

**PANDUAN MEMBANGUN SEBUAH
WEBSITE BESERTA STUDI KASUS
MENGGUNAKAN 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 membagun 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 perilisan tersebut skrip nya menjadi sumber terbuka untuk semua orang khususnya para programer/pemrogram sehingga banyak yang tertarik untuk ikut mengembangkannya. Tahun ke tahun skripnya banyak digunakan 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 programer web menggunakanya 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 pemograman 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 mengistall *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 pemograman yang lain? Selain karena cocok untuk pemula, PHP juga memiliki kelebihan dari bahasa pemograman 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 menampilkan *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 dalam 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* dating 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 menyiapkannya 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 dua buah *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 programer.

best practice, kumpulan aksi yang telah teruji oleh para programer *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 programer 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 dikembangkan 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 pemrograman PHP. Referensi yang begitu banyak menjadi alasannya juga.

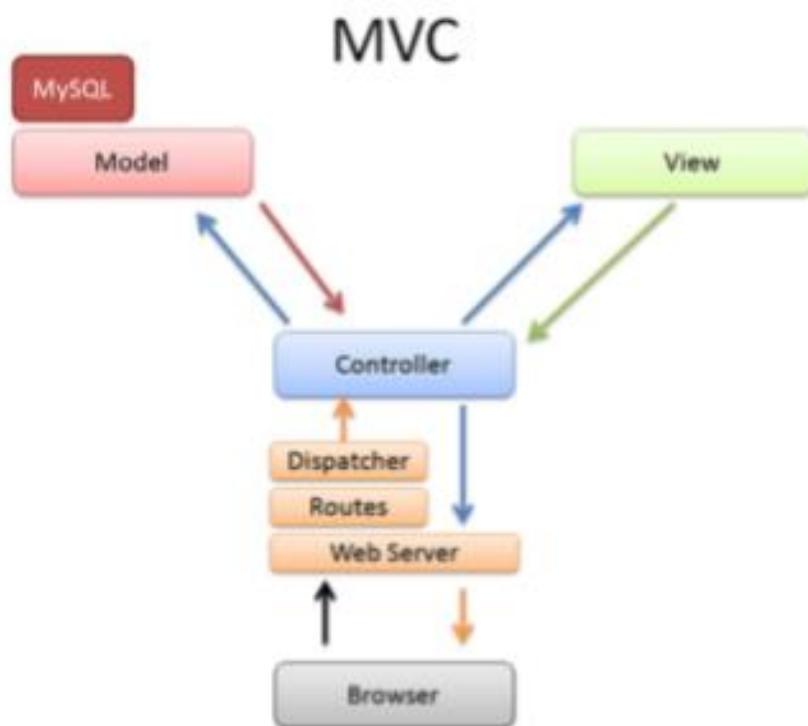
Codeigniter membantu para *developer* untuk mengerjakan aplikasi menjadi cepat daripada menulis skrip/codingan dari awal, *framework* ini memiliki library yang dapat mempermudah dalam pengembangan memuat file-file yang berekstensi PHP yang menyediakan *class* seperti *libraries*, *helpers*, *plugins*, 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 pemrograman 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 khsuskan 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 berjalannya 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



Bootstrap

Gambar 1.7 Bootstrap

Bootstrap merupakan *framework* dari *front-end*, Perancangnya adalah dua orang programer Twitter yaitu Mark Octo 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,*plungin-plugin* Javascript,icon, dan semua yang mereka kusai juga 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 jawabannya 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 programer 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 (programer 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 dijadikan produk turunan salah satu konsep utama dalam database sejak lama yaitu *Structured Query Language 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 hamat *resource memory* dibandingkan dengan *database* lain.

6. Banyak terintegrasi oleh basaha pemograman seperti, PHP,Python,Net,Java,dll.

Untuk penginstallan MySQL penulis sudah mempersiapkanya 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,bagunan,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 programer 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 dijadikan penamaan XAMPP itu sendiri yaitu,

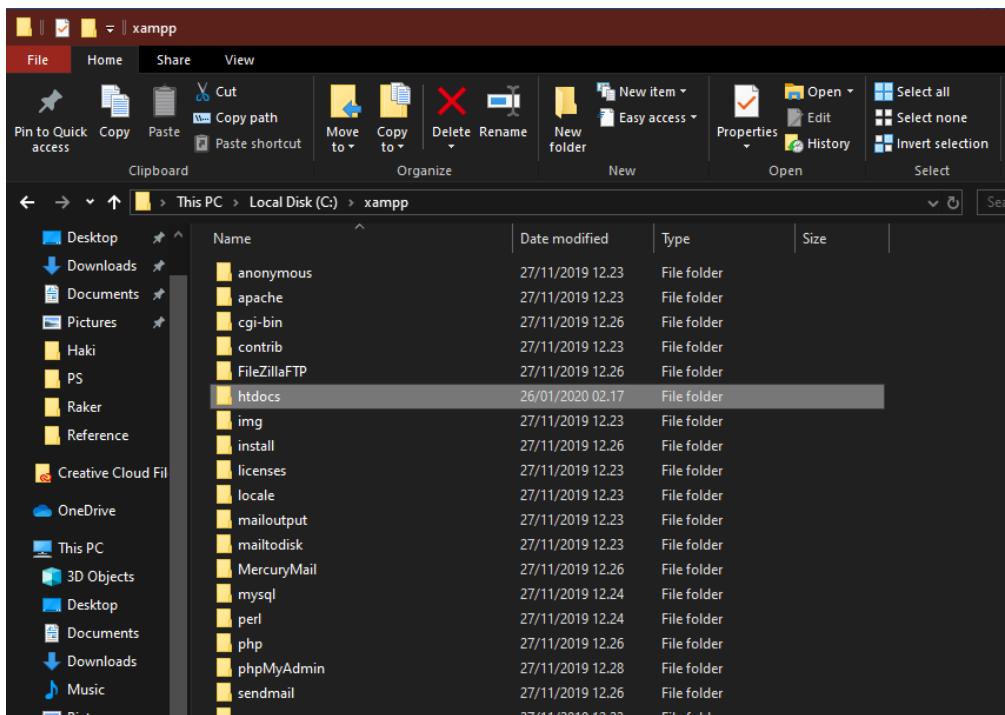
- 1.** **X**, dapat diartikan sebagai *cross platform* yang artinya banyak sistem operasi yang dapat menggunakan nya.
- 2.** **A**, dapat diartikan sebagai Apache merupakan program web server yang bersifat gratis dan *opensource*, apache disini bertugas untuk menghasilkan halaman web yang benar ke user berdasarkan skrip PHP yang telah dibangun sebelumnya oleh programer.
- 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, diantarnya adalah :

1. Htdocs

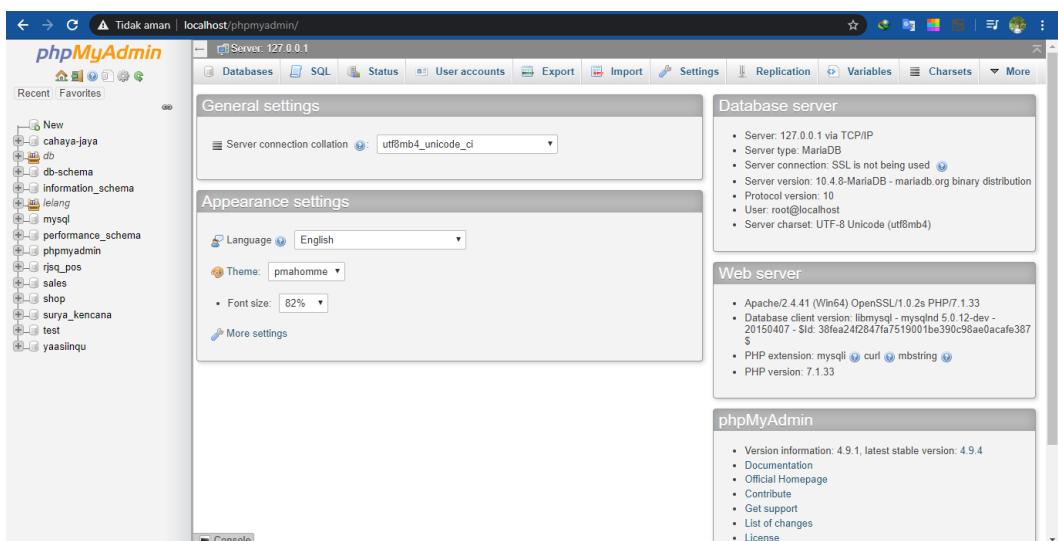


Gambar 2.2 Htdocs

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.

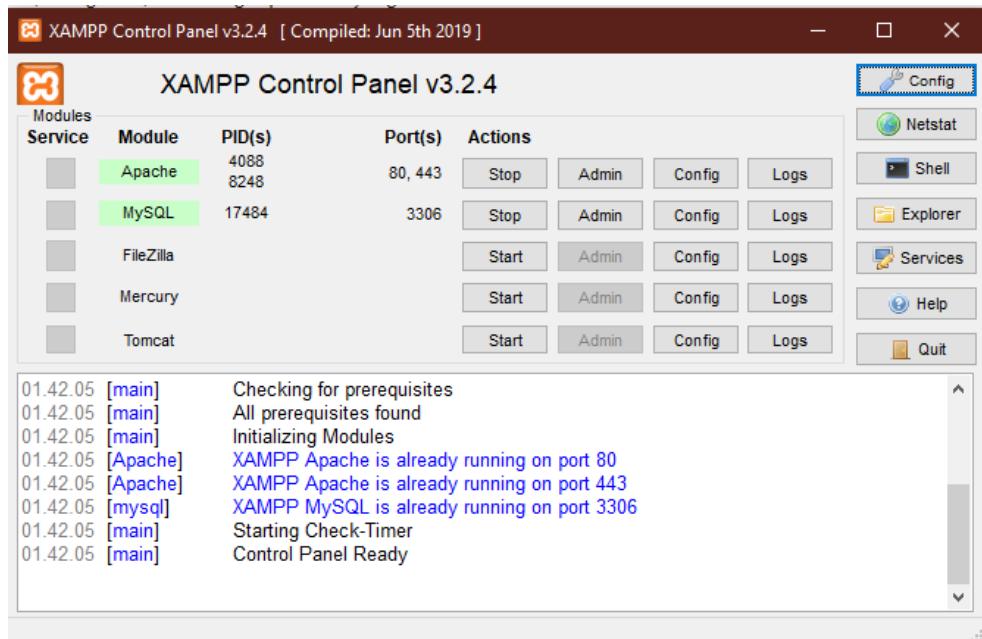
2. phpMyAdmin,



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,



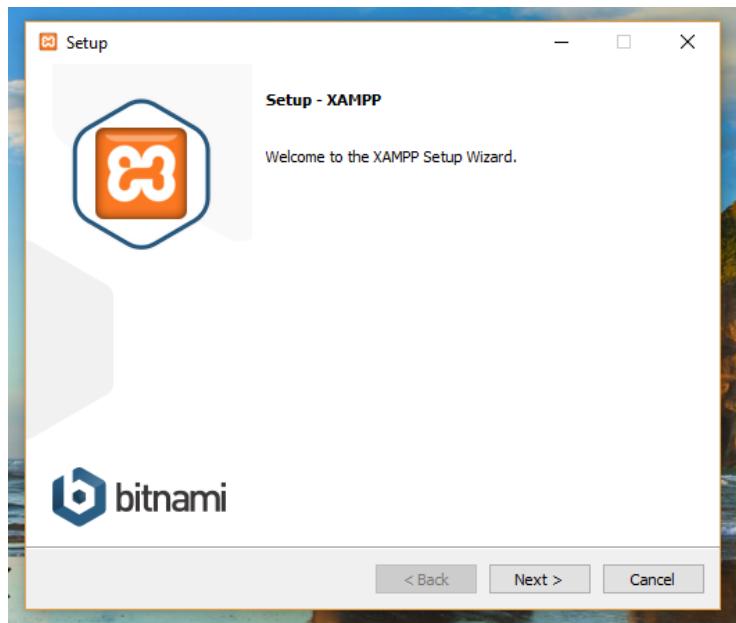
Gambar 2.4 Control Panel

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 Installasi XAMPP

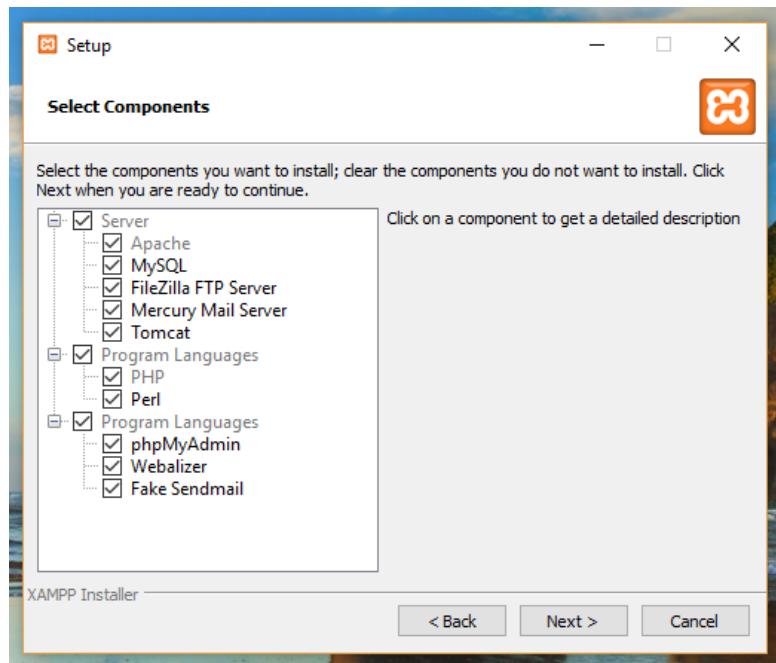
Setelah mengenal jauh program XAMPP seperti apa maka penulis berikan cara penginstallan xampp pada perangkat anda berikut cara installasinya,

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 isntallasi, pilih bahasa yang anda inginkan, lalu klik next.



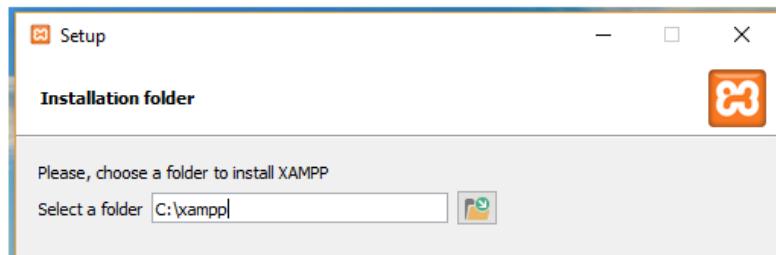
Gambar 2.5 Installasi Jendela Xampp

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.



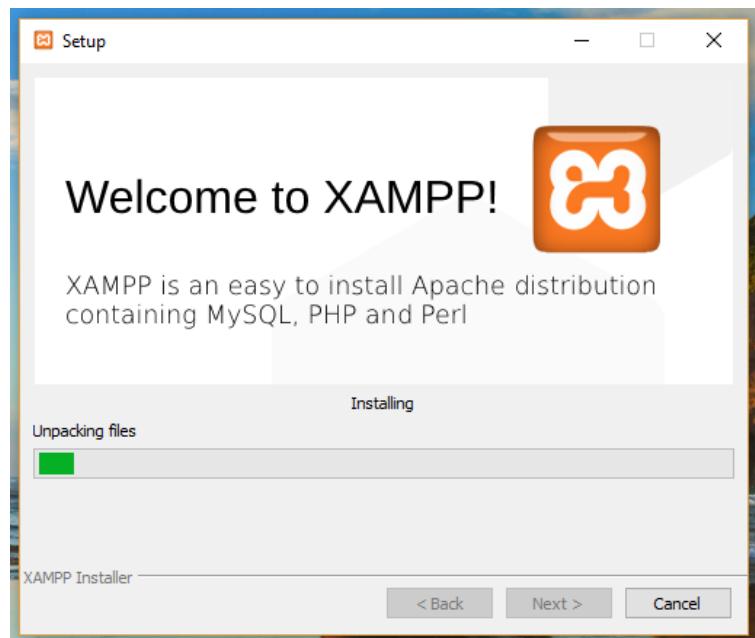
Gambar 2.6 Pilih komponen Installasi XAMPP

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 Installasi XAMPP

6. Tungu beberapa menit hingga progress installasi selesai dan berhasil.



Gambar 2.7 Progress Installasi XAMPP

7. Setelah selesai anda sudah bisa menggunakanya, 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 programer 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.

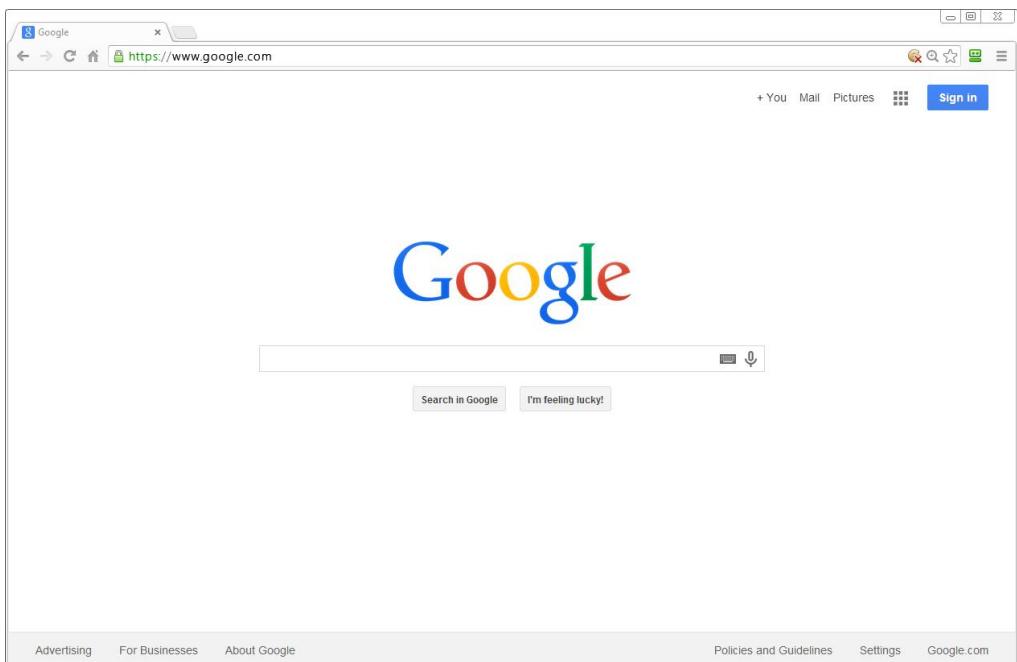
2 . 2 Web Browser

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* :

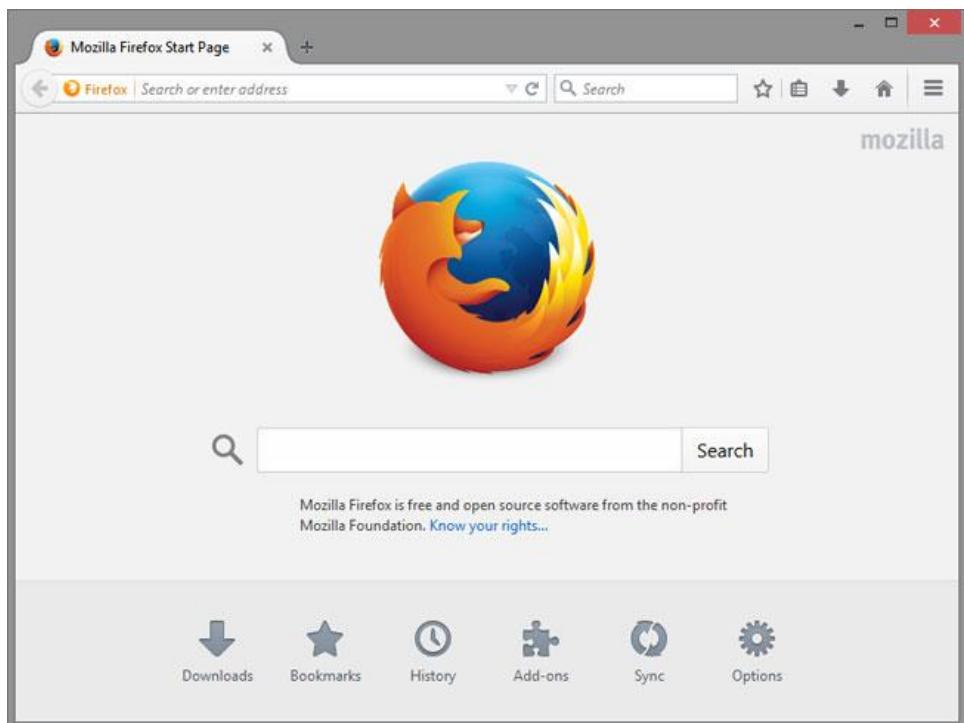
2.2.1 Google Chrome



Gambar 2.8 Google Chrome

Google Chrome merupakan browser yang di keluarkan oleh Google dan sudah banyak sekali yang menggunakanya khususnya negara-negara maju dan berkembang. dan hampir semua penggunanya mengatakan bahwa browser ini browser *fast and friendly*. Browser ini bisa anda dapat dengan mendownload di website resminya, 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 hal 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

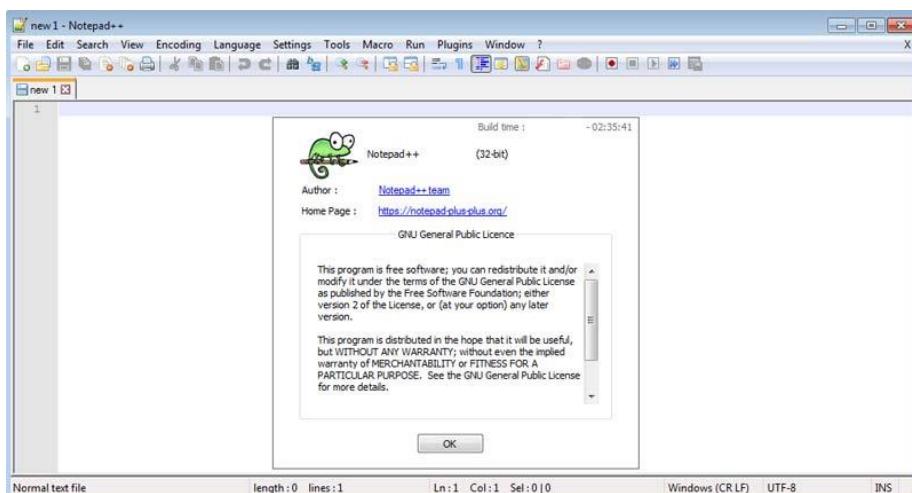
Safari merupakan browser besutan dari perusahaan raksasa *Apple*, termasuk yang paling populer dikarenakan kalangan dari pengguna *Apple* hampir semuanya menggunakan browser ini selain bentuknya yang elegan browser ini juga memiliki kecepatan *load* sebuah halaman website. Browser ini bisa anda dapat 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 programer 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 :

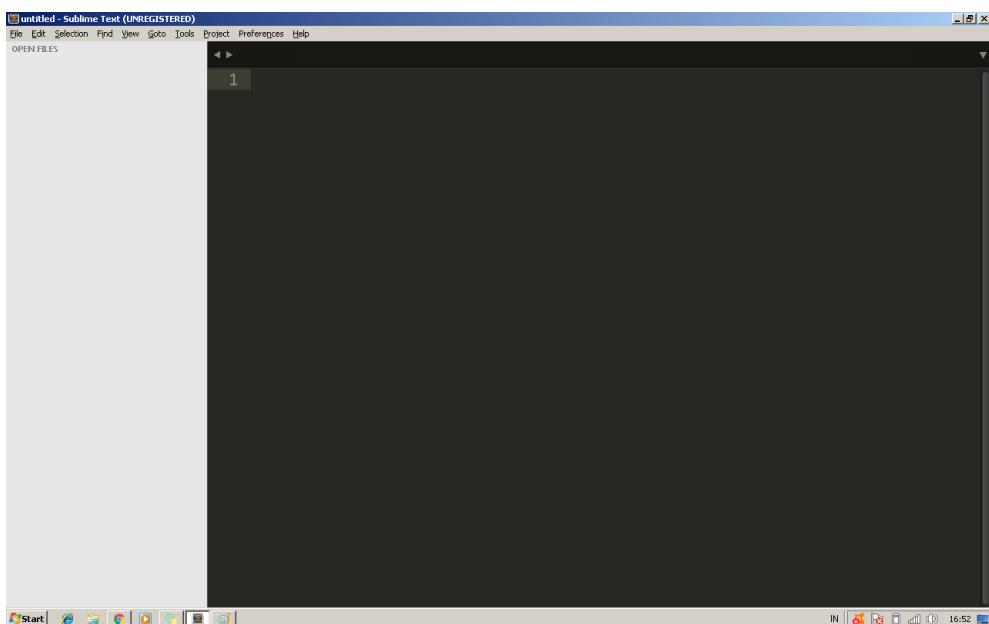
2 . 3 . 1 Notepad ++



Gambar 2.10 Notepad ++

Merupakan *Text Editor* yang tersedia untuk sistem operasi *Windows*. Dengan fitur-fitur yang menarik dan mendukung para programer dalam menulis *source code*. Merupakan *software* yang banyak diminati programer 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

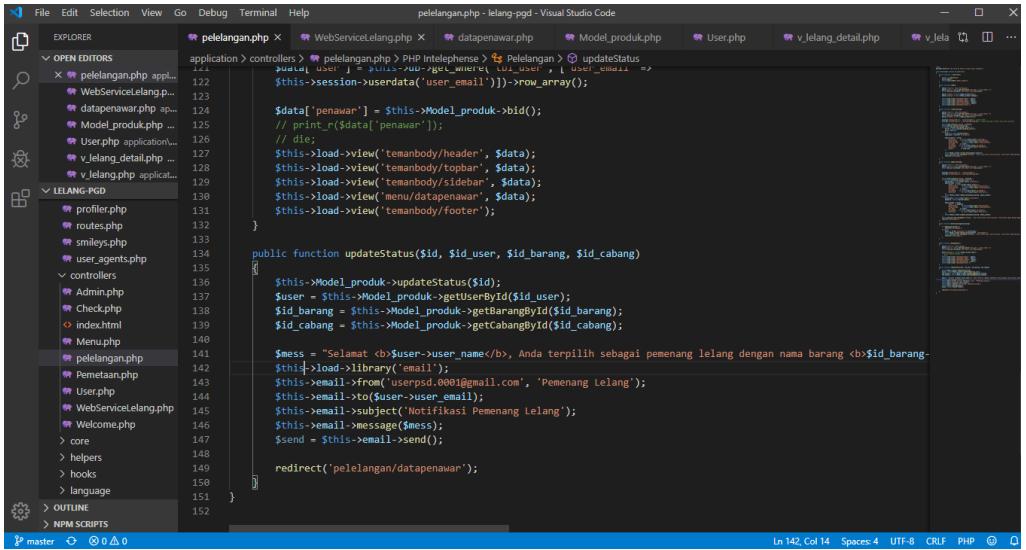


Gambar 2.11 Sublime Text

Sublime merupakan *text editor* yang populer di kalangan programer, tampilanya yang simpel namun enak di lihat serta fleksibel dan *powerfull*, *Software* atau aplikasi ini tersedia untuk sistem 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)



```
File Edit Selection View Go Debug Terminal Help pelelangan.php - lelang-pgd - Visual Studio Code

OPEN EDITORS
  pelelangan.php x  WebServiceLelang.php x  datapenawar.php x  Model_produk.php x  User.php x  v_Jelang_detail.php x  v_lela
  application > controllers > pelelangan.php > PHP Intelliphense > Pelelangan > updateStatus
  $data['user'] = $this->load->user->getWhere('id_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);

    $mess = "Selamat <b></b>$User->user_name</b>, Anda terpilih sebagai pemenang lelang dengan nama barang <b>$id_barang</b>";
    $this->load->library('email');
    $this->email->from('userspd.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');
```

Gambar 2.12 VS Code

VS Code merupakan *text editor* yang populer saat ini di kalangan programer 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 programer 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 membagun 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.

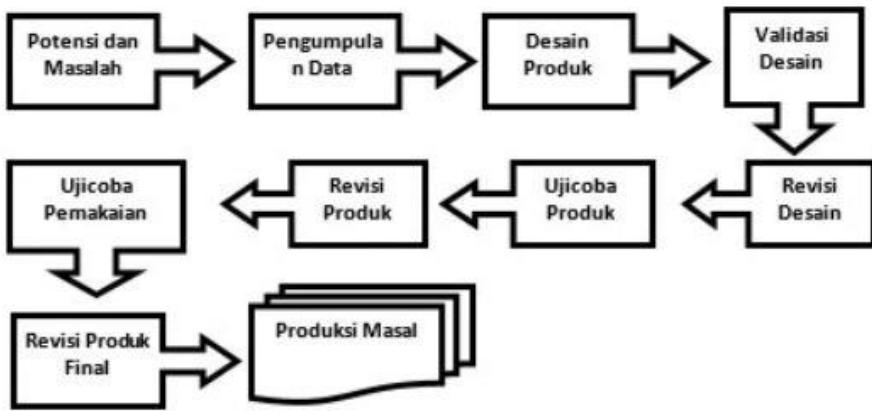
3.1 Penjelasan R&D



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*, pengembangan 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,

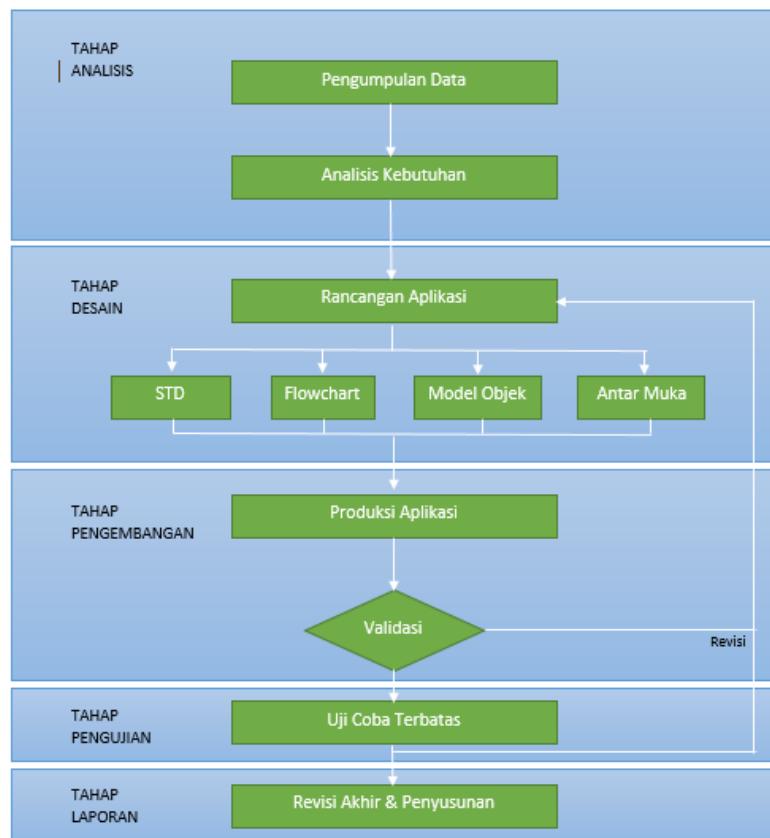
1. Potensi dan masalah. Penelitian ini didapatkan berdasarkan latar belakang masalah yang ada. Potensi merupakan sesuatu yang memiliki nilai tambah. Masalah adalah merupakan penyimpangan antara sesuatu yang diharapkan dengan apa yang sudah terjadi.
2. Mengumpulkan informasi. Setelah potensi permasalahan 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 bagun.

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.

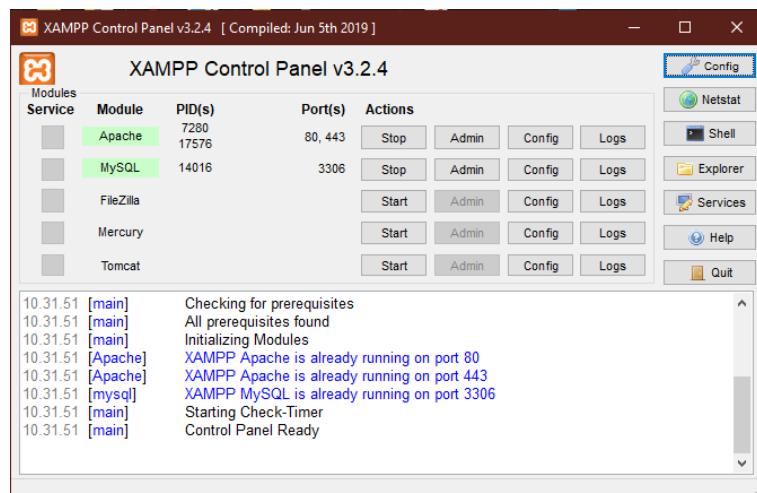
BAB 4

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.

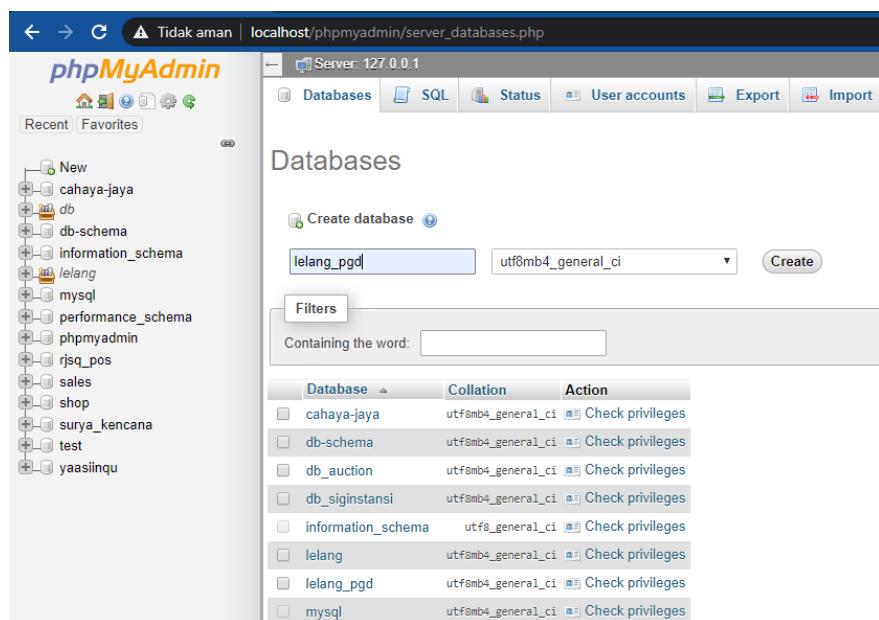
4.1 Langkah-langkah Membuat Database Pelelangan

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



Gambar 4.1 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 4.2 Pembuatan *Database*

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

Pembuatan tabel **bid**,

The screenshot shows the phpMyAdmin interface for creating a table named 'bid'. The left sidebar lists databases and tables under 'lelang_pg', including 'cabang', 'pelelangan', 'pemetaan', 'tbl_user', 'tbl_user_token', 'user_accessmenu', 'user_menu', 'user_role', and 'user_submenu'. The main panel displays the 'Table structure' tab for the 'bid' table. The table has seven columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_lelang_bid	int(11)			No	None	AUTO_INCREMENT		Change Drop More
2	id_barang	int(11)			No	None			Change Drop More
3	id_cabang	int(11)			No	None			Change Drop More
4	id_user	int(11)			No	None			Change Drop More
5	harga_lelang	double			No	None			Change Drop More
6	status	varchar(30)	utf8mb4_general_ci		No	pending			Change Drop More
7	tanggal	datetime			No	None			Change Drop More

Below the table structure, there are buttons for 'Check all', 'With selected:', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Fulltext', and 'Add to central color'. A dropdown menu shows '1 column(s)' and 'after tanggal', with a 'Go' button.

Gambar 4.3 Pembuatan Tabel bid

Pembuatan tabel **cabang**,

The screenshot shows the phpMyAdmin interface for creating a table named 'cabang'. The left sidebar lists databases and tables under 'lelang_pg', including 'bid', 'cabang', 'pelelangan', 'pemetaan', 'tbl_user', 'tbl_user_token', 'user_accessmenu', 'user_menu', 'user_role', and 'user_submenu'. The main panel displays the 'Table structure' tab for the 'cabang' table. The table has two columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT		Change Drop More
2	cabang	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More

Below the table structure, there are buttons for 'Check all', 'With selected:', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Fulltext', and 'Add to central color'. A dropdown menu shows '1 column(s)' and 'after cabang', with a 'Go' button. At the bottom, there is a section for 'Indexes' with a table:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	Id	12	A	No	

Below the indexes table, there is a button 'Create an index on 1 columns Go'.

Gambar 4.4 Pembuatan Tabel cabang

Pembuatan tabel pelelangan,

The screenshot shows the phpMyAdmin interface for creating a new table named 'pelelangan'. The left sidebar displays the database structure with tables like 'id_barang', 'nama_barang', 'harga_barang', 'id_cabang', 'tgl_buka', 'gambar', and 'deskripsi'. The main area shows the table structure with the following columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_barang	int(11)	utf8mb4_general_ci		No	None		AUTO_INCREMENT	Change Drop More
2	nama_barang	varchar(120)	utf8mb4_general_ci		No	None			Change Drop More
3	harga_barang	double			No	None			Change Drop More
4	id_cabang	int(11)			No	None			Change Drop More
5	tgl_buka	int(11)			No	None			Change Drop More
6	gambar	text	utf8mb4_general_ci		No				Change Drop More
7	deskripsi	text	utf8mb4_general_ci		No				Change Drop More

Below the table structure, there are buttons for 'Check all', 'With selected:', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Fulltext', 'Add to central columns', 'Print', 'Propose table structure', 'Track table', 'Move columns', 'Normalize', 'Add', 'column(s)', 'after deskripsi', and 'Go'.

Gambar 4.5 Pembuatan Tabel pelelangan

Pembuatan tabel pemetaan,

The screenshot shows the phpMyAdmin interface for creating a new table named 'pemetaan'. The left sidebar displays the database structure with tables like 'id_cabang', 'nama_cabang', 'alamat', 'kota', 'no_telp', 'image', 'latitude', 'longitude', and 'deskripsi'. The main area shows the table structure with the following columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_cabang	int(11)	utf8mb4_general_ci		No	None		AUTO_INCREMENT	Change Drop More
2	nama_cabang	varchar(120)	utf8mb4_general_ci		No	None			Change Drop More
3	alamat	text	utf8mb4_general_ci		No				Change Drop More
4	kota	varchar(120)	utf8mb4_general_ci		No	None			Change Drop More
5	no_telp	varchar(12)	utf8mb4_general_ci		No	None			Change Drop More
6	image	varchar(120)	utf8mb4_general_ci		No	None			Change Drop More
7	latitude	varchar(120)	utf8mb4_general_ci		No	None			Change Drop More
8	longitude	varchar(120)	utf8mb4_general_ci		No	None			Change Drop More
9	deskripsi	text	utf8mb4_general_ci		No				Change Drop More

Below the table structure, there are buttons for 'Check all', 'With selected:', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Fulltext', 'Add to central columns', 'Print', 'Propose table structure', 'Track table', 'Move columns', 'Normalize', 'Add', 'column(s)', and 'Go'.

Gambar 4.6 Pembuatan Tabel pemetaan

Pembuatan tabel **tbl_user**,

The screenshot shows the phpMyAdmin interface for creating a table named 'tbl_user'. The left sidebar lists databases and tables under 'lolang_pgsql'. The main area shows the 'Structure' tab for 'tbl_user' with the following columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	user_id	int(11)	latin1_swedish_ci	No	None			AUTO_INCREMENT	Change Drop More
2	user_name	varchar(120)	latin1_swedish_ci	No	None				Change Drop More
3	user_email	varchar(120)	latin1_swedish_ci	No	None				Change Drop More
4	image	varchar(120)	latin1_swedish_ci	No	None				Change Drop More
5	password	varchar(120)	latin1_swedish_ci	No	None				Change Drop More
6	role_id	int(11)		No	None				Change Drop More
7	is_active	int(1)		No	None				Change Drop More
8	date_created	int(11)		No	None				Change Drop More

Below the table structure, there are buttons for 'Check all', 'With selected:', and various actions like 'Browse', 'Change', 'Drop', etc. The bottom navigation bar includes 'Print', 'Propose table structure', 'Track table', 'Move columns', and 'Normalize'.

Gambar 4.7 Pembuatan Tabel **tbl_user**

Pembuatan tabel **tbl_user_token**,

The screenshot shows the phpMyAdmin interface for creating a table named 'tbl_user_token'. The left sidebar lists databases and tables under 'lolang_pgsql'. The main area shows the 'Structure' tab for 'tbl_user_token' with the following columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)		No	None			AUTO_INCREMENT	Change Drop More
2	email	varchar(128)	utf8mb4_general_ci	No	None				Change Drop More
3	token	varchar(128)	utf8mb4_general_ci	No	None				Change Drop More
4	date_created	int(11)		No	None				Change Drop More

Below the table structure, there are buttons for 'Check all', 'With selected:', and various actions like 'Browse', 'Change', 'Drop', etc. The bottom navigation bar includes 'Print', 'Propose table structure', 'Track table', 'Move columns', and 'Normalize'. A separate 'Indexes' section at the bottom shows a single primary key index:

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	4	A	No	

Gambar 4.8 Pembuatan Tabel **tbl_user_token**

Pembuatan tabel user_accessmenu,

The screenshot shows the phpMyAdmin interface for creating a table named 'user_accessmenu'. The left sidebar shows the database structure under 'lelang_pg' schema, including tables like 'tbl_user', 'tbl_user_token', 'user_accessmenu', 'user_menu', 'user_role', and 'user_submenu'. The main panel displays the table structure with three columns: 'id', 'role_id', and 'menu_id'. The 'id' column is defined as int(11) with AUTO_INCREMENT, 'role_id' as int(11), and 'menu_id' as int(11). An index named 'PRIMARY' is present on the 'id' column. Below the table structure, there is a section for adding columns and another for indexes.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	role_id	int(11)			No	None			Change Drop More
3	menu_id	int(11)			No	None			Change Drop More

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	7	A	No	

Gambar 4.9 Pembuatan Tabel user_accessmenu

Pembuatan tabel user_menu,

The screenshot shows the phpMyAdmin interface for creating a table named 'user_menu'. The left sidebar shows the database structure under 'lelang_pg' schema, including tables like 'tbl_user', 'tbl_user_token', 'user_accessmenu', 'user_menu', 'user_role', and 'user_submenu'. The main panel displays the table structure with two columns: 'id' and 'menu'. The 'id' column is defined as int(11) with AUTO_INCREMENT, and the 'menu' column is defined as varchar(120) with collation latin1_swedish_ci. An index named 'PRIMARY' is present on the 'id' column. Below the table structure, there is a section for adding columns and another for indexes.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	menu	varchar(120)	latin1_swedish_ci		No	None			Change Drop More

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	5	A	No	

Gambar 4.10 Pembuatan Tabel user_menu

Pembuatan tabel **user_role**,

The screenshot shows the phpMyAdmin interface for creating a new table named 'user_role'. The left sidebar lists databases and tables under 'lolang_pgsql'. The main area shows the table structure with two columns: 'id' (int(11)) and 'role' (varchar(120)). An index 'PRIMARY' is defined on the 'id' column.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)	latin1_swedish_ci		No	None		AUTO_INCREMENT	Change Drop More
2	role	varchar(120)	latin1_swedish_ci		No	None			Change Drop More

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	2	A	No	

Gambar 4.11 Pembuatan Tabel **user_role**

Pembuatan tabel **user_submenu**,

The screenshot shows the phpMyAdmin interface for creating a new table named 'user_submenu'. The left sidebar lists databases and tables under 'lolang_pgsql'. The main area shows the table structure with six columns: 'id' (int(11)), 'menu_id' (int(11)), 'title' (varchar(120)), 'url' (varchar(120)), 'icon' (varchar(120)), and 'is_active' (int(1)).

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)	latin1_swedish_ci		No	None		AUTO_INCREMENT	Change Drop More
2	menu_id	int(11)	latin1_swedish_ci		No	None			Change Drop More
3	title	varchar(120)	latin1_swedish_ci		No	None			Change Drop More
4	url	varchar(120)	latin1_swedish_ci		No	None			Change Drop More
5	icon	varchar(120)	latin1_swedish_ci		No	None			Change Drop More
6	is_active	int(1)			No	None			Change Drop More

Gambar 4.12 Pembuatan Tabel **user_submenu**

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

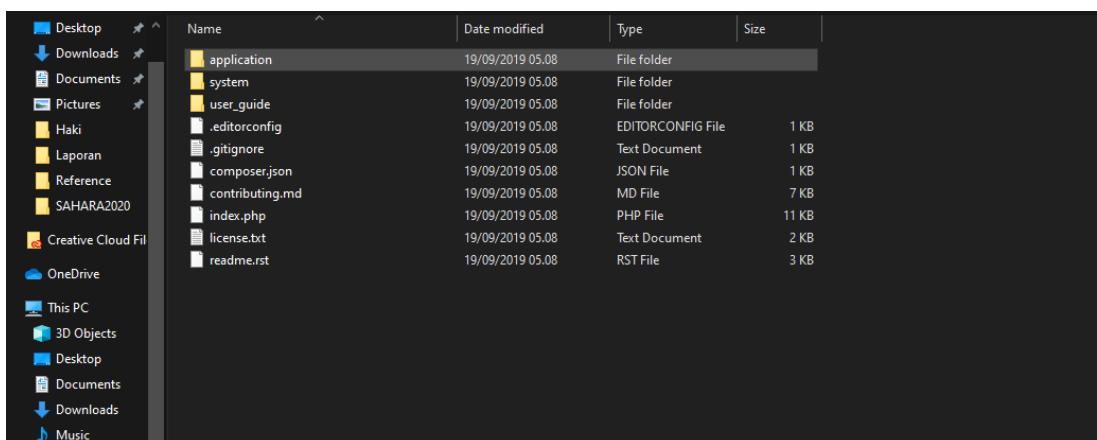
BAB 5

Pembuatan Aplikasi

Setelah selesai membuat semua tabel yang dibutuhkan untuk *database* selanjutnya adalah tahap pembuatan aplikasi, diawali dengan installasi *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

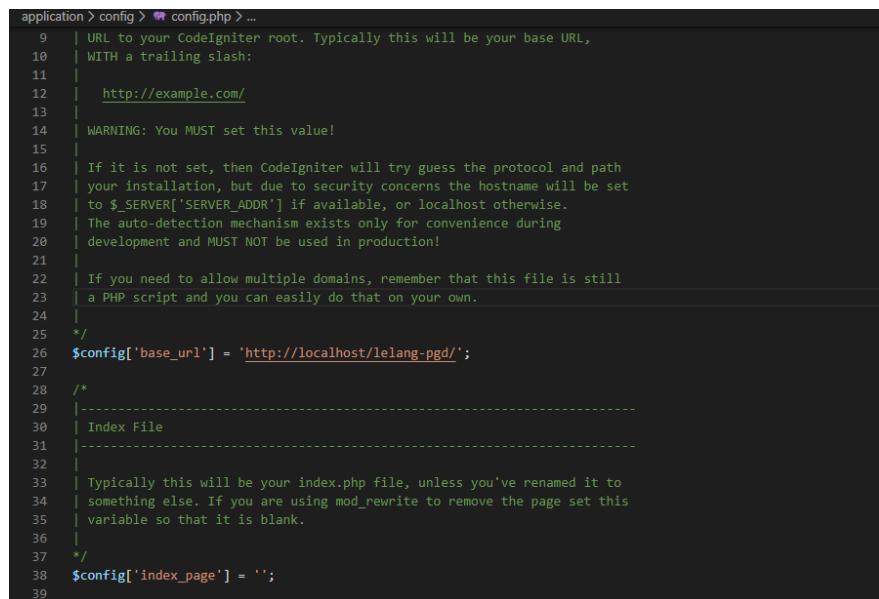
Langkah pertama pastikan anda sudah mendownload *codeigniter* di *website* resminya pada link berikut, <https://codeigniter.com/download> kita akan menggunakan pada versi *codeigniter 3*. Ekstrak file tersebut.



Gambar 5.1 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 | Index File
30 | -----
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 5.2 Konfigurasi config.php

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

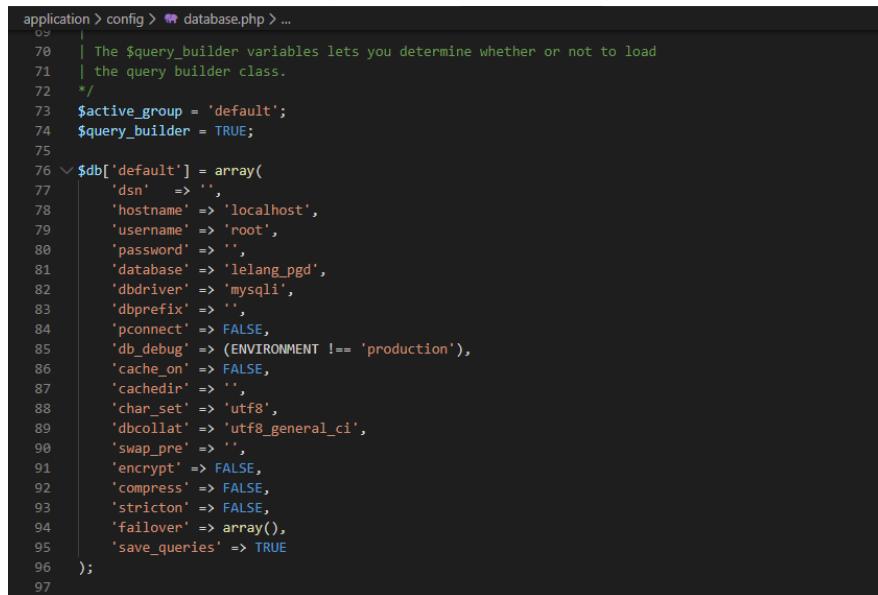
The image shows two code editors side-by-side, both displaying the contents of the `autoload.php` file located in the `application/config` directory. The code is written in PHP and defines various autoload configurations.

```
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 5.3 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 > ...
...
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 5.4 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
```

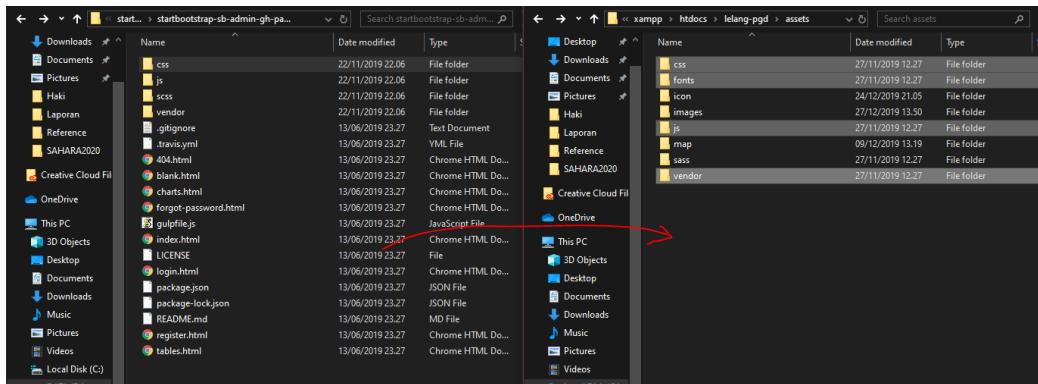
Gambar 5.5 konfigurasi routes.php

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 installasi & konfigurasi *codeigniter* untuk *back-end*. Pada bagian ini penulis akan memberikan cara installasi & konfigurasi pada sisi *front-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 :

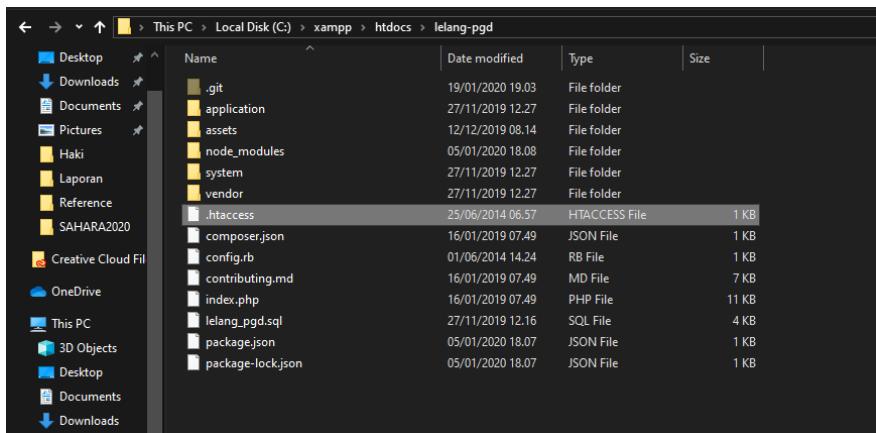
Setelah didownload bootstrapnya, ekstrak *file* tersebut dan pindahkan beberapa folder seperti **css,js,fonts & vendor** ke folder htdocs kalian dengan membuat folder baru dan beri nama **assets**, contohnya seperti berikut, (C:\xampp\htdocs\lelang-pgd\assets)



Gambar 5.6 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 5.7 *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 kefokusana 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 terlewati 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 dibawa 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_ur
l('assets/icon/favicon.png') ?&gt;" /&gt;
&lt;link href="<?= base_url(); ?&gt;assets/vendor/fontaw
esome-
free/css/all.min.css" rel="stylesheet" type="text/css"
&gt;

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

&lt;/head&gt;

&lt;body class="bg-dark"&gt;</pre
```

check_footer.php

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

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

&lt;/html&gt;</pre
```

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

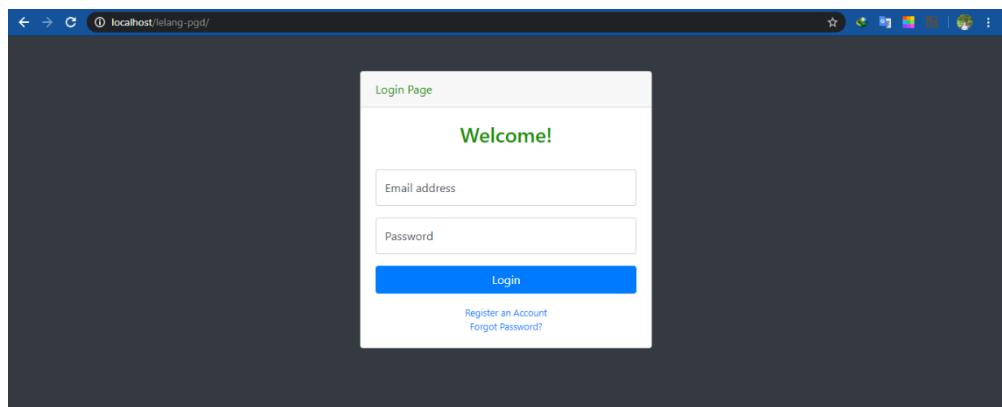
```
<div class="container">

    <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') ?&gt;"&gt;
                    &lt;div class="form-group"&gt;
                        &lt;div class="form-label-group"&gt;
                            &lt;input type="email" id="email" class="form-control" name="email" placeholder="Enter Your Email" a</pre
```

```
        autofocus="autofocus" value=<?= set_value('email'); ?>
    >
        <label for="email">Email address</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="password" class="form-control" name="password" placeholder="Password">
        <label for="password">Password</label>
        <?php echo form_error('password', '<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 5.8 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 tambahanya.

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|matches[password1]');

if ($this->form_validation->run() == false) {
    $data['title'] = 'Pegadaian Lelang Registration';
    $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($email),
        'image' => 'default.png',
        'password' => password_hash($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');
    }
}
```

Pada intinya fungsi diatas menjalakan sebuah perintah dengan kondisi berisi **session** untuk menyimpan informasi yang dibutuhkan oleh aplikasi dan dapat dibagi keseluruh halaman web. Lalu terdapat **form_validation** yang berisi rules/aturan ketika user melakukan penginputan seperti **trim** digunakan untuk menghapus spasi atau karakter dari sebuah string, **required** artinya kolom yang digunakan untuk penginputan tidak boleh di kosongkan, **valid_email** artinya email yang di inputkan oleh user harus email asli sesuai dengan format email sebenarnya, **is_unique** artinya email yang dininputkan untuk registrasi tidak boleh sama, jika email yang di inputkan sama dengan yang ada pada database sistem akan menolaknya, dan pada rules password ada **min_length[8]** artinya

user ketika penginputan password tidak boleh kurang dari 8 karakter, **matches[password2]** artinya konfirmasi penginputan password yang kedua harus sama dengan penginputan password yang pertama atau sebaliknya.

Setelah semua rules sudah benar, sistem akan melakukan input kedalam database dengan mengisi gambar/foto profil dari user secara otomatis dengan **default.png**, mengisi **role_id** dengan '2' yang artinya user registrasi sebagai member, **is_active '0'** yang artinya akun belum active dan **date_created time()** yang artinya user telah di daftarkan dengan tanggal dan waktu ketika user membuatnya. Lalu sistem akan mengirim sebuah token dengan fungsi tambahan **_sendEmail**, serta **base_encode64(random byte [32])** artinya mengirim token karakter secara acak dengan panjang 32 digit. Yang nantinya akan user terima berupa email masuk untuk mengaktifkan akun agar bisa login pada sistem.

Setelah itu kita akan membuat front-end atau antarmuka dari halaman register, source nya hampir sama seperti login tetapi ini hanya ada penambahan beberapa kolom saja, buat file baru beri nama **register.php** simpan pada folder **check** yang berada di dalam folder **view**.

Berikut source code untuk tampilan register,

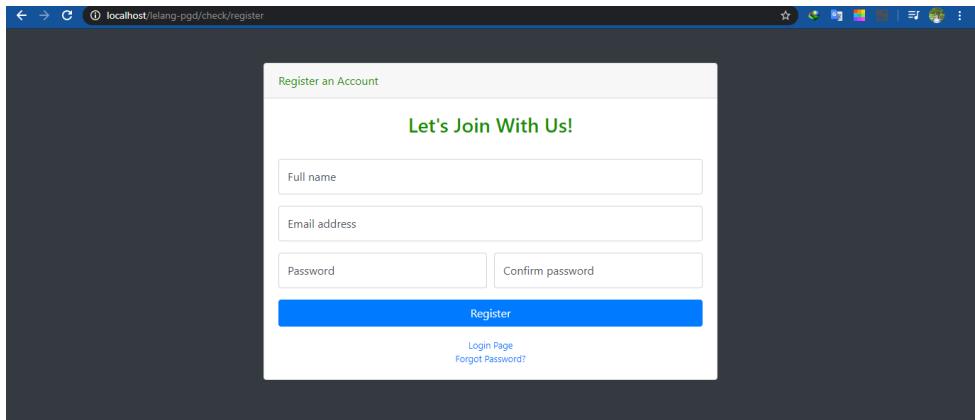
```
<body class="bg-dark">
```

```
<div class="container">
    <div class="card card-register mx-auto mt-5">
        <div class="card-
header">Register an Account</div>
        <div class="card-body">
            <form class="user" method="post" action=<?= base_url('check/register'); ?>>
                <div class="text-center">
                    <h3>Let's Join With Us!</h3>
                </div><br>
                <div class="form-group">
                    <div class="form-label-group">
                        <input type="text" id="name" name="name" class="form-
control" placeholder="Full name" value=<?= set_value(
'name'); ?>>
                            <label for="name">Full nam
e</label>
                            <?php echo form_error('nam
e', '<small class="text-danger pl-
3">', '</small>'); ?>
                    </div>
                </div>
                <div class="form-group">
                    <div class="form-label-group">
                        <input type="text" id="ema
il" name="email" class="form-
control" placeholder="Email address" value=<?= set_va
lue('email'); ?>>
                            <label for="email">Email a
ddress</label>
                            <?php echo form_error('ema
il', '<small class="text-danger pl-
3">', '</small>'); ?>
                    </div>
                </div>
                <div class="form-group">
```

```
        <div class="form-row">
            <div class="col-md-6">
                <div class="form-
label-group">
                    <input type="passw
ord" id="password1" name="password1" class="form-
control" placeholder="Password">
                    <label for="passwo
rd1">Password</label>
                    <?php echo form_er
ror('password1', '<small class="text-danger pl-
3">', '</small>'); ?>
                </div>
            </div>
            <div class="col-md-6">
                <div class="form-
label-group">
                    <input type="passw
ord" id="password2" name="password2" class="form-
control" placeholder="Confirm password">
                    <label for="passwo
rd2">Confirm password</label>
                </div>
            </div>
            <button class="btn btn-
primary btn-user btn-
block" type="submit">Register</button>
        </form>
        <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>
```

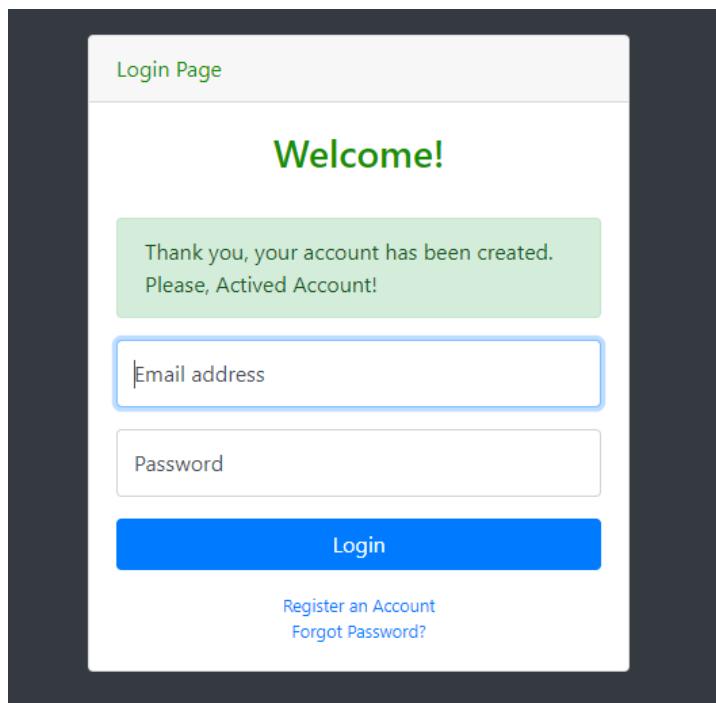
```
</div>
</div>
</div>
```

Dari source code diatas jika dijalankan pada web browser maka tampilanya akan seperti berikut,



Gambar 5.9 Antarmuka Register

Setelah user melakukan pendaftaran akan ada notif bahwa akun sudah dibuat tetapi belum aktif, Berikut tampilan halaman setelah pendaftaran akun,



Gambar 5.10 Registrasi

Fungsi _sendEmail

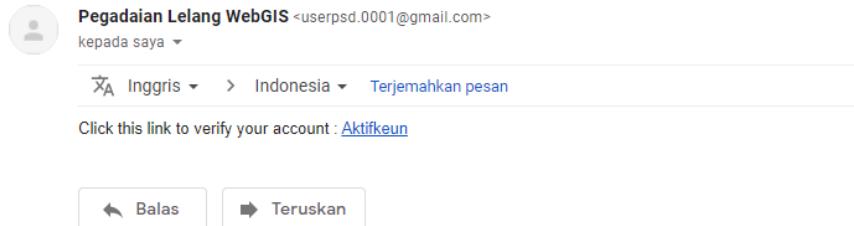
Masih di file yang sama yaitu **check.php** tambahkan kembali fungsi baru setelah fungsi register berikut source nya,

```
private function _sendEmail($token, $type)
```

```
{  
  
    $this->load->library('email');  
    $this->email-  
>from('userpsd.0001@gmail.com', 'Pegadaian Lelang WebG  
IS');  
    $this->email->to($this->input->post('email'));  
  
    if ($type == 'verify') {  
  
        $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) . '">Ak  
tifkeun</a>');  
        $em = $this->email->send();  
    }  
  
}
```

Fungsi di atas mengirimkan pesan melalui gmail yang sudah terisi otomatis pengirim, subjek dan isi pesannya yang berisi link untuk verifikasi email menjadi aktif agar bisa digunakan.

Account Verification Pegadaian Lelang WebGIS Kotak Masuk ×



Gambar 5.11 Verifikasi Gmail

Ketika proses pesan sudah masuk ke gmail pendaftar akan melewati sebuah fungsi yang bernama **verify()** untuk mengubah data dari user yang sebelumnya belum aktif menjadi aktif sehingga akun dapat digunakan untuk login masuk sistem.

Fungsi verify()

```
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 activated, Please Login!</div>');
                    redirect('check');
                } else {

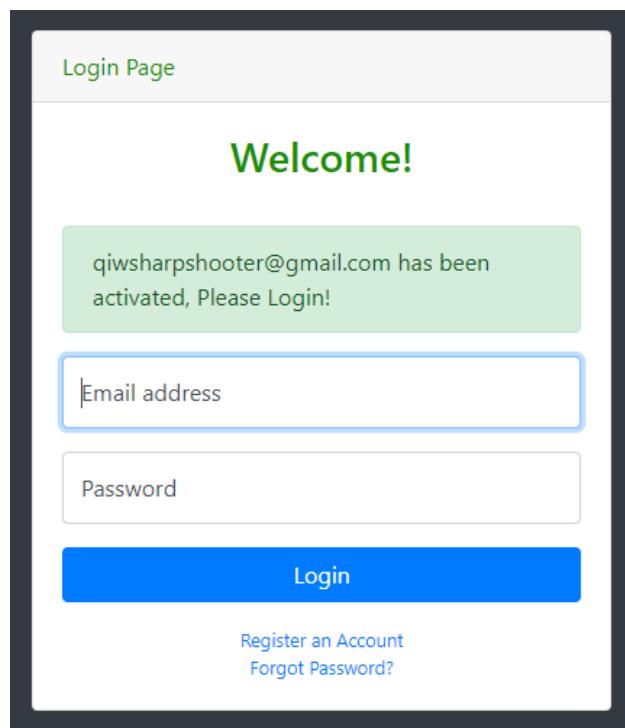
                    $this->db-
>delete('tbl_user', ['user_email' => $email]);
                    $this->db-
>delete('tbl_user_token', ['email' => $email]);

                    $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');
}
}
```

Pada intinya source diatas berfungsi mengubah data sebelumnya yang telah di inputkan oleh user. ketika user mengklik link aktivasi yang dikirim melalui email, secara otomatis data akan update dan kondisi dari field **is_active** akan berubah dari '0' menjadi '1' yang artinya akun sudah aktif dan dapat digunakan untuk login pada sistem.

Berikut tampilan setelah akun sudah di aktifkan melalui pesan di gmail,



Gambar 5.12 Email Aktif

Masih di file controller yang sama **check.php** kita akan langsung membuat fungsi logout pada sistem/aplikasi, berikut source,

Fungsi logout()

```
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');
}

}
```

Source logout ini berfungsi untuk mengeluarkan akun yang telah login sebelumnya dan mengembalikanya kembali ke halaman login.

Sebelumnya kita harus membuat sebuah file baru di folder config, beri nama **email.php** berikut source nya,

email.php

```
<?php
$config = [
    'mailtype'  => 'html',
    'charset'   => 'utf-8',
    'protocol'  => 'smtp',
    'smtp_host' => 'smtp.gmail.com',
    'smtp_user' => 'userpsd.0001@gmail.com', // Email
    'gmail
    'smtp_pass'  => 'pegadaian1', // Password gmail
    'smtp_crypto' => 'ssl',
```

```
'smtp_port' => 465,  
'crlf' => "\r\n",  
'newline' => "\r\n"  
];
```

Source diatas berfungsi untuk membantu jalanya pengiriman pesan melalui gmail menggunakan protokol smtp, ubah **smtp_user** dan **smtp_pass** dengan email dan password kalian yang akan digunakan sebagai pengirim pesan. Sebelumnya pastikan email kalian telah di aktifkan *less secure apps access* silahkan kunjungi link berikut, <https://myaccount.google.com/lesssecureapps>. Lalu ubah statusnya menjadi ON, seperti gambar berikut,

 Account

← Less secure app access

Some apps and devices use less secure sign-in technology, which makes your account more vulnerable. You can turn off access for these apps, which we recommend, or turn on access if you want to use them despite the risks. [Learn more](#)

Allow less secure apps:

Gambar 5.13 *Less Secure Apps Access*

5.5 Pembuatan Halaman Dashboard Admin

Buat file baru di folder controllers dengan nama **Admin.php** pada file ini kita akan membuat halaman admin dan pemberian hak akses menu pada user atau role.

Admin.php

```
<?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>');
    }
}

```

Terdapat empat fungsi didalam file **Admin.php** diantaranya adalah,

Index(), fungsi ini bertujuan sebagai index atau halaman utama dari admin, yang berisi menu-menu yang dapat diolah oleh admin sendiri.

```
$data['total_cabang'] = $this->Menu_model->hitungcabang();
```

Source diatas berfungsi untuk memanggil sebuah model yang membawa variabel berisi parameter,

Buat sebuah file baru, beri nama **Menu_model.php** simpan file tersebut di dalam folder models, berikut source dari **Menu_model.php**

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

class Menu_model extends CI_Model
{
    public function hitungcabang()
    {
        $query = $this->db->get('cabang');
        if ($query->num_rows() > 0) {
            return $query->num_rows();
        } else {
            return 0;
        }
    }
}
```

Source diatas berfungsi untuk menghitung jumlah data yang ada pada database **cabang**, sebelumnya isi manual terlebih dahulu data pada database **cabang** seperti berikut,

		+ Options			
		← →		id	cabang
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	Ujung Berung
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	Buah Batu
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	Ciparay
<input type="checkbox"/>	 Edit	 Copy	 Delete	4	Kiaracondong
<input type="checkbox"/>	 Edit	 Copy	 Delete	5	Metro
<input type="checkbox"/>	 Edit	 Copy	 Delete	8	Purwakarta
<input type="checkbox"/>	 Edit	 Copy	 Delete	9	Pasar Kordon
<input type="checkbox"/>	 Edit	 Copy	 Delete	10	Suci
<input type="checkbox"/>	 Edit	 Copy	 Delete	11	Sayati
<input type="checkbox"/>	 Edit	 Copy	 Delete	12	Padasuka
<input type="checkbox"/>	 Edit	 Copy	 Delete	13	Veteran
<input type="checkbox"/>	 Edit	 Copy	 Delete	14	Soreang
<input type="checkbox"/>	 Edit	 Copy	 Delete	15	Banjaran

Gambar 5.14 Isi Database Cabang

Kembali ke file controller **Admin.php** pada fungsi **index()** terdapat source seperti berikut,

```
$this->load->view('temanbody/header', $data);
```

```

$this->load->view('temanbody/topbar', $data);

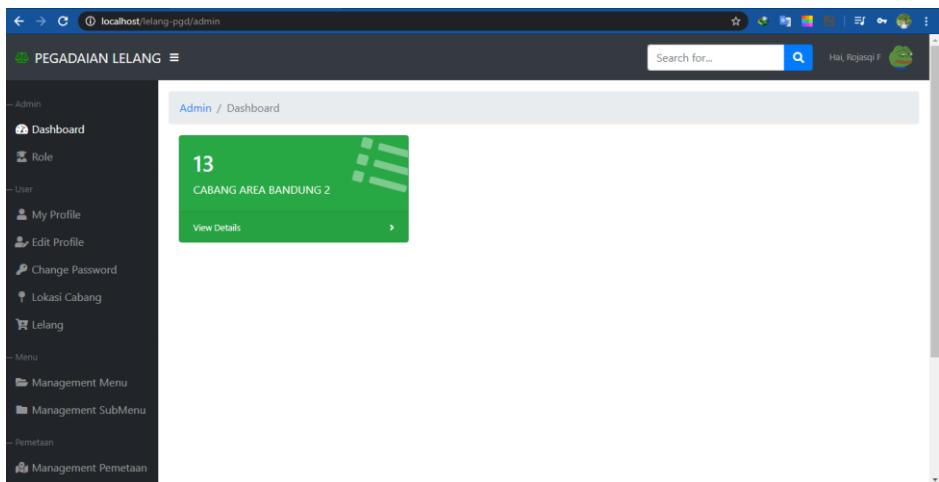
$this->load->view('temanbody/sidebar', $data);

$this->load->view('admin/index', $data);

$this->load->view('temanbody/footer');

```

Source diatas berfungsi me-load atau memanggil bagian-bagian view seperti topbar,sidebar,index dan footer yang membawa variabel data berisi parameter. Berikut tampilan dari view index.php dari admin,



Gambar 5.15 Antarmuka Index Admin

Berikut source code dari **index.php**

```

<div id="content-wrapper">

    <div class="container-fluid">

```

```
<!-- Breadcrumbs-->
<ol class="breadcrumb">
    <li class="breadcrumb-item">
        <a href="">Admin</a>
    </li>
    <li class="breadcrumb-
item active">Dashboard</li>
</ol>

<!-- Page Content -->
<div class="col-xl-4 col-sm-6 mb-3">
    <div class="card text-white bg-success o-
hidden h-100">
        <div class="card-body">
            <div class="card-body-icon">
                <i class="fas fa-fw fa-
list"></i>
            </div>
            <div class="mr-5">
                <h2><?= $total_cabang; ?></h2>
CABANG AREA BANDUNG 2
            </div>
            </div>
            <a class="card-footer text-
white clearfix small z-1" href="#">
                <span class="float-
left">View Details</span>
                <span class="float-right">
                    <i class="fas fa-angle-
right"></i>
                </span>
            </a>
        </div>
    </div>
</div>
```

```
</div>
<!-- /.container-fluid -->
```

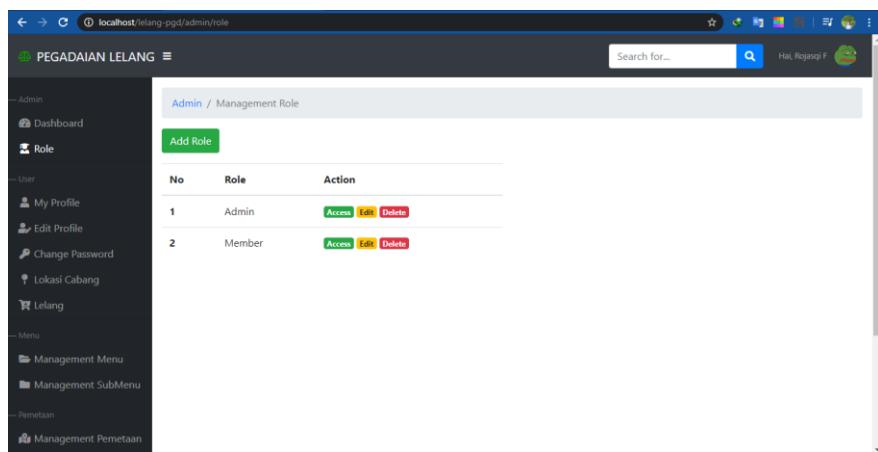
Fungsi **role()** bertujuan untuk menampilkan role dari database **user_role** yang sudah di relasikan dengan tabel **tbl_user**. sehingga mengetahui user yang login, apakah admin atau member, berikut isi dari database **user_role**



		+ Options	← →	▼	id	role
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	Admin	
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	Member	

Gambar 5.15 Tabel Role

Berikut adalah tampilan dan source code view dari menu **role.php** pada admin,



The screenshot shows a web application interface for managing roles. The URL is `localhost/lelang-psjd/admin/role`. The page title is "PEGADAIAN LELANG". On the left, there's a sidebar with navigation links for Admin, User, Menu, and Pemetaan. The main content area is titled "Admin / Management Role" and contains a table with two rows:

No	Role	Action
1	Admin	Access Edit Delete
2	Member	Access Edit Delete

Gambar 5.16 Antarmuka Role

role.php

```
<div id="content-wrapper">

    <div class="container-fluid">

        <!-- Breadcrumbs -->
        <ol class="breadcrumb">
            <li class="breadcrumb-item">
                <a href="">Admin</a>
            </li>
            <li class="breadcrumb-item active">Management Role</li>
        </ol>

        <!-- Page Content -->
        <div class="row">
            <div class="col-lg-6">
```

```
        <?= form_error('menu', '<div class="alert alert-danger" role="alert">', '</div>'); ?>

        <?= $this->session-
>flashdata('message'); ?>

        <a href="" class="btn btn-success mb-
3" data-toggle="modal" data-
target="#addRole">Add Role</a>
        <table class="table table-hover">
            <thead>
                <tr>
                    <th scope="col">No</th>
                    <th scope="col">Role</th>
                    <th scope="col">Action</th>
                </tr>
            </thead>
            <tbody>
                <?php $q = 1; ?>
                <?php foreach ($role as $r) :
?>
                <tr>
                    <th scope="row"><?= $q
; ?></th>
                    <td><?= $r['role']; ?>
</td>
                    <td>
                        <a href=<?= base_
url('admin/roleaccess/') . $r['id']; ?>" class="badge
badge-success">Access</a>
                        <a href="" class="badge
badge-warning">Edit</a>
                        <a href="" class="badge
badge-danger">Delete</a>
                    </td>
                </tr>
```

```
                <?php $q++; ?>
                <?php endforeach; ?>
            </tbody>
        </table>

    </div>
</div>

</div>
</div>
<!-- /.container-fluid -->

<!-- Modals -->
<div class="modal fade" id="addRole" tabindex="-1" role="dialog" aria-labelledby="addRoleLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-
title" id="addRoleLabel">Add New Role</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-
hidden="true">&times;</span>
                </button>
            </div>
            <form action=<?= base_url('role'); ?>" me-
thod="post">
                <div class="modal-body">
                    <div class="form-group">
                        <input type="text" class="form-
control" id="role" name="role" placeholder="Type new r
ole">
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```

        <div class="modal-footer">
            <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
            <button type="submit" class="btn btn-primary">Add</button>
        </div>
    </form>
</div>
</div>

```

Fungsi **roleAccess(\$role_id)** ini bertujuan untuk memberikan hak akses kepada user untuk menampilkan menu yang di pilih hanya oleh admin saja, selain yang bukan memiliki id ‘1’ yaitu admin, user yang lain tidak dapat mengatur hak akses menu dari aplikasi.

Berikut tampilan dan view source code dari **roleaccess.php**

No	Menu	Access
1	User	<input checked="" type="checkbox"/>
2	Menu	<input checked="" type="checkbox"/>
3	Pemetaan	<input checked="" type="checkbox"/>
4	Pelelangan	<input checked="" type="checkbox"/>

Gambar 5.17 Antarmuka RoleAccess

Berikut adalah source code view dari **roleaccess.php**

```
<div id="content-wrapper">

    <div class="container-fluid">

        <!-- Breadcrumbs-->
        <ol class="breadcrumb">
            <li class="breadcrumb-item">
                <a href="">Admin</a>
            </li>
            <li class="breadcrumb-item active">Management Role</li>
        </ol>

        <!-- Page Content -->
        <div class="row">
            <div class="col-lg-6">

                <?= $this->session->flashdata('message'); ?>
                    <h3>Role : <?= $role['role']; ?></h3>
                    <table class="table table-hover">
                        <thead>
                            <tr>
                                <th scope="col">No</th>
                                <th scope="col">Menu</th>
                                <th scope="col">Access</th>
                            </tr>
                        </thead>
                        <tbody>
                            <?php $q = 1; ?>
                            <?php foreach ($menu as $m) : ?>
                                <tr>
                                    <td><?= $q; ?></td>
                                    <td><?= $m['menu']; ?></td>
                                    <td><?= $m['access']; ?></td>
                                </tr>
                            <?php endforeach; ?>
                        </tbody>
                    </table>
            </div>
        </div>
    </div>
```

```

<th scope="row"><?= $q
; ?></th>
</td><?= $m[ 'menu' ]; ?>
<td>
<div class="form-
check">
<input class="form-
check-
input" type="checkbox" <?= check_access($role[ 'id' ], $m[ 'id' ]); ?> data-role=<?= $role[ 'id' ]; ?>" data-
menu=<?= $m[ 'id' ]; ?>">
</div>
</td>
</tr>
<?php $q++; ?>
<?php endforeach; ?>
</tbody>
</table>

</div>
</div>

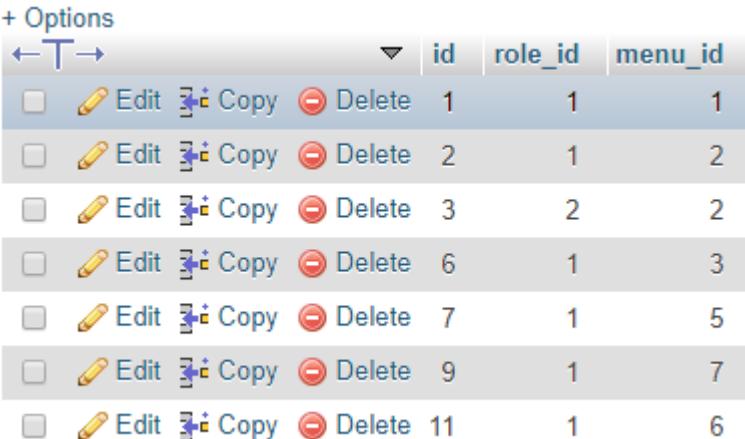
</div>
</div>
<!-- /.container-fluid -->

<!-- Modals -->

```

Fungsi **changeAccess()** ini bertujuan untuk memberikan hak akses menu yang di kelola oleh admin yang ketika admin memberi akses menu kepada salah satu user maka database

user_accessmenu akan langsung terupdate dan berubah, berikut isi dari database **user_accessmenu**



	+ Options	← →	id	role_id	menu_id	
<input type="checkbox"/>		Edit			1	1
<input type="checkbox"/>		Edit			2	1
<input type="checkbox"/>		Edit			3	2
<input type="checkbox"/>		Edit			6	1
<input type="checkbox"/>		Edit			7	1
<input type="checkbox"/>		Edit			9	1
<input type="checkbox"/>		Edit			11	6

Gambar 5.16 Tabel **user_accessmenu**

role_id ‘1’ adalah admin dan ‘2’ adalah member, menu_id adalah id-id yang berisi menu-menu pada aplikasi, dapat dilihat admin diberikan akses semua menu, sedangkan member hanya diberikan satu hak akses menu.

5 . 6 Pembuatan Halaman Management Menu Admin

Buat file baru pada folder controllers dan beri nama **Menu.php** pada file ini kita akan membuat management menu, yang nantinya semua menu dapat di olah seperti di tambahkan,di

edit, dan di hapus langsung di melalui interface tanpa mengubah kodingan.

Pertama buat terlebih dahulu fungsi index, berikut source nya

Fungsi **index()**

```
public function __construct()
{
    parent::__construct();
    check_login();
    $this->load->model('Menu_model');
}
public function index()
{
    $data['title'] = 'Management Menu';
    $data['user'] = $this->db-
>get_where('tbl_user', ['user_email' =>
    $this->session->userdata('user_email')])->row_array();

    $data['menu'] = $this->db->get('user_menu')-
>result_array();

    $this->form_validation-
>set_rules('menu', 'Menu', 'required');
    if ($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('menu/index', $data);
        $this->load->view('temanbody/footer');
    } else {
```

```
    $this->db-
>insert('user_menu', ['menu' => $this->input-
>post('menu')]);
    $this->session-
>set_flashdata('message', '<div class="alert alert-
success" role="alert">Menu Added!</div>');
    redirect('menu');
}
}
```

Fungsi **construct** pada source diatas bertujuan untuk memanggil library/model pada kasus ini yaitu library **check_login()** dan model **Menu_model**, fungsi ini digunakan untuk setiap fungsi yang ada pada file tersebut sehingga jika ada fungsi baru, tidak perlu mendeklarasikan kembali library dan model tersebut pada masing-masing fungsi baru.

Fungsi **index()** bertujuan untuk menampilkan dan menambahakan menu baru dengan rules tidak boleh dikosongkan (required) ketika menginput menu baru berikut sourcenyanya,

\$data['menu'] = \$this->db->get('user_menu')->result_array();
source ini bertujuan untuk menampilkan semua data yang ada pada tabel **user_menu** dengan variabel data dan parameter menu yang nantinya akan di panggil di view.

\$this->db->insert('user_menu', ['menu' => \$this->input-
>post('menu')]); bertujuan untuk menginput data kedalam database tapatnya pada tabel **user_menu**

setelah itu kita akan membuat fungsi edit menu berikut source nya,

fungsi **edit_menu()**

```
function edit_menu()
{
    $id = $this->input->post('id');
    $this->Menu_model->edit_menu($id);
    $this->session-
>set_flashdata('message', '<div class="alert alert-
success" role="alert">Menu Updated!</div>');
    redirect('menu');
}
```

Source diatas berfungsi untuk mengubah data yang prosesnya dikirim ke **Menu_model** fungsi **edit_menu(\$id)** berikut source dari fungsi **edit_menu(\$id)**,

```
public function edit_menu($id)
{
    $data = array(
        'menu' => $this->input->post('menu')
    );
    $this->db->where('id', $id);
    return $this->db->update('user_menu', $data);
}
```

Source diatas berfungsi mengubah data yang ada pada database tepatnya pada tabel **user_menu**

Setelah membuat fungsi edit, kita akan membuat fungsi delete atau hapus, berikut source nya,

```
function delete_menu()
{
    $id['id'] = $this->uri->segment(3);
    $this->Menu_model-
>delete_menu('user_menu', $id);
    redirect('menu');
}
```

Source diatas berfungsi untuk menghapus data yang prosesnya dilakukan di **Model_menu** fungsi **delete_menu(\$table, \$data)** berikut source fungsi **delete_menu(\$table,\$data)**,

```
public function delete_menu($table, $data)
{
    $this->db->delete($table, $data);
}
```

Source diatas berfungsi menghapus data yang ada pada database tepatnya pada tabel **user_menu**

Setelah membuat fungsi menu pada controller kita akan membuat tampilan antarmuka untuk management menu buat file baru di dalam folder view, sebelumnya buat terlebih dahulu folder baru lagi beri nama **menu** lalu masukan file php index disini dan beri nama **index.php** ini berikut sourcenyanya,

```
<div id="content-wrapper">

<div class="container-fluid">

    <!-- Breadcrumbs-->
    <ol class="breadcrumb">
        <li class="breadcrumb-item">
```

```
        <a href="">Admin</a>
    </li>
    <li class="breadcrumb-item active">Management Menu</li>
</ol>

<!-- Page Content -->
<div class="row">
    <div class="col-lg-6">
        <?= form_error('menu', '<div class="alert alert-danger" role="alert">', '</div>'); ?>

        <?= $this->session->flashdata('message'); ?>

        <a href="" class="btn btn-success mb-3" data-toggle="modal" data-target="#addMenu">Add Menu</a>
        <table class="table table-hover">
            <thead>
                <tr>
                    <th scope="col">No</th>
                    <th scope="col">Menu</th>
                    <th scope="col">Action</th>
                </tr>
            </thead>
            <tbody>

                <?php $q = 1; ?>
                <?php foreach ($menu as $m) : ?>
                    <tr>
                        <th scope="row"><?= $q ; ?></th>
                        <td><?= $m['menu']; ?>
                    </td>
```

```
<td>
    <a class="badge badge-warning" data-toggle="modal" href="" data-
target="#modalEdit<?= $m['id']; ?>">Edit</a>
    <a class="badge badge-
danger" href="<?= base_url('menu/delete_menu/' . $m['i
d']); ?>" onclick="return confirm('Are you sure?')">De
lete</a>
</td>
</tr>
<?php $q++; ?>

<!-- Modal EDIT --&gt;
&lt;div id="modalEdit&lt;?= $m['
id']; ?&gt;" tabindex="-1" class="modal fade"&gt;
    &lt;div class="modal-
dialog" role="document"&gt;
        &lt;div class="modal-
content"&gt;
            &lt;div class="mo
dal-header"&gt;
                &lt;h5 class=
"modal-title" id="modalEdit"&gt;Edit Menu&lt;/h5&gt;
                &lt;button ty
pe="button" class="close" data-dismiss="modal" aria-
label="Close"&gt;
                    &lt;span
aria-hidden="true"&gt;&amp;times; &lt;/span&gt;
                &lt;/button&gt;
            &lt;/div&gt;
            &lt;form action="
&lt;?= base_url('menu/edit_menu'); ?&gt;" method="post" enct
ype="multipart/form-data"&gt;
                &lt;div class
="modal-body"&gt;</pre>
```

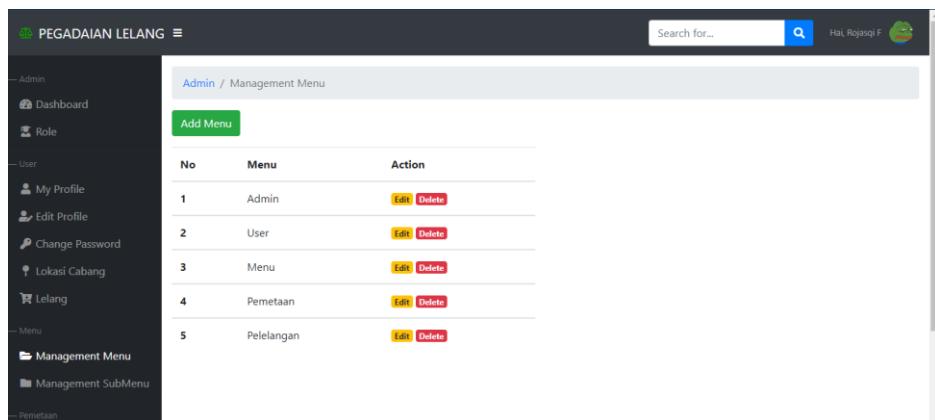
```
        <div c
lass="form-group">
    <i

```

```
<div class="modal fade" id="addMenu" tabindex="-1" role="dialog" aria-labelledby="addMenuLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-
title" id="addMenuLabel">Add New Menu</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-
hidden="true">&times;</span>
                </button>
            </div>
            <form action="= base_url('menu'); ?" me-
thod="post">
                <div class="modal-body">
                    <div class="form-group">
                        <input type="text" class="form-
control" id="menu" name="menu" placeholder="Type new m
enu">
                    </div>
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn b
tn-secondary" data-dismiss="modal">Close</button>
                    <button type="submit" class="btn b
tn-primary">Add</button>
                </div>
            </form>
        </div>
    </div>
</div>
```

Dalam source untuk tampilan management menu diatas admin dapat menambahkan, mengedit, dan menghapus menu

dengan cepat, karena disini kita menggunakan modal, menampilkan pop up window baru untuk mengolah semuanya tanpa pindah halaman. Berikut tampilan antarmuka dari management menu.



The screenshot shows a web application interface titled "PEGADAIAN LELANG". The left sidebar contains a navigation menu with categories like Admin, User, and Menu. Under the "Menu" category, there are "Management Menu" and "Management SubMenu". The main content area is titled "Admin / Management Menu" and displays a table titled "Add Menu". The table has columns "No", "Menu", and "Action". It lists five menu items: 1. Admin, 2. User, 3. Menu, 4. Pemetaan, and 5. Pelelangan. Each item has "Edit" and "Delete" buttons next to it. A search bar and user profile information are visible at the top right.

No	Menu	Action
1	Admin	Edit Delete
2	User	Edit Delete
3	Menu	Edit Delete
4	Pemetaan	Edit Delete
5	Pelelangan	Edit Delete

Gambar 5.17 Antarmuka Management Menu

5.6.1 Pembuatan Halaman Submenu

Kita akan membuat submenu pada management menu sebelumnya, masih dalam controller sama yaitu controller menu, tambahkan fungsi **submenu()**, berikut source nya

```
public function submenu()
{
    $data['title'] = 'Management SubMenu';
    $data['user'] = $this->db-
>get_where('tbl_user', ['user_email' =>
    $this->session->userdata('user_email')])->row_array();
```

```
$this->load->model('Menu_model', 'menu');
$data['subMenu'] = $this->menu->getSubMenu();
$data['menu'] = $this->db->get('user_menu')-
>result_array();

        $this->form_validation-
>set_rules('title', 'Title', 'required');
        $this->form_validation-
>set_rules('menu_id', 'Menu', 'required');
        $this->form_validation-
>set_rules('url', 'URL', 'required');
        $this->form_validation-
>set_rules('icon', 'Icon', 'required');

        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('menu_submenu', $data);
            $this->load->view('temanbody/footer');
        } else {
            $data = [
                'title' => $this->input-
>post('title'),
                'menu_id' => $this->input-
>post('menu_id'),
                'url' => $this->input->post('url'),
                'icon' => $this->input->post('icon'),
                'is_active' => $this->input-
>post('is_active')
            ];
            $this->db->insert('user_submenu', $data);
```

```
        $this->session-
>set_flashdata('message', '<div class="alert alert-
success" role="alert">SubMenu Added!</div>');
        redirect('menu/submenu');
    }
}
```

Source diatas berfungsi mememberikan rules required (tidak boleh kosong) pada form pengisian submenu baru, dan pada source,

```
$data = [
'title' => $this->input->post('title'),
'menu_id' => $this->input->post('menu_id'),
'url' => $this->input->post('url'),
'icon' => $this->input->post('icon'),
'is_active' => $this->input->post('is_active') ];
$this->db->insert('user_submenu', $data);
```

Source diatas melakukan proses penginputan data dengan method post kedalam masing-masing field pada database tepatnya tabel **user_submenu** berikut contoh isi dari tabel user_submenu,

	Edit	Copy	Delete	id	menu_id	title	url	icon	is_active
<input type="checkbox"/>	Edit	Copy	Delete	1	1	Dashboard	admin	fas fa-fw fa-tachometer-alt	1
<input type="checkbox"/>	Edit	Copy	Delete	2	2	My Profile	user	fas fa-fw fa-user	1
<input type="checkbox"/>	Edit	Copy	Delete	3	2	Edit Profile	user/edit	fas fa-fw fa-user-edit	1
<input type="checkbox"/>	Edit	Copy	Delete	4	3	Management Menu	menu	fas fa-fw fa-folder-open	1
<input type="checkbox"/>	Edit	Copy	Delete	5	3	Management SubMenu	menu_submenu	fas fa-fw fa-folder	1
<input type="checkbox"/>	Edit	Copy	Delete	6	1	Role	admin/role	fas fa-fw fa-user-graduate	1
<input type="checkbox"/>	Edit	Copy	Delete	7	2	Change Password	user/changepassword	fas fa-fw fa-key	1
<input type="checkbox"/>	Edit	Copy	Delete	9	6	Management Pemetaan	pemetaan	fas fa-fw fa-map-marked-alt	1
<input type="checkbox"/>	Edit	Copy	Delete	10	6	Data Pemetaan	pemetaan/datapemetaan	fas fa-fw fa-map-marker-alt	1
<input type="checkbox"/>	Edit	Copy	Delete	11	7	Data Pelelangan	pelelangan	fas fa-fw fa-balance-scale-left	1
<input type="checkbox"/>	Edit	Copy	Delete	12	2	Lokasi Cabang	user/tampilanpemetaan	fas fa-fw fa-map-pin	1
<input type="checkbox"/>	Edit	Copy	Delete	13	2	Lelang	user/lelang	fas fa-fw fa-cart-arrow-down	1
<input type="checkbox"/>	Edit	Copy	Delete	19	7	Data Penawar	pelelangan/datapenawar	fas fa-fw fa-handshake	1

Gambar 5.18 Tabel User_submenu

Pembuatan fungsi **edit_submenu**, berikut sourcenyanya,

```
function edit_submenu()
{
    $id = $this->input->post('id');
    $data = [
        'title' => $this->input->post('title'),
        'menu_id' => $this->input-
>post('menu_id'),
        'url' => $this->input->post('url'),
        'icon' => $this->input->post('icon'),
        'is_active' => $this->input-
>post('is_active')
    ];

    $this->Menu_model->edit_submenu($id, $data);
}
```

```
        $this->session-
>set_flashdata('message', '<div class="alert alert-
success" role="alert">SubMenu Updated!</div>');
    redirect('menu/submenu');
}
```

Source diatas bertujuan untuk mengubah atau melakukan proses update data proses dilakukan pada model Menu_model didalam fungsi **edit_submenu(\$id,\$data)**, berikut source fungsi yang berada pada Menu_model,

```
public function edit_submenu($id, $data)
{
    $this->db->where('id', $id);
    return $this->db-
>update('user_submenu', $data);
}
```

Source diatas bertujuan melakukan proses pengubahan data atau update data submenu, yang terhubung langsung dengan database tepatnya tabel **user_submenu**

Pembuatan fungsi **delete_submenu**, masih pada controller **menu.php** berikut source nya,

```
function delete_submenu()
{
    $id['id'] = $this->uri->segment(3);
    $this->Menu_model-
>delete_submenu('user_submenu', $id);
```

```
    redirect('menu/submenu');
}
```

Source diatas bertujuan untuk melakukan proses penghapusan data yang dikirim ke model Menu_model dengan fungsi baru yaitu **delete_submenu**, berikut source nya,

```
public function delete_submenu($table, $data)
{
    $this->db->delete($table, $data);
}
```

Source diatas merupakan proses penghapusan data pada database tepatnya pada tabel **user_submenu** yang sebelumnya sudah di inisiasi pada fungsi di controller menu.

Untuk tampilan dari management submenu, buat terlebih dahulu file di view folder menu dengan memberi nama **submenu.php** berikut source nya,

```
<div id="content-wrapper">

<div class="container-fluid">

    <!-- Breadcrumbs-->
    <ol class="breadcrumb">
        <li class="breadcrumb-item">
            <a href="">Admin</a>
```

```
        </li>
    <li class="breadcrumb-item active">Management Menu</li>
</ol>


<div class="row">
    <div class="col-lg">
        <?php if (validation_errors()) : ?>
            <div class="alert alert-danger" role="alert">
                <?= validation_errors(); ?>
            </div>
        <?php endif; ?>

        <?= $this->session->flashdata('message'); ?>

        <a href="" class="btn btn-success mb-3" data-toggle="modal" data-target="#addSubMenu">Add SubMenu</a>
        <table class="display nowrap" id="example" style="width:100%">
            <thead>
                <tr>
                    <th scope="col">No</th>
                    <th scope="col">Title</th>
                    <th scope="col">Menu</th>
                    <th scope="col">Url</th>
                    <th scope="col">Icon</th>
                    <th scope="col">Active</th>
                </tr>
            <thead>
                <tr>
                    <th scope="col">Action</th>
                </tr>
            <tbody>
```

```

        <?php $q = 1; ?>
        <?php foreach ($subMenu as $sm
) : ?>
            <tr>
                <th scope="row"><?= $q
; ?></th>
                <td><?= $sm['title'];
?></td>
                <td><?= $sm['menu']; ?>
</td>
                <td><?= $sm['url']; ?>
</td>
                <td><?= $sm['icon']; ?>
</td>
                <td><?= $sm['is_active
']; ?></td>
                <td>
                    <a data-
toggle="modal" data-
target="#modalEdit<?= $sm['id']; ?>" class="badge badge-
warning">Edit</a>
                    <a href=<?= base_
url('menu/delete_submenu/' . $sm['id']); ?>" onclick="
return confirm('Are you sure?')" class="badge badge-
danger">Delete</a>
                </td>
            </tr>
            <?php $q++; ?>

        <!-- Modal EDIT -->
        <div id="modalEdit<?= $sm[
'id']; ?>" tabindex="-1" class="modal fade">
            <div class="modal-
dialog" role="document">
                <div class="modal-
content">

```

```
        <div class="mo
dal-header">
            <h5 class=
"modal-title" id="modalEdit">Edit SubMenu</h5>
            <button ty
pe="button" class="close" data-dismiss="modal" aria-
label="Close">
                <span
aria-hidden="true">&times;</span>
            </button>
        </div>
        <form action="
<?= base_url('menu/edit_submenu'); ?>" method="post" e
nctype="multipart/form-data">
            <div class
="modal-body">
                <div c
lass="form-group">
                    <i
nput type="hidden" class="form-
control" id="id" name="id" value="<?= $sm['id']; ?>">
                    <l
abel>Title</label>
                    <i
nput type="text" class="form-
control" id="title" name="title" value="<?= $sm['title
']; ?>">
                </div>
                <div c
lass="form-group">
                    <l
abel>Menu</label>
                    <s
elect name="menu_id" id="menu_id" class="form-
control">
                        <option value=""><?= $sm['menu']; ?></option>
```

```
<?php foreach ($menu as $m) : ?>

    <option value="<?= $m['id']; ?>><?= $m['menu'];
?></option>

<?php endforeach; ?>
        </
select>
                    </div>
                <div c
lass="form-group">
                    <l
abel>Url</label>
                    <i
nput type="text" class="form-
control" id="url" name="url" value="<?= $sm['url']; ?>
">
                    </div>
                <div c
lass="form-group">
                    <l
abel>Icon</label>
                    <i
nput type="text" class="form-
control" id="icon" name="icon" value="<?= $sm['icon'];
?>>
                    </div>
                <div c
lass="form-group">
                    <d
iv class="custom-control custom-checkbox">

            <input type="checkbox" class="custom-control-
input" value="1" id="is_active" name="is_active" checked>
```

```
        <label class="custom-control-
label" for="is_active">Active</label>
        </
      div>
      </div>
      <div c
lass="modal-footer">
      <b
    utton type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
      <b
    utton type="submit" class="btn btn-
primary">Update</button>
      </div>
    </form>
  </div>
</div>
</div>

          <?php endforeach; ?>
        </tbody>
      </table>

      </div>
    </div>

  </div>
</div>
<!-- /.container-fluid -->

<!-- Modals -->
<div class="modal fade" id="addSubMenu" tabindex="-
1" role="dialog" aria-
labelledby="addSubMenuLabel" aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
```

```
<div class="modal-header">
    <h5 class="modal-
title" id="addSubMenuLabel">Add New SubMenu</h5>
    <button type="button" class="close" da-
ta-dismiss="modal" aria-label="Close">
        <span aria-
hidden="true">&times;</span>
    </button>
</div>
<form action="= base_url('menu/submenu') 
; ?&gt;" method="post"&gt;
    &lt;div class="modal-body"&gt;
        &lt;div class="form-group"&gt;
            &lt;input type="text" class="form-
control" id="title" name="title" placeholder="Type new
title"&gt;
        &lt;/div&gt;
        &lt;div class="form-group"&gt;
            &lt;select name="menu_id" id="men
u_id" class="form-control"&gt;
                &lt;option value=""&gt;Select me
nu&lt;/option&gt;
                &lt;?php foreach ($menu as $m
) : ?&gt;
                &lt;option value="<?= $m['id']; ?&gt;&gt;&lt;?= $m['menu']; ?&gt;&lt;/option&gt;
            &lt;?php endforeach; ?&gt;
        &lt;/select&gt;
    &lt;/div&gt;
    &lt;div class="form-group"&gt;
        &lt;input type="text" class="form-
control" id="url" name="url" placeholder="Type Submenu
url"&gt;
    &lt;/div&gt;
&lt;div class="form-group"&gt;</pre
```

```
        <input type="text" class="form-
-
control" id="icon" name="icon" placeholder="Type Subme
nu icon">
    </div>
    <div class="form-group">
        <div class="custom-
control custom-checkbox">
            <input type="checkbox" cla
ss="custom-control-
input" value="1" id="is_active" name="is_active" check
ed>
                <label class="custom-
control-label" for="is_active">Active</label>
            </div>
        </div>
        <div class="modal-footer">
            <button type="button" class="btn b
tn-secondary" data-dismiss="modal">Close</button>
            <button type="submit" class="btn b
tn-primary">Add</button>
        </div>
    </form>
</div>
</div>
```

Source tampilan diatas bertujuan untuk proses pengolahan data pada menu management submenu, seperti menambahkan, mengedit, dan menghapus data semua proses pengolahan menggunakan modal menampilkan pop up window baru.

Berikut tampilan dari antarmuka management submenu,

The screenshot shows a web application interface for managing submenus. On the left, there's a sidebar with a tree-like navigation menu. The 'Menu' section is expanded, showing 'Management Menu' and 'Management SubMenu'. The 'Management SubMenu' item is selected and highlighted in blue. The main content area has a title 'Admin / Management Menu' and a sub-section title 'Add SubMenu'. Below these are buttons for 'Copy', 'CSV', 'Excel', 'PDF', and 'Print'. A search bar is also present. The central part of the screen is a table listing five submenu entries:

No	Title	Menu	Url	Icon	Active	Action
1	Dashboard	Admin	admin	fas fa-fw fa-tachometer-alt	1	<button>Edit</button> <button>Delete</button>
2	My Profile	User	user	fas fa-fw fa-user	1	<button>Edit</button> <button>Delete</button>
3	Edit Profile	User	user/edit	fas fa-fw fa-user-edit	1	<button>Edit</button> <button>Delete</button>
4	Management Menu	Menu	menu	fas fa-fw fa-folder-open	1	<button>Edit</button> <button>Delete</button>
5	Management SubMenu	Menu	menu_submenu	fas fa-fw fa-folder	1	<button>Edit</button> <button>Delete</button>

At the bottom of the table, it says 'Showing 1 to 10 of 13 entries'. To the right of the table, there are buttons for 'Previous', '1', '2', and 'Next'.

Gambar 5.19 Antarmuka Management Submenu

Pada menu management submenu, admin dapat melakukan pengolahan data dan terdapat ekstensi tambahan seperti export menjadi csv,excel,pdf atau melakukan cetak data/print.

5.7 Pembuatan Halaman User

Pertama buat kembali file baru di controller dengan nama **User.php** buat beberapa fungsi,

Fungsi **index()**

```
<?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');
    }
}
```

Source ditas terdapat fungsi **construct** artinya sama seperti yang sebelumnya yaitu meginisiasi semua fungsi yang ada pada controller User.php sehingga semua fungsi di dalamnya tidak perlu melakukan deklarasi fungsi kembali, Fungsi **Index()** bertujuan untuk menampilkan halaman utama dari user.

Didalam halaman utama user kita akan membuat management user yang di dalamnya user dapat mengubah gambar foto profilenya, mengubah namanya dan mengubah passwordnya,

Pertama buat terlebih dahulu fungsi baru dalam file User.php buat fungsi **edit()** berikut source nya,

```
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');

        $upload_image = $_FILES['image']['name'];
        if ($upload_image) {
            $config['allowed_types'] = 'gif|jpg|png';
            $config['max_size'] = '2048';
            $config['upload_path'] = './assets/images/';
        }

        $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!<
/div>');
    redirect('user');
}
}
```

Source diatas memvalidasi setiap inputan yang dilakukan oleh user seperti penggantian foto profil, ubah password atau nama,

```
$upload_image = $_FILES['image']['name'];

if ($upload_image) {

$config['allowed_types'] = 'gif|jpg|png';
$config['max_size'] = '2048';
$config['upload_path'] = './assets/images/';

$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();  
  
}  
  
}
```

Pada source diatas bertujuan untuk mengecek gambar yang akan di upload apakah sudah sesuai dengan ketentuan yang diberikan seperti format yang harus di upload berekstensi jpg,png, atau gif dan ukuran dari file maksimal harus 2mb dan penyimpanan default ketika user upload akan di simpan secara otomatis pada folder assets/images, jika tidak sesuai gambar tidak bisa terubah.

Buat terlebih dahulu tampilan atau view untuk edit management user, buat di view sebelumnya buat dulu folder baru dan beri nama user di dalam folder tersebut buat kembali file php dan beri nama edit.php, beikut source tampilan dari **edit.php**

```
<div id="content-wrapper">
```

```
<div class="container-fluid">

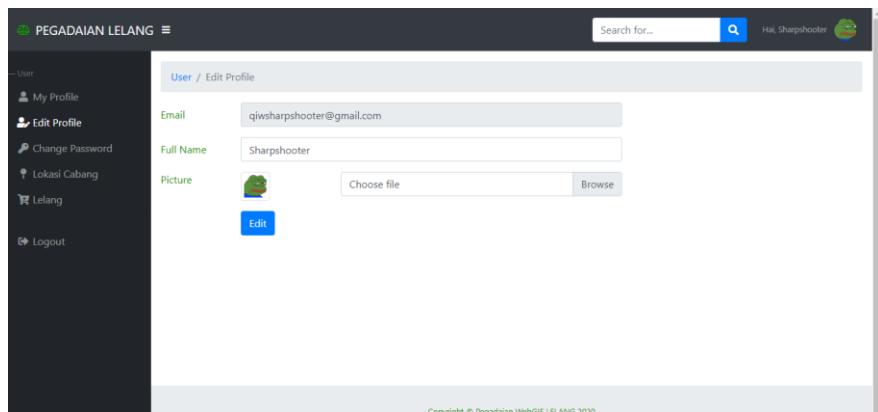
    <!-- Breadcrumbs-->
    <ol class="breadcrumb">
        <li class="breadcrumb-item">
            <a href="">User</a>
        </li>
        <li class="breadcrumb-item active">Edit Profile</li>
    </ol>

    <!-- Page Content -->
    <div class="row">
        <div class="col-lg-8">

            <?= form_open_multipart('user/edit');
?>
            <div class="form-group row">
                <label for="email" class="col-sm-2 col-form-label">Email</label>
                <div class="col-sm-10">
                    <input type="text" class="form-control" id="email" name="email" value="<?= $user['user_email']; ?>" readonly>
                </div>
            </div>
            <div class="form-group row">
                <label for="name" class="col-sm-2 col-form-label">Full Name</label>
                <div class="col-sm-10">
                    <input type="text" class="form-control" id="name" name="name" value="<?= $user['user_name']; ?>">
                    <?php echo form_error('name', '<small class="text-danger pl-3">', '</small>'); ?>
                </div>
            </div>
        </div>
    </div>
</div>
```

```
                </div>
            </div>
            <div class="form-group row">
                <div class="col-sm-2">Picture</div>
                <div class="col-sm-10">
                    <div class="row">
                        <div class="col-sm-3">
                            
<!-- /.container-fluid -->
```

Source diatas menampilkan antar muka jika user dalam kasus ini yaitu member akan melakukan perubahan / update data dirinya, berikut tampilan dari edit.php



Gambar 5.20 Antarmuka Edit user

Setelah itu kita membuat tampilan antarmuka dari user dengan membuat file baru pada view tapi sebelumnya buat folder baru beri nama user dan didalam folder user buat file php baru beri nama **index.php**, berikut source nya

```
<div id="content-wrapper">

    <div class="container-fluid">

        <!-- Breadcrumbs-->
        <ol class="breadcrumb">
            <li class="breadcrumb-item">
                <a href="">User</a>
            </li>

            <li class="breadcrumb-item active">My Profile</li>
        </ol>

        <div class="row">
            <div class="col-lg-5">
                <?= $this->session->flashdata('message'); ?>
            </div>
        </div>

        <!-- Page Content -->

        <div class="col-lg-4">
            <div class="card mb-3">
                <div class="card-header">
                    My Profile</div>
                <div class="card-body">
```

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

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

```
$this->form_validation-
>set_rules('passwordbaru2', 'Confirm New Password', 'r
equired|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, $da
ta['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
current password!</div>');
                    redirect('user/changepassword');
                } else {
                    // password benar
                    $password_hash = password_hash($pa
ssword_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');
    }
}
}
```

Source diatas berfungsi mengubah password dari pasword lama menjadi password baru dengan validasi rules tidak boleh di kosongkan (required) pada saat penginputan data.

Setelah itu buat tampilan dari ubah password ini, buat file baru di view dalam folder user berinama file tersebut dengan nama, **changepassword.php** berikut source nya,

```
<div id="content-wrapper">

<div class="container-fluid">

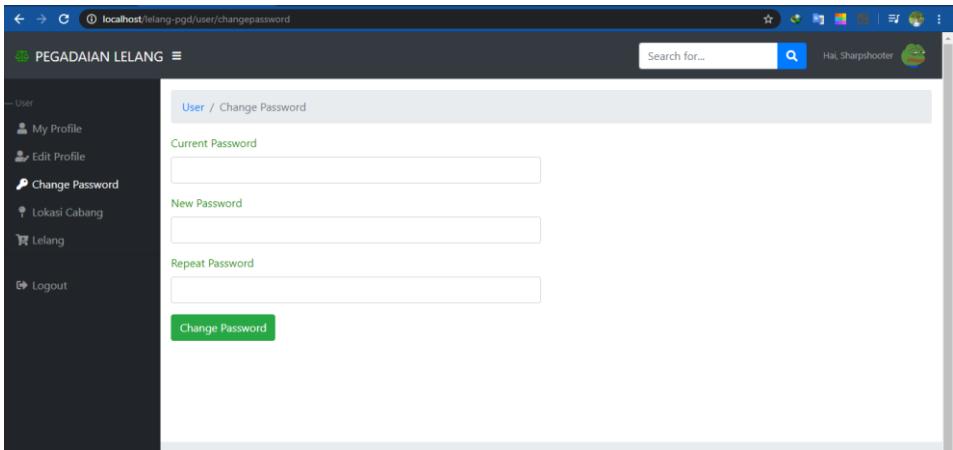
    <!-- Breadcrumbs-->
    <ol class="breadcrumb">
        <li class="breadcrumb-item">
            <a href="">User</a>
```

```
        </li>
        <li class="breadcrumb-item active">Change Password</li>
    </ol>

    <!-- Page Content -->
    <div class="row">
        <div class="col-lg-6">
            <?= $this->session->flashdata('message'); ?>
            <form action=<?= base_url('user/changepassword') ?>" method="post">
                <div class="form-group">
                    <label for="passwordsaatini">Current Password</label>
                    <input type="password" class="form-control" id="passwordsaatini" name="passwordsaatini">
                    <?php echo form_error('passwordsaatini', '<small class="text-danger pl-3">', '</small>'); ?>
                </div>
                <div class="form-group">
                    <label for="passwordbaru1">New Password</label>
                    <input type="password" class="form-control" id="passwordbaru1" name="passwordbaru1">
                    <?php echo form_error('passwordbaru1', '<small class="text-danger pl-3">', '</small>'); ?>
                </div>
                <div class="form-group">
                    <label for="passwordbaru2">Repeat Password</label>
                    <input type="password" class="form-control" id="passwordbaru2" name="passwordbaru2">
                </div>
            </form>
        </div>
    </div>
```

```
        <?php echo form_error('passwordbaru2', '<small class="text-danger pl-3">', '</small>'); ?>
            </div>
            <div class="form-group">
                <button type="submit" class="btn btn-success">Change Password</button>
            </div>
        </div>
    </div>
<!-- /.container-fluid -->
```

Source diatas merupakan form dengan inputan-inputan jika user akan mengubah data dirinya, berikut tampilan dari ubah password user,



Gambar 5.22 Antarmuka ChangePassword

Pada tampilan diatas user/member dapat mengubah passwordnya dengan mengisikan password lama dan password baru sebanyak duakali, klik button Change Password

5 . 8 Pembuatan Halaman Management Pemetaan

Pertama buat terlebih dahulu file baru pada controller beri nama **pemetaan.php** setelah itu buat fungsi baru dengan nama **index.php** berikut source nya,

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

```
class Pemetaan extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        check_login();
        $this->load->library(array('googlemaps'));
        $this->load->model('Gis_Model');
    }

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

        $this->load->library('googlemaps');
        $config['center'] = '-6.9032739, 107.5731165';
        $config['zoom'] = '15';
        $this->googlemaps->initialize($config);
        $marker['position'] = '-6.9032739, 107.5731165';
        $marker['draggable'] = true;
        $marker['ondragend'] = 'setMapToForm(event.latLng.lat(),event.latLng.lng());';
        $this->googlemaps->add_marker($marker);
        $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('menu/v_pemetaan', $data);
        $this->load->view('temanbody/footer');
    }
```

Source diatas terdapat fungsi contruct kembali yang berisi library google maps untuk menampilkan maps di web yang akan kita buat, dan model Gis_Model yang berisi proses pengolahan data pemetaan.

Fungsi Index()

Di dalam fungsi tersebut kita akan me load pusat titik koordinat yang kita buat saya memasukan latitude dan longitude pada titik kota bandung. Yang akan diload pada view , sebelumnya buat terlebih dahulu file view buat file baru di dalam folder menu beri nama **V_pemetaan.php** berikut sourcenyanya,

```
<div id="content-wrapper">
    <div class="container-fluid">
        <!-- Breadcrumbs-->
        <ol class="breadcrumb">
            <li class="breadcrumb-item">
                <a href="">Pemetaan</a>
            </li>
            <li class="breadcrumb-item active">Management Pemetaan</li>
        </ol>
        <div class="form-group">
            <?php if ($this->session->flashdata('message')) : ?>
```

```
        <button type="button" class="close" da-
ta-dismiss="alert" aria-hidden="true">&times;</button>
            <?php echo $this->session-
>flashdata('message'); ?>
        </div>
    <?php endif; ?>
</div>
<div class="row">
    <div class="col-lg-12">
        <div class="card bg-light mb-
6" style="max-width: 80rem;">
            <div class="card-
header">Form Input Pemetaan Cabang Bandung 2
            </div>
            <div class="card-body">
                <form action="<?= base_url() . 'pe-
metaan/inputcabang'; ?>" method="post" enctype="multip-
part/form-data">
                    <form class="form-horizontal">
                        <div class="form-group">
                            <label>Cabang Pembuka<
/label>
                            <select name="nama_cab
ang" class="form-control" required>
                                <?php foreach ($ca
bang as $g) : ?>
                                    <?= $g->id; ?>
                                    <option value=
"<?= $g->cabang; ?>"><?= $g->cabang; ?></option>
                                <?php endforeach ?>
                            </select>
                        </div>
                        <div class="form-group">
                            <label for="alamat">Al
amat :</label>
```

```
<div class="form-group">
    <textarea name="alamat" class="form-control" rows="3" placeholder="ex. Jl.Sukup Baru 1A Ujung Berung"></textarea>
    <p class="help-block"><?php echo form_error('alamat', '<small class="text-red">', '</small>'); ?></p>
</div>
<div class="form-group">
    <label for="kota">Kota</label>
    <input type="text" class="form-control" id="kota" required placeholder="ex. Bandung" name="kota">
    <?php echo form_error('kota', '<small class="text-danger pl-3">', '</small>'); ?>
</div>
<div class="form-group">
    <label for="no_telp">No Telp</label>
    <input type="text" class="form-control" id="no_telp" required placeholder="ex. 022222222" name="no_telp">
    <?php echo form_error('no_telp', '<small class="text-danger pl-3">', '</small>'); ?>
</div>
<div class="form-group">
    <label for="image">Image</label>
```

```
<div class="custom-
file">
    <input type="file"
        class="custom-file-input" id="image" name="image">
    <label class="cust
om-file-label" for="image">Choose file</label>
</div>
</div>
<div class="form-group">
    <label for="deskripsi">
        Deskripsi :</label>
        <div class="form-
group">
            <textarea name="de
skripsi" class="form-control" rows="6"></textarea>
            <p class="help-
block"><?php echo form_error('deskripsi', '<small clas
s="text-red">', '</small>'); ?></p>
        </div>
        </div>
        <div class="form-group">
            <label for="koordinat">
                Koordinat :</label>
                <label class="sr-
only" for="latitude">Latitude</label>
                <div class="input-
group mb-2 mr-sm-2">
                    <div class="input-
group-prepend">
                        <div class="in
put-group-text"><i class="fa fa-map-marker-
alt"></i></div>
                    </div>
                    <input type="text"
                        class="form-
control" id="latitude" name="latitude" value="<?php ec
ho set_value('latitude') ?>" placeholder="Latitude">
                </div>
            </div>
        </div>
    </div>
</div>
```

```
<p class="help-block"><?php echo form_error('latitude', '<small class="text-red">', '</small>'); ?></p>
</div>
<label class="sr-only" for="longitude">Longitude</label>
<div class="input-group mb-2 mr-sm-2">
<div class="input-group-prepend">
<div class="input-group-text"><i class="fa fa-map-marker-alt" alt=""/></i></div>
<input type="text" class="form-control" id="longitude" name="longitude" value="<?php echo set_value('longitude') ?>" placeholder="Longitude">
<p class="help-block"><?php echo form_error('longitude', '<small class="text-red">', '</small>'); ?></p>
</div>
<div class="col-md-offset-2">
<?php echo $map['html'] ?>
<?php echo $map['js'] ?>
</div>
</div>
<div class="form-group">
<button type="submit" class="btn btn-success mb-3"><i class="fa fa-save" alt="Save icon" /> Save</button>
<button class="btn btn-warning mb-3" type="button" data-dismiss="modal" alt="Close button" /> Close
</div>
```

```
3" onClick="refreshPage()"><i class="fa fa-trash-restore-alt"></i> Reset</button>
        </div>
    </div>
</div>
</div>
</form>
<br>
```

Source diatas menampilkan sebuah form pengisian untuk menambahkan titik baru, pada kasus ini adalah cabang/outlet, dengan validasi rules required tidak boleh di kosongkan pada salah satu inputan harus terisi semua, pada pengisian titik latitude dan longitude kita hanya perlu menggeserkan marker yang ada pada maps, otomatis titik koordinat akan terisi dengan sendirinya,

```
<?php echo $map['html'] ?>
```

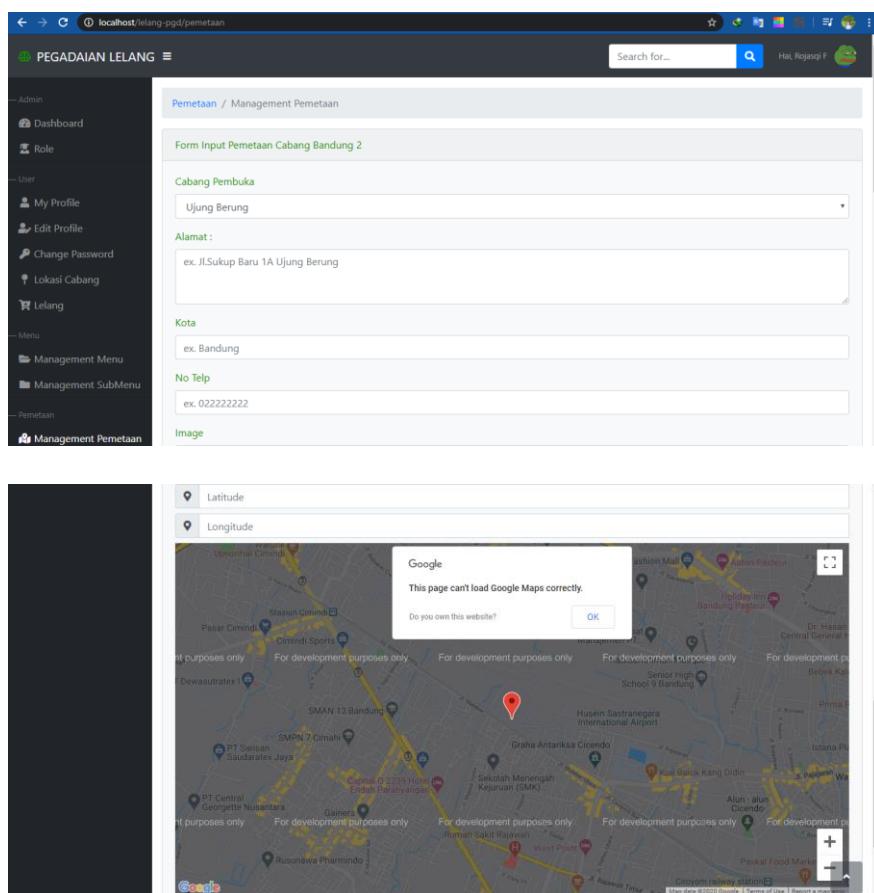
```
<?php echo $map['js'] ?>
```

Dua baris source ini yang berfungsi menampilkan peta yang telah di inisiasi pada controller sebelumnya,

Untuk menampilkan peta dari google maps tadi sebelumnya kita meload library googlemaps, buat terlebih dahulu library tersebut di folder **libraries**, buat file baru beri nama

googlemaps.php download terlebih dahulu library tersebut di link berikut, <https://gitlab.com/rojasqi.fadilla13/webgis-lelang/-/blob/master/application/libraries/googlemaps.php>

berikut tampilan dari halaman management pemetaan,



Gambar 5.23 Management Pemetaan

Setelah itu kita akan membuat tampilan dari hasil management pemetaan berupa maps dengan titik-titik koordinat dari cabang, buat fungsi baru dalam controller user, karena tampilan ini akan di tampilkan pada menu user, berikut sourcenyanya

Fungsi **tampilpemetaan()**

```
public function tampilpemetaan()
{
    $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 clas-
s="media" style="width:400px;">';
        $marker['infowindow_content'] .= '<div cla-
ss="media-left">';
        $marker['infowindow_content'] .= '<img src-
="" . base_url("assets/images/{$value-
>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>' . $value->alamat . '</p>';
            $marker['infowindow_content'] .= '<p>' . $value->deskripsi . '</p>';
            $marker['infowindow_content'] .= '</div>';
            $marker['infowindow_content'] .= '</div>';
            $marker['icon'] = base_url("assets/icon/pgd.png");
        $this->googlemaps->add_marker($marker);
    endforeach;

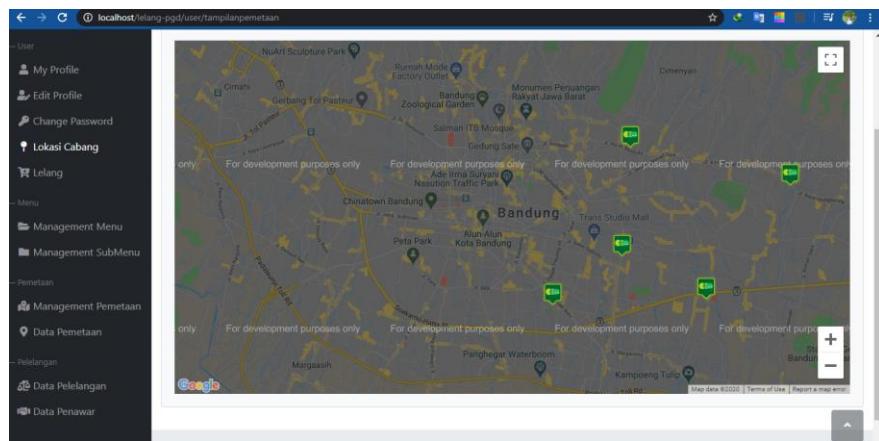
    $this->googlemaps->initialize($config);
    $data['map'] = $this->googlemaps-
>create_map();

    $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');
}

```

Source diatas menampilkan posisi-posisi dari koordinat yang telah di inputkan oleh admin sebelumnya, terdapat \$marker yang berfungsi memberi tanda shapes bewarna merah/icon, yang diambil dari \$marker['icon'] = base_url("assets/icon/pgd.png");

Ketika admin menginput data baru berupa titik koordinat dari cabang/outlet, maka secara tidak langsung akan menambah titik baru pada tampilan maps, seperti tampilan berikut,



Gambar 5.24 Tampil Pemetaan

Setelah itu kita akan membuat fungsi baru untuk menampilkan data-data cabang/oultet yang telah admin input sebelumnya buat fungsi baru di controller pemetaan, beri nama fungsi tersebut **datapemetaan()**, berikut source nya,

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

```

$data['datapemetaan'] = $this->db-
>get('pemetaan')->result_array();

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

```

Source diatas bertujuan untuk menampilkan isi dari tabel pemetaan, yang dikirim ke view dengan membawa variabel data berisi parameter datapemetaan yang nantinya akan dipanggil di view,

Berikut isi dari database tabel pemetaan,

	<input type="checkbox"/> + Options		id_cabang	nama_cabang	alamat	kota	no_telp	image	latitude	longitude	deskripsi
Click the drop-down arrow to toggle column's visibility.											
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	5	Ujung Berung	Jl. Sukup Baru No 1A, Pasir Endah, Kec. Ujung Beru...	Bandung	0227837998	uber.PNG	-6.9105164	107.6892345
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	6	Buah Batu	Jl. Buah Batu No 201A, Turannga, Kec. Lengkong	Bandung	0227303551	bubat.PNG	-6.9422204	107.6257295
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	21	Kiaracondong	Jl. Kiaracondong No 301 Kb. Kangkung, Kec. Kiar...	Bandung	0227311774	kircon.PNG	-6.929304366733729	107.64414139471432
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	22	Metro	Jl. Venus Raya No 1, Mentahlega, Kec. Rancasari	Bandung	0227569038	metro.PNG	-6.9407841	107.6667612
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	23	Ciparay	Jl. Raya Laswi, Pakutandang, Kec. Ciparay	Bandung	0222222222	ciparay.PNG	-7.0360844	107.7141001
<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	26	Suci	jl phh Mustopa No 101, Sukapada, Kec. Cikarang K...	Bandung	022321321321	suci1.PNG	-6.900447	107.6464085
<input type="checkbox"/>	<input type="checkbox"/> Console										

Gambar 5.25 Tabel Pemetaan

Untuk menampilkan data tersebut buat terlebih dahulu view dengan nama **v_datapemetaan.php** yang dibuat pada view folder menu berikut sourcennya,

```
<div id="content-wrapper">

    <div class="container-fluid">

        <!-- Breadcrumbs-->
        <ol class="breadcrumb">
            <li class="breadcrumb-item">
                <a href="">Pemetaan</a>
            </li>
            <li class="breadcrumb-item active">Data Pemetaan</li>
        </ol>

        <?= $this->session->flashdata('message'); ?>

        <table class="display nowrap" id="example" style="width:100%">
            <thead>
                <tr>
                    <th scope="col">No</th>
                    <th scope="col">Nama Cabang</th>
                    <th scope="col">Alamat</th>
                    <th scope="col">Kota</th>
                    <th scope="col">No Telp</th>
                    <th scope="col">Image</th>
                    <th scope="col">Latitude</th>
                    <th scope="col">Longitude</th>
                    <th scope="col">Deskripsi</th>
                    <th scope="col">Action</th>
                </tr>
            </thead>
            <tbody>
```

```
<?php $q = 1; ?>
<?php foreach ($datapemetaan as $sm) :
?>
<tr>
    <th scope="row"><?= $q; ?></th
>
    <td><?= $sm['nama_cabang']; ?>
</td>
    <td><?= $sm['alamat']; ?></td>
    <td><?= $sm['kota']; ?></td>
    <td><?= $sm['no_telp']; ?></td
>
    <td><?= $sm['image']; ?></td>
    <td><?= $sm['latitude']; ?></t
d>
    <td><?= $sm['longitude']; ?></
td>
    <td><?= $sm['deskripsi']; ?></
td>
    <td>
        <a href="<?= base_url('pem
etaan/updatecabang/' . $sm['id_cabang']); ?>" class="b
adge badge-warning">Edit</a>
        <a href="<?= base_url('pem
etaan/deletecabang/' . $sm['id_cabang']); ?>" onclick=
"return validasi();" class="badge badge-
danger">Delete</a>
    </td>
</tr>
<?php $q++; ?>
<?php endforeach; ?>
</tbody>
</table>

</div>
</div>
```

Source diatas menampilkan data-data dari database tabel pemetaan, berikut tampilan dari data pemetaan,

No	Nama Cabang	Alamat	Kota	No Telp	Image	Latitude
1	Ujung Berung	Jl. Sukup Baru No.1A, Pasir Endah, Kec. Ujung Berung	Bandung	0227837998	uber.PNG	-6.9105164
2	Buah Batu	Jl. Buah Batu No.201A, Turangga, Kec. Lengkong	Bandung	0227303551	bubat.PNG	-6.9422204
3	Kiaracondong	Jl. Kiaracondong No.301, Kb. Kangkung, Kec. Kiaracondong	Bandung	0227311774	kircon.PNG	-6.92930436
4	Metro	Jl. Venus Raya No.14, Manjahlega, Kec. Rancasari	Bandung	0227569038	metro.PNG	-6.9407841

Gambar 5.26 Antarmuka datapemetaan

Dari tampilan diatas, menampilkan data-data dari tabel pemetaan, dan admin dapat melakukan export data tersebut kedalam file berekstensi csv,excel,pdf dan dapat print.

Setelah itu kita akan membuat fungsi update cabang, yang berguna untuk mengubah data dari tabel pemetaan, buat fungsi baru dengan nama **updatecabang(\$param = 0)** berikut sourcenyanya,

```
public function updatecabang($param = 0)
{
    $data['title'] = 'Data Pemetaan';
    $data['user'] = $this->db-
>get_where('tbl_user', ['user_email' =>
    $this->session->userdata('user_email')])->row_array();
```

```

        $this->form_validation-
>set_rules('nama_cabang', 'Nama Cabang', 'trim|required');
        $this->form_validation-
>set_rules('alamat', 'Alamat', 'trim|required');
        $this->form_validation-
>set_rules('kota', 'Kota', 'trim|required');
        $this->form_validation-
>set_rules('no_telp', 'No Telp', 'trim|required');
        $this->form_validation-
>set_rules('latitude', 'Latitude', 'trim|required');
        $this->form_validation-
>set_rules('longitude', 'Longitude', 'trim|required');
        $this->form_validation-
>set_rules('deskripsi', 'deskripsi', 'trim|required');

        if ($this->form_validation->run() == TRUE) {
            $this->Gis_Model->updatecabang($param);

            redirect('pemetaan/datapemetaan');
        }

        $data['pemetaan'] = $this->Gis_Model-
>getcabang($param);
        $data['cabang'] = $this->Gis_Model->cabang();

        $config['map_div_id'] = "map-add";
        $config['map_height'] = "250px";
        $config['center'] = $data['pemetaan']-
>latitude . ',' . $data['pemetaan']->longitude;
        $config['zoom'] = '15';
        $config['map_height'] = '300px;';
        $this->googlemaps->initialize($config);

        $marker = array();

```

```

        $marker[ 'position' ] = $data[ 'pemetaan' ]-
>latitude . ',' . $data[ 'pemetaan' ]->longitude;
        $marker[ 'draggable' ] = true;
        $marker[ 'ondragend' ] = 'setMapToForm(event.lat
LnG.lat(), event.latLng.lng());';
        $this->googlemaps->add_marker($marker);
        $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('menu/v_updatepemetaan', $data);
        $this->load->view('temanbody/footer');
    }
}

```

Source diatas berfungsi untuk mengubah atau mengapdate data proses nya akan di lakukan di model **Gis_model.php** pada fungsi **updatecabang(\$param = 0)** berikut source nya,

```

public function updatecabang($param = 0)
{
    $pemetaan = $this->getcabang($param);

    $config[ 'upload_path' ] = './assets/images/';
    $config[ 'allowed_types' ] = 'gif|jpg|png';
    $config[ 'max_width' ] = 1024 * 3;
    $config[ 'max_height' ] = 768 * 3;

    $this->load->library('upload', $config);
    $this->upload->initialize($config);
    if (!$this->upload->do_upload('image')) {
        $img_name = $pemetaan->image;
    }
}

```

```
        $error = $this->upload->display_errors();
        print_r($error);
    } else {
        $photo = $this->upload->data();
        $img_name = $photo['file_name'];

        $object = array(
            'nama_cabang' => $this->input-
>post('nama_cabang'),
            'alamat' => $this->input-
>post('alamat'),
            'kota' => $this->input->post('kota'),
            'no_telp' => $this->input-
>post('no_telp'),
            'image' => $img_name,
            'latitude' => $this->input-
>post('latitude'),
            'longitude' => $this->input-
>post('longitude'),
            'deskripsi' => $this->input-
>post('deskripsi')
        );

        $this->db-
>update('pemetaan', $object, array('id_cabang' => $par
am));
        $this->session-
>set_flashdata('message', '<div class="alert alert-
success" role="alert">Cabang, Updated!</div>');
    }
}
```

Pada source diatas dilakukanlah proses pengupdatean data pada tabel pemetaan yang akan ditampilkan di view , buat terlebih dahulu view baru dalam folder menu beri nama **v_updatepemetaan.php** berikut sourcenyanya,

```
<div id="content-wrapper">
    <div class="container-fluid">
        <!-- Breadcrumbs-->
        <ol class="breadcrumb">
            <li class="breadcrumb-item">
                <a href="">Pemetaan</a>
            </li>
            <li class="breadcrumb-item active">Update Management Pemetaan</li>
        </ol>
        <?php echo $this->session->flashdata('message'); ?>
    </div>
    <div class="row">
        <div class="col-lg-12">
            <div class="container">
                <div class="card bg-light mb-6" style="max-width: 80rem;">
                    <div class="card-header">Form Update Pemetaan Cabang Bandung 2
                    </div>
                    <div class="card-body">
                        <form action="<?= current_url(); ?>" method="post" enctype="multipart/form-data">
                            <form class="form-horizontal">
                                <input type="hidden" class="form-control" id="id_cabang" name="id_cabang" value="<?= $pemetaan->id_cabang; ?>">
```

```
<div class="form-
group">
    <label>Cabang Pemb
uka</label>
    <select name="id_c
abang" class="form-control" required>
        <option select
ed value="<?= $pemetaan-
>id_cabang; ?>" required><?= $pemetaan-
>nama_cabang; ?></option>
        <?php foreach
($cabang as $c) { ?>
            <option va
lue="<?= $c->id ?>"><?= $c->cabang ?></option>
        <?php } ?>
    </select>
</div>
<div class="form-
group">
    <label for="alamat
">Alamat :</label>
    <div class="form-
group">
        <textarea name
="alamat" class="form-control" rows="3"><?= $pemetaan-
>alamat; ?></textarea>
        <p class="help
-
block"><?php echo form_error('alamat', '<small class="text-red">', '</small>'); ?></p>
    </div>
</div>
<div class="form-
group">
    <label for="kota">
Kota</label>
```

```
        <input type="text"
    class="form-
control" id="kota" name="kota" value="<?= $pemetaan-
>kota; ?>">
                <?php echo form_er
ror('kota', '<small class="text-danger pl-
3">', '</small>'); ?>
            </div>
            <div class="form-
group">
                <label for="no_tel
p">No Telp</label>
                <input type="text"
    class="form-
control" id="no_telp" name="no_telp" value="<?= $pemet
aan->no_telp; ?>">
                <?php echo form_er
ror('no_telp', '<small class="text-danger pl-
3">', '</small>'); ?>
            </div>
            <div class="form-
group">
                <label for="image"
>Image</label><br>
            <div class="custom-
file col-sm-4">
                <input type="f
ile" class="custom-file-
input" id="image" name="image">
                <label class="
custom-file-label" for="image">Choose file</label>
            </div>
            <div class="custom-
file col-md-4">
                <?php if ($pem
etaan->image != '') : ?>
```

```
image}") ?>" height="150" class="rounded">
<?php endif; ?
>
</div>
</div><br>
<div class="form-
group">
<label for="deskri
psi">Deskripsi :</label>
<div class="form-
group">
<textarea name
="deskripsi" class="form-
control" rows="6"><?= $pemetaan-
>deskripsi; ?></textarea>
<p class="help
-
block"><?php echo form_error('deskripsi', '<small clas
s="text-red">', '</small>'); ?></p>
</div>
</div class="form-
group">
<label for="koordi
nat">Koordinat :</label>
<label class="sr-
only" for="latitude">Latitude</label>
<div class="input-
group mb-2 mr-sm-2">
<div class="in
put-group-prepend">
<div class
="input-group-text"><i class="fa fa-map-marker-
alt"></i></div>
</div>
```

```
        <input type="t
ext" class="form-
control" id="latitude" name="latitude" value="<?= $pem
etaan->latitude; ?>">
        <p class="help
-
block"><?php echo form_error('latitude', '<small class
="text-red">', '</small>'); ?></p>
        </div>
        <label class="sr-
only" for="longitude">Longitude</label>
        <div class="input-
group mb-2 mr-sm-2">
        <div class="in
put-group-prepend">
            <div class
="input-group-text"><i class="fa fa-map-marker-
alt"></i></div>
            </div>
            <input type="t
ext" class="form-
control" id="longitude" name="longitude" value="<?= $p
emetaan->longitude; ?>">
            <p class="help
-
block"><?php echo form_error('longitude', '<small clas
s="text-red">', '</small>'); ?></p>
            </div>
            <div class="col-
md-offset-2">
                <?php echo $ma
p['html'] ?>
                <?php echo $ma
p['js'] ?>
            </div>
        </div>
    </div>
```

```
<div class="form-group">
    <button type="submit" class="btn btn-success mb-3"><i class="fa fa-save"></i> Save</button>
    <button class="btn btn-warning mb-3" onClick="refreshPage()"><i class="fa fa-trash-restore-alt"></i> Reset</button>
</div>
</div>
</div>
</div>
</form>
</div>
<br>
```

Source diatas merupakan antarmuka perngisian atau penginputan form yang akan di lakukan perubahan data, berikut tampilanya,

Gambar 5.27 Antarmuka Update Data Pemetaan

Dalam halaman ini admin dapat melakukan update atau perubahan data dengan mengisi semua form yang tersedia.

Pembuatan fungsi delete pemetaan, buat fungsi baru pada file controller **pemetaan.php** beri nama fungsi tersebut **deletecabang(\$id_cabang)** berikut sourcenyanya,

```

public function deletecabang($id_cabang)
{
    if (empty($id_cabang)) {
        redirect('pemetaan');
    } else {
        $where = array('id_cabang' => $id_cabang);
        $this->Gis_Model-
>deletecabang($where, 'pemetaan');
        $this->session-
>set_flashdata('message', '<div class="alert alert-
success" role="alert">Data Cabang Deleted!</div>');
        redirect('pemetaan/datapemetaan');
    }
}

```

```
        }
    }
}
```

Source diatas merupakan proses penghapusan data pada database tabel pemetaan, dengan id_cabang sebagai primary nya, proses akan di lakukan di model **Gis_model.php** pada fungsi **deletecabang()** , berikut sourcennya,

```
public function deletecabang($where, $table)
{
    $this->db->where($where);
    $this->db->delete($table);
}
```

Source diatas merupakan proses penghapusan data pada database tabel pemetaan.

5.9 Pembuatan Halaman Pelelangan

Pertama buat terlebih dahulu file baru di controller dengan nama **pelelangan.php** buat beberapa fungsi berikut source nya,

```
<?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');
    }
}
```

Source diatas terdapat fungsi construct yang berisi library check_login dan model yang bernama **Model_produk** , model ini berisi proses-proses seperti penambahan, pengubahan, dan penghapusan data.

Pada fungsi **index()** memberikan inisiasi pada variabel \$data yang berisi parameter produk dan cabang, yang nantinya di olah atau di proses di **Model_produk** dan akan di tampilkan pada view.

Fungsi **create_barang()** , berikut source lanjutanya,

```
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/' / 
path folder
    $config['allowed_types'] = 'gif|jpg|png|jpeg|b
mp'; //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');
    }
}

```

Source diatas bertujuan untuk melakukan penginputan barang lelangan yang akan kita tampilkan pada menu user nanti

```
$config['upload_path'] = './assets/images/';
```

Source diatas berfungsi sebagai path gambar yang akan kita upload akan di simpan di folder assets/images

```
$config['allowed_types'] = 'gif|jpg|png|jpeg|bmp';
```

Source diatas berfungsi untuk tipe yang dapat diupload oleh admin berformat seperti diatas bisa kalian sesuaikan

```

$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);

```

Source diatas berupa proses inisiasi penginputan menyesuaikan dengan field yang ada pada tabel pelelangan, yang nantinya akan di proses input data ke model dengan fungsi bernama **create_barang(\$data_product)** , berikut source model nya sebelumnya buat terlebih dahulu model dengan nama **Model_produk.php** dan isi source didalamnya seperti berikut,

```

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

class Model_produk extends CI_Model

```

```
{  
public function create_barang($data_produk)  
{  
    $this->db->insert('pelelangan', $data_produk);  
}
```

Source diatas merupakan proses penginputan data kedalam database tepatnya tabel pelelangan berikut, contoh gambar isi dari table pelelangan,

+ Options								
	← ↑ ↓ →	id_barang	nama_barang	harga_barang	id_cabang	tgl_buka	gambar	deskripsi
<input type="checkbox"/>		15	Vario 125 CCC 2014	7500000	5	1579424390	vario.jpg	Lorem ipsum dolor sit amet, consectetur adipiscing...
<input type="checkbox"/>		18	HRV Prestige 2018	25000000	2	1576711844	hrv.jpg	Lorem ipsum dolor sit amet, consectetur adipiscing...
<input type="checkbox"/>		19	TV LED 40 LG Putih	2500000	3	1576837453	tv.jpeg	Lorem ipsum dolor sit amet, consectetur adipiscing...
<input type="checkbox"/>		20	Samsung S10+	11200000	4	1576837702	hp.jpg	Lorem ipsum dolor sit amet, consectetur adipiscing...
<input type="checkbox"/>		21	Asus X456LJ Core i5	6000000	1	1577091718	Laptop.jpg	Intel Core i5-5200U Broadwell Layar 14'' RAM 8GB ...

Gambar 5.28 Tabel Pelelangan

Setelah membuat fungsi create tadi selanjutnya kita membuat fungsi edit atau update, buat fungsi baru di controller pelelangan eri nama **update_barang()** berikut sourcennya,

```
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|b  
mp';  
  
 $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'),
        );
        $this->Model_produ-
>update_barang($id_barang, $data_produk);
    }
    $this->session-
>set_flashdata('message', '<div class="alert alert-
success" role="alert">Data Barang Updated!</div>');
    redirect('pelelangan');
}
```

Source diatas berfungsi untuk prubahan data atau update data pada database tabel pelelangan dengan source yang hampir sama dengan inputan sebelumnya tetapi proses ini terdapat kondisi dan proses nya juga di lakukan pada model dengan fungsi yang berbeda, buat fungsi baru pada model Model_produk beri nama fungsi tersebut **update_barang(\$data_produk)** berikut sourcenyanya,

```
public function update_barang($id_barang, $data_pr
oduk)
{
    $this->db->where('id_barang', $id_barang);
    return $this->db-
>update('pelelangan', $data_produk);
}
```

Source diatas berfungsi sebagai proses perubahan data atau update data pada database tabel pelelangan dengan membawa variabel id_barang dan dataproduk serta ini siasi dari primary pada tabel pelelangan yaitu id_barang.

Setelah membuat fungsi update kita akan membuat fungsi baru yaitu delete kita akan membuat fungsi baru pada controller pelelangan beri nama **delete_barang(\$id_barang)** berikut sourcenya,

```
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');
    }
}
```

Source diatas bertujuan untuk proses inisiasi sebelum di eksekusi di model, dengan membawa variabel \$id_barang, buat fungsi baru pada model_produk beri nama **delete_barang(\$where,\$table)** , berikut sourcenyanya,

```
public function delete_barang($where, $table)
{
    $this->db->where($where);
    $this->db->delete($table);
}
```

Source diatas merupakan proses penghapusan data pada database tabel pelelangan, dengan membawa variabel where dan table yang sebelumnya sudah di inisiasi pada controller fungsi delete_barang(\$id_barang)

Setelah semua fungsi dibuat kita akan membuat tampilanya sebelumnya buat file baru dlu view di dalam folder menu, beri nama **v_pelelangan.php** berikut source tampilanya,

```

        <th width="20%">Nama B
arang</th>
        <th width="8%">Harga</
th>
        <th width="15%">Tanggal
Pembuka</th>
        <th width="10%">Cabang
Pembuka</th>
        <th width="10%">Deskri
psi</th>
        <th width="15%">Gambar
</th>
        <th width="15%">
            <button class="btn
btn-primary btn-sm" data-toggle="modal" data-
target="#addbarang"><i class="fas fa-fw fa-plus-
square" style="text-
align: center;"></i> Tambah</button>
        </th>
    </tr>
</thead>
<tbody>
    <?php $q = 1; ?>
    <?php foreach ($produk as
$barang) : ?>
        <tr>
            <td><?= $q; ?></td
>
            <td><?= $barang['n
ama_barang']; ?></td>
            <td><?= number_for
mat($barang['harga_barang'], 0, ',', '.'); ?></td>
            <td><?= date('d F
Y', $barang['tgl_buka']) ?></td>
            <td><?= $barang['c
abang']; ?></td>

```

```
<td><textarea read  
only style="width:300px; height:120px;"><?= $barang['d  
eskripsi']; ?></textarea></td>  
        <td></td>  
        <td>  
            <a class="badg  
e badge-warning" data-toggle="modal" data-  
target="#barangedit<?= $barang['id_barang']; ?>">Edit<  
/a>  
            <a class="badg  
e badge-  
danger" href="= base_url('pelelangan/delete_barang/'<br/. $barang['id_barang']); ?>" onclick="return validasi  
();">Delete</a>  
        </td>  
    </tr>  
<?php $q++; ?>  
  
        <!-- Modal Update-->  
        <div id="barangedit<?= $barang['id_barang']; ?>" tabindex="-  
1" class="modal fade">  
            <div class="modal-  
dialog" role="document">  
                <div class="mo  
dal-content">  
                    <div class  
="modal-header">  
                        <h5 cl  
ass="modal-title">Update Barang</h5>  
                        <button  
type="button" class="close" data-  
dismiss="modal" aria-label="Close">  
                            <s  
pan aria-hidden="true">&times;</span>
```

```
</button>
</div>
<div class="modal-body">
    <form action="= base_url('pelelangan/update_barang'); ?&gt;" method="post" enctype="multipart/form-data"&gt;
        &lt;i&gt;
            &lt;input style="font-size: 15px" style="font-size: 15px" type="hidden" class="form-control" name="id_barang" value="<?= $barang['id_barang']; ?&gt;"&gt;
        &lt;i&gt;
            &lt;input style="font-size: 15px" style="font-size: 15px" type="hidden" class="form-control" name="gambar2" value="<?= $barang['gambar']; ?&gt;"&gt;
        &lt;d&gt;
        &lt;div class="form-group"&gt;
            &lt;label&gt;Nama Barang&lt;/label&gt;
            &lt;input type="text" class="form-control" name="nama_barang" value="<?= $barang['nama_barang']; ?&gt;" required&gt;
        &lt;/div&gt;
        &lt;d&gt;
        &lt;div class="form-group"&gt;
            &lt;label&gt;Harga Barang&lt;/label&gt;
            &lt;input type="text" class="form-control" name="harga_barang" value="<?= $barang['harga_barang']; ?&gt;" onkeypress='return event.charCodeAt &gt;= 48 &amp;&amp; event.charCodeAt &lt;= 57' required&gt;
        &lt;/div&gt;
    &lt;/form&gt;
&lt;/div&gt;</pre
```

```
</
div>
<d
iv class="form-group">

    <label>Cabang Pembuka</label>

    <select name="id_cabang" class="form-
control" required>

        <option selected value=<?= $barang['id_cabang'] ?><?= $barang['cabang']; ?>></option>

        <?php foreach ($cabang as $c) { ?>

            <option value=<?= $c->id ?><?= $c-
>cabang ?>></option>

        <?php } ?>

    </select>
</
div>
<d
iv class="form-group">

    <label>Deskripsi</label>

    <textarea rows="5" class="form-
control" name="deskripsi" value='<?= $barang['deskripsi']; ?>' required><?= $barang['deskripsi']; ?></textar
ea>
</
div>
<d
iv class="form-group">
```

```
<div class="custom-file">

    <input type="file" class="custom-file-input" id="gambar" name="gambar">

    <label class="custom-file-label" for="gambar"><?= $barang['gambar']; ?></label>

</div>

<div class="col-md-4">

    <br>

</div>
</div>

<div class="modal-footer">

    <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

    <button type="submit" class="btn btn-primary">Update</button>
</div>
</div>
</form>
</div>
```

```
                </div>
            </div>
        </tbody>
    <?php endforeach; ?>
</table>
</div>
</div>
<div class=" col-md-1">
</div>
</div>
</div>
</div>
</div>
</div>

<div id="addbarang" class="modal fade" tabindex="-1" role="dialog">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-
title">Tambah Barang</h5>
                <button type="button" class="close" da
ta-dismiss="modal" aria-label="Close">
                    <span aria-
hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                <form action=<?= base_url() . 'pelela
ngan/create_barang'; ?>" method="post" enctype="multip
art/form-data">

                    <div class="form-group">
                        <label>Nama Barang</label>
                        <input type="text" required cl
ass="form-control" value="" name="nama_barang">
                    </div>
                    <div class="form-group">
```

```
        <label>Harga</label>
        <input type="text" required on
keypress='return event.charCodeAt >= 48 && event.charCodeAt <= 57' class="form-control border-
input" value="" name="harga_barang">
    </div>
    <!-- <div class="form-group">
        <label>Tanggal Pembuka</label>
        <input type="date" required cl
ass="form-control border-
input" value="" name="tgl_buka" hidden>
    </div> -->
    <div class="form-group">
        <label>Cabang Pembuka</label>
        <select name="id_cabang" class
="form-control" required>
            <?php foreach ($cabang as
$g) : ?>
                <option value="<?= $g-
>id ?><?= $g->cabang ?>></option>
            <?php endforeach ?>
        </select>
    </div>
    <div class="form-group">
        <label>Deskripsi</label>
        <textarea name="deskripsi" cla
ss="form-
control" required cols="30" rows="5"></textarea>
    </div>
    <div class="form-group">
        <div class="custom-file">
            <input type="file" class="
custom-file-input" id="gambar" name="gambar">
            <label class="custom-file-
label" for="gambar">Choose file</label>
        </div>
    </div>
```

```
<div class="clearfix"></div>
<div class="modal-footer">
    <button type="submit" class="button button-primary button-large" data-dismiss="modal" data-bbox="76 291 218 311">Add Barang</button>
    <button type="button" class="button button-secondary" data-dismiss="modal" data-bbox="76 311 218 331">Tutup</button>
</div>
</div>
</div>

<script>
$(document).ready(function() {
    $('#myTable').DataTable();
});
</script>
```

Source diatas merupakan tampilan dari menu pelelangan semua proses pengolahan data dibuat menggunakan modal berupa tampilan pop up window baru, dengan tambahan source js untuk membuat export file lebih cepat source diatas merupakan inisiasi dari controller dan model berikut tampilanya dari data pelelangan,

No	Nama Barang	Harga	Tanggal Pembuka	Cabang Pembuka	Deskripsi	Gambar	Aksi
1	Vario 125 CC 2014	7.500.000	19 January 2020	Metro	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		<button>Edit</button> <button>Delete</button>
2	HRV Prestige 2018	250.000.000	19 December 2019	Buah Batu	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		<button>Edit</button> <button>Delete</button>
3	TV LED 40 LG Putih	2.500.000	20 December 2019	Ciparay	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		<button>Edit</button> <button>Delete</button>

Gambar 5.28 Antarmuka Pelelangan

Pada tampilan diatas admin dapat melakukan pengolahan data seperti penambahan,pengeditan atau penghapusan data.

Untuk tampilan pada sisi member buat fungsi baru di controller user beri nama fungsi **lelang(\$id = 0)** berikut sourcenya,

```
public function lelang($id = 0)
{
    $data['title'] = "Lelang";
    $data['user'] = $this->db-
>get_where('tbl_user', ['user_email' =>
    $this->session->userdata('user_email')])->row_array();
```

```
        if ($id == 0) {
            $data['produk'] = $this->Model_produk-
>all();
            $data['cabang'] = $this->Model_produ-
>cabang();
            $this->load-
>view('temanbody/header', $data);
            $this->load-
>view('temanbody/topbar', $data);
            $this->load-
>view('temanbody/sidebar', $data);
            $this->load->view('menu/v_lelang', $data);
            $this->load->view('temanbody/footer');
        } else {
            $data['produk'] = $this->Model_produk-
>allcabang($id);
            $data['cabang'] = $this->Model_produ-
>cabang();
            $this->load-
>view('temanbody/header', $data);
            $this->load-
>view('temanbody/topbar', $data);
            $this->load-
>view('temanbody/sidebar', $data);
            $this->load->view('menu/v_lelang', $data);
            $this->load->view('temanbody/footer');
        }
    }
}
```

Source diatas merupakan perantara antara model dan view dengan kondisi variabel \$data yang berisi parameter produk untuk

menampilkan semua produk dan parameter cabang untuk menampilkan cabang pada produk yang di upload.

Buat fungsi baru pada Model_produk beri nama **all()** berikut sourcenya,

```
public function all()
{
    $query = "SELECT `pelelangan`.* , `cabang` . `caba
ng` .
    FROM `pelelangan` JOIN `cabang`
    ON `pelelangan` . `id_cabang` = `cabang` .
    `id` ;
    return $this->db->query($query)-
>result_array();
}
```

Source diatas merupakan proses menampilkan data pada database tepatnya tabel pelelangan yang di relasikan (join) dengan tabel cabang.

Setelah itu membuat fungsi baru beri nama **cabang()** , berikut sourcenya,

```
public function cabang()
{
    $a = $this->db->get('cabang');
    return $a->result();
}
```

Source diatas berfungsi untuk mengambil data pada tabel cabang.

Setelah pembuatan fungsi tersebut kita akan membuat tampilan pada view buat dalam folder menu beri nama **v_lelang.php** berikut sourcennya,

```
<div id="content-wrapper">

    <div class="container-fluid">

        <!-- Breadcrumbs-->
        <ol class="breadcrumb">
            <li class="breadcrumb-item">
                <a href="">User</a>
            </li>
            <li class="breadcrumb-item active">Lelang</li>
        </ol>

        <center>
            <div class="btn-group">
                <button type="button" class="btn btn-danger">Pilih Cabang</button>
                <button type="button" class="btn btn-danger dropdown-toggle dropdown-toggle-split" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                    <span class="sr-only">Toggle Dropdown</span>
                </button>
                <div class="dropdown-menu">
```

```
<a class="dropdown-item" href="= base_url('user/lelang/') ?&gt;"&gt;All&lt;/a&gt;
        &lt;?php foreach ($cabang as $key) {
?&gt;
            &lt;a class="dropdown-item" href="<?= base_url('user/lelang/' . $key-&gt;id); ?&gt;&lt;?= $key-&gt;cabang; ?&gt;&lt;/a&gt;
                &lt;?php } ?&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/center&gt;
&lt;br&gt;

&lt;div class="row"&gt;
    &lt;?php foreach ($produk as $key) { ?&gt;
        &lt;div class="col-lg-4"&gt;
            &lt;div class="card mb-3"&gt;
                &lt;div class="card-header"&gt;
                    Cabang Pembuka :
                    &lt;span class="badge badge-primary"&gt;&lt;?php echo $key['cabang']; ?&gt; &lt;/span&gt;&lt;/div&gt;
                &lt;div class="card-body"&gt;
                    &lt;img class="rounded" src="<?= base_url() . '/assets/images/' . $key['gambar']; ?&gt;" style="height:300px; width:300px" alt="product-img" /&gt;
                    &lt;a href="<?= base_url('user/produk_detail/' . $key['id_barang']); ?&gt;"&gt;&amp;nbsp;
                        &lt;center&gt;&lt;button class="btn btn-danger btn-sm" type="button"&gt;Detail Barang&lt;/button&gt;&lt;/center&gt;
                    &lt;/a&gt;
                &lt;/div&gt;
            &lt;div class="card-footer small text-muted"&gt;</pre
```

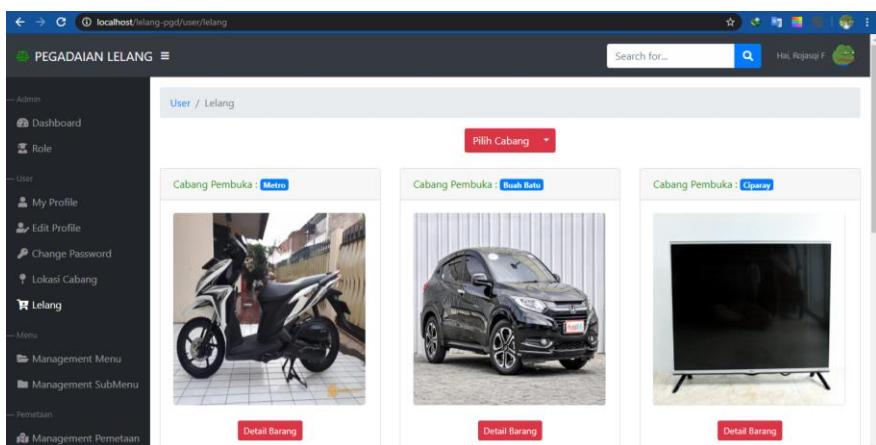
```

<h5 style="color: #0078FF;
"><?php echo $key['nama_barang'] ?></h5>
<p class="harga_barang"><b>
><?= 'Rp. ' . number_format($key['harga_barang'], 0, ', ','.') ; ?></b></p>
</div>
</div>
<?php } ?>
</div>

</div>
</div>

```

Source diatas bertujuan untuk menampilkan produk-produk lelang beserta harga,cabang,dll, dengan inisiasi dari controller sebelumnya. Berikut tampilan dari menu lelang ini,



Gambar 5.29 Antarmuka Lelang

Pada tampilan diatas user dapat mencari barang yang diinginkan untuk di beli user dapat melihat spesifikasi detail dari barang yang di tampilkan dengan cara mengklik button detail barang.

5.10 Pembuatan Halaman Penawar

Masih pada controller yang sama yaitu pelelangan dalam file tersebut tambahkan kembali fungsi baru beri nama **datapenawar()** berikut source nya,

```
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');
}
```

Source diatas merupakan perantara antara model dan view terdapat variabel data berisi parameter penawar yang nantinya akan di panggil pada view data penawar, data yang akan ditampilkan di view akan di proses terlebih dahulu di dalam model bernama Model_produk dengan fungsi baru bernama **bid()**, berikut source fungsi model baru dari **bid()**

```
public function bid()
{
    $this->db-
>join('pelelangan', 'bid.id_barang = pelelangan.id_bar-
ang', 'left');
    $this->db-
>join('cabang', 'bid.id_cabang = cabang.id', 'left');
    $this->db-
>join('tbl_user', 'bid.id_user = tbl_user.user_id', 'l-
eft');
    $w = $this->db->get('bid');
    return $w->result();
}
```

Source diatas bertujuan untuk merelasikan tabel-tabel (join) seperti tabel pelelangan,cabang,tbl_user dengan penggunaan fungsi left mengambil sejumlah data mulai dari kiri string, setelah di proses di model maka akan di tampilkan di view, lalu buat terlebih dahulu view baru di folder menu beri nama **datapenawar.php** berikut sourcenyanya,

```
<div id="content-wrapper">
```

```
<div class="container-fluid">

    <!-- Breadcrumbs-->
    <ol class="breadcrumb">
        <li class="breadcrumb-item">
            <a href="">Pelelangan</a>
        </li>
        <li class="breadcrumb-item active">Data Penawar</li>
    </ol>
    <!-- Page Content -->
    <div class="row">
        <div class="col-lg-12">
            <?= form_error('menu', '<div class="alert alert-danger" role="alert">', '</div>'); ?>

            <?= $this->session->flashdata('message'); ?>
            <table class="display" id="example" style="width:100%">
                <thead>
                    <tr>
                        <th scope="col">No</th>
                        <th scope="col">Nama Barang</th>
                        <th scope="col">Harga Barang</th>
                        <th scope="col">Jumlah Penawar</th>
                        <th scope="col">Cabang</th>
                        <th scope="col">Email Penawar</th>
                        <th scope="col">Tanggal</th>
                </thead>
```

```
<th scope="col">Action</th>
>
</tr>
</thead>
<tbody>

    <?php $q = 1; ?>
    <?php foreach ($penawar as $m) : ?>
        <tr>
            <th scope="row"><?= $q ; ?></th>
            <td><?= $m->nama_barang; ?></td>
            <td><?= $m->harga_barang; ?></td>
            <td><?= $m->harga_lelang; ?></td>
            <td><?= $m->cabang; ?></td>
            <td><?= $m->user_email;; ?></td>
            <td><?= $m->tanggal; ?></td>
            <td>
                <a class="badge badge-warning" id="status" href="<?= base_url(); ?>index.php/pelelangan/updateStatus/<?= $m->id_lelang_bid; ?>/<?= $m->id_user; ?>/<?= $m->id_barang; ?>/<?= $m->id_cabang; ?>"><?= $m->status; ?></a>
                <a class="badge badge-danger" href="<?= base_url('pelelangan/delete_bid/' . $m->
```

```
>id_lelang_bid); ?>" onclick="return confirm('Are you  
sure?')">Delete</a>  
        </td>  
    </tr>  
    <?php $q++; ?>  
  </tbody>  
  <?php endforeach; ?>  
 </table>  
  
      </div>  
    </div>  
  </div>  
</div>  
  
<script>  
  $(document).ready(function() {  
    $(document).on('click', '#status', function()  
{  
      console.log('aw')  
    })  
  
  })  
</script>
```

Source front-end diatas bertujuan untuk membuat sebuah tampilan