

Tutorial Integrasi Framework Code Igniter Dengan Whatsapp Untuk Proses Registrasi Pada Aplikasi Berbasis Web

Oleh

RYAN SITUMORANG

1804107

YUKI ARDIANSYAH

1184053

PROGRAM STUDI D4 TEKNIK INFORMATIKA

POLITENIK POS INDONESIA

2019

Penulis: Rolly Maulana Awangga

ISBN :

Editor: M. Yusril Helmi Setyawan

Penyunting:

Syafrial Fachrie Pane

Khaera Tunnisa

Diana Asri Wijayanti

Desain sampul dan Tataletak:

Deza Martha Akbar

Penerbit:

Kreatif Industri Nusantara

Redaksi:

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529 Email : awangga@kreatif.co.id

Distributor:

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

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

CONTRIBUTORS

M YUSRIL HELMI SETYAWAN, S.KOM., M.KOM., YUKI
ARDIANSYAH, RYAN SITUMORANG, Informatics Research Center.,
Politeknik Pos Indonesia, Bandung, Indonesia

ABSTRAK

E-commerce adalah sebuah media berbelanja bagi semua masyarakat yang ada di dunia, dengan adanya E-commerce memudahkan pengguna atau user dalam memilih barang yang akan dibeli dan digunakan, Bagaimanapun terbatasnya jangkauan pemasaran barang dan kurang efisiennya sistem promosi barang kepada masyarakat menyebabkan tidak tercapainya omzet yang maksimal dari vakansi.dengan berbasis web perkembangan teknologi pedagangdengan mudah memulai berjualan dalam media platform E-commerce karena efisien dan mudah digunakan

E-commerce di mulai dalam pembuatan akun yang akan digunakan pada pengguna atau user proses registrasi pada E-commerce juga perlu adanya proses penyambungan database untuk mengambil data yang akan digunakan sebagai Username dan Password

Kata Kunci: E-commerce, Web , Database , Sinstem Informasi

ABSTRACT

E-commerce is a shopping media for all people in the world, with E-commerce makes it easy for users or users to choose the items to be purchased and used, however the limited marketing reach of goods and the inefficient system of promotion of goods to the public causes not achieving turnover the maximum from vacancies. with the web-based development of technology traders easily start selling in the E-commerce platform media because it is efficient and easy to use

E-commerce starts in creating an account that will be used by the user or user registration process in E-commerce also needs a database connection process to retrieve data that will be used as a Username and Password

Keyword : Information System, Mail, E- Commerce, Web ,Database

KATA PENGANTAR

Puji syukur penulis panjatkan kehadapan Tuhan yang Maha Esa, karena atas berkat dan rahmat-Nya penulis dapat menyelesaikan penyusunan Tugas Project 2 yang berjudul “Pencarian Debug Pada Sistem Registrasi Akun Web E-Commerce Cahaya-Jaya” pada waktunya.

Dalam melakukan penulisan laporan ini banyak kendala yang penulis hadapi. Adanya kesulitan dalam mencari data-data dan informasi merupakan salah satu kendala yang penulis hadapi. Akan tetapi dengan semangat dan tekad yang kuat, penulis berhasil menyelesaikan laporan ini.

Kiranya tak ada ucapan yang lebih baik kecuali rasa terima kasih kepada :

1. Orang tua dengan dukungan dan do'a nya.
2. DR. Ir. Agus Purnomo, M.T. selaku Direktur Politeknik Pos Indonesia.
3. M. Yusril Helmi Setyawan, S.Kom., M.Kom. selaku Ketua Program Studi DIV Teknik Informatika.
4. Rolly Maulana Awangga, S.T., M.T. selaku Koordinator Proyek II.
5. Rd. Nuraini Siti Fathonah, S.S., M.Hum. selaku Pembimbing Proyek II yang telah memberikan pengarahan dan membimbing kepada penulis.
6. M. Harry K. Saputra, S.T., M.T.I.. selaku dosen wali kelas D4 TI 1C
7. Serta semua pihak yang tidak dapat penulis sebutkan satu persatu.

Akhir kata penulis berharap semoga laporan ini dapat bermanfaat bagi seluruh mahasiswa khususnya bagi penulis. Penulis sadar bahwa laporan ini masih jauh dari kata sempurna. Oleh karena itu penulis sangat mengharapkan adanya kritik dan saran yang membangun dari para pembaca untuk kesempurnaan laporan ini. Semoga Allah SWT selalu memberikan rahmat dan hidayah-Nya kepada kita semua. Amin

Bandung, 13 Februari 2020

DAFTAR ISI

Contents

DAFTAR ISI 5

BAB I 6

PENDAHULUAN 6

1.1	Latar Belakang Masalah	6
1.2	Identifikasi Masalah	6
1.3	Rumusan Masalah	7
1.4	Tujuan Penelitian	7
1.5	Ruang Lingkup	7
1.6	Sistematika Penulisan	8

DAFTAR GAMBAR

DAFTAR TABEL

Listings

ACKNOWLEDGMENTS

ACRONYMS

GLOSSARY

SYMBOLS

A Amplitude

& Propositional logic symbol

a Filter Coefficient

B Number of Beats

Terimakasih atas semua masukan dari dosen pembimbing keluarga dan teman-teman agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti. Terima kasih ini juga ditujukan khusus untuk team IRC yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Proyek II.

BAB I

PENDAHULUAN

CODE IGNITER



CodeIgniter adalah framework yang akan kita pakai karena aplikasi yang telah dibuat di dalam buku ini menggunakan framework CodeIgniter, sebelum membahas CodeIgniter sebelumnya kita telah membahas tentang apa itu Framework dan jenis-jenisnya.

Kita akan sedikit membahas kembali apa itu Framework adalah sebuah kerangka kerja atau kumpulan data yang dapat membantu programmer dalam menjali sebuah masalah dalam menjalankan beberapa Bahasa pemrograman dan koneksi ke dalam sebuah database sehingga membantu programmer dalam menjalankan tugasnya.

Sedangkan untuk Framework CodeIgniter ini sudahh banyak tersedia sebuah library yang biasanya diperlukan dalam pengembangan sebuah website ataupun aplikasi yang berbasis di dalam web, di dalam CodeIgniter (CI) terdapat library untuk mengoperasikan database sampai library untuk masalah lainnya

Framework CodeIgniter ini juga berdasarkan design patternkonsep MVC Model, View, Controller). Secara garis besar Konsep MVC ini adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan Data (Model) dari tampilan (View) serta cara memprosesnya (Controller).

Framework Codeigniter ini pertama kali dibuat oleh Rick Ellis seorang musisi sekaligus juga seorang programmer, dan kemudian Framework ini dirilis pada tanggal 28 Februari 2006 dan sejak tahun 2014 sampai saat ini Framework Codeigniter ini telah dimiliki dan dikembangkan oleh British Columbia Institute of Technology (BCIT) situs resminya beralamat di <https://www.codeigniter.com/>

Design Pattern adalah kumpulan penjelasan mengenai metode-metode bagaimana cara menyelesaikan suatu masalah yang umum ditemui dalam proses perancangan perangkat lunak (Software Design). Design pattern merupakan petunjuk bagaimana cara menyelesaikan suatu masalah di seputar dunia software design.

MVC adalah sebuah metode pendekatan yang dapat ditempuh untuk memisahkan aplikasi menjadi tiga bagian, yaitu :

a) Model

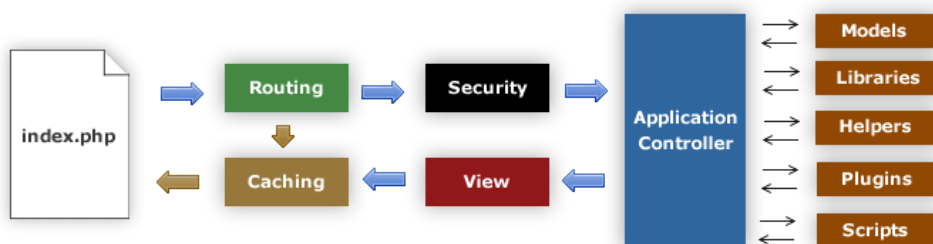
Model berfungsi untuk merepresentasikan data yang digunakan aplikasi. Contoh data misalnya database, RSS, atau data yang diperoleh dari pemanggilan API, dan aksi yang melibatkan operasi Create, Read, Update, dan Delete (CRUD) data.

b) View

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

c) Controller

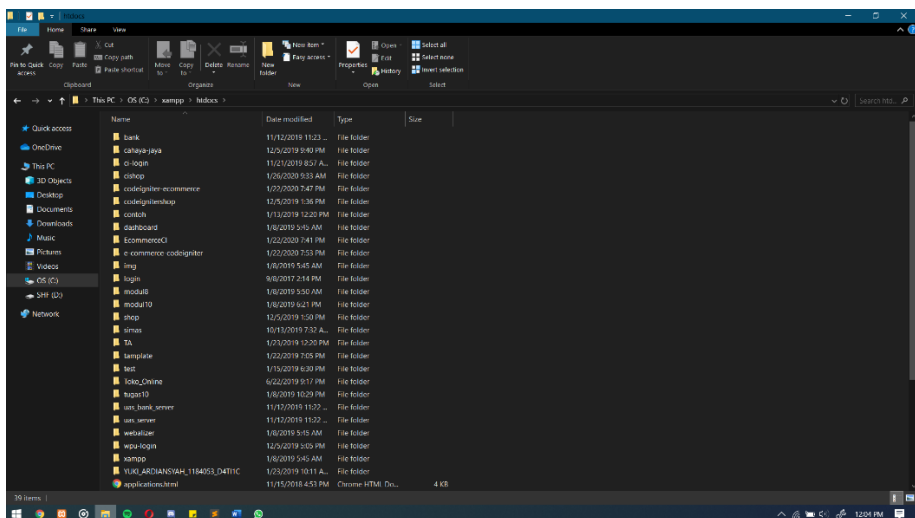
Controller adalah business logic yang bertugas sebagai jembatan antara Model dan View. Controller akan merespon HTTP request yang datang dari user (melalui browser), dari request ini controller akan menentukan apa yang harus dilakukan. Jika diperlukan, controller akan memanggil model untuk mengolah data, kemudian akan memanggil view yang sesuai untuk menampilkan data hasil olahan model dan akhirnya mengirim kembali halaman hasil olah data ke browser.



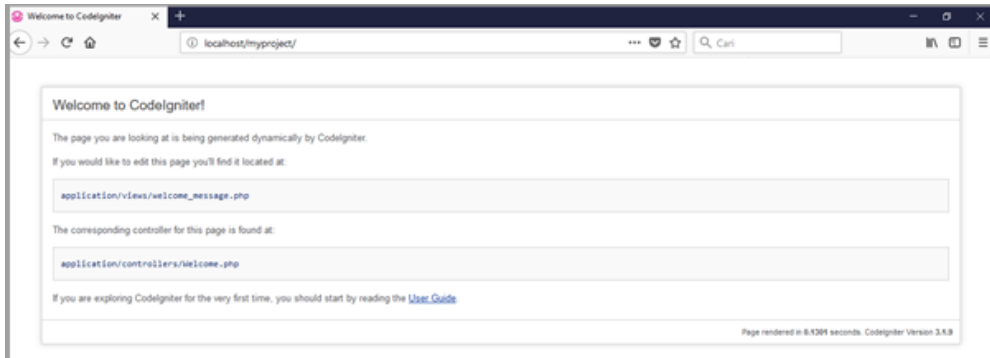
- A. Index.php berfungsi sebagai front controller, menginisialisasi base resource untuk menjalankan CodeIgniter.
- B. Router memeriksa HTTP request untuk menentukan apa yang harus dilakukan dengannya.
- C. Jika Cache aktif maka hasilnya akan langsung dikirimkan ke browser dengan mengabaikan aliran data normal.
- D. Security. Sebelum controller dimuat, HTTP request dan data yang dikirimkan user akan difilter untuk keamanan.
- E. Controller membuat model, core libraries, plugins, helper, dan semua resource yang diperlukan untuk memproses request.
- F. View yang dihasilkan akan dikirim ke browser. Jika cache aktif maka View akan disimpan sebagai cache dahulu sehingga para request berikutnya langsung ditampilkan

Langkah – Langkah Instalasi CodeIgniter

1. Langkah pertama pastikan web server telah terinstall di dalam computer masing-masing
2. Setelah itu unduh file codeigniter di situs resminya untuk mendapatkan templatennya www.codeigniter.com
3. Extract file Codeigniter.zip ke direktori C:/wamp/www/ (jika Anda menggunakan wampserver). Tetapi, jika Anda menggunakan XAMPP. Extract file Codeigniter.zip ke direktori C:/xampp/htdocs/
4. Pergi ke folder c:/wamp/www/ (jika Anda menggunakan WAMP) dan **rename** (ganti nama) file folder codeigniter yang baru di extraxt tadi menjadi nama project Anda.



5. Selanjutnya buka browser Anda. disini saya menggunakan Mozilla Firefox. Kemudian kunjungi URL berikut: <http://localhost/myproject>
Jika instalasi berhasil maka, akan terlihat seperti gambar berikut:



6. Selesai

Konfigurasi CodeIgniter

CodeIgniter dapat berjalan dengan konfigurasi default, tetapi untuk sebuah aplikasi yang nyata kita harus tetap melakukan konfigurasi, setidaknya pada bagian `base_url` dan `router`. Pengaturan `base_url` dan `router` sangat berguna ketika proses pengembangan aplikasi yang banyak menggunakan helper dan library.

File konfigurasi terletak dalam folder `application/config`. File yang terdapat pada direktori tersebut yang sering digunakan adalah file `autoload.php`, `config.php`, `database.php` dan `routes.php`.

1. File `autoload.php`. konfigurasi pada file ini bertujuan untuk menentukan sebuah sumber yang akan menjadi titik dari penyimpanan agar dapat dipanggil secara otomatis

File `autoload['libraries']` :

```
$autoload['libraries'] = array();
/*
-----
Auto-load Libraries
-----
These are the classes located in system/libraries/ or your
application/libraries/ directory, with the addition of the
'database' library, which is somewhat of a special case.

Prototype:
    $autoload['libraries'] = array('database', 'email', 'session');
You can also supply an alternative library name to be assigned
in the controller:
    $autoload['libraries'] = array('user_agent' => 'ua');
*/
File autoload.php bagian libraries
```

File `autoload['helper']` :

```

$autoload['helper'] = array();
/*
|-----|
| Auto-load Helper Files |
|-----|
| Prototype: |
| |
| $autoload['helper'] = array('url', 'file'); |
| |
|*/

```

File autoload.php bagian helper

2. File config.php. pada file ini kita akan mengkonfigurasi secara umum dalam CodeIgniter.

```

$config['base_url'] = '';
/*
|-----|
| Base Site URL |
|-----|
|
| URL to your CodeIgniter root. Typically this will be your base URL,
| WITH a trailing slash:
|
| http://example.com/
|
| WARNING: You MUST set this value!
|
| If it is not set, then CodeIgniter will try guess the protocol and path
| your installation, but due to security concerns the hostname will be set
| to $_SERVER['SERVER_ADDR'] if available, or localhost otherwise.
| The auto-detection mechanism exists only for convenience during
| development and MUST NOT be used in production!
|
| If you need to allow multiple domains, remember that this file is still
| a PHP script and you can easily do that on your own.
|
|*/

```

File config.php bagian base url

\$config['base_url'] – konfigurasi ini berisikan alamat url pada sebuah aplikasi yang akan dibuat. Sebagai contoh “C:xampp/htdocs/codeigniter”

3. File database.php. disini kita akan mengkonfigurasi koneksi database. Pada aplikasi yang digunakan dalam buku ini kita menggunakan database mysql yang terdapat pada aplikasi XAMPP

```

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => '',
    'password' => '',
    'database' => '',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);

```

File database.php

- A. Hostname menggunakan localhost karena di sini kita menggunakan database mysql dari aplikasi xampp yang terinstal di computer local.
 - B. Username di dalam sini kita isikan sesuai username yang telah kita buat didalam mysql atau secara default bisa diisikan dengan “root”
 - C. Password sesuaikan dengan mysql masing-masing
 - D. Database ini merupakan nama dari database yang akan kita pakai di dalam sebuah aplikasi yang dibuat
4. File routes.php. Konfigurasi routing digunakan untuk memetakan permintaan atau request kedalam controller didalam website yang dibuat. Misalnya kita membuka alamat <http://www.nama-website.com>, permintaan tersebut tidak menyertakan nama controller yang ingin dibuka tetapi kita bisa secara default mengarahkannya agar secara otomatis akan membuka controller sesuai yang definisikan. Untuk melakukan konfigurasi routing buka file konfigurasi routes.php. Settingan utama yang ada adalah sebagai berikut:

```
$route['default_controller'] = 'welcome';  
$route['404_override'] = '';  
$route['translate_uri_dashes'] = FALSE;
```

File routes.php

Artinya secara default semua permintaan yang tidak menyertakan nama controllernya akan diarahkan untuk membuka controller “welcome”. Sehingga saat alamat <http://www.nama-website.com> dibuka secara otomatis akan terbuka

Didalam codeigniter ada beberapa folder yang menjadi tempat bagi script php yang di simpan, pada setiap folder pun punya fungsi masing-masing untuk penempatan script seperti config, dll. Disini sebelum memberikan contoh sederhana alangkah baiknya memoerkenalkan setiap folder yang ada di dalam CodeIgniter

Folder Application adalah folder dasar yang meyimpan data aplikasi yang sedang dibuat

Folder Cache adalah folder yang menyimpan semua cache yang ada di aplikasi yang telah dibuat oleh cache library

Folder Config adalah folder yang berguna untuk menyimpan data atau informasi dari konfigurasi aplikasi yang telah dibuat seperti autoload, database, routes dan lainnya.

Folder Controller, adalah folder yang menyimpan controller dari aplikasi yang dapat digunakan untuk menyusun aktivitas program .

Folder Core, adalah folder untuk memperluas class class inti codeigniter.

Folder Helpers, merupakan folder untuk menyimpan helpers.

Folder Hooks, merupakan folder untuk menyimpan hooks untuk mengubah alur fungsi dari core Codeigniter

Folder Language, merupakan folder untuk menyimpan bahasa - bahasa yang akan digunakan.

Folder Libraries, merupakan folder untuk menyimpan library.

Folder Logs, merupakan folder untuk menyimpan semua error log apabila error log diaktifkan.

Folder Models, merupakan folder untuk menyimpan models yang akan mendefinisikan tabel dari database yang dapat kita gunakan oleh Controller yang kita buat untuk mengakses database.

Folder Third_party, merupakan folder untuk menyimpan fungsi fungsi tambahan dalam cara kerja codeigniter.

Folder Views, merupakan folder untuk menyimpan tampilan dari aplikasi yang kita buat.

Folder System, merupakan folder untuk menyimpan sistem inti dari Codeigniter.

Setelah saya akan memberikan contoh sederhana berupa menampilkan sebuah script yang akan memanggil hello word. Seperti yang telah disampaikan pada materi sebelumnya langkah pertama kita buka file yang telah kita buat di dalam /xampp/htdocs/namafolder kedalam aplikasi pemrograman seperti sublime atau yang lainnya, application seperti yang ditandai dengan kotak berwarna merah. Didalam folder application terdapat banyak folder, tetapi yang menjadi perhatian kita saat ini adalah folder controllers, views dan models.

1.1 Latar Belakang Masalah

Seiring dengan perkembangan teknologi dalam era globalisasi saat ini, pada setiap kegiatan yang dilakukan akan memberikan sebuah dampak positif terhadap kemajuan dan kelancaran pada sebuah institusi. Kebutuhan akan informasi menjadi dasar bagi setiap organisasi untuk melakukan

pengelolaan informasi dengan menggunakan teknologi informasi pada saat ini

Salah satu bidang yang mendapatkan dampak yang cukup berarti dengan perkembangan teknologi internet ini adalah bidang Online Shopping. Teknologi internet dapat dimanfaatkan untuk mempermudah dan memperlancar proses transaksi jual beli, sehingga dapat dikatakan teknologi internet dapat meningkatkan peluang bisnis para pengusaha. Beberapa bagian dari fashion seperti penjualan, pemasaran, pembelian dan transaksi jual beli mendapatkan sentuhan media teknologi informasi dalam hal ini internet, sehingga mencetuskan lahirnya ide tentang penjualan jarak jauh atau yang biasa disebut penjualan online atau e-commerce (Electronic Commerce).

Sistem registrasi juga diperlukan dalam membuka sebuah E-Commerce yang akan kita tuju sebagai contoh web E-Commerce Cahaya Jaya yang menjual beberapa makanan pada proses registrasi yang dilakukan menggunakan sistem yang dipakai dalam metode registrasi dalam mengambil data dari database yang dipakai untuk masuk ke dalam web yang dituju

Dengan berbasis web dengan pemrograman yang dipakai adalah PHP aplikasi E-Commerce ini juga memakai sistem framework CI(CodeIgniter) yang berguna sebagai template dalam melakukan kodingan di E-Commerce

1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah yang telah dikemukakan di atas identifikasi masalahnya adalah :

1. Bagaimana mengelola debug yang ada pada E-Commerce
2. Bagaimana sistem informasi registrasi yang dilakukan dalam mengelola akun
3. Bagaimana pengelolaan database dalam menginput data dalam web

1.3 Rumusan Masalah

Sesuai dengan latar belakang yang tertera diatas maka dapat dirumuskan dengan system registrasi diatas dapat melakukan pencarian debug yang dapat melakukan proses pendaftaran dalam E-Commerce dapat melakukan proses registrasi yang dilakukan oleh user maupun pedagang

1.4 Tujuan Penelitian

Berdasarkan latar belakang masalah dan identifikasi masalah, maka tujuan yang ingin dicapai adalah :

2. Sistem informasi ini dapat memberikan notifikasi status registrasi melalui email terhadap user di dalam E-Commerce.
3. Sistem informasi ini dapat mengelola user secara terkomputerisasi ke dalam database.

1.5 Ruang Lingkup

Dalam perancangan sistem ini ada beberapa hal yang perlu diperhatikan, yaitu:

1. Sistem informasi ini mengikuti pola user dalam melakukan proses registrasi yang akan dipakai

2. Sistem informasi ini hanya melayani pengajuan akun pada web E-commerce cahaya jaya.
3. Proses register ini akan melakukan pengambilan data yang diperlukan dalam dataset untuk proses registrasi

1.6 Sistematika Penulisan

Penyajian laporan proyek dibagi menjadi beberapa Bab dengan tujuan untuk mempermudah pencarian data atau informasi yang dibutuhkan, serta menunjukkan penyelesaian pekerjaan yang sistematis. Pembagian Bab tersebut adalah sebagai berikut:

BAB I Pendahuluan, Latar Belakang Masalah berisi ulasan ringkas mengenai keadaan atau kondisi yang ada dan kekurangan dari sistem yang diamati sehingga muncul topik yang diambil. Identifikasi Masalah berisi berbagai masalah yang sudah dikenali dan akan diberikan solusinya melalui fungsi dan aplikasi yang akan dibuat. Tujuan berisi tujuan untuk apa aplikasi dibuat. Ruang Lingkup berisi batasan-batasan proyek yang akan dibangun. Sistematika Penulisan menjelaskan isi yang ada didalam proyek.

BAB II Landasan Teori, membahas tentang konsep dasar dan pengertian aplikasi yang mendukung terbentuknya proyek “Pengembangan Sistem Informasi Pengelolaan Arsip Surat dan Dokumen Prodi D4 Teknik Informatika Berbasis Web”.

BAB III Analisis dan Perancangan adalah proses untuk menentukan bentuk dari kebutuhan aplikasi baik berupa kebutuhan pada saat membangun aplikasi maupun pada saat implementasi. Serta penjelasan perancangan aplikasi yang akan dibuat terdiri dari perancangan alir program (Flow Chart), data, maupun perancangan input dan output aplikasi.

BAB IV Implementasi dan Pengujian adalah aplikasi yang dibuat dengan merinci komponen-komponen pendukung berupa program, lingkungan

implementasi, tampilan antarmuka, petunjuk pemakaian, petunjuk instalasi. Serta cara untuk mengetahui apakah aplikasi yang dibuat sesuai rancangan dan menuliskan hasil ujinya.

BAB V Kesimpulan dan Saran adalah berisi pencapaian tujuan dari aplikasi yang dibuat. Dan berisi hal-hal atau tujuan dari pembuatan aplikasi yang dirasa belum sempurna atau tidak tercapai. Juga bisa berupa kondisi implementasi yang optimal bagi aplikasi yang dibuat'

Jenis – jenis web

□ Jenis-Jenis Website Berdasarkan Platform: Tersedia berbagai cara untuk membuat website. Setidaknya Anda bisa membangun website dengan tiga metode, yaitu menggunakan CMS, website builder, atau kode, berikut penjelasannya :

1. CMS

Metode pertama adalah cara yang paling populer, yaitu menggunakan Content Management System (CMS). CMS merupakan perangkat lunak atau sistem yang digunakan untuk membuat dan mengatur suatu konten digital. Membuat sebuah website menjadi lebih cepat dan mudah. Beberapa CMS yang paling banyak digunakan adalah WordPress, Joomla!, dan Drupal.

2. Website Builder

Website builder disebut juga sebagai website gratis yang cocok digunakan untuk anda yang ingin membuat website dalam waktu singkat, tidak mempunyai kemampuan teknis dan tidak ada waktu untuk mempelajarinya.

Beberapa contoh website builder populer adalah Wix, Site Builder, dan Weebly. Website Builder juga memiliki kelebihan dan kekurangan antara lain sebagai berikut:

Kelebihan dari website builder adalah pengguna mendapatkan paket lengkap membuat website, dari hosting, domain, hingga pilihan template. Pengguna hanya perlu mengganti konten sesuai yang diinginkan dan bisa dibantu oleh support dari penyedia website builder.

Kekurangan dari website builder adalah:

- a. Keterbatasan pada jumlah halaman
- b. Tanpa nama domain profesional
- c. Konsumen sulit untuk menemukan website anda
- d. Dukungan customer service yang minim e. Keterbatasan fitur lainnya

3. Kode (HTML dan CSS)

Pada metode pembuatan website terakhir ini memerlukan sebuah pengetahuan tentang coding dan pemrograman menggunakan HTML dan CSS. Anda bisa membuat kode tersebut untuk website dapat menggunakan software seperti Notepad++, Text Wrangler, atau Sublime Text

Dashboard

Dashboard merupakan sebuah model aplikasi sistem informasi yang disediakan bagi para manajer untuk menyajikan informasi kualitas kerja, dari sebuah perusahaan atau lembaga organisasi (Ilhamsyah, 2017). Dashboard merupakan control panel center berplatform yang berfungsi untuk mengatur semua kegiatan di sebuah website. Dalam sistem informasi manajemen, dashboard dikatakan sebagai alat untuk membaca, menampilkan presentasi grafis status saat ini (snapshot), real-time user interface, sejarah tren indikator kinerja utama organisasi untuk diaktifkan seketika dan harus membuat informasi keputusan sekilas dan halaman suatu situs. Dapat disimpulkan bahwa dashboard merupakan sebuah sistem aplikasi yang berfokus kepada data atau informasi yang disajikan agar terlihat begitu menarik dan dapat dipahami dengan mudah yang bertujuan dalam membantu pengambilan keputusan. Dashboard memiliki nama lain, yaitu “laporan kemajuan” atau “manajemen laporan”. Dashboard sering ditampilkan pada halaman situs yang terhubung dengan database yang memungkinkan laporan diperbaharui terus-menerus. Dashboard memiliki beberapa, berikut manfaat yang dihasilkan dari Dashboard:

Media informasi yang menyajikan data atau informasi dengan grafik ataupun kalimat ringkasan dari data atau informasi yang disajikan secara efisien. Media monitoring yang memantau perkembangan atau progress dari suatu kegiatan. Sebagai sistem penunjang keputusan bagi Middle-Top Level Management sebuah perusahaan, organisasi, ataupun instansi.

Apakah dashboard itu penting? Ya, dashboard itu penting, contohnya pada sebuah perusahaan ketika penjualan dan produksi produknya sangat tinggi, tiba-tiba produksi berkurang dan penjualannya pun merosok jauh dari jumlah produktivitas sebelumnya, maka pemilik perusahaan pun pasti bingung tentang hal apa yang menyebabkan produktivitas produk perusahaannya menurun sehingga mempengaruhi angka penjualan, lalu pemilik perusahaan tersebut meminta laporan mengenai perkembangan produktivitas dan penjualan produk perusahaannya dan bagaimana jika laporan yang diberikan atau diterima kepada pemilik perusahaan atau atasan tersebut berupa laporan yang berisi angka yang sangat banyak, sehingga untuk mencari akar permasalahan dari menurunnya produktivitas dan penjualan perusahaannya sulit ditemukan, bisa dilihat seperti dibawah ini.

BAB II

Pengertian Singkat

2.1 Pengertian Sistem

Sistem adalah kumpulan unsur-unsur yang bergabung menjadi sebuah kesatuan dan mempunyai tujuan yang sama. Unsur dalam sistem saling terhubung satu sama lain sehingga sebuah arus informasi dengan mudah dicapai dengan bersama

2.2 Pengertian Website

Website adalah sering juga disebut Web, dapat diartikan suatu kumpulan-kumpulan halaman yang menampilkan berbagai macam informasi teks, data, gambar diam ataupun bergerak, data animasi, suara, video maupun gabungan dari semuanya, baik itu yang bersifat statis maupun yang dinamis, yang dimana membentuk satu rangkaian bangunan yang saling berkaitan dimana masing-masing dihubungkan dengan jaringan halaman atau hyperlink.

2.3 Pengertian E-Commerce

E-commerce adalah transaksi yang bersifat non finansial antara dua pihak yang saling membutuhkan melalui suatu media elektronik. setiap pertukaran informasi antara perusahaan dengan pihak stakeholder eksternalnya (pihak ketiga yang memiliki kesepakatan tertentu dengan perusahaan) melalui media elektronik.

2.4 Pengertian Database

Basis Data adalah himpunan kelompok data yang saling berhubungan dan tanpa pengulangan, yang disimpan dalam media elektronik dan diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan dengan cepat mudah.

2.5 Analisis dan Perancangan Sistem

Definisi Analisis dan Perancangan Sistem menurut para ahli :

- A. Analisis Sistem adalah suatu studi dari sistem yang telah ada dengan tujuan untuk merancang sistem yang baru atau memperbaiki kekurangan dari sistem yang telah ada.
- B. Perancangan sistem adalah penentuan proses dan data yang diperlukan oleh sistem baru, jika sistem itu berbasis komputer, perancangan dapat dinyatakan spesifikasi peralatan yang digunakan.

2.5.1 Flowchart

Suatu gambaran yang mempresentasikan suatu algoritma atau prosedur menyelesaikan suatu masalah. . Flowchart terbagi dua, yaitu flowchart sistem dan flowchart program.

2.5.2 UML (Unified Modelling Language)

Menurut Henderi (2008:5) “UML adalah bahasa spesifikasi standar untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi”. Jenis-jenis diagram UML:

- A. Use Case Diagram Use case diagram secara grafis menggambarkan interaksi antara sistem, sistem eksternal dan pengguna. Dengan kata lain use case diagram secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna (user) mengharapkan interaksi dengan sistem itu. Use case secara naratif digunakan untuk secara tekstual menggambarkan sekuensi langkah-langkah dari setiap interaksi.
- B. Class Diagram Menggambarkan struktur object sistem. Diagram ini menunjukkan class object yang menyusun sistem dan juga hubungan antara class object tersebut.

C. Sequence Diagram Secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah use case atau operasi.

D. State Chart Diagram Digunakan untuk memodelkan behaviour objek khusus yang dinamis. Diagram ini mengilustrasikan siklus hidup objek berbagai keadaan yang dapat diasumsikan oleh objek dan event-event (kejadian) yang menyebabkan objek beralih dari satu state ke state yang lain.

E. Activity Diagram Secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun use case. Activity diagram dapat juga digunakan untuk memodelkan action yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari action tersebut.

2.6. Perancangan Basis Data

Perancangan basis data adalah sebuah pendekatan struktur yang mencakup prosedur, teknik, alat bantu perancangan untuk merencanakan, mengatur dan mengevaluasi pengembangan dari proyek pembuatan basis data tersebut . Menurut Conolly metodologi perancangan basis data terbagi menjadi 2 bagian yaitu:

1. Perancangan Fisik Merupakan proses pendefinisian entitas dan relasi (Relationship) dari dunia nyata yang dirancang, berdasarkan kebutuhan informasi dan pengolahan data dari organisasi yang bersangkutan.

A. Entity / entitas adalah sekumpulan objek yang dapat diidentifikasi dan dibedakan di lingkungan pemakai

B. Relasi adalah hubungan yang terjadi antar kelompok entitas.

2. Perancangan Logik Merupakan proses untuk mengimplementasikan hasil perancangan logika ke dalam komputer secara fisik yang bergantung kepada software DBMS yang dipilih. Proses yang dilakukan :

- A. Menentukan struktur untuk setiap tabel, meliputi nama field, jenis, lebar dan field kuncinya.
- B. Menentukan nama basis data dan nama setiap tabel, serta lokasi tempat penyimpanannya (drive, directory / folder).
- C. Menghitung perkiraan tempat (space) yg dibutuhkan untuk seluruh tabel dan untuk seluruh index.
- D. Implementasi dengan menggunakan software DBMS.

2.7. DBMS (Data Base Management System)

DBMS ialah perantara untuk user dengan basis data, untuk dapat berinteraksi dengan DBMS dapat memakai bahasa basis data yang sudah ditentukan oleh perusahaan DBMS. Bahasa basis data umumnya terdiri dari berbagai macam instruksi yang diformulasikan sehingga instruksi tersebut dapat di proses oleh DBMS.

1. XAMPP

XAMPP adalah program aplikasi pengembang yang berguna untuk pengembangan website berbasis PHP dan MySQL. Melalui program ini, programmer web dapat menguji aplikasi web yang dikembangkan dan mempresentasikannya ke pihak lain secara langsung dari komputer, tanpa perlu terkoneksi ke internet. XAMPP juga dilengkapi fitur manajemen database PHPMyAdmin seperti pada server hosting sungguhan, sehingga pengembang web dapat mengembangkan aplikasi web berbasis database secara mudah.

2. MySQL

MySQL merupakan aplikasi database server. Perkembangannya disebut SQL yang merupakan kepanjangan dari Structured Query Language. SQL merupakan bahasa terstruktur yang digunakan untuk mengolah database. MySQL dapat digunakan untuk membuat dan mengelola database beserta isinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah, dan menghapus data yang berada dalam database.

2.8. Bahasa Pemrograman

Bahasa pemrograman adalah sebuah instruksi standar untuk memerintah komputer agar mempunyai fungsi tertentu. Berikut ini adalah beberapa contoh bahasa pemrograman yang biasa digunakan :

1. HTML (Hyper Text Markup Language)

Pengertian HTML : “HTML adalah dokumen text yang bisa dibaca untuk dipublikasikan di World Wide Web (WWW), dan semua nama file dokumen HTML mempunyai ekstensi html atau htm”.

Script dasar pada HTML :

```
<HTML>
```

```
<HEAD>
```

Deskripsi dokumen

```
</HEAD>
```

```
<BODY> Isi dokumen </BODY>
```

</HTML>

2. PHP (Hyper Text Preprocessor)

PHP singkatan dari Hypertext Processor yang merupakan server-side programming, yaitu bahasa pemrograman yang diproses di sisi server. Fungsi utama PHP dalam membangun website adalah untuk melakukan pengolahan data pada database. Data website akan dimasukkan ke database, diedit, dihapus, dan ditampilkan pada website yang diatur oleh PHP.

Script dasar pada PHP :

```
<html>
```

```
<head>
```

```
<title>Judul</title>
```

```
</head>
```

```
<body>
```

```
<?php echo("Isi dokumen php"); ?>
```

```
</body>
```

```
</html>
```

2.9. CI Framework

CodeIgniter merupakan salah satu framework PHP yang bermodelkan 3 komponen utama MVC (Model, View, dan Controller) selain itu CI juga merupakan sebuah web application network yang memiliki sifat open source bisa digunakan untuk membuat dan membangun sebuah aplikasi php yang dinamis. Jika kita membangun sebuah website menggunakan CodeIgniter

maka website yang dibuat akan ringan dan juga cepat dalam proses loadingnya. Kelebihan lain dari CodeIgniter adalah implementasi kode yang lengkap dan disertai dokumentasi yang lengkap juga di dalamnya. Karena adanya dokumentasi yang lengkap ini, hal ini yang menjadikan CI menjadi pilihan framework pilihan

2.10. Use Case Diagram

Use case menggambarkan external view dari sistem yang akan kita buat modelnya (Prabowo Pudjo Widodo, 2011) Model use case dapat dijabarkan dalam diagram use case, tetapi perlu diingat, diagram tidak identik dengan model karena model lebih luas dari diagram. (Pooley, 2003:15). Use case harus mampu menggambarkan urutan aktor yang menghasilkan nilai terukur (Prabowo Pudjo Widodo, 2011).

2.11. Activity Diagram

Diagram activity menunjukkan aktivitas sistem dalam bentuk kumpulan aksiaksi, bagaimana masing-masing aksi tersebut dimulai, keputusan yang mungkin terjadi hingga berakhirnya aksi. Activity diagram juga dapat

menggambarkan proses lebih dari satu aksi dalam waktu bersamaan. “Diagram activity adalah aktifitas-aktifitas, objek, state, transisi state dan event. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas” (Haviluddin, 2011).

2.12. Sequence Diagram

“Sequence diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan use case diagram.” (Haviluddin, 2011).

2.2.8 Class Diagram

Kelas sebagai suatu set objek yang memiliki atribut dan perilaku yang sama, kelas kadang disebut kelas objek (Whitten, 2004:410). Class memiliki tiga area pokok yaitu : 1) Nama, kelas harus mempunyai sebuah nama. 2) Atribut, adalah kelengkapan yang melekat pada kelas. Nilai dari suatu kelas hanya bisa diproses sebatas atribut yang dimiliki. 3) Operasi, adalah proses yang dapat dilakukan oleh sebuah kelas, baik pada kelas itu sendiri ataupun kepada kelas lainnya.

Pengertian Aplikasi E-Commerce

Sebelum masuk ke Aplikasi E-Commerce yang kami buat, terlebih dahulu kami menjelaskan pengertian dari E-Commerce.

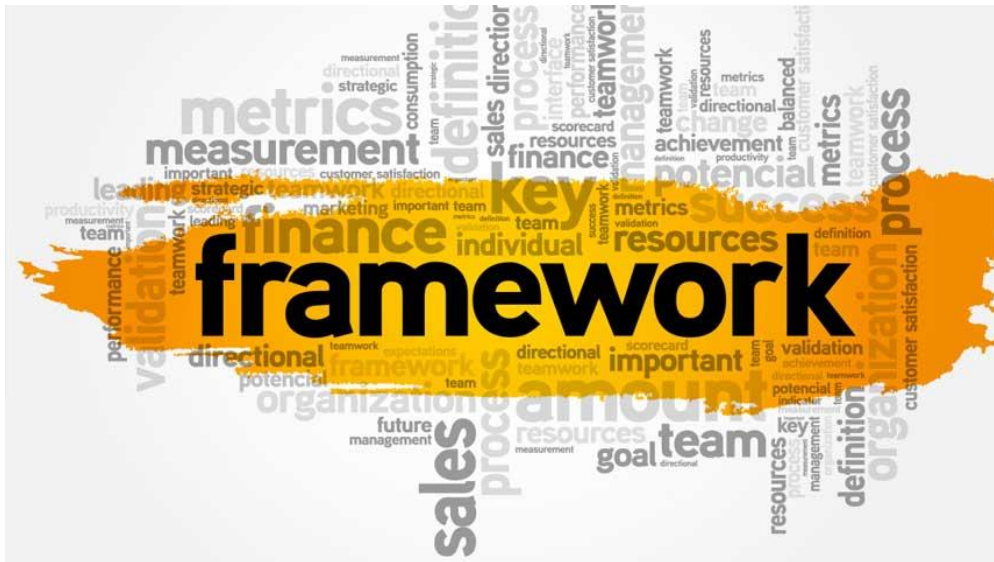
Pengertian E-Commerce adalah aktivitas penyebaran, penjualan, pembelian, pemasaran produk (barang dan jasa), dengan memanfaatkan jaringan telekomunikasi seperti internet dan jaringan komputer.

Arti E-commerce (*Electronic Commerce*) dapat juga didefinisikan sebagai aktivitas penggunaan teknologi informasi dan komunikasi pengolahan digital dalam melakukan transaksi bisnis untuk menciptakan, mengubah, dan mendefinisikan kembali hubungan antara penjual dan pembeli.

Disini kami akan menjelaskan Aplikasi E-Commerce yang kami buat,

1. Aplikasi E-Commerce yang kami buat adalah E-Commerce tentang penjualan Makanan dan Minuman, dimana di perkembangan Dunia Teknologi 4,0 kami tertarik untuk membuat Aplikasi ini.
2. Dengan kami merancang Aplikasi E-Commerce ini kami bisa membantu para pedagang Makanan dan Minuman untuk mempermudah menjual makanan dan minuman yang dia jual

FRAMEWORK



Gambar framework

Framework adalah sebuah kerangka kerja atau kumpulan data yang dapat membantu programmer dalam menjali sebuah masalah dalam menjalankan beberapa Bahasa pemrograman dan koneksi ke dalam sebuah database sehingga membantu programmer dalam memjalankan tugasnya.

Framework juga bisa disebut sebagai sebuah komponen yang dapat ditambahkan ke dalam berbagai system operasi seperti Microsoft windows, linux, dll. Selain itu framework juga dapat menyediakan sejumlah solusi dalam melakukan pemrograman untuk membantu dalam memenuhi kebutuhan

Pada sebuah framework ada beberapa macam jenis yang terdiri dari Bahasa pemrograman yang akan kita pakai. Bahasa pemrograman yang sering menggunakan framework antara lain PHP, Javascript dan Ruby dan masih banyak lagi Bahasa pemrograman yang sering dipakai untuk membuat sebuah program. Framework yang di gunakan pada E-Commerce ini adalah Code Igniter (CI) ada beberapa jenis framework juga yang memakai Bahasa pemrograman PHP seperti CakePHP, Symphony, Yii dan Kohana. Framework PHP banyak sekali programmer yang memakainya, diantara lain

yang sangat populer dikalangan programmer adalah framework Zend dan Code Igniter yang mudah dipahami oleh programer pemula karena strukturnya yang digunakan sangat sederhana dan lengkapnya data. Dan ini list framework PHP terbaik yang bisa digunakan oleh anda

Framework PHP Terbaik yang populer saat ini :

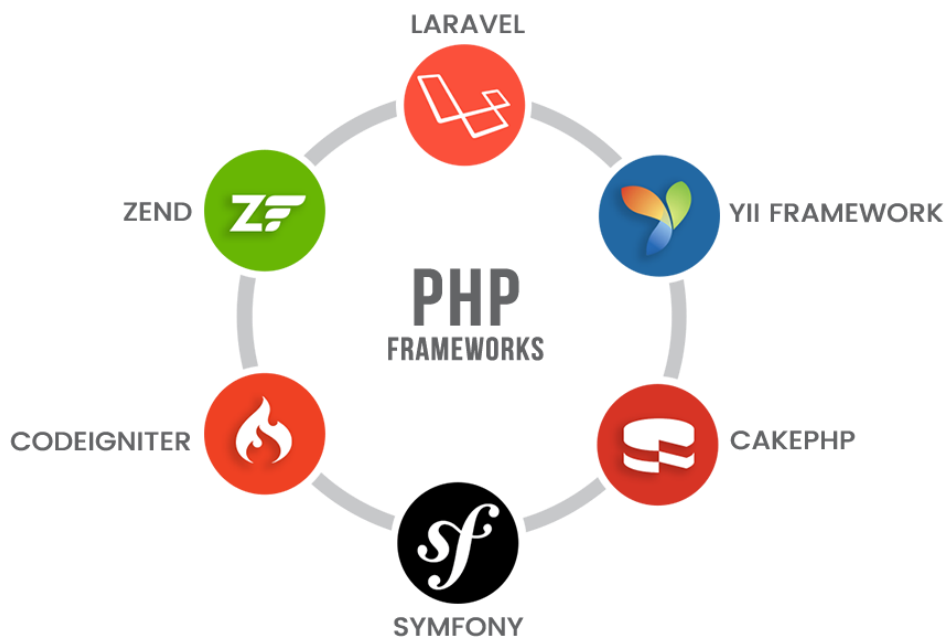
1. Yii
2. Code Igniter
3. CakePHP
4. Zend
5. Symphony
6. PHP Dev Shell
7. Prade
8. Akelos
9. QPHP
10. ZooP

Ada dua jenis Framework dalam dunia pemrograman yaitu Desktop Framework dan Web Framework, keduanya memiliki fungsi yang berbeda, Desktop Framework digunakan untuk membangun aplikasi berbasis desktop sedangkan Web Framework digunakan untuk membangun aplikasi berbasis web.

Pada Desktop Framework program yang dijalankan tidak berjalan langsung pada perangkat keras, akan tetapi aplikasi berjalan dilingkungan perangkat lunak dengan memanfaatkan Common Language Runtime. Sedangkan aplikasi yang dibangun menggunakan Web Framework juga tidak langsung dieksekusi oleh web server, aplikasi web terlebih dahulu diproses oleh core

Secara umum Framework tersusun dengan struktur MVC (Model View Controller) yang memungkinkan pengembang dapat mengelompokkan fungsi – fungsi seperti fungsi inputan, proses dan output dari sebuah aplikasi. Bagaimana dari sini sobat komputer pastinya sudah memahami apa itu Framework.

Ada beberapa fungsi yang harus anda ketahui dalam penerapan framework di dunia kerja atau yang lainnya. Pada sebuah framework yang kita pakai bisa mempercepat proses pembuatan aplikasi maupun di Desktop, Mobile maupun Web. Dapat membantu para developer dalam pembuatan, perencanaan dan pemeliharaan dari aplikasi tersebut. Framework juga dapat menghasilkan aplikasi yang stabil dan mudah digunakan, dikarenakan framework sudah diuji oleh beberapa programmer sehingga dapat membantu membaca program lebih mudah dan pencarian bugn dalam aplikasi dapat dengan mudah dicari. Memiliki tingkat keamanan yang lebih, hal ini dikarenakan Framework telah mengantisipasi celah – celah keamanan yang mungkin timbul. Sehingga mempermudah developer dalam mendokumentasikan aplikasi – aplikasi yang sedang dibangun.



PHP Framework berbeda jika dibandingkan dengan CMS (Content Management System), meski sama-sama meringankan dalam pembuatan web. Jika menggunakan CMS, kita tidak perlu pusing menulis script. Semuanya telah dibuat menjadi Fix dan kita hanya perlu mengatur bagian content dan interfacenya saja. Tidak demikian dengan Framework. Membangun sebuah aplikasi web dengan menggunakan framework, tetapi tetap harus menuliskan kode perintah PHP sesuai dengan ruang lingkungan yang disediakan oleh framework.

Dalam menggunakan framework php, kita menggunakan metode pengembangan berbasis “MVC”. MVC sendiri merupakan suatu metode untuk memisahkan bagian-bagian dari suatu web aplikasi. MVC adalah kependekan dari Model View Controller. MVC terdiri dari tiga bagian yaitu;

Model :

Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu kita dalam pengelolaan database seperti memasukkan data ke database, update data dan lain-lain.

View :

View adalah bagian yang mengatur tampilan ke user. Bisa di katakan berupa halaman web.

Controller :

Controller merupakan bagian yang menjembatani model dan view. Controller berisi script-script php yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman web.

Kelebihan Framework

1. Lebih cepat dan efisien

Jika Anda mengerjakan proyek besar, maka penggunaan framework dapat membantu mempercepat proses pengembangan. Pada umumnya, framework memiliki beragam fungsi dan plugin yang bisa Anda manfaatkan.

Dengan kerangka kerja ini, maka proses pengembangan proyek jauh lebih cepat daripada Anda harus menulis kode dari awal. Selain itu, Anda juga tidak perlu menulis berulang-ulang untuk kode yang bersifat repetitif.

2. Menghemat biaya

Sebagian besar kerangka populer bersifat open source dan gratis untuk digunakan. Biaya yang harus dikeluarkan oleh client Anda juga akan menjadi lebih kecil karena proses pengerjaan yang lebih simple dan lebih cepat.

3. Memperhatikan faktor keamanan

Framework populer telah banyak digunakan oleh banyak pengembang, dan kemungkinan adanya masalah keamanan atau pun bug telah diperbaiki. Selain itu, framework biasanya juga memiliki komunitas besar yang dapat berperan sebagai penguji jangka panjang. Setiap kali pengguna menemukan celah keamanan, maka mereka dapat memberi tahu tim untuk segera memperbaikinya.

Kelemahan Framework

1. Kurangnya pemahaman bahasa pemrograman

Jika Anda bekerja menggunakan kerangka kerja dan hanya mengetahui sedikit tentang bahasa pemrograman yang digunakan dibalikinya, maka Anda hanya mempelajari mengenai kerangka tersebut. Sehingga pemahaman mengenai bahasa pemrograman menjadi tidak berkembang.

2. Memiliki batasan

Dalam penggunaannya, kerangka ini juga memiliki beberapa batasan yang tidak dapat Anda modifikasi. Sehingga Anda harus bekerja sesuai dengan standar yang digunakan di dalamnya. Oleh karena itu, ketika

mengembangkan sebuah aplikasi, Anda harus menggunakan kerangka kerja yang sesuai dengan kebutuhan Anda.

3. Kode Publik

Karena bersifat publik, maka kode dapat digunakan siapa saja termasuk pihak-pihak yang mempunyai niat buruk. Mereka dapat mempelajari cara kerja kode untuk menemukan kelemahan untuk menyerang Anda.

CODE IGNITER



CodeIgniter adalah framework yang akan kita pakai karena aplikasi yang telah dibuat di dalam buku ini menggunakan framework CodeIgniter, sebelum membahas CodeIgniter sebelumnya kita telah membahas tentang apa itu Framework dan jenis-jenisnya.

Kita akan sedikit membahas kembali apa itu Framework adalah sebuah kerangka kerja atau kumpulan data yang dapat membantu programmer dalam menjali sebuah masalah dalam menjalankan beberapa Bahasa pemrograman

dan koneksi ke dalam sebuah database sehingga membantu programmer dalam menjalankan tugasnya.

Sedangkan untuk Framework CodeIgniter ini sudahh banyak tersedia sebuah library yang biasanya diperlukan dalam pengembangan sebuah website ataupun aplikasi yang berbasis di dalam web, di dalam CodeIgniter (CI) terdapat library untuk mengoperasikan database sampai library untuk masalah lainnya

Framework CodeIgniter ini juga berdasarkan design patternkonsep MVC Model, View, Controller). Secara garis besar Konsep MVC ini adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan Data (Model) dari tampilan (View) serta cara memprosesnya (Controller).

Framework Codeigniter ini pertama kali dibuat oleh Rick Ellis seorang musisi sekaligus juga seorang programmer, dan kemudian Framework ini dirilis pada tanggal 28 Februari 2006 dan sejak tahun 2014 sampai saat ini Framework Codeigniter ini telah dimiliki dan dikembangkan oleh British Columbia Institute of Technology (BCIT) situs resminya beralamat di <https://www.codeigniter.com/>

Design Pattern adalah kumpulan penjelasan mengenai metode-metode bagaimana cara menyelesaikan suatu masalah yang umum ditemui dalam proses perancangan perangkat lunak (Software Design). Design pattern merupakan petunjuk bagaimana cara menyelesaikan suatu masalah di seputar dunia software design.

MVC adalah sebuah metode pendekatan yang dapat ditempuh untuk memisahkan aplikasi menjadi tiga bagian, yaitu :

a) Model

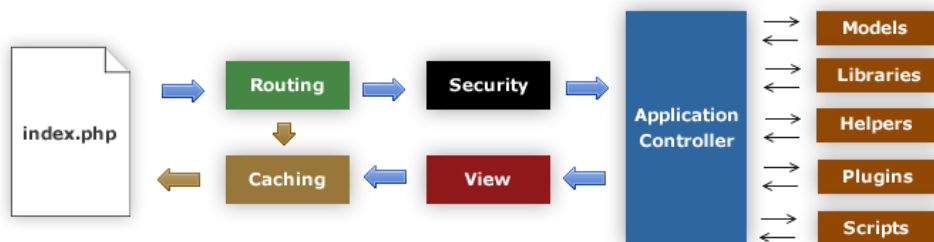
Model berfungsi untuk merepresentasikan data yang digunakan aplikasi. Contoh data misalnya database, RSS, atau data yang diperoleh dari pemanggilan API, dan aksi yang melibatkan operasi Create, Read, Update, dan Delete (CRUD) data.

b) View

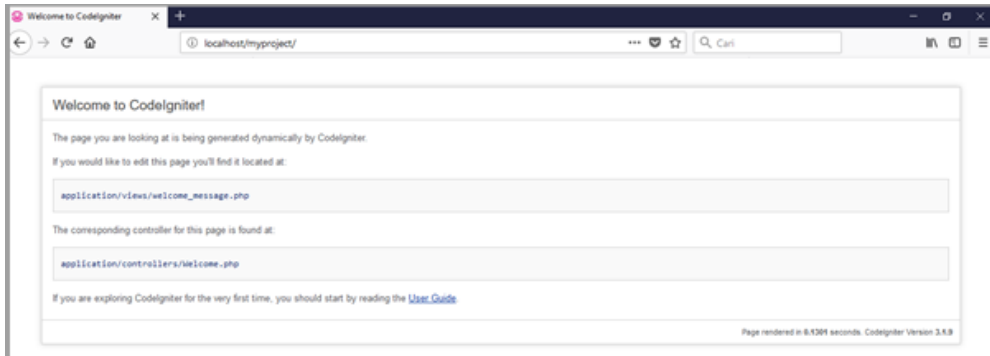
View adalah informasi yang ditampilkan kepada user melalui browser. Biasanya berupa file HTML atau kode PHP yang menyusun template untuk sebuah website. Pada codeigniter, view dapat berupa bagian-bagian sebuah halaman, template atau jenis lain dari halaman atau template

c) Controller

Controller adalah business logic yang bertugas sebagai jembatan antara Model dan View. Controller akan merespon HTTP request yang datang dari user (melalui browser), dari request ini controller akan menentukan apa yang harus dilakukan. Jika diperlukan, controller akan memanggil model untuk mengolah data, kemudian akan memanggil view yang sesuai untuk menampilkan data hasil olahan model dan akhirnya mengirim kembali halaman hasil olah data ke browser.



- G. Index.php berfungsi sebagai front controller, menginisialisasi base resource untuk menjalankan CodeIgniter.
- H. Router memeriksa HTTP request untuk menentukan apa yang harus dilakukan dengannya.
- I. Jika Cache aktif maka hasilnya akan langsung dikirimkan ke browser dengan mengabaikan aliran data normal.
- J. Security. Sebelum controller dimuat, HTTP request dan data yang dikirimkan user akan difilter untuk keamanan.
- K. Controller membuat model, core libraries, plugins, helper, dan semua resource yang diperlukan untuk memproses request.
- L. View yang dihasilkan akan dikirim ke browser. Jika cache aktif maka View akan disimpan sebagai cache dahulu sehingga para request berikutnya langsung ditampilkan



12. Selesai

Konfigurasi CodeIgniter

CodeIgniter dapat berjalan dengan konfigurasi default, tetapi untuk sebuah aplikasi yang nyata kita harus tetap melakukan konfigurasi, setidaknya pada bagian base_url dan router. Pengaturan base_url dan router sangat berguna ketika proses pengembangan aplikasi yang banyak menggunakan helper dan library.

File konfigurasi terletak dalam folder application/config. File yang terdapat pada direktori tersebut yang sering digunakan adalah file autoload.php, config.php, database.php dan routes.php.

5. File autoload.php. konfigurasi pada file ini bertujuan untuk menentukan sebuah sumber yang akan menjadi titik dari penyimpanan agar dapat dipanggil secara otomatis

File autoload['libraries'] :

```
$autoload['libraries'] = array();
/*
 * Auto-load Libraries
 *
 * These are the classes located in system/libraries/ or your
 * application/libraries/ directory, with the addition of the
 * 'database' library, which is somewhat of a special case.
 *
 * Prototype:
 *
 * $autoload['libraries'] = array('database', 'email', 'session');
 *
 * You can also supply an alternative library name to be assigned
 * in the controller:
 *
 * $autoload['libraries'] = array('user_agent' => 'ua');
 */
File autoload.php bagian libraries
```

File autoload['helper'] :

```

$autoload['helper'] = array();
/*
|-----
| Auto-load Helper Files
|-----
| Prototype:
|
| $autoload['helper'] = array('url', 'file');
|*/

```

File autoload.php bagian helper

6. File config.php. pada file ini kita akan mengkonfigurasi secara umum dalam CodeIgniter.

```

$config['base_url'] = '';
/*
|-----
| Base Site URL
|-----
|
| URL to your CodeIgniter root. Typically this will be your base URL,
| WITH a trailing slash:
|
| http://example.com/
|
| WARNING: You MUST set this value!
|
| If it is not set, then CodeIgniter will try guess the protocol and path
| your installation, but due to security concerns the hostname will be set
| to $_SERVER['SERVER_ADDR'] if available, or localhost otherwise.
| The auto-detection mechanism exists only for convenience during
| development and MUST NOT be used in production!
|
| If you need to allow multiple domains, remember that this file is still
| a PHP script and you can easily do that on your own.
|*/

```

File config.php bagian base url

\$config['base_url'] – konfigurasi ini berisikan alamat url pada sebuah aplikasi yang akan dibuat. Sebagai contoh “C:xampp/htdocs/codeigniter”

7. File database.php. disini kita akan mengkonfigurasi koneksi database. Pada aplikasi yang digunakan dalam buku ini kita menggunakan database mysql yang terdapat pada aplikasi XAMPP

```

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => '',
    'password' => '',
    'database' => '',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);

```

File database.php

- E. Hostname menggunakan localhost karena di sini kita menggunakan database mysql dari aplikasi xampp yang terinstal di computer local.
 - F. Username di dalam sini kita isikan sesuai username yang telah kita buat didalam mysql atau secara default bisa diisikan dengan “root”
 - G. Password sesuaikan dengan mysql masing-masing
 - H. Database ini merupakan nama dari database yang akan kita pakai di dalam sebuah aplikasi yang dibuat
8. File routes.php. Konfigurasi routing digunakan untuk memetakan permintaan atau request kedalam controller didalam website yang dibuat. Misalnya kita membuka alamat <http://www.nama-website.com>, permintaan tersebut tidak menyertakan nama controller yang ingin dibuka tetapi kita bisa secara default mengarahkannya agar secara otomatis akan membuka controller sesuai yang definisikan. Untuk melakukan konfigurasi routing buka file konfigurasi routes.php. Settingan utama yang ada adalah sebagai berikut:

```
$route['default_controller'] = 'welcome';  
$route['404_override'] = '';  
$route['translate_uri_dashes'] = FALSE;
```

File routes.php

Artinya secara default semua permintaan yang tidak menyertakan nama controllernya akan diarahkan untuk membuka controller “welcome”. Sehingga saat alamat <http://www.nama-website.com> dibuka secara otomatis akan terbuka

Didalam codeigniter ada beberapa folder yang menjadi tempat bagi script php yang di simpan, pada setiap folder pun punya fungsi masing-masing untuk penempatan script seperti config, dll. Disini sebelum memberikan contoh sederhana alangkah baiknya memoerkenalkan setiap folder yang ada di dalam CodeIgniter

Folder Application adalah folder dasar yang meyimpan data aplikasi yang sedang dibuat

Folder Cache adalah folder yang menyimpan semua cache yang ada di aplikasi yang telah dibuat oleh cache library

Folder Config adalah folder yang berguna untuk menyimpan data atau informasi dari konfigurasi aplikasi yang telah dibuat seperti autoload, database, routes dan lainnya.

Folder Controller, adalah folder yang menyimpan controller dari aplikasi yang dapat digunakan untuk menyusun aktivitas program .

Folder Core, adalah folder untuk memperluas class class inti codeigniter.

Folder Helpers, merupakan folder untuk menyimpan helpers.

Folder Hooks, merupakan folder untuk menyimpan hooks untuk mengubah alur fungsi dari core Codeigniter

Folder Language, merupakan folder untuk menyimpan bahasa - bahasa yang akan digunakan.

Folder Libraries, merupakan folder untuk menyimpan library.

Folder Logs, merupakan folder untuk menyimpan semua error log apabila error log diaktifkan.

Folder Models, merupakan folder untuk menyimpan models yang akan mendefinisikan tabel dari database yang dapat kita gunakan oleh Controller yang kita buat untuk mengakses database.

Folder Third_party, merupakan folder untuk menyimpan fungsi fungsi tambahan dalam cara kerja codeigniter.

Folder Views, merupakan folder untuk menyimpan tampilan dari aplikasi yang kita buat.

Folder System, merupakan folder untuk menyimpan sistem inti dari Codeigniter.

Setelah saya akan memberikan contoh sederhana berupa menampilkan sebuah script yang akan memanggil hello word. Seperti yang telah disampaikan pada materi sebelumnya langkah pertama kita buka file yang telah kita buat di dalam /xampp/htdocs/namafolder kedalam aplikasi pemrograman seperti sublime atau yang lainnya, application seperti yang ditandai dengan kotak berwarna merah. Didalam folder application terdapat banyak folder, tetapi yang menjadi perhatian kita saat ini adalah folder controllers, views dan models.

Langkah pertama

Buatlah file bernama helloworld.php dan di simpan di dalam folder controllers, kemudian ketikkan script seperti dibawah ini:

```

1  <?php
2
3  class HelloWorld extends Controller {
4
5      function HelloWorld()
6      {
7          parent::Controller();
8      }
9
10     function index()
11     {
12         $this->load->view('helloworld_view');
13     }
14
15 }

```

Fungsi HelloWorld() merupakan fungsi constructor pada class HelloWorld. Permasalahan ini bisa dijelaskan pada tutorial lain yang membahas mengenai Object Oriented Programming pada bahasa pemrograman PHP. Singkatnya nama class harus sama dengan nama fungsi constructor-nya. Dan juga nama file harus sama dengan nama classnya. Pada script `$this->load->view('helloworld_view')` berfungsi `index()` memanggil sebuah view bernama `helloworld_view.php` dan jika kita memanggil di dalam browser dengan `http://localhost/CodeIgniter/index.php/HelloWorld` melalui browser, maka hasilnya akan muncul error pada layar browser. Mengapa terjadi error? Karena kita belum membuat file view yang bernama `helloworld_view.php` pada folder `views`

Langkah kedua

Buatlah file bernama `helloworld_view.php` dan taruh dalam folder `views`. Kemudian ketikkan sintaks seperti dibawah :

```

<html>

<head>

<title>Hello World</title>

</head>

<body>

<h1>Hello Word</h1>

```


</body>

</html>

Akses di dalam browser dengan URL maka setelah itu di dalam browser akan muncul script hello world

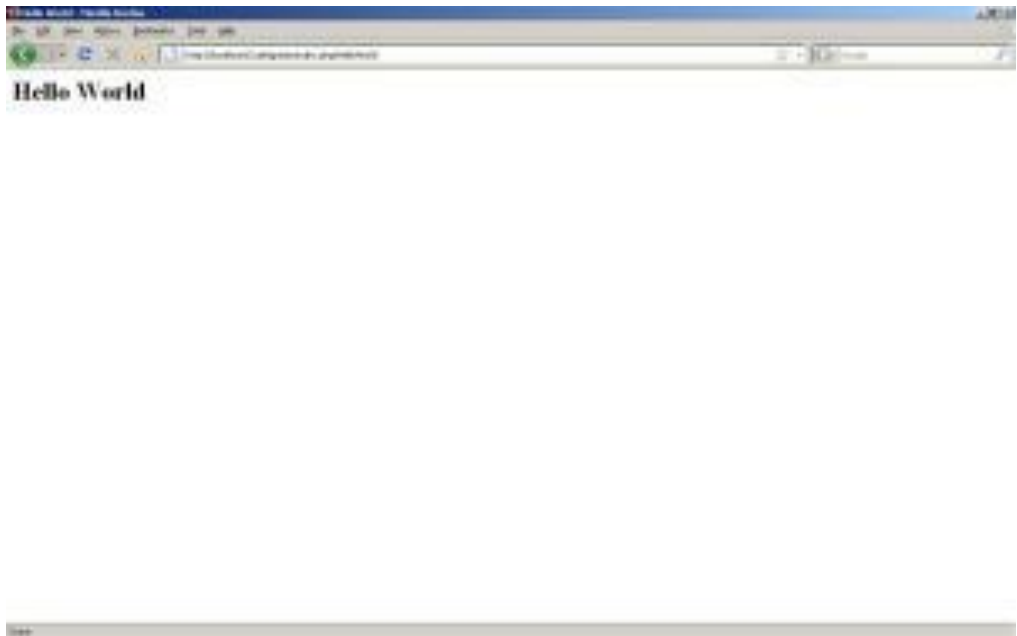


Setelah itu jika telah berhasil dengan cara yang di atas, cara yang kedua adalah berupa menampilkan sebuah variable dari controller ke halaman view dengan mengubah script pada file helloworld.php pada folder controller seperti yang ada di dalam gambar ini

```
1 <html>
2 <head>
3 <title><? = $title ?></title>
4 </head>
5 <body>
6 <h1><? = $header ?></h1>
7 </body>
8 </html>
```

Setelah itu buka kembali halaman browser untuk membuka script yang telah diubah tersebut dengan menuliskan

<http://localhost/CodeIgniterindex.php/helloworld>



Dari pembahasan yang telah di dapat pada materi ini menarik kesimpulan bahwa framework CodeIgniter ini merupakan frameworkPHP yang kuat dengan jalin yang sangat kecil, dibuat untuk memenuhi program yang akan kita buat atau pengembangan yang membutuhkan kerangka sederhana dan elegan untuk membuat aplikasi berbasis web dengan fitur yang lengkap, yang banyak memiliki keunggulan pada penggunaan untuk framework yang ada.

BOOTSTRAP



Bootstrap adalah kerangka kerja dari front end yang dibuat untuk pengembangan web yang lebih dinamis dan mudah. Bootstrap mencakup beberapa template desain berbasis pada HTML dan CSS untuk menentukan sebuah tipografi, bentuk, tabel, navigasi, tombol, gambar dan masih banyak yang lainnya, bootstrap data memberikan pengalaman pada programmer dalam membuat desain yang responsif pada front end aplikasi yang telah dibuat dengan mudah. Adapun pengertian Bootstrap adalah sebuah alat bantu untuk membuat sebuah tampilan halaman website yang dapat mempercepat pekerjaan seorang pengembang website ataupun pendesain halaman website.

Pada saat ini untuk pengembangan front-end para web developer menggunakan bootstrap sebagai pilihan agar bisa dengan mudah terlaksana dan cepat dalam pengerjaan. karena mereka hanya harus menambahkan class tertentu. seperti membuat grid, tombol, navigasi, dan lain sebagainya. Pada bootstrap juga sudah tersedia berbagai komponen dasar class interface yang bertujuan untuk membuat tampilan pada website menjadi lebih menarik. untuk dilihat oleh para konsumen dari web tersebut

Bootstrap juga menyediakan kumpulan kelas yang sudah siap untuk digunakan. Misalnya ketika Anda hendak membuat tombol tapi Anda tidak harus merancang dan juga mengetik syntax css. Dengan bootstrap, pembuatan

atau penambahan tombol tersebut bisa Anda wujudkan dalam waktu yang singkat. Anda hanya perlu menambah class 'btn'. Selanjutnya Anda bisa menambahkan class btn tambahan.

Perbedaan Bootstrap 3 & Bootstrap 4

Bootstrap sampai dengan saat ini masih memperbarui sistemnya. Tidak hanya berhenti di angka 3, tapi sudah masuk ke versi 4. Pada versi 4 ini ada beberapa pembaharuan lebih banyak berhubungan dengan tema. Contohnya saja perubahan tampilan styling, warna, fonts, dan lain sebagainya.

1. Global

Bootstrap menyediakan ukuran font global (default). Bootstrap 3 menggunakan ukuran font 14px, sedangkan pada Bootstrap 4 diganti dengan ukuran yang lebih besar, yaitu 16px.

Ketika penggunaan Bootstrap 3 dan di upgrade menjadi versi 4 tulisan pada tampilan akan terlihat lebih besar dibandingkan dengan versi sebelumnya.. ketika beralih dari Bootstrap 3 ke versi 4, perlu penyesuaian dalam menentukan ukuran pixel menjadi lebih kecil

2. Grid

Bootstrap 4 tidak memerlukan lagi col- pada setiap class grid. Ini berbeda dengan Bootstrap 3 yang sebelumnya menggunakan col- di depan, misalnya ketika menggunakan class "col-md-offset-3" untuk membatasi kolom. Bootstrap 4 menghilangkan col-xs dan diwakilkan oleh col-sm untuk perangkat super kecil (extra small)

3. Tabel

Bootstrap 4 mendukung gaya untuk header atau judul tabel – yang sebelumnya di Bootstrap 3 tidak didukung. Selain mendukung gaya header, Bootstrap 4 juga dapat memberikan warna tabel header menjadi gelap (.thead-dark) atau terang (.thead-light).

Fungsi Dan Kegunaan Bootstrap



Fungsi tertentu dari bootstrap memiliki sebuah kegunaan atau fungsi yang sudah mempermudah pekerjaan web developer dalam mengolah sebuah website. Beberapa fungsi yang telah dimiliki oleh bootstrap antara lain adalah:

1. Bootstrap dapat membantu dalam membuat atau mendesain sebuah website yang dinamis dan juga responsive. Sehingga, tampilan dari website yang telah dibuat melalui bootstrap dapat menyesuaikan dengan perangkat yang digunakan oleh pengguna itu sendiri
2. Bootstrap juga akan membantu di dalam membuat dan mengembangkan sebuah website yang bersifat dinamis maupun statis.
3. Bootstrap juga sudah menyediakan berbagai class yang berguna bagi developer dalam sebuah website agar mudah diakses

4. Bootstrap juga membantu di dalam melakukan pengaturan pada layout sebuah page di website dilengkapi dengan fitur grid.
5. menambahkan CSS atau class sendiri. Dengan begitu, developer dapat mendesain website sehingga memiliki tampilan yang lebih variatif.

Kelebihan dan Kekurangan Bootstrap



Pada setiap program pasti mempunyai kekurangan dan kelebihan termasuk juga bootstrap. Adapun beberapa hal yang termasuk ke dalam kelebihan dan juga kekurangan pada bootstrap

Kelebihan Bootstrap

1. Bisa digunakan untuk membantu mempercepat ketika membuat front-end pada website.
2. Bootstrap memiliki tampilan yang modern sehingga bisa membuat website Anda memiliki tampilan yang menarik.
3. Bootstrap lebih lengkap. Hal ini karena ia sudah mencakup CSS, HTML, dan juga Javascript.
4. Responsive Layout dan 12-column grid system. Dengan Responsive Layout maka aplikasi web yang didesain dengan menggunakan Bootstrap akan langsung menyesuaikan dengan lebar dari media perambahnya.
5. tampilan web akan tetap rapi dibuka dengan media apapun baik itu handphone, tablet, laptop ataupun PC desktop

Kekurangan Bootstrap

1. Dikarenakan penggunaan CSS3, akibatnya jika pengguna menambahkan beberapa gambar bootstrap tidak selalu menerima gambar.
2. Dikarenakan terlalu fokus pada coding, sehingga membuat kita kurang begitu kreatif untuk mendesain website sesuai dengan apa yang benar-benar diinginkan.
3. Bootstrap masih belum mampu untuk dapat menampilkan tampilan website yang sama di seluruh browser. beberapa browser tertentu yang dapat menampilkan tampilan website dari bootstrap sesuai dengan layar perangkat yang dipakai

Install Bootstrap



Penerapan bootstrap pada websites sebelumnya harus melakukan penginstalan pada computer dengan 2 cara online dan offline. Bisa dengan composer, bower. Langkah pertama sebelum melakukan penginstalan adalah melakukan install package seperti composer, bower atau npm

Menggunakan bower `bower install bootstrap`

Menggunakan npm `npm install bootstrap`

Menggunakan composer `composer require twbs/bootstrap`

Cara yang banyak dilakukan oleh pengguna adalah dengan cara offline atau online

Menginstal bootstrap online

Bisa dengan memilih dari 3 contoh yang akan diberikan untuk melakukan penginstalan bootstrap pada computer dengan menaruh script letakan pada bagian `<head>` pada halaman HTML

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css
" integrity="sha384-
```

```
1q8mTJOASx8j1Au+a5WDVnPi2lkFfwWEAa8hDDdjZlpLegxhjVME1fgj  
WPGmkzs7" crossorigin="anonymous">
```

```
<!-- Optional theme -->
```

```
<link rel="stylesheet"  
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap-  
theme.min.css" integrity="sha384-  
fLW2N01lMqjakBkx3l/M9EahuwPsfENvV63J5ezn3uZzapT0u7EYsXMjQ  
V+0En5r" crossorigin="anonymous">
```

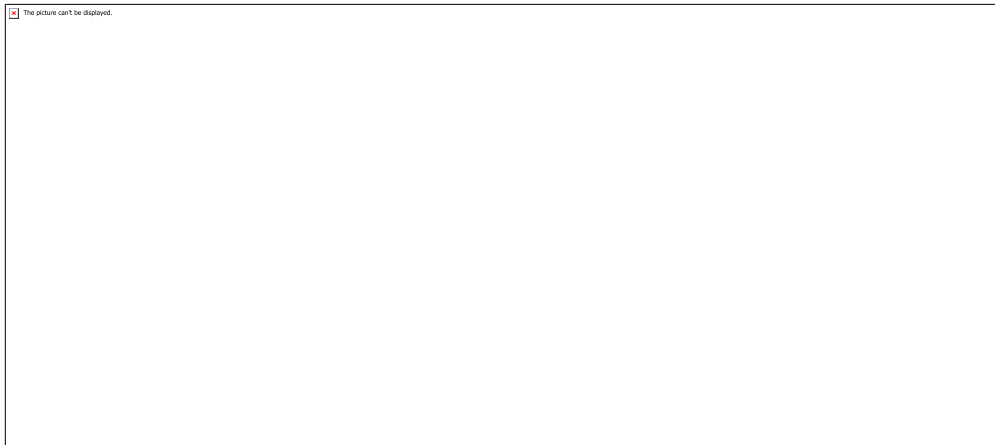
```
<!-- Latest compiled and minified JavaScript -->
```

```
<script  
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"  
integrity="sha384-  
0mSbJDEHialfmuBBQP6A4Qrprq5OVfW37PRR3j5ELqxs1yVqOtnepnH  
VP9aJ7xS" crossorigin="anonymous"></script>
```

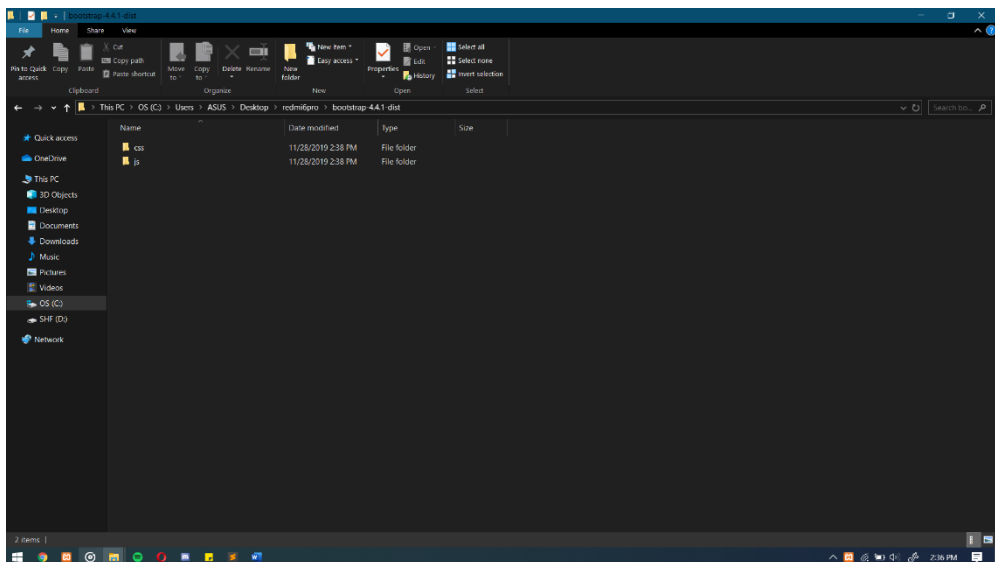
Menginstal bootstrap offline

Untuk menginstall boot strap secara offline, langkah pertama yang perlu anda lakukan adalah mendownload file bootsrap melalui website

<https://getbootstrap.com>



Kemudian, pilih tombol download untuk mendownload file bootstrap. Nah, setelah mendownload file tersebut dalam bentuk zip. Ekstrak file tersebut dalam folder baru. langkah terakhir adalah kita perlu membuat file .html. Buka text editor kalian, kemudian paste kode dibawah ini.



```
<!Doctype html>
```

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

```
<head>
```

```
<meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<title>Belajar Bootstrap</title>

<link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">

<script src="bootstrap/js/jquery-3.2.1.js"></script>

<script src="bootstrap/js/bootstrap.min.js"></script>

</head>

<body>

<h1> Instal Bootstrap Berhasil</h1>

</html>
```

Bootstrap merupakan suatu aplikasi dimana tampilan web akan tetap rapih dibuka dengan media apapun baik itu seperti handphone, tablet, laptop ataupun PC desktop. . Dengan menggunakan bootstrap, web developer bisa dengan mudah membuat halaman web yang responsive. dengan bootstrap juga bisa membangun web dinamis ataupun statis.

DATABASE



Database atau basis data adalah sebuah kumpulan data yang disimpan secara sistematis di dalam komputer untuk dapat diolah atau dimanipulasi menggunakan aplikasi database atau perangkat lunak yang dapat menghasilkan informasi. Basis data meliputi terdefinisi sebagai tipe data, struktur data dan juga pembatasan pada data yang akan disimpan

Basis data juga merupakan aspek penting yang meliputi sistem informasi karena berfungsi sebagai tempat penyimpanan data yang akan diolah lebih lanjut oleh sistem yang ada. Basis data menjadi sangat penting karena dapat mengorganisasi data yang diterima, menghindari adanya data yang sama berulang kali, dan menghindari relasi antar data yang tidak diinginkan. Proses untuk pemasukan data juga mengambilnya untuk di transfer sebagai media penyimpanan data pun memerlukan sebuah software yang disebut juga sebagai sistem manajemen basis data (database management system | DBMS). DBMS merupakan sistem perangkat lunak yang memungkinkan pengguna untuk mengelola basis data untuk memeliharanya, mengontrol dan mengakses data secara praktis dan efisien.

Dengan kata lain, akses ke semua basis data dapat ditangani oleh DBMS. DBMS ini juga menjadi lapisan yang akan menghubungkan basis data dengan program yang telah di buat oleh pengguna dapat digabungkan di dalam aplikasi untuk memastikan bahwa data tetap terorganisasi secara konsisten dan dapat diakses dengan mudah.

Ada beberapa fungsi yang harus ditangani DBMS seperti pendefinisian data, menangani permintaan pengguna untuk mengakses data, memeriksa sekuriti dan integriti data yang didefinisikan oleh DBA (Database Administrator), menangani kegagalan dalam pengaksesan data yang disebabkan oleh kerusakan sistem maupun media penyimpanan (disk) dan juga menangani unjuk kerja semua fungsi secara efisien.

Untuk lebih memahami tentang perbedaan basis data dan DBMS, Anda dapat membaca artikel Memahami Basis Data (Database), DBMS dan Program Aplikasi Di MySQL

Pembagian basis data.

Menurut jenisnya, basis data dapat dibagi menjadi:

1. Basis data flat-file.

Basis data ini sangat cocok untuk data berukuran kecil dan dapat dirubah dengan mudah. Pada awalnya, basis data flat-file ini tersusun dari sekumpulan data string yang ada di dalam satu atau lebih file yang dapat diuraikan untuk mendapatkan informasi yang dapat disimpan. Basis data flat-file akan menjadi sangat rumit apabila digunakan untuk menyimpan data dengan struktur kompleks.

Ada beberapa kendala pada penggunaan basis data jenis flatfile ini adalah rentan pada file yang akan korupsi data digunakan atau dimodifikasi juga adanya duplikasi data yang mungkin sulit dihindari. Salah satu tipe basis data flat-file adalah file CSV yang menggunakan pemisah koma untuk setiap nilainya.

2. Basis data relasional.

Basis data ini mempunyai sebuah struktur yang lebih logis dan ekonomis terkait dengan cara menyimpan data tersebut. Kata “relasional” berasal dari kata kenyataan bahwa table yang ada harus berlerasi akan dapat membantu basis data untuk menghubungkan tabel yang ada satu dengan yang lainnya. Basis data relasional menggunakan sekumpulan tabel dua dimensi yang masing-masing tabel tersusun atas baris (tupel) dan kolom (atribut).

Untuk menghubungkan antara dua atau lebih tabel, digunakan key (atribut kunci) yaitu primary key di salah satu tabel dan foreign key di tabel yang lain. Saat ini, basis data relasional menjadi pilihan utama karena keunggulannya. Program aplikasi untuk mengakses basis data relasional menjadi lebih mudah dibuat dan dikembangkan dibandingkan dengan penggunaan basis data flat-file.

Beberapa kekurangan yang mungkin dirasakan di basis data jenis ini adalah implementasi yang lebih sulit untuk data dalam jumlah besar dengan tingkat kompleksitasnya yang tinggi. Selain itu, proses pencarian informasi juga menjadi lebih lambat karena perlu menghubungkan tabel-tabel terlebih dahulu apabila datanya tersebar di beberapa tabel.

Namun, terlepas dari beberapa kekurangannya, basis data relasional telah digunakan secara luas. Saat ini, basis data relasional telah banyak dimanfaatkan oleh perusahaan-perusahaan dari skala kecil, menengah hingga besar. Beberapa basis data ternama yang ada saat ini, baik yang berasal dari sumber terbuka (open source) atau yang komersil, adalah juga basis data relasional.

3. Database Warehouse

Jenis database ini adalah database warehouse atau bisa disebut juga data warehouse (DW atau DWH). Database warehouse adalah sistem database yang lebih digunakan kepada pelaporan dan analisis data. Database warehouse menjadi tempat menyimpan repository sentral dari satu database atau lebih. Database yang berada di warehouse akan diunggah dari sistem operasional yang ada seperti pada sistem pemasaran atau penjualan. Contoh warehouse database adalah Microsoft SQL Server,

Pengertian Database Menurut Para Ahli

Beberapa ahli di bidang teknologi informasi pernah menjelaskan tentang definisi database, diantaranya adalah:

1. Jogiyanto

Menurut Jogiyanto pengertian database adalah kumpulan informasi dan data yang saling berhubungan satu sama lain, dimana data tersebut tersimpan di simpanan luar komputer dan diperlukan software tertentu untuk memanipulasinya.

2. S. Atte

Menurut S. Atte pengertian database atau basis data adalah sebuah koleksi data-data yang saling berhubungan yang ada dalam suatu organisasi atau enetrprise dengan berbagai penggunaan.

3. Gordon C. Everest

Menurut Gordon C. Everest, definisi database adalah suatu kumpulan data yang bersifat mekanis, terbagi, terdefinisi secara formal, dan terkontrol. Pengontrolan tersebut terpusat pada suatu organisasi.

4. Fabbri dan Schwab

Menurut Fabbri dan Schwab definisi database atau basis adalah suatu sistem berkas terpadu yang dirancang khusus untuk dapat meminimalkan pengulangan atau redundancy data.

5. C.J. Date

Menurut C.J. Date pengertian basis data adalah koleksi data/ informasi operasional yang sengaja disimpan dan juga digunakan oleh sistem aplikasi sebuah organisasi.

6. Toni Fabbri

Menurut Toni Fabbri pengertian database atau basis data adalah suatu sistem file-file dan data yang terintegrasi dimana file (baca: pengertian file) dan data tersebut yang mempunyai sebuah primary key untuk melakukan pengulangan data.

7. Abdul Kadir

Menurut Abdul Kadir pengertian database atau basis data adalah suatu pengorganisasian sekumpulan data yang saling terhubung sehingga memudahkan aktivitas untuk memperoleh informasi.

Fungsi Database

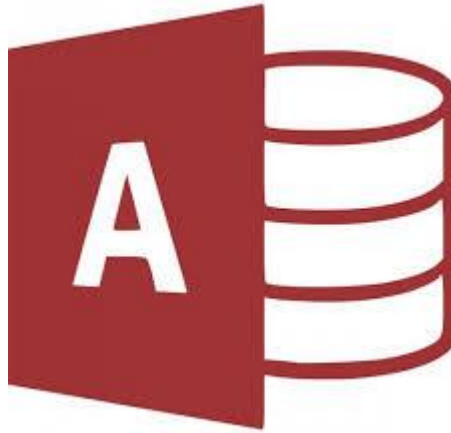
Dalam pemrogram aplikasi atau web sangat diperlukan sebuah tempat penampungan data atau basisdata semuanya memiliki fungsi dan spesifikasi yang berbeda-beda. Adapun berikut beberapa fungsi database yang perlu Anda ketahui:

1. Database berfungsi sebagai alat pengelompokan data dan untuk mempermudah dalam proses indentifikasi data. Database akan menampilkan data yang sesuai dengan permintaan dari program atau user dengan memanggil informasi melalui proses yang cepat dengan bantuan sebuah software Database Management Systems (DBMS).
2. Software DBMS akan menghindarkan data ganda dan inkonsistensi pada database. Database akan memberikan notif secara langsung jika terjadi duplikasi data.
3. memudahkan dalam akses, edit, tambah, delete dan penyimpanan data. Untuk mempermudah user dalam pengelolaan data dan semua kegiatan dalam jumlah yang besar
4. Menjaga kualitas database yang masuk agar sesuai dengan saat proses entry data dan setelah entry.
5. Menjadi solusi untuk mengatasi masalah penyimpanan data konvensional yang memerlukan ruang yang besar dan memakan biaya banyak.

Jenis-Jenis Software Database

Software database merupakan salah satu jenis software yang juga menjadi bagian yang sangat penting didalam komputer. Ada berbagai software database yang bisa menjadi bagian dalam pengolahan pada komputer.

1. Microsoft Access



Software database ini adalah yang paling sering digunakan oleh prmbuat aplikasi dalam pengelolaan datanya. Microsoft access sangat cocok digunakan bagi sebagian besar koomputer yang relasional. Selain itu, dapat memilih sistem database ini karena sangat ringan digunakan dan format datanya sangat umum sehingga memudahkan pembacaan.

2. Oracle



Software database ini sangat mampu untuk menyimpan data dengan ukuran yang maksimum hingga tera byte. Oracle peling banyak digunakan oleh perusahaan karane dapat diaksek dengan gratis sehingga bagi perusahaan yang baru sangat mudah untuk mengembangkan data dari perusahaan.

Itulah pembahasan mengenai pengertian database beserta fungsi dan jenis-jenis database yang paling umum digunakan. Dan basisdata merupakan hal yang penting bagi sebuah program karena dapat membantu dalam pengelolaan data

MYSQL



MySQL adalah sebuah database management system (manajemen basis data) menggunakan perintah dasar SQL (Structured Query Language) yang cukup terkenal. Database management system (DBMS) MySQL multi pengguna dan multi alur.

MySQL adalah DBMS yang open source dengan dua bentuk lisensi, yaitu Free Software (perangkat lunak bebas) dan Shareware (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi MySQL adalah database server yang gratis dengan lisensi GNU General Public License (GPL) sehingga dapat Anda pakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada.

Seperti yang sudah disinggung di atas, MySQL masuk ke dalam jenis RDBMS (Relational Database Management System). Maka dari itu, istilah semacam baris, kolom, tabel, dipakai pada MySQL. Contohnya di dalam MySQL sebuah database terdapat satu atau beberapa tabel.

Sejarah MySQL

MySQL AB merupakan perusahaan komersial Swedia yang mensponsori dan yang memiliki MySQL. Pendiri MySQL AB adalah dua orang Swedia yang bernama David Axmark, Allan Larsson dan satu orang Finlandia bernama Michael “Monty”. Setiap pengguna MySQL dapat menggunakannya secara bebas yang didistribusikan gratis dibawah lisensi GPL(General Public License) namun tidak boleh menjadikan produk turunan yang bersifat komersial.

Pada saat ini MySQL merupakan database server yang sangat terkenal di dunia, semua itu tak lain karena bahasa dasar yang digunakan untuk mengakses database yaitu SQL. SQL (Structured Query Language) pertama kali diterapkan pada sebuah proyek riset pada laboratorium riset San Jose, IBM yang bernama system R. Kemudian SQL juga dikembangkan oleh Oracle, Informix dan Sybase. Dengan menggunakan SQL, proses pengaksesan database lebih user-friendly dibandingkan dengan yang lain, misalnya dBase atau Clipper karena mereka masih menggunakan perintah-perintah pemrograman murni.

Fungsi – Fungsi Dalam MySQL

Aplikasi database MySQL memiliki sangat banyak rumus yang bisa digunakan oleh para programmer. Mereka yang masih pemula mungkin menganggap bahwa hal ini berat untuk dilakukan, padahal sebenarnya tidak perlu untuk semuanya dihapalkan. Minimal para pemula mengetahui beberapa fungsi dasar yang sering digunakan dalam pemrograman menggunakan mysql.

beberapa fungsi yang sering digunakan tersebut :

1. Sum, Fungsi ini untuk menjumlahkan field yang berisi data angka atau yang dalam bahasa pemrograman sering disebut integer.

```
SELECT SUM(barang_harga) AS tot_harga FROM tbl_barang;
```

2. Count, Hampir mirip dengan fungsi Sum, tapi untuk rumus ini digunakan menghitung baris (record) dalam satu atau lebih tabel.

```
SELECT COUNT(nama_field) AS nama_alias FROM nama_table;
```

3. Min, Fungsi ini bisa digunakan oleh para programmer untuk menyeleksi nilai minimal dari field dengan data integer. Jika diterapkan polanya hampir sama dengan rumus Sum ataupun Count.

```
SELECT MIN(barang_harga) AS tot_harga FROM tbl_barang;
```

4. Max, Fungsi ini digunakan oleh para programmer untuk menyeleksi nilai tertinggi dari field berdata integer.

```
SELECT MAX(barang_harga) AS tot_harga FROM tbl_barang;
```

5. Now, Bisa digunakan untuk mengambil waktu aktual. Waktu tersebut berpijak pada pengaturan tanggal dan waktu di komputer server.

```
SELECT NOW() AS waktu_sekarang;
```

6. Day, Fungsi ini digunakan untuk mengambil hari, khusus pada field-field dengan data date atau datetime.

```
SELECT DAY(jual_tanggal) AS hari FROM tbl_jual;
```

7. Avg, Fungsi ini digunakan untuk menghitung nilai rata-rata dari field yang berdata integer.

```
SELECT AVG(barang_harga) AS tot_harga FROM tbl_barang;
```

8. Curdate, Fungsi ini digunakan untuk mengambil data tanggal pada hari tersebut.

```
SELECT CURDATE() AS tanggal_sekarang;
```

9. Year, Seperti artinya dalam bahasa inggris yang berarti tahun, fungsi ini bisa digunakan oleh programmer untuk menampilkan data tahun pada field yang bertipe date atau datetime.

```
SELECT YEAR(jual_tanggal) AS thn_jual FROM tbl_jual;
```

10. Date_format, Sebagaimana dijelaskan dalam poin diatas bahwa aplikasi ini bersifat open source atau terbuka terhadap pengembangan. Hal ini termasuk dalam penanggalan juga bisa dicustomize oleh programmer dengan fungsi date_format ini.

```
SELECT DATE_FORMAT(jual_tanggal,'%d %M %Y')
```

11. Date, Fungsi ini bisa digunakan untuk mengambil tanggal pada field dengan data datetime.

```
SELECT DATE(jual_tanggal) AS tgl_jual FROM tbl_jual;
```

Kelebihan dan Kekurangan MySQL

Meskipun menjadi database yang cukup populer, MySQL tentu mempunyai beberapa kelebihan dan kekurangan dibandingkan dengan database server lainnya. Salah satu kekurangan MySQL adalah performanya turun di saat beberapa database manajemen sistem mampu bekerja baik pada pengelolaan database yang besar.

Kelebihan MySQL

1. Mendukung Integrasi Pada Bahasa Pemrograman Lain.

Website atau perangkat lunak terkadang dikembangkan dengan cara menggunakan berbagai macam bahasa pemrograman, pengguna tidak perlu khawatir pada penggunaan MySQL. Maka dari itu, MySQL dapat membantu untuk mengembangkan perangkat lunak yang lebih efektif dan tentu lebih mudah dengan integrasi antara bahasa pemrograman.

2. Mendukung Multi User.

MySQL dapat dipakai oleh beberapa user dalam waktu yang bersamaan tanpa membuatnya crash atau berhenti bekerja. Ini dapat dimanfaatkan ketika mengerjakan proyek yang sifatnya tim sehingga seluruh tim dapat bekerja dalam waktu bersamaan tanpa harus menunggu user lain selesai.

3. Keamanan yang Terjamin.

Open source MySQL tidak menyediakan keamanan yang buruk. Tetapi sebaliknya, MySQL mempunyai fitur keamanan yang cukup apik. Ada beberapa lapisan keamanan yang diterapkan oleh MySQL, seperti level nama host, dan subnetmask. Selain itu MySQL juga dapat mengatur hak akses user dengan enkripsi password tingkat tinggi.⁷

4. Tipe Data yang Bervariasi.

MySQL dapat mendukung berbagai macam data yang bisa di gunakan dalam MySQL. Contohnya float, integer, date, char, text, timestamp, double, dan lain sebagainya. Jadi manajemen database sistem sangat membantu Anda untuk mengembangkan perangkat lunak yang berguna untuk pengelolaan database di server.

5. Bersifat Open Source

MySQL adalah sistem manajemen database yang bersifat gratis. Meskipun gratis, bukan berarti database ini mempunyai kinerja buruk. Apalagi lisensi gratis yang dipakai adalah GPL di bawah pengelolaan Oracle sehingga kualitasnya termasuk baik. Selain itu, Anda juga tidak perlu khawatir jika terjadi masalah karena banyak komunitas dan dokumentasi yang membahas soal MySQL.

6. Tidak Membutuhkan RAM Besar.

MySQL dapat dipasang pada server dengan spesifikasi kecil. khawatir jika Anda hanya mempunyai server dengan kapasitas 1 GB masih bisa digunakan dengan mengakses database pengguna, maka tidak dibutuhkan spesifikasi minimal komputer yang tinggi,

7. Struktur Tabel yang Fleksibel.

MySQL mempunyai struktur tabel yang mudah dipakai dan fleksibel. Contohnya saat MySQL memproses ALTER TABLE dan lain sebagainya. Jika dibandingkan dengan database lain seperti Oracle dan PostgreSQL, MySQL tergolong lebih mudah.

Kekurangan MySQL

1. Kurang Cocok untuk Aplikasi Game dan Mobile

DBMS mysql ini sangat kurang digunakan untuk aplikasi Game dan Mobile application. Jadi jika anda ingin mengembangkan dua jenis aplikasi ini, MYSQL bukan teman yang tepat.

2. Sulit Mengelola Database yang Besar

Pengembangan aplikasi atau sistem di perusahaan dengan database yang cukup besar, ada baiknya jika menggunakan database manajemen sistem selain MySQL. MySQL dikembangkan supaya ramah dengan perangkat yang mempunyai spesifikasi rendah, itulah mengapa MySQL tidak memiliki fitur yang lengkap seperti aplikasi lainnya

Mengenal SQL

SQL merupakan bahasa pemrograman yang perlu Anda pahami karena dapat merelasikan antara beberapa tabel dengan database maupun antar database. Ada tiga bentuk SQL yang perlu Anda ketahui, yaitu Data Definition Language (DDL), Data Manipulation Language(DML), dan Data Control Language (DCL)

CREATE	Dipakai untuk membuat tabel dan database.
DROP	Dipakai untuk menghapus database dan tabel.
ALTER	Alter dapat mengganti field menggunakan perintah “Change”, menambahkan field menggunakan perintah “Add”, atau menghapus field menggunakan perintah “drop”, dan mengubah namanya menggunakan perintah “Rename”.

Data Definition Language (DDL)

DDL berguna pada saat Anda ingin mendefinisikan data di dalam database. Terdapat beberapa query yang dikelompokkan ke dalam DDL. DDL dapat Anda pahami karena merupakan dasar penggunaan SQL di tahap awal pembuatan database. Jika belum menjalankan perintah “CREATE”, Anda belum bisa melanjutkan penggunaan argumen yang lainnya.

Data Manipulation Language (DML)

DML dapat Anda pakai setelah menjalankan perintah DDL. DML berfungsi untuk memanipulasi, mengubah, atau mengganti isi dari database (tabel) yang sudah ada.

INSERT	Dipakai untuk memasukkan data ke dalam tabel pada database.
UPDATE	Dipakai untuk mengubah data yang ada di dalam tabel pada database.
DELETE	Dipakai untuk menghapus data di dalam tabel pada database.

Data Control Language (DCL)

Pada penggunaan user dan ingin mengatur hak akses masing-masing user, Anda sebaiknya memahami berbagai macam jenis DCL dan cara penggunaannya. DCL berguna untuk memberikan hak akses database, mendefinisikan space, mengalokasikan space, dan melakukan audit penggunaan database.

GRANT	Dipakai untuk memberikan izin kepada user untuk mengakses database.
REVOKE	Dipakai untuk membatalkan izin user untuk mengakses database.
COMMIT	Dipakai untuk menetapkan penyimpanan pada database.
ROLLBACK	Dipakai untuk membatalkan penyimpanan pada database.

Opsi di MySQL Server

MySQL server menyediakan beberapa bantuan yang bisa Anda akses untuk mengubah dan mendokumentasikan server, yaitu dengan mengetikkan perintah “\h” atau “\?” ke dalam koneksi MySQL yang sedang aktif.

Bantuan	\h atau \?	Dipakai untuk menampilkan opsi bantuan yang tersedia di dalam MySQL server.
Clear	\c	Dipakai untuk menghapus atau membatalkan semua perintah yang berjalan pada satu perintah.
Connect	\r	Dipakai untuk me-refresh koneksi ke database yang ada di dalam Server Host.
Ego	\G	Dipakai untuk menampilkan data di dalam database server secara horizontal

Go	\g	Dipakai untuk mengeksekusi perintah yang sudah dimasukkan.
Tee	\T	Dipakai untuk mengatur lokasi file untuk perintah yang ingin didokumentasikan.

Perintah Administrasi di MySQL Server

Pengertian MySQL server lainnya yaitu sebuah database manajemen sistem yang bersifat client/server. Jadi sangat penting untuk mengatur hak akses setiap user. Pengaturan hak akses ini supaya semua user mempunyai wewenang yang berbeda. Hak akses juga dapat mengatur user supaya hanya dapat mengakses database tertentu saja.

1. user. Tabel ini digunakan untuk menyimpan informasi user MySQL yang mencakup informasi user, password dan host user, serta informasi hak akses user.
2. db. Tabel ini digunakan untuk menyimpan informasi mengenai hak akses user terhadap database.
3. host. Tabel ini digunakan untuk menyimpan daftar komputer (bisa berupa alamat IP, nama komputer, atau %) yang berhak mengakses suatu database.
4. tables_priv. Tabel ini digunakan untuk menyimpan informasi mengenai hak akses user terhadap tabel. Dengan kata lain menyimpan tabel ini dapat diakses oleh siapa dengan hak akses apa saja.
5. columns_priv. Tabel ini digunakan untuk menyimpan informasi mengenai hak akses user terhadap kolom.
6. procs_priv. Tabel ini digunakan untuk menyimpan informasi mengenai hak akses user terhadap procedure.
7. proc. Tabel ini digunakan untuk menyimpan informasi mengenai daftar procedure dalam MySQL.
8. func. Tabel ini digunakan untuk menyimpan informasi mengenai function yang didefinisikan di MySQL.

SUBLIME TEXT EDITOR



Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform operating system dengan menggunakan teknologi Phyton API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim, Aplikasi ini sangatlah fleksibel dan powerfull. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan sublime-packages. Sublime Text bukanlah aplikasi opensource dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (packages) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki linsensi aplikasi gratis.

Sublime Text mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur syntax highlight hampir di semua bahasa pemrograman yang didukung ataupun dikembangkan oleh komunitas seperti; C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile and XML. Biasanya bagi bahasa pemrograman yang didukung ataupun belum terdukung secara default dapat lebih dimaksimalkan atau didukung dengan menggunakan add-ons yang bisa didownload sesuai kebutuhan user.

Kelebihan dari Sublime Text Editor

1. Multiple Selection

Fitur ini dapat memungkinkan user untuk mengubah secara interaktif banyak baris sekaligus, mengubah nama variabel dengan mudah, dan memanipulasi file lebih cepat dari sebelumnya.

2. Goto Anything

Fitur yang sangat membantu dalam membuka file ataupun menjelajahi isi dari file hanya dengan beberapa keystrokes.

3. Command Pallete

Dengan hanya beberapa keystrokes, user dapat dengan cepat mencari fungsi yang diinginkan, tanpa harus menavigasi melalui menu. 4. Distraction Free Mode Ketika user memerlukan fokus penuh pada sublime text, fitur ini dapat membantu user dengan memberikan tampilan layar penuh.

4. Split Editing

Dapatkan hasil yang maksimal dari monitor layar lebar dengan dukungan editing perpecahan (Split). Menulis sisi file dengan sisi, atau menulis dua lokasi di satu file. Anda dapat mengedit dengan banyak baris dan kolom yang Anda inginkan.

5. Instant Project Switch

Menangkap semua file yang dimasukkan ke dalam project pada sublime text. Terintegrasi dengan fitur Goto Anything untuk menjelajahi semua file yang ada ataupun untuk beralih ke file dalam project lainnya dengan cepat.

6. Plugin API

Dilengkapi dengan plugin API berbasis bahasa pemrograman Python sehingga membuat aplikasi ini sangat tangguh.

7. Customize Anything

Aplikasi ini memberikan user kemudahan dalam hal pengaturan fungsional dalam sublime text.

8. Cross Platform

Aplikasi ini dapat berjalan hamper pada semua operating system modern seperti Windows, OS X, dan Linux based operating system.

Bahasa Pemrograman PHP



Dalam pembuatan Aplikasi yang kami buat, kami merancang Aplikasi tersebut dengan menggunakan Bahasa Pemrograman PHP beserta Frameworks-nya dan CSS. Disini kami akan menjelaskan apa itu Bahasa Pemrograman PHP, Fungsi Bahasa Pemrograman PHP dalam sebuah Web

Pengertian Bahasa Pemrograman PHP

PHP adalah bahasa pemrograman script server-side yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum. PHP di kembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group. Bahwasannya PHP mempunyai Situs resmi beralamat di <http://www.php.net>. Bahasa pemrograman PHP biasanya tidak digunakan pada keseluruhan pengembangan website, melainkan dikombinasikan dengan beberapa bahasa pemrograman lain. Misalnya saja untuk mengatur tampilan, layout, dan berbagai macam menu menggunakan CSS.

Selain itu, terdapat juga beberapa framework PHP; Laravel, Phalcon, Codigniter, Symfoni yang saat ini banyak tersedia di internet untuk

memudahkan proses pengembangan website menggunakan bahasa pemrograman tersebut.

Bahasa pemrograman ini menggunakan sistem server-side. *Server-side programming* adalah jenis bahasa pemrograman yang nantinya script/program tersebut akan dijalankan/diproses oleh server. Kelebihannya adalah mudah digunakan, sederhana, dan mudah untuk dimengerti dan dipelajari.

Semenjak PHP menjadi bahasa pemrograman yang open source, pengembang tidak perlu menunggu sampai dengan update terbaru rilis.

Pengguna PHP akan lebih baik jika menggunakan versi terbaru. Sehingga jika ada rilis terbaru Anda harus menyesuaikan sistem Anda dengan versi PHP yang paling baru. Meskipun harus menggunakan versi terbaru, biaya untuk maintenance dan web development sangat terjangkau.

Di bawah ini adalah contoh penggunaan PHP untuk menghubungkan dan membuat database.

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
    die("Koneksi gagal: " . $conn->connect_error);
}
// Create database
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
```



```
    echo "Database berhasil dibuat!";  
} else {  
    echo "Error membuat database: " . $conn->error;  
}  
$conn->close();  
?>
```

Fungsi PHP dalam Pemrograman Web

Bahasa pemrograman PHP merupakan Bahasa Pemrograman yang mungkin banyak digunakan untuk mengembangkan website.

Jadi sebenarnya untuk membuat sistem berbasis web tidak harus menggunakan Bahasa Pemrograman PHP. Namun karena fitur dari Bahasa Pemrograman PHP yang menarik dan dirasa memudahkan kalangan programmer , maka kebanyakan pengembangan menggunakan bahasa pemrograman ini.

Sebelum ada PHP, pengembang kebanyakan menggunakan bahasa pemrograman HTML. Bahasa pemrograman ini sudah dapat membuat tampilan website, akan tetapi tampilan yang dihasilkan masih biasa saja(Sederhana). Biasa saja karena website masih bersifat statis atau tetap, tidak bisa menyesuaikan tampilan dengan kondisi pengguna.

Semenjak adanya PHP maka pembuatan website dapat dikembangkan dan diatur agar menjadi website yang dinamis. Dimana website dapat menyesuaikan tampilan sesuai dengan perangkat yang digunakan oleh pengguna. Tidak hanya itu, website juga dapat melakukan input output dengan database.

PHP sendiri biasanya disisipkan ke dalam kode HTML dan dikombinasikan dengan bahasa pemrograman lain, seperti CSS.

Kelebihan

1. Memiliki Community yang besar

Programmer Web mana yang tidak mengetahui PHP, semua web programmer paling tidak pasti pernah mencoba PHP. Banyak sekali website yang menggunakan PHP sebagai bahasa pemrograman untuk membuat aplikasi web atau website nya. Facebook, Yahoo, Wikipedia, WordPress adalah contoh website terkenal yang menggunakan PHP. Forum untuk membahas dan juga saling bertukar pikiran dalam pemrograman PHP juga telah banyak muncul di berbagai situs. Kebanyakan kuliah di bidang IT mengajarkan PHP sebagai bahasa pemrograman awal untuk mahasiswanya yang berkuliah di jurusan website development (baca juga: Daftar Mata Kuliah Teknik Informatika)

2. Mudah Dipelajari

PHP mudah di install dan dikonfigurasi. membuatnya menjadi bahasa pemrograman tingkat entry level yang mudah dipelajari bagi seseorang yang baru memulai belajar pengembangan web. Tutorial untuk memulai belajar pemrograman PHP dapat diperoleh dengan mudah secara online, di toko buku, ataupun di lembaga bimbingan kursus pengembangan website.

3. Pengembangan Cepat

Membuat Aplikasi menggunakan PHP jauh lebih cepat daripada mengembangkan aplikasi web menggunakan bahasa pemrograman lain. banyak sekali tools, boiler yang tersedia secara open source untuk bahasa pemrograman PHP. hal ini mempercepat proses dari start sampai dengan finish sebuah projek pembuatan aplikasi web.

4. Ringkas

Bagi Programmer web yang pernah mencoba bahasa ASP maupun java pasti mengetahui betul satu kelebihan ini. Mulai dari proses install yang tidak perlu setting berlebihan, konfigurasi dengan database yang mudah. hingga proses pengembangan yang tidak memerlukan waktu kompilasi. membuat PHP terasa sangat ringkas dan praktis berbeda dengan bahasa pemrograman lain yang membutuhkan proses kompilasi untuk dapat melihat website yang telah diselesaikan pembuatannya. Bahkan, bahasa pemrograman php dapat digunakan didalam dokumen html, hmm ringkas bukan. (baca juga: Dasar Dasar HTML)

5. Maintenance mudah

Sekali web yang menggunakan PHP berjalan, programmer dapat dengan mudah melakukan update dari software PHP dengan mudah jika memang diperlukan. karena sifat PHP yang merupakan interpreter. Aplikasi web yang dibuat dengan menggunakan PHP dapat dengan mudah diupgrade versi PHP tanpa harus melakukan kompilasi ulang source code. berbeda sekali dengan bahasa pemrograman lain yang membutuhkan kompilasi ulang jika melakukan upgrade versi dari bahasa pemrograman. PHP juga dapat berjalan pada berbagai macam web server seperti apache, nginx, dan IIS. (baca juga : Pengertian Web Server Menurut Para Ahli)

6. Open Source

PHP merupakan sebuah projek Open source dengan license yang dikeluarkan oleh PHP group yaitu PHP license V3.01. Inti dari license ini adalah setiap pengguna program PHP bebas menggunakan PHP secara gratis tanpa harus memberikan royalty apapun ke PHP group namun tetap wajib mencantumkan licensi atas PHP yang dimiliki PHP Group. Dengan kata lain selama pemakai program PHP tidak mengakui produk PHP adalah buatannya maka penjual belian program yang menggunakan PHP diperbolehkan tanpa harus membayar licensi apapun.

7. Perkembangan Pesat

Karena sifat PHP yang open source, banyak sekali bermunculan proyek proyek open source besar yang menggunakan PHP seperti Prestashop, WordPress, Drupal, dan lain lain. Hal ini menjadi keunggulan yang sangat besar bagi orang yang menguasai pemrograman PHP. Dengan sangat luasnya perkembangan PHP, maka kesempatan untuk bisnis ataupun kerja pada bidang pemrograman PHP sangatlah luas

Pengertian CSS



Cascading Style Sheets atau lebih dikenal dengan CSS adalah bahasa pemrograman desain yang berguna untuk menyederhanakan proses pembuatan website. Pengertian CSS secara umumnya adalah bahasa pemrograman yang dipakai untuk mendesain halaman depan atau tampilan website (front end).

CSS menangani tampilan dari halaman website. Ada banyak hal yang dapat Anda lakukan menggunakan CSS dibandingkan dengan bahasa pemrograman inti seperti HTML dan PHP. Ketika menggunakan CSS, Anda dapat mengatur warna teks, jenis font, baris antar paragraf, ukuran kolom,

jenis background yang dipakai, desain layout, variasi tampilan di berbagai perangkat yang berbeda, dan berbagai efek yang dipakai di dalam website.

CSS sangat mudah dipelajari, tapi juga powerful karena dapat mengontrol penyajian tampilan dari dokumen HTML mulai dari yang simpel sampai kompleks. Tidak heran jika saat ini CSS hampir dipakai di berbagai website untuk dikombinasikan dengan HTML maupun PHP.

Fungsi CSS pada Web

Anda pernah mencoba mengembangkan website menggunakan HTML atau PHP? Jika sudah, Anda pasti merasakan sedikit kesulitan atau kewalahan untuk mendesain tampilannya. Apalagi jika membuat tampilan yang cukup kompleks di proyek yang besar. Maklum, kedua bahasa pemrograman tersebut memang bukan dikembangkan untuk mendesain tampilan depan website.

Namun Anda tidak perlu khawatir karena semua proses pengembangan tampilan website dapat Anda permudah dan perindah menggunakan CSS. Tombol tersebut dapat Anda buat hanya menggunakan HTML atau PHP. Sedangkan tampilan yang lebih menarik dan beragam seperti di bawah ini dapat Anda buat menggunakan CSS dengan lebih mudah.

Lebih berwarna dan menarik yang di bawah, kan? Tombol di bawah bisa saja Anda buat menggunakan PHP dan HTML, hanya saja membutuhkan lebih banyak baris kode dan sedikit ribet. Namun jika menggunakan CSS, Anda cukup menggunakan satu baris kode saja.

CSS tentu tidak hanya mengatur tombol saja. Anda bisa mendesain berbagai macam tampilan menggunakan CSS sehingga sesuai dengan kebutuhan. Selain itu, ada beberapa keuntungan yang bisa Anda dapatkan ketika menggunakan CSS, seperti ini:

1. Mempercepat Proses Desain

Apa yang akan Anda lakukan jika menggunakan desain yang sama di beberapa halaman HTML? Anda mungkin akan menyalin satu per satu ke setiap file halaman.

Sesuai dengan pengertian CSS, jika menggunakan CSS, Anda tidak perlu melakukan itu lagi. Anda mungkin akan menyalin satu per satu ke setiap file halaman. Jika menggunakan CSS, Anda tidak perlu melakukan itu lagi.

Ketika menggunakan CSS, Anda cukup mengetikkan satu kali fungsi CSS kemudian menggunakannya di berbagai halaman HTML. Fungsi CSS yang Anda buat dalam satu file dapat Anda panggil ke berbagai halaman web tanpa harus menyalin baris kode fungsi berkali-kali.

2. Halaman Lebih Cepat Dimuat

Jika menggunakan CSS, Anda tidak perlu menuliskan atribut tag HTML di setiap file. Anda hanya cukup menulis satu aturan CSS dan menerapkannya di berbagai file yang membutuhkannya hanya dengan memanggilnya.

Jadi satu file hanya mengandung sedikit baris kode yang dimuat. sedikit baris kode inilah yang akan membuat proses *download* menjadi lebih cepat.

3. Proses Pemeliharaan Mudah

CSS memudahkan Anda untuk mengubah tampilan di berbagai halaman. Hanya dengan mengubah fungsi style di file CSS maka seluruh tampilan yang menggunakan fungsi tersebut akan berubah secara otomatis.

4. Style Lebih Beragam Dibanding HTML

CSS mempunyai atribut lebih beragam dibandingkan dengan HTML. Apa keuntungannya? Anda mempunyai lebih banyak pilihan tampilan halaman website.

5. Kompatibel Dengan Berbagai Macam Perangkat

CSS memungkinkan konten Anda dapat dioptimasi di lebih dari satu perangkat. Misal ketika memproses sebuah dokumen. Jika menggunakan CSS, Anda bisa menyesuaikan tampilan dokumen di perangkat versi lama sekaligus di versi yang baru.

6. CSS Menjadi Standar Pengembangan Website

Hampir seluruh website yang ada di internet menggunakan CSS di dalamnya. Selain tampilannya yang lebih menarik, kebanyakan browser populer saat ini juga mendukung CSS. Jadi jika Anda ingin mengembangkan website yang sesuai standar browser populer, sebaiknya gunakan CSS.

Cara Kerja CSS

Setelah mengetahui pengertian CSS dan fungsinya, pertanyaan selanjutnya ialah bagaimana cara kerja CSS? Jawabannya, CSS membuat style di dalam halaman web dapat berinteraksi dengan elemen HTML. Elemen merupakan komponen HTML dari sebuah halaman web. Berikut adalah salah satu contoh elemen di HTML:

```
<p>This is my paragraph!</p>
```

Jika Anda ingin membuat paragraf di atas muncul dengan warna yang lain dan bold, Anda bisa menambahkan baris CSS seperti di bawah ini:

```
p { color:pink; font-weight:bold; }
```

Pada contoh di atas, “p” (paragraf) merupakan bagian baris kode CSS untuk menspesifikasikan elemen HTML atau biasa dikenal dengan nama ‘selector’.

Di dalam CSS, selector ditulis di bagian kiri sebelum tanda kurung kurawal (*curly bracket*) pertama. Sedangkan informasi yang ada di dalam tanda kurung kurawal merupakan sebuah deklarasi yang di dalamnya mengandung properti dan nilai yang diaplikasikan ke dalam selector.

Contoh dari properti adalah ukuran font, warna, layout, format, dan lain sebagainya. Sedangkan nilai yang dimaksud adalah pengaturan yang diterapkan ke dalam properti. Misalnya dari contoh di atas, “color” dan “font-weight” merupakan properti, sedangkan “pink” dan “bold” merupakan sebuah nilai.

Contoh dari properti adalah ukuran font, warna, layout, format, dan lain sebagainya. Sedangkan nilai yang dimaksud adalah pengaturan yang diterapkan ke dalam properti. Misalnya dari contoh di atas, “color” dan “font-weight” merupakan properti, sedangkan “pink” dan “bold” merupakan sebuah nilai.

Jadi seluruh baris kode di dalam tanda kurung kurawal { color:pink; font-weight:bold; } merupakan sebuah deklarasi, sedangkan “p” (HTML paragraf) merupakan sebuah selector. Baik deklarasi dan selector merupakan prinsip dasar dari penggunaan CSS dan HTML. Prinsip ini juga dapat digunakan untuk mengatur ukuran font, warna background, format teks, dan lain sebagainya.

Sebagai tambahan, baris kode ..

```
body { background-color:lightblue; }
```

.. akan membuat halaman background menjadi biru cerah, atau ..

```
p { font-size:20px; color:red; }
```

.. akan menerapkan ukuran font 20 dengan warna merah.

3 Macam CSS di Dalam Kode HTML

Ada tiga macam penempatan CSS yang dapat dipakai ke dalam baris kode HTML, yaitu *external*, *internal*, atau *inline*. Bagian ini akan membahas tiga macam CSS ini dengan lebih detail.

Inline CSS

CSS inline merupakan penempatan CSS yang langsung dilakukan di dalam konten HTML. Penempatan CSS secara inline hanya akan mempengaruhi satu bagian baris kode. Berikut adalah contohnya.

```
<h1 style="font-size:30px;color:blue;">Cek berita utama ini!</h1>
```

CSS di atas berarti hanya akan mempengaruhi paragraf yang spesifik di dalam satu file .html dan muncul dengan warna biru (*blue*) ukuran font 30px.

External CSS

External CSS adalah penempatan baris kode CSS menggunakan file .css. Data ini dapat digunakan untuk mendeklarasikan sebagian besar pengaturan tampilan website secara keseluruhan.

Cara ini lebih sederhana dan simpel daripada menambahkan baris kode di setiap elemen HTML yang ingin Anda atur tampilannya. Jika menggunakan penempatan secara eksternal, Anda harus menambahkan header untuk memanggil file .css di dalam file HTML seperti contoh di bawah ini.

```
<head>  
<link rel="stylesheet" type="text/css" href=fileCSSAnda.css">  
</head>
```

Baris kode di atas akan menghubungkan file .html dengan eksternal style sheet (CSS). Pada contoh di atas file CSS menggunakan nama 'fileCSSAnda.css' yang dapat Anda sesuaikan dengan letak file CSS di dalam direktori website. Ketika file HTML sudah terhubung dengan file CSS maka seluruh pengaturan CSS akan diterapkan di dalam kode HTML.

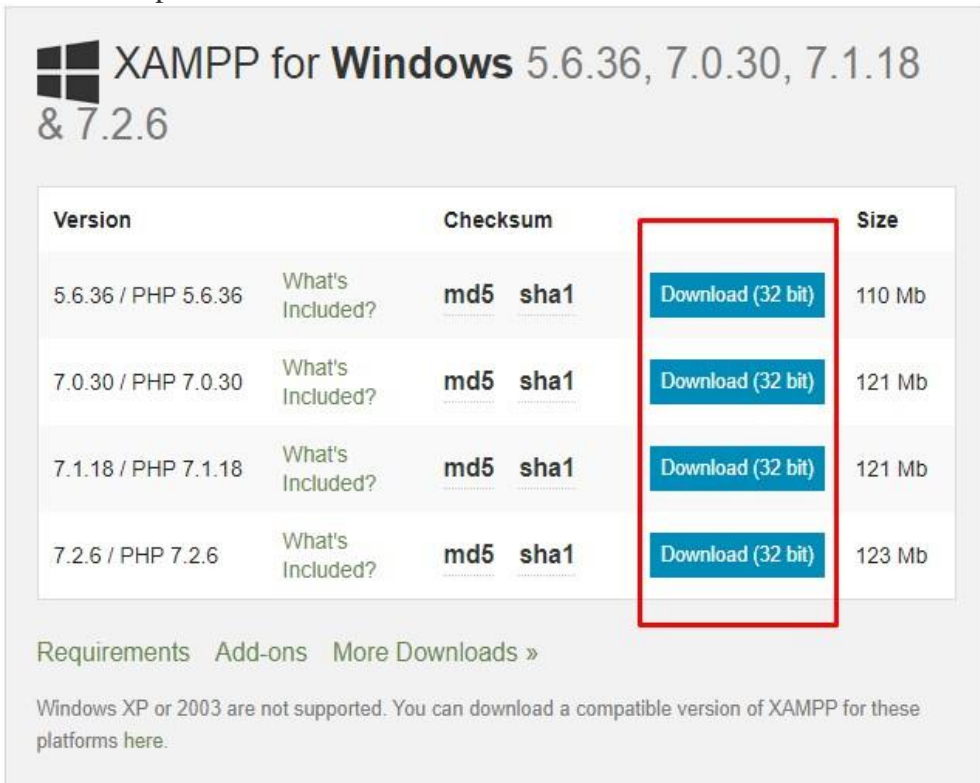
XAMPP



XAMPP adalah free software yang mendukung banyak platform sistem operasi yang berfungsi sebagai server yang berdiri sendiri (localhost). XAMPP adalah perangkat lunak web server Apache yang tertanam server MySQL di dalamnya, XAMPP didukung dengan bahasa pemrograman PHP. XAMPP adalah paket PHP dan MySQL yang berbasis open source yang digunakan sebagai alat untuk membantu pengembangan aplikasi yang berbasis PHP. XAMPP memiliki beberapa paket perangkat lunak yang berbeda ke dalam satu paket lalu dikombinasikan.

Sebelum kita masuk ke dalam Aplikasi yang kami buat, terlebih dahulu kita membuka Xampp, jika tidak mempunyai Xampp silahkan di Install Xampp nya dengan cara dibawah ini:

Download aplikasi XAMPP terbaru Pilih salah satu



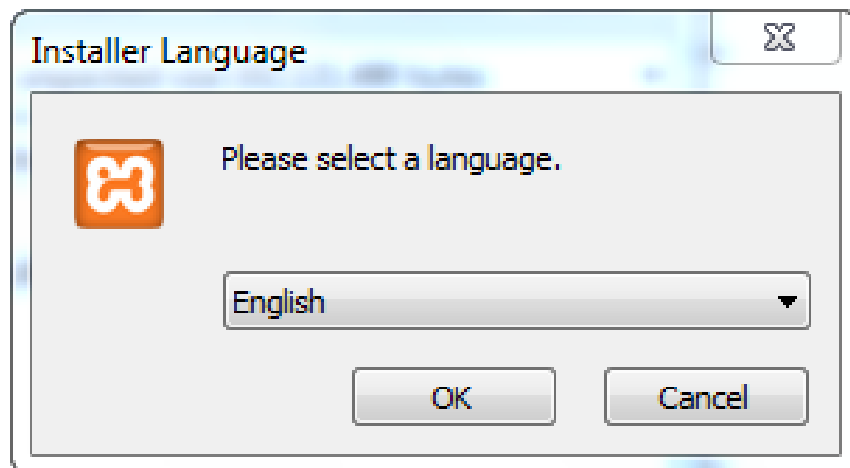
XAMPP for Windows 5.6.36, 7.0.30, 7.1.18 & 7.2.6

Version	Checksum	Download (32 bit)	Size
5.6.36 / PHP 5.6.36 What's Included?	md5 sha1	Download (32 bit)	110 Mb
7.0.30 / PHP 7.0.30 What's Included?	md5 sha1	Download (32 bit)	121 Mb
7.1.18 / PHP 7.1.18 What's Included?	md5 sha1	Download (32 bit)	121 Mb
7.2.6 / PHP 7.2.6 What's Included?	md5 sha1	Download (32 bit)	123 Mb

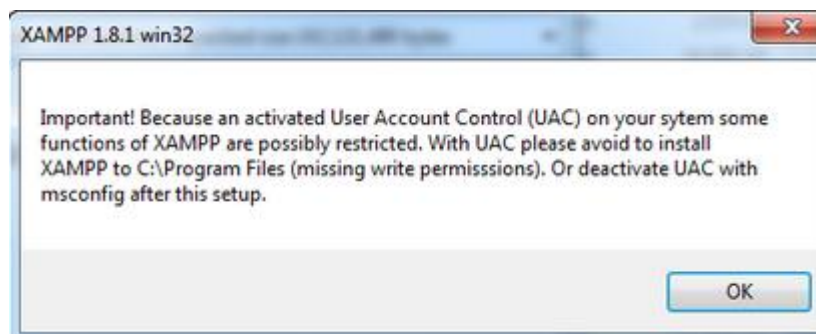
[Requirements](#) [Add-ons](#) [More Downloads »](#)

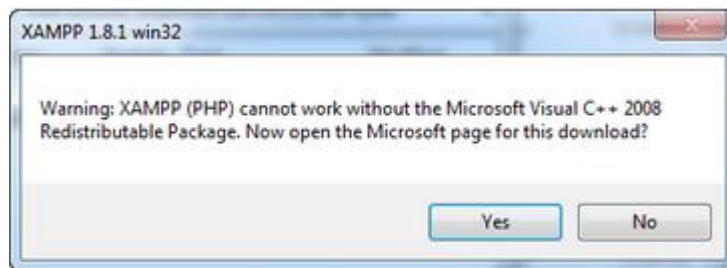
Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

1. Dobel klik file XAMPP yang baru saja Anda download, nanti selanjutnya akan muncul jendela “installer language” seperti di bawah ini:



2. Selanjutnya pilih bahasa. Pilih yang Bahasa yang mudah dimengerti, dimana saya memilih bahasa Inggris (English), setelah kita memilihnya Klik OK.
3. Kadang pada proses ini muncul pesan error. Jika ada, abaikan saja dan lanjutkan dengan klik OK dan YES.

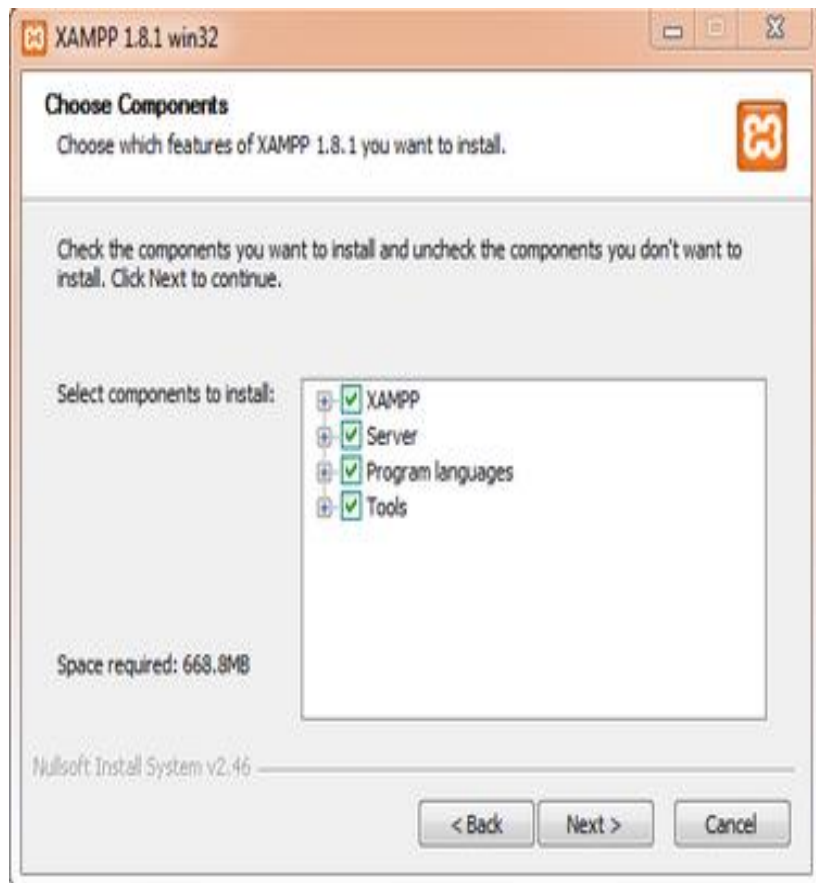




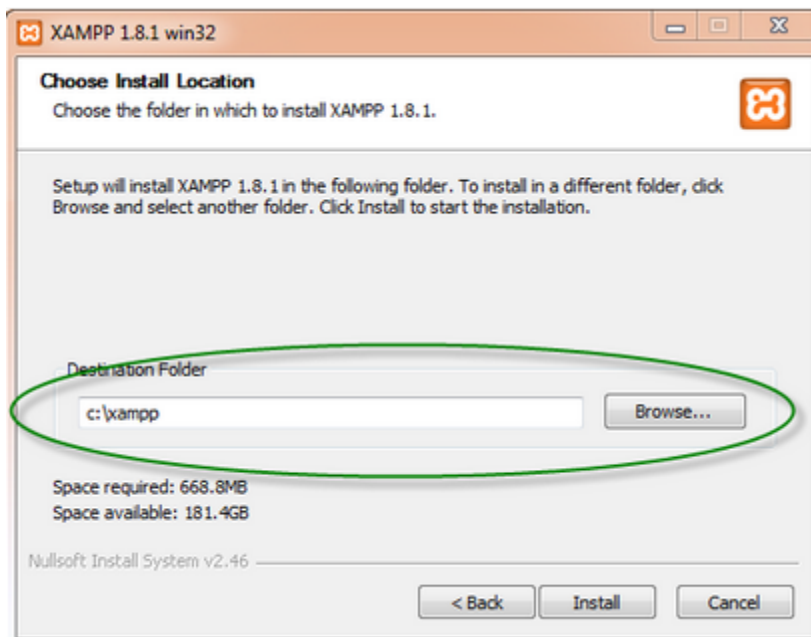
4. Berikutnya akan muncul jendela yang isinya meminta Anda menutup semua aplikasi yang sedang berjalan. Jika semua aplikasi sudah ditutup, maka klik tombol **Next**.



5. Selanjutnya Anda akan diminta untuk memilih aplikasi yang mau diinstal. Centang saja semua pilihan dan klik tombol **Next**.



6. Kemudian Anda akan diminta untuk menentukan lokasi folder penyimpanan file-file dan folder XAMPP. Secara default akan diarahkan ke lokasi **c:\xampp**. Namun jika Anda ingin menyimpannya di folder lain bisa klik **browse** dan tentukan secara manual folder yang ingin digunakan. Jika sudah selesai, lanjutkan dan klik tombol **Install**.



7. Disini kita disuru Menunggu beberapa menit hingga proses instalasi selesai. Jika sudah muncul jendela seperti di bawah ini, klik tombol **Finish** untuk menyelesaikannya.

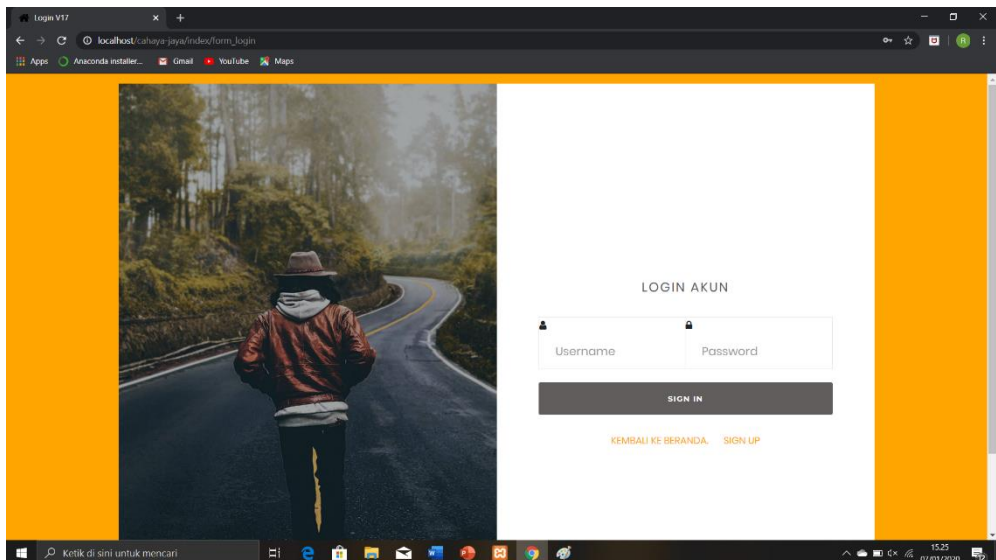


- a. Setelah kita melakukan Instalasin Xampp, kita buka Xampp kita dan jalankan Apache dan MySQL.
- b. Setelah dijalankan Apache dan MySQL tersebut kita pergi ke Google Chrome dengan cara mengetik localhost/cahaya-jaya/ lalu klik Enter,maka kita langsung dituju ke Aplikasi yang kami buat dan akanmuncul tampilan awal Aplikasi kami

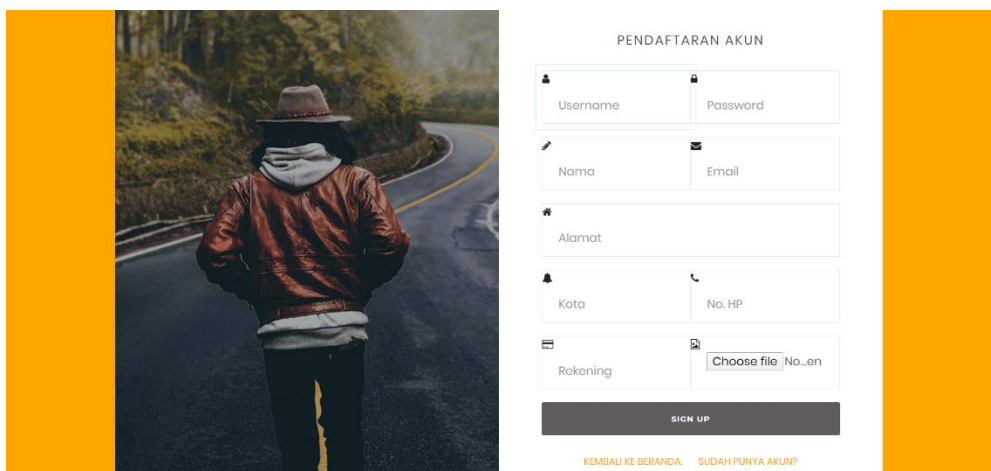


- c. Didalam Aplikasi yang kami buat, terdapat beberapa fitur seperti Search,Home,Account,Buy Now dan fitur-fitur tersebut mempunyai fungsi-fungsi masing-masing.

- d. Sebelum kita melakukan pembelian kita terlebih dahulu melakukan Login, seperti gambar dibawah



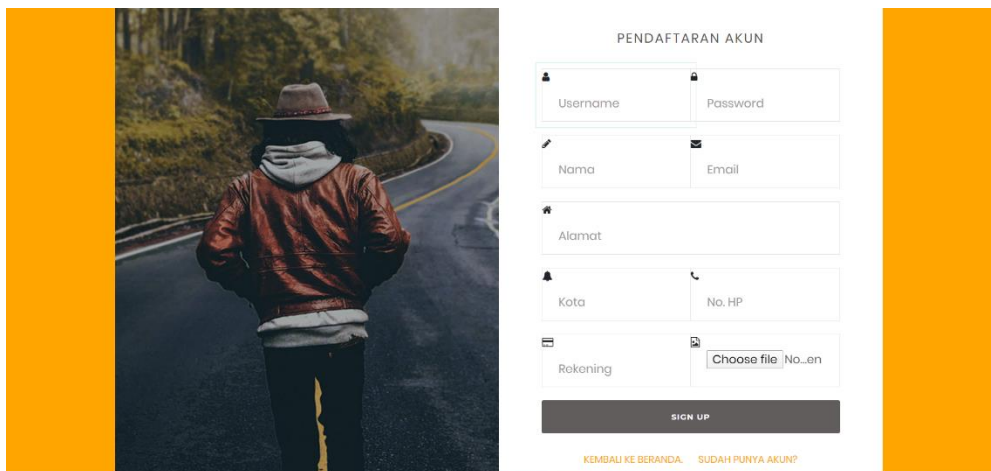
- e. Jika kita tidak mempunyai Account untuk melakukan Pembelian kita diharapkan untuk melakukan Registrasi Account. Seperti Gambar dibawah



- f. Setelah melakukan registrasi akun, kembali lagi ke menu login dan masukkan akun yang telah kita daftar untuk melakukan pembelian pada Aplikasi yang kami buat.
- g. Pada menu Search, kita dapat melihat berbagai jenis makanan dan harga yang kita inginkan.

Registrasi pada Aplikasi E-Commerce

Registrasi adalah sebuah pendaftaran setiap program yang berfungsi untuk menghubungkan data pribadi kita di Aplikasi tersebut. Didalam Aplikasi yang kami buat, kami menciptakan sebuah fitur Registrasi yang seperti gambar dibawah ini:



The image displays a registration form titled "PENDAFTARAN AKUN" (Account Registration) overlaid on a background image of a person in a red jacket and hat walking away on a winding road. The form contains the following fields:

- Username
- Password
- Nama (Name)
- Email
- Alamat (Address)
- Kota (City)
- No. HP (Phone Number)
- Rokoning (Profile Picture) with a "Choose file" button and ".No..en" extension

At the bottom of the form is a "SIGN UP" button. Below the button, there are two links: "KEMBALI KE BERANDA" (Back to Home) and "SUDAH PUNYA AKUN?" (Already have an account?).

Terdapat beberapa kolom yang saling berkaitan data diri kita, dan kita tidak boleh mengosongkan 1 kolom tentang data diri kita, jika kita mengosongkan maka terjadi kegagalan pendaftaran akun. Dimana fitur Registrasi yang kami buat telah tersambung dengan Database, supaya akun yang kita daftar bisa terhubung dengan server yang kami buat. Dan fitur Registrasi ini mempunyai sebuah keunikan jika kita mendaftarkan akun, dimana setelah kita mengisi semua kolom yang terdapat pada fitur registrasi tersebut, kita mengklik **Sign**

Up, setelah kita mengklik kita akan menunggu beberapa menit untuk mendapatkan Notifikasi Konfirmasi Akun melalui Whatsapp, dimana kita akan disuru mengonfirmasi akun yang telah kita daftar. Setelah melakukan Konfirmasi, akan ada pemberitahuan bahwasannya akun Anda telah terdaftar.

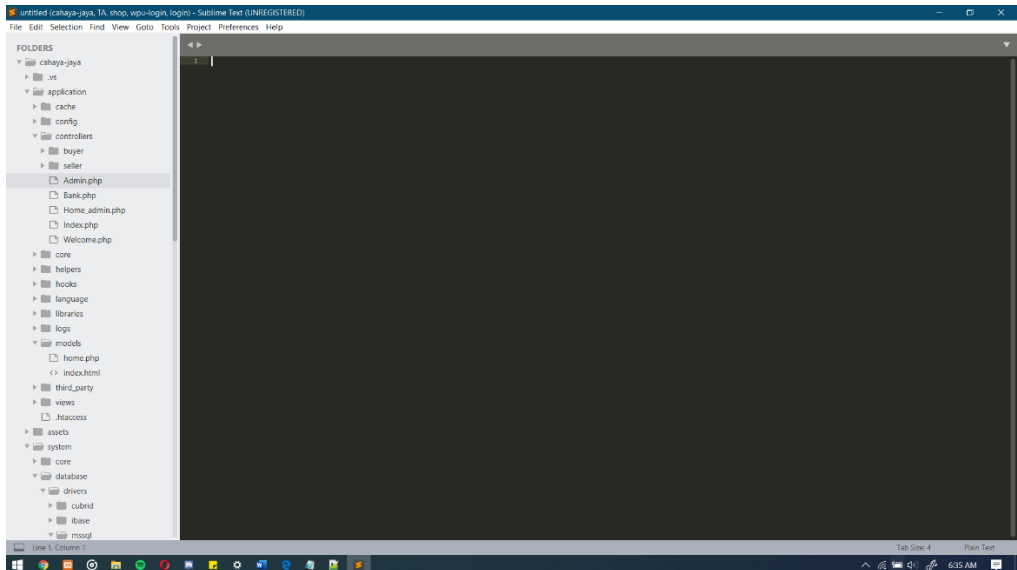
Langkah-Langkah Untuk Melakukan Transaksi Pembelian

1. Anda akan disuruh Mendaftarkan Akun.
2. Klik fitur ACCOUNT, dan Pilih fitur Registration, jika belum punya akun, jika sudah punya akun Anda diutamakan langsung ke fitur Login.
3. Setelah Anda mendaftarkan akun tersebut Anda akan disarankan ke menu login,
4. Di menu Login Anda disur memasukkan akun yang telah daftarkan, setelah memasukkan akun yang telah kita daftar kita akan disuru ke fitur Login dan Anda disuru langsung memasukkan Akun yang telah Anda daftar, Lalu klik Login
5. Setelah melakukan registrasi dan berhasil Login, berarti akun Anda telah terdaftar dan Anda bisa melakukan Transaksi Pembelian pada Aplikasi yang saya ciptakan bersama teman saya.

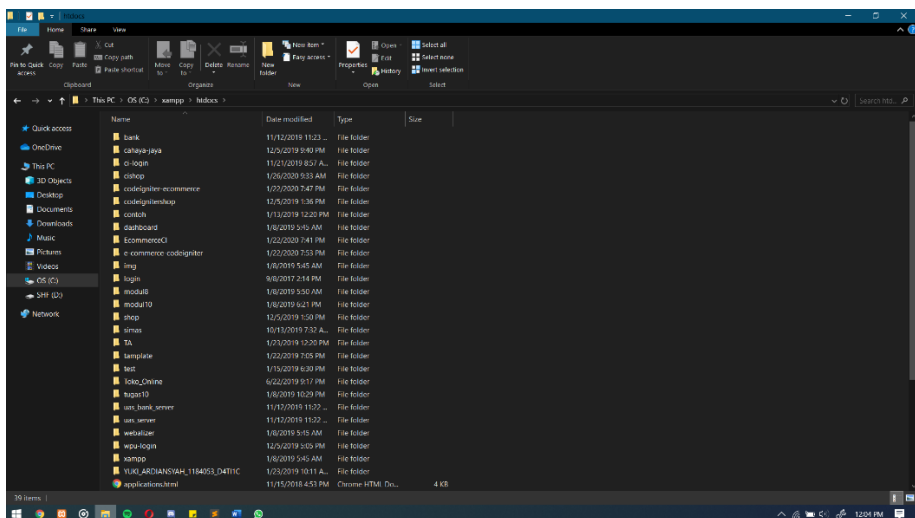
BAB III

KONFIGURASI APLIKASI

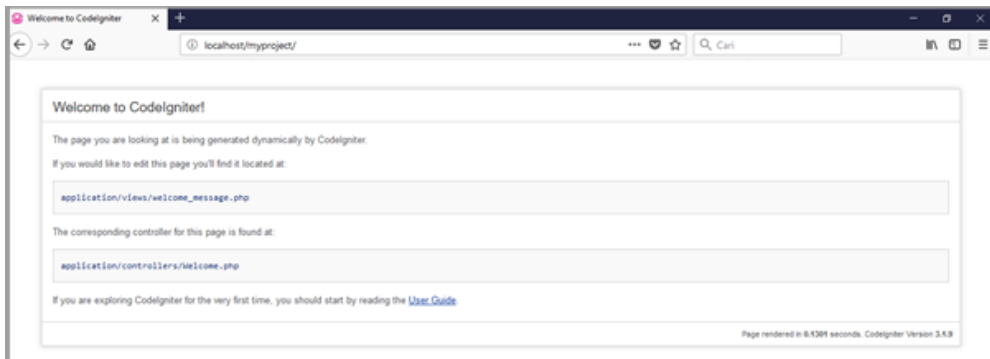
1. Buka Sublime Text Editor, lalu klik menu "file-open folder", selanjutnya cari folder akademik yang disimpan di C:/xampp/htdocs/cahaya-jaya, atau sesuai dengan dimana Anda menyimpan folder e-commerce di Xampphtdocs. Hasilnya akan nampak seperti di bawah.



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



14. Selanjutnya buka browser Anda. disini saya menggunakan Mozilla Firefox. Kemudian kunjungi URL berikut: <http://localhost/myproject>
Jika instalasi berhasil maka, akan terlihat seperti gambar berikut:



15. Selesai Konfigurasi CodeIgniter

CodeIgniter dapat berjalan dengan konfigurasi default, tetapi untuk sebuah aplikasi yang nyata kita harus tetap melakukan konfigurasi, setidaknya pada bagian base_url dan router. Pengaturan base_url dan router sangat berguna ketika proses pengembangan aplikasi yang banyak menggunakan helper dan library.

File konfigurasi terletak dalam folder application/config. File yang terdapat pada direktori tersebut yang sering digunakan adalah file autoload.php, config.php, database.php dan routes.php.

9. File autoload.php. konfigurasi pada file ini bertujuan untuk menentukan sebuah sumber yang akan menjadi titik dari penyimpanan agar dapat dipanggil secara otomatis

File autoload['libraries'] :

```
$autoload['libraries'] = array();
/*
 * Auto-load Libraries
 *
 * These are the classes located in system/libraries/ or your
 * application/libraries/ directory, with the addition of the
 * 'database' library, which is somewhat of a special case.
 *
 * Prototype:
 *
 *   $autoload['libraries'] = array('database', 'email', 'session');
 *
 * You can also supply an alternative library name to be assigned
 * in the controller:
 *
 *   $autoload['libraries'] = array('user_agent' => 'ua');
 */
File autoload.php bagian libraries
```

File autoload['helper'] :

```

$autoload['helper'] = array();
/*
|-----|
| Auto-load Helper Files |
|-----|
| Prototype: |
| $autoload['helper'] = array('url', 'file'); |
|*/

```

File autoload.php bagian helper

10. File config.php. pada file ini kita akan mengkonfigurasi secara umum dalam CodeIgniter.

```

$config['base_url'] = '';
/*
|-----|
| Base Site URL |
|-----|
|
| URL to your CodeIgniter root. Typically this will be your base URL,
| WITH a trailing slash:
|
| http://example.com/
|
| WARNING: You MUST set this value!
|
| If it is not set, then CodeIgniter will try guess the protocol and path
| your installation, but due to security concerns the hostname will be set
| to $_SERVER['SERVER_ADDR'] if available, or localhost otherwise.
| The auto-detection mechanism exists only for convenience during
| development and MUST NOT be used in production!
|
| If you need to allow multiple domains, remember that this file is still
| a PHP script and you can easily do that on your own.
|*/

```

File config.php bagian base url

\$config['base_url'] – konfigurasi ini berisikan alamat url pada sebuah aplikasi yang akan dibuat. Sebagai contoh “C:xampp/htdocs/codeigniter”

11. File database.php. disini kita akan mengkonfigurasi koneksi database. Pada aplikasi yang digunakan dalam buku ini kita menggunakan database mysql yang terdapat pada aplikasi XAMPP


```

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => '',
    'password' => '',
    'database' => '',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);

```

File database.php

- I. Hostname menggunakan localhost karena di sini kita menggunakan database mysql dari aplikasi xampp yang terinstal di computer local.
 - J. Username di dalam sini kita isikan sesuai username yang telah kita buat didalam mysql atau secara default bisa diisikan dengan “root”
 - K. Password sesuaikan dengan mysql masing-masing
 - L. Database ini merupakan nama dari database yang akan kita pakai di dalam sebuah aplikasi yang dibuat
12. File routes.php. Konfigurasi routing digunakan untuk memetakan permintaan atau request kedalam controller didalam website yang dibuat. Misalnya kita membuka alamat <http://www.nama-website.com>, permintaan tersebut tidak menyertakan nama controller yang ingin dibuka tetapi kita bisa secara default mengarahkannya agar secara otomatis akan membuka controller sesuai yang definisikan. Untuk melakukan konfigurasi routing buka file konfigurasi routes.php. Settingan utama yang ada adalah sebagai berikut:

```
$route['default_controller'] = 'welcome';  
$route['404_override'] = '';  
$route['translate_uri_dashes'] = FALSE;
```

File routes.php

Artinya secara default semua permintaan yang tidak menyertakan nama controllernya akan diarahkan untuk membuka controller “welcome”. Sehingga saat alamat <http://www.nama-website.com> dibuka secara otomatis akan terbuka

BAB IV

SCRIPT APLIKASI CAHAYA-JAYA

1. Autoload.php

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

/*
| -----
| -----
| AUTO-LOADER
| -----
| -----
| This file specifies which systems should be loaded by
| default.
|
| In order to keep the framework as light-weight as possible
| only the
| absolute minimal resources are loaded by default. For
| example,
| the database is not connected to automatically since no
| assumption
| is made regarding whether you intend to use it. This file
| lets
| you globally define which systems you would like loaded
| with every
| request.
|
| -----
| -----
| Instructions
| -----
| -----
|
| These are the things you can load automatically:
|
| 1. Packages
| 2. Libraries
| 3. Drivers
| 4. Helper files
| 5. Custom config files
| 6. Language files
| 7. Models
|
| */

/*
| -----
| -----
| Auto-load Packages
| -----
| -----
```

```

| Prototype:
|
| $autoload['packages'] = array(APPPATH.'third_party',
| '/usr/local/shared');
|
| */
$autoload['packages'] = array();

/*
| -----
|
| Auto-load Libraries
| -----
|
| These are the classes located in system/libraries/ or your
| application/libraries/ directory, with the addition of the
| 'database' library, which is somewhat of a special case.
|
| Prototype:
|
| $autoload['libraries'] = array('database', 'email',
| 'session');
|
| You can also supply an alternative library name to be
| assigned
| in the controller:
|
| $autoload['libraries'] = array('user_agent' => 'ua');
| */
$autoload['libraries'] = array();

/*
| -----
|
| Auto-load Drivers
| -----
|
| These classes are located in system/libraries/ or in your
| application/libraries/ directory, but are also placed
| inside their
| own subdirectory and they extend the CI_Driver_Library
| class. They
| offer multiple interchangeable driver options.
|
| Prototype:
|
| $autoload['drivers'] = array('cache');
|
| You can also supply an alternative property name to be
| assigned in
| the controller:

```

```

|
|   $autoload['drivers'] = array('cache' => 'cch');
|
| */
$autoload['drivers'] = array();

/*
| -----
| -----
|   Auto-load Helper Files
| -----
| -----
| Prototype:
|
|   $autoload['helper'] = array('url', 'file');
| */
$autoload['helper'] = array();

/*
| -----
| -----
|   Auto-load Config files
| -----
| -----
| Prototype:
|
|   $autoload['config'] = array('config1', 'config2');
|
| NOTE: This item is intended for use ONLY if you have
created custom
| config files. Otherwise, leave it blank.
|
| */
$autoload['config'] = array();

/*
| -----
| -----
|   Auto-load Language files
| -----
| -----
| Prototype:
|
|   $autoload['language'] = array('lang1', 'lang2');
|
| NOTE: Do not include the "_lang" part of your file. For
example
| "codeigniter_lang.php" would be referenced as
array('codeigniter');
|
| */

```

```

$autoload['language'] = array();

/*
| -----
|   Auto-load Models
| -----
|
| Prototype:
|
|   $autoload['model'] = array('first_model',
| 'second_model');
|
| You can also supply an alternative model name to be
assigned
| in the controller:
|
|   $autoload['model'] = array('first_model' => 'first');
*/
$autoload['model'] = array();

```

2. Database.php

```

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

/*
| -----
|   DATABASE CONNECTIVITY SETTINGS
| -----
|
| This file will contain the settings needed to access your
database.
|
| For complete instructions please consult the 'Database
Connection'
| page of the User Guide.
|
| -----
|   EXPLANATION OF VARIABLES
| -----
|
|   ['dsn']      The full DSN string describe a connection to
the database.
|   ['hostname'] The hostname of your database server.
|   ['username'] The username used to connect to the database
|   ['password'] The password used to connect to the database

```

```

|   ['database'] The name of the database you want to connect
to
|   ['dbdriver'] The database driver. e.g.: mysqli.
|               Currently supported:
|               cubrid, ibase, mssql, mysql, mysqli, oci8,
|               odbc, pdo, postgre, sqlite, sqlite3, sqlsrv
|   ['dbprefix'] You can add an optional prefix, which will
be added
|               to the table name when using the Query
Builder class
|   ['pconnect'] TRUE/FALSE - Whether to use a persistent
connection
|   ['db_debug'] TRUE/FALSE - Whether database errors should
be displayed.
|   ['cache_on'] TRUE/FALSE - Enables/disables query caching
|   ['cachedir'] The path to the folder where cache files
should be stored
|   ['char_set'] The character set used in communicating with
the database
|   ['dbcollat'] The character collation used in
communicating with the database
|               NOTE: For MySQL and MySQLi databases, this
setting is only used
|               as a backup if your server is running PHP <
5.2.3 or MySQL < 5.0.7
|               (and in table creation queries made with DB
Forge).
|               There is an incompatibility in PHP with
mysql_real_escape_string() which
|               can make your site vulnerable to SQL
injection if you are using a
|               multi-byte character set and are running
versions lower than these.
|               Sites using Latin-1 or UTF-8 database
character set and collation are unaffected.
|   ['swap_pre'] A default table prefix that should be
swapped with the dbprefix
|   ['encrypt'] Whether or not to use an encrypted
connection.
|
|               'mysql' (deprecated), 'sqlsrv' and 'pdo/sqlsrv'
drivers accept TRUE/FALSE
|               'mysqli' and 'pdo/mysqli' drivers accept an array
with the following options:
|
|               'ssl_key'      - Path to the private key file
|               'ssl_cert'     - Path to the public key
certificate file
|               'ssl_ca'       - Path to the certificate
authority file

```



```

|           'ssl_cpath' - Path to a directory containing
trusted CA certificates in PEM format
|           'ssl_cipher' - List of *allowed* ciphers to
be used for the encryption, separated by colons (':')
|           'ssl_verify' - TRUE/FALSE; Whether verify the
server certificate or not ('mysqli' only)
|
|   ['compress'] Whether or not to use client compression
(MySQL only)
|   ['stricton'] TRUE/FALSE - forces 'Strict Mode'
connections
|
|           - good for ensuring strict SQL
while developing
|   ['ssl_options'] Used to set various SSL options that can
be used when making SSL connections.
|   ['failover'] array - A array with 0 or more data for
connections if the main should fail.
|   ['save_queries'] TRUE/FALSE - Whether to "save" all
executed queries.
|
|           NOTE: Disabling this will also effectively
disable both
|
|           $this->db->last_query() and profiling of DB
queries.
|
|           When you run a query, with this setting set
to TRUE (default),
|
|           CodeIgniter will store the SQL statement for
debugging purposes.
|
|           However, this may cause high memory usage,
especially if you run
|
|           a lot of SQL queries ... disable this to
avoid that problem.
|
| The $active_group variable lets you choose which connection
group to
| make active. By default there is only one group (the
'default' group).
|
| The $query_builder variables lets you determine whether or
not to load
| the query builder class.
*/
$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'cahaya-jaya',
    'dbdriver' => 'mysqli',

```

```

'dbprefix' => '',
'pconnect' => FALSE,
'db_debug' => (ENVIRONMENT !== 'production'),
'cache_on' => FALSE,
'cachedir' => '',
'char_set' => 'utf8',
'dbcollat' => 'utf8_general_ci',
'swap_pre' => '',
'encrypt' => FALSE,
'compress' => FALSE,
'stricton' => FALSE,
'failover' => array(),
'save_queries' => TRUE
);

```

3. Home_buyer.php

```

<?php
Class Home_buyer extends CI_Controller{

    var $API = "";

    function __construct() {
        parent::__construct();
        // print_r($_SESSION); die;
        $this->API="http://localhost/uas_server/index.php";
        $this->load->library('session');
        $this->load->library('curl');
        $this->load->library('cart');
        $this->load->library('pagination');
        $this->load->helper('form');
        $this->load->helper('url');
        $this->load->model('home');

        if($this->session->userdata('account_id') == "" and
            $this->session->userdata('name') == ""){
            redirect(base_url("index"));
        }
    }

    function index(){
        $key = array(
            'search' => $this->input->get('search')
        );
        if ($key == '') {
            // print_r($uri); die;
            $data['barang'] = json_decode($this->curl->
>simple_get($this->API.'/product'));
            $data['category'] = json_decode($this->curl->
>simple_get($this->API.'/category'));
            $this->load->view('komponen_buyer/header');
            $this->load->view('buyer/index_buyer',$data);

```

```

        $this->load->view('komponen_buyer/footer');
    }elseif ($this->uri->segment(4)){
        $uri = array(
            'uri' => $this->uri->segment(4)
        );
        $data['barang'] = json_decode($this->curl-
>simple_get($this->API.'/ProductCategory',$uri));
        $data['category'] = json_decode($this->curl-
>simple_get($this->API.'/category'));
        $this->load->view('komponen_buyer/header');
        $this->load->view('buyer/index_buyer',$data);
        $this->load->view('komponen_buyer/footer');
    }elseif ($key != ''){
        $data['barang'] = json_decode($this->curl-
>simple_get($this->API.'/productspec',$key));
        $data['category'] = json_decode($this->curl-
>simple_get($this->API.'/category'));
        $this->load->view('komponen_buyer/header');
        $this->load->view('buyer/index_buyer',$data);
        $this->load->view('komponen_buyer/footer');
    }
}

function shop(){
    $key = array(
        'search' => $this->input->get('search')
    );
    if ($key == '') {
        // print_r($uri); die;
        $data['barang'] = json_decode($this->curl-
>simple_get($this->API.'/product'));
        $data['category'] = json_decode($this->curl-
>simple_get($this->API.'/category'));
        $this->load->view('komponen_buyer/header');
        $this->load->view('buyer/shop',$data);
        $this->load->view('komponen_buyer/footer');
    }elseif ($this->uri->segment(4)){
        $uri = array(
            'uri' => $this->uri->segment(4)
        );
        $data['barang'] = json_decode($this->curl-
>simple_get($this->API.'/ProductCategory',$uri));
        $data['category'] = json_decode($this->curl-
>simple_get($this->API.'/category'));
        $this->load->view('komponen_buyer/header');
        $this->load->view('buyer/shop',$data);
        $this->load->view('komponen_buyer/footer');
    }elseif ($key != ''){
        $data['barang'] = json_decode($this->curl-
>simple_get($this->API.'/productspec',$key));

```

```

        $data['category'] = json_decode($this->curl-
>simple_get($this->API.'/category'));
        $this->load->view('komponen_buyer/header');
        $this->load->view('buyer/shop',$data);
        $this->load->view('komponen_buyer/footer');
    }
}

function account(){
    $account_id = $this->session->userdata('account_id');
    $where = array('account_id' => $account_id);
    $data['account'] = json_decode($this->curl-
>simple_get($this->API.'/account',$where));
    // print_r($data); die;
    $this->load->view('komponen_buyer/header');
    $this->load->view('buyer/account',$data);
    $this->load->view('komponen_buyer/footer');
}

function invoice_detail($invoice_id){
    // $invoice_id = $this->input->post('invoice_id');
    // print_r($invoice_id); die;
    $data['detail'] = $this->home-
>invoice_detail($invoice_id);
    $data['total_invoice'] = $this->home-
>total_invoice($invoice_id);
    $this->load->view('komponen_buyer/header');
    $this->load->view('buyer/invoice_detail',$data);
    $this->load->view('komponen_buyer/footer');
}

function product_detail($id,$kid){
    $where = array(
        'product_id' => $id,
        'categori_id' => $kid
    );
    $data['barang'] = json_decode($this->curl-
>simple_get($this->API.'/product_detail', $where));
    $data['related'] = json_decode($this->curl-
>simple_get($this->API.'/chose_category', $where));
    $data['comment'] = json_decode($this->curl-
>simple_get($this->API.'/comment', $where));
    $this->load->view('komponen_buyer/header');
    $this->load->view('buyer/product_detail',$data);
}

public function update_account()
{

```

```

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

        $this->load->library('upload', $config);
        if ( ! $this->upload->do_upload('image')){

                // $id = $this->input->post('account_id');
                // $data = array('account_id' => $id);
                $data_account = array (
                        'image' => $this->input-
>post('image2'),
                                'username' => $this->input-
>post('username'),
                                'password' => $this->input-
>post('password'),
                                'name' => $this->input->post('name'),
                                'email' => $this->input-
>post('email'),
                                'address' => $this->input-
>post('address'),
                                'city' => $this->input->post('city'),
                                'telp' => $this->input->post('telp'),
                                'rekening' => $this->input-
>post('rekening'),
                                'id' => $this->input->post('id')
                );
                // print_r($data_product); die;
                $this->curl->simple_put($this-
>API.'/account', $data_account, array(CURLOPT_BUFFERSIZE =>
10));

                redirect('buyer/home_buyer/account');
        }else{
                // $id = $this->input->post('account_id');
                // $data = array('account_id' => $id);

                $image = $this->upload->data();
                $data_account = array (
                        'image' =>
$image['file_name'] ,
                                'username' => $this->input-
>post('username'),
                                'password' => $this->input-
>post('password'),
                                'name' => $this->input-
>post('name'),
                                'email' => $this->input-
>post('email'),

```

```

        'address'          => $this->input-
>post('address'),
        'city'             => $this->input-
>post('city'),
        'telp'             => $this->input-
>post('telp'),
        'rekening'        => $this->input-
>post('rekening'),
        'id'               => $this->input-
>post('id')
    );
    // print_r($data_product); die;
    $this->curl->simple_put($this-
>API.'/account', $data_account, array(CURLOPT_BUFFERSIZE =>
10));

redirect(base_url('buyer/home_buyer/account'));
    }

}

public function add_to_cart(){
    $data = array(
        'id'                => $this->input-
>post('product_id'),
        'account_id'        => $this->input-
>post('account_id'),
        'name'              => $this->input-
>post('product_name'),
        'qty'                => $this->input-
>post('quantity'),
        'product_image'     => $this->input-
>post('product_image'),
        'price'              => $this->input-
>post('price'),
        'nama'               => $this->input-
>post('nama'),
        'seller_id'         => $this->input-
>post('id_seller'),
    );
    $a=$this->cart->insert($data);
    // print_r($a); die;

    redirect(base_url('buyer/Home_buyer/cart'));
}

function cart(){
    $data['stock'] = $this->home->stock_cart();
    $this->load->view('komponen_buyer/header');
    $this->load->view('buyer/cart',$data);
    $this->load->view('komponen_buyer/footer');
}

```

```

    }

    public function clear_cart(){
        $this->cart->destroy();
        redirect(base_url('buyer/Home_buyer'));
    }

    public function remove_item($id){
        // $this->cart->remove($id);
        $data = array(
            'rowid'    => $id,
            'qty'      => 0
        );
        $this->cart->update($data);
        // if($i){
        //     echo 'aw'; die;
        // }else{
        //     echo 'awa'; die;
        // }
        redirect(base_url('buyer/Home_buyer/cart'));
    }

    public function shopping_history(){
        $account_id = $this->session->userdata('account_id');
        $exp = array('payment_status' => 'Cancelled');
        // $where = array('account_id' => $account_id);
        // $params = array('product_id'=> $this->uri-
>segment(3));
        // $data['order'] = json_decode($this->curl-
>simple_get($this->API.'/order',$where));
        $data['order'] = $this->home-
>shopping_history($account_id);
        // print_r($data['order']); die;
        $this->load->view('komponen_buyer/header');
        $this->load->view('buyer/shopping_history',$data);
        $this->load->view('komponen_buyer/footer');
    }

    public function shopping_history2(){
        $account_id = $this->session->userdata('account_id');
        $exp = array('payment_status' => 'Cancelled');
        // $where = array('account_id' => $account_id);
        // $params = array('product_id'=> $this->uri-
>segment(3));
        // $data['order'] = json_decode($this->curl-
>simple_get($this->API.'/order',$where));
        $data['order'] = $this->home-
>shopping_history2($account_id);
        // print_r($data['order']); die;
        $this->load->view('komponen_buyer/header');
        $this->load->view('buyer/shopping_history',$data);
    }

```

```

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

    function shipping_status(){
        // print_r($id); die;
        $id = $this->input->post('order_id');
        $total = $this->input->post('total');
        $account_id = $this->input->post('account_id');
        $status = 'Already Received';
        $this->home->shipping_status_buyer($id,$status);
        $this->home->pay_to_seller($account_id,$total);
        redirect('buyer/home_buyer/shopping_history');
    }

    function comment(){
        // print_r($id); die;
        $a = $this->input->post('product_id');
        $b = $this->input->post('categori_id');
        $this->home->comment();

        redirect('buyer/home_buyer/product_detail/' . $a . '/' . $b);
    }

    public function invoice_history(){
        $account_id = $this->session->userdata('account_id');
        // print_r($account_id); die;

        // $where = array('order_id' => $id);
        // $params = array('product_id'=> $this->uri-
>segment(3));
        $data['invoice'] = $this->home-
>invoice_history($account_id);
        $data['saldo'] = $this->home-
>saldo_invoice($account_id);
        // print_r($data['saldo']); die;
        $this->load->view('komponen_buyer/header');
        $this->load->view('buyer/invoice_history',$data);
        $this->load->view('komponen_buyer/footer');
    }

    public function invoice_history2(){
        $account_id = $this->session->userdata('account_id');
        // print_r($account_id); die;

        // $where = array('order_id' => $id);
        // $params = array('product_id'=> $this->uri-
>segment(3));
        $data['invoice'] = $this->home-
>invoice_history2($account_id);

```



```

        $data['saldo'] = $this->home-
>saldo_invoice($account_id);
        // print_r($data['saldo']); die;
        $this->load->view('komponen_buyer/header');
        $this->load->view('buyer/invoice_history',$data);
        $this->load->view('komponen_buyer/footer');
    }

    public function payment(){
        // $account_id = $this->session-
>userdata('account_id');
        $invoice_id = $this->input->post('invoice_id');
        $price = $this->input->post('price');
        // $where = array('order_id' => $id);
        // $params = array('product_id'=> $this->uri-
>segment(3));
        $this->home->payment($invoice_id);
        $this->home->update_saldo($price);
        // print_r($data); die;
        redirect('buyer/home_buyer/invoice_history');
    }

    function verification(){
        $this->load->view('komponen_buyer/header');
        $this->load->view('buyer/verification');
        $this->load->view('komponen_buyer/footer');
    }

    public function order(){
        $is_processed = $this->home->process();
        foreach ($this->cart->contents() as $key) {
            $id = $key['id'];
            $qtyAwal = $this->db-
>get_where('products',array('product_id' => $id))->row();
            $qtyAwalFix = $qtyAwal->stock;
            $qty = $key['qty'];
            $qtyAkhir = $qtyAwalFix - $qty;
            $stock = $this->home->stock($id,$qtyAkhir);
        }
        // $qty = $this->input->post('qty');
        // print_r($qtyAkhir); die;

        if($is_processed){
            $this->cart->destroy();
            redirect('buyer/Home_buyer/verification');
        }else{
            $this->session->set_flashdata('error','Gagal
untuk memproses pesanan anda, tolong coba lagi!');
            redirect('buyer/Home_buyer/cart');
        }
    }
}

```

```
// public function saldo()
// {
//     $sid = $this->session->userdata('account_id');
//     $saldo = $this->input->post('nominal');
//     $voucher_id = $this->input->post('voucher_id');
//     $this->home->saldo($sid,$saldo);
//     $this->home->status_voucher($voucher_id);
//     redirect('buyer/home_buyer/invoice_history');
// }
```

```
public function voucher()
{
    $sid = $this->session->userdata('account_id');
    $data2['code'] = $this->home->saldo($sid);
    // print_r($data2); die;
    $this->load->view('komponen_buyer/header');
    $this->load->view('buyer/topup2', $data2);
    $this->load->view('komponen_buyer/footer');
}
```

```
public function voucher2()
{
    $sid = $this->session->userdata('account_id');
    $data2['code'] = $this->home->saldo2($sid);
    // print_r($data2); die;
    $this->load->view('komponen_buyer/header');
    $this->load->view('buyer/topup', $data2);
    $this->load->view('komponen_buyer/footer');
}
```

```
function logout(){
    $this->cart->destroy();
    $this->session->sess_destroy();
    redirect('index');
}
```

```
function topup(){
    $nominal = $this->input->post('nominal');
    $status = 'Not Used';
    $sid = $this->session->userdata('account_id');
    $this->home->topup($nominal,$sid,$status);
    redirect('buyer/home_buyer/voucher');
}
```

```
}
```

4. Home_seller.php

```
<?php
Class Home_seller extends CI_Controller{

    var $API = "";

    function __construct() {
        parent::__construct();
        $this->API="http://localhost/uas_server/index.php";
        $this->load->library('session');
        $this->load->library('curl');
        $this->load->library('pagination');
        $this->load->helper('form');
        $this->load->helper('url');
        $this->load->model('home');

        if($this->session->userdata('account_id') == "" and
            $this->session->userdata('name') == ""){
            redirect(base_url("index"));
        }
    }

    function form_create_product(){
        $data['category'] = $this->home->category();
        $this->load->view('komponen_seller/header');
        $this->load->view('seller/create_product',$data);
        $this->load->view('komponen_seller/footer');
    }

    function index(){
        $account_id = $this->session->userdata('account_id');
        $exp = array('payment_status' => 'Cancelled');
        // $where = array('account_id' => $account_id);
        // $params = array('product_id'=> $this->uri-
>segment(3));
        // $data['order'] = json_decode($this->curl-
>simple_get($this->API.'/order',$where));
        $data['product'] = $this->home-
>my_product($account_id);
        $data['category'] = $this->home-
>category_product($account_id);
        // print_r($data); die;
        $this->load->view('komponen_seller/header');
        $this->load->view('seller/index_seller',$data);
        $this->load->view('komponen_seller/footer');
    }

    function create_product() {
        $config['upload_path'] = './assets/gambar_product';
        //path folder
```

```

        // $config['upload_path'] = "echo
base_url()/assets/gambar_product"; //path folder
        $config['allowed_types'] = 'gif|jpg|png|jpeg|bmp';
//type yang dapat diakses bisa anda sesuaikan

```

```

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

            $data_product = array(
                'account_id'      => $this->input-
>post('account_id'),
                'product_name'    => $this->input-
>post('product_name'),
                'description'      => $this->input-
>post('description'),
                'categori_id'     => $this->input-
>post('categori_id'),
                'price'           => $this->input-
>post('price'),
                'stock'           => $this->input-
>post('stock'),
                'product_image'    => $img_name
            );

            $insert = $this->curl->simple_post($this-
>API.'/product', $data_product, array(CURLOPT_BUFFERSIZE =>
10));

            if($insert)
            {
                // print_r($data); die;
                echo"<script>
                alert('Succes')
                </script>";
                redirect('seller/home_seller');
            }
            else
            {
                echo"<script>
                alert('Fail making product, check data
again!')

                </script>";
                $this->load->view('komponen_seller/header');
                $this->load->view('seller/create_product');
                $this->load->view('komponen_seller/footer');
            }
        }
    }
}

```

```

    }

    function delete_product($product_id){
        if(empty($product_id)){
            redirect('seller/home_seller');
        }else{
            $delete = $this->curl->simple_delete($this->API.'/product', array('product_id'=>$product_id),
            array(CURLOPT_BUFFERSIZE => 10));
            redirect('seller/home_seller');
        }
    }

    public function order_in(){
        $account_id = $this->session->userdata('account_id');
        $data['order'] = $this->home->order_in($account_id);
        // print_r($data); die;
        $this->load->view('komponen_seller/header');
        $this->load->view('seller/order_in',$data);
        $this->load->view('komponen_seller/footer');
    }

    public function order_in2(){
        $account_id = $this->session->userdata('account_id');
        $data['order'] = $this->home->order_in2($account_id);
        // print_r($data); die;
        $this->load->view('komponen_seller/header');
        $this->load->view('seller/order_in2',$data);
        $this->load->view('komponen_seller/footer');
    }

    public function order_in3(){
        $account_id = $this->session->userdata('account_id');
        $data['order'] = $this->home->order_in3($account_id);
        // print_r($data); die;
        $this->load->view('komponen_seller/header');
        $this->load->view('seller/order_in2',$data);
        $this->load->view('komponen_seller/footer');
    }

    function shipping_status(){
        // print_r($id); die;
        $id = $this->input->post('order_id');
        $resi = $this->input->post('resi');
        $courier = $this->input->post('courier');
        $status = 'On Delivery';
        $this->home->shipping_status($id,$status,$resi,$courier);
        redirect('seller/home_seller/order_in2');
    }

```

```

    public function update_product()
    {
        $config['upload_path'] = './assets/gambar_product';
        //path folder
        $config['allowed_types'] = 'gif|jpg|png|jpeg|bmp';
        //type yang dapat diakses bisa anda sesuaikan

        $this->load->library('upload', $config);
        if ( ! $this->upload->do_upload('product_image')){

            // $id = $this->input->post('account_id');
            // $data = array('account_id' => $id);
            $data = array (
                'product_image' => $this->input-
>post('product_image2'),
                'product_name' => $this->input-
>post('product_name'),
                'description' => $this->input-
>post('description'),
                'categori_id' => $this->input-
>post('categori_id'),
                'price' => $this->input-
>post('price'),
                'stock' => $this->input-
>post('stock'),
                'product_id' => $this->input-
>post('product_id')
            );
            // print_r($data_product); die;
            $this->curl->simple_put($this-
>API.'/product', $data, array(CURLOPT_BUFFERSIZE => 10));
            redirect('seller/home_seller');
        }else{
            // $id = $this->input->post('account_id');
            // $data = array('account_id' => $id);

            $image = $this->upload->data();
            $data = array (
                'product_image' =>
                $image['file_name'] ,
                'product_name' => $this->input-
>post('product_name'),
                'description' => $this->input-
>post('description'),
                'categori_id' => $this->input-
>post('categori_id'),
                'price' => $this->input-
>post('price'),

```

```

        'stock' => $this->input-
>post('stock'),
        'product_id' => $this->input-
>post('product_id')
    );
    // print_r($data_product); die;
    $this->curl->simple_put($this-
>API.'/product', $data, array(CURLOPT_BUFFERSIZE => 10));
    redirect(base_url('seller/home_seller'));
}

}

}

```

5. Admin.php

```

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

class Admin extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        is_logged_in();
    }

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

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

```

6. Bank.php

```

<?php
Class Bank extends CI_Controller{

    var $API = "";

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

```

```

        $this->
>API="http://localhost/uas_bank_server/index.php";
        $this->load->library('session');
        $this->load->library('curl');
        $this->load->library('pagination');
        $this->load->helper('form');
        $this->load->helper('url');
        $this->load->model('home');
    }

    function topup(){
        $id = $this->session->userdata('account_id');
        $uniq_key = $this->input->post('uniq_key');
        $nominal = $this->input->post('nominal');
        $where = array(
            'uniq_key' => $uniq_key,
            'nominal' => $nominal
        );
        $cek = json_decode($this->curl->simple_get($this->
>API.'/bank',$where));
        // print_r($cek); die;
        if ($cek) {
            // echo "aw"; die;
            $this->home->tambah_saldo($id,$nominal);
            $this->home->status_topup($uniq_key);
            // $this->session->set_flashdata('hasil','Saldo
Was Successfully Added');
            echo "<script>alert('Saldo Was Successfully
Added');history.go(-1);</script>";
        }else{
            echo "<script>alert('Pay This Unique Code In
Bank!');history.go(-1);</script>";
        }
    }
}

```

7. Home_admin.php

```

<?php
Class Home_admin extends CI_Controller{

    var $API = "";

    function __construct() {
        parent::__construct();
        $this->API="http://localhost/uas_server/index.php";
    }
}

```



```

        $this->load->library('session');
        $this->load->library('curl');
        $this->load->library('cart');
        $this->load->library('pagination');
        $this->load->helper('form');
        $this->load->helper('url');
        $this->load->model('home');

        if($this->session->userdata('account_id') == "" and
            $this->session->userdata('grup') == ""){
            redirect(base_url("index"));
        }
    }

    public function index(){
        $data['order'] = $this->home-
>shopping_history_admin();
        // $data['buyer'] = $this->home-
>shopping_history_admin_buyer();
        // print_r($data['order']); die;
        $this->load->view('admin/index_admin',$data);
    }

    public function index2(){
        $data['order'] = $this->home-
>shopping_history_admin2();
        // $data['buyer'] = $this->home-
>shopping_history_admin_buyer();
        // print_r($data['order']); die;
        $this->load->view('admin/index_admin',$data);
    }

    public function clear_payment(){
        $price = $this->input->post('price');
        $account_id = $this->input->post('account_id');
        $order_id = $this->input->post('order_id');
        $this->home->clear_payment($price,$account_id);
        $this->home->clear_status($order_id);
        // $data['buyer'] = $this->home-
>shopping_history_admin_buyer();
        // print_r($data['order']); die;
        redirect('home_admin/index2');
    }

    public function category(){
        $data['category'] = $this->home->category_admin();
        // $data['buyer'] = $this->home-
>shopping_history_admin_buyer();
        // print_r($data['order']); die;
        $this->load->view('admin/category',$data);
    }
}

```

```

function delete_category($categori_id){
    if(empty($categori_id)){
        redirect('home_admin/category');
    }else{
        $delete = $this->curl->simple_delete($this->API.'/category', array('categori_id'=>$categori_id), array(CURLOPT_BUFFERSIZE => 10));
        redirect('home_admin/category');
    }
}

function input(){
    $data = array(
        'category' => $this->input->post('category')
    );
    $this->curl->simple_post($this->API.'/category', $data, array(CURLOPT_BUFFERSIZE => 10));
    redirect('home_admin/category');
}
}

```

8. Index.php

```

<?php
Class Index extends CI_Controller{

    var $API = "";

    function __construct() {
        parent::__construct();
        $this->API="http://localhost/uas_server/index.php";
        $this->load->library('session');
        $this->load->library('curl');
        $this->load->library('pagination');
        $this->load->helper('form');
        $this->load->helper('url');
        $this->load->model('home');
    }

    function index($key = null){
        $key = array(
            'search' => $this->input->get('search')
        );
        if ($key == '') {
            // print_r($uri); die;
            $data['barang'] = json_decode($this->curl->simple_get($this->API.'/product'));

```

```

        $data['category'] = json_decode($this->curl-
>simple_get($this->API.'/category'));
        $this->load->view('index', $data);
    }elseif ($this->uri->segment(3)){
        $uri = array(
            'uri' => $this->uri->segment(3)
        );
        $data['barang'] = json_decode($this->curl-
>simple_get($this->API.'/ProductCategory', $uri));
        $data['category'] = json_decode($this->curl-
>simple_get($this->API.'/category'));
        $this->load->view('index', $data);
    }elseif ($key != ''){
        $data['barang'] = json_decode($this->curl-
>simple_get($this->API.'/productspec', $key));
        // $ab = $this->input->get('search');
        // print_r($ab); die;
        $data['category'] = json_decode($this->curl-
>simple_get($this->API.'/category'));
        $this->load->view('index', $data);
    }
}

```

```

function product_detail($id, $kid){
    $where = array(
        'product_id' => $id,
        'categori_id' => $kid
    );
    $data['barang'] = json_decode($this->curl-
>simple_get($this->API.'/product detail', $where));
    $data['related'] = json_decode($this->curl-
>simple_get($this->API.'/chose_category', $where));
    $data['comment'] = json_decode($this->curl-
>simple_get($this->API.'/comment', $where));
    $this->load->view('product_detail', $data);
}

```

```

function form_login(){
    $this->load->view('login');
}

```

```

function form_registration(){
    $this->load->view('registration');
}

```

```

function registration(){
    $config['upload_path'] = './assets/profile'; //path
folder
    // $config['upload_path'] = "echo
base_url()/assets/gambar_product"; //path folder

```

```
    $config['allowed_types'] = 'gif|jpg|png|jpeg|bmp';  
    //type yang dapat diakses bisa anda sesuaikan
```

```
    $this->load->library('upload', $config);  
    if (!$this->upload->do_upload('image')){  
        $error = $this->upload->display_errors();  
        print_r($error);  
    }else{  
        $result = $this->upload->data();  
        $img_name = $result['file_name'];  
  
        $data_account = array (  
            'image' => $img_name ,  
            'username' => $this->input->  
>post('username'),  
            'password' => $this->input->  
>post('password'),  
            'name' => $this->input->post('name'),  
            'email' => $this->input->  
>post('email'),  
            'address' => $this->input->  
>post('address'),  
            'city' => $this->input->post('city'),  
            'telp' => $this->input->post('telp'),  
            'rekening' => $this->input->  
>post('rekening'),  
            'grup' => '2'  
        );  
  
        #$token = base64_encode(random_bytes(32));  
        // var_dump($token);  
        // $user_token = [  
        //     'email' => $this->$email,  
        //     'token' => $token,  
        //     'date_created' => time()  
        // ];  
  
        // $this->_sendEmail();  
  
        // // $this->db->insert('user_token');  
  
        $insert = $this->curl->simple_post($this->  
>API.'/account', $data_account, array(CURLOPT_BUFFERSIZE =>  
10));  
  
        if($insert)  
        {
```

```

        echo '<script language="javascript">
        alert ("Registrasi Berhasil Di Lakukan!");
        window.location="index.php";
        </script>';
        redirect('index/form_login');
    }
    else
    {
        echo"<script>
        alert('gagal menambahkan akun, check data
lagi!')
        </script>";
    }
}

}

// function _sendemail(){
//
//     $config =[
//
//         'protocol'=> 'smtp',
//         'smtp_host'=>
//         'ssl://smtp.googlemail.com',
//         'smtp_user'=>
//         'cahayajayahome@gmail.com',
//         'smtp_pass'=> 'ecommerce',
//         'smtp_port'=> 465,
//         'mailtype'=> 'html',
//         'charset'=> 'utf-8',
//         'newline'=> "\r\n",
//     ];
//
//     $this->load->library('email', $config);
//
//     $this->email->from('cahayajayahome@gmail.com', 'Cahaya Jaya');
//
//     $this->email->to($this->input->post('email'));
//     //nginput email dari pendaftaran
//
//     $this->email->subject('activation');
//
//     $this->email->message('click this link to
verify your account');
//
//     if ($this->email->send()) {
//         return true;
//     } else {

```

```

//          echo $this->email->print_debugger();
//          }
//      }

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

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

//         if ($user) {
//             $user_token = $this->db-
>get_where('user_token', ['token' => $token])->row_array();

//             if ($user_token) {
//                 if (time() - $user_token['date_created'] <
(60 * 60 * 24)) {
//                     $this->db->where('email', $email);
//                     $this->db->update('user');

//                     $this->db->delete('user_token',
['email' => $email]);

//                     $this->session-
>set_flashdata('message', '<div class= "alert alert-success"
role="alert">' . $email . ' has been activated! Please
login.</div>');
//                     redirect('index');
//                 } else {
//                     $this->db->delete('user', ['email' =>
$email]);
//                     $this->db->delete('user_token',
['email' => $email]);

//                     $this->session-
>set_flashdata('message', '<div class="alert alert-danger"
role="alert">Account activation failed! Token
expired.</div>');
//                     redirect('index');
//                 }
//             } else {
//                 $this->session->set_flashdata('message',
'<div class="alert alert-danger" role="alert">Account
activation failed! Wrong token.</div>');
//                 redirect('index');
//             }
//         } else {

```

```

//          $this->session->set_flashdata('message', '<div
class="alert alert-danger" role="alert">Account activation
failed! Wrong email.</div>');
//          redirect('index');
//      }
//  }

public function login()
{
    $username = $this->input->post('username');
    $password = $this->input->post('password');

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

}

function cek_login(){
    $where = array(
        'username' => $this->input->post('username'),
        'password' => $this->input->post('password')
    );
    $cek = json_decode($this->curl->simple_get($this-
>API.'/login_seller',$where));
    if ($cek) {
        foreach ($cek as $sess) {
            $sess_data['account_id'] = $sess->account_id;
            $sess_data['name'] = $sess->name;
            $sess_data['grup'] = $sess->grup;
            $this->session->set_userdata($sess_data);
        }
        if ($this->session->userdata('grup')==1) {
            redirect('home_admin');
        }
        elseif ($this->session->userdata('grup')==2) {
            redirect('buyer/home_buyer');
        }
    }else{
        echo "<script>alert('Gagal login: Cek username,
password!');history.go(-1);</script>";
    }
}
}

```

9. Admin.php

```
<?php

class home extends CI_Model{

public function __construct() {
    $this->load->database();
}

    function getByCategory($uri){
        $this->db->join('categories c','p.categori_id =
c.categori_id');
        $q = $this->db->get_where('products p',array('p
categori_id' => $uri));
        return $q->result();
    }

    function process(){
        // $item =$this->cart->contents();

        $invoice = array(
            'buyer_id' => $this->session-
>userdata('account_id'),
            'price' => $this->cart->total(),
            'payment_status' => 'In Progress'
        );
        $this->db->insert('invoices',$invoice);
        $invoice_id = $this->db->insert_id();

        foreach($this->cart->contents() as $item){
            $a=$item['price'];
            $b=$item['qty'];
            $d=$item['account_id'];
            $c= $a * $b;
            $order = array(
                'invoice_id' => $invoice_id,
                'buyer_id' => $this->session-
>userdata('account_id'),
                'seller_id' => $item['account_id'],
                'product_id' => $item['id'],
                'price' => $c,
                'quantity' => $item['qty'],
                'shipping_status' => 'On Progress'
            );
            $this->db->insert('orders',$order);
        }

        return TRUE;
    }

public function stock($product_id,$qtyAkhir)
```



```

    {
        $this->db->where('product_id', $product_id);
        $data = array(
            'stock' => $qtyAkhir
        );
        return $this->db->update('products', $data);
    }

function shopping_history($account_id)
{
    $status = 'Already Received';
    $this->db->select('invoices.invoice_id,
products.product_id, products.product_name,
products.categori_id, products.product_image, account.name,
orders.price as p, orders.quantity, orders.shipping_status,
orders.order_id, account.account_id, orders.resi,
orders.courier');
    // $this->db->from('orders');
    $this->db->join('products', 'orders.product_id =
products.product_id');
    $this->db->join('invoices', 'orders.invoice_id =
invoices.invoice_id');
    $this->db->join('account', 'orders.buyer_id =
account.account_id');
    $this->db->join('account o', 'orders.seller_id =
o.account_id');
    $this->db->order_by("order_id", "desc");
    $data = $this->db-
>get_where('orders', array('orders.buyer_id' => $account_id,
'shipping_status' => $status)); // menampilkan data orderan
yang id user nya sama dengan session toko id
    return $data->result_array();
}

function shopping_history2($account_id)
{
    $data = $this->db->query("SELECT invoices.invoice_id,
products.product_id, products.product_name,
products.categori_id, products.product_image, account.name,
orders.price as p, orders.quantity, orders.shipping_status,
orders.order_id, account.account_id, orders.resi,
orders.courier FROM orders JOIN products ON orders.product_id
= products.product_id JOIN invoices ON orders.invoice_id =
invoices.invoice_id JOIN account ON orders.seller_id =
account.account_id WHERE shipping_status = 'On Progress' OR
shipping_status = 'On Delivery' order by order_id desc");
    return $data->result_array();
}

function shopping_history_admin()
{

```

```

        $data = $this->db->query("SELECT invoices.invoice_id,
invoices.payment_status, products.product_name,
invoices.date, account.name, orders.price,
orders.shipping_status, orders.resi, account.account_id,
orders.order_id FROM orders JOIN products ON
orders.product_id = products.product_id JOIN invoices ON
orders.invoice_id = invoices.invoice_id JOIN account ON
orders.seller_id = account.account_id WHERE shipping_status =
'On Progress' OR shipping_status = 'On Delivery' order by
order_id desc");

```

```

        return $data->result_array();

```

```

    }

```

```

function shopping_history_admin2()

```

```

{

```

```

    $this->db->select('invoices.invoice_id,
invoices.payment_status, products.product_name,
invoices.date, account.name, orders.price,
orders.shipping_status, orders.resi, account.account_id,
orders.order_id');

```

```

    // $this->db->from('orders');

```

```

    $this->db->join('products','orders.product_id =
products.product_id');

```

```

    $this->db->join('invoices','orders.invoice_id =
invoices.invoice_id');

```

```

    $this->db->join('account','orders.seller_id =
account.account_id');

```

```

    $this->db->order_by("order_id", "desc");

```

```

    $data = $this->db-
>get_where('orders',array('orders.shipping_status' =>
'Already Received'));

```

```

    return $data->result_array();

```

```

}

```

```

function invoice_detail($invoice_id)

```

```

{

```

```

    $this->db->select('orders.invoice_id,
products.product_id, products.categori_id,
products.product_image, orders.order_id, account.account_id,
invoices.price as i, products.product_name, account.name,
orders.quantity, products.price as p, orders.price as o,
orders.shipping_status, orders.resi, orders.courier');

```

```

    // $this->db->from('orders');

```

```

    $this->db->join('products','orders.product_id =
products.product_id');

```

```

    $this->db->join('invoices','orders.invoice_id =
invoices.invoice_id');

```

```

    $this->db->join('account','orders.seller_id =
account.account_id');

```

```

    $this->db->order_by("order_id", "desc");

```

```

        $data = $this->db-
>get_where('orders',array('orders.invoice_id' =>
$invoice_id));
        return $data->result_array();
    }

    function total_invoice($invoice_id)
    {
        $data = $this->db->query("SELECT price FROM invoices
where invoice_id = $invoice_id");
        return $data->result_array();
    }

    function order_in($account_id)
    {
        $status = 'On Progress';
        $this->db->select('orders.price as a,
invoices.payment_status, products.product_image,
products.product_name, orders.quantity, orders.order_id,
account.name, account.address, account.telp,
orders.shipping_status');
        // $this->db->from('orders');
        $this->db->join('products','orders.product_id =
products.product_id');
        $this->db->join('invoices','orders.invoice_id =
invoices.invoice_id');
        $this->db->join('account as o','orders.seller_id
= o.account_id');
        $this->db->join('account','orders.buyer_id =
account.account_id');
        $this->db->order_by("order_id", "desc");
        $data = $this->db-
>get_where('orders',array('orders.seller_id' => $account_id,
'orders.shipping_status' => $status)); // menampilkan data
orderan yang id user nya sama dengan session toko id
        return $data->result_array();
    }

    function order_in2($account_id)
    {
        $status = 'On Delivery';
        $this->db->select('orders.price as a,
invoices.payment_status, products.product_image,
products.product_name, orders.quantity, orders.order_id,
account.name, account.address, account.telp,
orders.shipping_status, orders.resi, orders.courier');
        // $this->db->from('orders');
        $this->db->join('products','orders.product_id =
products.product_id');
        $this->db->join('invoices','orders.invoice_id =
invoices.invoice_id');
    }

```

```

        $this->db->join('account as o','orders.seller_id
= o.account_id');
        $this->db->join('account','orders.buyer_id =
account.account_id');
        $this->db->order_by("order_id", "desc");
        $data = $this->db-
>get_where('orders',array('orders.seller_id' => $account_id,
'orders.shipping_status' => $status)); // menampilkan data
orderan yang id user nya sama dengan session toko
payment_status
        return $data->result_array();
    }

    function order_in3($account_id)
    {
        $status = 'Already Received';
        $this->db->select('orders.price as a,
invoices.payment_status, products.product_image,
products.product_name, orders.quantity, orders.order_id,
account.name, account.address, account.telp,
orders.shipping_status, orders.resi, orders.courier');
        // $this->db->from('orders');
        $this->db->join('products','orders.product_id =
products.product_id');
        $this->db->join('invoices','orders.invoice_id =
invoices.invoice_id');
        $this->db->join('account as o','orders.seller_id
= o.account_id');
        $this->db->join('account','orders.buyer_id =
account.account_id');
        $this->db->order_by("order_id", "desc");
        $data = $this->db-
>get_where('orders',array('orders.seller_id' => $account_id,
'orders.shipping_status' => $status)); // menampilkan data
orderan yang id user nya sama dengan session toko
payment_status
        return $data->result_array();
    }

    function my_product($account_id)
    {
        $this->db->select('*');
        // $this->db->from('orders');
        $this->db->join('account','products.account_id =
account.account_id');
        $this->db-
>join('categories','products.categori_id =
categories.categori_id');
        $data = $this->db-
>get_where('products',array('products.account_id' =>

```

```

$account_id)); // menampilkan data orderan yang id user nya
sama dengan session toko id
    return $data->result();
}

function invoice_history($account_id)
{
    $status = 'Paid';
    $this->db->select('*');
    // $this->db->from('invoices as c');
    // $this->db->join('products k','u.product_id =
k.product_id');
    $this->db->join('account','invoices.buyer_id =
account.account_id');
    // $this->db->join('account as o','u.seller_id =
o.account_id');
    $this->db->order_by("date", "desc");
    $data = $this->db-
>get_where('invoices',array('invoices.buyer_id' =>
$account_id, 'payment_status' => $status)); // menampilkan
data orderan yang id user nya sama dengan session toko id
    return $data->result();
}

function invoice_history2($account_id)
{
    $status = 'In Progress';
    $this->db->select('*');
    // $this->db->from('invoices as c');
    // $this->db->join('products k','u.product_id =
k.product_id');
    $this->db->join('account','invoices.buyer_id =
account.account_id');
    // $this->db->join('account as o','u.seller_id =
o.account_id');
    $this->db->order_by("date", "desc");
    $data = $this->db-
>get_where('invoices',array('invoices.buyer_id' =>
$account_id, 'payment_status' => $status)); // menampilkan
data orderan yang id user nya sama dengan session toko id
    return $data->result();
}

function saldo_invoice($account_id)
{
    $this->db->where('account_id',$account_id); //
menampilkan data orderan yang id user nya sama dengan session
toko id
    $data = $this->db->get('account');
    return $data->result();
}

```

```

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

function stock_cart(){
    return $this->db->get('products');
}

function update_account($id,$data_account){
    // print_r($u); die;
    return $this->db->
>update('account',$data_account,array('account_id'=>$id));
}

function category(){
    // print_r($u); die;
    $a=$this->db->get('categories');
    return $a->result();
}

function category_admin(){
    // print_r($u); die;
    $a=$this->db->get('categories');
    return $a->result_array();
}

function category_product($id){
    // print_r($u); die;
    $a=$this->db->get('categories');
    return $a->result();
}

function shipping_status($id,$status,$resi,$courier){
    $data = $this->db->query("UPDATE orders set
shipping_status = '$status',resi = '$resi',courier =
'$courier' Where order_id = $id");
    return $data;
}

function shipping_status_buyer($id,$status){
    $data = $this->db->query("UPDATE orders set
shipping_status = '$status' Where order_id = $id");
    return $data;
}

function comment(){
    $data = array(
        'account_id' => $this->session-
>userdata('account_id'),
        'product_id' => $this->input->post('product_id'),

```

```

        'comment' => $this->input->post('comment')
    );
    $a = $this->db->insert('comment', $data);
    return $a;
}

public function voucher()
{
    $code = $this->input->GET('code', TRUE);
    $data = $this->db->query("SELECT * from voucher where
code = '$code' and status = 'Not used'");
    return $data->result();
}

public function saldo($id)
{
    $status = 'Not Used';
    $this->db->order_by("date", "desc");
    $this->db->where(array('account_id' => $id, 'status'
=> $status)); // menampilkan data orderan yang id user nya
sama dengan session toko id
    $data = $this->db->get('topup');
    return $data->result();
}

public function saldo2($id)
{
    $status = 'Used';
    $this->db->order_by("date", "desc");
    $this->db->where(array('account_id' => $id, 'status'
=> $status)); // menampilkan data orderan yang id user nya
sama dengan session toko id
    $data = $this->db->get('topup');
    return $data->result();
}

public function status_voucher($id)
{
    $a = 'Used';
    $data = $this->db->query("UPDATE voucher set status =
'$a' Where voucher_id = $id");
    return $data;
}

public function clear_payment($price,$account_id)
{
    $data = $this->db->query("UPDATE account set saldo =
saldo + $price Where account_id = $account_id");
    return $data;
}

```

```

public function clear_status($id)
{
    $a = 'Already Received';
    $data = $this->db->query("UPDATE orders set
shipping_status = '$a' Where order_id = $id");
    return $data;
}

public function tambah_saldo($id,$nominal)
{
    $a = 'Used';
    $data = $this->db->query("UPDATE account set saldo =
saldo + $nominal Where account_id = $id");
    return $data;
}

public function status_topup($uniq_key)
{
    $a = 'Used';
    $data = $this->db->query("UPDATE topup set status =
'$a' Where uniq_key = $uniq_key");
    return $data;
}

public function pay_to_seller($id,$total)
{
    $data = $this->db->query("UPDATE account set saldo =
saldo + $total Where account_id = $id");
    return $data;
}

public function payment($invoice_id)
{
    $data = $this->db->query("UPDATE invoices set
payment_status = 'Paid' Where invoice_id = $invoice_id");
    return $data;
}

public function update_saldo($price)
{
    $account_id = $this->session->userdata('account_id');
    $data = $this->db->query("UPDATE account set saldo =
saldo - $price Where account_id = $account_id");
    return $data;
}

function topup($nominal,$id,$status){
    $angka = range(0,9);
    shuffle($angka);
    $uniq = array_rand($angka,6);

```



```

        $unique = implode($uniq);
        $data = array(
            'account_id' => $this->session-
>userdata('account_id'),
            'uniq_key' => $unique,
            'nominal' => $nominal,
            'status' => $status
        );
        $a = $this->db->insert('topup', $data);
        return $a;
    }

    public function registration($data)
    {
        $data = $this->db->insert("account", $data);
    }
}

```

10. Index_admin.php

```

<!DOCTYPE html>
<!--[if lt IE 7]>      <html class="no-js lt-ie9 lt-ie8 lt-
ie7"> <![endif]-->
<!--[if IE 7]>        <html class="no-js lt-ie9 lt-ie8">
<![endif]-->
<!--[if IE 8]>        <html class="no-js lt-ie9"> <![endif]-
->
<!--[if gt IE 8]><!-->
<html class="no-js"> <!--<![endif]-->
<head>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
    <meta name="description" content="Cahaya Jaya Poltekpos">

    <meta name="author" content="Themefisher.com">

    <title>Cahaya Jaya</title>

    <!-- Mobile Specific Meta-->
    <meta name="viewport" content="width=device-width, initial-
scale=1">

    <!-- Favicon -->
    <link rel="shortcut icon" type="image/x-icon" href="<?php
echo base_url('assets/images/favicon.png')?>" />

    <!-- Themefisher Icon font -->
    <link rel="stylesheet" href="<?php echo
base_url('assets/plugins/themefisher-font/style.css')?>">
    <!-- bootstrap.min css -->

```

```

<link rel="stylesheet" href="<?php echo
base_url('assets/plugins/bootstrap/css/bootstrap.min.css')?>"
>

<link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/plugins/slick-
carousel/slick/slick.css')?>" />
<link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/plugins/slick-carousel/slick/slick-
theme.css')?>" />

<!-- Main Stylesheet -->
<link rel="stylesheet" href="<?php echo
base_url('assets/css/style.css')?>">

</head>

<body id="body">

<!-- Start Top Header Bar -->
<section class="top-header">
<div class="container">
<div class="row">
<div class="col-md-4 col-xs-12 col-sm-4">
<div class="contact-number">
<i class="tf-ion-ios-telephone"></i>
<span>0129- 12323-123123</span>
</div>
</div>
<div class="col-md-4 col-xs-12 col-sm-4">
<!-- Site Logo -->
<div class="logo text-center">
<a href="index.html">
<!-- replace logo here -->
<svg width="270px" height="29px" viewBox="0 0 155
29" version="1.1" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">
<g id="Page-1" stroke="none" stroke-width="1"
fill="none" fill-rule="evenodd" font-size="30" font-
family="AustinBold, Austin" font-weight="bold">
<g id="Group" transform="translate(-
108.000000, -297.000000)" fill="#000000">
<text id="AVIATO">
<tspan x="80.94" y="325">CAHAYA
JAYA</tspan>
</text>
</g>
</g>
</svg>
</a>
</div>

```

```

    </div>
    <div class="col-md-4 col-xs-12 col-sm-4">
    <!-- Cart -->
    <ul class="top-menu text-right list-inline">

        <li>
            <a href="<?php echo base_url().
'buyer/home_buyer/logout'; ?>"><i class="tf-ion-log-out"></i>
Logout</a>
        </li>

    </ul><!-- / .nav .navbar-nav .navbar-right -->
    </div>
</div>
</div>
</section><!-- End Top Header Bar -->

<!-- Main Menu Section -->
<section class="menu">
    <nav class="navbar navigation">
        <div class="container">
            <div class="navbar-header">
                <h2 class="menu-title">Main Menu</h2>
                <button type="button" class="navbar-toggle
collapsed" data-toggle="collapse" data-target="#navbar" aria-
expanded="false" aria-controls="navbar">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>

            </div><!-- / .navbar-header -->

            <!-- Navbar Links -->
            <div id="navbar" class="navbar-collapse collapse
text-center">
                <ul class="nav navbar-nav">

                    <li class="dropdown dropdown-slide">
                        <a class="dropdown-toggle" data-
toggle="dropdown" data-hover="dropdown" data-delay="350"
aria-haspopup="true" aria-expanded="false"><i class="tf-ion-
home"></i>&nbsp;Home<span class="tf-ion-ios-arrow-
down"></span></a>
                        <ul class="dropdown-menu">
                            <li><a href="<?php echo base_url(). 'home_admin';
?>">Progress</a></li>
                            <li><a href="<?php echo base_url().
'home_admin/index2'; ?>">Clear</a></li>

```

```

        </ul>
    </li>

    <!-- Elements -->
    <li>
        <a href="<?php echo base_url().
'home_admin/category'; ?>"><i class="tf-ion-
archive"></i>&nbsp;Categories</a>
    </li><!-- / Elements -->
    <!-- Elements -->

    </ul><!-- / .nav .navbar-nav
    </div><!--/.navbar-collapse -->
    </div><!-- / .container -->
</nav>
</section>

<section class="products section bg-gray">
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <div class="title text-center">
                    <h2><i class="tf-ion-ios-
briefcase"></i>&nbsp;Shopping History Customer</h2>
                </div>
            </div>
        </div>
        <table class="table">
            <thead>
                <tr>
                    <th width="3%" scope="col">No.</th>
                    <th width="3%" scope="col">Invoice</th>
                    <th width="15%" scope="col">Payment Status</th>
                    <th width="15%" scope="col">Name Product</th>
                    <th width="15%" scope="col">Order Date</th>
                    <th width="15%" scope="col">Seller</th>
                    <th width="10%" scope="col">Total</th>
                    <th width="13%" scope="col">Shipping Status</th>
                    <th width="13%" scope="col">Resi</th>
                    <th width="13%" scope="col">Action</th>
                </tr>
            </thead>
            <?php
            $i=0;
            foreach ($order as $key) {
                $i++;
                ?>
            <tbody>
                <tr>
                    <th scope="row"><?php echo $i ?></th>
                    <td><?php echo $key['invoice_id'] ?></td>

```

```

        <td><?php echo $key['payment_status'] ?></td>
        <td><?php echo $key['product_name'] ?></td>
        <td><?php echo $key['date'] ?></td>
        <td><?php echo $key['name'] ?></td>
        <td><b><?php echo $key['Rp'] ?></b></td>
        <td><?php echo $key['price'],0,'','.' ?></td>
        <td><?php echo $key['shipping_status'] ?> <br>
        <td><?php echo $key['resi'] ?> <br>
        <td>
            <form action="<?php echo
base_url() ?>home_admin/clear_payment" method="post">
                <?php if ($key['payment_status'] == 'Paid') { ?>
                    <?php if ($key['shipping_status'] == 'Already
Received') { ?>
                        <button type="submit" class='btn btn-sm btn-
primary' disabled>Clear</a>
                        <?php }elseif ($key['shipping_status'] == 'On
Delivery') { ?>
                            <input type="hidden" name="price" value="<?php echo
$key['price'] ?>"></input>
                            <input type="hidden" name="account_id" value="<?php
echo $key['account_id'] ?>"></input>
                            <input type="hidden" name="order_id" value="<?php
echo $key['order_id'] ?>"></input>
                            <button type="submit" onclick="return
validasi();" class='btn btn-sm btn-primary'>Clear</a>
                            <?php }elseif ($key['shipping_status'] == 'On
Progress') { ?>
                                <button type="submit" class='btn btn-sm btn-
primary' disabled>Clear</a>
                                <?php } ?>
                            <?php }elseif ($key['payment_status'] == 'In Progress')
{ ?>
                                <?php if ($key['shipping_status'] == 'Already
Received') { ?>
                                    <button type="submit" class='btn btn-sm btn-
primary' disabled>Clear</a>
                                    <?php }elseif ($key['shipping_status'] == 'On
Progress') { ?>
                                        <input type="hidden" name="price" value="<?php echo
$key['price'] ?>"></input>
                                        <input type="hidden" name="account_id" value="<?php
echo $key['account_id'] ?>"></input>
                                        <input type="hidden" name="order_id" value="<?php
echo $key['order_id'] ?>"></input>
                                        <button type="submit" class='btn btn-sm btn-
primary' disabled>Clear</a>
                                        <?php } ?>
                                    <?php } ?>
                                </form>
                            </td>

```

```
        </tr>
    </form>
</tbody>
<?php } ?>
```

```
</table>
```

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

```
<!--
```

```
=====
=====
```

```
> -->
```

```
<!--
```

```
Start Call To Action
```

```
===== -->
```

```
<section class="section instagram-feed">
```

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

```
        <div class="row">
```

```
            <div class="col-md-12">
```

```
                </div>
```

```
        </div>
```

```
    </div>
```

```
</section>
```

```
<footer class="footer section text-center">
```

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

```
        <div class="row">
```

```
            <div class="col-md-12">
```

```
                <ul class="social-media">
```

```
                    <li>
```

```
                        <a href="">
```

```
                            <i class="tf-ion-social-facebook"></i>
```

```
                        </a>
```

```
                    </li>
```

```
                    <li>
```

```
                        <a href="">
```

```
                            <i class="tf-ion-social-instagram"></i>
```

```
                        </a>
```

```
                    </li>
```

```
                    <li>
```

```
                        <a href="">
```

```
                            <i class="tf-ion-social-twitter"></i>
```

```
                        </a>
```

```
                    </li>
```

```
                    <li>
```

```

        <a href="">
            <i class="tf-ion-social-pinterest"></i>
        </a>
    </li>
</ul>
<ul class="footer-menu">
    <li>
        <a href="">CONTACT</a>
    </li>
    <li>
        <a href="">SHIPPING</a>
    </li>
    <li>
        <a href="">TERMS OF SERVICE</a>
    </li>
    <li>
        <a href="">PRIVACY POLICY</a>
    </li>
</ul>
<p class="copyright-text">Powered by Bootstrap</p>
</div>
</div>
</div>
</footer>

```

```

<!--
Essential Scripts
=====-->

```

```

<!-- Main jQuery -->
<script src="https://code.jquery.com/jquery-
git.min.js"></script>
<!-- Bootstrap 3.1 -->
<script src="<?php echo
base_url('assets/plugins/bootstrap/js/bootstrap.min.js')?>">
</script>
<!-- Instagram Feed Js -->
<script src="<?php echo
base_url('assets/plugins/instafeed.js/instafeed.min.js')?>">
</script>
<!-- Slick Carousel -->
<script src="<?php echo base_url('assets/plugins/slick-
carousel/slick/slick.min.js')?>"></script>
<!-- Google Map js -->
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBItrd4
sQ_aXlQG_fvEzsxvuYyaWnJKETk&callback=initMap"></script>

```

```

        <script type="text/javascript" src="<?php echo
base_url('assets/plugins/bootstrap/jquery/dist/jquery.js')?>"
></script>
        <script type="text/javascript" src="<?php echo
base_url('assets/plugins/bootstrap/js/bootstrap.js')?>"></scr
ipt>

        <!-- Main Js File -->
        <script src="<?php echo
base_url('assets/js/script.js')?>"></script>

        <script type="text/javascript">
            function validasi() {
                var msg= confirm("Apakah Anda Yakin Ingin
Mengubah Data Ini?");
                if (msg){
                    return true ;
                }else{
                    return false ;
                }
            }
        </script>

</body>
</html>

```

11.

```

<section class="page-header">
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <div class="content">
                    <h1 class="page-name">Account</h1>
                </div>
            </div>
        </div>
    </div>
</section>

```

```

<section class="page-wrapper bg-gray">
    <div class="contact-section">
        <div class="container">
            <div class="row">

```



```

<!-- Contact Details -->

<?php foreach ($account as $key) { ?>
<div class="contact-details col-md-6 " >
    <div class="google-map">
        
    </div>

</div>

<div class="contact-details col-md-6 " >
<ul class="contact-short-info" >
    <li>
        <i class="tf-ion-edit" style="font-size:
30px"></i>
        <span style="font-size: 20px"> Name: <?php
echo $key->name ?></span>
    </li>
    <li>
        <i class="tf-ion-person" style="font-size:
30px"></i>
        <span style="font-size: 20px"> Username: <?php
echo $key->username ?></span>
    </li>
    <li>
        <i class="tf-ion-key" style="font-size:
30px"></i>
        <span style="font-size: 20px"> Password:
*****</span>
    </li>
    <li>
        <i class="tf-ion-android-mail" style="font-
size: 30px"></i>
        <span style="font-size: 20px"> Email: <?php
echo $key->email ?></span>
    </li>
    <li>
        <i class="tf-ion-ios-home" style="font-size:
30px"></i>
        <span style="font-size: 20px"> Address: <?php
echo $key->address ?></span>
    </li>
    <li>
        <i class="tf-ion-map" style="font-size:
30px"></i>
        <span style="font-size: 20px"> City: <?php
echo $key->city ?></span>
    </li>

```

```

        <li>
            <i class="tf-ion-android-phone-portrait"
style="font-size: 30px"></i>
            <span style="font-size: 20px"> Phone: <?php
echo $key->telp ?></span>
        </li>

        <li>
            <i class="tf-ion-card" style="font-size:
30px"></i>
            <span style="font-size: 20px"> Rekening: <?php
echo $key->rekening ?></span>
        </li>

        <li>
            <i class="tf-ion-cash" style="font-size:
30px"></i>
            <span style="font-size: 20px"><?= 'Rp.
'.number_format($key->saldo,0,',','.'); ?></span>
        </li>
        <br>
        <li>
            <!-- <span data-toggle="modal" data-
target="#product-modal<?php echo $key->account_id?>"
class="btn btn-main">Edit Profile</span> -->
            <button type="button" class='btn btn-main' data-
toggle='modal' data-target='#product-modal<?= $key-
>account_id; ?>'>Edit Profile</button>
        </li>

    </ul>

</div>

<!-- Modal -->
<div class="modal fade" id="product-modal<?php echo
$key->account_id;?>">
    <button type="button" class="close" data-
dismiss="modal" aria-label="Close">
        <i class="tf-ion-close"></i>
    </button>
    <div class="modal-dialog " role="document">
        <div class="modal-content">

            <div class="modal-header">
                <h5 class="modal-title">Edit
Profile</h5>
                <button type="button"
class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-
hidden="true">&times;</span>

```

```

        </button>
    </div>
<div class="modal-body">

    <form action="<?php echo base_url().
'buyer/home_buyer/update_account'; ?>" method="post"
enctype="multipart/form-data">

        <input style="font-size: 15px"
style="font-size: 15px" type="hidden" class="form-control"
name="id" value="<?php echo $key->account_id?>">
        <input style="font-size: 15px"
style="font-size: 15px" type="hidden" class="form-control"
name="image2" value="<?php echo $key->image?>">

        <div class="row">
            <div class="col-md-6">
                <div class="form-group">
                    <span>Nama</span>
                    <input style="font-size: 15px"
style="font-size: 15px" type="text" class="form-control"
name="name" value="<?php echo $key->name?>">
                </div>
            </div>

            <div class="col-md-6">
                <div class="form-group">
                    <span>Foto</span>
                    <input type="file" class="form-control"
name="image">
                </div>
            </div>
        </div>

        <div class="row">
            <div class="col-md-6">
                <div class="form-group">
                    <span>Username</span>
                    <input style="font-size: 15px"
type="text" class="form-control" name="username" value="<?php
echo $key->username?>">
                </div>
            </div>

            <div class="col-md-6">
                <div class="form-group">
                    <span>Password</span>
                    <input style="font-size: 15px"
type="text" class="form-control" name="password" value="<?php
echo $key->password?>">
                </div>
            </div>
        </div>
    </div>

```

```

    </div>
  </div>

  <div class="row">
    <div class="col-md-6">
      <div class="form-group">
        <span>Email</span>
        <input style="font-size: 15px"
type="text" class="form-control" name="email" value="<?php
echo $key->email?>">
      </div>
    </div>

    <div class="col-md-6">
      <div class="form-group">
        <span>No. Telp</span>
        <input style="font-size: 15px"
type="text" class="form-control" name="telp" value="<?php
echo $key->telp?>">
      </div>
    </div>
  </div>

  <div class="row">
    <div class="col-md-12">
      <div class="form-group">
        <span>Alamat</span>
        <textarea
name="address" value="<?= $key->address;?>" class="form-
control border-input" required cols="30" rows="5"><?php echo
$key->address?></textarea>
      </div>
    </div>
  </div>

  <div class="row">
    <div class="col-md-6">
      <div class="form-group">
        <span>Kota</span>
        <input style="font-size: 15px"
type="text" class="form-control" name="city" value="<?php
echo $key->city?>">
      </div>
    </div>

    <div class="col-md-6">
      <div class="form-group">
        <span>Rekening</span>
        <input style="font-size: 15px"
type="text" class="form-control" name="rekening" value="<?php
echo $key->rekening?>">
      </div>
    </div>
  </div>

```

```

        </div>
    </div>
    <button type="submit" class="btn btn-
main">Edit</button>
    <button type="button" class="btn btn-
secondary" data-dismiss="modal">Close</button>
    </form>
</div>
</div>
</div>
</div>

```

```
<?php } ?>
```

```

    </div> <!-- end row -->
</div> <!-- end container -->
</div>
</section>

```

12. Cart.php

```

<section class="products section bg-gray">
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <div class="title text-center">
                    <h2>Shopping Cart</h2>
                </div>
            </div>
        </div>
        <table class="table">
            <thead>
                <tr>

                    <th width="2%" scope="col">No.</th>
                    <th width="10%" scope="col">Image</th>
                    <th width="33%" scope="col">Name Product</th>
                    <th width="17%" scope="col">Seller</th>
                    <th width="8%" scope="col">Quantity</th>
                    <th width="15%" scope="col">Price */stock</th>
                    <th width="15%" scope="col">Total</th>
                    <th width="10%" scope="col">Remove</th>
                </tr>
            </thead>
            <tbody>
                <?php

```

```

        $i=0;
        $total=0;
        // $this-> cart contents[$rowid]['rowid'] = $rowid;
        foreach ($this->cart->contents() as $items):
            $subtotal = ($items['qty'] * $items['price']);
            // $grand_total=$total + $items['']
            $i++;
        ?>
    <tr>
        <th scope="row"><?php echo $i?></th>
        <td></td>
        <td hidden><?php echo $items['product_id'] ?></td>
        <td><?php echo $items['name'] ?></td>
        <td><?php echo $items['nama'] ?></td>
        <td><?php echo $items['qty'] ?></td>
        <td><?php echo 'Rp.
'.number_format($items['price'],0,',','.') ?></td>

        <td><?php echo 'Rp.
'.number_format($subtotal,0,',','.') ?></td>

    <td>
    <center>
    <a href="<?php echo
base_url('buyer/home_buyer/remove_item/' . $items['rowid']);
?>" class="remove"><i class="tf-ion-close"></i></a>
    <?php endforeach; ?>
    </center>
    </td>
    </tr>
</tbody>
<tr>
    <td colspan="3"><b>Order Total: <?= 'Rp.
'.number_format($this->cart->total(),0,',','.') ?></b></td>
    <td colspan="4" align="right">
    <a href="<?php echo
base_url()??>buyer/home_buyer/clear_cart" class='btn btn-sm
btn-danger'>Empty The Cart</a>
    <a href="<?php echo base_url()??>buyer/home_buyer/shop"
class='btn btn-sm btn-success' type="submit">Update Cart</a>

    <a href="<?php echo base_url()??>buyer/home_buyer/order"
class='btn btn-sm btn-primary'>Check Out</a>
    </td>
</tr>
</table>

</div>

```

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

Kodingan patchregister

Autoload.php

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

/*
| -----
| AUTO-LOADER
| -----
| This file specifies which systems should be loaded by
| default.
|
| In order to keep the framework as light-weight as possible
| only the
| absolute minimal resources are loaded by default. For
| example,
| the database is not connected to automatically since no
| assumption
| is made regarding whether you intend to use it. This file
| lets
| you globally define which systems you would like loaded
| with every
| request.
|
| -----
| Instructions
| -----
|
| These are the things you can load automatically:
|
| 1. Packages
| 2. Libraries
| 3. Drivers
| 4. Helper files
| 5. Custom config files
| 6. Language files
```

```

| 7. Models
|
*/

/*
| -----
|
| Auto-load Packages
| -----
|
| Prototype:
|
| $autoload['packages'] = array(APPPATH.'third_party',
| '/usr/local/shared');
|
*/
$autoload['packages'] = array();

/*
| -----
|
| Auto-load Libraries
| -----
|
| These are the classes located in system/libraries/ or your
| application/libraries/ directory, with the addition of the
| 'database' library, which is somewhat of a special case.
|
| Prototype:
|
| $autoload['libraries'] = array('database', 'email',
| 'session');
|
| You can also supply an alternative library name to be
| assigned
| in the controller:
|
| $autoload['libraries'] = array('user_agent' => 'ua');
*/
$autoload['libraries'] = array('email', 'session',
'database', 'form_validation');

/*
| -----
|
| Auto-load Drivers
| -----
|
| These classes are located in system/libraries/ or in your
| application/libraries/ directory, but are also placed
| inside their

```



```

| own subdirectory and they extend the CI_Driver_Library
class. They
| offer multiple interchangeable driver options.
|
| Prototype:
|
|     $autoload['drivers'] = array('cache');
|
| You can also supply an alternative property name to be
assigned in
| the controller:
|
|     $autoload['drivers'] = array('cache' => 'cch');
|
| */
$autoload['drivers'] = array();

/*
| -----
|
| Auto-load Helper Files
| -----
|
| Prototype:
|
|     $autoload['helper'] = array('url', 'file');
| */
$autoload['helper'] = array('url', 'file', 'security',
'wpu');

/*
| -----
|
| Auto-load Config files
| -----
|
| Prototype:
|
|     $autoload['config'] = array('config1', 'config2');
|
| NOTE: This item is intended for use ONLY if you have
created custom
| config files. Otherwise, leave it blank.
|
| */
$autoload['config'] = array();

/*
| -----
|
| Auto-load Language files

```

```

| -----
| -----
| Prototype:
|
|     $autoload['language'] = array('lang1', 'lang2');
|
| NOTE: Do not include the "_lang" part of your file.  For
example
| "codeigniter_lang.php" would be referenced as
array('codeigniter');
|
| */
$autoload['language'] = array();

/*
| -----
| -----
| Auto-load Models
| -----
| -----
| Prototype:
|
|     $autoload['model'] = array('first_model',
'second_model');
|
| You can also supply an alternative model name to be
assigned
| in the controller:
|
|     $autoload['model'] = array('first_model' => 'first');
| */
$autoload['model'] = array();

```

Database.php

```

$active_group = 'default';
$query_builder = true;

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'patchregister',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => false,
    'db_debug' => (ENVIRONMENT !== 'production'),

```

```

        'cache_on' => false,
        'cachedir' => '',
        'char_set' => 'utf8',
        'dbcollat' => 'utf8_general_ci',
        'swap_pre' => '',
        'encrypt' => false,
        'compress' => false,
        'stricton' => false,
        'failover' => array(),
        'save_queries' => true
    );

```

Admin.php

```

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

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

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

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

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

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

```

```

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

    public function roleAccess($role_id)
    {
        $data['title'] = 'Role Access';
        $data['user'] = $this->db->get_where('user', ['email'
=> $this->session->userdata('email')])>row_array();

        $data['role'] = $this->db->get_where('user_role',
['id' => $role_id])>row_array();

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

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

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

        $data = [
            'role_id' => $role_id,
            'menu_id' => $menu_id
        ];

        $result = $this->db->get_where('user_access_menu',
$data);

        if ($result->num_rows() < 1) {
            $this->db->insert('user_access_menu', $data);
        } else {
            $this->db->delete('user_access_menu', $data);
        }
    }

```

```

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

```

Auth.php

```

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

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

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

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

        if ($this->form_validation->run() == false) {
            $data['title'] = 'Login Page';
            $this->load->view('templates/auth_header',
$data);
            $this->load->view('auth/login');
            $this->load->view('templates/auth_footer');
        } else {
            // validasinya success
            $this->_login();
        }
    }

    private function _login()
    {

```

```

$email = $this->input->post('email');
$password = $this->input->post('password');

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

// jika usernya ada
if ($user) {
    // jika usernya aktif
    if ($user['is_active'] == 1) {
        // cek password
        if (password_verify($password,
$user['password'])) {
            $data = [
                'email' => $user['email'],
                'role_id' => $user['role_id']
            ];
            $this->session->set_userdata($data);
            if ($user['role_id'] == 1) {
                redirect('admin');
            } else {
                redirect('user');
            }
        } else {
            $this->session->set_flashdata('message',
'<div class="alert alert-danger" role="password
salah!</div>');
            redirect('auth');
        }
    } else {
        $this->session->set_flashdata('message',
'<div class="alert alert-danger" role="alert">This email has
not been activated!</div>');
        redirect('auth');
    }
} else {
    $this->session->set_flashdata('message', '<div
class="alert alert-danger" role="alert">Email is not
registered!</div>');
    redirect('auth');
}
}

public function registration()
{
    if ($this->session->userdata('email')) {
        redirect('user');
    }
    $this->form_validation->set_rules('username',
'username', 'required|trim');

```

```

        $this->form_validation->set_rules('name', 'Name',
'required|trim');
        $this->form_validation->set_rules('telp', 'telp',
'required|trim');
        $this->form_validation->set_rules('email', 'Email',
'required|trim|valid_email|is_unique[user.email]', [
            'is_unique' => 'This email has already
registered!'
        ]);
        $this->form_validation->set_rules('password1',
'Password', 'required|trim|min_length[3]|matches[password2]', [
            'matches' => 'Password dont match!',
            'min_length' => 'Password too short!'
        ]);
        $this->form_validation->set_rules('password2',
'Password', 'required|trim|matches[password1]');

        if ($this->form_validation->run() == false) {
            $data['title'] = 'WPU User Registration';
            $this->load->view('templates/auth_header',
$data);
            $this->load->view('auth/registration');
            $this->load->view('templates/auth_footer');
        } else {
            $email = $this->input->post('email', true);
            $data = [
                'username' => htmlspecialchars($this->input-
>post('username', true)),
                'name' => htmlspecialchars($this->input-
>post('name', true)),
                'telp' => htmlspecialchars($this->input-
>post('telp', true)),
                'email' => htmlspecialchars($email),
                'image' => 'default.jpg',
                'password' => password_hash($this->input-
>post('password1'), PASSWORD_DEFAULT),
                'role_id' => 2,
                'is_active' => 0,
                'date_created' => time()
            ];

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

            $this->db->insert('user', $data);

```

```

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

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

        $this->session->set_flashdata('message', '<div
class="alert alert-success" role="alert">Selamat Account Anda
Telah Dibuat. Mohon activate account anda</div>');
        redirect('auth');
    }
}

private function _sendwhatsapp($token, $type)
{
    $my_apikey = "MBJ569AQF9GNRQREX75E";
    $destination = "DESTINATION";
    $message = "MESSAGE TO SEND";
    $api_url =
"http://panel.rapiwha.com/send_message.php";
    $api_url .= "?apikey=" . urlencode ($my_apikey);
    $api_url .= "&number=" . urlencode ($destination);
    $api_url .= "&text=" . urlencode ($message);
    $my_result_object =
json_decode(file_get_contents($api_url, false));
    echo "<br>Result: " . $my_result_object->success;
    echo "<br>Description: " . $my_result_object->
>description;
    echo "<br>Code: " . $my_result_object->result_code;

    $this->message->initialize($config);

    $this->message->from('cahayajayahome@gmail.com',
'contoh');
    $this->message->to($this->input->post('email'));

    if ($type == 'verify') {
        $this->message->subject('Account Verification');
        $this->message->message('Click this link to
verify you account : <a href="' . base_url() .
'auth/verify?email=' . $this->input->post('email') .
'&token=' . urlencode($token) . '"'>Activate</a>');
    } else if ($type == 'forgot') {
        $this->message->subject('Reset Password');
        $this->message->message('Click this link to reset
your password : <a href="' . base_url() .
'auth/resetpassword?email=' . $this->input->post('email') .
'&token=' . urlencode($token) . '"'>Reset Password</a>');
    }

    if ($this->message->send()) {
        return true;
    } else {

```



```

        echo $this->message->print_debugger();
        die;
    }
}
private function _sendEmail($token, $type)
{
    $config = [
        'protocol' => 'smtp',
        'smtp_host' => 'ssl://smtp.googlemail.com',
        'smtp_user' => 'cahayajayahome@gmail.com',
        'smtp_pass' => 'ecommerce',
        'smtp_port' => 465,
        'mailtype' => 'html',
        'charset' => 'utf-8',
        'newline' => "\r\n"
    ];

    $this->email->initialize($config);

    $this->email->from('cahayajayahome@gmail.com',
'contoh');
    $this->email->to($this->input->post('email'));

    if ($type == 'verify') {
        $this->email->subject('Account Verification');
        $this->email->message('Click this link to verify
you account : <a href="' . base_url() . 'auth/verify?email='
. $this->input->post('email') . '&token=' . urlencode($token)
. '">Activate</a>');
    } else if ($type == 'forgot') {
        $this->email->subject('Reset Password');
        $this->email->message('Click this link to reset
your password : <a href="' . base_url() .
'auth/resetpassword?email=' . $this->input->post('email') .
'&token=' . urlencode($token) . '">Reset Password</a>');
    }

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

public function verifywa()
{
    $message = $this->input->get('telp');
    $token = $this->input->get('token');

```

```

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

        if ($user) {
            $user_token = $this->db->get_where('user_token',
['token' => $token])->row_array();

            if ($user_token) {
                if (time() - $user_token['date_created'] <
(60 * 60 * 24)) {
                    $this->db->set('is_active', 1);
                    $this->db->where('telp', $stelp);
                    $this->db->update('user');

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

                    $this->session->set_flashdata('message',
'<div class="alert alert-success" role="alert">' . $email . '
has been activated! Please login.</div>');
                    redirect('auth');
                } else {
                    $this->db->delete('user', ['email' =>
$email]);
                    $this->db->delete('user_token', ['email'
=> $email]);

                    $this->session->set_flashdata('message',
'<div class="alert alert-danger" role="alert">Account
activation failed! Token expired.</div>');
                    redirect('auth');
                }
            } else {
                $this->session->set_flashdata('message',
'<div class="alert alert-danger" role="alert">Account
activation failed! Wrong token.</div>');
                redirect('auth');
            }
        } else {
            $this->session->set_flashdata('message', '
<div class="alert alert-danger" role="alert">Account activation
failed! Wrong email.</div>');
            redirect('auth');
        }
    }

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

```

```

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

        if ($user) {
            $user_token = $this->db->get_where('user_token',
['token' => $token]);->row_array();

            if ($user_token) {
                if (time() - $user_token['date_created'] <
(60 * 60 * 24)) {
                    $this->db->set('is_active', 1);
                    $this->db->where('email', $email);
                    $this->db->update('user');

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

                    $this->session->set_flashdata('message',
'<div class="alert alert-success" role="alert">' . $email . '
has been activated! Please login.</div>');
                    redirect('auth');
                } else {
                    $this->db->delete('user', ['email' =>
$email]);
                    $this->db->delete('user_token', ['email'
=> $email]);

                    $this->session->set_flashdata('message',
'<div class="alert alert-danger" role="alert">Account
activation failed! Token expired.</div>');
                    redirect('auth');
                }
            } else {
                $this->session->set_flashdata('message',
'<div class="alert alert-danger" role="alert">Account
activation failed! Wrong token.</div>');
                redirect('auth');
            }
        } else {
            $this->session->set_flashdata('message', '<div
class="alert alert-danger" role="alert">Account activation
failed! Wrong email.</div>');
            redirect('auth');
        }
    }

    public function logout()
    {
        $this->session->unset_userdata('email');
    }
}

```

```

        $this->session->unset_userdata('role_id');

        $this->session->set_flashdata('message', '<div
class="alert alert-success" role="alert">You have been logged
out!</div>');
        redirect('auth');
    }

    public function blocked()
    {
        $this->load->view('auth/blocked');
    }

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

        if ($this->form_validation->run() == false) {
            $data['title'] = 'Forgot Password';
            $this->load->view('templates/auth_header',
$data);
            $this->load->view('auth/forgot-password');
            $this->load->view('templates/auth_footer');
        } else {
            $email = $this->input->post('email');
            $user = $this->db->get_where('user', ['email' =>
$email, 'is_active' => 1])->row_array();

            if ($user) {
                $token = base64_encode(random_bytes(32));
                $user_token = [
                    'email' => $email,
                    'token' => $token,
                    'date_created' => time()
                ];

                $this->db->insert('user_token', $user_token);
                $this->_sendEmail($token, 'forgot');

                $this->session->set_flashdata('message',
'<div class="alert alert-success" role="alert">Please check
your email to reset your password!</div>');
                redirect('auth/forgotpassword');
            } else {
                $this->session->set_flashdata('message',
'<div class="alert alert-danger" role="alert">Email is not
registered or activated!</div>');
                redirect('auth/forgotpassword');
            }
        }
    }

```

```

    }
}

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

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

    if ($user) {
        $user_token = $this->db->get_where('user_token',
['token' => $token])->row_array();

        if ($user_token) {
            $this->session->set_userdata('reset_email',
$email);

            $this->changePassword();
        } else {
            $this->session->set_flashdata('message',
'<div class="alert alert-danger" role="alert">Reset password
failed! Wrong token.</div>');
            redirect('auth');
        }
    } else {
        $this->session->set_flashdata('message', '
<div class="alert alert-danger" role="alert">Reset password
failed! Wrong email.</div>');
        redirect('auth');
    }
}

public function changePassword()
{
    if (!$this->session->userdata('reset_email')) {
        redirect('auth');
    }

    $this->form_validation->set_rules('password1',
'Password',
'trim|required|min_length[3]|matches[password2]');
    $this->form_validation->set_rules('password2',
'Repeat Password',
'trim|required|min_length[3]|matches[password1]');

    if ($this->form_validation->run() == false) {
        $data['title'] = 'Change Password';
    }
}

```

```

        $this->load->view('templates/auth_header',
$data);
        $this->load->view('auth/change-password');
        $this->load->view('templates/auth_footer');
    } else {
        $password = password_hash($this->input-
>post('password1'), PASSWORD_DEFAULT);
        $email = $this->session->userdata('reset_email');

        $this->db->set('password', $password);
        $this->db->where('email', $email);
        $this->db->update('user');

        $this->session->unset_userdata('reset_email');

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

        $this->session->set_flashdata('message', '<div
class="alert alert-success" role="alert">Password has been
changed! Please login.</div>');
        redirect('auth');
    }
}
}

```

User.php

```

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

class User extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        is_logged_in();
    }

    public function index()
    {
        $data['title'] = 'My Profile';
        $data['user'] = $this->db->get_where('user', ['email'
=> $this->session->userdata('email')])>row_array();

        $this->load->view('templates/header', $data);
        $this->load->view('templates/sidebar', $data);
    }
}

```

```

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

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

        $this->form_validation->set_rules('name', 'Full
Name', 'required|trim');

        if ($this->form_validation->run() == false) {
            $this->load->view('templates/header', $data);
            $this->load->view('templates/sidebar', $data);
            $this->load->view('templates/topbar', $data);
            $this->load->view('user/edit', $data);
            $this->load->view('templates/footer');
        } else {
            $name = $this->input->post('name');
            $email = $this->input->post('email');

            // cek jika ada gambar yang akan diupload
            $upload_image = $_FILES['image']['name'];

            if ($upload_image) {
                $config['allowed_types'] = 'gif|jpg|png';
                $config['max_size']      = '2048';
                $config['upload_path'] =
'./assets/img/profile/';

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

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

            $this->db->set('name', $name);

```

```

        $this->db->where('email', $email);
        $this->db->update('user');

        $this->session->set_flashdata('message', '<div
class="alert alert-success" role="alert">Your profile has
been updated!</div>');
        redirect('user');
    }
}

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

    $this->form_validation->set_rules('current_password',
'Current Password', 'required|trim');
    $this->form_validation->set_rules('new_password1',
'New Password',
'required|trim|min_length[3]|matches[new_password2]');
    $this->form_validation->set_rules('new_password2',
'Confirm New Password',
'required|trim|min_length[3]|matches[new_password1]');

    if ($this->form_validation->run() == false) {
        $this->load->view('templates/header', $data);
        $this->load->view('templates/sidebar', $data);
        $this->load->view('templates/topbar', $data);
        $this->load->view('user/changepassword', $data);
        $this->load->view('templates/footer');
    } else {
        $current_password = $this->input-
>post('current_password');
        $new_password = $this->input-
>post('new_password1');
        if (!password_verify($current_password,
$data['user']['password'])) {
            $this->session->set_flashdata('message',
'<div class="alert alert-danger" role="alert">Wrong current
password!</div>');
            redirect('user/changepassword');
        } else {
            if ($current_password == $new_password) {
                $this->session->set_flashdata('message',
'<div class="alert alert-danger" role="alert">New password
cannot be the same as current password!</div>');
                redirect('user/changepassword');
            } else {
                // password sudah ok

```



```

        $password hash =
password_hash($new_password, PASSWORD_DEFAULT);

        $this->db->set('password',
$password_hash);
        $this->db->where('email', $this->session-
>userdata('email'));
        $this->db->update('user');

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

```

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

```
<!-- Outer Row -->
```

```
<div class="row justify-content-center">
```

```
<div class="col-lg-7">
```

```
<div class="card o-hidden border-0 shadow-lg my-5">
```

```
<div class="card-body p-0">
```

```
<!-- Nested Row within Card Body -->
```

```
<div class="row">
```

```
<div class="col-lg">
```

```
<div class="p-5">
```

```
<div class="text-center">
```

```
<h1 class="h4 text-gray-900 mb-4">Login
```

```
Page</h1>
```

</div>

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

<form class="user" method="post" action="<?= base_url('auth'); ?>">

<div class="form-group">

<input type="text" class="form-control form-control-user" id="email" name="email" placeholder="Masukkan Email" value="<?= set_value('email'); ?>">

<?= form_error('email', '<small class="text-danger pl-3">', '</small>'); ?>

</div>

<div class="form-group">

<input type="password" class="form-control form-control-user" id="password" name="password" placeholder="Masukkan Password">

<?= form_error('password', '<small class="text-danger pl-3">', '</small>'); ?>

</div>

<button type="submit" class="btn btn-primary btn-user btn-block">

Login

</button>

</form>

<hr>

<div class="text-center">

<a class="small" href="<?= base_url('auth/forgotpassword'); ?>">lupa password?

```

        </div>

        <div class="text-center">

            <a class="small" href="<?=
base_url('auth/registration'); ?>">buat account!</a>

        </div>

    </div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

```

13. Index_admin.php

```

<!DOCTYPE html>
<!--[if lt IE 7]>          <html class="no-js lt-ie9 lt-ie8 lt-
ie7"> <![endif]-->
<!--[if IE 7]>            <html class="no-js lt-ie9 lt-ie8">
<![endif]-->
<!--[if IE 8]>            <html class="no-js lt-ie9"> <![endif]-
->
<!--[if gt IE 8]><!-->
<html class="no-js"> <!--<![endif]-->
<head>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
    <meta name="description" content="Cahaya Jaya Poltekpos">

```

```

<meta name="author" content="Themefisher.com">

<title>Cahaya Jaya</title>

<!-- Mobile Specific Meta-->
<meta name="viewport" content="width=device-width, initial-
scale=1">

<!-- Favicon -->
<link rel="shortcut icon" type="image/x-icon" href="<?php
echo base_url('assets/images/favicon.png')?>" />

<!-- Themefisher Icon font -->
<link rel="stylesheet" href="<?php echo
base_url('assets/plugins/themefisher-font/style.css')?>">
<!-- bootstrap.min css -->
<link rel="stylesheet" href="<?php echo
base_url('assets/plugins/bootstrap/css/bootstrap.min.css')?>"
>

<link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/plugins/slick-
carousel/slick/slick.css')?>" />
<link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/plugins/slick-carousel/slick/slick-
theme.css')?>" />

<!-- Main Stylesheet -->
<link rel="stylesheet" href="<?php echo
base_url('assets/css/style.css')?>">

</head>

<body id="body">

<!-- Start Top Header Bar -->
<section class="top-header">
<div class="container">
<div class="row">
<div class="col-md-4 col-xs-12 col-sm-4">
<div class="contact-number">
<i class="tf-ion-ios-telephone"></i>
<span>0129- 12323-123123</span>
</div>
</div>
<div class="col-md-4 col-xs-12 col-sm-4">
<!-- Site Logo -->
<div class="logo text-center">
<a href="index.html">
<!-- replace logo here -->

```

```

        <svg width="270px" height="29px" viewBox="0 0 155
29" version="1.1" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">
        <g id="Page-1" stroke="none" stroke-width="1"
fill="none" fill-rule="evenodd" font-size="30" font-
family="AustinBold, Austin" font-weight="bold">
            <g id="Group" transform="translate(-
108.000000, -297.000000)" fill="#000000">
                <text id="AVIATO">
                    <tspan x="80.94" y="325">CAHAYA
JAYA</tspan>
                </text>
            </g>
        </g>
    </svg>
</a>
</div>
</div>
<div class="col-md-4 col-xs-12 col-sm-4">
<!-- Cart -->
<ul class="top-menu text-right list-inline">

    <li>
        <a href="<?php echo base_url().
'buyer/home_buyer/logout'; ?>"><i class="tf-ion-log-out"></i>
Logout</a>
    </li>

</ul><!-- / .nav .navbar-nav .navbar-right -->
</div>
</div>
</div>
</section><!-- End Top Header Bar -->

<!-- Main Menu Section -->
<section class="menu">
    <nav class="navbar navigation">
        <div class="container">
            <div class="navbar-header">
                <h2 class="menu-title">Main Menu</h2>
                <button type="button" class="navbar-toggle
collapsed" data-toggle="collapse" data-target="#navbar" aria-
expanded="false" aria-controls="navbar">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>

            </div><!-- / .navbar-header -->

```

```

        <!-- Navbar Links -->
        <div id="navbar" class="navbar-collapse collapse
text-center">
            <ul class="nav navbar-nav">

                <li class="dropdown dropdown-slide">
                    <a class="dropdown-toggle" data-
toggle="dropdown" data-hover="dropdown" data-delay="350"
aria-haspopup="true" aria-expanded="false"><i class="tf-ion-
home"></i>&nbsp;Home<span class="tf-ion-ios-arrow-
down"></span></a>
                    <ul class="dropdown-menu">
                        <li><a href="<?php echo base_url(). 'home_admin';
?>">Progress</a></li>
                        <li><a href="<?php echo base_url().
'home_admin/index2'; ?>">Clear</a></li>
                    </ul>
                </li>

                <!-- Elements -->
                <li>
                    <a href="<?php echo base_url().
'home_admin/category'; ?>"><i class="tf-ion-
archive"></i>&nbsp;Categories</a>
                </li><!-- / Elements -->
                <!-- Elements -->

            </ul><!-- / .nav .navbar-nav
        </div><!-- /.navbar-collapse -->
    </div><!-- / .container -->
</nav>
</section>

<section class="products section bg-gray">
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <div class="title text-center">
                    <h2><i class="tf-ion-ios-
briefcase"></i>&nbsp;Shopping History Customer</h2>
                </div>
            </div>
            <table class="table">
                <thead>
                    <tr>
                        <th width="3%" scope="col">No.</th>
                        <th width="3%" scope="col">Invoice</th>
                        <th width="15%" scope="col">Payment Status</th>
                        <th width="15%" scope="col">Name Product</th>

```

```

        <th width="15%" scope="col">Order Date</th>
        <th width="15%" scope="col">Seller</th>
        <th width="10%" scope="col">Total</th>
        <th width="13%" scope="col">Shipping Status</th>
        <th width="13%" scope="col">Resi</th>
        <th width="13%" scope="col">Action</th>
    </tr>
</thead>
<?php
$i=0;
foreach ($order as $key) {
    $i++;
    ?>
<tbody>
    <tr>
        <th scope="row"><?php echo $i ?></th>
        <td><?php echo $key['invoice_id'] ?></td>
        <td><?php echo $key['payment_status'] ?></td>
        <td><?php echo $key['product_name'] ?></td>
        <td><?php echo $key['date'] ?></td>
        <td><?php echo $key['name'] ?></td>
        <td><b><?php echo 'Rp.
'.number_format($key['price'],0,',','.'); ?></b></td>
        <td><?php echo $key['shipping_status'] ?> <br>
        <td><?php echo $key['resi'] ?> <br>
        <td>
            <form action="<?php echo
base_url() ?>home_admin/clear_payment" method="post">
                <?php if ($key['payment_status'] == 'Paid') { ?>
                    <?php if ($key['shipping_status'] == 'Already
Received') { ?>
                        <button type="submit" class ='btn btn-sm btn-
primary' disabled>Clear</a>
                        <?php }elseif ($key['shipping_status'] == 'On
Delivery') { ?>
                            <input type="hidden" name="price" value="<?php echo
$key['price'] ?>"></input>
                            <input type="hidden" name="account_id" value="<?php
echo $key['account_id'] ?>"></input>
                            <input type="hidden" name="order_id" value="<?php
echo $key['order_id'] ?>"></input>
                            <button type="submit" onclick="return
validasi();" class ='btn btn-sm btn-primary'>Clear</a>
                            <?php }elseif ($key['shipping_status'] == 'On
Progress') { ?>
                                <button type="submit" class ='btn btn-sm btn-
primary' disabled>Clear</a>
                                <?php } ?>
                            <?php }elseif ($key['payment_status'] == 'In Progress')
{ ?>

```

```

        <?php if ($key['shipping_status'] == 'Already
Received') { ?>
            <button type="submit" class ='btn btn-sm btn-
primary' disabled>Clear</a>
            <?php }elseif ($key['shipping_status'] == 'On
Progress') { ?>
                <input type="hidden" name="price" value="<?php echo
$key['price'] ?>"></input>
                <input type="hidden" name="account_id" value="<?php
echo $key['account_id'] ?>"></input>
                <input type="hidden" name="order_id" value="<?php
echo $key['order_id'] ?>"></input>
                <button type="submit" class ='btn btn-sm btn-
primary' disabled>Clear</a>
            <?php } ?>
        <?php } ?>
    </form>
</td>
</tr>
</form>
</tbody>
<?php } ?>
</table>

```

```

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

```

```

<!--
=====
=====
> -->

```

```

<!--
Start Call To Action
===== -->
<section class="section instagram-feed">
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                </div>
            </div>
        </div>
    </div>
</section>

<footer class="footer section text-center">
    <div class="container">
        <div class="row">

```



```

<div class="col-md-12">
  <ul class="social-media">
    <li>
      <a href="">
        <i class="tf-ion-social-facebook"></i>
      </a>
    </li>
    <li>
      <a href="">
        <i class="tf-ion-social-instagram"></i>
      </a>
    </li>
    <li>
      <a href="">
        <i class="tf-ion-social-twitter"></i>
      </a>
    </li>
    <li>
      <a href="">
        <i class="tf-ion-social-pinterest"></i>
      </a>
    </li>
  </ul>
  <ul class="footer-menu">
    <li>
      <a href="">CONTACT</a>
    </li>
    <li>
      <a href="">SHIPPING</a>
    </li>
    <li>
      <a href="">TERMS OF SERVICE</a>
    </li>
    <li>
      <a href="">PRIVACY POLICY</a>
    </li>
  </ul>
  <p class="copyright-text">Powered by Bootstrap</p>
</div>
</div>
</footer>

```

```

<!--

```

```

Essential Scripts

```

```

=====-->

```

```

<!-- Main jQuery -->

```

```

    <script src="https://code.jquery.com/jquery-
git.min.js"></script>
    <!-- Bootstrap 3.1 -->
    <script src="<?php echo
base_url('assets/plugins/bootstrap/js/bootstrap.min.js')?>"><
/script>
    <!-- Instagram Feed Js -->
    <script src="<?php echo
base_url('assets/plugins/instafeed.js/instafeed.min.js')?>"><
/script>
    <!-- Slick Carousel -->
    <script src="<?php echo base_url('assets/plugins/slick-
carousel/slick/slick.min.js')?>"></script>
    <!-- Google Map js -->
    <script async defer
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBITrd4
sQ_aXlQG_fvEzsvuYyaWnJKETk&callback=initMap"></script>

    <script type="text/javascript" src="<?php echo
base_url('assets/plugins/bootstrap/jquery/dist/jquery.js')?>"
></script>
    <script type="text/javascript" src="<?php echo
base_url('assets/plugins/bootstrap/js/bootstrap.js')?>"></scr
ipt>

    <!-- Main Js File -->
    <script src="<?php echo
base_url('assets/js/script.js')?>"></script>

    <script type="text/javascript">
        function validasi() {
            var msg= confirm("Apakah Anda Yakin Ingin
Mengubah Data Ini?");
            if (msg){
                return true ;
            }else{
                return false ;
            }
        }
    </script>

</body>
</html>

```

14.

```
<section class="page-header">
```

```

<div class="container">
  <div class="row">
    <div class="col-md-12">
      <div class="content">
        <h1 class="page-name">Account</h1>
      </div>
    </div>
  </div>
</div>
</div>
</section>

```

```

<section class="page-wrapper bg-gray">
  <div class="contact-section">
    <div class="container">
      <div class="row">

        <!-- Contact Details -->

        <?php foreach ($account as $key) { ?>
          <div class="contact-details col-md-6 " >
            <div class="google-map">
              
            </div>

          </div>

          <div class="contact-details col-md-6 " >
            <ul class="contact-short-info" >
              <li>
                <i class="tf-ion-edit" style="font-size:
30px"></i>
                <span style="font-size: 20px"> Name: <?php
echo $key->name ?></span>
              </li>
              <li>
                <i class="tf-ion-person" style="font-size:
30px"></i>
                <span style="font-size: 20px"> Username: <?php
echo $key->username ?></span>
              </li>
              <li>
                <i class="tf-ion-key" style="font-size:
30px"></i>
                <span style="font-size: 20px"> Password:
*****</span>
              </li>
            </ul>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        <li>
            <i class="tf-ion-android-mail" style="font-size: 30px"></i>
            <span style="font-size: 20px"> Email: <?php
echo $key->email ?></span>
        </li>
        <li>
            <i class="tf-ion-ios-home" style="font-size: 30px"></i>
            <span style="font-size: 20px"> Address: <?php
echo $key->address ?></span>
        </li>
        <li>
            <i class="tf-ion-map" style="font-size: 30px"></i>
            <span style="font-size: 20px"> City: <?php
echo $key->city ?></span>
        </li>
        <li>
            <i class="tf-ion-android-phone-portrait" style="font-size: 30px"></i>
            <span style="font-size: 20px"> Phone: <?php
echo $key->telp ?></span>
        </li>
        <li>
            <i class="tf-ion-card" style="font-size: 30px"></i>
            <span style="font-size: 20px"> Rekening: <?php
echo $key->rekening ?></span>
        </li>
        <li>
            <i class="tf-ion-cash" style="font-size: 30px"></i>
            <span style="font-size: 20px"><?= 'Rp.
'.number_format($key->saldo,0,',','.'.''); ?></span>
        </li>
        <br>
        <li>
            <!-- <span data-toggle="modal" data-target="#product-modal<?php echo $key->account_id?>"
class="btn btn-main">Edit Profile</span> -->
            <button type="button" class='btn btn-main' data-toggle='modal' data-target='#product-modal<?= $key->account_id; ?>'>Edit Profile</button>
        </li>
    </ul>

```

```

        </div>
        <!-- Modal -->
        <div class="modal fade" id="product-modal">?php echo
$key->account_id;?>
        <button type="button" class="close" data-
dismiss="modal" aria-label="Close">
            <i class="tf-ion-close"></i>
        </button>
        <div class="modal-dialog " role="document">
            <div class="modal-content">

                <div class="modal-header">
                    <h5 class="modal-title">Edit
Profile</h5>

                    <button type="button"
class="close" data-dismiss="modal" aria-label="Close">
                        <span aria-
hidden="true">&times;</span>
                    </button>
                </div>
                <div class="modal-body">

                    <form action="<?php echo base_url().
'buyer/home_buyer/update_account'; ?>" method="post"
enctype="multipart/form-data">

                        <input style="font-size: 15px"
style="font-size: 15px" type="hidden" class="form-control"
name="id" value="<?php echo $key->account_id?>">
                        <input style="font-size: 15px"
style="font-size: 15px" type="hidden" class="form-control"
name="image2" value="<?php echo $key->image?>">

                        <div class="row">
                            <div class="col-md-6">
                                <div class="form-group">
                                    <span>Nama</span>
                                    <input style="font-size: 15px"
style="font-size: 15px" type="text" class="form-control"
name="name" value="<?php echo $key->name?>">
                                </div>
                            </div>

                            <div class="col-md-6">
                                <div class="form-group">
                                    <span>Foto</span>
                                    <input type="file" class="form-control"
name="image">
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

        <div class="row">
            <div class="col-md-6">
                <div class="form-group">
                    <span>Username</span>
                    <input style="font-size: 15px"
type="text" class="form-control" name="username" value="<?php
echo $key->username?>">
                </div>
            </div>

            <div class="col-md-6">
                <div class="form-group">
                    <span>Password</span>
                    <input style="font-size: 15px"
type="text" class="form-control" name="password" value="<?php
echo $key->password?>">
                </div>
            </div>
        </div>

        <div class="row">
            <div class="col-md-6">
                <div class="form-group">
                    <span>Email</span>
                    <input style="font-size: 15px"
type="text" class="form-control" name="email" value="<?php
echo $key->email?>">
                </div>
            </div>

            <div class="col-md-6">
                <div class="form-group">
                    <span>No. Telp</span>
                    <input style="font-size: 15px"
type="text" class="form-control" name="telp" value="<?php
echo $key->telp?>">
                </div>
            </div>
        </div>

        <div class="row">
            <div class="col-md-12">
                <div class="form-group">
                    <span>Alamat</span>
                    <textarea
name="address" value="<?=$key->address;?>" class="form-
control border-input" required cols="30" rows="5"><?php echo
$key->address?></textarea>
                </div>
            </div>
        </div>
    </div>

```

```

        <div class="row">
            <div class="col-md-6">
                <div class="form-group">
                    <span>Kota</span>
                    <input style="font-size: 15px"
type="text" class="form-control" name="city" value="<?php
echo $key->city?>">
                </div>
            </div>

            <div class="col-md-6">
                <div class="form-group">
                    <span>Rekening</span>
                    <input style="font-size: 15px"
type="text" class="form-control" name="rekening" value="<?php
echo $key->rekening?>">
                </div>
            </div>
        </div>
        <button type="submit" class="btn btn-
main">Edit</button>
        <button type="button" class="btn btn-
secondary" data-dismiss="modal">Close</button>
    </form>
</div>
</div>
</div>
</div>
</div>

<?php } ?>

```

```

    </div> <!-- end row -->
</div> <!-- end container -->
</div>
</section>

```

15. Cart.php

```

<section class="products section bg-gray">
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <div class="title text-center">
                    <h2>Shopping Cart</h2>
                </div>
            </div>
        </div>
    </div>
</section>

```

```

        </div>
    </div>
<table class="table">
    <thead>
        <tr>

            <th width="2%" scope="col">No.</th>
            <th width="10%" scope="col">Image</th>
            <th width="33%" scope="col">Name Product</th>
            <th width="17%" scope="col">Seller</th>
            <th width="8%" scope="col">Quantity</th>
            <th width="15%" scope="col">Price */stock</th>
            <th width="15%" scope="col">Total</th>
            <th width="10%" scope="col">Remove</th>
        </tr>
    </thead>
    <tbody>
        <?php
            $i=0;
            $total=0;
            // $this-> cart contents[$rowid]['rowid'] = $rowid;
            foreach ($this->cart->contents() as $items):
                $subtotal = ($items['qty'] * $items['price']);
                // $grand_total=$total + $items['']
                $i++;
            ?>
        <tr>
            <th scope="row"><?php echo $i?></th>
            <td></td>
            <td hidden><?php echo $items['product_id'] ?></td>
            <td><?php echo $items['name'] ?></td>
            <td><?php echo $items['nama'] ?></td>
            <td><?php echo $items['qty'] ?></td>
            <td><?php echo 'Rp.
'.number_format($items['price'],0,',','.') ?></td>

            <td><?php echo 'Rp.
'.number_format($subtotal,0,',','.') ?></td>

            <td>
                <center>
                    <a href="<?php echo
base_url('buyer/home_buyer/remove_item/' . $items['rowid']);
?>" class="remove"><i class="tf-ion-close"></i></a>
                    <?php endforeach; ?>
                </center>
            </td>
        </tr>
    </tbody>

```



```

        <tr>
            <td colspan="3"><b>Order Total: <?= 'Rp.
'.number_format($this->cart->total(),0,',','.''); ?></b></td>
            <td colspan="4" align="right">
                <a href="<?php echo
base_url()>buyer/home_buyer/clear_cart" class='btn btn-sm
btn-danger'>Empty The Cart</a>
                <a href="<?php echo base_url()>buyer/home_buyer/shop"
class='btn btn-sm btn-success' type="submit">Update Cart</a>

                <a href="<?php echo base_url()>buyer/home_buyer/order"
class='btn btn-sm btn-primary'>Check Out</a>
            </td>
        </tr>
    </table>

```

```

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

```