

**PANDUAN MEMBANGUN SEBUAH
APLIKASI NOTIFIKASI OTOMASI
PEMELIHARAAN ASSET MENGGUNAKAN
BAHASA PEMOGRAMAN PHP DENGAN
*FRAMEWORK CODEIGNITER***

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 RAD (*Rapid Application Development*).

BAB 1

Pengenalan Dasar

Dalam buku panduan/tutorial ini penulis akan membangun sebuah aplikasi berbasis website dinamis dan responsif dengan studi kasus notifikasi otomasi pemeliharaan *asset*, dikembangkan dengan metode RAD (*Research Application Development*) akan dibangun menggunakan bahasa pemograman PHP pada *Back-end* dan dibantu *framework Codeigniter* serta untuk *front-end* menggunakan *Bootstrap* dengan sedikit tambahan jQuery dan *Database MYSQL*.

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

Pertama kali PHP di luncurkan memiliki kepanjangan dari *Personal Home Page* (Situs personal). Rasmus Lerdorf adalah orang yang membuat/menciptakan pertama kali atau pada tahun 1995. Saat itu PHP masih berwujud berupa sekumpulan kode/skrip yang digunakan untuk mengolah data formulir dari web.

Rasmus Lerdorf merilis skripnya secara umum dan menamainya dengan PHP/FI, dengan perilisan PHP/FI tersebut, skrip nya menjadi sumber terbuka untuk semua orang khususnya para pemrogram banyak yang tertarik untuk ikut mengembangkanya. Kemudian pada tahun 1997 Rasmus Lerdorf merilis PHP/FI 2.0 dimana interpreter PHP ini sudah di implementasikan dalam pemrograman C dan juga skripnya banyak digunakan dan dikembangkan, sampai sebuah perusahaan bernama Zend menulis ulang skrip PHP yang dibuat oleh Rasmus

Lerdorf menjadi lebih baik dan lebih cepat daripada sebelumnya. Pada bulan Juni tahun 1998 perusahaan Zend menulis ulang dan skrip PHP dengan gaya baru dan diresmikan menjadi PHP 3.0 dan merubah kepanjangannya menjadi *PHP : Hypertext Preprocessor*.

Pada pertengahan tahun 1999 Perusahaan Zend meng *upgrade* lagi PHP dan merilisnya dengan nama PHP 4.0. 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>Contoh Code PHP</title>

</head>

<body>

<?php

$ulangi = 1;

while ($ulangi <= 32) {

    echo "<p>Ini adalah perulangan ke-$ulangi</p>";

    $ulangi++;

}

?>

</body>

</html>
```

Pada sintaks diatas akan menampilkan *output* perulangan Ini adalah perulangan ke - 1 sampai dengan perulangan ke – 32. Seperti gambar berikut :

Ini adalah contoh perulangan ke-1
Ini adalah contoh perulangan ke-2
Ini adalah contoh perulangan ke-3
Ini adalah contoh perulangan ke-4
Ini adalah contoh perulangan ke-5
Ini adalah contoh perulangan ke-6
Ini adalah contoh perulangan ke-7
Ini adalah contoh perulangan ke-8
Ini adalah contoh perulangan ke-9
Ini adalah contoh perulangan ke-10
Ini adalah contoh perulangan ke-11
Ini adalah contoh perulangan ke-12
Ini adalah contoh perulangan ke-13
Ini adalah contoh perulangan ke-14
Ini adalah contoh perulangan ke-15
Ini adalah contoh perulangan ke-16
Ini adalah contoh perulangan ke-17
Ini adalah contoh perulangan ke-18
Ini adalah contoh perulangan ke-19
Ini adalah contoh perulangan ke-20
Ini adalah contoh perulangan ke-21
Ini adalah contoh perulangan ke-22
Ini adalah contoh perulangan ke-23
Ini adalah contoh perulangan ke-24
Ini adalah contoh perulangan ke-25
Ini adalah contoh perulangan ke-26
Ini adalah contoh perulangan ke-27
Ini adalah contoh perulangan ke-28
Ini adalah contoh perulangan ke-29
Ini adalah contoh perulangan ke-30
Ini adalah contoh perulangan ke-31
Ini adalah contoh perulangan ke-32

Gambar 1.2 Contoh Sintaks PHP

1.2 Pengenalan jQuery

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

Kemudian para pengembang bertemu untuk membantu menyempurnakan *library* yang dibuat oleh John Resig, dan pada tahun 2006 John resig dan para pengembang berhasil berhasil merilis jQuery versi 1.0. Sejak itu, jQuery telah banyak berkembang, dan terus di *upgrade* kemampuannya dengan rilis versi 1.7.1.

1.2.1 Penjelasan jQuery



Gambar 1.3 jQuery

jQuery adalah modul/*library* dari sebuah Javascript lintas *platform*, sebelumnya jika anda kurang memahami apa itu Javascript anda dapat mencari tau lebih detailnya di buku ataupun internet. Begitu banyak para *pengembang khususnya web* menggunakan *library* dari javascript ini, karena penulisan skrip codingnya yang pendek dan mudah dimengerti.

jQuery dibuat untuk memperingkas dan menyederhanakan skrip-skrip yang ada di javascript. JQuery merupakan *library* yang cepat dan ringan untuk mengelola dan mengatasi dokumen HTML, Event, 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. jQuery sediri merupakan *library* yang membuat sebuah aplikasi web dari sisi *client*, tidak terlihat sebagai *script JavaScript* yang biasa pada umumnya, 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 script Javascript biasa.
2. AJAX : Mengambil informasi dari server tanpa me-refresh seluruh form halaman.

3. Mengatasi masalah kompatibilitas antar-browser.
4. Kemudahan dalam mengakses elemen-elemen HTML.
5. Memanipulasi elemen HTML,CSS dan Penanganan event HTML

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 menyisipkanya ke skrip javascript anda,

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

Berikut contohnya.

```
<!DOCTYPE html>

<html>

<head>

    <script
    src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.mi
n.js">
```

```
</script>

<script>
$(document).ready(function() {
    $("button").click(function() {
        $("p").text("Hello world!");
    });
</script>

</head>

<body>
    <button>Ubah isi kata untuk semua element </button>
    <p>Ubah kata satu.</p>
    <p>ubah kata dua.</p>
</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,

Ubah isi kata untuk semua element

Ubah kata satu.

ubah kata dua.

Ubah isi kata untuk semua element

Hello world!

Hello world!

Gambar 1.4 Contoh Sintaks jQuery

1.3 Pengenalan Framework

kenapa harus menggunakan *framework*? alasanya ialah begitu banyak kemudahan dalam membangun sebuah *web* menggunakan *framework* sehingga membuat skrip menjadi singkat dan praktis. Dalam *framework* juga sudah tersedia struktur pengaplikasian yang baik seperti,

standard coding, standar penulisan skrip/code yang harus di ikuti oleh programmer/develop.

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

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

dan ***common function***, adanya fungsi/library (pustaka) yang umum yang sering di kembangkan dalam membagun sebuah sistem/aplikasi.

Dengan adanya bantuan *framework* ini kita dapat langsung mempermudah dan fokus pada bisnis proses / latar belakang masalah yang di hadapi dalam membangun sebuah *web* tanpa memikirkan struktu dalam penulisan skrip/coding,dsb. Dapat diartikan *framework* merupakan kumpulan dari fungsi/prosedur dan kelas untuk tujuan mempercepat proses pengembangan aplikasi.

1.3.1 Codeigniter



Gambar 1.5 *Codeigniter*

Codeigniter merupakan salah satu *framework* dari *Back-end*. Yang ditemukan pertama kali oleh Rick Ellis pada tanggal 28 Februari 2006, Pada 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 pemograman PHP. Referensi yang begitu banyak menjadi alasanya juga.

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

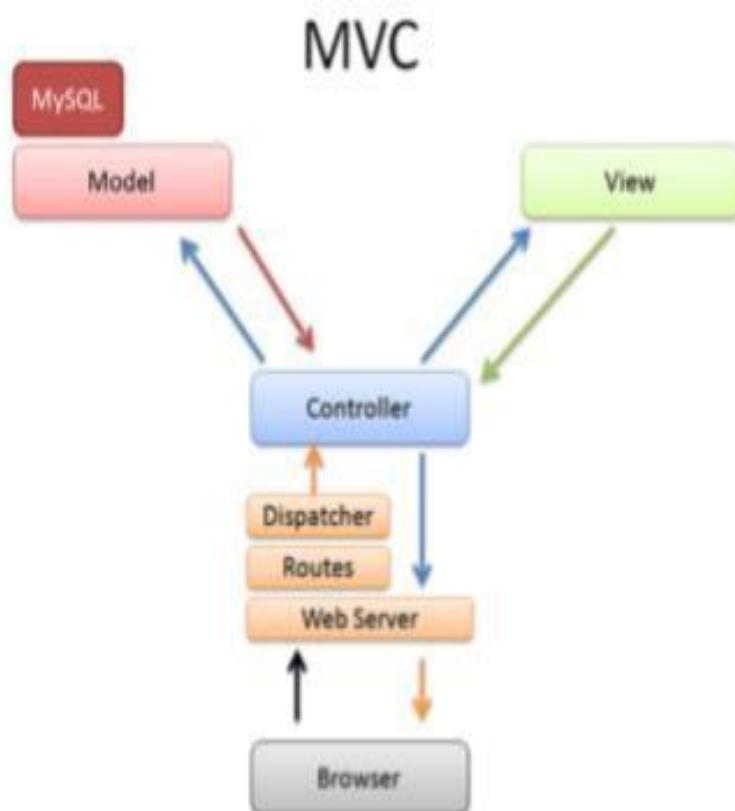
1.3.2 Model MVC

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

1. **Model** merupakan yang berhubungan langsung dengan data atau berinteraksi langsung dengan *Database/webservice*. Model disini merupakan tempat merepresentasikan struktur data. Biasanya dalam model berisi script fungsi seperti tambah data, update data, menampilkan data dan menghapus data. Model secara langsung berhubungan dengan *query-query* SQL yang nantinya akan masuk ke *database*.
2. **View** merupakan bagian yang berhubungan langsung dengan tampilan antarmuka/*user interface* isinya dapat berupa skrip HTML,CSS,Javascript,dsb. Dibagian ini berisi variabel-variabel data yang siap untuk ditampilkan kepada *user*. Di dalam *view* ini hanya di khususkan untuk menampilkan data yang telah di proses oleh controller dan model.
3. **Controller** merupakan bagian yang menjembatani antara model dengan *view*. Bisa dikatakan *controller*

merupakan pusat dari berjalannya sebuah aplikasi. Bagian ini merupakan kelas-kelas dan fungsi-fungsi di proses dari permintaan *view* kedalam struktur data yang ada di dalam model. Controller mengerjakan proses *logix*, 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, 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*.

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\pm).

1.5 Pengenalan MySQL



Gambar 1.8 MySQL

MySQL: Machine Learning bermula di awal abad 20, seorang penemu Spanyol, Torres y Quevedo, membuat sebuah mesin learning setelah ditemukannya komputer digital. Machine Learning pada dasarnya merupakan proses komputer untuk belajar dari data (Learn from data). Sejak pertama kali komputer diciptakan manusia sudah memikirkan bagaimana caranya agar komputer dapat belajar dari pengalaman. Gagasan ini terbukti yaitu pada tahun 1952, Arthur Samuel menciptakan sebuah program, game of checkers, pada sebuah komputer IBM.

Program tersebut dapat mempelajari gerakan untuk memenangkan permainan checkers dan menyimpan gerakan tersebut kedalam memorinya. Istilah teknologi machine learning

pada dasarnya adalah proses komputer untuk belajar dari data (*learn from data*). Jika tidak ada data, maka komputer tidak akan bisa belajar apa-apa. Maka dari itu ketika kita ingin belajar machine learning, pasti akan terus berinteraksi dengan data. Semua pengetahuan machine learning pasti akan melibatkan data. Data bisa saja sama, akan tetapi algoritma dan pendekatannya berbeda-beda untuk mendapatkan hasil yang optimal.

1.5.1 Penjelasan MySQL

Pada awalnya *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 dibaratkan sebagai wadah yang menyimpan data *string (text based)* data. Gambar, media, file, file audio, dll.

1.5.2 Keunggulan MySQL

Kenapa 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 CSS



Gambar 1.8 Cascading Style Sheet

Cascading Style Sheet ataupun CSS pertama kali muncul pada tahun 1970. Kemudian CSS mengalami perkembangan yang sangat pesat, dan format dasar CSS yang sampai saat ini kita gunakan ini merupakan sebuah ide dari programmer bernama Hakon Wium Lie yang berisi dalam bentuk proposalnya tentang Cascading HTML Style Sheet (CHSS) pada bulan Oktober tahun 1994 pada konferensi W3C di Chicago, Illinois. Kemudian, Hakon Wium Lie bersama-sama dengan seorang temannya yang bernama Bert Bos mengembangkan suatu standard CSS.

Pada akhir tahun 1996, CSS telah resmi dipublikasikan. Penggerjaan proyek ini didukung juga oleh seorang programmer dari Perusahaan Microsoft bernama Thomas Reardon. Cascading Style Sheet sangat digunakan oleh penulis maupun pembaca halaman pada sebuah web untuk menentukan warna, jenis huruf, tata letak, dan berbagai aspek tampilan dokumen. CSS sangat digunakan

terutama untuk memisahkan antara isi dokumen yang ditulis dengan HTML atau bahasa markup lainnya dengan presentasi dokumen yang ditulis dengan CSS.

1.6.1 Penjelasan CSS

Cascading Style Sheet atau CSS merupakan salah satu bahasa yang digunakan dalam mendesain sebuah web (style sheet language) dimana CSS mengontrol format tampilan pada sebuah halaman web yang ditulis dengan menggunakan penanda markup laguage. Dulu CSS digunakan untuk mendesain sebuah halaman HTML dan XHTML, namun dengan adanya pengembangan CSS sekarang bisa diaplikasikan/diimplementasikan untuk segala dokumen contohnya XML,SVG,XUL,ANDROID.

CSS dibuat dengan tujuan untuk memisahkan sebuah konten utama dengan tampilan dokumen yang meliputi layout, warna dan font. Dengan melakukan pemisahan tersebut, maka dapat meningkatkan daya pada akses konten sebuah web, CSS menyediakan lebih banyak fleksibilitas dan kontrol dalam spesifikasi dari sebuah karakteristik pada sebuah tampilan, CSS memungkinkan programmer untuk membagi halaman untuk sebuah formatting dan mengurangi kerumitan dalam penulisan kode, struktur dari konten.

Dengan menggunakan Cascading Style Sheet memungkinkan untuk sebuah halaman untuk ditampilkan dalam berbagai style dengan metode pembawaan yang berbeda pula, seperti on-screen, in-print, by voice, dll. Sedangkan pemilik konten web bisa menentukan link yang menghubungkan konten dengan file CSS.

Tujuan utama dari penggunaan Cascading Style Sheet diciptakan untuk membedakan konten dari dokumen dan dari tampilan dokumen, dengan cara itu, pembuat ataupun pengembang web akan lebih mudah dilakukan. Hal ini termasuk dalam desain web diantaranya adalah warna, ukuran dan formatting. Dengan adanya CSS, konten dan desain web akan mudah membedakannya, sehingga akan memudahkan dalam membuat halaman web yang banyak, yang pada akhirnya dapat memangkas waktu pembuatan web.

1.5.2 Keunggulan CSS

Kenapa menggunakan CSS dalam pembuatan aplikasinya? Karena menggunakan CSS akan memudahkan dalam mengatur tampilan pada sebuah web, CSS ini juga memiliki kelebihan diantaranya adalah :

1. Memisahkan antara desain dengan sebuah konten pada halaman web.

2. *Desain dapat diatur dengan seefesien mungkin..*
3. Dapat mengubah suatu tema pada halaman web hanya dengan modifikasi pada bagian Cascading Style Sheet saja.
4. Melangkapi permintaan yang tidak dapat dilakukan oleh HTML.
5. Ukuran File yang sangat kecil.
6. CSS dapat digunakan untuk lebih dari satu halaman.

BAB 2

Persiapan Tools

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

2.1 XAMPP



Gambar 2.1

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

1. **X** = Cross Platform

Merupakan kode penanda untuk software cross platform atau yang bisa berjalan di banyak sistem operasi.

2. **A** = Apache

Apache adalah aplikasi web server yang bersifat gratis dan bisa dikembangkan oleh banyak orang (open source).

3. **M** = MySQL / MariaDB MySQL atau MariaDB merupakan aplikasi database server yang dikembangkan oleh orang yang sama. MySQL berperan dalam mengolah, mengedit, dan menghapus daftar melalui database.

4. **P** = PHP

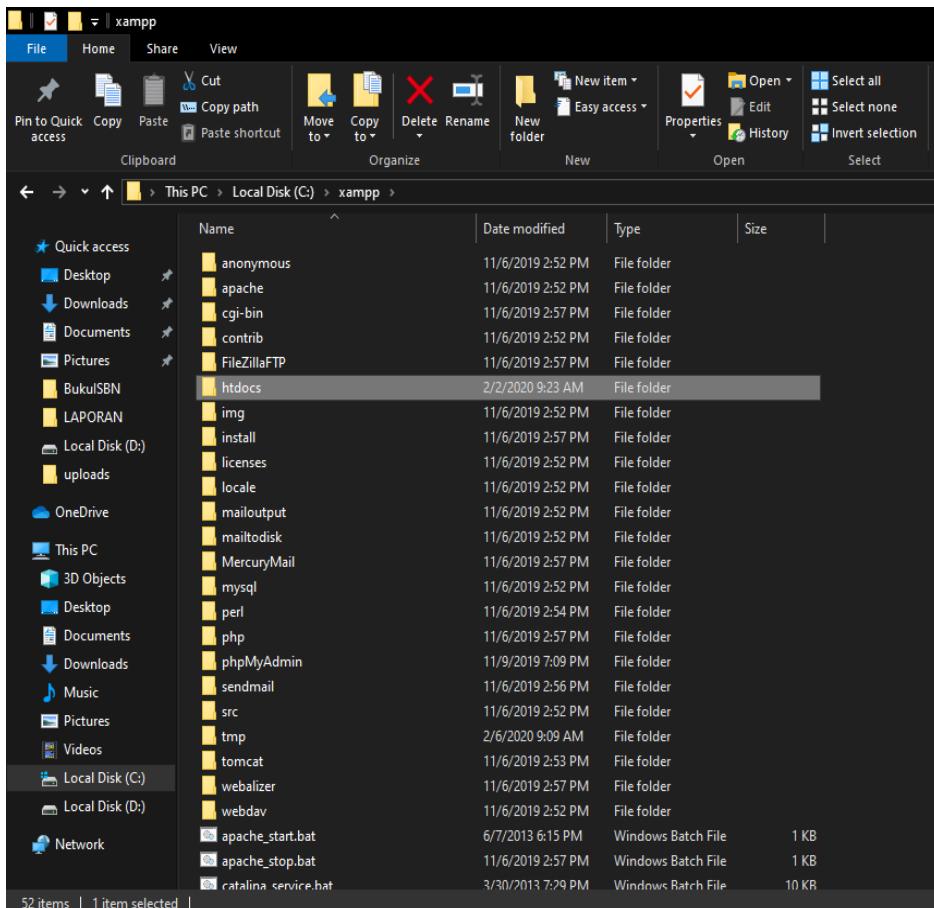
Huruf "P" yang pertama dari akronim kata XAMPP adalah inisial untuk menunjukkan eksistensi bahasa pemrograman PHP. Bahasa pemrograman ini biasanya digunakan untuk membuat website dinamis, contohnya dalam website berbasis CMS WordPress.

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

2.1.1 Bagian Penting pada XAMPP

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

1. Htdocs

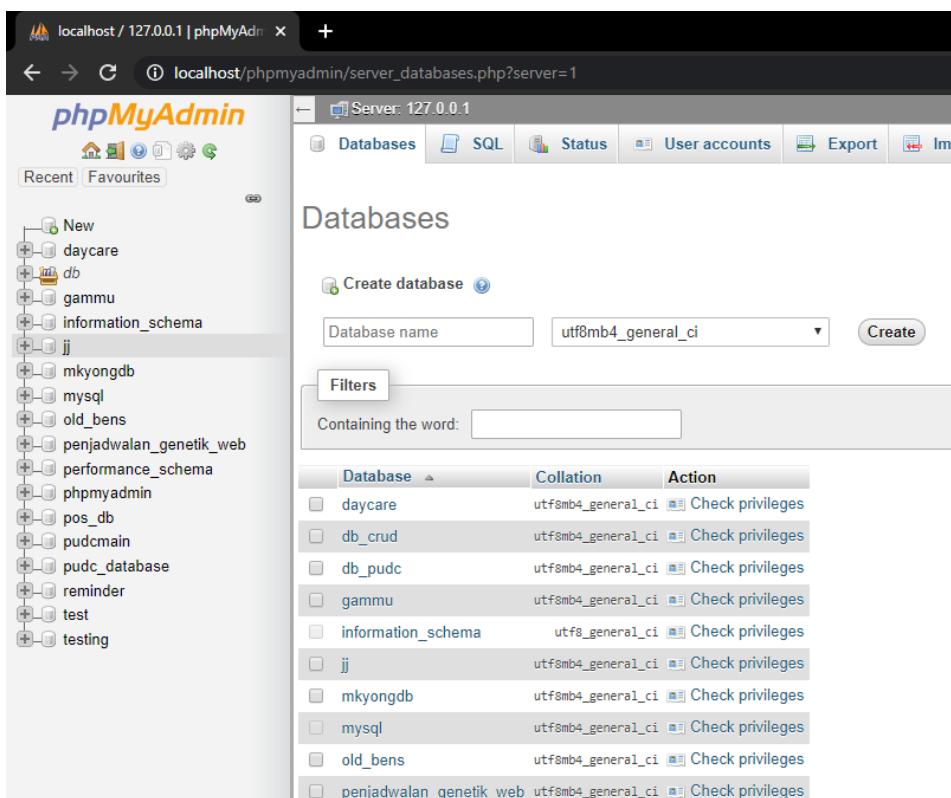


Gambar 2.2 Htdocs

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,



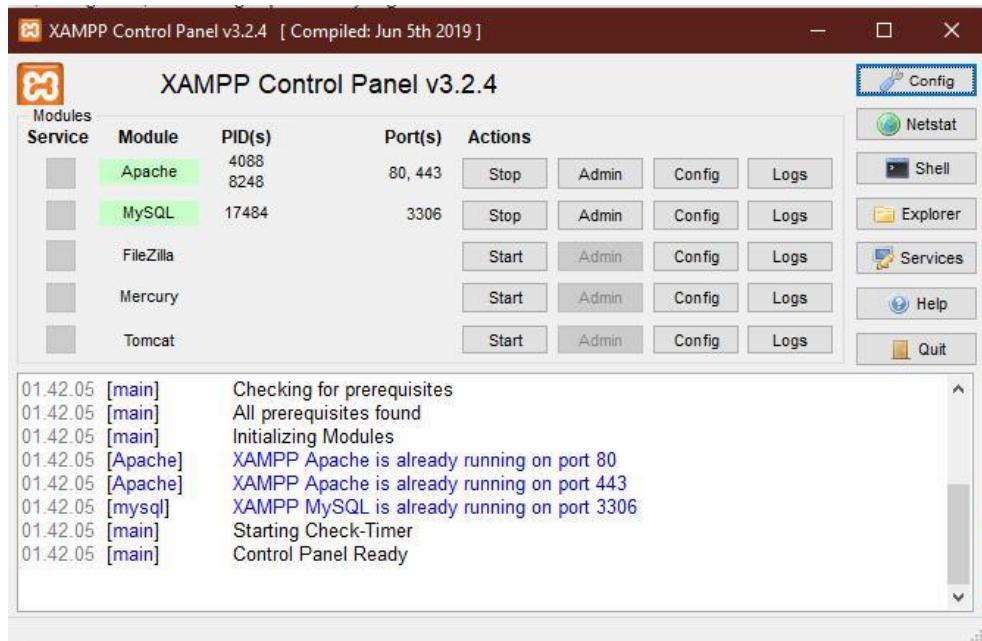
The screenshot shows the phpMyAdmin interface for MySQL management. The left sidebar lists various databases: daycare, db, gammu, information_schema, jj, mkyongdb, mysql, old_bens, penjadwalan_genetik_web, performance_schema, phpmyadmin, pos_db, pudcmain, pudc_database, reminder, test, and testing. The main panel is titled 'Databases' and contains a 'Create database' button and a 'Filters' section with a search input field. Below these are two tables: one for 'Database' and one for 'Collation'. The 'Database' table lists all the databases from the sidebar, each with a checkbox and a 'Check privileges' link. The 'Collation' table lists the same databases with their respective collations and 'Check privileges' links.

Database	Collation	Action
daycare	utf8mb4_general_ci	Check privileges
db_crud	utf8mb4_general_ci	Check privileges
db_pudc	utf8mb4_general_ci	Check privileges
gammu	utf8mb4_general_ci	Check privileges
information_schema	utf8_general_ci	Check privileges
jj	utf8mb4_general_ci	Check privileges
mkyongdb	utf8mb4_general_ci	Check privileges
mysql	utf8mb4_general_ci	Check privileges
old_bens	utf8mb4_general_ci	Check privileges
penjadwalan_genetik_web	utf8mb4_general_ci	Check privileges

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,



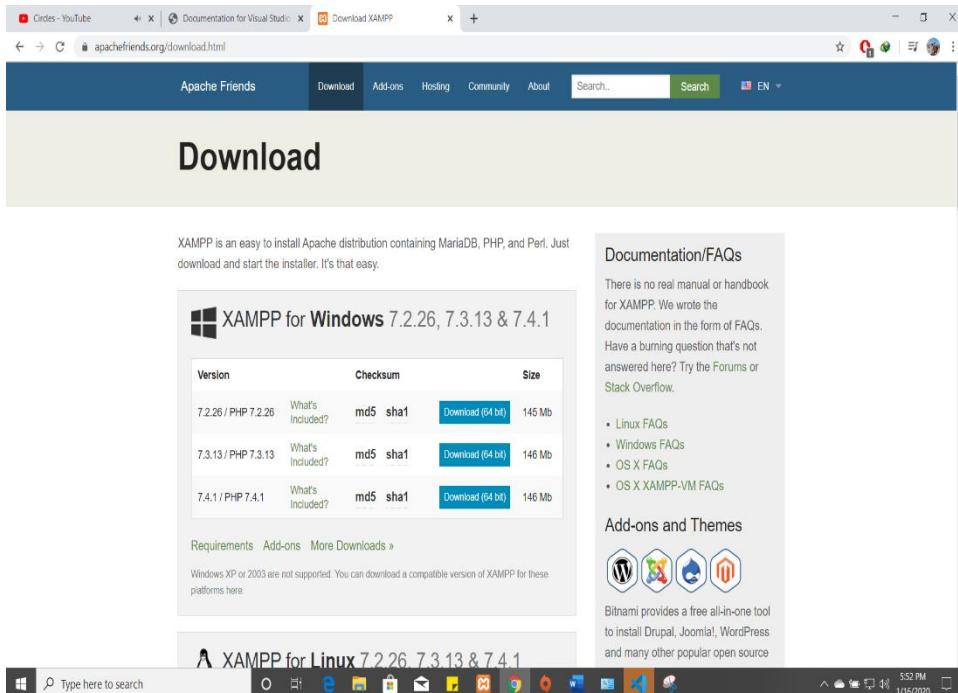
Gambar 2.4 Control Panel

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.

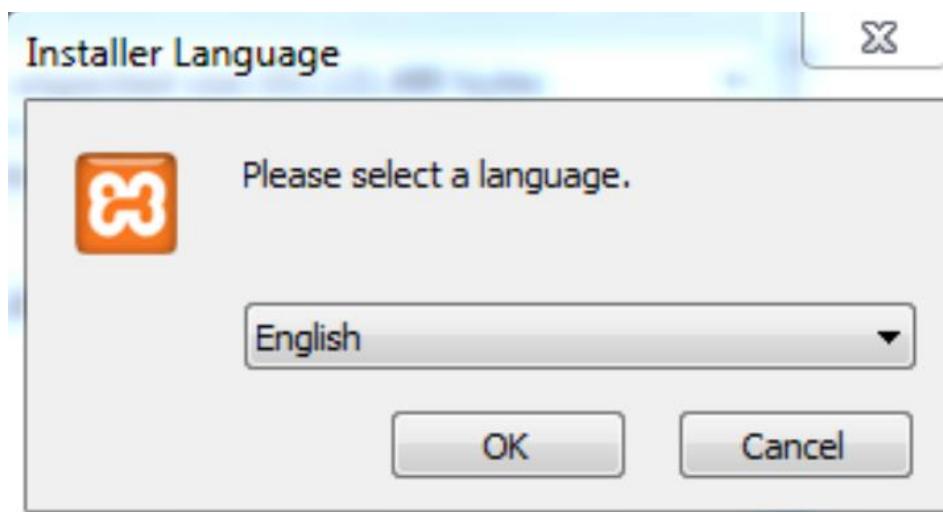


Gambar 2.5 Installasi Jendela Xampp

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.

- Akan tampil jendela baru, untuk memulai proses instalasi, pilih bahasa yang anda inginkan, lalu klik next

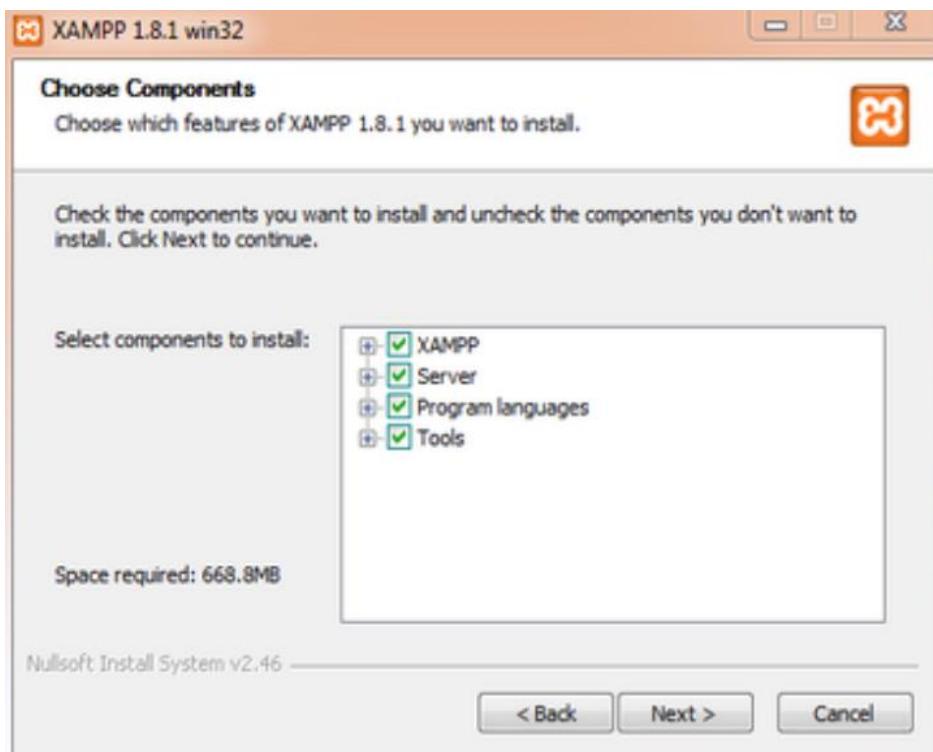


Gambar 2.6 Pilih Bahasa



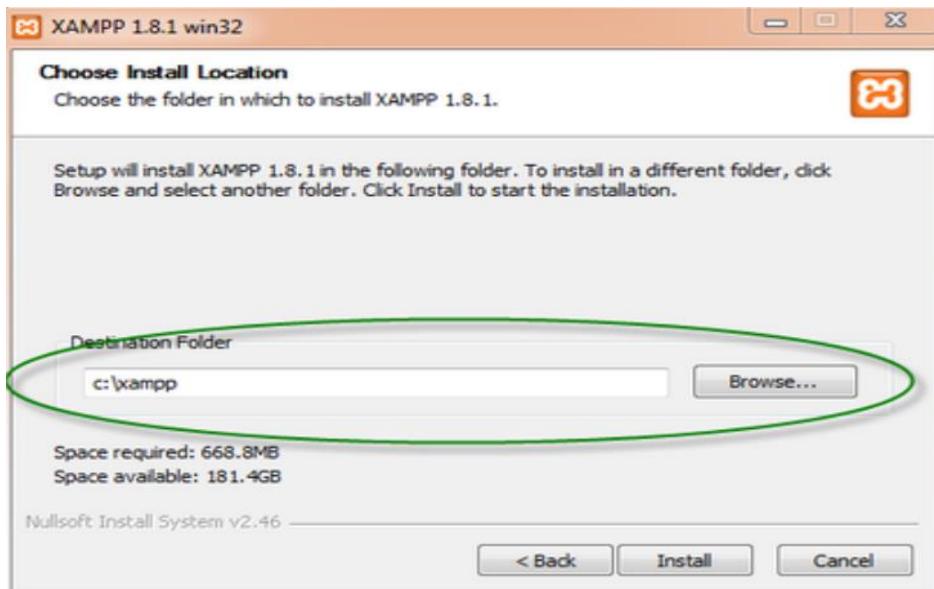
Gambar 2.7 Instalasi Xampp

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.

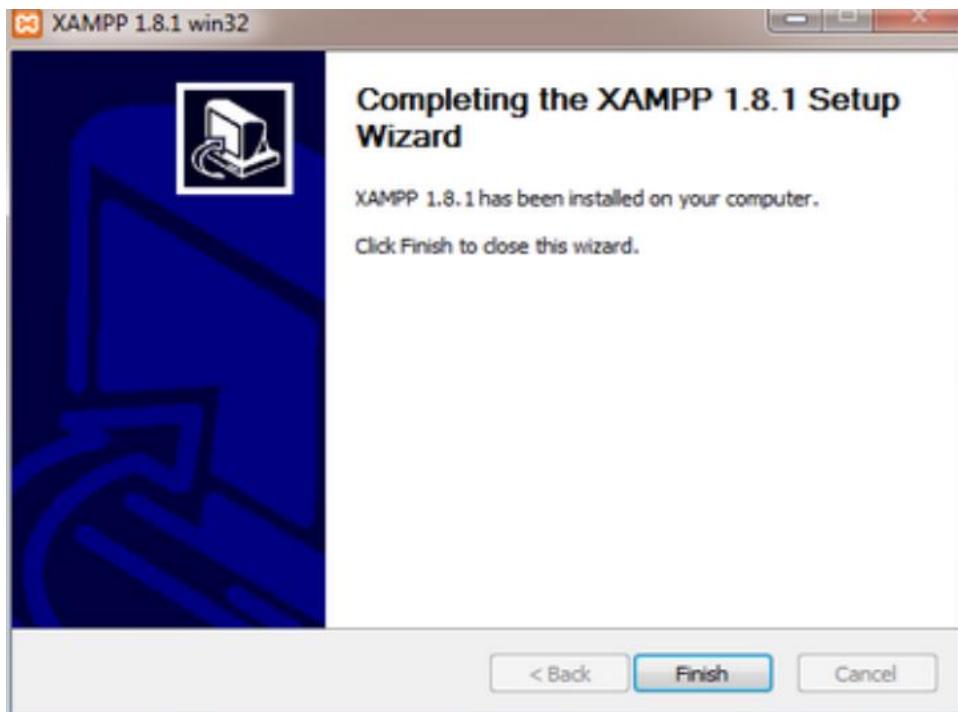


Gambar 2.7 Lokasi Installasi XAMPP

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



Gambar 2.8 Progress Installasi XAMPP



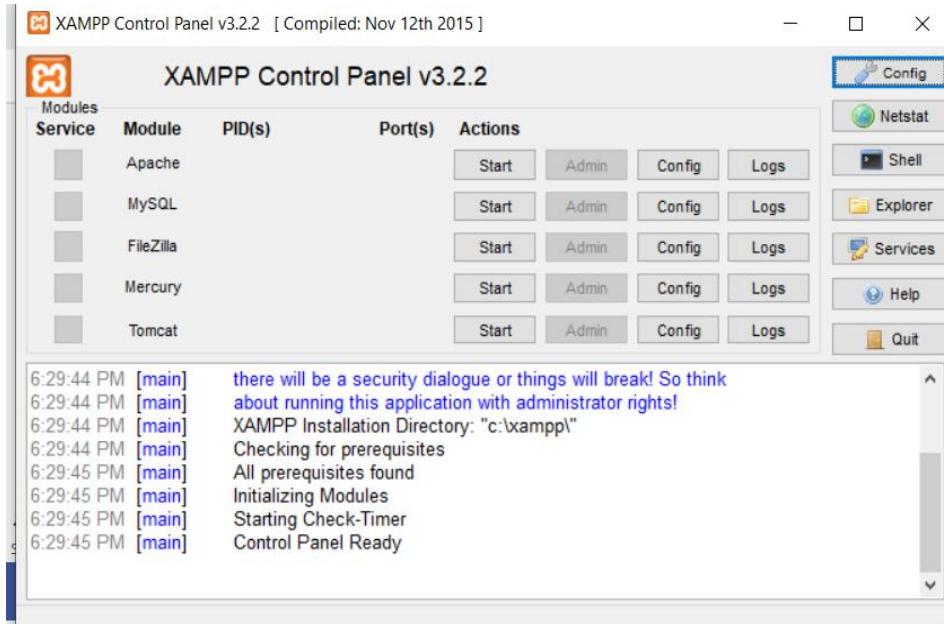
Gambar 2.9 Progress Installasi XAMPP

7. Berikutnya, akan muncul jendela dialog seperti gambar di bawah ini yang menanyakan Anda apakah mau langsung menjalankan aplikasi XAMPP atau tidak. Jika ya, maka klik YES.

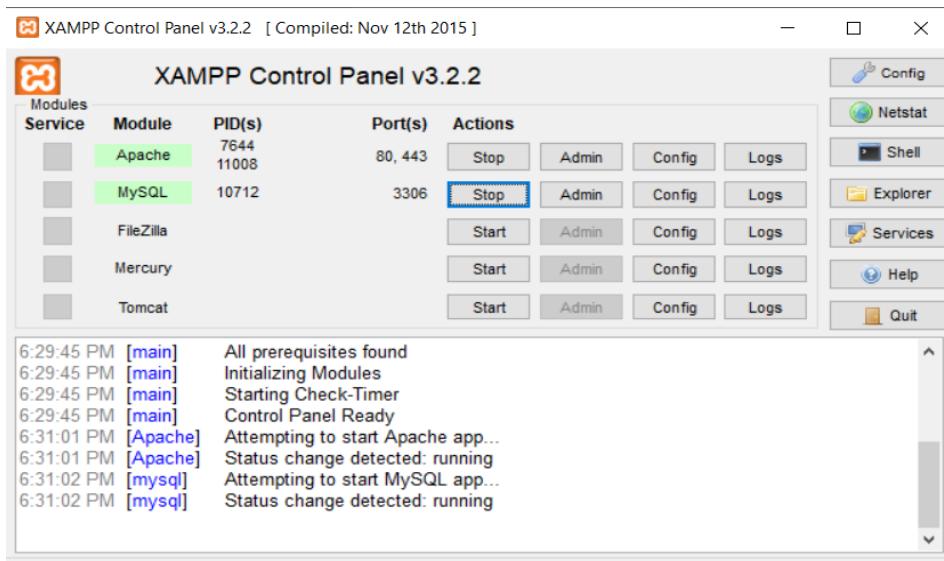


Gambar 2.10 Progress Installasi XAMPP

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



Gambar 2.11 Menjalankan XAMPP



Gambar 2.12 Menjalankan XAMPP

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.

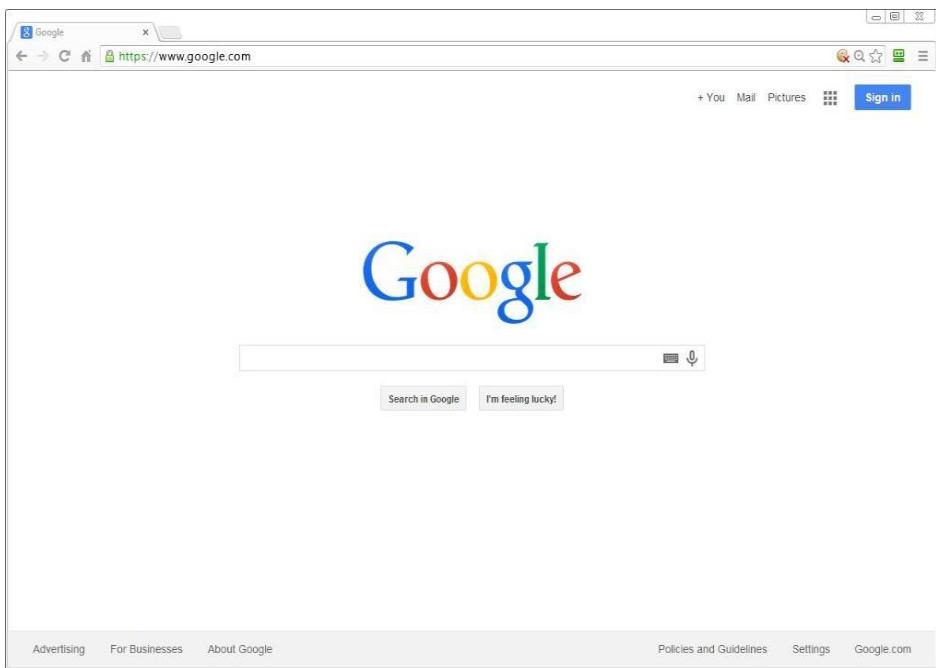
2 . 2 Web Browser

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 2.13 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



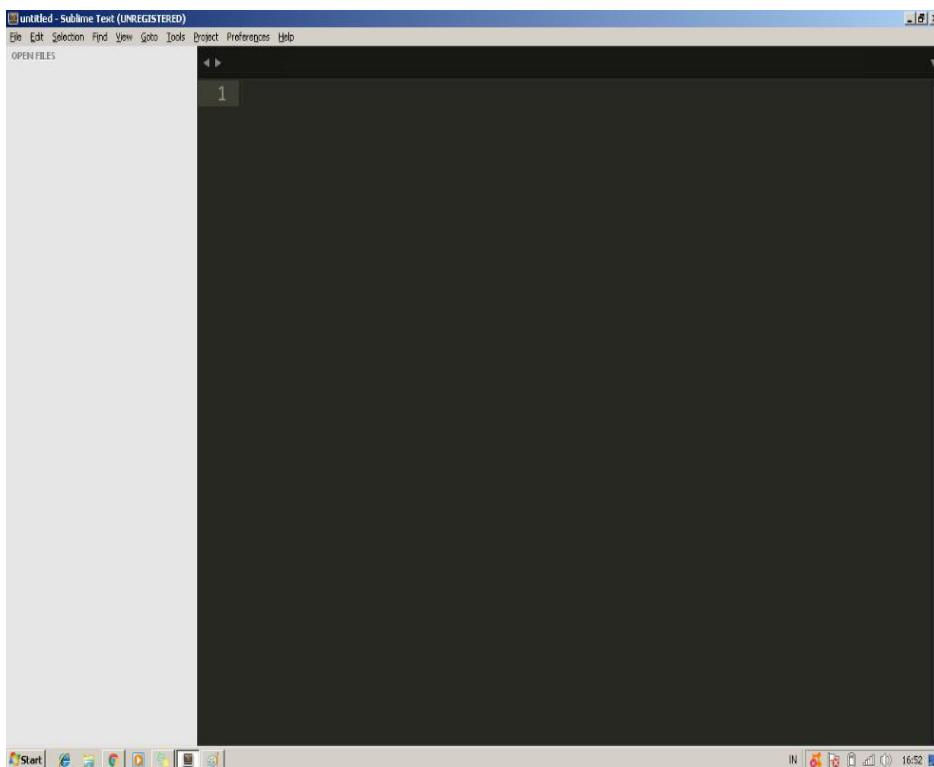
Gambar 2.14 Mozilla Firefox

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/>

2.3 Text Editor

Text Editor merupakan *software tools* penting, *software* atau aplikasi ini memang dikhkususkan untuk para programmer/developer dalam berimajinasi dan 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*.

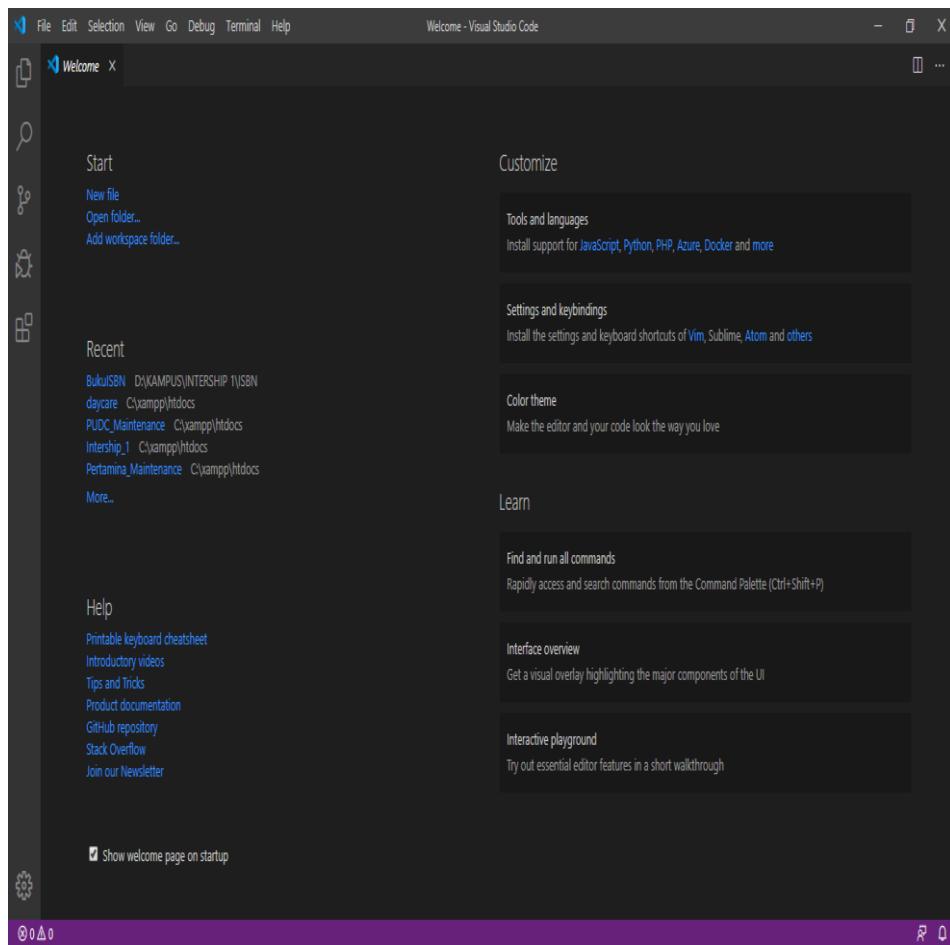
2.3.1 Sublime Text



Gambar 2.15 Sublime Text

Sublime merupakan *text editor* yang populer di kalangan programer, tampilanya yang simpel ini, 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



Gambar 2.16 Visual Studio 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,

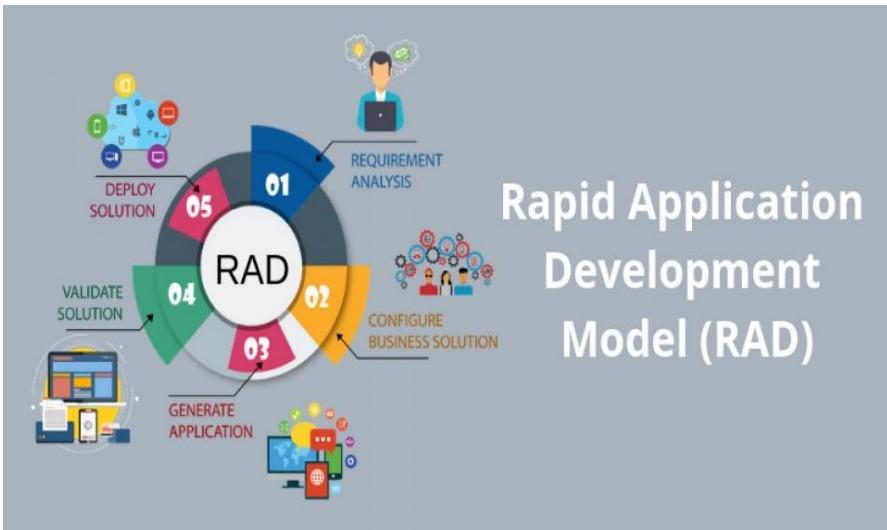
<https://code.visualstudio.com/download>

BAB 3

Metode Pengembangan

Pada pembuatan *web* yang di bangun dengan studi kasus sebuah aplikasi notifikasi otomasi pemeliharaan asset, kita akan membagun aplikasi notifikasi otomasi berbasis *web* berdasarkan jadwal pemeliharaan asset yang masih dilakukan pengecekan jadwal pemeliharaan asset secara manual. Kita akan mengubah metode tersebut dengan pengaplikasian nya menjadi sistem berbasis web, dengan bantuan metode pengembangan *Rapid Application Development* atau biasa dikenal RAD. Metode ini digunakan penulis untuk membangun dan menghasilkan produk/sistem baru yang berupaya melakukan pengembangan dari sistem atau proses bisnis yang sudah ada sebelumnya dengan waktu pembuatan yang singkat.

3.1 Penjelasan RAD



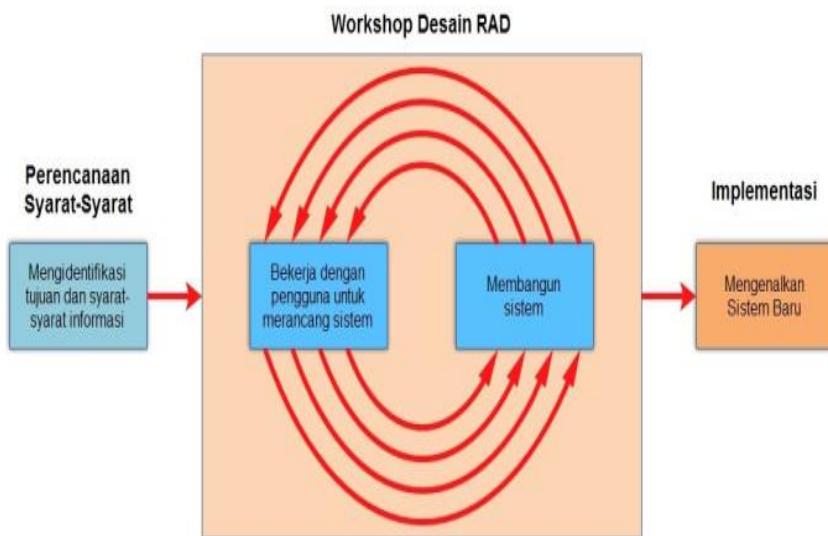
Gambar 3.1 R&D

Rapid Application Development (RAD) merupakan model yang diresmikan oleh James Martin pada tahun 1991 sebagai alternatif dari *waterfall*. Pendekatan *waterfall* bekerja sempurna dalam konstruksi dan industri lainnya di mana perubahan ruang lingkup yang langka dan mahal. Jika Anda mulai membangun jembatan, tidak mungkin bahwa Anda akan swap untuk setengah feri melalui proses.

Pengembangan perangkat lunak jauh lebih fleksibel. Ada lebih variasi bagaimana tantangan bisnis yang sama dapat diselesaikan, dan pada saat yang sama perubahan lebih murah. Akibatnya, desain dan perencanaan super rinci sering kalah dengan pendekatan *trial and error*. Selain itu, pengguna cenderung untuk

memberikan umpan balik yang lebih baik hanya ketika mereka melihat sesuatu kerja.

3.2 Tahapan-tahapan Metode RAD



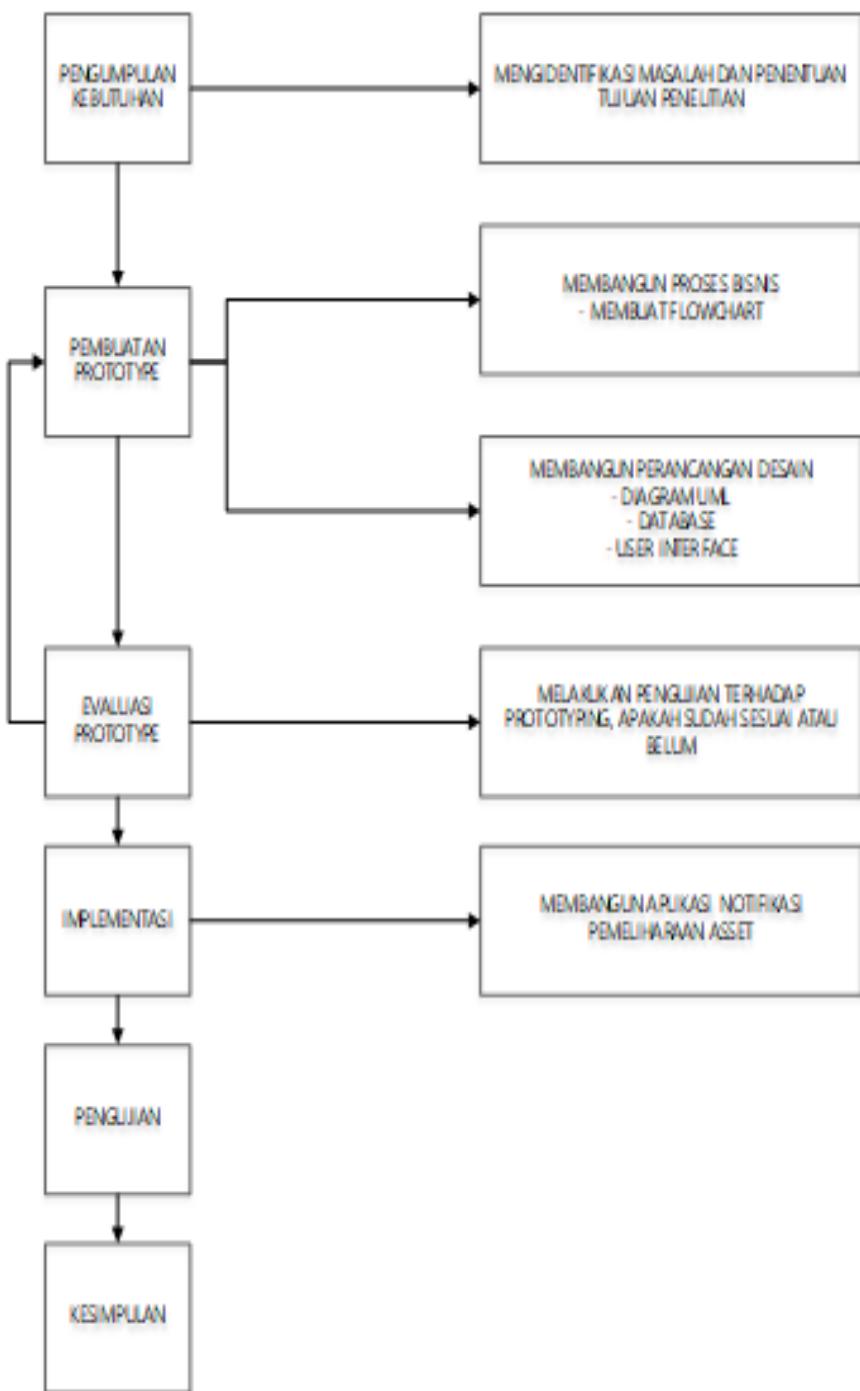
Gambar 3.2 Tahapan-tahapan RAD

Pada umumnya penelitian pembangunan ini terdapat 3 alur tahapan seperti gambar berikut,

1. *Planning* (Perencanaan Syarat-Syarat) Dalam fase ini, pengguna dan penganalisis bertemu untuk mengidentifikasi tujuan-tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Orientasi dalam fase ini adalah menyelesaikan masalah-masalah

perusahaan. Meskipun teknologi informasi dan sistem bisa mengarahkan sebagian dari sistem yang diajukan, fokusnya akan selalu tetap pada upaya pencapaian tujuan-tujuan perusahaan.

2. *RAD Design Workshop* (*Workshop Desain RAD*) Fase ini adalah fase untuk merancang dan memperbaiki yang bisa digambarkan sebagai *workshop*. Penganalisis dan pemrogram dapat bekerja membangun dan menunjukkan representasi visual desain dan pola kerja kepada pengguna. *Workshop* desain ini dapat dilakukan selama beberapa hari tergantung dari ukuran aplikasi yang akan dikembangkan. Selama *workshop* desain RAD, pengguna merespon prototipe yang ada dan penganalisis memperbaiki modul-modul yang dirancang berdasarkan respon pengguna. Apabila sorang pengembangnya merupakan pengembang atau pengguna yang berpengalaman, Kendall menilai bahwa usaha kreatif ini dapat mendorong pengembangan sampai pada tingkat terakselerasi
3. *Implementation* (*Implementasi*) Pada fase implementasi ini, penganalisis bekerja dengan para pengguna secara intens selama *workshop* dan merancang aspek-aspek bisnis dan nonteknis perusahaan. Segera setelah aspek-aspek ini disetujui dan sistem-sistem dibangun dan disaring, sistem-sistem baru atau bagian dari sistem diujicoba dan kemudian diperkenalkan kepada organisasi



Gambar 3.3 Tahapan Ringkas R&D

Berikut merupakan penjelasan tahapan alur diagram metode RAD yang penulis bangun :

1. Tahapan Pengumpulan Kebutuhan

Melibuti kegiatan mengidentifikasi suatu masalah dan penentuan tujuan penelitian.

analisa kebutuhan dan pengumpulan data. Pengumpulan data dilakukan dengan cara meminta data dari tempat magang dan melihat proses bisnis yang ada saat melakukan pemeliharaan asset.

2. Tahapan Pembuatan Prototype

Pada tahapan ini programmer memiliki 2 kegiatan yaitu membangun proses bisnis, proses bisnis yang dimaksud contohnya membuat flowchart ataupun flowmap dan membangun perancangan, contoh perancangan desain ini antara lain merancang diagram uml, merancang database dan merancang user interface.

3. Tahapan Evaluasi Prototype

Merupakan tahap dimana hasil rancangan prototype yang telah dibuat sebelumnya di uji apakah sudah sesuai apa belum, apabila belum sesuai dengan rancangan desain dan rancangan proses bisnis yang sudah dibuat maka, programmer harus melakukan pengecekan prototype kembali pada tahap sebelumnya, jika prototype saat evaluasi sesuai dengan proses bisnis dan rancangan desain maka programmer dapat melangkah pada tahap berikutnya.

4. Tahapan Implementasi

Pada tahap ini programmer mulai membuat website dengan mengimplementasikan sesuai dengan proses bisnis dan rancangan desain yang sudah di evaluasi pada tahap sebelumnya

5. Tahapan Pengujian

Tahapan ini merupakan tahapan pengujian apakah ada bug ataupun error dalam aplikasi tersebut dan dibuatlah catatan agar memperbaiki kesalahan yang ada pada skrip ataupun kesalahan yang ada pada tampilan

6. Tahapan Kesimpulan

Tahap kesimpulan ini merupakan penjelasan dari tahap awal sampai dengan tahap akhir ini, yang merinci semua kegiatan dari pengumpulan data sampai dengan pengujian aplikasi tersebut.

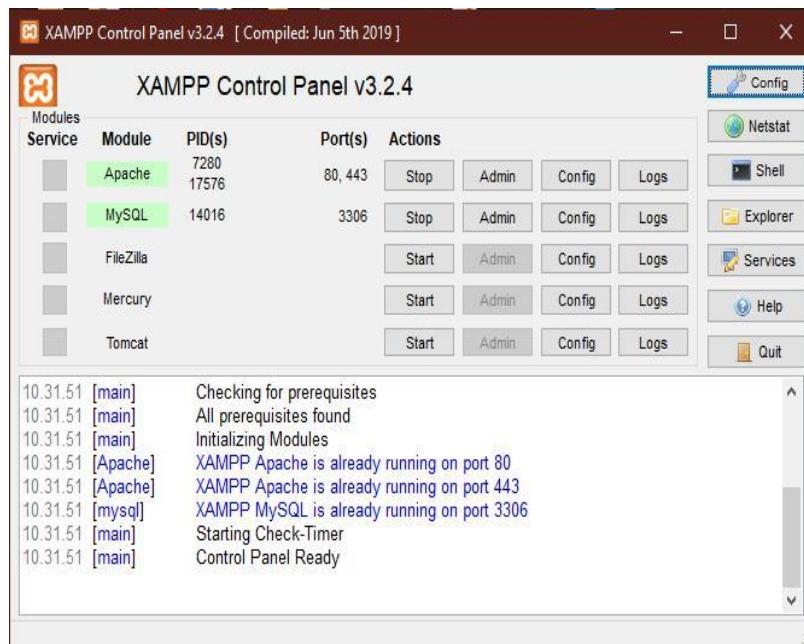
BAB 4

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 4.4 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 sesuai dengan keinginan anda. Disini saya membuatnya dengan nama **pudcmain**.

The screenshot shows the phpMyAdmin interface for a MySQL server at 127.0.0.1. The left sidebar lists various databases: daycare, db, gammu, information_schema, jj, mkyongdb, mysql, old_bens, penjadwalan_genetik_web, performance_schema, phpmyadmin, pos_db, pudc_database, reminder, test, and testing. The main area is titled 'Databases' and contains a 'Create database' form where 'pudcmain' is entered in the database name field, and 'utf8mb4_general_ci' is selected in the collation dropdown. A 'Create' button is visible. Below the form is a 'Filters' section with a search input field. A table lists existing databases with their collations and actions like 'Check privileges':

Database	Collation	Action
daycare	utf8mb4_general_ci	Check privileges
db_crud	utf8mb4_general_ci	Check privileges
db_pudc	utf8mb4_general_ci	Check privileges
gammu	utf8mb4_general_ci	Check privileges
information_schema	utf8_general_ci	Check privileges
jj	utf8mb4_general_ci	Check privileges
mkyongdb	utf8mb4_general_ci	Check privileges
mysql	utf8mb4_general_ci	Check privileges
old_bens	utf8mb4_general_ci	Check privileges
penjadwalan_genetik_web	utf8mb4_general_ci	Check privileges
performance_schema	utf8_general_ci	Check privileges
phpmyadmin	utf8_bin	Check privileges
pos_db	utf8mb4_general_ci	Check privileges
Console		

Gambar 3.5 Pembuatan Database

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

Pembuatan tabel **pm_asset_assignment**,

The screenshot shows the phpMyAdmin interface for creating a new table named 'pm_asset_assignment'. The left sidebar lists various databases and tables. The main area shows the table structure with three columns: ID_ASSIGNMENT, ID_EMPLOYEE, and ID_TYPE. The 'Indexes' section shows three indexes: PRIMARY, FK_ID_EMPLOYEE, and FK_ID_TYPE. A warning message at the bottom states 'No partitioning defined!'. Navigation tabs include 'Table structure' and 'Relation view'.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ID_ASSIGNMENT	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	ID_EMPLOYEE	int(11)			No	None			Change Drop More
3	ID_TYPE	int(11)			No	None			Change Drop More

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Drop	PRIMARY	BTREE	Yes	No	ID_ASSIGNMENT	3	A	No	
Drop	FK_ID_EMPLOYEE	BTREE	No	No	ID_EMPLOYEE	3	A	No	
Drop	FK_ID_TYPE	BTREE	No	No	ID_TYPE	3	A	No	

Gambar 3.6 Pembuatan Tabel pm_asset_assignment

Pembuatan tabel **cabang**,

localhost / 127.0.0.1 / pudmain

localhost/phpmyadmin/tbl_structure.php?db=pudmain&table=pm_email_sender

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_email	int(11)	utf8mb4_general_ci		No	None		AUTO_INCREMENT	Change Drop More
2	email	varchar(50)	utf8mb4_general_ci		No	None		Change Drop More	
3	password	varchar(50)	utf8mb4_general_ci		No	None		Change Drop More	

[Check all](#) [With selected:](#) [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Fulltext](#) [Add to central columns](#) [Remove from central columns](#)

[Print](#) [Propose table structure](#) [Track table](#) [Move columns](#) [Normalise](#)

[Add](#) column(s) [after password](#) [Go](#)

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id_email	0	A	No	

Create an index on columns [Go](#)

Partitions

[No partitioning defined](#)

Gambar 3.7 Pembuatan Tabel pm_email_sender

Pembuatan tabel pm_maintenance_trans,

localhost / 127.0.0.1 / pudmain

localhost/phpmyadmin/tbl_structure.php?db=pudmain&table=pm_maintenance_trans

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ID_M	int(11)	utf8mb4_general_ci		No	None		AUTO_INCREMENT	Change Drop More
2	ID_TYPE	int(12)	utf8mb4_general_ci		No	None		Change Drop More	
3	DATE_H_MIN	varchar(30)	utf8mb4_general_ci		No	None		Change Drop More	
4	DATE_H	varchar(30)	utf8mb4_general_ci		No	None		Change Drop More	
5	DATE_H_PLUS	varchar(30)	utf8mb4_general_ci		No	None		Change Drop More	
6	STATUS	int(11)	utf8mb4_general_ci		No	None		Change Drop More	

[Check all](#) [With selected:](#) [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Fulltext](#) [Add to central columns](#) [Remove from central columns](#)

[Print](#) [Propose table structure](#) [Track table](#) [Move columns](#) [Normalise](#)

[Add](#) column(s) [after STATUS](#) [Go](#)

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	ID_M	86	A	No	
Edit Drop	FK_ID_TYPE_TRANS	BTREE	No	No	ID_TYPE	43	A	No	

Create an index on columns [Go](#)

Gambar 3.8 Pembuatan Tabel pm_maintenance_trans

Pembuatan tabel pm_master_area,

localhost / 127.0.0.1 / pudcmain

localhost/phpmyadmin/tbl_structure.php?db=pudcmain&table=pm_master_area

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	AREA_NAME	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	ID	3	A	No	

Create an index on 1 columns [Go](#)

Partitions

Warning: No partitioning defined!

Gambar 3.9 Pembuatan Tabel pm_master_area

Pembuatan tabel pm_master_asset,

localhost / 127.0.0.1 / pudcmain

localhost/phpmyadmin/tbl_structure.php?db=pudcmain&table=pm_master_asset

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ID_ASSET	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	ASSET_TYPE	int(11)			No	None			Change Drop More
3	ASSET_NAME	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
4	AREA	int(11)			No	None			Change Drop More
5	ACTIVE	enum('1', '2')	utf8mb4_general_ci		No	None			Change Drop More

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	ID_ASSET	42	A	No	
Edit Drop	FK_ID_TYPE_ASSET	BTREE	No	No	ASSET_TYPE	42	A	No	
Edit Drop	FK_ID_AREA	BTREE	No	No	AREA	4	A	No	

Create an index on 1 columns [Go](#)

Gambar 3.10 Pembuatan Tabel pm_master_Asset
Pembuatan tabel pm_master_employee,

The screenshot shows the phpMyAdmin interface for the pudmain database. The left sidebar lists various databases and tables. The main area displays the structure of the 'pm_master_employee' table. The table has 10 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ID	int(11)	utf8mb4_general_ci		No	None	AUTO_INCREMENT		Change Drop More
2	IMAGE	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
3	NAME	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
4	DATE_OF_BIRTH	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
5	PHONE_NUMBER	varchar(15)	utf8mb4_general_ci		No	None			Change Drop More
6	EMAIL	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
7	USERNAME	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
8	PASSWORD	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
9	ACCESS	int(11)			No	None			Change Drop More
10	ACTIVE	enum('Y', 'N')	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, Add to central column, Remove from central columns, Print, Propose table structure, Track table, Move columns, Normalise, and a Go button.

Gambar 3.11 Pembuatan pm_master_employee

Pembuatan tabel pm_message,

The screenshot shows the phpMyAdmin interface for the pudmain database. The left sidebar lists various databases and tables. The main area displays the structure of the 'pm_message' table. The table has 2 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ID	int(11)	utf8mb4_general_ci		No	None	AUTO_INCREMENT		Change Drop More
2	MESSAGE	varchar(1000)	utf8mb4_general_ci		No	None			Change Drop More

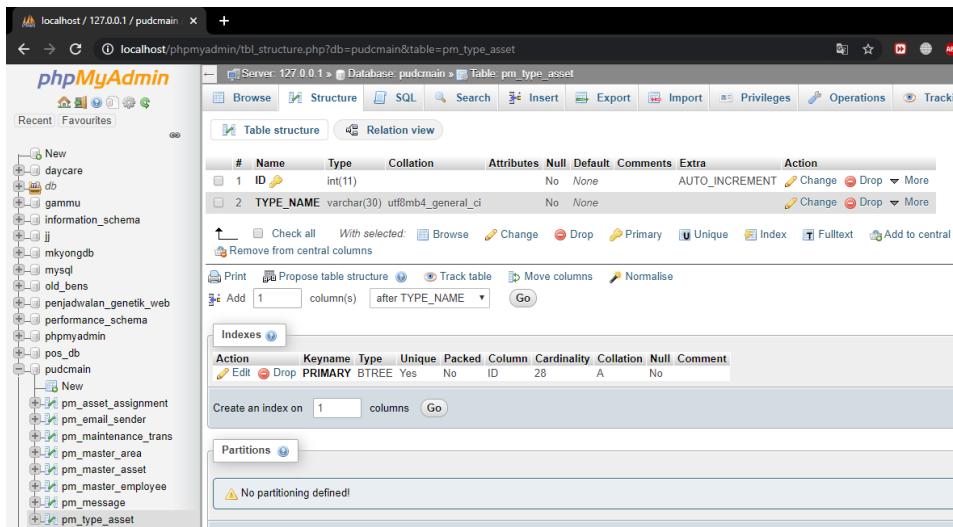
Below the table structure, there are buttons for Check all, With selected, Browse, Change, Drop, Primary, Unique, Index, Fulltext, Add to central column, Remove from central columns, Print, Propose table structure, Track table, Move columns, Normalise, and a Go button.

Under the table structure, there is an 'Indexes' section with a table showing the primary key index:

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

Below the indexes table, there is a 'Create an index on' input field with a Go button. Underneath that is a 'Partitions' section with a message: 'No partitioning defined!'

Gambar 3.12 Pembuatan Tabel pm_message
Pembuatan tabel pm_type_asset,



Gambar 3.13 Pembuatan Tabel pm_type_asset

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

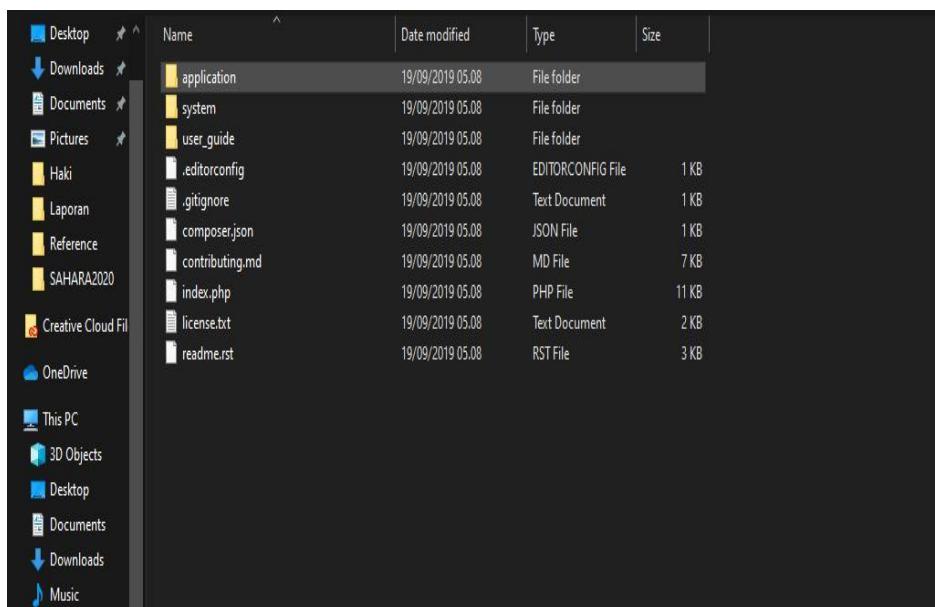
BAB 5

Pembuatan Aplikasi

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

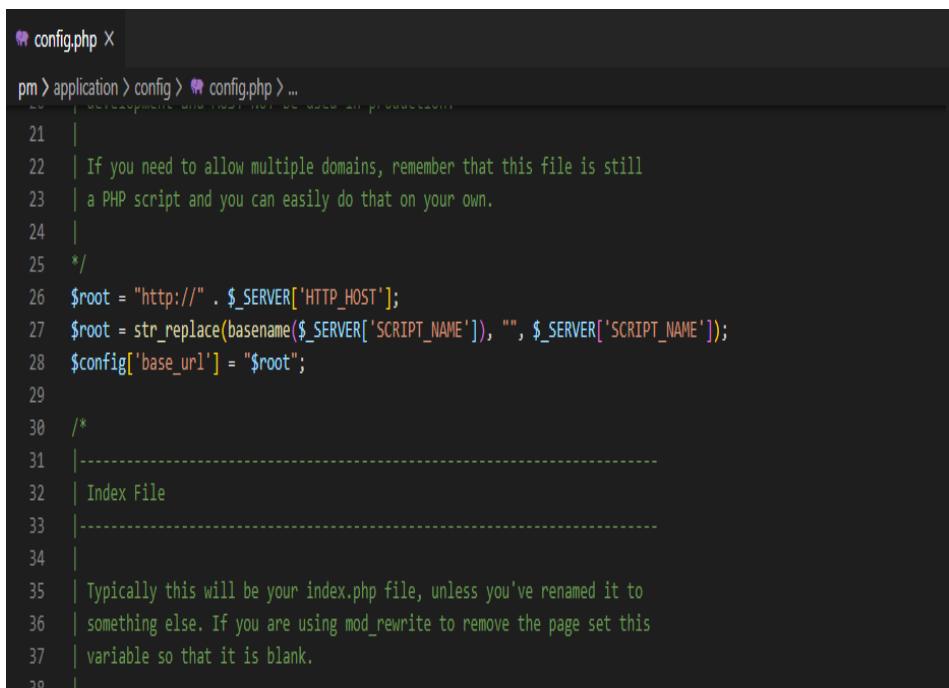
Langkah pertama pastikan anda sudah mendownload *codeigniter* pada *website* resminya pada link berikut, <https://codeigniter.com/download> kita menggunakan versi *codeigniter 3*. Ekstrak *file* tersebut.



Gambar 3.16 Ekstrak *file* Codeigniter

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

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



```
config.php X
pm > application > config > config.php > ...
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 $root = "http://" . $_SERVER['HTTP_HOST'];
27 $root = str_replace(basename($_SERVER['SCRIPT_NAME']), "", $_SERVER['SCRIPT_NAME']);
28 $config['base_url'] = "$root";
29
30 /*
31 | -----
32 | Index File
33 | -----
34 |
35 | Typically this will be your index.php file, unless you've renamed it to
36 | something else. If you are using mod_rewrite to remove the page set this
37 | variable so that it is blank.
38 |
```

Gambar 3.17 Konfigurasi config.php

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

```
autoload.php <
pm > application > config > autoload.php > ...
33  /*
34  | -----
35  | Auto-load Packages
36  | -----
37  | Prototype:
38  |
39  | $autoload['packages'] = array(APPPATH.'third_party', '/usr/local/shared');
40  |
41  */
42 $autoload['packages'] = array();
43
44 /**
45 | -----
46 | Auto-load Libraries
47 | -----
48 | These are the classes located in system/libraries/ or your
49 | application/libraries/ directory, with the addition of the
50 | 'database' library, which is somewhat of a special case.
51 |
52 | Prototype:
53 |
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', 'curl');
62
63 /**
64 | -----
65 | Auto-load Drivers
66 | -----
67
68 | -----
69 | Auto-load Helper Files
70 | -----
71 | Prototype:
72 |
73 | $autoload['helper'] = array('url', 'file');
74 */
75 $autoload['helper'] = array('url');
76
77 /**
78 | -----
79 | Auto-load Config files
80 | -----
81 | Prototype:
82 |
83 | $autoload['config'] = array('config1', 'config2');
84 |
85 | NOTE: This item is intended for use ONLY if you have created custom
86 | config files. Otherwise, leave it blank.
87 |
88 */
89 $autoload['config'] = array();
90
91 /**
92 | -----
93 | Auto-load Language files
94 | -----
95 | Prototype:
```

```

113 |
114 |     $autoload['language'] = array('lang1', 'lang2');
115 |
116 |     NOTE: Do not include the "_lang" part of your file. For example
117 |     "codeigniter_lang.php" would be referenced as array('codeigniter');
118 |
119 */
120 $autoload['language'] = array();
121 /*
122 | -----
123 |     Auto-load Models
124 | -----
125 |
126 | Prototype:
127 |
128 |     $autoload['model'] = array('first_model', 'second_model');
129 |
130 | You can also supply an alternative model name to be assigned
131 | in the controller:
132 |
133 |     $autoload['model'] = array('first_model' => 'first');
134 */
135 $autoload['model'] = array('m_admin');
136

```

Gambar 3.18 Konfigurasi autoload.php

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

```

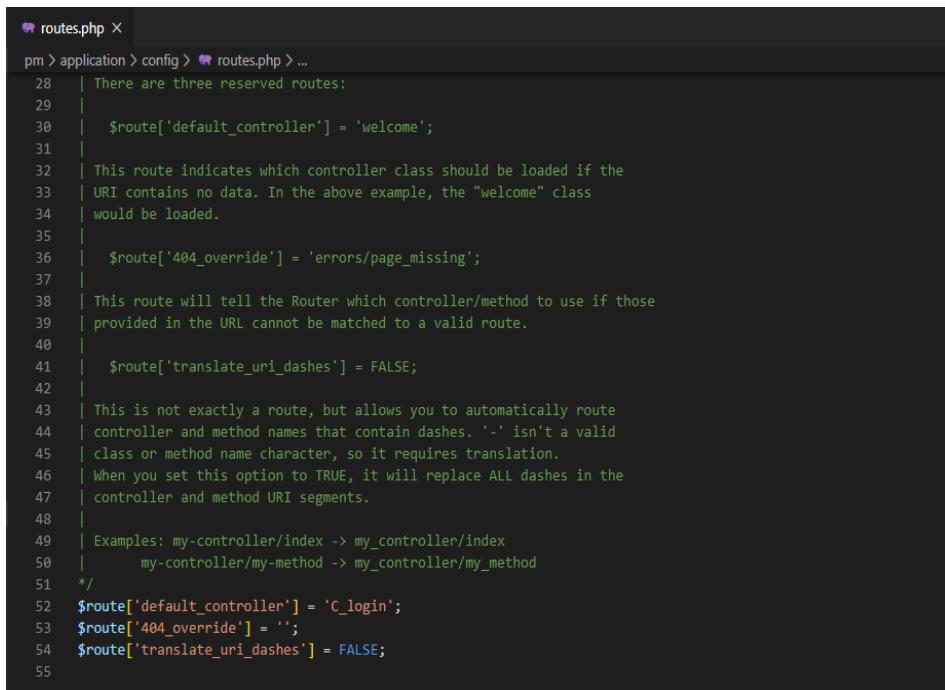
database.php X
pm > application > config > database.php > ...
69 |
70 |     The $query_builder variables lets you determine whether or not to load
71 |     the query builder class.
72 */
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname'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

```

Gambar 3.19 Konfigurasi database.php

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

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



```
routes.php X
pm > application > config > routes.php > ...
28 | There are three reserved routes:
29 |
30 | $route['default_controller'] = 'welcome';
31 |
32 | This route indicates which controller class should be loaded if the
33 | URI contains no data. In the above example, the "welcome" class
34 | would be loaded.
35 |
36 | $route['404_override'] = 'errors/page_missing';
37 |
38 | This route will tell the Router which controller/method to use if those
39 | provided in the URL cannot be matched to a valid route.
40 |
41 | $route['translate_uri_dashes'] = FALSE;
42 |
43 | This is not exactly a route, but allows you to automatically route
44 | controller and method names that contain dashes. '-' isn't a valid
45 | class or method name character, so it requires translation.
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'] = 'C_login';
53 $route['404_override'] = '';
54 $route['translate_uri_dashes'] = FALSE;
55
```

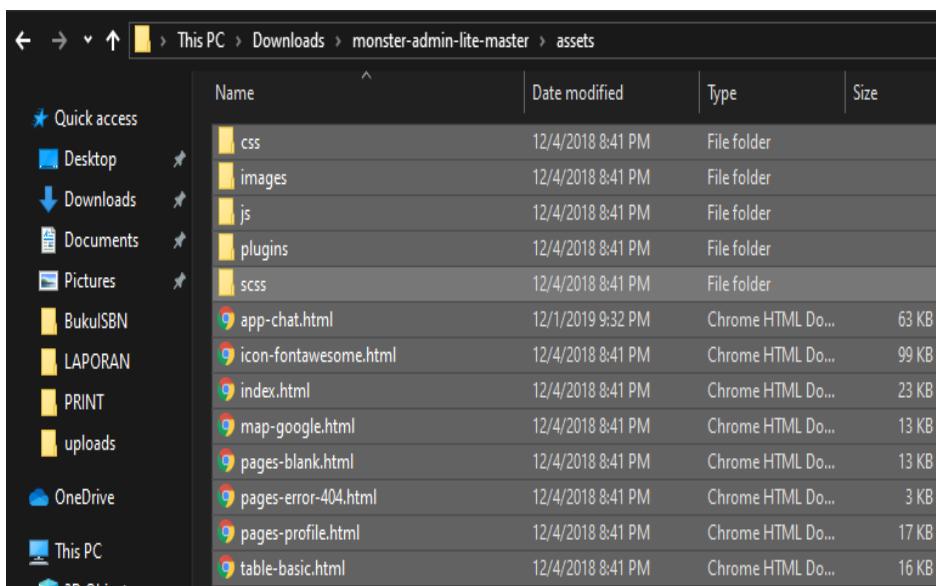
Gambar 3.20 konfigurasi routes.php

Ubah isi **default_controller** dengan controller utama anda, disini penulis memasukan **C_login**, **C_login** ini merupakan nama controller yang nantinya ketika kita membuka *web* kita, controller tersebutlah yang akan *load* pertama. Untuk pembuatan controller **C_login** 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://www.wrappixel.com/templates/monster-admin-lite/>, 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,fonds & vendor** ke folder htdocs kalian dengan membuat folder baru dan beri nama **assets**, contohnya seperti berikut, (C:\xampp\htdocs\pm\assets)



This PC > Downloads > monster-admin-lite-master > assets				
	Name	Date modified	Type	Size
Quick access	css	12/4/2018 8:41 PM	File folder	
Desktop	images	12/4/2018 8:41 PM	File folder	
Downloads	js	12/4/2018 8:41 PM	File folder	
Documents	plugins	12/4/2018 8:41 PM	File folder	
Pictures	scss	12/4/2018 8:41 PM	File folder	
BukulISBN	app-chat.html	12/1/2019 9:32 PM	Chrome HTML Do...	63 KB
LAPORAN	icon-fontawesome.html	12/4/2018 8:41 PM	Chrome HTML Do...	99 KB
PRINT	index.html	12/4/2018 8:41 PM	Chrome HTML Do...	23 KB
uploads	map-google.html	12/4/2018 8:41 PM	Chrome HTML Do...	13 KB
OneDrive	pages-blank.html	12/4/2018 8:41 PM	Chrome HTML Do...	13 KB
This PC	pages-error-404.html	12/4/2018 8:41 PM	Chrome HTML Do...	3 KB
	pages-profile.html	12/4/2018 8:41 PM	Chrome HTML Do...	17 KB
	table-basic.html	12/4/2018 8:41 PM	Chrome HTML Do...	16 KB

	Name	Date modified	Type	Size
Quick access				
Desktop				
Downloads				
Documents				
Pictures				
BukulISBN				
LAPORAN				
PRINT				
uploads				
OneDrive				
This PC				
3D Objects				
Desktop				
Documents				
Downloads				
Music				
Pictures				
Videos				
Local Disk (C:)				
Local Disk (D:)				
Network				
css	12/12/2019 7:29 PM	File folder		
images	12/16/2019 1:16 AM	File folder		
js	12/12/2019 7:29 PM	File folder		
plugins	2/11/2019 9:45 PM	File folder		
scss	12/12/2019 7:29 PM	File folder		
uploads	1/10/2020 10:24 AM	File folder		
app-calendar.html	1/5/2019 4:58 PM	Chrome HTML Do...	61 KB	
app-chat.html	1/5/2019 4:58 PM	Chrome HTML Do...	63 KB	
app-compose.html	1/5/2019 4:58 PM	Chrome HTML Do...	59 KB	
app-contact.html	1/5/2019 4:58 PM	Chrome HTML Do...	75 KB	
app-contact2.html	1/5/2019 4:58 PM	Chrome HTML Do...	68 KB	
app-contact-detail.html	1/5/2019 4:58 PM	Chrome HTML Do...	72 KB	
app-email.html	1/5/2019 4:58 PM	Chrome HTML Do...	80 KB	
app-email-detail.html	1/5/2019 4:58 PM	Chrome HTML Do...	64 KB	
app-ticket.html	1/5/2019 4:58 PM	Chrome HTML Do...	76 KB	
chart-chartist.html	1/5/2019 4:58 PM	Chrome HTML Do...	59 KB	
chart-chart.js.html	1/5/2019 4:58 PM	Chrome HTML Do...	58 KB	
chart-echart.html	1/5/2019 4:58 PM	Chrome HTML Do...	58 KB	
chart-extra-chart.html	1/5/2019 4:58 PM	Chrome HTML Do...	64 KB	
chart-flot.html	1/5/2019 4:58 PM	Chrome HTML Do...	58 KB	
chart-knob.html	1/5/2019 4:58 PM	Chrome HTML Do...	62 KB	
chart-morris.html	1/5/2019 4:58 PM	Chrome HTML Do...	60 KB	
chart-peity.html	1/5/2019 4:58 PM	Chrome HTML Do...	61 KB	
chart-sparkline.html	1/5/2019 4:58 PM	Chrome HTML Do...	63 KB	
cropped.html	2/11/2019 6:39 PM	Chrome HTML Do...	36 KB	

Gambar 3.21 Konfigurasi Bootstrap Monster Admin

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,

	Name	Date modified	Type	Size
Quick access				
Desktop				
Downloads				
Documents				
Pictures				
BukulISBN				
LAPORAN				
PRINT				
uploads				
OneDrive				
This PC				
3D Objects				
Desktop				
Documents				
.bat	12/25/2019 9:51 AM	File folder		
application	12/12/2019 7:23 PM	File folder		
assets	12/20/2019 9:24 AM	File folder		
database	1/27/2020 9:57 PM	File folder		
system	12/12/2019 7:23 PM	File folder		
user_guide	12/12/2019 7:24 PM	File folder		
vendor	12/12/2019 7:25 PM	File folder		
.editorconfig	12/12/2019 7:23 PM	EDITORCONFIG File	1 KB	
.gitignore	12/12/2019 7:23 PM	Text Document	1 KB	
.htaccess	4/18/2017 8:12 PM	HTACCESS File	1 KB	
composer.json	12/12/2019 7:23 PM	JSON File	1 KB	
composer.lock	12/12/2019 7:25 PM	LOCK File	50 KB	
contributing.md	12/12/2019 7:23 PM	MD File	7 KB	
index.php	12/12/2019 7:23 PM	PHP File	11 KB	
license.txt	12/12/2019 7:23 PM	Text Document	2 KB	
readme.rst	12/12/2019 7:23 PM	RST File	3 KB	

Gambar 3.22 *file htaccess*

Berikut isi dari file .htaccess

```
X
RewriteEngine On

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/pm> ketika sudah membuat file .htaccess ini tidak perlu lagi mengetikan index.php jadi langsung saja <http://localhost/pm/>

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 **C_login.php** , pastikan ikuti semua petunjuk karena penulis membuatnya secara berurutan, jika ada yang terlewat dipastikan aplikasi/program akan ada menu yang tidak berkerja.

C_login.php

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

class C_login extends CI_Controller
{

    public function index()
    {
        $data['tes'] = $this->session-
>flashdata('msg');
        $this->load->view('login/index', $data);
    }

    function aksi_login()
    {
        $data = array(
            'username' => $this->input-
>post('username'),
            'password' => $this->input-
>post('password')
        );
        $hasil = $this->m_admin->cek_login($data);
        if ($hasil->num_rows() == 1) {
            foreach ($hasil->result() as $sess) {
                $sess_data['ID'] = $sess->ID;
                $sess_data['IMAGE'] = $sess->IMAGE;
                $sess_data['NAME'] = $sess->NAME;
                $sess_data['DATE_OF_BIRTH'] = $sess-
>DATE_OF_BIRTH;
                $sess_data['PHONE_NUMBER'] = $sess-
>PHONE_NUMBER;
                $sess_data['ACCESS'] = $sess->ACCESS;
                $sess_data['EMAIL'] = $sess->EMAIL;
                $sess_data['ACTIVE'] = $sess->ACTIVE;
                $sess_data['USERNAME'] = $sess-
>USERNAME;
```

```
        $sess_data['PASSWORD'] = $sess-
>PASSWORD;
                // $sess_data['last_login_time'] = tim
e();
                $this->session-
>set_userdata($sess_data);
            }
            if ($this->session-
>userdata('ACTIVE') == 'Y') {
                if ($this->session-
>userdata('ACCESS') == '1') {
                    redirect('C_Admin');
                } elseif ($this->session-
>userdata('ACCESS') == '2') {
                    // redirect('toko');
                    redirect('C_officer');
                }
            } elseif ($this->session-
>userdata('ACTIVE') == 'N') {

                $this->session-
>set_flashdata('msg', 'Account Is Not Active, Please C
ontact Admin');
                redirect('C_login');
                $this->session->sess_destroy();
                // echo "<script>alert('Login Failed :
Account Is Not Active, Please Contact Admin ');histor
y.go(-1);</script>";
            }
        } else {
            $this->session-
>set_flashdata('msg', 'Login Failed: Cek username, pas
sword!');
            redirect('C_login');
        }
    }

    function logout()
{
    $this->session->sess_destroy();
    redirect('C_login');
```

```
    }
}
```

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

```
function aksi_login()
{
    $data = array(
        'username' => $this->input-
>post('username'),
        'password' => $this->input-
>post('password')
    );
    $hasil = $this->m_admin->cek_login($data);
    if ($hasil->num_rows() == 1) {
        foreach ($hasil->result() as $sess) {
            $sess_data[ 'ID' ] = $sess->ID;
            $sess_data[ 'IMAGE' ] = $sess->IMAGE;
            $sess_data[ 'NAME' ] = $sess->NAME;
            $sess_data[ 'DATE_OF_BIRTH' ] = $sess-
>DATE_OF_BIRTH;
            $sess_data[ 'PHONE_NUMBER' ] = $sess-
>PHONE_NUMBER;
            $sess_data[ 'ACCESS' ] = $sess->ACCESS;
            $sess_data[ 'EMAIL' ] = $sess->EMAIL;
            $sess_data[ 'ACTIVE' ] = $sess->ACTIVE;
            $sess_data[ 'USERNAME' ] = $sess-
>USERNAME;
            $sess_data[ 'PASSWORD' ] = $sess-
>PASSWORD;
```

```
// $sess_data['last_login_time'] = time();
$this->session->set_userdata($sess_data);
}
if ($this->session->userdata('ACTIVE') == 'Y') {
    if ($this->session->userdata('ACCESS') == '1') {
        redirect('C_Admin');
    } elseif ($this->session->userdata('ACCESS') == '2') {
        // redirect('toko');
        redirect('C_officer');
    }
} elseif ($this->session->userdata('ACTIVE') == 'N') {

    $this->session->set_flashdata('msg', 'Account Is Not Active, Please Contact Admin');
    redirect('C_login');
    $this->session->sess_destroy();
    // echo "<script>alert('Login Failed : Account Is Not Active, Please Contact Admin ');history.go(-1);</script>";
}
} else {
    $this->session->set_flashdata('msg', 'Login Failed: Cek username, password!');
    redirect('C_login');
}
}
```

Private Function `aksi_login` ini berperan mengecek *username* dan *password* yang dimasukan oleh *user* apakah dia login sebagai admin atau officer, jika *username* dan *password* benar serta sesuai dengan yang ada di *database* dan telah di aktivasi emailnya jika belum di aktivasi akan muncul sebuah notif bahwa email belum di aktivasi dan akan di bawa kembali ke halaman login, jika email belum terdaftar akan muncul sebuah notif bahwa email belum teregistrasi dan akan di bawa kembali ke halaman login, dan jika password yang dimasukan salah maka akan muncul notif bahwa password salah, dan akan dibawa kembali ke halaman login, jika yang di masukan sesuai ia akan langsung masuk kehalaman utama masing-masing dari *user* yang login tersebut. Setelah itu kita akan membuat *view* atau tampilan dari halaman login,

Login/index.php

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

<!-- Mirrored from www.wrappixel.com/demos/admin-templates/monster-admin/main/login.html by HTTrack Website Copier/3.x [XR&amp;CO'2014], Mon, 11 Feb 2019 11:35:42 GMT --&gt;

&lt;head&gt;
    &lt;meta charset="utf-8"&gt;
    &lt;meta http-equiv="X-UA-Compatible" content="IE=edge"&gt;
        &lt;meta name="viewport" content="width=device-width, initial-scale=1"&gt;
        &lt;meta name="description" content=""&gt;</pre>
```

```
<meta name="author" content="">
<link rel="icon" type="image/png/icon" sizes="16x16" href="<?php echo base_url(); ?>assets/images/pertamina.ico">
<title>Preventive Maintenance</title>
<link href="<?php echo base_url(); ?>assets/plugins/bootstrap/css/bootstrap.min.css" rel="stylesheet">
<link href="<?php echo base_url(); ?>assets/css/style.css" rel="stylesheet">
<link href="<?php echo base_url(); ?>assets/css/colors/blue.css" id="theme" rel="stylesheet">
</head>

<body>
<section id="wrapper">
    <div class="login-register" style="background-image:url(<?php echo base_url(); ?>assets/images/background/pudc.jpg);">
        <br><br><br><br><br>
        <div class="login-box card">
            <div class="card-body">
                <form action="<?php echo base_url(); ?>C_login/aksi_login" method="post">
                    <center>
                        <h3 class="box-title mb-3">Preventive Maintenance</h3>
                        <?php echo $tes; ?>
                    </center>
                    <div class="form-group">
                        <div class="col-xs-12">
                            <input class="form-control" type="text" required="" placeholder="Username" name="username"> </div>
                        </div>
                    <div class="form-group">
                        <div class="col-xs-12">
                            <input class="form-control" type="password" required="" placeholder="Password" name="password"> </div>
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>
</section>

```

```
</div>

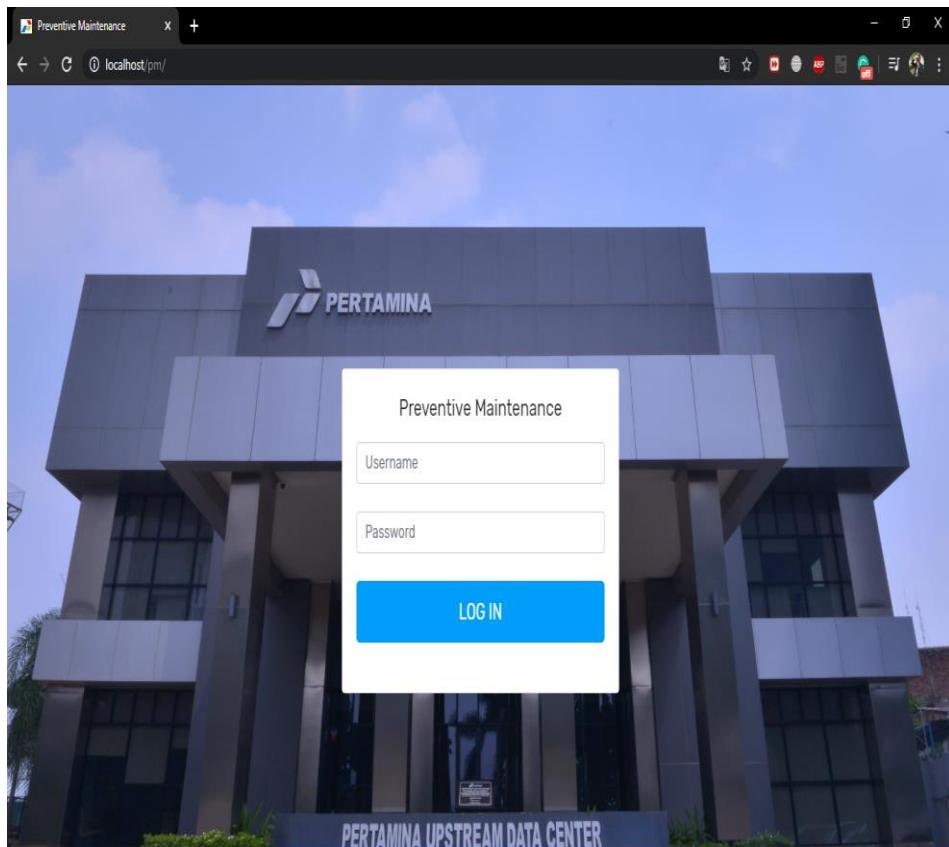
        <div class="form-group text-
center mt-3">
            <div class="col-xs-12">
                <button class="btn btn
-info btn-lg btn-block text-uppercase waves-
effect waves-light" type="submit">Log In</button>
            </div>
        </div>
    </div>

</section>
<script src=<?php echo base_url(); ?>assets/plugi
ns/jquery/jquery.min.js"></script>
<script src=<?php echo base_url(); ?>assets/plugi
ns/bootstrap/js/popper.min.js"></script>
<script src=<?php echo base_url(); ?>assets/plugi
ns/bootstrap/js/bootstrap.min.js"></script>
<script src=<?php echo base_url(); ?>assets/js/jq
uery.slimscroll.js"></script>
<script src=<?php echo base_url(); ?>assets/js/wa
ves.js"></script>
<script src=<?php echo base_url(); ?>assets/js/si
debarmenu.js"></script>
<script src=<?php echo base_url(); ?>assets/plugi
ns/sticky-kit-master/dist/sticky-kit.min.js"></script>
<script src=<?php echo base_url(); ?>assets/js/cu
stom.min.js"></script>
<script src=<?php echo base_url(); ?>assets/plugi
ns/styleswitcher/jQuery.style.switcher.js"></script>
</body>

</html>
```

Pada intinya codingan diatas membuat tampilan layout baru yang berisi *label* dan *textbox username password*, lalu *button login* untuk melakukan proses login.

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/pm/> maka secara otomatis tampilan akan otomatis muncul seperti berikut,



Gambar 3.23 Halaman Login

5 . 4 Pembuatan Halaman Admin

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

C_admin.php

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

class C_admin extends CI_Controller
{

    var $API = "";

    public function __construct()
    {
        parent::__construct();
        // date_default_timezone_set('Asia/Jakarta');
        $this->
>API = "http://localhost/api_sms/index.php";
        if (!$this->session->userdata('USERNAME')) {
            redirect('C_login');
        }
    }

    public function index()
    {
        $this->load->view('resource/admin/header');
        $this->load->view('resource/admin/sidebar');
        $this->load->view('admin/home/index');
        $this->load->view('resource/admin/footer');
        $this->load->view('style/style');
```

```
}

// ====== EM
PLOYEE MANAGEMENT ======
=====

    public function employeeManagement()
    {
        $data['employee'] = $this->m_admin-
>getAllEmployee();
        $data['tes'] =  $this->session-
>flashdata('msg');

        $this->load->view('resource/admin/header');
        $this->load->view('resource/admin/sidebar');
        $this->load-
>view('admin/employee/employeeView', $data);
        $this->load->view('resource/admin/footer');
        $this->load->view('style/style');
    }

    public function employeeForm()
    {
        $this->load->view('resource/admin/header');
        $this->load->view('resource/admin/sidebar');
        $this->load-
>view('admin/employee/employeeForm');
        $this->load->view('resource/admin/footer');
        $this->load->view('style/style');
    }

    public function addEmployee()
    {
        $data = array(
            'NAME' => $this->input->post('NAME'),
            'DATE_OF_BIRTH' => $this->input-
>post('DATE_OF_BIRTH'),
            'EMAIL' => $this->input->post('EMAIL'),
```

```

        'PHONE_NUMBER' => $this->input-
>post('PHONE_NUMBER'),
        'USERNAME' => $this->input-
>post('USERNAME'),
        'PASSWORD' => $this->input-
>post('PASSWORD'),
        'ACCESS' => $this->input->post('ACCESS'),
        'ACTIVE' => $this->input->post('ACTIVE'),
        'IMAGE' => $this->input->post('IMAGE'),
    );
    if (!empty($_FILES['IMAGE']['name'])) {
        $upload = $this->_do_upload();
        $data['IMAGE'] = $upload;
    }

    // print_r($data);
    // die;
    $this->m_admin-
>InsertData('pm_master_employee', $data);
    redirect("C_admin/employeeManagement");
}

private function _do_upload()
{
    $config['upload_path']      = 'assets/uploads/';
;
    $config['allowed_types']   = 'gif|jpg|png|jpeg
';
    $config['max_size']         = 10000;
    $config['max_width']        = 10000;
    $config['max_height']       = 10000;
    $config['file_name']        = round(microti
me(true) * 1000);

    $this->load->library('upload', $config);
    if (!$this->upload->do_upload('IMAGE')) {
        $this->session->set_flashdata('msg', $this-
>upload->display_errors(' ', ''));
        redirect('C_admin/employeeManagement');
    }
}

```

```
        }

        return $this->upload->data('file_name');
    }

    public function employeeEdit($ID)
    {
        $where = array('ID' => $ID);
        $data['employee'] = $this->m_admin-
>EditAllData($where, 'pm_master_employee');

        $this->load->view('resource/admin/header');
        $this->load->view('resource//admin/sidebar');
        $this->load-
>view('admin/employee/employeeEdit', $data);
        $this->load->view('resource/admin/footer');
        $this->load->view('style/style');
    }

    public function employeeEditPhotos($ID)
    {
        $where = array('ID' => $ID);
        $data['employee'] = $this->m_admin-
>EditAllData($where, 'pm_master_employee');

        $this->load->view('resource/admin/header');
        $this->load->view('resource//admin/sidebar');
        $this->load-
>view('admin/employee/employeeEditPhotos', $data);
        $this->load->view('resource/admin/footer');
        $this->load->view('style/style');
    }

    public function employeeUpdate()
    {

        $data = array(
            #database
            'NAME' => $this->input->post('NAME'),
    }
```

```

        'DATE_OF_BIRTH' => $this->input-
>post('DATE_OF_BIRTH'),
        'EMAIL' => $this->input->post('EMAIL'),
        'PHONE_NUMBER' => $this->input-
>post('PHONE_NUMBER'),
        'USERNAME' => $this->input-
>post('USERNAME'),
        'PASSWORD' => $this->input-
>post('PASSWORD'),
        'ACCESS' => $this->input->post('ACCESS'),
        'ACTIVE' => $this->input->post('ACTIVE'),
        // 'IMAGE' => $this->input->post('IMAGE'),
    );
    // if (!empty($_FILES['IMAGE']['name'])) {
    //     $upload = $this->_do_upload();
    //     $data['IMAGE'] = $upload;
    // }
    $where = array(
        'ID' => $this->input->post('ID')
    );
    // print_r($data);die;
    $this->m_admin-
>UpdateAllData($where, $data, 'pm_master_employee');
    redirect('C_admin/employeeManagement');
}
public function employeePhotosUpdate()
{
    $data = array(
        #database
        'IMAGE' => $this->input->post('IMAGE'),
    );
    if (!empty($_FILES['IMAGE']['name'])) {
        $upload = $this->_do_upload();
        $data['IMAGE'] = $upload;
    }
    $where = array(
        'ID' => $this->input->post('ID')
    );
}

```

```
// print_r($data);die;
$this->m_admin-
>UpdateAllData($where, $data, 'pm_master_employee');
    redirect('C_admin/employeeManagement');
}

public function employeeDelete($ID)
{
    $where = array('ID' => $ID);
    $this->m_admin-
>DeleteAllData($where, 'pm_master_employee');
    redirect('C_admin/employeeManagement');
}

// ===== AS
SET MANAGEMENT =====
=====
// =====
== DATA AREA =====
=====
public function area()
{
    $data = array(
        'area' => $this->m_admin-
>getAllData('pm_master_area')
    );

    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
    $this->load-
>view('admin/assetManagement/area', $data);
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
}

public function addArea()
{
    $data = array(
```

```
        'AREA_NAME' => $this->input-
>post('AREA_NAME'),
    );
    $this->m_admin-
>InsertData('pm_master_area', $data);
    redirect("C_admin/area");
}

public function areaForm()
{
    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
    $this->load-
>view('admin/assetManagement/areaForm');
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
}

public function areaEdit($ID)
{
    $where = array('ID' => $ID);
    $data['area'] = $this->m_admin-
>EditAllData($where, 'pm_master_area');

    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
    $this->load-
>view('admin/assetManagement/areaEdit', $data);
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
}

public function areaUpdate()
{

    $data = array(
        #database
        'AREA_NAME' => $this->input-
>post('AREA_NAME'),
```

```

    );
    $where = array(
        'ID' => $this->input->post('ID')
    );
    // print_r($data);die;
    $this->m_admin-
>UpdateAllData($where, $data, 'pm_master_area');
    redirect('C_admin/area');
}

public function areaDelete($ID)
{
    $where = array('ID' => $ID);
    $this->m_admin-
>DeleteAllData($where, 'pm_master_area');
    redirect('C_admin/area');
}
// =====
== END AREA =====
==

public function datatype()
{
    $keyword     = $this->input->post('keyword');
    $tb = 'pm_type_asset';
    $data = array(
        'type' => $this->m_admin-
>getDataType($tb, $keyword)
    );
    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
    $this->load-
>view('admin/assetManagement/dataType', $data);
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
}
public function dataTypeForm()
{

```

```
        $this->load->view('resource/admin/header');
        $this->load->view('resource/admin/sidebar');
        $this->load-
>view('admin/assetManagement/dataTypeForm');
        $this->load->view('resource/admin/footer');
        $this->load->view('style/style');
    }

    public function adddataType()
    {
        $data = array(
            'TYPE_NAME' => $this->input-
>post('TYPE_NAME'),
        );
        $this->m_admin-
>InsertData('pm_type_asset', $data);
        redirect("C_admin/dataType");
    }

    public function dataTypeEdit($ID)
    {
        $where = array('ID' => $ID);
        $data['type'] = $this->m_admin-
>EditAllData($where, 'pm_type_asset');

        $this->load->view('resource/admin/header');
        $this->load->view('resource/admin/sidebar');
        $this->load-
>view('admin/assetManagement/dataTypeEdit', $data);
        $this->load->view('resource/admin/footer');
        $this->load->view('style/style');
    }

    public function dataTypeUpdate()
    {

        $data = array(
            #database
```

```

        'TYPE_NAME' => $this->input-
>post('TYPE_NAME'),
    );
    $where = array(
        'ID' => $this->input->post('ID')
    );
    // print_r($data);die;
    $this->m_admin-
>UpdateAllData($where, $data, 'pm_type_asset');
    redirect('C_admin/dataType');
}

public function dataTypeDelete($ID)
{
    $where = array('ID' => $ID);
    $this->m_admin-
>DeleteAllData($where, 'pm_type_asset');
    redirect('C_admin/dataType');
}

// ===== END DATA TYPE =====
=
// ===== ASSET =====
SET =
=
public function assetdata()
{
    $data = array(
        'asset' => $this->m_admin->getAsset()
    );

    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
    $this->load-
>view('admin/assetManagement/assetData', $data);
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
}

```

```
}

public function assetForm()
{
    $data = array(
        'type' => $this->m_admin-
>getAllData('pm_type_asset'),
        'area' => $this->m_admin-
>getAllData('pm_master_area')

    );
    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
    $this->load-
>view('admin/assetManagement/assetForm', $data);
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
}

public function adddataAsset()
{
    $data = array(
        'ASSET_NAME' => $this->input-
>post('ASSET_NAME'),
        'ASSET_TYPE' => $this->input-
>post('ASSET_TYPE'),
        'AREA' => $this->input->post('AREA'),
        'ACTIVE' => $this->input->post('ACTIVE'),
    );
    // print_r($data);
    // die;
    $this->m_admin-
>InsertData('pm_master_asset', $data);
    redirect("C_admin/assetdata");
}
public function assetEdit($ID)
{
    $where = array('ID_ASSET' => $ID);
```

```
        $data['asset'] = $this->m_admin-
>EditAllData($where, 'pm_master_asset');
        $data['type'] = $this->m_admin-
>getAllData('pm_type_asset');
        $data['area'] = $this->m_admin-
>getAllData('pm_master_area');

        $this->load->view('resource/admin/header');
        $this->load->view('resource/admin/sidebar');
        $this->load-
>view('admin/assetManagement/assetEdit', $data);
        $this->load->view('resource/admin/footer');
        $this->load->view('style/style');
    }

    public function assetUpdate()
    {

        $data = array(
            #database
            'ASSET_NAME' => $this->input-
>post('ASSET_NAME'),
            'ASSET_TYPE' => $this->input-
>post('ASSET_TYPE'),
            'AREA' => $this->input->post('AREA'),
            'ACTIVE' => $this->input->post('ACTIVE'),
        );
        $where = array(
            'ID_ASSET' => $this->input-
>post('ID_ASSET')
        );
        // print_r($data);
        // die;
        $this->m_admin-
>UpdateAllData($where, $data, 'pm_master_asset');
        redirect('C_admin/assetdata');
    }

    public function assetDelete($ID)
```

```

{
    $where = array('ID_ASSET' => $ID);
    $this->m_admin-
>DeleteAllData($where, 'pm_master_asset');
    redirect('C_admin/assetdata');
}

// ===== END DATA TYPE =====
=====
// ===== ASSET ASSIGNMENT =====
=====

public function assetAssignment()
{
    $data = array(
        'assignment' => $this->m_admin-
>getAssignment()
    );

    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
    $this->load-
>view('admin/assetAssignment/assignment', $data);
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
}

public function assignmentForm()
{
    $data['employee'] = $this->m_admin-
>getAllEmployeeOfficer();
    $data['type'] = $this->m_admin-
>getAllData('pm_type_Asset');

    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
}

```

```

        $this->load-
>view('admin/assetAssignment/assignmentForm', $data);
        $this->load->view('resource/admin/footer');
        $this->load->view('style/style');
    }

    public function addAssignment()
    {
        $data = array(
            'ID_EMPLOYEE' => $this->input-
>post('ID_EMPLOYEE'),
            'ID_TYPE' => $this->input->post('ID_TYPE')
        );
        // print_r($data);
        // die;
        $this->m_admin-
>InsertData('pm_asset_assignment', $data);
        redirect("C_admin/assetAssignment");
    }

    public function assignmentEdit($ID)
    {
        $where = array('ID_ASSIGNMENT' => $ID);
        $data['assignment'] = $this->m_admin-
>EditAllData($where, 'pm_asset_assignment');
        $data['employee'] = $this->m_admin-
>getEmployeeAssignment($ID);
        $data['type'] = $this->m_admin-
>getAllData('pm_type_Asset');

        $this->load->view('resource/admin/header');
        $this->load->view('resource/admin/sidebar');
        $this->load-
>view('admin/assetAssignment/assigmentEdit', $data);
        $this->load->view('resource/admin/footer');
        $this->load->view('style/style');
    }

```

```
public function assignmentUpdate()
{
    $data = array(
        'ID_TYPE' => $this->input->post('ID_TYPE')
    );

    $where = array(
        'ID_ASSIGNMENT' => $this->input-
>post('ID_ASSIGNMENT')
    );
    // print_r($data);
    // die;
    $this->m_admin-
>UpdateAllData($where, $data, 'pm_asset_assignment');
    redirect('C_admin/assetAssignment');
}

public function assignmentDelete($ID)
{
    $where = array('ID_ASSIGNMENT' => $ID);
    $this->m_admin-
>DeleteAllData($where, 'pm_asset_assignment');
    redirect('C_admin/assetAssignment');
}

// ===== END AS
SET ASSIGNMENT =====
=====
// ===== MESSAG
E REMINDER =====
=====
public function defaultMessage()
{
    $data = array(
        'message' => $this->m_admin-
>getAllData('pm_message')
    );
    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
```

```
        $this->load-
>view('admin/messageReminder/defaultMessage', $data);
        $this->load->view('resource/admin/footer');
        $this->load->view('style/style');
    }

    public function defaultMessageEdit($ID)
{
    $where = array('ID' => $ID);
    $data['message'] = $this->m_admin-
>EditAllData($where, 'pm_message');

    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
    $this->load-
>view('admin/messageReminder/defaultMessageEdit', $data
);
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
}

public function defaultMessageUpdate()
{
    $data = array(
        'MESSAGE' => $this->input->post('MESSAGE')
    );

    $where = array(
        'ID' => $this->input->post('ID')
    );
    // print_r($data);
    // die;
    $this->m_admin-
>UpdateAllData($where, $data, 'pm_message');
    redirect('C_admin/defaultMessage');
}
```

```
public function email()
{
    $data['employee'] = $this->m_admin-
>getAllData('pm_master_employee');
    $data['email'] = $this->m_admin-
>DefaultEmail_1();

    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
    $this->load-
>view('admin/messageReminder/email', $data);
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
}

public function chat()
{
    $data['employee'] = $this->m_admin-
>getAllData('pm_master_employee');
    $data['chat'] = $this->m_admin-
>DefaultChat_1();
    $data['tes'] = $this->session-
>flashdata('msg');

    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
    $this->load-
>view('admin/messageReminder/chat', $data);
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
}

public function kirim_email()
{
    // Konfigurasi email
    $config = [
        'mailtype' => 'html',
        'charset' => 'utf-8',
        'protocol' => 'smtp',
        'host' => '127.0.0.1',
        'port' => 25,
        'username' => 'root',
        'password' => '',
        'timeout' => 60
    ];
    $this->load->view('admin/messageReminder/kirim_email', $config);
}
```

```
'smtp_host' => 'smtp.gmail.com',
'smtp_user' => '', // Email gmail
'smtp_pass' => '', // Password gmail
'smtp_crypto' => 'ssl',
'smtp_port' => 465,
'crlf' => "\r\n",
'newline' => "\r\n"
];

$to = $this->input->post('To');
$subject = $this->input->post('Subject');
$body = $this->input->post('Body');

// foreach ($to as $too) {
// print_r($too);
$this->load->library('email', $config);
$this->email-
>from('', 'Preventive Maintenance');
$this->email->to($to);
$this->email->subject($subject);
$this->email->message($body);
// }

// die;

if ($this->email->send()) {
    echo 'Sukses! email berhasil dikirim.';
    redirect('C_admin/email');
} else {
    echo 'Error! email tidak dapat dikirim.';
    redirect('C_admin/email');
}
}

public function kirim_pesan()
{
    $number = $this->input->post('To');
    $body = $this->input->post('Body');
    $data = [
```

```
        'phone' => $number, // Receivers phone
        'body' => $body, // Message
    ];
$json = json_encode($data);
$url = 'https://eu38.chat-
api.com/instance91716/message?token=s8iaiqtjshawptel';

// Make a POST request
$options = stream_context_create([
    'http' => [
        'method' => 'POST',
        'header' => 'Content-
type: application/json',
        'content' => $json
    ]
]);

// Send a request
$result = file_get_contents($url, false, $options);
// print_r($result);
// }

$d = date('d F Y');
$r = 'PUDC Preventive Maintenance : mengingatka
n kembali untuk melakukan pemeliharaan pada tanggal ' .
$d . ' dan silahkan login pada pm untuk lebih lengkapn
ya';
// print_r($r);
// die;
$data1 = array(
    'DestinationNumber' => $this->input-
>post('To'),
    'TextDecoded' => $body,
    'CreatorID' => 'PM Maintenance',
);

// print_r($data1);
// die;
```

```
        $insert = $this->curl->simple_post($this->API . '/Api_sms', $data1, array(CURLOPT_BUFFERSIZE => 10));
        if ($insert) {
            $this->session->set_flashdata('msg', 'Insert data berhasil');
        } else {
            $this->session->set_flashdata('msg', 'Insert Data Gagal');
        }
        redirect('C_admin/chat');
    }

// ===== END MESSAGE REMINDER =====
====

// ===== MAINTENANCE =====
====

public function maintenanceSchedulling()
{
    $data = array(
        'maintenance' => $this->m_admin->getSchedule());
}

$this->load->view('resource/admin/header');
$this->load->view('resource/admin/sidebar');
$this->load->view('admin/maintenance/maintenance', $data);
$this->load->view('resource/admin/footer');
$this->load->view('style/style');
}

public function Konfirmasi_Status()
{
```

```
$data = array(
    #database
    'STATUS' => 1,
);
$where = array(
    'ID_M' => $this->uri->segment('3')
);
// print_r($data);
// die;
$this->m_admin-
>UpdateAllData($where, $data, 'pm_maintenance_trans');
redirect('C_admin/maintenanceSchedulling');
}

public function Konfirmasi_Status_Belum()
{
    $data = array(
        #database
        'STATUS' => 0,
    );
    $where = array(
        'ID_M' => $this->uri->segment('3')
    );
    // print_r($data);
    // die;
    $this->m_admin-
>UpdateAllData($where, $data, 'pm_maintenance_trans');
    redirect('C_admin/maintenanceSchedulling');
}

public function maintenanceSchedullingForm()
{
    $data = array(
        'maintenance' => $this->m_admin-
>getSchedule()
    );
}
```

```
        $data['type'] = $this->m_admin->getAllData('pm_type_Asset');

        $this->load->view('resource/admin/header');
        $this->load->view('resource/admin/sidebar');
        $this->load->view('admin/maintenance/maintenanceForm', $data);
        $this->load->view('resource/admin/footer');
        $this->load->view('style/style');
    }

    public function addMaintenance()
    {
        $tujuh_hari = mktime(0, 0, 0, date("n"), date("j") + 7, date("Y"));
        $kurangsehari = mktime(0, 0, 0, date("n"), date("j") - 1, date("Y"));
        $dateH = $this->input->post('DATE_H');
        // $dateH = strtotime($dateH);
        // print_r($dateH);
        // die;
        $date1Plus = date('Y-m-d', strtotime($dateH . ' + 1 days'));
        $date1Min = date('Y-m-d', strtotime($dateH . ' - 1 days'));
        // strtotime()
        // print_r($date1Plus);
        // die;
        $data = array(
            'ID_TYPE' => $this->input->post('ID_TYPE'),
            'DATE_H_MIN' => $date1Min,
            'DATE_H' => $this->input->post('DATE_H'),
            'DATE_H_PLUS' => $date1Plus,
        );
        // print_r($data);
        // die;
        $this->m_admin->InsertData('pm_maintenance_trans', $data);
    }
}
```

```
        redirect("C_admin/maintenanceSchedulling");
    }

    public function mantenancetDelete($ID)
    {
        $where = array('ID_M' => $ID);
        $this->m_admin-
>DeleteAllData($where, 'pm_maintenance_trans');
        redirect('C_admin/maintenanceSchedulling');
    }

    public function tes()
    {
        $data['tes'] = $this->session-
>flashdata('msg');
        $this->load->view('resource/admin/header');
        $this->load->view('resource/admin/sidebar');
        $this->load-
>view('admin/maintenance/tes', $data);
        $this->load->view('resource/admin/footer');
        $this->load->view('style/style');
    }

    public function tes_tanggal()
    {
        $date = date('Y-m-d');
        $q = $this->db->get('pm_maintenance_trans')-
>result_array();
        foreach ($q as $row) {
            if ($date == $row['DATE_H_MIN']) {
                if ($row['STATUS'] == 0) {
                    $ID_TYPE = $row['ID_TYPE'];
                    $q2 = $this->db-
>get_where('pm_asset_assignment', ['ID_TYPE' => $ID_TYP
E])->result_array();
                }
            }
        }
    }
}
```

```

        foreach ($q2 as $q2) {
            $ID_EMPLOYEE = $q2['ID_EMPLOYEE'];
        ];
        $q3 = $this->db->get_where('pm_master_employee', ['ID' => $ID_EMPLOYEE])->row();
        $email = $q3->EMAIL;
        $config = [
            'mailtype' => 'html',
            'charset' => 'utf-8',
            'protocol' => 'smtp',
            'smtp_host' => 'smtp.gmail.com',
            'smtp_user' => '', // Email
            'smtp_pass' => '', // Password
            'smtp_crypto' => 'ssl',
            'smtp_port' => 465,
            'crlf' => "\r\n",
            'newline' => "\r\n"
        ];
        $date1 = date('d F Y');
        $date_coba = date('d F Y', strtotime($date1 . ' + 1 days'));
        $date_subject = date('d F Y');
        $subject = 'Reminder Maintenance,' . $date_coba;
        // $body = $this->m_admin->DefaultEmail();
        // $body = $body->MESSAGE;
        $name = $q3->NAME;

        $body = "
<h1>Reminder Maintenance</h1>
<p><h3>Dear, $name</h3></p>

```

```

<p>Mengingatkan kembali kepada <b>$name </b> untuk melakukan pemeliharaan pada tanggal $date_coba </p>
<p><b>$name </b> dapat melakukan login pada pada Preventive Maintenance <b> http://192.168.43.9/pm/ </b> untuk melihat daftar jadwal lainnya</p>
<p>Terima Kasih</p>
<h3><h3>Regards,<br> Admin PUDC</h3>
></p>";

        $this->load-
>library('email', $config);
        $this->email-
>from('', 'Preventive Maintenance');
        $this->email->to($email);
        $this->email-
>subject($subject);
        $this->email->message($body);
        $this->email->send();

// =====
=====KIRIM WHATSAPP=====
=====

$phone_number = $q3-
>PHONE_NUMBER;
$name = $q3->NAME;
// $body_chat = $this->m_admin-
>DefaultChat();
// $body_chat = $body_chat-
>MESSAGE;
$body_chat = "*Reminder Mainten
ance*
*Dear, $name* Mengingatkan kembali
kepada $name untuk melakukan pemeliharaan pada tanggal
$date_coba. Untuk lebih lengkapnya $name dapat melakuka

```

```
n login pada Preventive Maintenance *http://192.168.43.9/pm/*
Terima Kasih
*Regards,* 
*Admin PUDC*';

$data = [
    'phone' => $phone_number,
    / Receivers phone
    'body' => $body_chat, // Me
ssage
];
$json = json_encode($data);
$url = 'https://eu38.chat-
api.com/instance91716/message?token=s8iaiqtjshawptel';

$options = stream_context_creat
e([
    'http' => [
        'method' => 'POST',
        'header' => 'Content-
type: application/json',
        'content' => $json
    ]
]);
$result = file_get_contents($ur
l, false, $options);
// print_r($result);

$r = "PUDC Preventive Maintenance : mengingatkan kembali untuk melakukan pemeliharaan pada tanggal $date_coba dan silahkan login pada pm untuk lebih lengkapnya";
// print_r($r);
// die;
$data1 = array(
    'DestinationNumber' =>
$phone_number,
```

```

        'TextDecoded'      => $r,
        'CreatorID'       => 'PM Main
tenance',
    );

        // print_r($data1);
        // die;
        $insert = $this->curl-
>simple_post($this-
>API . '/Api_sms', $data1, array(CURLOPT_BUFFERSIZE =>
10));
        if ($insert) {
            $this->session-
>set_flashdata('msg', 'Insert data berhasil');
        } else {
            $this->session-
>set_flashdata('msg', 'Insert Data Gagal');
        }
    }
} elseif ($date == $row['DATE_H']) {
    if ($row['STATUS'] == 0) {
        $ID_TYPE = $row['ID_TYPE'];
        $q2 = $this->db-
>get_where('pm_asset_assignment', ['ID_TYPE' => $ID_TYPE])->result_array();

        foreach ($q2 as $q2) {
            $ID_EMPLOYEE = $q2['ID_EMPLOYEE'];
        }
        $q3 = $this->db-
>get_where('pm_master_employee', ['ID' => $ID_EMPLOYEE])->row();
        $email = $q3->EMAIL;
        $config = [
            'mailtype'  => 'html',
            'charset'   => 'utf-8',
            'protocol'  => 'smtp',

```

```

        'smtp_host' => 'smtp.gmail.
com',
        'smtp_user' => '', // Email
        'smtp_pass' => '', // Pa
        'smtp_crypto' => 'ssl',
        'smtp_port' => 465,
        'crlf' => "\r\n",
        'newline' => "\r\n"
    ];
$date1 = date('d F Y');
$date_coba = date('d F Y', strtotime($date1 . ' + 0 days'));
$date_subject = date('d F Y');
$subject = 'Reminder Maintenance,' . $date_coba;
// $body = $this->m_admin-
>DefaultEmail();
// $body = $body->MESSAGE;
$name = $q3->NAME;

$body = "
<h1>Reminder Maintenance</h1>
<p><h3>Dear, $name</h3></p>
<p>Mengingatkan kembali kepada <b>$name </b> untuk melakukan pemeliharaan pada tanggal $date_coba </p>
<p><b> $name </b> dapat melakukan login pada pada Preventive Maintenance <b> http://192.168.43.9/pm/ </b> untuk melihat daftar jadwal lainnya</p>
<p>Terima Kasih</p>
<h3><h3>Regards,<br> Admin PUDC</h3>
></p>";

$this->load-
>library('email', $config);
        $this->email-
>from('', 'Preventive Maintenance');

```

```
        $this->email->to($email);
        $this->email-
>subject($subject);
        $this->email->message($body);
        $this->email->send();

        // =====
=====KIRIM WHATSAPP=====
=====

$phone_number = $q3-
>PHONE_NUMBER;
$name = $q3->NAME;
// $body_chat = $this->m_admin-
>DefaultChat();
// $body_chat = $body_chat-
>MESSAGE;
$body_chat = "*Reminder Maintenance*
*Dear, $name* Mengingatkan kembali
kepada $name untuk melakukan pemeliharaan pada tanggal
$date_coba. Untuk lebih lengkapnya $name dapat melakuka
n login pada Preventive Maintenance *http://192.168.43.
9/pm/*

Terima Kasih
*Regards,*  

*Admin PUDC*";

$data = [
    'phone' => $phone_number, /
/ Receivers phone
    'body' => $body_chat, // Me
ssage
];
$json = json_encode($data);
$url = 'https://eu38.chat-
api.com/instance91716/message?token=s8iaiqtjshawptel';
```

```
        $options = stream_context_create([
            'http' => [
                'method' => 'POST',
                'header' => 'Content-type: application/json',
                'content' => $json
            ]
        ]);

        $result = file_get_contents($url, false, $options);
        // print_r($result);

        $r = "PUDC Preventive Maintenance : mengingatkan kembali untuk melakukan pemeliharaan pada tanggal $date_coba dan silahkan login pada pm untuk lebih lengkapnya";
        // print_r($r);
        // die;
        $data1 = array(
            'DestinationNumber' => $phone_number,
            'TextDecoded' => $r,
            'CreatorID' => 'PM Maintenance',
        );
        // print_r($data1);
        // die;
        $insert = $this->curl->simple_post($this->API . '/Api_sms', $data1, array(CURLOPT_BUFFERSIZE => 10));
        if ($insert) {
            $this->session->set_flashdata('msg', 'Insert data berhasil');
        } else {
```

```

        $this->session-
>set_flashdata('msg', 'Insert Data Gagal');
    }
}
} elseif ($date == $row['DATE_H_PLUS']) {
    if ($row['STATUS'] == 0) {
        $ID_TYPE = $row['ID_TYPE'];
        $q2 = $this->db-
>get_where('pm_asset_assignment', ['ID_TYPE' => $ID_TYP
E])->result_array();

        foreach ($q2 as $q2) {
            $ID_EMPLOYEE = $q2['ID_EMPLOYEE
'];
            $q3 = $this->db-
>get_where('pm_master_employee', ['ID' => $ID_EMPLOYEE]
)->row();
            $email = $q3->EMAIL;
            $config = [
                'mailtype' => 'html',
                'charset' => 'utf-8',
                'protocol' => 'smtp',
                'smtp_host' => 'smtp.gmail.
com',
                'smtp_user' => '', // Email
                'smtp_pass' => '', // Password
                'smtp_crypto' => 'ssl',
                'smtp_port' => 465,
                'crlf' => "\r\n",
                'newline' => "\r\n"
            ];
            $date1 = date('d F Y');
            $date_coba = date('d F Y', strtotime($date1 . ' + 0 days'));
            $date_subject = date('d F Y');

```

```

$subject = 'Reminder Maintenance
e,' . $date_coba;
// $body = $this->m_admin-
>DefaultEmail();
// $body = $body->MESSAGE;
$name = $q3->NAME;

$body = "
<h1>Reminder Maintenance</h1>
<p><h3>Dear, $name</h3></p>
<p>Mengingatkan kembali kepada <b>
$name </b> untuk melakukan pemeliharaan pada tanggal $d
ate_coba </p>
<p><b> $name </b> dapat melakukan l
ogin pada pada Preventive Maintenance <b> http://192.16
8.43.9/pm/ untuk melihat daftar jadwal lainnya</p>
<p>Terima Kasih</p>
<h3><h3>Regards,<br> Admin PUDC</h3
></p>";

$this->load-
>library('email', $config);
$this->email-
>from('', 'Preventive Maintenance');
$this->email->to($email);
$this->email-
>subject($subject);
$this->email->message($body);
$this->email->send();

=====KIRIM WHATSAPP=====
=====

$phone_number = $q3-
>PHONE_NUMBER;
$name = $q3->NAME;

```

```
// $body_chat = $this->m_admin->DefaultChat();
// $body_chat = $body_chat->MESSAGE;
$body_chat = "*Reminder Maintenance*
*Dear, $name* Mengingatkan kembali kepada $name untuk melakukan pemeliharaan pada tanggal $date_coba. Untuk lebih lengkapnya $name dapat melakukan login pada Preventive Maintenance *http://192.168.43.9/pm/*
Terima Kasih
*Regards,* 
*Admin PUDC*;

$data = [
    'phone' => $phone_number, /
/ Receivers phone
    'body' => $body_chat, // Message
];
$json = json_encode($data);
$url = 'https://eu38.chat-api.com/instance91716/message?token=s8iaiqtjshawptel';

$options = stream_context_create([
    'http' => [
        'method' => 'POST',
        'header' => 'Content-type: application/json',
        'content' => $json
    ]
]);
$result = file_get_contents($url, false, $options);
// print_r($result);
```

```
$r = "PUDC Preventive Maintenance : mengingatkan kembali untuk melakukan pemeliharaan pada tanggal $date_coba dan silahkan login pada pm untuk lebih lengkapnya";
// print_r($r);
// die;
$data1 = array(
    'DestinationNumber' =>
$phone_number,
    'TextDecoded' => $r,
    'CreatorID' => 'PM Maintenance',
);
// print_r($data1);
// die;
$insert = $this->curl->simple_post($this->API . '/Api_sms', $data1, array(CURLOPT_BUFFERSIZE => 10));
if ($insert) {
    $this->session->set_flashdata('msg', 'Insert data berhasil');
} else {
    $this->session->set_flashdata('msg', 'Insert Data Gagal');
}
}
}
}
}
}
redirect('C_admin/tes');
}
}
```

Tidak hanya C_admin saja untuk membuat Halaman admin, maka dibutuhkanlah sebuah model yang akan mengeksekusi query query dari form View ke Controller

M_admin.php

```
<?php
class m_admin extends ci_model
{
    public function getAllData($table)
    {
        return $this->db->get($table)->result_array();
    }

    function InsertData($table, $data)
    {
        $this->db->insert($table, $data);
    }

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

    function UpdateAllData($where, $data, $table)
    {
        $this->db->where($where);
        $this->db->update($table, $data);
    }
    function DeleteAllData($where, $table)
    {
        $this->db->where($where);
        $this->db->delete($table);
    }
}
```

```
function getAllEmployee()
{
    return $this->db-
>query("SELECT * FROM pm_master_employee
        order by pm_master_employee.NAME asc")-
>result_array();
}

function getAllEmployeeOfficer()
{
    return $this->db-
>query("SELECT * FROM pm_master_employee
        where pm_master_employee.ACCESS = 2 order by pm
_master_employee.NAME asc ")->result_array();
}

function getEmployeeAssignment($ID)
{
    return $this->db-
>query("SELECT * FROM pm_asset_assignment paa
        left join pm_master_employee pme
        on paa.ID_EMPLOYEE = pme.ID WHERE paa.ID_ASSIGN
MENT='$ID'")->result_array();
}

function DefaultEmail_1()
{
    return $this->db-
>query("SELECT * FROM pm_message pm WHERE pm.ID = 2")-
>result_array();
}

function DefaultEmail()
{
    return $this->db-
>query("SELECT * FROM pm_message pm WHERE pm.ID = 2")-
>row();
}

function DefaultChat()
{
```

```
        return $this->db-
>query("SELECT * FROM pm_message pm WHERE pm.ID = 1")-
>row();
    }

    function DefaultChat_1()
{
    return $this->db-
>query("SELECT * FROM pm_message pm WHERE pm.ID = 1")-
>result_array();
}

function cek_tanggal()
{
    return $this->db-
>query("SELECT * FROM pm_maintenance_trans")-
>result_array();
}

function cek_tanggal_type()
{
    return $this->db-
>query("SELECT * FROM pm_maintenance_trans pmt
        left join pm_type_asset pma
        on pmt.ID_TYPE = pma.ID")->result_array();
}

function cek_login($data)
{
    return $this->db-
>get_where('pm_master_employee', $data);
}

function asset_officer($ID)
{
    return $this->db-
>query("SELECT * FROM pm_master_asset pma
        left join pm_type_asset pta
        on pma.ASSET_TYPE = pta.ID")-
```

```
        left join pm_master_area pmar
        on pma.AREA = pmar.ID
        left join pm_asset_assignment pmas
        on pma.ASSET_TYPE = pmas.ID_TYPE WHERE pmas.ID_
EMPLOYEE = $ID and pma.ACTIVE=1")->result_array();
    }

function data_maintenance($ID)
{
    return $this->db-
>query("SELECT *,  DATE_FORMAT(DATE_H, '%d %M %Y') AS D
ATE_ FROM pm_master_employee pme
    inner join pm_asset_assignment paa
    inner join pm_type_asset pta
    on paa.ID_TYPE = pta.ID
    inner join pm_maintenance_trans pmt
    on pmt.ID_TYPE = paa.ID_TYPE
    on pme.ID = paa.ID_EMPLOYEE where pme.ID = $ID
order by pmt.DATE_H asc")->result_array();
}

function product_list()
{
    $hasil = $this->db->get('pm_master_area');
    return $hasil->result();
}

function getAssetReport($ID)
{
    $hasil = $this->db-
>query("SELECT * FROM pm_master_asset a
```

```
        LEFT JOIN pm_master_area b
        on a.AREA = b.ID where a.ASSET_TYPE= $ID and a.
ACTIVE = 1 ");
            return $hasil->result_array();
        }

        function save_product()
        {
            $data = array(
                'product_code' => $this->input-
>post('product_code'),
                'product_name' => $this->input-
>post('product_name'),
                'product_price' => $this->input-
>post('price'),
            );
            $result = $this->db->insert('product', $data);
            return $result;
        }

        function update_product()
        {
            $product_code = $this->input-
>post('product_code');
            $product_name = $this->input-
>post('product_name');
            $product_price = $this->input->post('price');

            $this->db->set('product_name', $product_name);
            $this->db-
>set('product_price', $product_price);
            $this->db-
>where('product_code', $product_code);
            $result = $this->db->update('product');
            return $result;
        }

        function delete_product()
        {
```

```
    $product_code = $this->input-
>post('product_code');
    $this->db-
>where('product_code', $product_code);
    $result = $this->db->delete('product');
    return $result;
}

public function getDataEmployee($table, $keyword)
{
    $this->db->like('NAME', $keyword);
    $this->db->OR_LIKE('AKSES', $keyword);
    return $this->db->get($table)->result_array();
}

public function getDataArea($table, $keyword)
{
    $this->db->like('AREA_NAME', $keyword);
    $this->db->OR_LIKE('ID', $keyword);
    return $this->db->get($table)->result_array();
}

public function getDataType($table, $keyword)
{
    $this->db->like('TYPE_NAME', $keyword);
    $this->db->OR_LIKE('ID', $keyword);
    return $this->db->get($table)->result_array();
}

public function GetVendor()
{
    return $this->db->get('tbl_vendor')-
>result_array();
}
```

```
public function GetLantai()
{
    return $this->db->get('tbl_lantai')-
>result_array();
}

public function GetBarang()
{
    return $this->db->get('tbl_barang')-
>result_array();
}

public function GetKategoriBarang()
{
    return $this->db->get('tbl_kategori_barang')-
>result_array();
}

public function GetKodeVendor()
{
    $q = $this->db-
>query("select MAX(RIGHT(KODE_VENDOR,3)) as kd_max from
tbl_vendor");
    $kd = "";
    if ($q->num_rows() > 0) {
        foreach ($q->result() as $k) {
            $tmp = ((int) $k->kd_max) + 1;
            $kd = sprintf("%03s", $tmp);
        }
    } else {
        $kd = "0001";
    }
    return "VNDR-" . $kd;
}

public function getKodeLantai()
{
```

```
$q = $this->db->query("select MAX(RIGHT(KODE_LANTAI,3)) as kd_max from tbl_lantai");
$kd = "";
if ($q->num_rows() > 0) {
    foreach ($q->result() as $k) {
        $tmp = ((int) $k->kd_max) + 1;
        $kd = sprintf("%03s", $tmp);
    }
} else {
    $kd = "001";
}
return "LT-" . $kd;
}

public function getKodeLokasi()
{
    $q = $this->db->query("select MAX(RIGHT(KODE_LOKASI,3)) as kd_max from tbl_lokasi");
    $kd = "";
    if ($q->num_rows() > 0) {
        foreach ($q->result() as $k) {
            $tmp = ((int) $k->kd_max) + 1;
            $kd = sprintf("%03s", $tmp);
        }
    } else {
        $kd = "001";
    }
    return "LKS-" . $kd;
}

public function getKodeKategoriBarang()
{
    $q = $this->db->query("select MAX(RIGHT(KODE_KATEGORI,3)) as kd_max from tbl_kategori_barang");
    $kd = "";
    if ($q->num_rows() > 0) {
```

```
        foreach ($q->result() as $k) {
            $tmp = ((int) $k->kd_max) + 1;
            $kd = sprintf("%03s", $tmp);
        }
    } else {
        $kd = "001";
    }
    return "KTG-" . $kd;
}

public function getKodeBarang()
{
    $q = $this->db-
>query("select MAX(RIGHT(KODE_BARANG,3)) as kd_max from
tbl_barang");
    $kd = "";
    if ($q->num_rows() > 0) {
        foreach ($q->result() as $k) {
            $tmp = ((int) $k->kd_max) + 1;
            $kd = sprintf("%03s", $tmp);
        }
    } else {
        $kd = "001";
    }
    return "AST-BRG-" . $kd;
}

public function GetDataLokasiLantai()
{
    return $this->db->query("SELECT * FROM
tbl_lokasi a
INNER JOIN tbl_lantai b
on a.KODE_LANTAI = b.KODE_LANTAI")-
>result_array();
}

function getMessages()
{
```

```
        return $this->db-
>query("select * from tbl_pesanan where id_pesanan = 1")-
>result();
    }

    function getTanggal()
    {
        return $this->db-
>query("select * from tbl_schedule")->result_array();
    }

    function getAssignment()
    {
        return $this->db->query("select * from
pm_asset_assignment pas
inner join pm_master_employee pme
on pas.ID_EMPLOYEE = pme.ID
inner join pm_type_asset pma
on pma.ID = pas.ID_TYPE
")->result_array();
    }

    function getAsset()
    {
        return $this->db->query("select * from
pm_master_Asset pma
left join pm_type_asset pta
on pma.ASSET_TYPE = pta.ID
left join pm_master_area pmar
on pma.AREA = pmar.ID
")->result_array();
    }
}
```

```

        function getSchedule()
        {
            return $this->db-
>query("SELECT *, DATE_FORMAT(DATE_H, '%d %M %Y') AS DA
TE_ from
            pm_maintenance_trans pmt
            left join pm_type_asset pta
            on pmt.ID_TYPE = pta.ID order by pta.TYPE_NAME
asc
            ")->result_array();
        }

        function search($keyword)
        {
            $this->db->like('Asset_Name', $keyword);
            $query = $this->db->get('tbl_barang');
            return $query->result();
        }
    }
}

```

Pada initinya C_admin diatas berisi perintah perintah dimana perintah tersebut di khususkan dalam hak akses Admin. Dalam pembentukan halaman admin ini saya membagi 4 bagian yaitu header, sidebar dan footer untuk admin dan penulis letakan pada file php pada folder view/resource/admin/ dan tidak pula memisahkan CSS pada folder view/style/style

Admin/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">
        <meta name="description" content="">
        <meta name="author" content="">
        <link rel="icon" type="image/png/icon" sizes="16x16" href=<?php echo base_url(); ?>assets/images/pertamina.ico">
        <title>Preventive Maintenance</title>
</head>

<body class="fix-sidebar fix-header logo-center card-no-border">
    <div class="preloader">
        <svg class="circular" viewBox="25 25 50 50">
            <circle class="path" cx="50" cy="50" r="20" fill="none" stroke-width="2" stroke-miterlimit="10" /> </svg>
    </div>

    <header class="topbar">
        <nav class="navbar top-navbar navbar-expand-md navbar-light">
            <div class="navbar-collapse">
                <ul class="navbar-nav mr-auto mt-md-0 ">
                    <li class="nav-item"> <a class="nav-link nav-toggler hidden-md-up text-muted waves-effect waves-dark" href="javascript:void(0)"><i class="ti-menu"></i></a> </li>
                    <li class="nav-item"> <a class="nav-link sidebar-toggler hidden-sm-down text-muted waves-effect waves-dark" href="javascript:void(0)"><i class="icon-arrow-left-circle"></i></a> </li>
                </ul>
            </div>
        </nav>
    </header>

```

```

        <ul class="navbar-nav my-lg-0">
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-
toggle text-muted waves-effect waves-
dark" href="#" data-toggle="dropdown" aria-
haspopup="true" aria-
expanded="false"></a>
                <div class="dropdown-
menu dropdown-menu-right animated flipInY">
                    <ul class="dropdown-user">
                        <li>
                            <div class="dw-
user-box">
                                <center>
                                    <div class=
"u-
img"><br></div>
                                </center><br>
                                <center>
                                    <h4><?php e
cho $this->session->userdata( 'NAME' ); ?></h4>
                                    <p><?php ec
ho $this->session->userdata( 'EMAIL' ); ?></p>
                                </center>
                            </div>
                        </li>
                    </ul>
                </div>
            </li>
        </ul>
    </div>

```

Admin/sidebar.php

```

<aside class="left-sidebar">
    <div class="scroll-sidebar">
        <div class="user-profile">
            <div class="profile-
img">  </div>
    <div class="profile-
text"> <?php echo $this->session-
>userdata('NAME'); ?><span class="caret"></span>
    </div>
</div>
<nav class="sidebar-nav">
    <ul id="sidebarnav">
        <center>
            <li class="nav-small-cap">
                <?php echo date('d F Y'); ?><p
id="jam"></p>
                <!--
- <?php echo date('d m y'); ?> -->
                </li>
            </center>
            <li><a href=<?php echo base_url(); ?>C
_Admin/employeeManagement" aria-
expanded="false"><i class="fas fa-address-
card"></i><span class="hide-
menu">Employee Management</span></a> </li>
            <li>
                <a class="has-arrow" href="#" aria-
expanded="false"><i class="fas fa-
archive"></i><span class="hide-
menu">Asset Management</span></a>
                <ul aria-
expanded="false" class="collapse">
                    <li><a href=<?php echo base_ur
l(); ?>C_Admin/area">Data Area</a></li>
                    <li><a href=<?php echo base_ur
l(); ?>C_Admin/datatype">Data Type</a></li>
                    <li><a href=<?php echo base_ur
l(); ?>C_Admin/assetdata">Asset Data</a></li>
                </ul>
            </li>
            <li><a href=<?php echo base_url(); ?>C
_admin/assetAssignment" aria-
```

```
expanded="false">><i class="fas fa-address-
card"></i><span class="hide-
menu">Asset Assignment</span></a> </li>
            <li>
                <a class="has-
arrow " href="#" aria-
expanded="false">><i class="mdi mdi-
email"></i><span class="hide-
menu">Message Reminder</span></a>
                    <ul aria-
expanded="false" class="collapse">
                        <li><a href="php echo base_ur
l(); ?&gt;C_Admin/defaultMessage"&gt;Default Message&lt;/a&gt;&lt;/li&gt;
                        &lt;li&gt;&lt;a href="<?php echo base_ur
l(); ?&gt;C_Admin/chat"&gt;SMS &amp; Whatsapp&lt;/a&gt;&lt;/li&gt;
                        &lt;li&gt;&lt;a href="<?php echo base_ur
l(); ?&gt;C_Admin/email"&gt;E-mail&lt;/a&gt;&lt;/li&gt;
                    &lt;/ul&gt;
                &lt;/li&gt;
                &lt;li&gt;&lt;a href="<?php echo base_url(); ?&gt;C
_Admin/maintenanceSchedulling"&gt;&lt;i class="far fa-
calendar-alt"&gt;&lt;/i&gt;&lt;span class="hide-
menu"&gt;Maintenance&lt;/span&gt;&lt;/a&gt; &lt;/li&gt;
                    &lt;li&gt;&lt;a href="<?php echo base_url(); ?&gt;C
_admin/tes" aria-expanded="false"><i class="fas fa-
address-card"></i><span class="hide-
menu">tes</span></a> </li>
                </ul>
            </nav>
        </div>
    </aside>

<script>
    window.setTimeout("waktu()", 1000);

    function waktu() {
        var waktu = new Date();
        setTimeout("waktu()", 1000);
    }

```

```
        document.getElementById("jam").innerHTML = waktu.getHours() + ":" + waktu.getMinutes() + ":" + waktu.getSeconds();
        // document.getElementById("menit").innerHTML =
waktu.getMinutes();
        // document.getElementById("detik").innerHTML =
waktu.getSeconds();
    }
</script>
```

Admin/footer.php

```
<footer class="footer">

    <?php $tahun = date('Y'); ?>

    © <?php echo $tahun ?> Pertamina Upstraem Data Center

</footer>

</body>

<!--

-->

Mirrored from www.wrappixel.com/demos/admin-templates
/monster-admin/main/layout-logo-
center.html by HTTrack Website Copier/3.x [XR&CO'2014],

Mon, 11 Feb 2019 11:36:58 GMT

-->

</html>
```

```
var $API = "";

public function __construct()
{
    parent::__construct();
    // date_default_timezone_set('Asia/Jakarta');
    $this->API = "http://localhost/api_sms/index.php";
    if (!$this->session->userdata('USERNAME')) {
        redirect('C_login');
    }
}
```

pada `public function __construct()` ini berfungsi untuk membaca pertama kali saat file `C_admin` ini di jalankan, jika saat loggin session tidak terbaca username dari user yang melakukan login maka aplikasi akan otomatis redirect ke `C_kogin`. Dan `$this->api` ini berfungsi untuk menampilkan data pada aplikasi lainnya menggunakan restfull Api.

```
public function index()
{
    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
    $this->load->view('admin/home/index');
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
```

pada `public function index()` ini merupakan hal yang pertama di jalankan setelah membaca `construct` yang ada di dalam

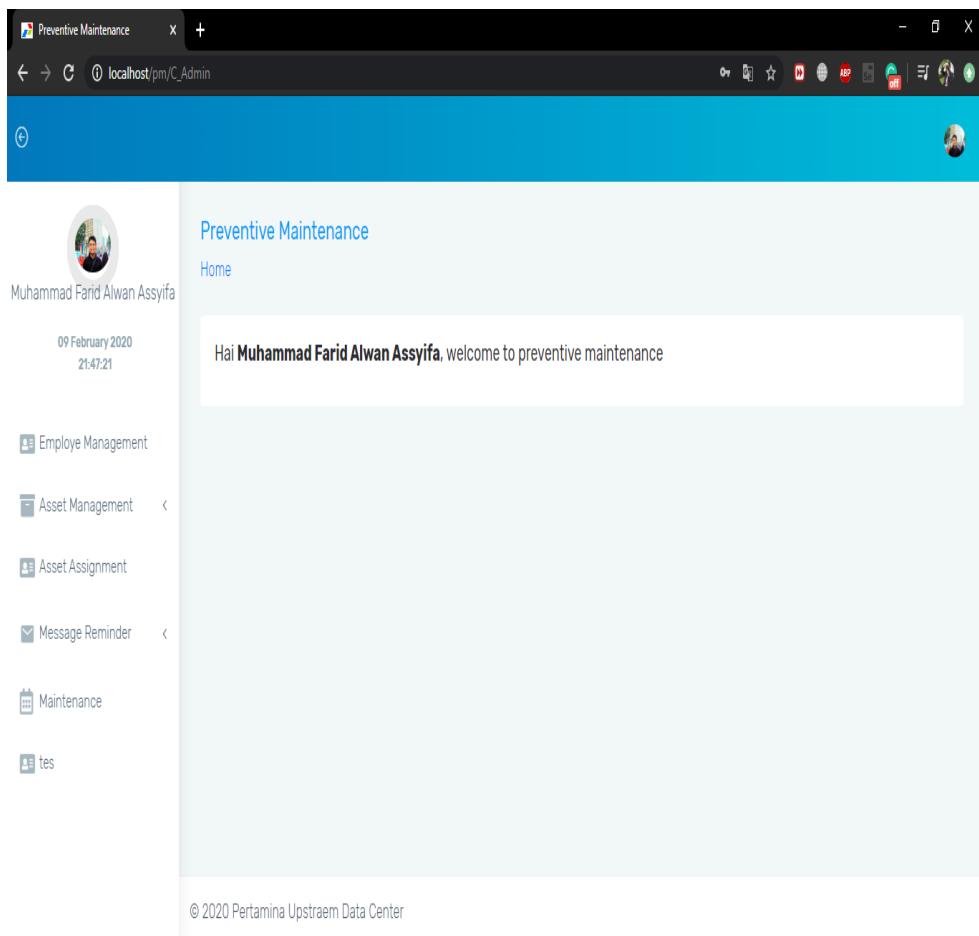
C_admin, function ini berfungsi untuk menampilkan halaman Admin dan memanggil header, sidebar, footer dan CSS yang berada pada alamat masing alamat foldernya.

Admin/home/index.php

```
<div class="page-wrapper">
    <div class="container-fluid">
        <div class="row page-titles">
            <div class="col-md-6 col-8 align-self-center">
                <h3 class="text-themecolor mb-0 mt-0">Preventive Maintenance</h3>
                <ol class="breadcrumb">
                    <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin>Home</a></li>
                    <!-- <li class="breadcrumb-item active">Asset Assignment</li> -->
                </ol>
            </div>
        </div>

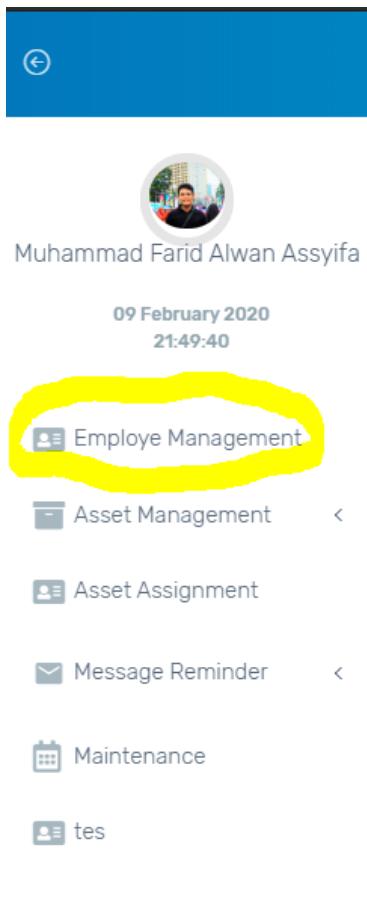
        <div class="row">
            <div class="col-12">
                <div class="card">
                    <div class="card-body">
                        <h4 class="card-title">Hai <b><?php echo $this->session->userdata('NAME'); ?></b>, welcome to preventive maintenance</h4>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

Jadi apabila kita sudah berhasil login dengan hak akses sebagai admin maka aplikasi akan membaca C_admin dan function index() lah yang akan erbaca pertama kali dan akan menampilkan gambar seperti di bawah ini.



5.4.1 Pembuatan Halaman Employee Management

Apabila kita menekan menu Employee Management pada gambar di bawah ini



Maka aplikasi akan memproses C_admin dan menjalankan script di bawah ini

```
public function employeeManagement()
{
```

```
        $data['employee'] = $this->m_admin-
>getAllEmployee();
        $this->load->view('resource/admin/header');
        $this->load->view('resource/admin/sidebar');
        $this->load->view(
            'admin/employee/employeeView', $data);
        $this->load->view('resource/admin/footer');
        $this->load->view('style/style');
    }
```

Pada function ini juga, memanggil function **getAllEmployee()** pada model m_admin. Script **getAllemployee()**

```
function getAllEmployee()
{
    return $this->db-
>query("SELECT * FROM pm_master_employee
        order by pm_master_employee.NAME asc")-
>result_array();
}
```

Kemudian data yang di ambil dari **getAllEmployee()** ini akan di parsing ke file view yang benama **employeeView** pada folder '**admin/employee**' kemudian didalam file **employeeView** tersebut kita masukan *query* untuk menampilkan data atau untuk lebih jelasnya bias lihat script **employeeView** di bawah ini

admin/employee/employeeView.php

```
<div class="page-wrapper">
    <div class="container-fluid">
        <div class="row page-titles">
```

```

<div class="col-md-6 col-8 align-self-center">
    <h3 class="text-themecolor mb-0 mt-0">Preventive Maintenance</h3>

    <ol class="breadcrumb">
        <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin>Home</a></li>
        <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin/employee Management>Employee Management</a></li>
            <!-- <li class="breadcrumb-item active">Asset Assignment</li> -->
        </ol>
    </div>
</div>
<?php echo $tes ?>

<div class="row">
    <div class="col-12">
        <div class="card">
            <div class="card-body">
                <div class="table-responsive m-t-40">
                    <table id="myTable" class="table table-bordered table-striped">
                        <thead>
                            <tr>
                                <th>#</th>
                                <th>Image</th>
                                <th>Name</th>
                                <th>Phone Number</th>
                                <th>Email</th>
                                <th>Access</th>
                                <th>
                                    <center>

```

```
          <a href=
"<?php echo base_url(); ?>C_admin/employeeForm" class="b
tn btn-outline-primary btn-rounded float-
center"><i class="fas fa-plus"></i></a>
          </center>
        </th>
      </tr>
    </thead>
  <tbody>
    <?php
      $no = 1;
      // $q = $this->uri-
>segment('1');
      // echo $q;
      foreach ($employee a
s $row) {
      ?>

      <tr>
        <td><?php ec
ho $no++ ?></td>
        <td>
          <div cla
ss="user-profile">
            
          </div>
        </td>
        <td><?php ec
ho $row['NAME']; ?></td>
        <td><?php ec
ho $row['PHONE_NUMBER']; ?></td>
        <td><?php ec
ho $row['EMAIL']; ?></td>
        <td>
          <?php
```

```
        if ($row
['ACCESS'] == 1) { ?>
    Admi
n
    <?php ?>
else { ?>
    Offi
cer
    <?php ?>
?>
    </td>
<td>
    <center>
        <a h
ref=<?php echo base_url('C_admin/employeeEditPhotos/' .
$row['ID']); ?>" class="btn btn-outline-primary btn-
rounded"><i class="fas fa-image"></i></a>
        <a h
ref=<?php echo base_url('C_admin/employeeEdit/' . $row[
'ID']); ?>" class="btn btn-outline-warning btn-
rounded"><i class="fas fa-pencil-alt"></i></a>
        <a h
ref=<?php echo base_url('C_admin/employeeDelete/' . $ro
w['ID']); ?>" class="btn btn-outline-danger btn-
rounded" onclick="return confirm('Are you sure you want
to delete <?php echo $row['NAME'] ?> ?')"><i class="fas
fa-trash"></i></a>
    </center>
>
    </td>
</tr>
<?php } ?>
</tbody>
</table>
</div>
</div>
</div>
```

```

        </div>
    </div>

    </div>
</div>
```

Setelah mengikuti script seperti diatas, maka saat menekan menu ***employee Management*** maka akan menampilkan gambar seperti diabwah ini

#	Name	Phone Number	Email	Access	
1	Iko Suropo	+6285822742574	iko.suropotomunari@gmail.com	Officer	
2	Cikro Edi Prawiro	+6282284216111	cikroed2@gmail.com	Officer	
3	Muhammad Fird Alwan Assyifa	+6285200584859	muhammadfirdalwan@gmail.com	Admin	
4	Muhammad Lutfi	+6285333977734	lutfimoeftonmad@gmail.com	Officer	
5	Naufal Ramadhan	+6285640435363	ramnaufal@gmail.com	Officer	
6	Nur Ima Tahirah	+6282289960144	nurismatirah@gmail.com	Officer	
7	Pak Roni Andaryan	+6285220237364	roniandaryan@politeknik.ac.id	Officer	
8	Puadi Hamzani	+6282129812729	puadhamzani@gmail.com	Officer	
9	Regina Putri Anjani	+6285236108839	reginaputri@gmail.com	Officer	

Kemudian untuk menambahkan ***employee*** kita dapat menekan pada sebelah kanan atas, ketika menekan btton tersebut maka dia akan memanggil *function employeeForm()* pada *controller C_admin.*

```
public function employeeForm()
{
    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
    $this->load->view('admin/employee/employeeForm');
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
}
```

Pada function ***employeeForm()*** ini, dia memanggil *view* pada folder ***admin/employee/employeeForm.php***

EmployeeForm.php

```
<div class="page-wrapper">
    <div class="container-fluid">
        <div class="row page-titles">
            <div class="col-md-6 col-8 align-self-center">
                <h3 class="text-themecolor mb-0 mt-0">Preventive Maintenance</h3>
                <ol class="breadcrumb">
                    <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin">Home</a></li>
                    <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin/employeeManagement">Employee Management</a></li>
                    <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin/employeeForm">Add Employee</a></li>
                    <!-- <li class="breadcrumb-item active">Asset Assignment</li> -->
                </ol>
            </div>
        </div>
    </div>
```

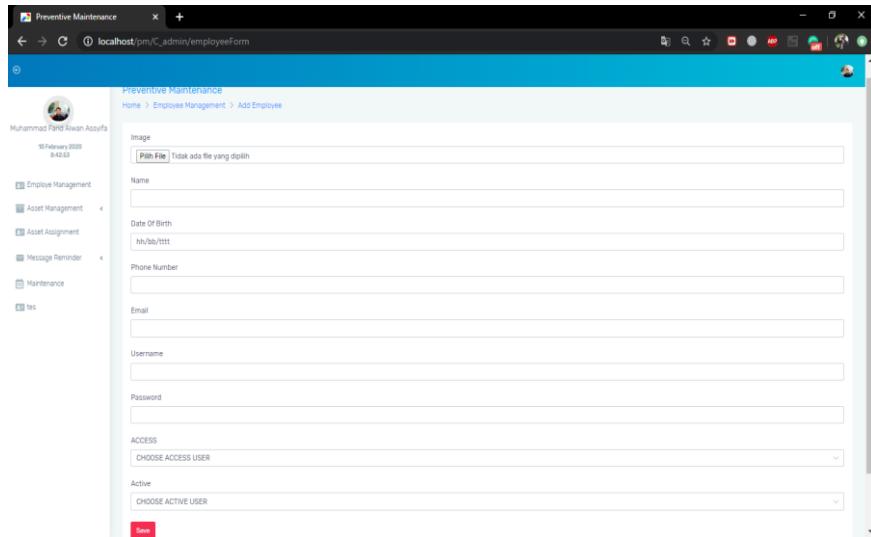
```
<div class="row">
    <div class="col-12">
        <div class="card">
            <div class="card-body">
                <form action="=base_url(); ?&gt;C_admin/addEmployee" method="post" enctype="multipart/form-data"&gt;
                    &lt;div class="form-group"&gt;
                        &lt;label class="control-label"&gt;Image&lt;/label&gt;
                        &lt;input type="file" class="form-control" name="IMAGE" required&gt;
                    &lt;/div&gt;
                    &lt;div class="form-group"&gt;
                        &lt;label class="control-label"&gt;Name&lt;/label&gt;
                        &lt;input type="text" class="form-control" name="NAME" required&gt;
                    &lt;/div&gt;
                    &lt;div class="form-group"&gt;
                        &lt;label class="control-label"&gt;Date Of Birth&lt;/label&gt;
                        &lt;input type="date" class="form-control" name="DATE_OF_BIRTH" required&gt;
                    &lt;/div&gt;
                    &lt;div class="form-group"&gt;
                        &lt;label class="control-label"&gt;Phone Number&lt;/label&gt;
                        &lt;input type="text" class="form-control" name="PHONE_NUMBER" required&gt;
                    &lt;/div&gt;
                    &lt;div class="form-group"&gt;
                        &lt;label class="control-label"&gt;Email&lt;/label&gt;
                        &lt;input type="email" class="form-control" name="EMAIL" required&gt;
                    &lt;/div&gt;
                &lt;div class="form-group"&gt;</pre
```

```
        <label class="control-
label">Username</label>
        <input type="text" class
="form-control" name="USERNAME" required>
    </div>
    <div class="form-group">
        <label class="control-
label">Password</label>
        <input type="password" c
lass="form-control" name="PASSWORD" required>
    </div>
    <div class="form-group">
        <label>ACCESS</label>
        <select class="form-
control custom-select" name="ACCESS" required>
            <option value="">CHO
OSE ACCESS USER</option>
            <option value="1">AD
MIN</option>
            <option value="2">OF
FICER</option>
        </select>
    </div>
    <div class="form-group">
        <label>Active</label>
        <select class="form-
control custom-select" name="ACTIVE" required>
            <option value="">CHO
OSE ACTIVE USER</option>
            <option value="N">NO
</option>
            <option value="Y">YE
S</option>
        </select>
    </div>
    <button type="submit" class=
"btn btn-danger waves-effect waves-light">Save</button>
</form>
</div>
```

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

</div>
</div>
```

Maka sistem akan menampilkan halaman seperti di bawah ini.



Kemudian kita dapat menginputkan data pada textbox yang ada disana kemudian kita atur function untuk menyimpan terlebih dahulu pada form ini sehingga dapat untuk menyimpan data employee nanti, pertama buka c_admin dan ketikan script seperti di bawah ini.

```
public function addEmployee()
{
    $data = array(
        'NAME' => $this->input->post('NAME'),
        'DATE_OF_BIRTH' => $this->input-
>post('DATE_OF_BIRTH'),
        'EMAIL' => $this->input->post('EMAIL'),
        'PHONE_NUMBER' => $this->input-
>post('PHONE_NUMBER'),
        'USERNAME' => $this->input-
>post('USERNAME'),
        'PASSWORD' => $this->input-
>post('PASSWORD'),
        'ACCESS' => $this->input->post('ACCESS'),
        'ACTIVE' => $this->input->post('ACTIVE'),
        'IMAGE' => $this->input->post('IMAGE'),
    );
    if (!empty($_FILES['IMAGE']['name'])) {
        $upload = $this->_do_upload();
        $data['IMAGE'] = $upload;
    }

    // print_r($data);
    // die;
    $this->m_admin-
>InsertData('pm_master_employee', $data);
    redirect("C_admin/employeeManagement");
}

private function _do_upload()
{
    $config['upload_path']      = 'assets/uploads/';
;
    $config['allowed_types']    = 'gif|jpg|png|jpeg
';
    $config['max_size']         = 10000;
    $config['max_width']        = 10000;
    $config['max_height']       = 10000;
```

```
$config['file_name'] = round(microtime(true) * 1000);

        $this->load->library('upload', $config);
        if (!$this->upload->do_upload('IMAGE')) {
            $this->session->set_flashdata('msg', $this-
>upload->display_errors('', ''));  
            redirect('C_admin/employeeManagement');
        }
        return $this->upload->data('file_name');
    }
```

Fungsi dari script diatas ini adalah untuk menerima inputan dari view dimana inputan data tersebut akan diparsing kedalam **database pm_master_employee** melalui model m_admin pada function **InsertData()**

```
function InsertData($table, $data)
{
    $this->db->insert($table, $data);
}
```

Kemudian kita dapat menginputkan data untuk **employee** yang akan ditambahkan pada sistem pertama pilih foto yang akan dijadikan profile kemudian inputkan **name “Zalwan”, date of birth “10 Februari 1998”, phone “+6285200684899”, email “zalwan@gmail.com”, username “zalwan”, password “zalwan”, access ” OFFICER” dan active “yes”**. Maka hasilnya seperti gambar pada halaman selanjutnya.

Preventive Maintenance

Home > Employee Management > Add Employee

Image

Pilih File [157691735905.png]

Name

Zalwan

Date Of Birth

10/02/1998

Phone Number

085943468059

Email

Zalwan@gmail.com

Username

Zalwan

Password

ACCESS

OFFICER

Active

YES

Save

Kemudian kita tekan **button save** mberpindah halaka otomatis akan pindah ke halaman employee management dan data yang di inputkan sebelumnya akan tampil pada halaman tersebut

Preventive Maintenance

Home > Employee Management

#	Image	Name	Phone Number	Email	Access	
1	[User Icon]	ap supripto	+6283822742574	apsupripto@gmail.com	Officer	
2	[User Icon]	Coko Edi Prawiro	+6282204216111	cokoredi@gmail.com	Officer	
3	[User Icon]	Muhammad Fird Alwan Assyifa	+6285200884899	muhammadfirdalwana@gmail.com	Admin	
4	[User Icon]	Muhammad Lutfi	+628533397734	lutfmoenam@gmail.com	Officer	
5	[User Icon]	Naufal Ramadhan	+6285640435583	rjmnaufal@gmail.com	Officer	
6	[User Icon]	Nur Ima Tarian	+6282285950164	nuriatariana@gmail.com	Officer	
7	[User Icon]	Pak Roni Andarsyah	+6285220267364	roniandarsyah@politeknikipos.ac.id	Officer	
8	[User Icon]	Puadi Hamzani	+628298812729	puadihamzani@gmail.com	Officer	
9	[User Icon]	Regina Putri Anjani	+6285236108839	reginaputri@gmail.com	Officer	
10	[User Icon]	Zalwan	+6285640435583	Zalwan@gmail.com	Officer	

Pada ***employee management*** ini ***admin*** dapat mengupdate foto ***profile*** dari setiap ***employee*** yang ada contohnya kita dapat mengubah foto ***employee*** zalwan, dengan cara menekan button () ini, kemudian sistem akan memanggil ***function employeeEditPhotos()***.

```
public function employeeEditPhotos($ID)
{
    $where = array('ID' => $ID);
    $data['employee'] = $this->m_admin-
>EditAllData($where, 'pm_master_employee');

    $this->load->view('resource/admin/header');
    $this->load->view('resource//admin/sidebar');
    $this->load-
>view('admin/employee/employeeEditPhotos', $data);
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
}
```

Kemudian setelah menekan button tersebut C_admin pun memanggil file  view ***EmployeeEditPhotos.php*** pada folder ***admin/employee***

EmployeeEditPhotos.php

```
<div class="page-wrapper">
    <div class="container-fluid">
        <div class="row page-titles">
            <div class="col-md-6 col-8 align-self-
center">
                <h3 class="text-themecolor mb-0 mt-
0">Preventive Maintenance</h3>
                <ol class="breadcrumb">
```

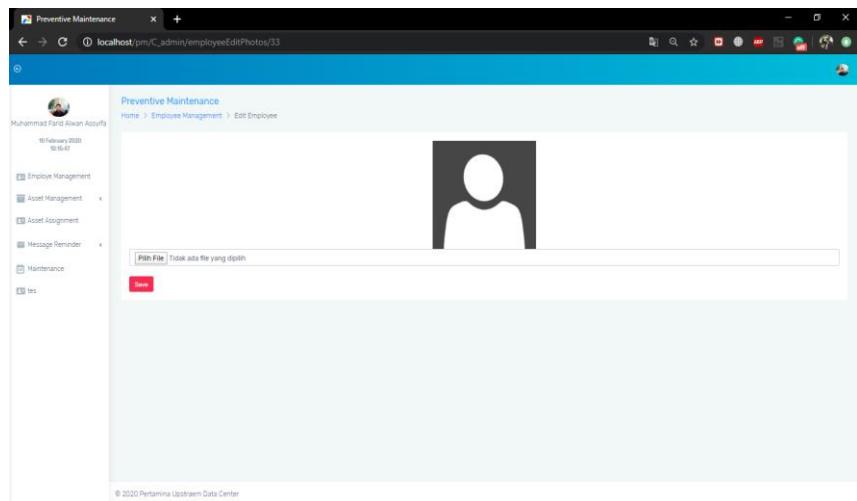
```
<li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin>Home</a></li>
                <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin/employeeManagement">Employee Management</a></li>
                    <li class="breadcrumb-item">Edit Employee</li>
                    <!-- <li class=" breadcrumb-item active">Asset Assignment</li> -->
                </ol>
            </div>
        </div>

        <div class="row">
            <div class="col-12">
                <div class="card">
                    <div class="card-body">
                        <form action=<?php echo base_url(); ?>C_admin/employeePhotosUpdate" method="post" enctype="multipart/form-data">
                            <?php
                            foreach ($employee as $row)
                            {
                                ?>
                            <div class="form-group">
                                <center>
                                    
                                </center>
                                <!--
                                - <label class="control-label">Image</label> -->
                                <input type="hidden"
                            " class="form-
```

```
control" name="ID" required value=<?php echo $row['ID'];
]; ?>>
                                <input type="file"
class="form-
control" name="IMAGE" value=<?php echo $row['IMAGE'];
?>>
                                </div>

                                <button type="submit" c
lass="btn btn-danger waves-effect waves-
light">Save</button>
                                <?php } ?>
                            </form>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
```

Maka akan tampil halaman seperti di bawah ini ketika menekan *button* () pada employee Zalwan.

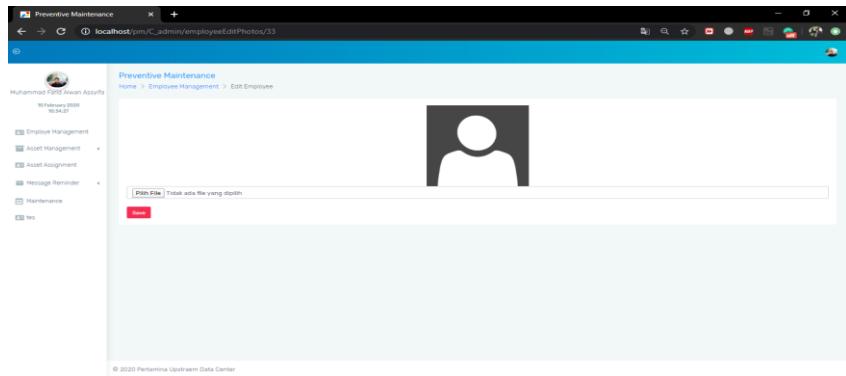


kemudian pilih foto yang akan di upload dan tekan button save sehingga. Ketika button save di tekan maka sistem akan memanggil **function employeePhotosUpdate()**

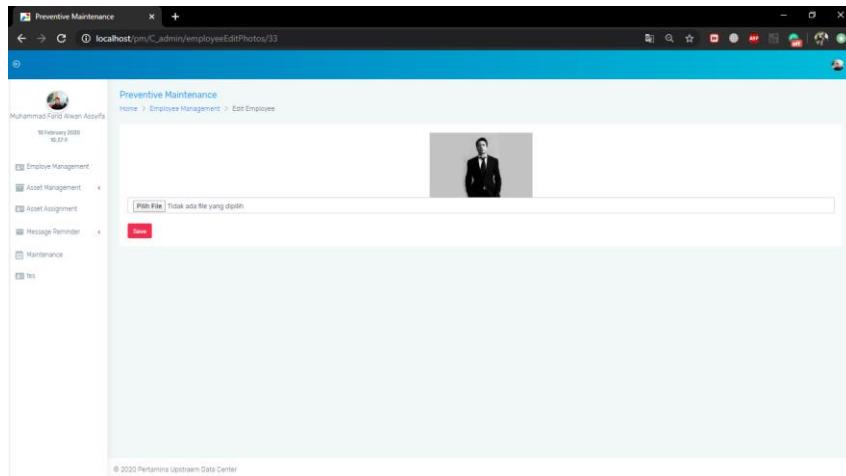
```
public function employeePhotosUpdate()
{
    $data = array(
        #database
        'IMAGE' => $this->input->post('IMAGE'),
    );
    if (!empty($_FILES['IMAGE']['name'])) {
        $upload = $this->_do_upload();
        $data['IMAGE'] = $upload;
    }
    $where = array(
        'ID' => $this->input->post('ID')
    );
    // print_r($data);die;
    $this->m_admin->UpdateAllData($where, $data, 'pm_master_employee');
    redirect('C_admin/employeeManagement');
}
```

Kemudian akan mengupdate foro baru yang telah di pilih sebelumnya dana akanter update dengan foto terbaru

Before



After



Kemudian kita dapat menekan *button* (✎) ini pada employee zalwan sehingga akan akan memanggil *function employeeEdit()* pada **C_Admin**. Kemudian sistem akan

menampilkan data yang akan diedit sesuai dengan id zalwan dikarenakan button yang ditekan adalah milik zalwan.

```
public function employeeEdit($ID)
{
    $where = array('ID' => $ID);
    $data['employee'] = $this->m_admin-
>EditAllData($where, 'pm_master_employee');

    $this->load->view('resource/admin/header');
    $this->load->view('resource//admin/sidebar');
    $this->load-
>view('admin/employee/employeeEdit', $data);
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
}
```

Pada function ini juga memanggil model m_Admin **function EditAllData()** untuk mengambil data *employee by ID_Employee*

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

Kemudian data tersebut akan di parsing pada view *employeeEdit.php* pada folder *admin/employee/*.

employeeEdit.php

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

```

        <div class="row page-titles">
            <div class="col-md-6 col-8 align-self-center">
                <h3 class="text-themecolor mb-0 mt-0">Preventive Maintenance</h3>
                <ol class="breadcrumb">
                    <li class="breadcrumb-item"><a href="php echo base_url(); ?&gt;C_admin"&gt;Home&lt;/a&gt;&lt;/li&gt;
                    &lt;li class="breadcrumb-item"&gt;&lt;a href="<?php echo base_url(); ?&gt;C_admin/employeeManagement"&gt;Employee Management&lt;/a&gt;&lt;/li&gt;
                    &lt;li class="breadcrumb-item"&gt;Edit Employee&lt;/li&gt;
                    &lt;!-- &lt;li class=" breadcrumb-item active"&gt;Asset Assignment&lt;/li&gt; --&gt;
                &lt;/ol&gt;
            &lt;/div&gt;
        &lt;/div&gt;

        &lt;div class="row"&gt;
            &lt;div class="col-12"&gt;
                &lt;div class="card"&gt;
                    &lt;div class="card-body"&gt;
                        &lt;form action="<?php echo base_url(); ?&gt;C_admin/employeeUpdate" method="post" enctype="multipart/form-data"&gt;
                            &lt;?php
                            foreach ($employee as $row)
                            {
                                ?&gt;
                            &lt;div class="form-group"&gt;
                                &lt;center&gt;
                                    &lt;img src="<?php echo base_url(); ?&gt;assets/uploads/&lt;?php echo $row['IMAGE']; ?&gt;" style="width:250px; height:80" /&gt;
                                &lt;/center&gt;
                            &lt;/div&gt;
                        &lt;/form&gt;
                    &lt;/div&gt;
                &lt;/div&gt;
            &lt;/div&gt;
        &lt;/div&gt;
    </pre

```

```
< ! -  
- <label class="control-label">Image</label> -->  
  
                                <input type="hidden"  
" class="form-  
control" name="ID" required value="<?php echo $row['ID'  
]; ?>">  
                                < ! -  
- <input type="file" class="form-  
control" name="IMAGE" value="<?php echo $row['IMAGE'];  
?>"> -->  
                                </div>  
<div class="form-  
group">  
                                <label class="contr  
ol-label">Name</label>  
                                <input type="text"  
class="form-  
control" name="NAME" required value="<?php echo $row['N  
AME']; ?>">  
                                </div>  
<div class="form-  
group">  
                                <label class="contr  
ol-label">Date Of Birth</label>  
                                <input type="text"  
class="form-  
control" name="DATE_OF_BIRTH" required value="<?php ech  
o $row['DATE_OF_BIRTH']; ?>">  
                                </div>  
<div class="form-  
group">  
                                <label class="contr  
ol-label">Phone Number</label>  
                                <input type="text"  
class="form-  
control" name="PHONE_NUMBER" required value="<?php echo  
$row['PHONE_NUMBER']; ?>">  
                                </div>
```

```
<div class="form-group">
    <label class="control-label">Email</label>
    <input type="email" class="form-control" name="EMAIL" required value=<?php echo $row['EMAIL']; ?>>
</div>
<div class="form-group">
    <label class="control-label">Username</label>
    <input type="text" class="form-control" name="USERNAME" required value=<?php echo $row['USERNAME']; ?>>
</div>
<div class="form-group">
    <label class="control-label">Password</label>
    <input type="password" class="form-control" name="PASSWORD" required value=<?php echo $row['PASSWORD']; ?>>
</div>
<div class="form-group">
    <label>ACCESS</label>
    <select class="form-control custom-select" name="ACCESS" required>
        <!--
        - <option value="">CHOOSE ACCESS USER</option> -->
        <?php if ($row['ACCESS'] == 1) { ?>
            <option value="1">ADMIN</option>
        <?php } ?>
    </select>
</div>
```

```
<option val
ue="2">OFICER</option>

        <?php } else {
?>
        <option val
ue="2">OFFICER</option>
        <option val
ue="1">ADMIN</option>
        <?php } ?>
        </select>
    </div>
    <div class="form-
group">
        <label>Active</labe
l>
        <select class="form
-control custom-select" name="ACTIVE" required>
            <?php if ($row[
'ACTIVE'] == 'Y') { ?>
            <option val
ue="Y">ACTIVE</option>
            <option val
ue="N">DEACTIVE</option>
            <?php } else {
?>
            <option val
ue="N">DEACTIVE</option>
            <option val
ue="Y">ACTIVE</option>
            <?php } ?>
            </select>
        </div>
        <button type="submit" c
lass="btn btn-danger waves-effect waves-
light">Save</button>
        <?php } ?>
```

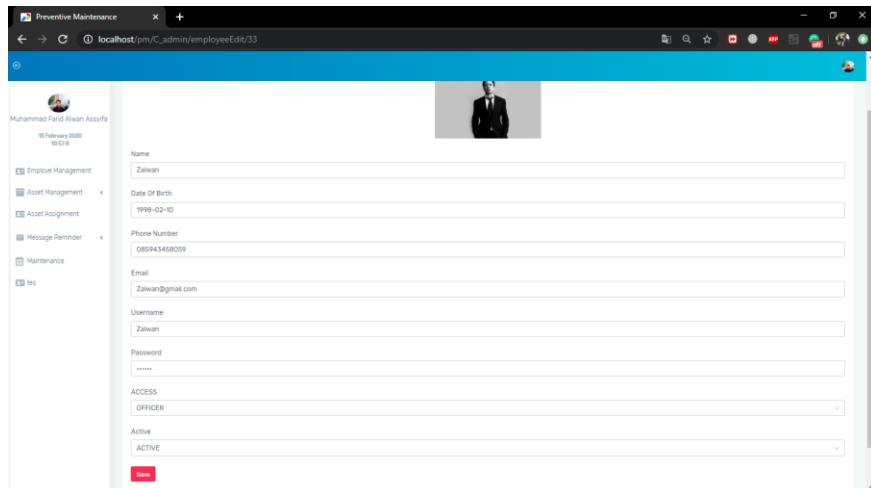
```

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

</div>
</div>

```

Maka sistem akan menampilkan halaman edit employee atas nama zalwan seperti gambar di bawah ini.



Kemudian kita ganti name dari zalwan menjadi “zalwan 123” kemudian tekan save pada form tersebut. Ketika button save di tekan maka sistem akan memanggil function ***employeeUpdate()***

```

public function employeeUpdate()
{
    $data = array(
        #database
        'NAME' => $this->input->post('NAME'),

```

```

        'DATE_OF_BIRTH' => $this->input-
>post('DATE_OF_BIRTH'),
        'EMAIL' => $this->input->post('EMAIL'),
        'PHONE_NUMBER' => $this->input-
>post('PHONE_NUMBER'),
        'USERNAME' => $this->input-
>post('USERNAME'),
        'PASSWORD' => $this->input-
>post('PASSWORD'),
        'ACCESS' => $this->input->post('ACCESS'),
        'ACTIVE' => $this->input->post('ACTIVE'),
        // 'IMAGE' => $this->input->post('IMAGE'),
    );
    // if (!empty($_FILES['IMAGE']['name'])) {
    //     $upload = $this->_do_upload();
    //     $data['IMAGE'] = $upload;
    // }
    $where = array(
        'ID' => $this->input->post('ID')
    );
    // print_r($data);die;
    $this->m_admin-
>UpdateAllData($where, $data, 'pm_master_employee');
    redirect('C_admin/employeeManagement');
}

```

Kemudian function ***employeeUpdate()*** akan mengirim hasil inputan tersebut, dimana akan dikirimkannya data yang telah di inputkan saat update pada halaman sebelumnya menuju model *m_Admin* pada ***function UpdateAllData()***.

```

function UpdateAllData($where, $data, $table)
{
    $this->db->where($where);
    $this->db->update($table, $data);
}

```

Before Update

10		Zalwan	085943458059	Zalwan@gmail.com
----	-----------------------------------------------------------------------------------	--------	--------------	------------------

After Update

10		Zalwan123	085943458059	Zalwan@gmail.com
----	-----------------------------------------------------------------------------------	-----------	--------------	------------------

Setelah itu kita dapat menekan button () pada **employee zalwan123** maka sistem akan memanggil **function employeeDelete()** pada C_admin dan juga memarsing data pada model m_admin pada **function DeleteAllData()**.

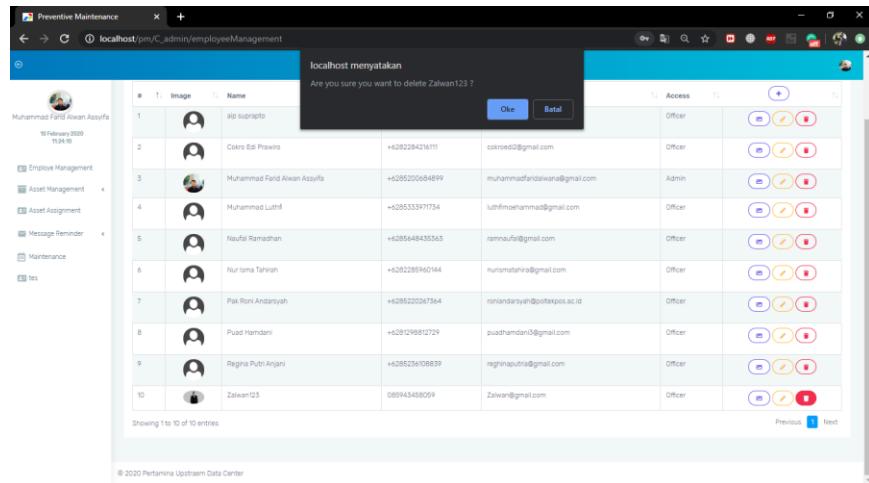
C_admin-employeeDelete()

```
public function employeeDelete($ID)
{
    $where = array('ID' => $ID);
    $this->m_admin-
>DeleteAllData($where, 'pm_master_employee');
    redirect('C_admin/employeeManagement');
}
```

M_admin-DeleteAllData()

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

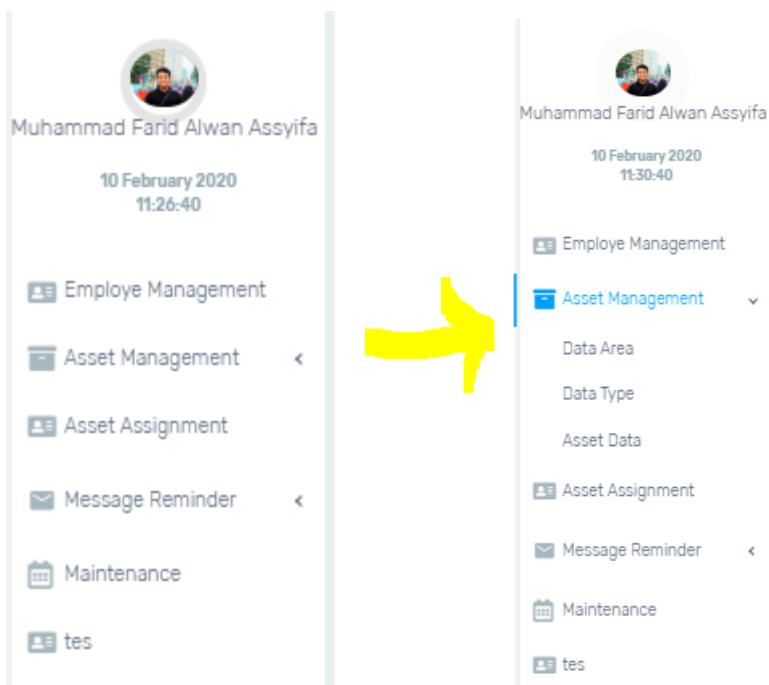
Kemudian sistem akan menampilkan konfirmasi pada saat akan menghapus data **employee zalwan123** seperti gambar dibawah ini .



Jika menekan oke pada alert yang muncul maka data **employee zalwan123** akan terhapus dari sistem.

5.4.2 Pembuatan Halaman Asset Management

Kemudian kita dapat menekan menu asset management, maka akan memunculkan 3 sub menu yaitu data area, data type dan asset data.



Untuk menampilkan menu tampilan asset management seperti di atas maka script nya seperti dibawah ini.

```
<li>
<a class="has-arrow" href="#" aria-
expanded="false"><i class="fas fa-
archive"></i><span class="hide-
menu">Asset Management</span></a>
<ul aria-expanded="false" class="collapse">
```

```
        <li><a href="?>C\_Admin/area">Data Area</a></li>
        <li><a href="?>C\\_Admin/datatype">Data Type</a></li>
        <li><a href="?>C\\\_Admin/assetdata">Asset Data</a></li>
        </ul>
    </li>
```

5.4.2.1 Pembuatan Halaman Data Area

Kemudian ketika memilih sub menu data area maka sistem akan memanggil ***function area()*** pada ***controller C_admin*** untuk menampilkan Data Area pada sistem.

C_Admin-area()

```
public function area()
{
    $data = array(
        'area' => $this->m_admin-
>getAllData('pm_master_area')
    );

    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
    $this->load-
>view('admin/assetManagement/area', $data);
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
}
```

Pada function ini juga memanggil ***function getAllData()*** dari ***model m_Admin*** untuk menampilkan data dari database ke view ***area*** pada folder ***admin/assetManagement/***

M_admin-getAllData()

```
public function getAllData($table)
{
    return $this->db->get($table)->result_array();
}
```

Area.php

```
<div class="page-wrapper">
    <div class="container-fluid">
        <div class="row page-titles">
            <div class="col-md-6 col-8 align-self-center">
                <h3 class="text-themecolor mb-0 mt-0">Preventive Maintenance</h3>
                <ol class="breadcrumb">
                    <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin">Home</a></li>
                    <li class="breadcrumb-item"><a href="#">Asset Management</a></li>
                    <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin/area">Area Data</a></li>
                        <!-- <li class="breadcrumb-item active">Asset Assignment</li> -->
                </ol>
            </div>
        </div>
        <div class="row">
```

```
<div class="col-12">
    <div class="card">
        <div class="card-body">
            <div class="table-responsive m-
t-40">
                <table id="myTable" class="table table-bordered table-striped">

                    <thead>

                        <tr>

                            <th>#</th>

                            <th>Name</th>

                            <th>

                                <center>

                                    <a href
="<?php echo base_url(); ?>C_admin/areaForm" class="btn
btn-outline-primary btn-rounded float-
center"><i class="fas fa-plus"></i></a>

                                </center>
                            </th>

                        </tr>
                    </thead>

                    <tbody>

                        <?php
                            $no = 1;

```

```
        foreach ($area as $  
row) {  
            ?>  
            <tr>  
                <td><?php e  
cho $no++ ?></td>  
                <td><?php e  
cho $row[ 'AREA_NAME' ] ; ?></td>  
                <td>  
                    <center  
>  
                        <a  
                            href="<?php echo base_url('C_admin/areaEdit/' . $row['I  
D']); ?>" class="btn btn-outline-warning btn-  
rounded"><i class="fas fa-pencil-alt"></i></a>  
                        <a  
                            href="<?php echo base_url('C_admin/areaDelete/' . $row[  
'ID']); ?>" class="btn btn-outline-danger btn-  
rounded" onclick="return confirm('Are you sure you want  
to delete <?php echo $row['AREA_NAME'] ?> ?')><i clas  
s="fas fa-trash"></i></a>  
                    </center>  
                </td>  
            </tr>  
        <?php } ?>  
    </tbody>  
    </table>  
    </div>  
    </div>  
    </div>  
    </div>
```

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

Setelah sistem selesai memanggil dan memproses function `area()` pada `C_Admin` maka sistem akan menampilkan halaman seperti di bawah ini.

#	Name	Action
1	LANTAI 1 - PUDC	(+)
2	LANTAI 2 - PUDC	(+)
3	Warehouse	(+)

Kemudian silahkan tekan button (+) maka sistem akan memanggil `function areaForm()`.

`C_admin-areaForm()`

```
public function areaForm()
{
    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
```

```
        $this->load-
>view('admin/assetManagement/areaForm');
        $this->load->view('resource/admin/footer');
        $this->load->view('style/style');
    }
```

Setelah memanggil ***function areaForm()*** maka sistem akan menampilkan halaman ***areaForm*** dari view dalam folder ***admin/assetManagement/areaForm***

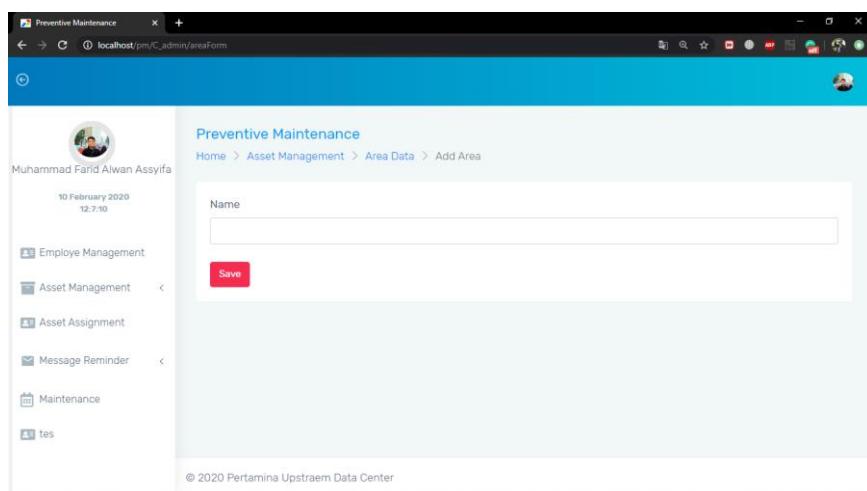
areaForm.php

```
<div class="page-wrapper">
    <div class="container-fluid">
        <div class="row page-titles">
            <div class="col-md-6 col-8 align-self-center">
                <h3 class="text-themecolor mb-0 mt-0">Preventive Maintenance</h3>
                <ol class="breadcrumb">
                    <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin">Home</a></li>
                    <li class="breadcrumb-item"><a href="#">Asset Management</a></li>
                    <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin/area">Area Data</a></li>
                    <li class="breadcrumb-item active">Add Area</li>
                    <!-- <li class="breadcrumb-item active">Asset Assignment</li> -->
                </ol>
            </div>
        </div>
```

```

<div class="row">
    <div class="col-12">
        <div class="card">
            <div class="card-body">
                <form action=<?php echo base_u
rl(); ?>C_admin/addArea" method="post">
                    <div class="form-group">
                        <label class="control-
label">Name</label>
                        <input type="text" clas
s="form-control" name="AREA_NAME" required>
                    </div>
                    <button type="submit" class
="btn btn-danger waves-effect waves-
light">Save</button>
                </form>
            </div>
        </div>
    </div>
</div>

```



Kemudian kita inputkan data misalnya kita ketikan Gedung Utama pada textbos tersebut dan menekan *button save*, ketika *button save* ditekan maka sistem akan memanggil ***function addArea()*** pada ***Controller C_Admin***. ***C_Admin*** akan mengirim data yang telah diinputkan ke model ***m_admin*** pada ***function InsertData()***

C_admin-addArea()

```
public function addArea()
{
    $data = array(
        'AREA_NAME' => $this->input-
>post('AREA_NAME'),
    );
    $this->m_admin-
>InsertData('pm_master_area', $data);
    redirect("C_admin/area");
}
```

m_admin-InsertData()

```
function InsertData($table, $data)
{
    $this->db->insert($table, $data);
}
```

Setelah data selesai diproses maka sistem akan secara otomatis akan kembali pada halaman data area dan anda bisa melihat data yang baru diinputkan tadi pada halaman data area

#	Name	
1	LANTAI 1 - PUDC	
2	LANTAI 2 - PUDC	
3	Warehouse	
4	Gedung Utama	

Kemudian pilih button () pada data Gedung Utama, kemudian sistem akan memanggil ***function areaEdit()*** dari ***Controller C_Admin.*** ***C_admin*** akan memanggil function EditAllData() pada model m_admin untuk menampilkan data Gedung Utama.

C_Admin-areaRdit()

```
public function areaEdit($ID)
{
    $where = array('ID' => $ID);
    $data['area'] = $this->m_admin-
>EditAllData($where, 'pm_master_area');

    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
```

```

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

        $this->load->view('resource/admin/footer');
        $this->load->view('style/style');
    }

m_admin>EditAllData()

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

```

C_admin akan memarsing data dari *model m_Admin* tersebut menuju *view areaEdit* pada folder *admin/assetManagement/*

admin/assetManagement/areaEdit.php

```

<div class="page-wrapper">
    <div class="container-fluid">
        <div class="row page-titles">
            <div class="col-md-6 col-8 align-self-center">
                <h3 class="text-themecolor mb-0 mt-0">Preventive Maintenance</h3>
                <ol class="breadcrumb">
                    <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin">Home</a></li>
                    <li class="breadcrumb-item"><a href="#">Asset Management</a></li>
                    <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin/area">Area Data</a></li>
                    <li class="breadcrumb-item active">Area Edit</li>
                
```

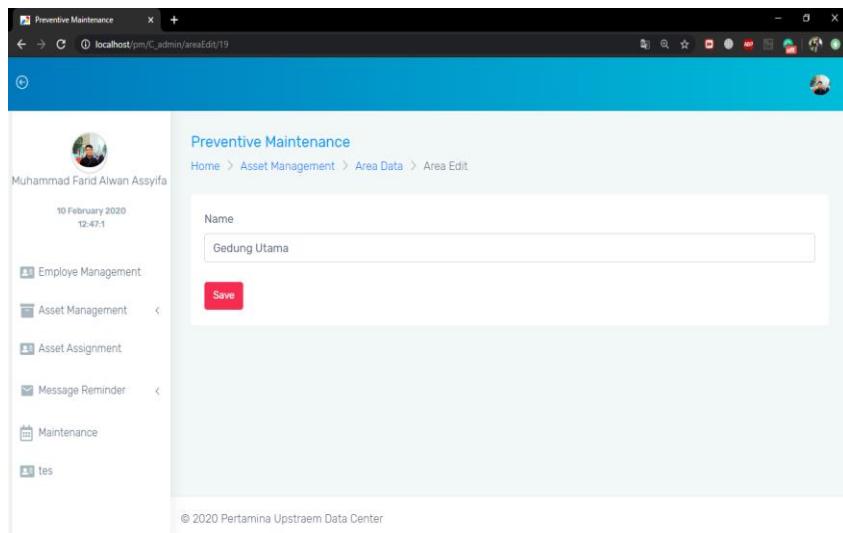
```
        <!-- <li class="breadcrumb-item active">Asset Assignment</li> -->
            </ol>
        </div>
    </div>

    <div class="row">
        <div class="col-12">
            <div class="card">
                <div class="card-body">
                    <?php
                    foreach ($area as $row) {
                        ?>
                        <form action=<?php echo base_url(); ?>C_admin/areaUpdate" method="post">
                            <div class="form-group">
                                <label class="control-label">Name</label>
                                <input type="text" class="form-control" name="AREA_NAME" value=<?php echo $row['AREA_NAME']; ?>" required>
                                <input type="hidden" class="form-control" name="ID" value=<?php echo $row['ID'] ?>" required>
                            </div>
                            <button type="submit" class="btn btn-danger waves-effect waves-light" onclick="return confirm('Are you sure you want to update this data ?')">Save</button>
                        </form>
                    <?php } ?>
                </div>
            </div>
```

```
        </div>
    </div>

    </div>
</div>
```

Apabila sistem sudah selesai memproses data dari C_Admin/ areaEdit() maka sistem akan menampilkan halaman tersebut sesuai dengan data yang akan di edit.



kemudian kita inputkan data terbaru untuk meng update data tersebut misalkan dari '**Gedung Utama**' menjadi '**Gedung Utama PUDC**' kemudian tekan button save. Ketika button save ditekan maka sistem akan memanggil function areaUpdate() pada controller C_Admin.

```
public function areaUpdate()
{
```

```
$data = array(
    #database
    'AREA_NAME' => $this->input-
>post('AREA_NAME'),
);
$where = array(
    'ID' => $this->input->post('ID')
);
// print_r($data);die;
$this->m_admin-
>UpdateAllData($where, $data, 'pm_master_area');
redirect('c_admin/area');
```

Kemudian pada areaUpdate ini data akan di parsing ke model m_Admin pada function UpdateAllData(). Setelah data berhasil di parsing kedalam UpdateAllData maka dia akan otomatis kembali kehalaman awal area

m_admin-UpdateAllData()

```
function UpdateAllData($where, $data, $table)
{
    $this->db->where($where);
    $this->db->update($table, $data);

}
```

Preventive Maintenance																									
localhost/pm/C_admin/area																									
Muhammad Farid Alwan Assyifa																									
10 February 2020 12:53:57																									
<table border="1"> <thead> <tr> <th>Show</th> <th>10</th> <th>entries</th> <th>Search:</th> </tr> </thead> <tbody> <tr> <td>#</td> <td>↑</td> <td>Name</td> <td>↑↓</td> </tr> <tr> <td>1</td> <td></td> <td>LANTAI 1 - PUDC</td> <td></td> </tr> <tr> <td>2</td> <td></td> <td>LANTAI 2 - PUDC</td> <td></td> </tr> <tr> <td>3</td> <td></td> <td>Warehouse</td> <td></td> </tr> <tr> <td>4</td> <td></td> <td>Gedung Utama PUDC</td> <td></td> </tr> </tbody> </table>		Show	10	entries	Search:	#	↑	Name	↑↓	1		LANTAI 1 - PUDC		2		LANTAI 2 - PUDC		3		Warehouse		4		Gedung Utama PUDC	
Show	10	entries	Search:																						
#	↑	Name	↑↓																						
1		LANTAI 1 - PUDC																							
2		LANTAI 2 - PUDC																							
3		Warehouse																							
4		Gedung Utama PUDC																							
<p>Showing 1 to 4 of 4 entries</p>																									
<p>Previous </p>																									
<p></p>																									

kemudian tekan button () pada data Gedung Utama PUDC maka sistem akan akan memanggil ***function areaDelete()*** pada C_admin dan juga memarsing data pada model m_admin pada ***function DeleteAllData()***.

C_admin-areaDelete()

```
public function areaDelete($ID)
{
    $where = array('ID' => $ID);
    $this->m_admin-
>DeleteAllData($where, 'pm_master_area');
    redirect('C_admin/area');
}
```

M_admin-DeleteAllData()

```
function DeleteAllData($where, $table)
{
    $this->db->where($where);

    $this->db->delete($table);
```

```
}
```

Kemudian sistem akan menampilkan konfirmasi pada saat akan menghapus data **Gedung Utama PUDC** seperti gambar dibawah ini .

The screenshot shows a web browser window titled "Preventive Maintenance" with the URL "localhost:pm/C_admin/area". A confirmation dialog box is displayed in the center, asking "localhost menyatakan Are you sure you want to delete Gedung Utama PUDC ?" with "Oke" and "Batal" buttons. Below the dialog, the main content area shows a table of asset data. The table has columns for #, Name, and actions (edit and delete). The data is as follows:

#	Name	
1	LANTAI 1 - PUDC	
2	LANTAI 2 - PUDC	
3	Warehouse	
4	Gedung Utama PUDC	

At the bottom of the table, it says "Showing 1 to 4 of 4 entries".

Jika menekan oke pada alert yang muncul maka data **Gedung Utama PUDC** akan terhapus dari sistem.

5.4.2.2 Pembuatan Halaman Data Type

Kemudian ketika memilih sub menu data type maka sistem akan memanggil **function dataType()** pada **controller C_admin** untuk menampilkan Data Type pada sistem.

C_Admin-dataType()

```
public function datatype()
{
    $keyword      =  $this->input->post('keyword');
    $tb = 'pm_type_asset';
    $data = array(
        'type' => $this->m_admin-
>getDataType($tb, $keyword)
    );
    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
    $this->load-
>view('admin/assetManagement/dataType', $data);
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
}
```

Pada function ini juga memanggil **function getDataType()** dari **model m_Admin** untuk menampilkan data dari database ke view **area** pada folder **admin/dataType/**

M_admin-getDataType()

```
public function getDataType($table, $keyword)
{
    $this->db->like('TYPE_NAME', $keyword);
    $this->db->OR_LIKE('ID', $keyword);
    return $this->db->get($table)->result_array();
}
```

dataType.php

```

<th>
    <center>
        <a href
="<?php echo base_url(); ?>C_admin/dataTypeForm" class=
"btn btn-outline-primary btn-rounded"><i class="fas fa-
plus"></i></a>
    </center>
</th>
</tr>
</thead>
<tbody>
<?php
$no = 1;
foreach ($type as $row) {
    ?>
<tr>
    <td><?php e
cho $no++ ?></td>
    <td><?php e
cho $row['TYPE_NAME']; ?></td>
    <td>
        <center
>
            <a
href="<?php echo base_url('C_admin/dataTypeEdit/' . $ro
w['ID']); ?>" class="btn btn-outline-warning btn-
rounded"><i class="fas fa-pencil-alt"></i></a>
            <a
href="<?php echo base_url('C_admin/dataTypeDelete/' . $
row['ID']); ?>" class="btn btn-outline-danger btn-
rounded" onclick="return confirm('Are you sure you want
to delete <?php echo $row['TYPE_NAME'] ?> ?')"><i clas
s="fas fa-trash"></i></a>
        </center>
    </td>
</tr>
<?php } ?>

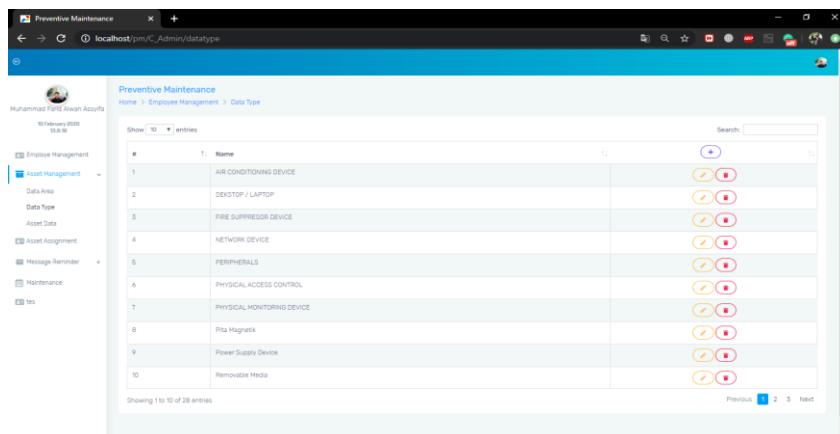
```

```
        </tbody>
    </table>
</div>
</div>
</div>

</div>
</div>

</div>
</div>
```

Setelah sistem selesai memanggil dan memproses function dataType() pada C_Admin maka sistem akan menampilkan halaman seperti di bawah ini.



The screenshot shows a web browser window titled "Preventive Maintenance" with the URL "localhost:pm/C_Admin/datatype". The page displays a table of data types, with the first 10 entries listed:

#	Name	Action Buttons
1	AIR CONDITIONING DEVICE	[Edit] [Delete]
2	DESKTOP / LAPTOP	[Edit] [Delete]
3	FIRE SUPPRESSOR DEVICE	[Edit] [Delete]
4	NETWORK DEVICE	[Edit] [Delete]
5	PERIPHERALS	[Edit] [Delete]
6	PHYSICAL ACCESS CONTROL	[Edit] [Delete]
7	PHYSICAL MONITORING DEVICE	[Edit] [Delete]
8	Pita Magnetik	[Edit] [Delete]
9	Power Supply Device	[Edit] [Delete]
10	Removable Media	[Edit] [Delete]

At the bottom of the table, it says "Showing 1 to 10 of 28 entries". Below the table, there are navigation links: "Previous" (disabled), "1", "2", "3", and "Next".

Kemudian silahkan tekan *button* (+) maka sistem akan memanggil ***function areaForm()***.

C_amin-dataTypeForm()

```
public function dataTypeForm()
{
    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
    $this->load-
>view('admin/assetManagement/dataTypeForm');
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
}
```

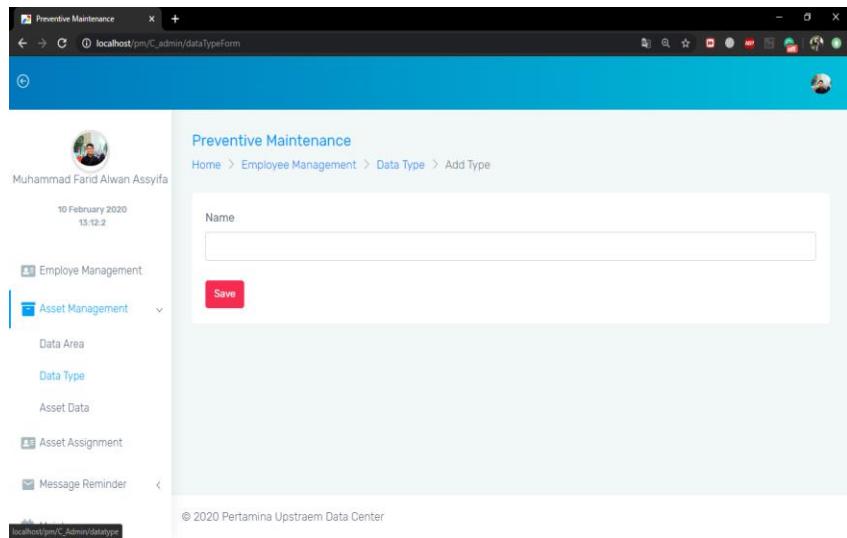
Setelah memanggil ***function dataTypeForm()*** maka sistem akan menampilkan halaman ***dataTypeForm*** dari view dalam folder ***admin/assetManagement/dataTypeForm.php***

dataTypeForm.php

```
<div class="page-wrapper">
    <div class="container-fluid">
        <div class="row page-titles">
            <div class="col-md-6 col-8 align-self-
center">
                <h3 class="text-themecolor mb-0 mt-
0">Preventive Maintenance</h3>
                <ol class="breadcrumb">
                    <li class="breadcrumb-
item"><a href="=base_url(); ?&gt;C_admin"&gt;Home&lt;/
a&gt;&lt;/li&gt;
                    &lt;li class="breadcrumb-
item"&gt;&lt;a href="#"&gt;Employee Management&lt;/a&gt;&lt;/li&gt;</pre
```

```
        <li class="breadcrumb-item"><a href="=base_url('C_admin/dataType')?&gt;Data Type&lt;/a&gt;&lt;/li&gt;
            &lt;li class="breadcrumb-item active"&gt;Add Type&lt;/li&gt;
                &lt;!-- &lt;li class="breadcrumb-item active"&gt;Asset Assignment&lt;/li&gt; --&gt;
            &lt;/ol&gt;
        &lt;/div&gt;
    &lt;/div&gt;

    &lt;div class="row"&gt;
        &lt;div class="col-12"&gt;
            &lt;div class="card"&gt;
                &lt;div class="card-body"&gt;
                    &lt;form action="<?=base_url('C_admin/adddataType')?&gt;" method="post"&gt;
                        &lt;div class="form-group"&gt;
                            &lt;label class="control-label"&gt;Name&lt;/label&gt;
                            &lt;input type="text" class="form-control" name="TYPE_NAME" required&gt;
                        &lt;/div&gt;
                        &lt;button type="submit" class="btn btn-danger waves-effect waves-light"&gt;Save&lt;/button&gt;
                    &lt;/form&gt;
                &lt;/div&gt;
            &lt;/div&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;</pre
```



Kemudian kita inputkan data misalnya kita ketikan **APAR 1** pada textbos tersebut dan menekan *button save*, ketika *button save* ditekan maka sistem akan memanggil **function adddataType()** pada **Controller C_Admin** . **C_Admin** akan mengirim data yang telah diinputkan ke model **m_admin** pada **function InsertData()**

C_admin-adddataType()

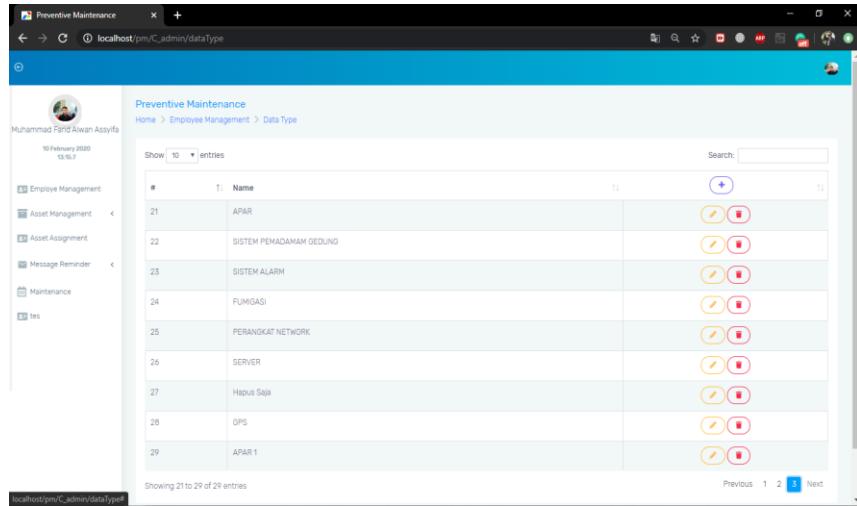
```
public function adddataType()
{
    $data = array(
        'TYPE_NAME' => $this->input-
>post('TYPE_NAME'),
    );
    $this->m_admin-
>InsertData('pm_type_asset', $data);

    redirect("C_admin/dataType");
}
```

m_admin-InsertData()

```
function InsertData($table, $data)
{
    $this->db->insert($table, $data);
}
```

Setelah data selesai diproses maka sistem akan secara otomatis akan kembali pada halaman data type dan anda bisa melihat data yang baru diinputkan tadi pada halaman data type



#	Name	Action	Action
21	APAR		
22	SISTEM PEMADAMAN GEDUNG		
23	SISTEM ALARM		
24	FUMIGASI		
25	PERANGKAT NETWORK		
26	SERVER		
27	Hapus Saja		
28	GPS		
29	APAR 1		

Kemudian pilih button () pada data **APAR 1**, kemudian sistem akan memanggil **function dataTypeEdit()** dari **Controller C_Admin**. **C_admin** akan memanggil **function EditAlldata()** pada model m_admin untuk menampilkan data APAR 1.

C_Admin-dataTypeEdit()

```
public function dataTypeEdit($ID)
{
    $where = array('ID' => $ID);
    $data['type'] = $this->m_admin-
>EditAllData($where, 'pm_type_asset');

    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
    $this->load-
>view('admin/assetManagement/dataTypeEdit', $data);
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
}
```

m_admin>EditAllData()

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

C_admin akan memarsing data dari ***model m_Admin*** tersebut menuju ***view dataTypeEdit*** pada folder ***admin/assetManagement/***

admin/assetManagement/dataTypeEdit.php

```
<div class="page-wrapper">
    <div class="container-fluid">
        <div class="row page-titles">
            <div class="col-md-6 col-8 align-self-
center">
                <h3 class="text-themecolor mb-0 mt-
0">Preventive Maintenance</h3>
                <ol class="breadcrumb">
```

```
        <li class="breadcrumb-item"><a href="=base_url(); ?&gt;C_admin"&gt;Home&lt;/a&gt;&lt;/li&gt;
            &lt;li class="breadcrumb-item"&gt;&lt;a href="#"&gt;Employee Management&lt;/a&gt;&lt;/li&gt;
                &lt;li class="breadcrumb-item"&gt;&lt;a href="<?=base_url(); ?&gt;C_admin/dataType"&gt;Data Type&lt;/a&gt;&lt;/li&gt;
                    &lt;li class="breadcrumb-item active"&gt;Edit Type&lt;/li&gt;
                    &lt;!-- &lt;li class="breadcrumb-item active"&gt;Asset Assignment&lt;/li&gt; --&gt;
                &lt;/ol&gt;
            &lt;/div&gt;
        &lt;/div&gt;

        &lt;div class="row"&gt;
            &lt;div class="col-12"&gt;
                &lt;div class="card"&gt;
                    &lt;div class="card-body"&gt;
                        &lt;?php
                            foreach ($type as $row) {
                                ?
                                &lt;form action="<?=base_url(); ?&gt;C_admin/dataTypeUpdate" method="post"&gt;
                                    &lt;div class="form-group"&gt;
                                        &lt;label class="control-label"&gt;Name&lt;/label&gt;
                                        &lt;input type="text" class="form-control" name="TYPE_NAME" value="<?=echo $row['TYPE_NAME']?" required>
                                        <input type="hidden" class="form-control" name="ID" value="=echo $row['ID']?" required>
                                    </div>
                            
```

```

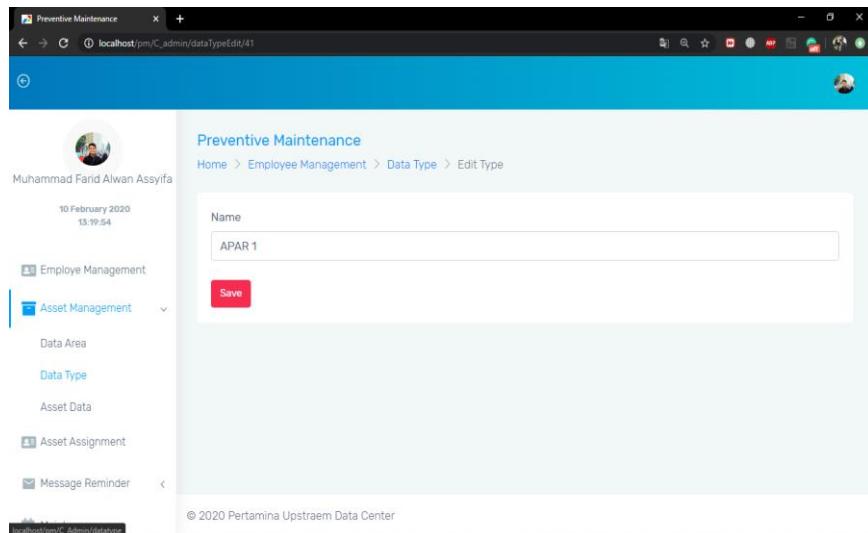
        <button type="submit" c
lass="btn btn-danger waves-effect waves-
light" onclick="return confirm('Are you sure you want t
o update this data ?')">Save</button>
        </form>
        <?php } ?>
    </div>

    </div>
</div>

</div>
</div>

```

Apabila sistem sudah selesai memproses data dari **C_Admin/dataTypeEdit()** maka sistem akan menampilkan halaman tersebut sesuai dengan data yang akan di edit.



kemudian kita inputkan data terbaru untuk meng update data tersebut misalkan dari ‘APAR 1’ menjadi ‘APAR UTAMA’ kemudian tekan *button save*. Ketika *button save* ditekan maka sistem akan memanggil **function dataTypeUpdate()** pada **controller C_Admin**.

```
public function dataTypeUpdate()
{
    $data = array(
        #database
        'TYPE_NAME' => $this->input-
>post('TYPE_NAME'),
    );
    $where = array(
        'ID' => $this->input->post('ID')
    );
    // print_r($data);die;
    $this->m_admin-
>UpdateAllData($where, $data, 'pm_type_asset');
    redirect('C_admin/dataType');
}
```

Kemudian pada areaUpdate ini data akan di parsing ke model **m_Admin** pada function **UpdateAllData()**. Setelah data berhasil di parsing kedalam **UpdateAllData** maka dia akan otomatis kembali ke halaman awal data Type

m_admin-UpdateAllData()

```
function UpdateAllData($where, $data, $table)
{
    $this->db->where($where);
    $this->db->update($table, $data);
    $this->db->reset();
}
```

```

        $this->db->update($table, $data);

    }

```

#	Name		
21	APAR		
22	SISTEM PEMADAMAM GEDUNG		
23	SISTEM ALARM		
24	FUMIGASI		
25	PERANGKAT NETWORK		
26	SERVER		
27	Hapus Saja		
28	GPS		
29	APAR PERTAMA		

Showing 21 to 29 of 29 entries

Previous 1 2 3 Next

localhost/pmc/Admin/datatype © 2020 Pertamina Upstream Data Center

kemudian tekan button () pada data Gedung Utama PUDC maka sistem akan akan memanggil **function dataTypeDelete()** pada **C_admin** dan juga memarsing data pada **model m_admin** pada **function DeleteAllData()**.

C_admin-dataTypeDelete()

```

public function dataTypeDelete($ID)
{
    $where = array('ID' => $ID);

    $this->m_admin-
>DeleteAllData($where, 'pm_type_asset');

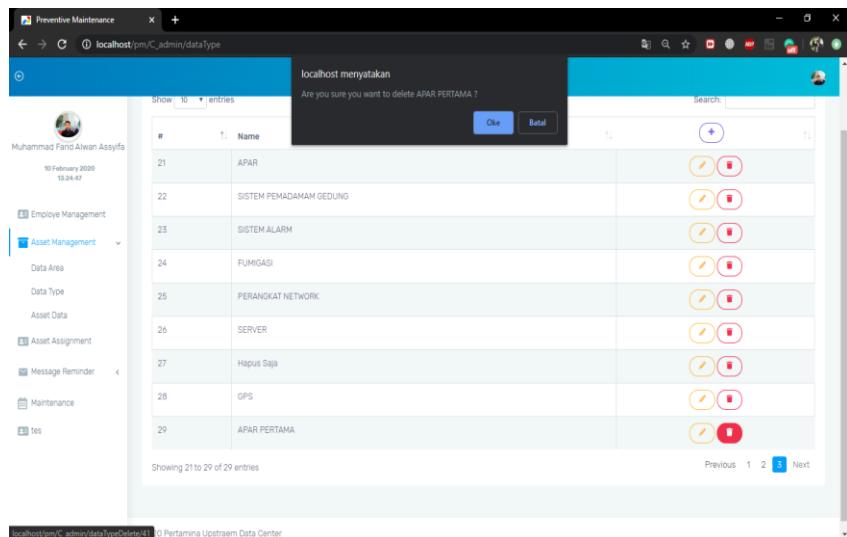
    redirect('C_admin/datatype');
}

```

M_admin-DeleteAllData()

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

Kemudian sistem akan menampilkan konfirmasi pada saat akan menghapus data **APAR PERTAMA** seperti gambar dibawah ini .



Jika menekan oke pada alert yang muncul maka data **Gedung Utama PUDC** akan terhapus dari sistem.

5.4.2.3 Pembuatan Halaman Asset Data

Kemudian ketika memilih sub menu data type maka sistem akan memanggil ***function assetdata()*** pada ***controller C_admin*** untuk menampilkan Data Type pada sistem.

C_Admin-assetdata()

```
public function assetdata()
{
    $data = array(
        'asset' => $this->m_admin->getAsset()
    );

    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
    $this->load-
>view('admin/assetManagement/assetData', $data);
    $this->load->view('resource/admin/footer');
    $this->load->view('style/style');
}
```

Pada function ini juga memanggil ***function getAsset()*** dari ***model m_Admin*** untuk menampilkan data dari database ke view ***assetData*** pada folder ***admin/assetManagement***

M_admin-getAsset()

```
function getAsset()
{
    return $this->db->query("select * from
pm_master_Asset pma
left join pm_type_asset pta
on pma.ASSET_TYPE = pta.ID
left join pm_master_area pmar
```

```
    on pma.AREA = pmar.ID  
    ")->result_array();  
}
```

assetData.php

```
<div class="page-wrapper">  
    <div class="container-fluid">  
        <div class="row page-titles">  
            <div class="col-md-6 col-8 align-self-center">  
                <h3 class="text-themecolor mb-0 mt-0">Preventive Maintenance</h3>  
                <ol class="breadcrumb">  
                    <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin">Home</a></li>  
                    <li class="breadcrumb-item"><a href="#">Asset Management</a></li>  
                    <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin/assetdata">Asset Data</a></li>  
                    <!-- <li class="breadcrumb-item active">Asset Assignment</li> -->  
                </ol>  
            </div>  
        </div>  
  
        <div class="row">  
            <div class="col-12">  
                <div class="card">  
                    <div class="card-body">  
                        <div class="table-responsive mt-40">  
                            <table id="myTable" class="table table-bordered table-striped">  
                                <thead>
```

```
<tr>
    <th>#</th>
    <th>Type</th>
    <th>Name</th>
    <th>Area</th>
    <th>Status</th>
    <th>
        <center>
            <a href
="<?php echo base_url(); ?>C_admin/assetForm" class="bt
n btn-outline-primary btn-rounded"><i class="fas fa-
plus"></i></a>
        </center>
    </th>
</tr>
</thead>
<tbody>
    <?php
        $no = 1;
        foreach ($asset as
$row) {
    ?>
    <tr>
        <td><?php e
cho $no++ ?></td>
        <td><?php e
cho $row[ 'TYPE_NAME' ] ?></td>
        <td><?php e
cho $row[ 'ASSET_NAME' ] ?></td>
        <td><?php e
cho $row[ 'AREA_NAME' ] ?></td>
        <td>
            <center>
>
            <?p
hp
if ($row[ 'ACTIVE' ] == "1") { ?>
```

```

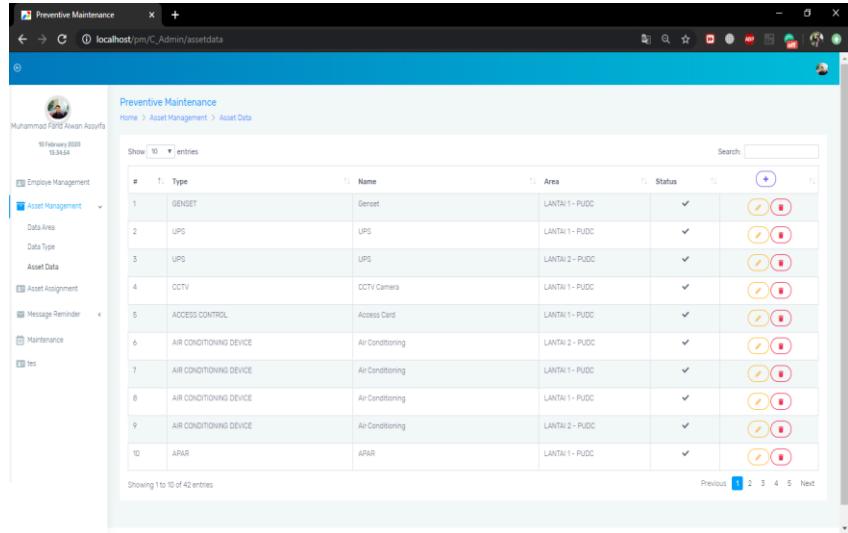
<i class="fa fa-check"></i>
<?p
hp } else { ?>

<i class="fa fa-times"></i>
<?p
hp } ?>
</center
r>
</td>

<td>
<center
>
<a
href=<?php echo base_url('C_admin/assetEdit/' . $row['
ID_ASSET']); ?>" class="btn btn-outline-warning btn-
rounded"><i class="fas fa-pencil-alt"></i></a>
<a
href=<?php echo base_url('C_admin/assetDelete/' . $row
['ID_ASSET']); ?>" class="btn btn-outline-danger btn-
rounded" onclick="return confirm('Are you sure you want
to delete <?php echo $row['AREA_NAME'] ?> ?')"><i clas
s="fas fa-trash"></i></a>
</center
r>
</td>
</tr>
<?php } ?>
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>

```

Setelah sistem selesai memanggil dan **memproses function assetdata()** pada **C_Admin** maka sistem akan menampilkan halaman seperti di bawah ini.



#	Type	Name	Area	Status	
1	GENSET	Genset	LANTAI 1 - PUDD	✓	
2	UPS	UPS	LANTAI 1 - PUDD	✓	
3	UPS	UPS	LANTAI 2 - PUDD	✓	
4	CCTV	CCTV Camera	LANTAI 1 - PUDD	✓	
5	ACCESS CONTROL	Access Card	LANTAI 1 - PUDD	✓	
6	AIR CONDITIONING DEVICE	Air Conditioning	LANTAI 2 - PUDD	✓	
7	AIR CONDITIONING DEVICE	Air Conditioning	LANTAI 1 - PUDD	✓	
8	AIR CONDITIONING DEVICE	Air Conditioning	LANTAI 1 - PUDD	✓	
9	AIR CONDITIONING DEVICE	Air Conditioning	LANTAI 2 - PUDD	✓	
10	APAR	APAR	LANTAI 1 - PUDD	✓	

Kemudian silahkan tekan button () maka sistem akan memanggil **function assetForm()** .

C_admin-assetForm()

```
public function assetForm()
{
    $data = array(
        'type' => $this->m_admin-
>getAllData('pm_type_asset'),
        'area' => $this->m_admin-
>getAllData('pm_master_area')

    );
    $this->load->view('resource/admin/header');
    $this->load->view('resource/admin/sidebar');
```

```
        $this->load-
>view('admin/assetManagement/assetForm', $data);
        $this->load->view('resource/admin/footer');
        $this->load->view('style/style');
    }
```

Setelah memanggil ***function assetForm()*** maka sistem akan menampilkan halaman ***assetForm*** dari view dalam folder ***admin/assetManagement/assetForm.php***

assetForm.php

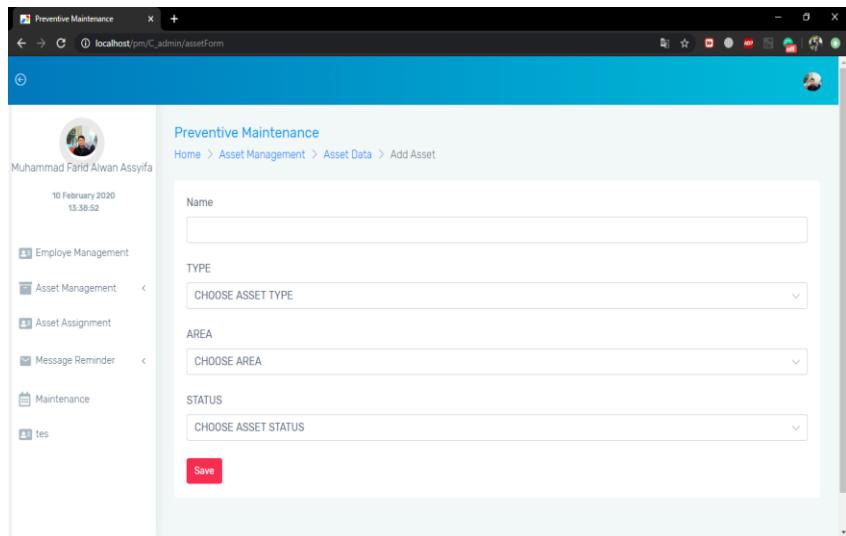
```
<div class="page-wrapper">
    <div class="container-fluid">
        <div class="row page-titles">
            <div class="col-md-6 col-8 align-self-center">
                <h3 class="text-themecolor mb-0 mt-0">Preventive Maintenance</h3>
                <ol class="breadcrumb">
                    <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin">Home</a></li>
                    <li class="breadcrumb-item"><a href="#">Asset Management</a></li>
                    <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin/assetdata">Asset Data</a></li>
                    <li class="breadcrumb-item active">Add Asset</li>
                    <!-- <li class="breadcrumb-item active">Asset Assignment</li> -->
                </ol>
            </div>
        </div>
```



```
                ?>
                <option value="

<?php echo $row['ID'] ?><?php echo $row['AREA_NAME'] ?></option>
                <?php } ?>
            </select>
        </div>
        <div class="form-group">
            <label>STATUS</label>
            <select class="form-
control custom-select" name="ACTIVE" required>
                <option value="">CH
OOSE ASSET STATUS</option>
                <option value="1">A
CTIVATED</option>
                <option value="2">D
EACTIVATED</option>
            </select>
        </div>
        <button type="submit" class
="btn btn-danger waves-effect waves-
light">Save</button>
    </form>
</div>
</div>
</div>

</div>
</div>
```



Kemudian kita inputkan data misalnya kita ketikan **name='APAR'**, **type='APAR UTAMA'**,**area='Gedung Utama PUDC'**, **status='ACTIVATED'** pada form tersebut dan menekan *button save*, ketika *button save* ditekan maka sistem akan memanggil *function adddataAsset()* pada **Controller C_Admin**. **C_Admin** akan mengirim data yang telah diinputkan ke model **m_admin** pada *function InsertData()*

C_admin-adddataAsset()

```
public function adddataAsset()
{
    $data = array(
        'ASSET_NAME' => $this->input->post('ASSET_NAME'),
        'ASSET_TYPE' => $this->input->post('ASSET_TYPE'),
        'AREA' => $this->input->post('AREA'),
        'ACTIVE' => $this->input->post('ACTIVE'),
    );
}
```

```

        // print_r($data);
        // die;
        $this->m_admin-
>InsertData('pm_master_asset', $data);

        redirect("C_admin/assetdata");

    }

```

m_admin-InsertData()

```

function InsertData($table, $data)
{
    $this->db->insert($table, $data);
}

```

Setelah data selesai diproses maka sistem akan secara otomatis akan kembali pada halaman data asset dan anda bisa melihat data yang baru diinputkan tadi pada halaman data asset

#	Type	Name	Area	Status	Action
41	PERANGKAT NETWORK	ZTE Modem	LANTAI 1 - PUDC	✓	Edit Delete
42	PERIPHERALS	Flashdisk	LANTAI 1 - PUDC	✗	Edit Delete
43	APAR PERTAMA	APAR	Gedung Utama PUDC	✓	Edit Delete

Showing 41 to 43 of 43 entries

© 2020 Pertamina Upstream Data Center

Kemudian pilih button (✎) pada data **APAR**, kemudian sistem akan memanggil **function assetEdit()** dari **Controller C_Admin**. **C_admin** akan memanggil **function EditAllData()** pada model **m_admin** untuk menampilkan data **APAR**.

C_Admin-assetEdit()

```
public function assetEdit($ID)
{
    $where = array('ID_ASSET' => $ID);
    $data['asset'] = $this->m_admin-
>EditAllData($where, 'pm_master_asset');
    $data['type'] = $this->m_admin-
>getAllData('pm_type_asset');
    $data['area'] = $this->m_admin-
>getAllData('pm_master_area');

    $this->load->view('resource/admin/header');

    $this->load->view('resource/admin/sidebar');

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

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

    $this->load->view('style/style');
}
```

m_admin>EditAllData()

```
function EditAllData($where, $table)
{
    return $this->db->get_where($table, $where)-
>result_array();

}
```

C_admin akan memarsing data dari *model m_Admin* tersebut menuju *view assetEdit* pada folder *admin/assetManagement/*

admin/assetManagement/assetEdit.php

```
<div class="page-wrapper">
    <div class="container-fluid">
        <div class="row page-titles">
            <div class="col-md-6 col-8 align-self-center">
                <h3 class="text-themecolor mb-0 mt-0">Preventive Maintenance</h3>
                <ol class="breadcrumb">
                    <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin>Home</a></li>
                    <li class="breadcrumb-item"><a href="#">Asset Management</a></li>
                    <li class="breadcrumb-item"><a href=<?php echo base_url(); ?>C_admin/assetdata">Asset Data</a></li>
                    <li class="breadcrumb-item active">Edit Asset</li>
                    <!-- <li class="breadcrumb-item active">Asset Assignment</li> -->
                </ol>
            </div>
        </div>

        <div class="row">
            <div class="col-12">
                <div class="card">
                    <div class="card-body">
                        <form action=<?php echo base_url(); ?>C_admin/assetUpdate" method="post">
                            <?php
```

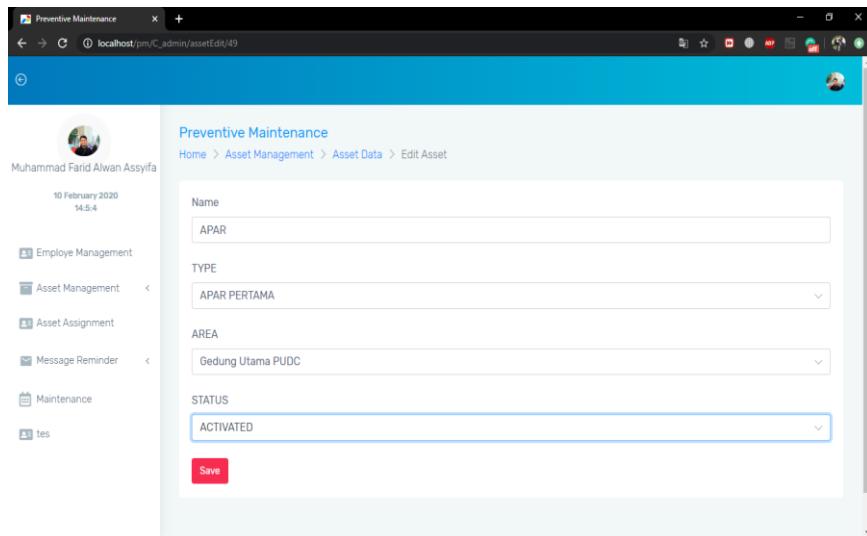
```
foreach ($asset as $row) {
    ?>
    <div class="form-
group">
        <label class="contr
ol-label">Name</label>
        <input type="text"
class="form-
control" name="ASSET_NAME" value=<?php echo $row[ 'ASSE
T_NAME' ] ?>" required>
        <input type="hidden
" class="form-
control" name="ID_ASSET" value=<?php echo $row[ 'ID_ASS
ET' ] ?>" required>
    </div>
    <?php } ?>
    <div class="form-group">
        <label>TYPE</label>
        <select class="form-
control custom-select" name="ASSET_TYPE" required>
            <option value="">CH
OOSE ASSET TYPE</option>
            <?php
            foreach ($type as $
row) {
                ?>
                <option value="
<?php echo $row['ID'] ?>"><?php echo $row[ 'TYPE_NAME' ]
?></option>
            <?php } ?>
        </select>
    </div>
    <div class="form-group">
        <label>AREA</label>
        <select class="form-
control custom-select" name="AREA" required>
            <option value="">CH
OOSE AREA</option>
            <?php
```

```

        foreach ($area as $row) {
            ?>
            <option value="<?php echo $row['ID'] ?>"><?php echo $row['AREA_NAME'] ?></option>
                <?php } ?>
            </select>
        </div>
        <div class="form-group">
            <label>STATUS</label>
            <select class="form-control custom-select" name="ACTIVE" required>
                <option value="">CHOOSE ASSET STATUS</option>
                <option value="1">ACTIVATED</option>
                <option value="2">DEACTIVATED</option>
            </select>
        </div>
        <button type="submit" class="btn btn-danger waves-effect waves-light" onclick="return confirm('Are you sure you want to update this data ?')">Save</button>
    </form>
</div>
</div>
</div>
</div>
</div>

```

Apabila sistem sudah selesai memproses data dari **C_Admin/assetEdit()** maka sistem akan menampilkan halaman tersebut sesuai dengan data yang akan di edit.



kemudian kita inputkan data terbaru untuk meng update data tersebut misalkan dari '**ACTIVATED**' menjadi '**DEACTIVATED**' kemudian tekan *button save*. Ketika *button save* ditekan maka sistem akan memanggil **function assetUpdate()** pada **controller C_Admin.**

```
public function assetUpdate()
{
    $data = array(
        #database
        'ASSET_NAME' => $this->input->post('ASSET_NAME'),
        'ASSET_TYPE' => $this->input->post('ASSET_TYPE'),
        'AREA' => $this->input->post('AREA'),
        'ACTIVE' => $this->input->post('ACTIVE'),
    );
    $where = array(

```

```
'ID_ASSET' => $this->input-
>post('ID_ASSET')
);
// print_r($data);
// die;
$this->m_admin-
>UpdateAllData($where, $data, 'pm_master_asset');
redirect('C_admin/assetdata');
}
```

Kemudian pada areaUpdate ini data akan di parsing ke model **m_Admin** pada function **UpdateAllData()**. Setelah data berhasil di parsing kedalam **UpdateAllData** maka dia akan otomatis kembali kehalaman awal asset data

m_admin-UpdateAllData()

```
function UpdateAllData($where, $data, $table)
{
    $this->db->where($where);
    $this->db->update($table, $data);

}
```

The screenshot shows a table with the following data:

#	Type	Name	Area	Status
41	PERANGKAT NETWORK	ZTE Modem	LANTAI 1 - PUDC	
42	PERIPHERALS	Flashdisk	LANTAI 1 - PUDC	
43	APAR PERTAMA	APAR	Gedung Utama PUDC	

kemudian tekan button () pada data Gedung Utama PUDC maka sistem akan akan memanggil ***function assetDelete()*** pada **C_admin** dan juga memarsing data pada **model m_admin** pada ***function DeleteAllData()***.

C_admin-assetDelete()

```
public function assetDelete($ID)
{
    $where = array('ID_ASSET' => $ID);
    $this->m_admin-
>DeleteAllData($where, 'pm_master_asset');
    redirect('C_admin/assetdata');
}
```

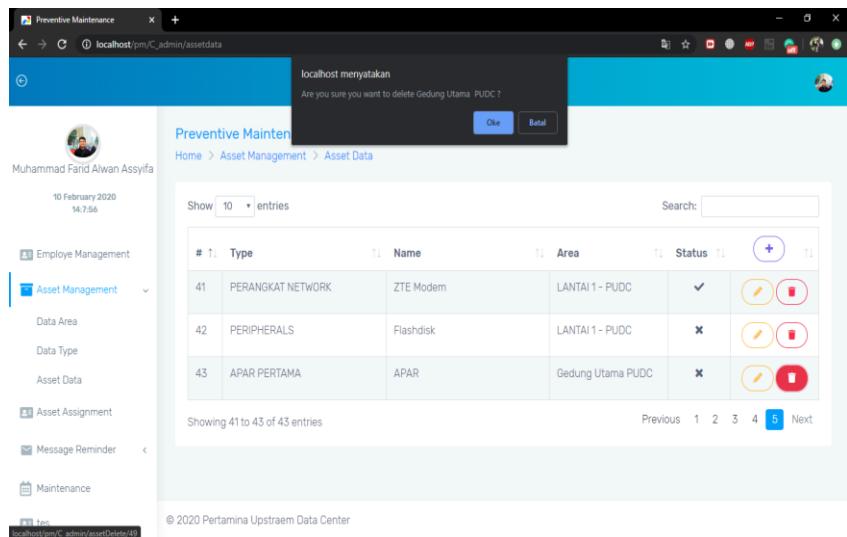
M_admin-DeleteAllData()

```
function DeleteAllData($where, $table)
{
    $this->db->where($where);
```

```
        $this->db->delete($table);  
  
    }  

```

Kemudian sistem akan menampilkan konfirmasi pada saat akan menghapus data **APAR PERTAMA** seperti gambar dibawah ini .



Jika menekan oke pada alert yang muncul maka data **APAR PERTAMA** akan terhapus dari sistem.