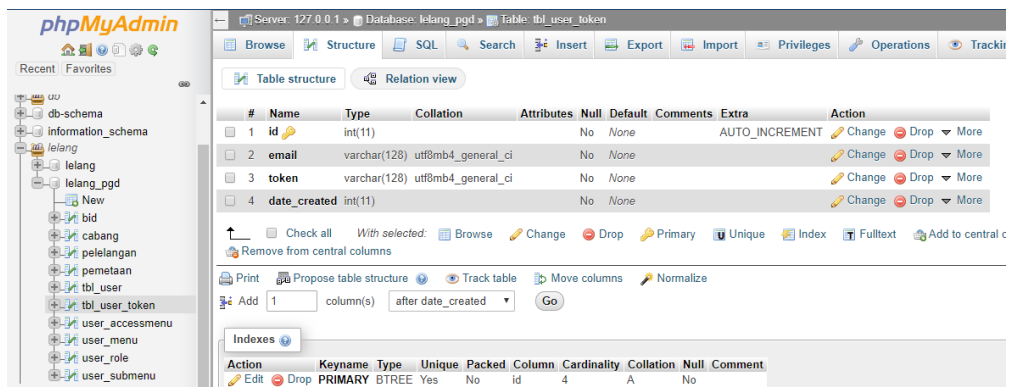
Gambar 3.9 Pembuatan Tabel pemetaan

## Pembuatan tabel **tbl\_user**,



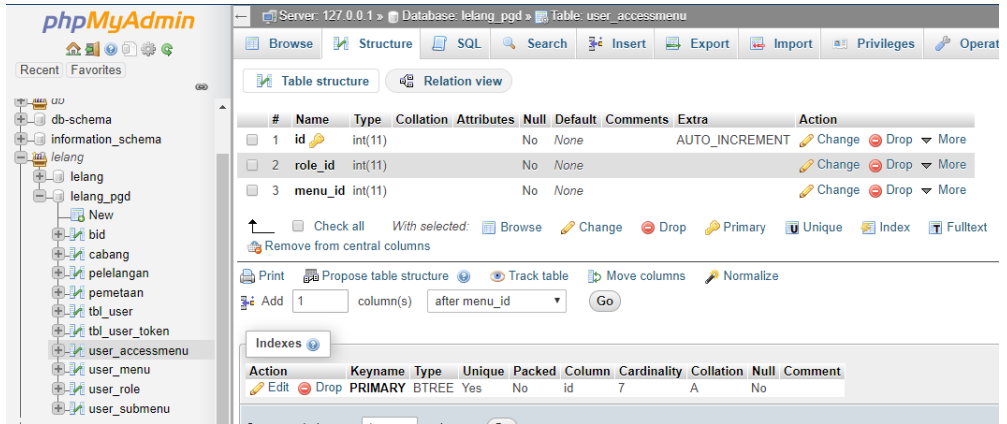
Gambar 3.10 Pembuatan Tabel **tbl\_user**

## Pembuatan tabel **tbl\_user\_token**,



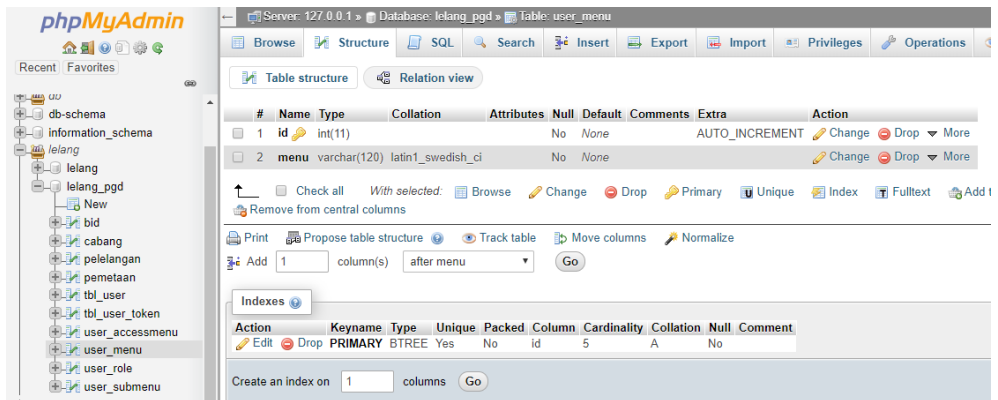
Gambar 3.11 Pembuatan Tabel **tbl\_user\_token**

### Pembuatan tabel `user_accessmenu`,



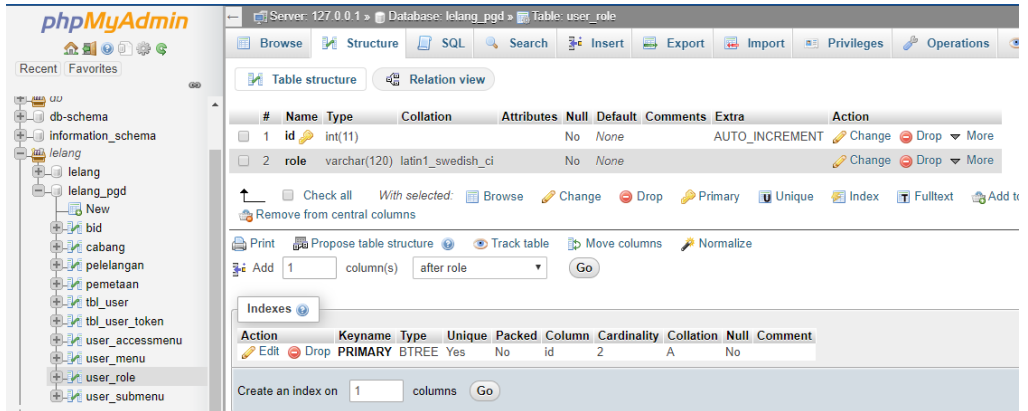
Gambar 3.12 Pembuatan Tabel `user_accessmenu`

### Pembuatan tabel `user_menu`,



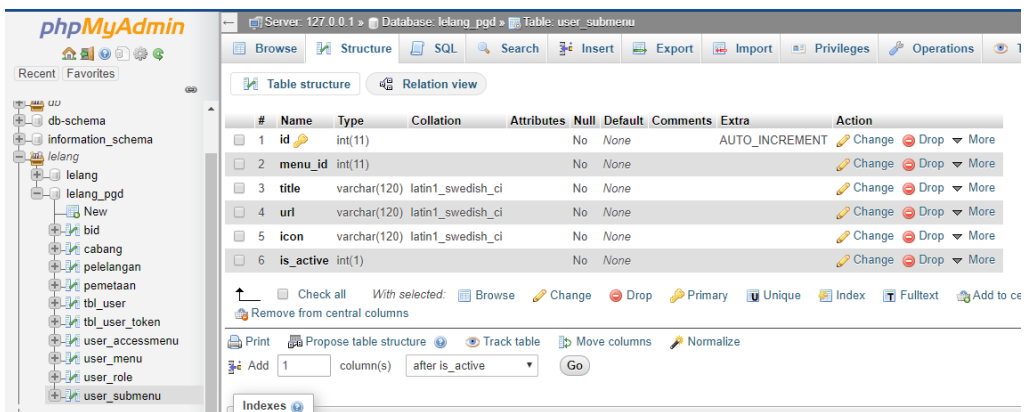
Gambar 3.13 Pembuatan Tabel `user_menu`

## Pembuatan tabel **user\_role**,



Gambar 3.14 Pembuatan Tabel user\_role

## Pembuatan tabel **user\_submenu**,



Gambar 3.15 Pembuatan Tabel user\_submenu

Setelah semua tabel telah dibuat dengan sesuai maka selanjutnya adalah tahap pengcodingan akan di jelaskan di bab 5.

# BAB 5

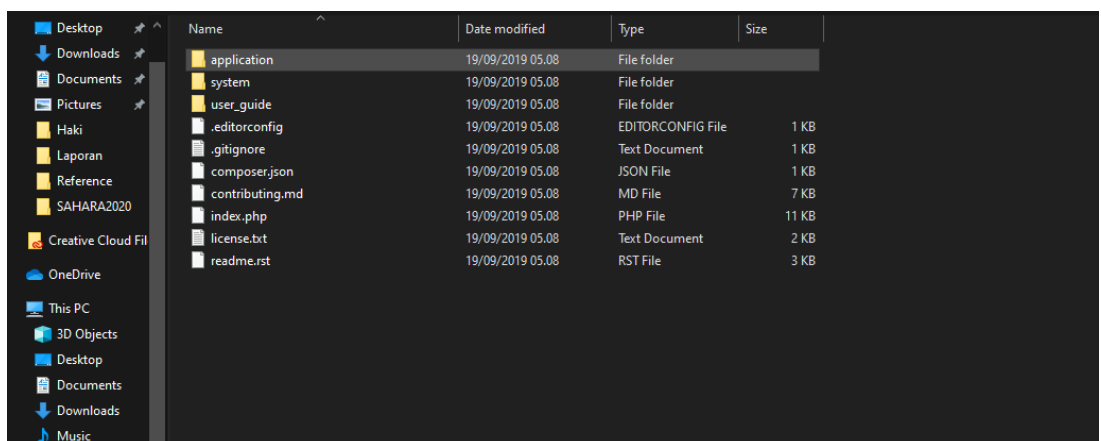
## Pembuatan Aplikasi

---

Setelah selesai membuat semua tabel yang dibutuhkan untuk *database* selanjutnya adalah tahap pembuatan aplikasi, diawali dengan instalasi *codeigniter* untuk sisi *back-end* dan *framework* Bootstrap untuk sisi *front-end*, yang sebelumnya sudah dijelaskan pada bab-bab sebelumnya.

### 5.1 Konfigurasi Awal Codeigniter

Langkah pertama pastikan anda sudah mendownload codeigniter di *website* resminya pada link berikut, <https://codeigniter.com/download> kita akan menggunakan pada versi *codeigniter* 3. Ekstrak *file* tersebut.



Gambar 3.16 Ekstrak *file* Codeigniter



Setelah di ekstrak, pindahkan atau copy semua file tersebut ke direktori penginstallan XAMPP tepatnya pada folder htdocs, contoh ( **C:\xampp\htdocs\lelang-pgd** ) lelang-pgd adalah folder yang berisi *file-file* codeigniter tadi yang telah di ekstrak.

Lalu sedikit konfigurasi, masuk ke folder ( **C:\xampp\htdocs\lelang-pgd\application\config** ) , lalu buka file **config.php**, di *text editor* kalian masing-masing, kemudia edit dibagian `$config['base_url']` menjadi **`$config['base_url'] = 'http://localhost/lelang-pgd/';`** lelang-pgd adalah nama folder yang kita miliki, yang nantinya akan kita panggil di browser. Berikut gambar jelasnya,

```
application > config > config.php > ...
 9  | URL to your CodeIgniter root. Typically this will be your base URL,
10  | WITH a trailing slash:
11  |
12  |   http://example.com/
13  |
14  | WARNING: You MUST set this value!
15  |
16  | If it is not set, then CodeIgniter will try guess the protocol and path
17  | your installation, but due to security concerns the hostname will be set
18  | to $_SERVER['SERVER_ADDR'] if available, or localhost otherwise.
19  | The auto-detection mechanism exists only for convenience during
20  | development and MUST NOT be used in production!
21  |
22  | If you need to allow multiple domains, remember that this file is still
23  | a PHP script and you can easily do that on your own.
24  |
25  | */
26  | $config['base_url'] = 'http://localhost/lelang-pgd/';
27  |
28  | /*
29  | -----
30  |   Index File
31  | -----
32  |
33  | Typically this will be your index.php file, unless you've renamed it to
34  | something else. If you are using mod_rewrite to remove the page set this
35  | variable so that it is blank.
36  |
37  | */
38  | $config['index_page'] = '';
39  |
```

Gambar 3.17 Konfigurasi config.php

Masih di folder yang sama, sekarang buka file yang bernama **autoload.php** lalu edit kembali didalamnya, pada bagian seperti berikut,

```
application > config > autoload.php > ...
54 | $autoload['libraries'] = array('database', 'email', 'session');
55 |
56 | You can also supply an alternative library name to be assigned
57 | in the controller:
58 |
59 | $autoload['libraries'] = array('user_agent' => 'ua');
60 | */
61 | $autoload['libraries'] = array('database', 'session', 'cart', 'upload', 'email', 'form_validation', 'googlemaps');
62 |
63 | /*
64 | -----
65 | Auto-load Drivers
66 | -----
67 | These classes are located in system/libraries/ or in your
68 | application/libraries/ directory, but are also placed inside their
69 | own subdirectory and they extend the CI_Driver_Library class. They
70 | offer multiple interchangeable driver options.
71 |
72 | Prototype:
73 |
74 | $autoload['drivers'] = array('cache');
75 |
```

```
application > config > autoload.php > ...
85 | -----
86 | Auto-load Helper Files
87 | -----
88 | Prototype:
89 |
90 | $autoload['helper'] = array('url', 'file');
91 | */
92 | $autoload['helper'] = array('url', 'file', 'form', 'security', 'check_login');
93 |
94 | /*
95 | -----
96 | Auto-load Config files
97 | -----
98 | Prototype:
99 |
100 | $autoload['config'] = array('config1', 'config2');
101 |
102 | NOTE: This item is intended for use ONLY if you have created custom
103 | config files. Otherwise, leave it blank.
104 |
105 | */
```

Gambar 3.18 Konfigurasi autoload.php

Masih di folder yang sama, sekarang buka file yang bernama **database.php** lalu edit kembali didalamnya, pada bagian seperti berikut,

```
application > config > database.php > ...
69 |
70 | The $query_builder variables lets you determine whether or not to load
71 | the query builder class.
72 | */
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => 'root',
80     'password' => '',
81     'database' => 'lelang_pgd',
82     'dbdriver' => 'mysqli',
83     'dbprefix' => '',
84     'pconnect' => FALSE,
85     'db_debug' => (ENVIRONMENT !== 'production'),
86     'cache_on' => FALSE,
87     'cachedir' => '',
88     'char_set' => 'utf8',
89     'dbcollat' => 'utf8_general_ci',
90     'swap_pre' => '',
91     'encrypt' => FALSE,
92     'compress' => FALSE,
93     'stricton' => FALSE,
94     'failover' => array(),
95     'save_queries' => TRUE
96 );
97
```

Gambar 3.19 Konfigurasi database.php

Ubah pada bagian, **hostname** ,**username** ,**password** ,**database** ,**dbdriver**. Sesuaikan dengan yang anda miliki. Password saya kosongkan karena saya tidak men *setting* password pada localhost saya.

Kemudian, masih di folder yang sama buka *file* bernama **routes.php** edit kembali isi di dalamnya, seperti berikut.

```
application > config > routes.php > ...
46 | When you set this option to TRUE, it will replace ALL dashes in the
47 | controller and method URI segments.
48 |
49 | Examples: my-controller/index -> my_controller/index
50 |           my-controller/my-method -> my_controller/my_method
51 | */
52 $route['default_controller'] = 'check';
53 $route['404_override'] = '';
54 $route['translate_uri_dashes'] = FALSE;
55
```

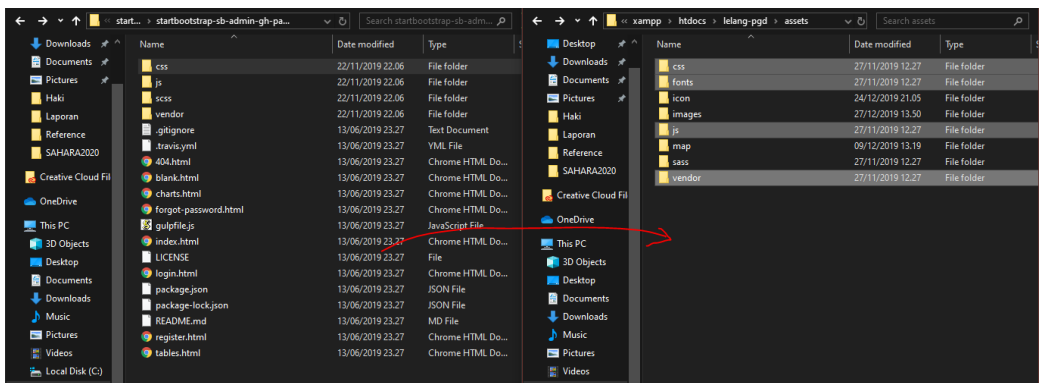
Gambar 3.20 konfigurasi routes.php

Ubah isi **default\_controller** dengan controller utama anda, disini penulis memasukan **check**, check ini merupakan nama controller yang nantinya ketika kita membuka *web* kita, controller tersebutlah yang akan *load* pertama. Untuk pembuatan controller **check** nanti akan penulis jelaskan di halaman-halaman berikutnya.

## 5.2 Konfigurasi Awal Bootstrap

Setelah melakukan instalasi & konfigurasi *codeigniter* untuk *back-end*. Pada bagian ini penulis akan memberikan cara instalasi & konfigurasi pada sisi *fornt-end* dengan menggunakan bootstrap dengan *template* SB ADMIN, dapat anda *download* secara gratis pada link berikut, <https://startbootstrap.com/templates/sb-admin/> , setelah anda berhasil mendownloadnya, ikuti langkah berikut untuk memodifikasi *template* nya, kita akan mengambil apa yang dibutuhkan saja :

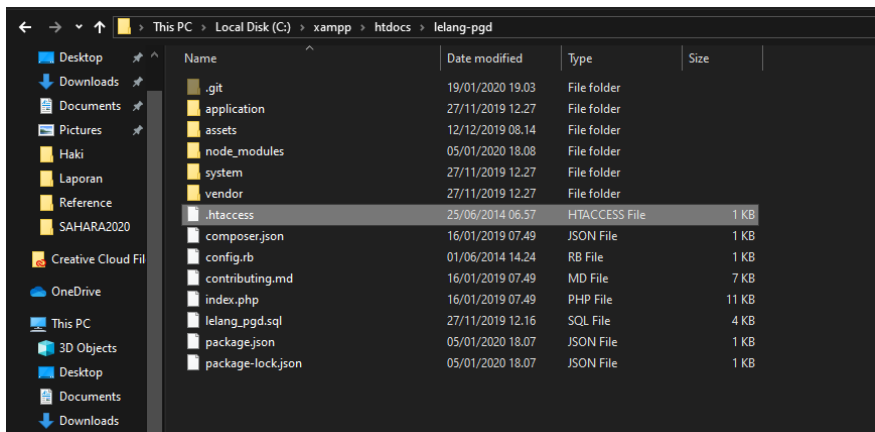
Setelah didownload bootstrapnya, ekstrak *file* tersebut dan pindahkan beberapa folder seperti **css,js,fonts & vendor** ke folder *htdocs* kalian dengan membuat folder baru dan beri nama **assets**, contohnya seperti berikut, ( C:\xampp\htdocs\lelang-pgd\assets )



Gambar 3.21 Konfigurasi Bootstrap

Kemudian buat juga folder **icon**, **images**, **map**. Untuk isinya biarkan kosongkan saja terlebih dahulu, nanti akan kita isi dibagian pengcodingan.

Selanjutnya, agar ketika kita memanggil *url* di browser kita tidak kepanjangan, kita membuat sebuah *file* bernama **.htaccess** , buat *file* tersebut langsung di dalam folder utama kalian, seperti pada gambar berikut,



Gambar 3.22 *file* htaccess

Berikut isi dari *file* **.htaccess**

```
RewriteEngine On
```

```
RewriteCond %{REQUEST_URI} ^/system.*
```

```
RewriteRule ^(.*)$ index.php?/$1 [L]
```

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

Pastikan nama sesuai *file* sesuai. Inti pada codingan diatas adalah membuat otomatis ketika kita memanggil sebuah halaman *web* kita tidak perlu lagi menggunakan `index.php` dari yang awalnya ketika memanggil halam web seperti ini, <http://localhost/lelang-pgd/index.php> ketika sudah membuat *file* `.htaccess` ini tidak perlu lagi mengetikan `index.php` jadi langsung saja <http://localhost/lelang-pgd/>

Setelah semuanya telah dilakukan konfigurasi, sekarang saatnya kita masuk ke tahap pengcodingan aplikasi siapkan mental dan kefokuskan anda karena nanti kita akan membuat banyak sekali halaman *web*, buka *text editor* kalian masing-masing. Disini saya menggunakan VsCode.

## 5.3 Pembuatan Halaman Login

Langkah pertama, buat sebuah *file* baru di controller dengan nama **check.php** , pastikan ikuti semua petunjuk karena penulis membuatnya secara berurutan, jika ada yang terlewat dipastikan aplikasi/program akan ada menu yang tidak berkerja.

## Check.php

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

class Check extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->library('form_validation');
    }

    public function index()
    {
        if ($this->session->userdata('user_email')) {
            redirect('user');
        }

        $this->form_validation-
>set_rules('email', 'Email', 'trim|required|valid_email');

        $this->form_validation-
>set_rules('password', 'Password', 'trim|required');

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

            $data['title'] = 'Pegadaian Lelang Login Page';

            $this->load-
>view('temanbody/check_header', $data);
            $this->load->view('check/login');
            $this->load-
>view('temanbody/check_footer');
        } else {
            // kalo sukses
            $this->_login();
        }
    }
}
```



```
}
}
```

Pada initinya codingan diatas membuat sebuah fungsi login pada halaman *web* dengan aturan/ketentuan *email* dan *password* yang di inputkan sesuai, jika *email* dan *password* dimasukan sesuai maka akan di lanjutkan ke fungsi selanjutnya yaitu fungsi **\_login** yang bersifat *private function* hanya dapat diakses oleh controller **check** saja.

```
private function _login()
{
    $user_email = $this->input->post('email');
    $password = $this->input->post('password');

    $user = $this->db-
>get_where('tbl_user', ['user_email' => $user_email])-
>row_array();

    // jika usernya ada
    if ($user) {
        // jika usernya aktif
        if ($user['is_active'] == 1) {
            // cek pas
            if (password_verify($password, $user['pa
ssword'])) {
                $data = [
                    'user_email' => $user['user_email'],
                    'role_id' => $user['role_id'],
                    'id_user' => $user['user_id']
                ];
            }
        }
    }
}
```

```

        $this->session->set_userdata($data);
        if ($user['role_id'] == 1) {
            redirect('admin');
        } else
            redirect('user');
    } else {
        $this->session-
>set_flashdata('message', '<div class="alert alert-
danger" role="alert">Your password is wrong!</div>');
        redirect('check');
    }
    } else {
        $this->session-
>set_flashdata('message', '<div class="alert alert-
danger" role="alert">Email is has not been Activated!</d
iv>');
        redirect('check');
    }
    } else {
        $this->session-
>set_flashdata('message', '<div class="alert alert-
danger" role="alert">Email is not registered!</div>');
        redirect('check');
    }
}
}
}

```

**Private Function \_login** ini berperan mengecek *email* dan *password* yang dimasukan oleh *user* apakah dia login sebagai admin atau member, jika *email* dan *password* benar serta sesuai dengan yang ada di *database* dan telah di aktivasi emailnya jika belum di aktivasi akan muncul sebuah notif bahwa email belum di aktivasi dan akan di bawa kembali ke halaman login, jika email belum terdaftar akan muncul sebuah notif bahwa email belum teregistrasi

dan akan di bawa kembali ke halaman login, dan jika password yang dimasukan salah maka akan muncul notif bahwa password salah, dan akan dibawa kembali ke halaman login, jika yang di masukan sesuai ia akan langsung masuk kehalaman utama masing-masing dari *user* yang login tersebut.

Setelah itu kita akan membuat *view* atau tampilan dari halaman login, tetapi sebelumnya kita akan membagi *layout* dari halam web yang kita buat agar nanti mudah me *manage* nya. Dengan cara membuat folder baru didalam folder *view* yang isinya *layout* seperti *header* dan *footer* pada halaman *web* kita. Penulis menamainya folder **temanbody** , buat dua *file* php di dalam folder tersebut, **check\_footer.php** dan **check\_header.php** dua *file* tersebut akan menjadi tampilan dari halam login.

### check\_header.php

```
<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-
Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-
width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">

    <title><?= $title; ?></title>
```

```

        <!-- Custom fonts for this template-->
        <link rel="shortcut icon" href="<?php echo base_url('assets/icon/favicon.png') ?>" />
        <link href="<?= base_url(); ?>assets/vendor/fontawesome-free/css/all.min.css" rel="stylesheet" type="text/css">

        <!-- Custom styles for this template-->
        <link href="<?= base_url(); ?>assets/css/sb-admin.css" rel="stylesheet">

</head>

<body class="bg-dark">

```

#### check\_footer.php

```

<!-- Bootstrap core JavaScript-->
<script src="<?= base_url(); ?>assets/vendor/jquery/jquery.min.js"></script>
<script src="<?= base_url(); ?>assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<!-- Core plugin JavaScript-->
<script src="<?= base_url(); ?>assets/vendor/jquery-easing/jquery.easing.min.js"></script>
</body>

</html>

```

Kedua *file* diatas sebenarnya satu kesatuan yang diambil dari *template* login Bootstrap SB ADMIN yang sebelumnya telah di download, perlu di perhatikan pastikan semua **script src** mengarah ke folder Assets kita, karena semua desain *layout* dari *web* kita disimpan disitu. Seperti contoh berikut,

```
<link href="<?= base_url(); ?>assets/css/sb-admin.css" rel="stylesheet">
```

Setelah membuat kedua *file* tersebut kalian membuat satu *file* lagi untuk halaman utama dari login, agar terlihat rapih buat folder baru dengan nama **check** di dalam folder **view**, buat *file* php baru dengan nama **login.php** didalam folder **check**.

### login.php

```
<div class="container">

    <div class="card card-login mx-auto mt-5">
        <div class="card-header">Login Page</div>
        <div class="card-body">
            <div class="text-center">
                <h3>Welcome!</h3>
            </div><br>
            <?= $this->session-
>flashdata('message'); ?>
            <form class="user" method="post" action="<
?= base_url('check') ?>">
                <div class="form-group">
                    <div class="form-label-group">
                        <input type="email" id="email"
class="form-
control" name="email" placeholder="Enter Your Email" a
```

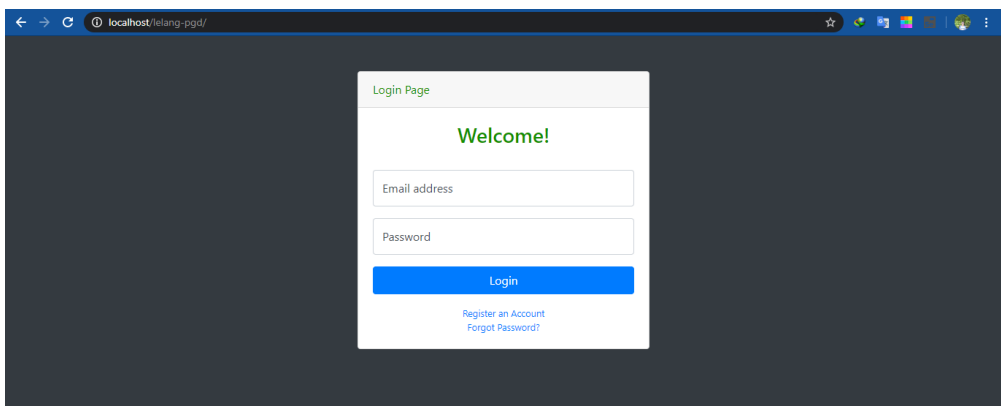
```

utofocus="autofocus" value="<?= set_value('email'); ?>
">
        <label for="email">Email addre
ss</label>
        <?php echo form_error('email',
        '<small class="text-danger pl-3">', '</small>'); ?>
        </div>
    </div>
    <div class="form-group">
        <div class="form-label-group">
            <input type="password" id="pas
sword" class="form-
control" name="password" placeholder="Password">
            <label for="password">Password
</label>
            <?php echo form_error('passwor
d', '<small class="text-danger pl-
3">', '</small>'); ?>
            </div>
        </div>
        <button class="btn btn-primary btn-
block" type="submit">Login</button>
    </form>
    <div class="text-center">
        <a class="d-block small mt-
3" href="<?= base_url('check/register'); ?>">Register
an Account</a>
        <a class="d-block small" href="forgot-
password.html">Forgot Password?</a>
    </div>
</div>
</div>

```

Pada intinya codingan diatas membuat tampilan layout baru yang berisi *label* dan *textbox email password*, lalu *button login* untuk melakukan proses login, serta label *register an account* untuk melakukan registrasi email, nanti dibagian selanjutnya kita akan membuat fungsi registrasi pada halaman login kita.

Setelah itu kalian dapat langsung melihat hasilnya di browser kalian sebelumnya pastikan terlebih dahulu Apache dan MySQL sudah di jalankan di control panel XAMPP kalian masing-masing, setelah itu dengan mengetikan <http://localhost/lelang-pgd/> maka secara otomatis tampilan akan otomatis muncul seperti berikut,



Gambar 3.23 Halaman Login

## 5.4 Pembuatan Halaman Registrasi

Buka kembali *file check.php* kalian yang berada di controller, tambahkan 4 fungsi tambahan di dalam *file* tersebut kita akan membuat halaman login kita hanya dapat diakses oleh email yang sudah registrasi dan mengaktifkannya melalui email, kita akan mengirimkan sebuah email ke member baru untuk melakukan aktivasi jika ingin login. Berikut fungsi tambahannya.

### check.php

```
public function register()
{
    if ($this->session->userdata('user_email')) {
        redirect('user');
    }

    $this->form_validation-
>set_rules('name', 'Name', 'required|trim');
    $this->form_validation-
>set_rules('email', 'Email', 'required|trim|valid_email|is_unique[tbl_user.user_email]', [
        'is_unique' => 'Email tos aya nungagunakeu
n bray!',
    ]);
    $this->form_validation-
>set_rules('password1', 'Password', 'required|trim|min_length[8]|matches[password2]', [
        'matches' => 'Password te cocok bray!',
        'min_length' => 'Password pendek teing!'
    ]);
}
```



```

        $this->form_validation-
>set_rules('password2', 'Password', 'required|trim|mat
ches[password1]');

        if ($this->form_validation->run() == false) {
            $data['title'] = 'Pegadaian Lelang Registr
ation';

            $this->load-
>view('temanbody/check_header', $data);
            $this->load->view('check/register');
            $this->load-
>view('temanbody/check_footer');
        } else {
            $email = $this->input-
>post('email', true);
            $data = [
                'user_name' => htmlspecialchars($this-
>input->post('name', true)),
                'user_email' => htmlspecialchars($emai
l),

                'image' => 'default.png',
                'password' => password_hash($this-
>input->post('password1'), PASSWORD_DEFAULT),
                'role_id' => 2,
                'is_active' => 0,
                'date_created' => time()
            ];

            //token
            $token = base64_encode(random_bytes(32));
            $user_token = [
                'email' => $email,
                'token' => $token,
                'date_created' => time()
            ];
            // var_dump($token);
            // die;

```

```

        $this->db->insert('tbl_user', $data);
        $this->db->insert('tbl_user_token', $user_token);

        $this->_sendEmail($token, 'verify');

        $this->session->set_flashdata('message', '<div class="alert alert-
success" role="alert">Thank you, your account has been
created. Please, Activated Account!</div>');
        redirect('check');
    }
}

private function _sendEmail($token, $type)
{
    $this->load->library('email');
    $this->email->from('userpsd.0001@gmail.com', 'Pegadaian Lelang WebGIS');

    $this->email->to($this->input->post('email'));

    if ($type == 'verify') {
        // print_r('coco');
        // die;
        $this->email->subject('Account Verification Pegadaian Lelang WebGIS');

        $this->email->message('Click this link to verify your account :
                                <a href="' . base_url(
) . 'check/verify?email=' . $this->input->post('email') . '&token=' . urlencode($token) . '">Aktifkeun</a>');
    }
}

```

```

        $em = $this->email->send();
        // print_r($em);
        // die;
    }

    // if ($this->email->send()) {
    //     return true;
    // } else {
    //     echo $this->email->print_debugger();
    //     die;
    // }
}

public function verify()
{
    $email = $this->input->get('email');
    $token = $this->input->get('token');

    $user = $this->db-
>get_where('tbl_user', ['user_email' => $email])-
>row_array();

    if ($user) {
        $user_token = $this->db-
>get_where('tbl_user_token', ['token' => $token])-
>row_array();
        if ($user_token) {
            if (time() - $user_token['date_created
'] < (86400)) {
                $this->db->set('is_active', 1);
                $this->db-
>where('user_email', $email);
                $this->db->update('tbl_user');

                $this->db-
>delete('tbl_user_token', ['email' => $email]);
            }
        }
    }
}

```

```

        $this->session-
>set_flashdata('message', '<div class="alert alert-
success" role="alert">' . $email . ' has been activate
d, Please Login!</div>');
        redirect('check');
    } else {

        $this->db-
>delete('tbl_user', ['user_email' => $email]);
        $this->db-
>delete('tbl_user_token', ['email' => $email]);

        $this->session-
>set_flashdata('message', '<div class="alert alert-
danger" role="alert">Activation account failed! cause,
Token expired!</div>');
        redirect('check');
    }
} else {
    $this->session-
>set_flashdata('message', '<div class="alert alert-
danger" role="alert">Activation account failed! cause,
Invalid token!</div>');
    redirect('check');
}
} else {
    $this->session-
>set_flashdata('message', '<div class="alert alert-
danger" role="alert">Activation account failed! cause,
Invalid email!</div>');
    redirect('check');
}
}
}

```

```
public function logout()
{
    $this->session->unset_userdata('user_email');
    $this->session->unset_userdata('role_id');
    $this->session-
>set_flashdata('message', '<div class="alert alert-
primary" role="alert">You have been logout, Thankyou!<
/div>');
    redirect('check');
}
```



```

        <div class="form-row">
            <div class="col-md-6">
                <div class="form-label-
group">
                    <input type="password" id="password1" name="password1" class="form-control" placeholder="Password">
                    <label for="password1">Password</label>
                    <?php echo form_error('password1', '<small class="text-danger pl-3">', '</small>'); ?>
                </div>
            </div>
            <div class="col-md-6">
                <div class="form-label-
group">
                    <input type="password" id="password2" name="password2" class="form-control" placeholder="Confirm password">
                    <label for="password2">Confirm password</label>
                </div>
            </div>
        </div>
        <div>
            <button class="btn btn-primary btn-user btn-block" type="submit">Register</button>
        </div>
        <div class="text-center">
            <a class="d-block small mt-3" href="<?= base_url('check'); ?>">Login Page</a>
            <a class="d-block small" href="forgot-password.html">Forgot Password?</a>
        </div>
    </div>

```

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



```

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

class Admin extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        check_login();
    }

    public function index()
    {
        $data['title'] = 'Dashboard';
        $data['user'] = $this->db->get_where('tbl_user', ['user_email' =>
            $this->session->userdata('user_email')])->row_array();
        $this->load->model('Menu_model');
        $data['total_cabang'] = $this->Menu_model->hitungcabang();

        $this->load->view('temanbody/header', $data);
        $this->load->view('temanbody/topbar', $data);
        $this->load->view('temanbody/sidebar', $data);
        $this->load->view('admin/index', $data);
        $this->load->view('temanbody/footer');
    }

    public function role()
    {
        $data['title'] = 'Role';
        $data['user'] = $this->db->get_where('tbl_user', ['user_email' =>
            $this->session->userdata('user_email')])->row_array();
    }
}

```

```

        $data['role'] = $this->db->get('user_role')-
>result_array();

        $this->load->view('temanbody/header', $data);
        $this->load->view('temanbody/topbar', $data);
        $this->load->view('temanbody/sidebar', $data);
        $this->load->view('admin/role', $data);
        $this->load->view('temanbody/footer');
    }

    public function roleAccess($role_id)
    {
        $data['title'] = 'Role';
        $data['user'] = $this->db-
>get_where('tbl_user', ['user_email' =>
        $this->session->userdata('user_email')])>-
>row_array();
        $data['role'] = $this->db-
>get_where('user_role', ['id' => $role_id])>-
>row_array();

        $this->db->where('id !=', 1);
        $data['menu'] = $this->db->get('user_menu')-
>result_array();

        $this->load->view('temanbody/header', $data);
        $this->load->view('temanbody/topbar', $data);
        $this->load->view('temanbody/sidebar', $data);
        $this->load->view('admin/roleaccess', $data);
        $this->load->view('temanbody/footer');
    }

    public function changeAccess()
    {
        $menu_id = $this->input->post('menuId');
        $role_id = $this->input->post('roleId');
    }

```

```
        $data = [  
            'role_id' => $role_id,  
            'menu_id' => $menu_id  
        ];  
  
        $result = $this->db->  
>get_where('user_accessmenu', $data);  
        if ($result->num_rows() < 1) {  
            $this->db->  
>insert('user_accessmenu', $data);  
        } else {  
            $this->db->  
>delete('user_accessmenu', $data);  
        }  
  
        $this->session->  
>set_flashdata('message', '<div class="alert alert-  
success" role="alert">Access, Changed!</div>');  
    }  
}
```

```

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

class Pelelangan extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        check_login();
        $this->load->model('Model_produk');
    }

    public function index()
    {
        $data['title'] = 'Data Pelelangan';
        $data['user'] = $this->db-
>get_where('tbl_user', ['user_email' =>
        $this->session->userdata('user_email')])-
>row_array();

        $data['produk'] = $this->Model_produk->all();
        $data['cabang'] = $this->Model_produk->cabang();

        $this->load->view('temanbody/header', $data);
        $this->load->view('temanbody/topbar', $data);
        $this->load->view('temanbody/sidebar', $data);
        $this->load->view('menu/v_pelelangan', $data);
        $this->load->view('temanbody/footer');
    }

    public function create_barang()
    {
        $data['title'] = 'Data Pelelangan';
        $data['user'] = $this->db-
>get_where('tbl_user', ['user_email' =>

```

```

        $this->session->userdata('user_email')))-
>row_array();
        $data['produk'] = $this->Model_produk->all();

        $config['upload_path'] = './assets/images/'; //p
ath folder
        $config['allowed_types'] = 'gif|jpg|png|jpeg|bmp
'; //type yang dapat diakses bisa anda sesuaikan

        $this->load->library('upload', $config);
        $this->upload->initialize($config);
        if (!$this->upload->do_upload('gambar')) {
            $error = $this->upload->display_errors();
            print_r($error);
        } else {
            $result = $this->upload->data();
            $img_name = $result['file_name'];

            $data_product = array(
                'id_barang'      => $this->input-
>post('id_barang'),
                'nama_barang'    => $this->input-
>post('nama_barang'),
                'harga_barang'   => $this->input-
>post('harga_barang'),
                'tgl_buka'        => time(),
                'id_cabang'       => $this->input-
>post('id_cabang'),
                'deskripsi'       => $this->input-
>post('deskripsi'),
                'gambar'          => $img_name
            );

            $this->Model_produk-
>create_barang($data_product);

```

```

        $this->session-
>set_flashdata('message', '<div class="alert alert-
success" role="alert">Data Barang Added!</div>');
        redirect('pelelangan');
    }
}

public function update_barang()
{
    $data['title'] = 'Data Pelelangan';
    $data['user'] = $this->db-
>get_where('tbl_user', ['user_email' =>
        $this->session->userdata('user_email')])-
>row_array();

    $config['upload_path'] = './assets/images/';
    $config['allowed_types'] = 'gif|jpg|png|jpeg|bmp
';

    $this->load->library('upload', $config);
    if (!$this->upload->do_upload('gambar')) {
        $id_barang = $this->input-
>post('id_barang');
        $data_produk = array(
            'gambar' => $this->input-
>post('gambar2'),
            'nama_barang'      => $this->input-
>post('nama_barang'),
            'harga_barang'    => $this->input-
>post('harga_barang'),
            'tgl_buka'         => time(),
            'id_cabang'       => $this->input-
>post('id_cabang'),
            'deskripsi'       => $this->input-
>post('deskripsi'),
        );
    }
}

```

```

        $this->Model_produk-
>update_barang($id_barang, $data_produk);
    } else {
        $id_barang = $this->input-
>post('id_barang');
        $gambar2 = $this->upload->data();

        $data_produk = array(
            'gambar' => $gambar2,
            'nama_barang' => $this->input-
>post('nama_barang'),
            'harga_barang' => $this->input-
>post('harga_barang'),
            'tgl_buka' => time(),
            'id_cabang' => $this->input-
>post('id_cabang'),
            'deskripsi' => $this->input-
>post('deskripsi'),
        );
        $this->Model_produk-
>update_barang($id_barang, $data_produk);
    }
    $this->session-
>set_flashdata('message', '<div class="alert alert-
success" role="alert">Data Barang Updated!</div>');
    redirect('pelelangan');
}

public function delete_barang($id_barang)
{
    if (empty($id_barang)) {
        redirect('pelelangan');
    } else {
        $where = array('id_barang' => $id_barang);
        $this->Model_produk-
>delete_barang($where, 'pelelangan');
    }
}

```

```

        $this->session->
>set_flashdata('message', '<div class="alert alert-
success" role="alert">Data Barang Deleted!</div>');
        redirect('pelelangan');
    }
}

public function datapenawar()
{
    $data['title'] = 'Data Penawar';
    $data['user'] = $this->db->
>get_where('tbl_user', ['user_email' =>
        $this->session->userdata('user_email')])-
>row_array();

    $data['penawar'] = $this->Model_produk->bid();
    // print_r($data['penawar']);
    // die;
    $this->load->view('temanbody/header', $data);
    $this->load->view('temanbody/topbar', $data);
    $this->load->view('temanbody/sidebar', $data);
    $this->load->view('menu/datapenawar', $data);
    $this->load->view('temanbody/footer');
}

public function updateStatus($id, $id_user, $id_barang, $id_cabang)
{
    $this->Model_produk->updateStatus($id);
    $user = $this->Model_produk->
>getUserById($id_user);
    $id_barang = $this->Model_produk->
>getBarangById($id_barang);
    $id_cabang = $this->Model_produk->
>getCabangById($id_cabang);

```



```

        $mess = "Selamat <b>$user->user_name</b>, Anda terpilih sebagai pemenang lelang dengan nama barang <b>$id_barang->nama_barang</b>, Silahkan segera mendatangi cabang <b>$id_cabang->cabang</b> untuk melakukan transaksi dan pengambilan barang, Jika lebih dari 2x24 Jam tidak datang maka dianggap Batal, Terimakasih.";
        $this->load->library('email');
        $this->email->from('userpsd.0001@gmail.com', 'Pemenang Lelang');
        $this->email->to($user->user_email);
        $this->email->subject('Notifikasi Pemenang Lelang');
        $this->email->message($mess);
        $send = $this->email->send();

        redirect('pelelangan/datapenawar');
    }

    public function delete_bid($id_lelang_bid)
    {
        if (empty($id_lelang_bid)) {
            redirect('pelelangan');
        } else {
            $where = array('id_lelang_bid' => $id_lelang_bid);
            $this->Model_produk->delete_bid($where, 'bid');
            $this->session->set_flashdata('message', '<div class="alert alert-success" role="alert">Data Deleted!</div>');
            redirect('pelelangan/datapenawar');
        }
    }
}

```

```

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

class User extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        check_login();
        $this->load->library(array('googlemaps'));
        $this->load->model('Gis_Model');
        $this->load->model('Model_produk');
    }
    public function index()
    {
        $data['title'] = 'My Profile';
        $data['user'] = $this->db->get_where('tbl_user', ['user_email' =>
            $this->session->userdata('user_email')])->row_array();

        $this->load->view('temanbody/header', $data);
        $this->load->view('temanbody/topbar', $data);
        $this->load->view('temanbody/sidebar', $data);
        $this->load->view('user/index', $data);
        $this->load->view('temanbody/footer');
    }

    public function edit()
    {
        $data['title'] = 'Edit Profile';
        $data['user'] = $this->db->get_where('tbl_user', ['user_email' =>
            $this->session->userdata('user_email')])->row_array();
    }
}

```

```

        $this->form_validation-
>set_rules('name', 'Full Name', 'required|trim');
        if ($this->form_validation->run() == false) {
            $this->load-
>view('temanbody/header', $data);
            $this->load-
>view('temanbody/topbar', $data);
            $this->load-
>view('temanbody/sidebar', $data);
            $this->load->view('user/edit', $data);
            $this->load->view('temanbody/footer');
        } else {
            $name = $this->input->post('name');
            $email = $this->input->post('email');

            //cek jika ada gambar yg mau di upload
            $upload_image = $_FILES['image']['name'];
            // var_dump($upload_image);
            // die;
            if ($upload_image) {
                $config['allowed_types'] = 'gif|jpg|png'
;
                $config['max_size'] = '2048';
                $config['upload_path'] = './assets/image
s/';

                $this->load->library('upload', $config);

                if ($this->upload->do_upload('image')) {
                    $old_image = $data['user']['image'];
                    if ($old_image != 'default.png') {
                        unlink(FCPATH . 'assets/images/'
. $old_image);
                    }
                    $new_image = $this->upload-
>data('file_name');
                    $this->db->set('image', $new_image);

```

```

        } else {
            echo $this->upload-
>display_errors();
        }
    }

    $this->db->set('user_name', $name);
    $this->db->where('user_email', $email);
    $this->db->update('tbl_user');

    $this->session-
>set_flashdata('message', '<div class="alert alert-
success" role="alert">Your profile has been, Updated!</d
iv>');

    redirect('user');
}

}

public function changePassword()
{
    $data['title'] = 'Change Password';
    $data['user'] = $this->db-
>get_where('tbl_user', ['user_email' =>
    $this->session->userdata('user_email')])-
>row_array();

    $this->form_validation-
>set_rules('passwordsaatini', 'Current Password', 'requi
red|trim');

    $this->form_validation-
>set_rules('passwordbaru1', 'New Password', 'required|tr
im|min_length[8]|matches[passwordbaru2]');

    $this->form_validation-
>set_rules('passwordbaru2', 'Confirm New Password', 'req
uired|trim|min_length[8]|matches[passwordbaru1]');

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

```

```

        $this->load-
>view('temanbody/header', $data);
        $this->load-
>view('temanbody/topbar', $data);
        $this->load-
>view('temanbody/sidebar', $data);
        $this->load-
>view('user/changepassword', $data);
        $this->load->view('temanbody/footer');
    } else {
        $passwordsaatini = $this->input-
>post('passwordsaatini');
        $password_baru = $this->input-
>post('passwordbaru1');
        if (!password_verify($passwordsaatini, $data
['user']['password'])) {
            $this->session-
>set_flashdata('message', '<div class="alert alert-
danger" role="alert">Wrong, Current Password!</div>');
            redirect('user/changepassword');
        } else {
            if ($passwordsaatini == $password_baru)
{
                $this->session-
>set_flashdata('message', '<div class="alert alert-
danger" role="alert">New password cannot be same with cu
rrent password!</div>');
                redirect('user/changepassword');
            } else {
                // password benar
                $password_hash = password_hash($pass
word_baru, PASSWORD_DEFAULT);
                $this->db-
>set('password', $password_hash);
                $this->db-
>where('user_email', $this->session-
>userdata('user_email'));

```

```

        $this->db->update('tbl_user');

        $this->session->
>set_flashdata('message', '<div class="alert alert-
success" role="alert">Password, Changed!</div>');
        redirect('user/changepassword');
    }
}
}

public function tampilanpemetaan()
{
    $data['title'] = "Lokasi Cabang";
    $data['user'] = $this->db->
>get_where('tbl_user', ['user_email' =>
    $this->session->userdata('user_email')])->
>row_array();

    $this->load->library('googlemaps');
    $config['center'] = '-6.9187126, 107.6148755';
    $config['zoom'] = '13';
    $this->googlemaps->initialize($config);

    $lokasi = $this->Gis_Model->datalokasi();
    foreach ($lokasi as $key => $value) :
        $marker = array();
        $marker['position'] = "$value->
>latitude, $value->longitude";

        $marker['animation'] = 'DROP';
        $marker['infowindow_content'] = '<div class=
"media" style="width:400px;">';
        $marker['infowindow_content'] .= '<div class
="media-left">';
        $marker['infowindow_content'] .= '

```

```

>image}")) . '" class="media-
object" style="width:150px">';
        $marker['infowindow_content'] .= '</div>';
        $marker['infowindow_content'] .= '<div class
="media-body">';
        $marker['infowindow_content'] .= '<h5 class=
"media-heading">' . $value->nama_cabang . '</h5>';
        $marker['infowindow_content'] .= '<p>No Telp
: <strong>' . $value->no_telp . '</strong></p>';
        $marker['infowindow_content'] .= '<p>' . $va
lue->alamat . '</p>';
        $marker['infowindow_content'] .= '<p>' . $va
lue->deskripsi . '</p>';
        $marker['infowindow_content'] .= '</div>';
        $marker['infowindow_content'] .= '</div>';
        $marker['icon'] = base_url("assets/icon/pgd.
png");

        $this->googlemaps->add_marker($marker);
    endforeach;

    $this->googlemaps->initialize($config);
    $data['map'] = $this->googlemaps->create_map();

    $this->load->view('temanbody/header', $data);
    $this->load->view('temanbody/topbar', $data);
    $this->load->view('temanbody/sidebar', $data);
    $this->load->view('user/lokasi', $data);
    $this->load->view('temanbody/footer');
}

```

