

Belajar Dengan Peraktek

[Date]

Codelgniter :

Implementasi Metode Entropy
Pada Pemerograman PHP

Cokro Edi Prawiro
POLITEKNIK POS INDONESIA

BAB I

CodeIgniter

Pada bab ini akan membahas mengenai sejarah dari codeigniter, keunggulan dari codeigniter, sarana atau dasar untuk memperajari codeigniter setelah itu dilanjutkan dengan tools pendukung untuk codeigniter sekilas tentang MVC pada codeigniter kemudian struktur direktori pada paket yang telah di sediakan oleh codeigniter kemudian editor teks yang di anjurkan dan contoh MVC sederhana

Sejarah CodeIgniter

Codeigniter merupakan *framework* web yang digunakan untuk bahasa pemerograman PHP yang dibuat oleh Rick Ellis pada tahun 2006, penemu dan pendiri EllisLab (www.ellislab.com). Ellislab merupakan suatu tim kerja yang berdiri pada tahun 2002 dan bergerak di bidang pembuatan *software* dan *tool* untuk para pengembang web. Sejak tahun 2014 sampai sekarang, EllisLab telah menyerahkan hak kepemilikan codeigniter ke British Columbia Institute of Technology (BCIT) untuk proses pengembangan lebih lanjut. Saat ini situs resmi codeigniter telah berubah dari www.ellislab.com ke www.codeigniter.com.

Dibandingkan web *framework* yang lain codeIgniter memiliki desain yang lebih sederhana dan bersifat tidak kaku (fleksibel). CodeIgniter masih mengizinkan atau memberikan kebebasan kepada para pengembang untuk menulis code-code tertentu di dalam aplikasi menggunakan cara konvesional atau tanpa menggunakan framework.

Beberapa Keunggulan CodeIgniter

CodeIgniter merupakan *toolkit* untuk orang-orang yang ingin membuat atau membangun aplikasi web menggunakan bahasa pemerograman PHP, adapun beberapa keunggulan yang ditawarkan oleh CodeIgniter adalah sebagai berikut:

1. CodeIgniter merupakan framework yang bersifat gratis atau *open-source*
2. CodeIgniter memiliki ukuran file yang relatif kecil dibandingkan *Framework* php lain. Setelah di download dan di ekstrak file codeigniter memiliki total ukuran kurang lebih 11 MB dengan

ketentuan folder user_guide (dokumentasi CodeIgniter) kurang lebih sebesar 9 MB dan folder aplikasi dan sistem dengan ukurang kurang lebih 2 MB

3. Aplikasi yang dibuat menggunakan codeigniter dapat berjalan dengan cepat
4. CodeIgniter Menggunakan pola desain Model-View-Controller (MVC) yang memungkinkan pada satu file tidak akan berisi banyak *code*. Halini mengakibatkan kode menjadi mudah untuk di baca, dipahami dan dikembangkan atau dilakukan *maintaining* (pemeriharaan) di kemudian hari.
5. Codeigniter dapat diperluas sesuai dengan kebutuhan.
6. Codeigniter juga terdokumentasi dengan baik atau memiliki dokumentasi yang sangat baik. Informasi tentang *class* dan *function* yang terdapat pada codeigniter dapat diperoleh melalui dokumentasi yang disediakan pada paket distribusinya.

Persiapan Untuk Menggunakan CodeIgniter

CodeIgniter merupakan *framework* PHP. Untuk dapat menggunakan terlebih dahulu programmer harus sudah familiar dengan penggunaan bahasa pemerograman PHP atau sudah mahir menggunakan bahasa pemerograman PHP. Selain itu, CodeIgniter merupakan framework yang memiliki konsep MVC maka pada saat melakukan pemerograman menggunakan framework CodeIgniter pasti bersinggungan dengan *model*, *view* dan *controller* dimana isi dari *model* dan *controller* merupakan *class* yang merupakan inti dari pemerograman yang berorientasi objek. Maka dari itu untuk menggunakan codeigniter harus mengetahui konsep pemerograman berorientasi objek menggunakan PHP.

Tools yang di Perlukan

Untuk dapat menggunakan codeigniter setidaknya harus memiliki beberapa tools yang harus di install pada komputer. Adapun tolls yang harus di gunakan yaitu :

1. PHP

PHP merupakan base bahasa pemrograman yang di gunakan oleh CodeIgniter sehingga PHP harus terinstall terlebih dahulu pada komputer yang akan di gunakan untuk pemrograman sehingga dapat CodeIgniter dapat digunakan

2. Web Server

Web server atau server web merupakan tool yang digunakan untuk mengeksekusi code PHP sehingga hasil dari code PHP tersebut dapat terlihat, sedangkan web server yang di gunakan merupakan apache. Merupakan web server lokal yang harus di istall pada komputer yang akan digunakan untuk pemrograman, jika apache telah terinstall dapat di coba di panggil menggunakan localhost.

3. Server Database

Setelah web server swlanjutnya yaitu Server database yang merupakan database atau pusat penyimpanan data dari web yang di buat, biasanya server database yang sering di gunakan merupakan database MySql, selain dari database MySql juga dapat di gunakan yang penting database tersebut termasuk pada jenis database yang dapat di hubungkan melalui Open Database Connectivity (ODBC).

Untuk opsi lain dalam mengistall ke tiga tools tersebut dapat menggunakan XAMPP, XAMPP merupakan aplikasi yang di dalamnya

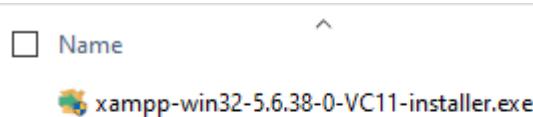
terdapat dari kumpulan aplikasi yang digunakan untuk pengembangan dan pembuatan website berupa Apache, MySQL, PHP, dan Perl, dengan menggunakan XAMPP dapat mempersingkat pekerjaan yang tadinya harus menginstall PHP, Apache, dan MySQL secara terpisah menjadi satu, hanya dengan menggunakan satu aplikasi XAMPP semua aplikasi tersebut telah teristall.

Catatan:

Untuk xampp dari versi 5 sampai 7 sekarang sudah tidak menggunakan database MySQL melainkan menggunakan database MariaDB, namun tidak perlu kawathir dikarenakan MariaDB basenya masih Menggunakan MySQL.

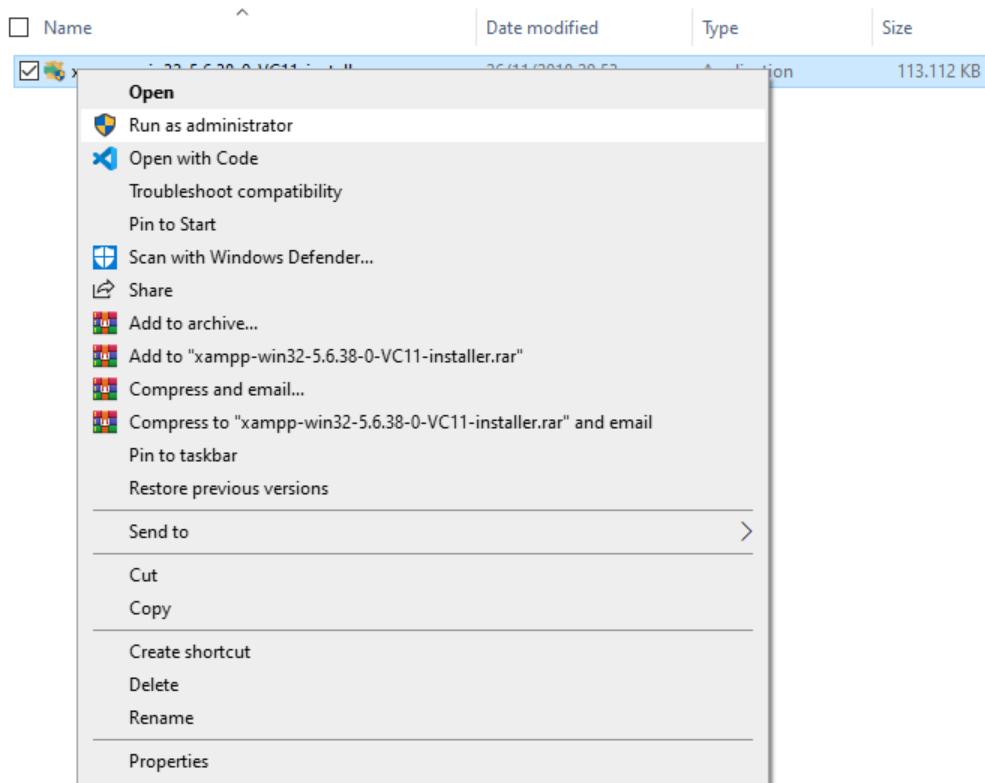
Berikut merupakan langkah langkah beserta screenshot untuk instalasi XAMPP.

1. Download terlebih dahulu file XAMPP pada website resminya pada link berikut <https://www.apachefriends.org/download.html>
2. setelah terdownload filenya seperti pada gambar 1 berikut



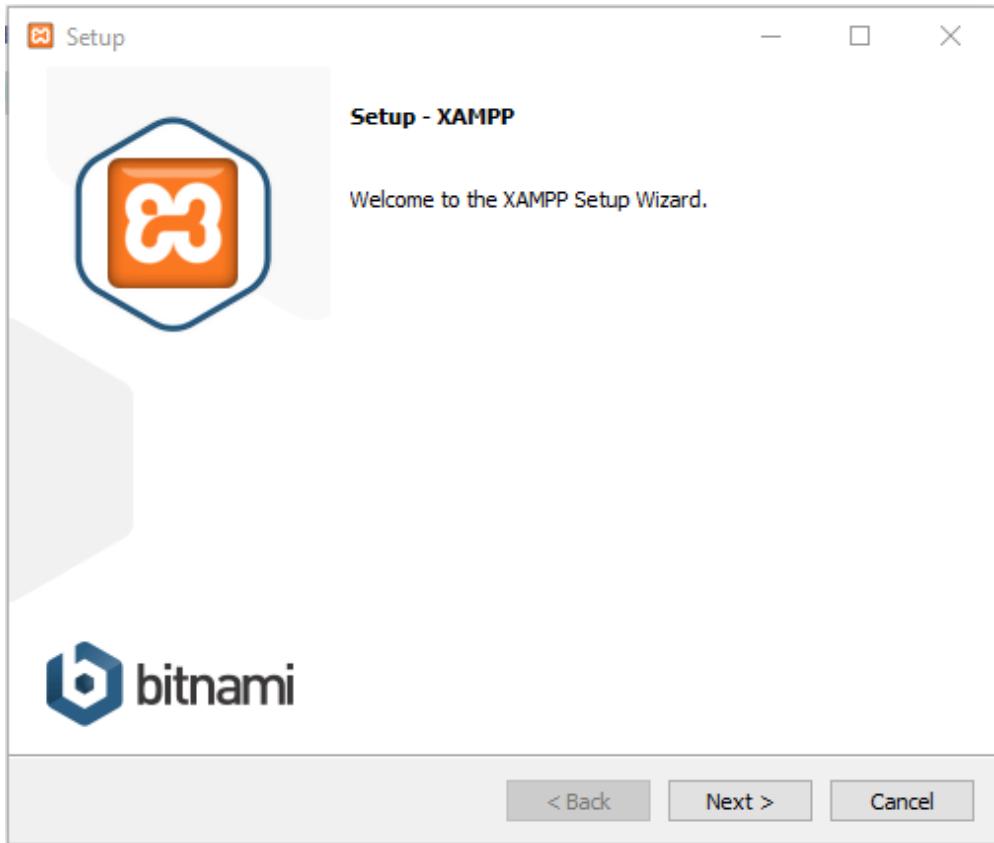
Gambar 1 file xampp telah di download

3. Kemudian setelah itu jalankan file tersebut dengan cara klik kanan pilih run administrator seperti pada gambar 2 berikut :



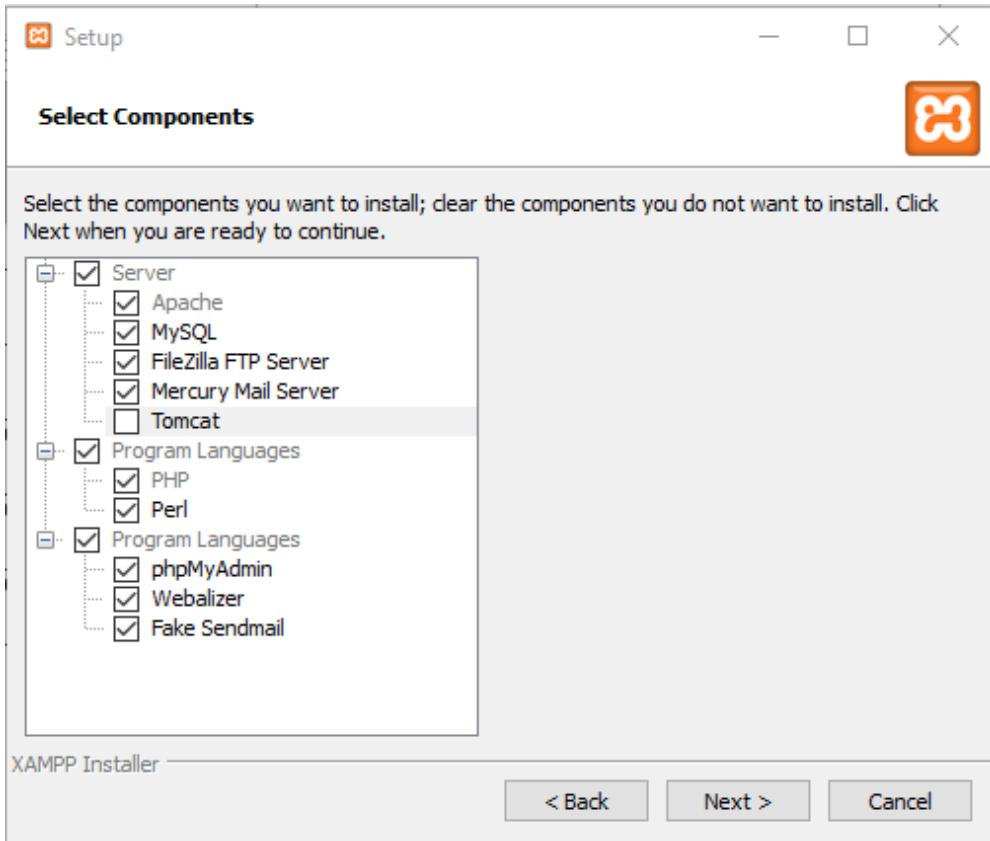
Gambar 2 run as administrator XAMPP

4. Kemudian jika muncul popup pilihan untuk memasang aplikasi pada komputer pilih yes, lalu tunggu beberapa saat maka akan muncul setup XAMPP seperti pada gambar 3.



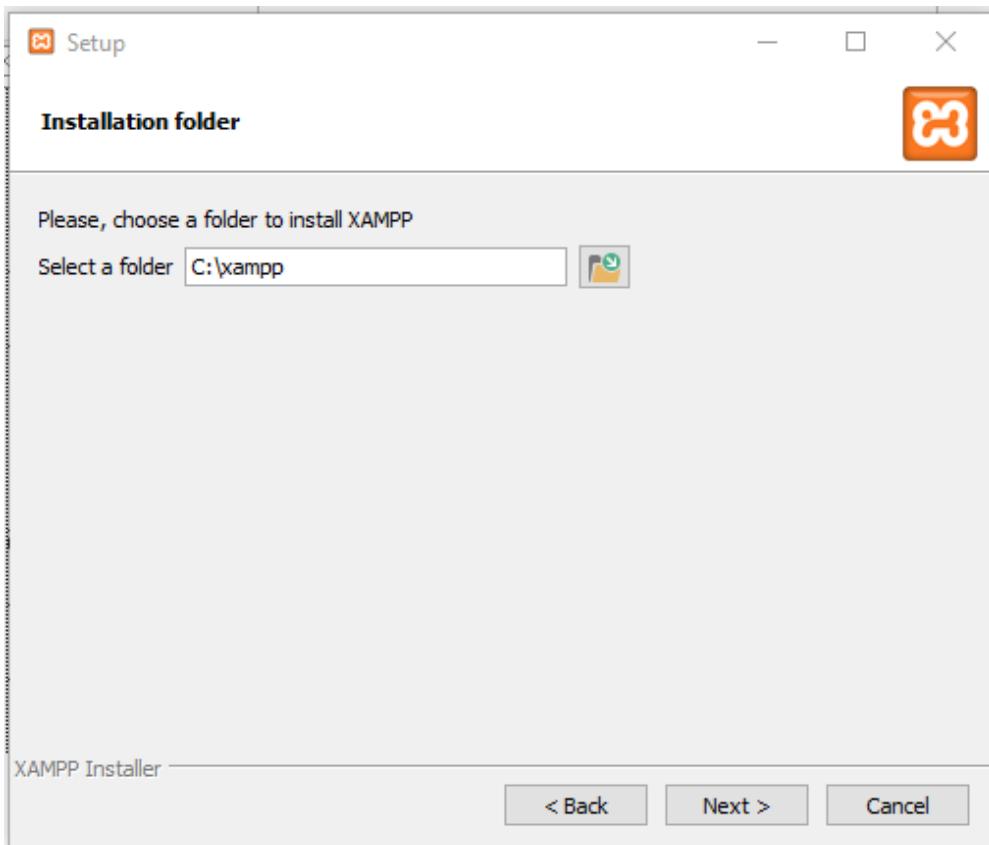
Gambar 3 Setup XAMPP

5. Kemudian klik **Next** untuk melanjutkan peroses Instalasi



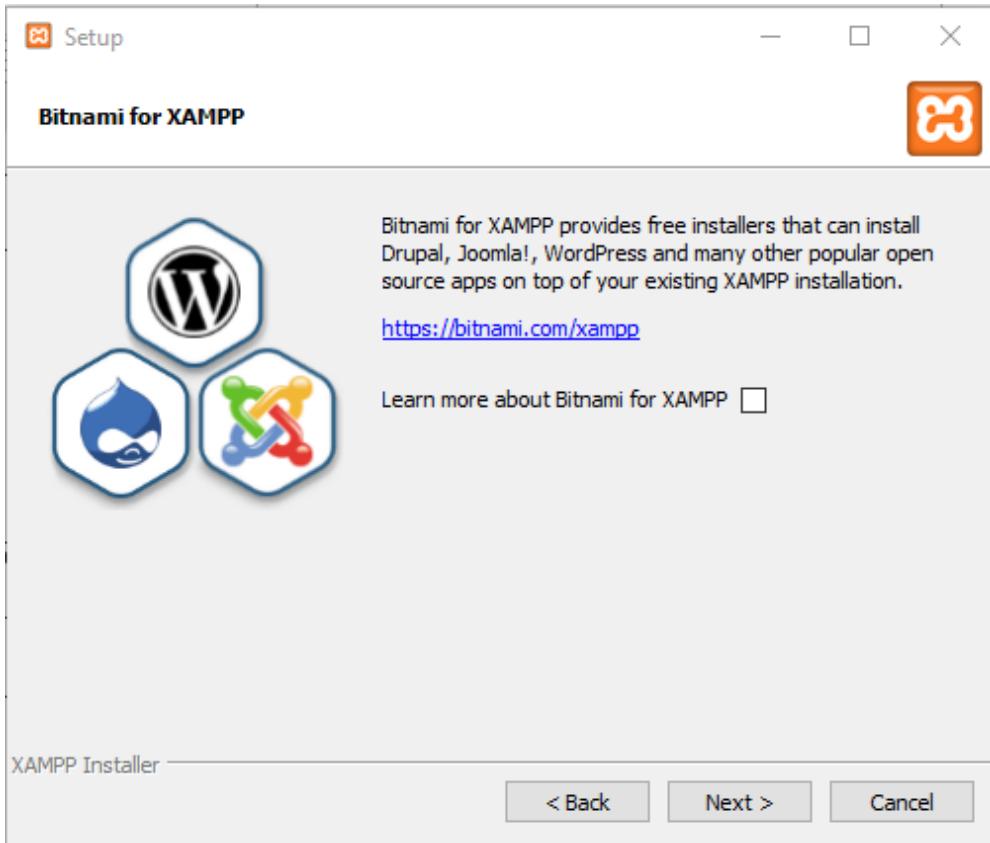
Gambar 4 Memilih Software

6. Pada gambar 4 merupakan peroses memilih software yang akan di pasang pada komputer sebagai contoh hilangkan tanda checklist pada check box Tomcat, kemudian kelik tombol **Next**.



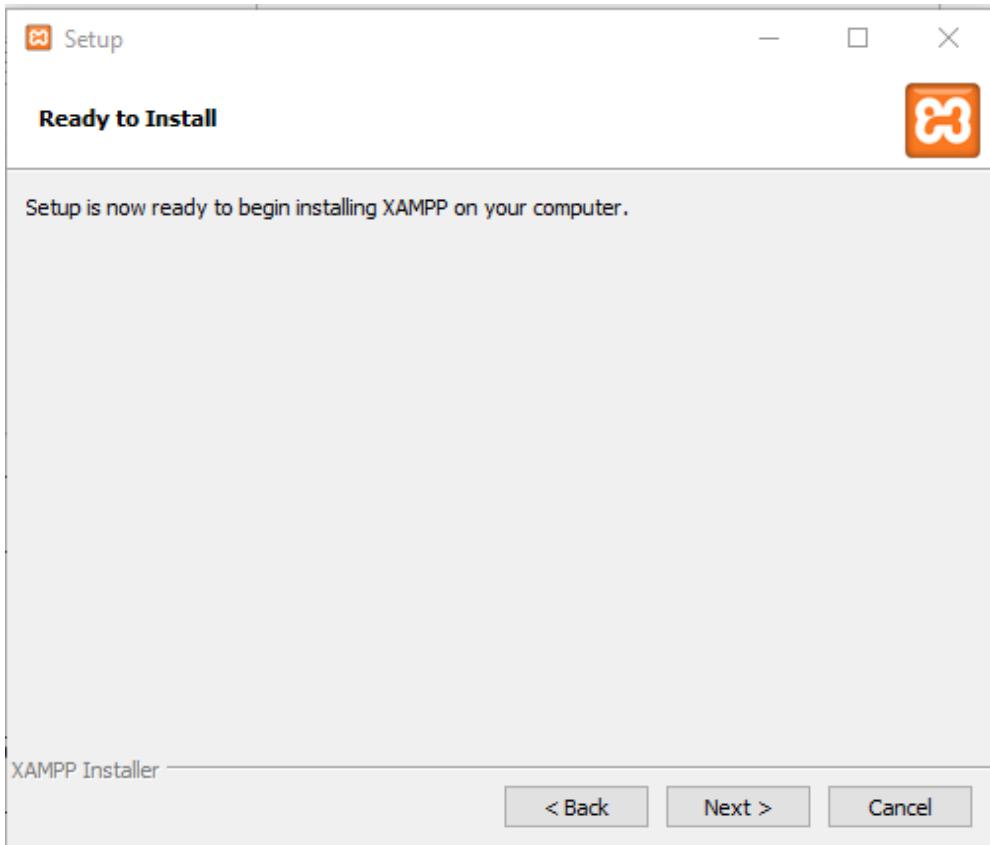
Gambar 5 Menentukan tempat instal xampp

7. Pada gambar 5 merupakan menentukan tujuan instalasi XAMPP, secara default xampp akan teristal pada direktori C:\xampp, jika tidak akan menginstall di direktori C maka datapat memilih direktori lain dengan cara klik tmbol browser yang bergambar folder dengan anak panah. Kemudian klik tombol **Next** untuk melanjutkan peroses.



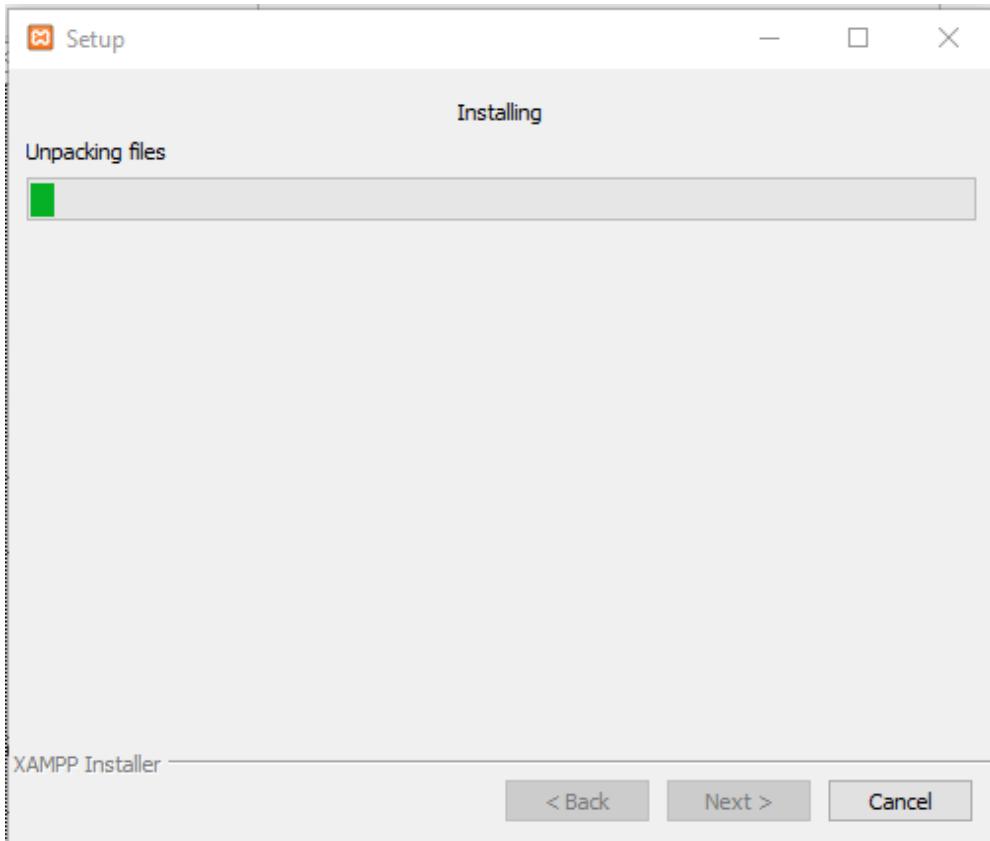
Gambar 6 Bitami For XAMPP

8. Pada gambar 6 uncheck checklist yang terdapat pada halaman tersebut, kemudian klik tombol **Next** untuk melanjutkan peroses instalasi.



Gambar 7 semua aplikasi siap diinstall

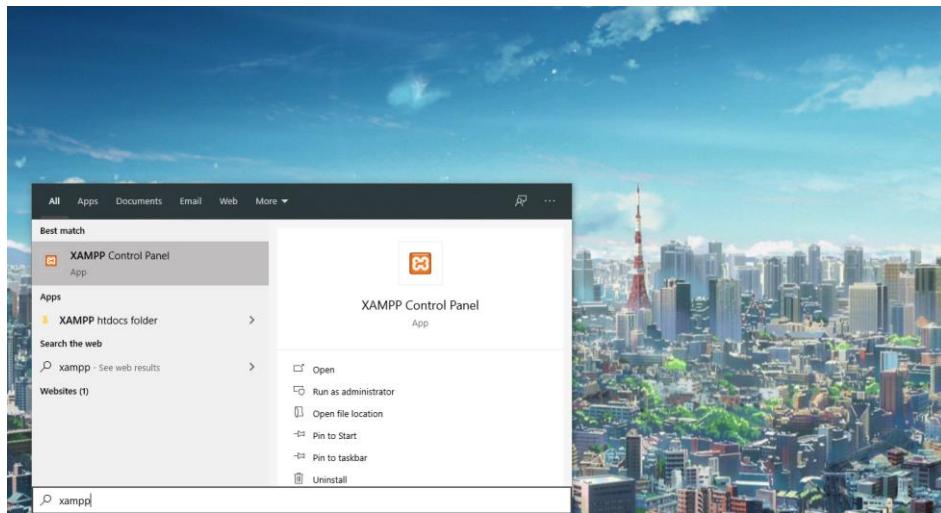
9. Pada gambar 7 menunjukan bahwa semua aplikasi yangtelah di cheklis tadi dan telah di tentukan tempat installnya, telah siap untuk diinstal, kemudian klik tombol **Next** untuk melanjutkan proses instal



Gambar 8 Proses Install

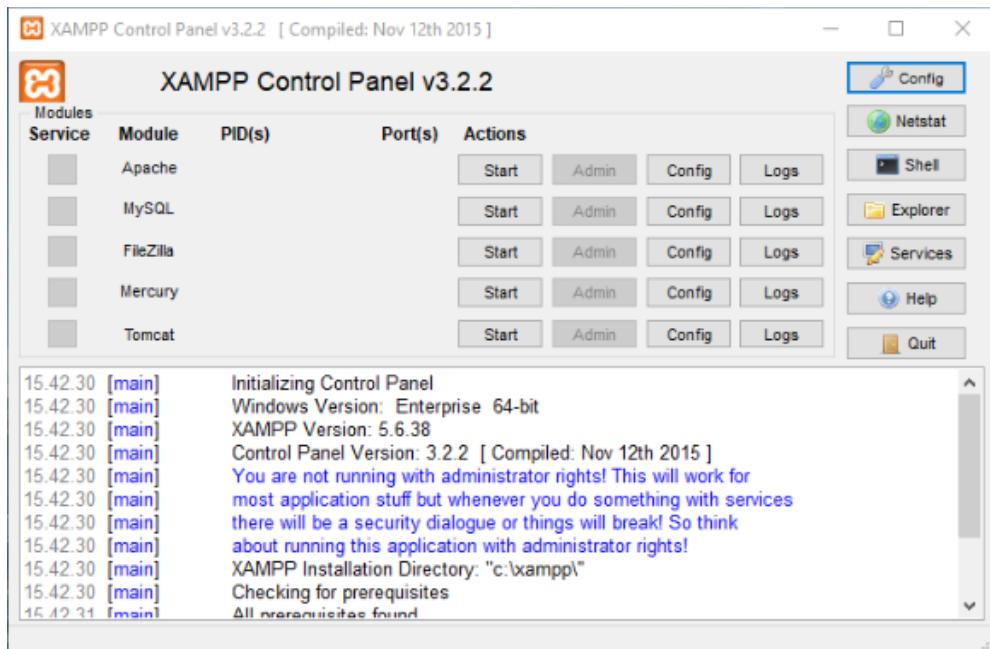
10. Pada gambar 8 menunjukan peroses install aplikasi pada peroses ini tunggu instal aplikasinya beres jika muncul popup klik finish untuk mengakhiri proses istalasi

Setelah peroses instalasi selesai jalankan xampp controlpanel dengan cara klik icon search pada taskbar di windows kemudian search xampp maka hasilnya seperti pada gambar 8 berikut.



Gambar 8 search XAMPP

Pada gambar 8 tersebut kemudian klik tombol open untuk memunculkan XAMPP control panel



Gambar 9 XAMPP control Panel

Pada gambar 9 tersebut jalankan service apache dan MySQL dengan cara klik tombol start yang terdapat di sebelah tulisan apache dan MySQL.

Catatan:

Untuk melatekan dokumen codeigniter pada xampp dapat di simpan pada dokumen root apache yang terletak pada C:\xampp\htdocs.

Instalasi CodeIgniter

Framework code igniter dapat di unduh website resminya yaitu www.codeigniter.com untuk tampilannya seperti pada gambar 10 berikut



Gambar 10 website codeigniter

Pada buku ini akan menggunakan codeigniter versi 3.1.11 yang terbaru pada saat buku ini di tulis. Untuk dapat mengunduhnya dapat menekan menu download yang terdapat pada halaman utama web resmi codeigniter atau dengan cara menekan menu download yang terdapat pada navigator bar maka akan pindah halaman ke halaman download pada halaman tersebut pilih menu codeigniter 3 dan download seperti pada gambar 11 berikut

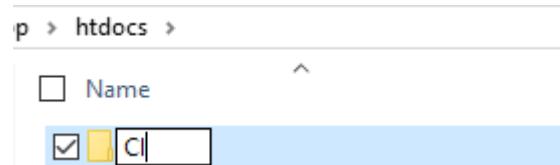


Gambar 11 halaman Download

Setelah di download maka hasil file nya berupa zip dapat dilihat pada gambar 12, setelah itu ekstrak file zip tersebut kemudian pindahkan ke direktori C:\xampp\htdocs, lalu agar mempermudah pemanggilan terhadap folder codeigniter bisa dilakukan dengan cara me rename folder codeigniter sebut misalkan menjadi CI sehingga hasilnya seperti pada gambar 13.

Name	Date modified	Type	Size
CodeIgniter-3.1.11.zip	26/01/2020 17.33	WinRAR ZIP archive	2.685 KB

Gambar 12 hasil download codeigniter

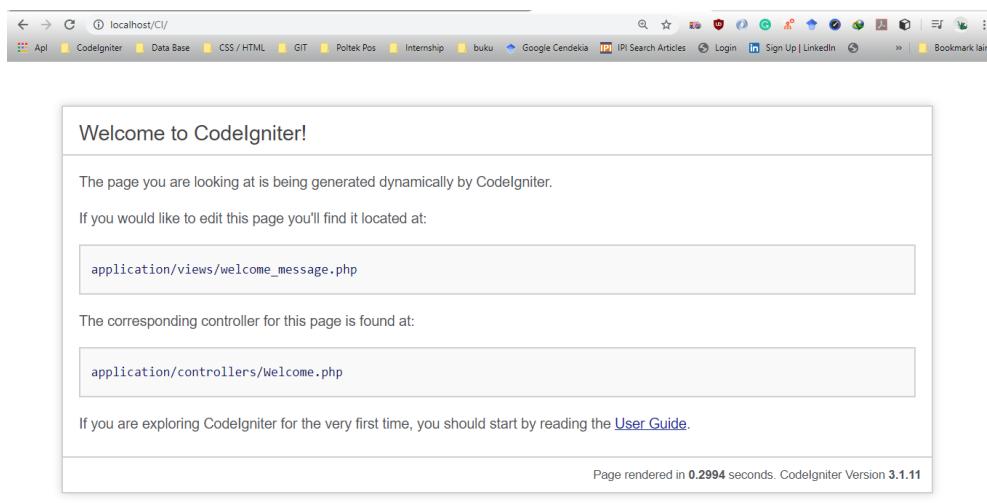


Gambar 13 memindahkan folder codeigniter

Kemudian untuk memeriksa apakah codeigniter telah terpasang dengan benar atau belum dapat dilakukan dengan cara menuliskan alamat berikut :

<http://localhost/CI>

pada web browser yang di gunakan, jika codeigniter berjalan dengan baik maka hasilnya akan seperti berikut:



Gambar 14 Hasil Codeigniter

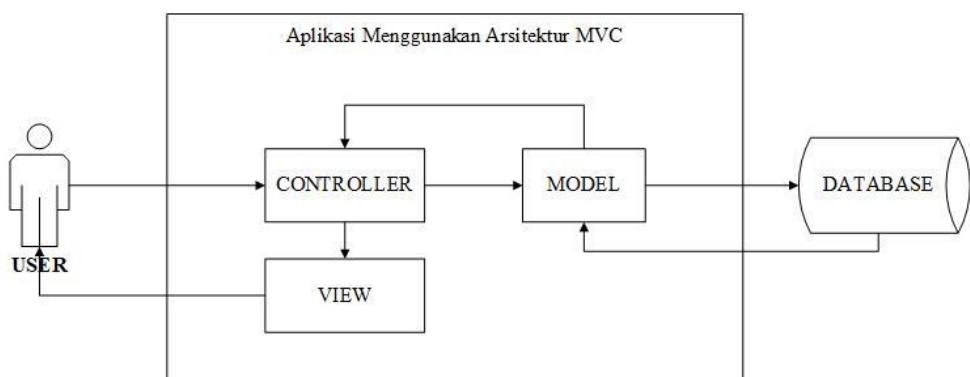
Desain MVC

Pada teknik pemerograman berorientasi objek, MVC atau model-view-controller merupakan sebuah metodelogi atau pola desain (desain pattern) yang digunakan untuk merelasikan data dan user interface dari suatu sistem agar menjadi efisien. Awalmulanya MVC digunakan untuk pemerograman berbasis dekstop khususnya untuk aplikasi-aplikasi yang dikembangkan menggunakan bahasa pemerograman C++, Java, dan Smalltalk, dengan semakin berkembangnya teknologi kini pengaplikasian model MVC tersebut diadopsi pada aplikasi yang berbasis web, sehingga hampir semua framework yang di gunakan untuk pengembangan web menggunakan konsep MVC.

Adapun komponen pada MVC dibagi menjadi tiga bagian yaitu:

1. **Model** yang berfungsi untuk mempresentasikan struktur data
2. **View** berfungsi untuk epresentasi keluaran atau output dari model yang berkaitan.
3. **Controller** yang berfungsi untuk mengambil masukan dari user atau inputan dari user dan mengubahnya menjadi perintah untuk mengeksekusi model dan/atau view

Umumnya pola MVC dapat digambarkan seperti berikut :



Gambar 15 pola MVC

Isi Folder CodeIgniter

Isi folde atau susunan direktori pada codeigniter, pada satu paket codeigniter yang telah di download di dalamnya terdapat tiga folder atau tiga direktori yaitu :

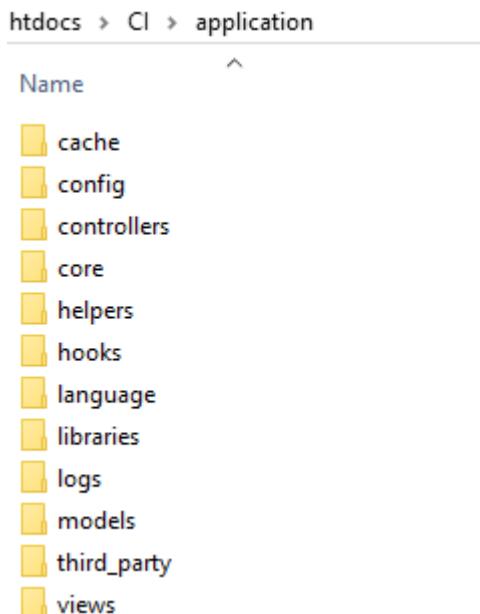
1. Application
2. System
3. User_guide

Berikut merupakan penjelasan isi setiap folder yang terdapat pada satu paket codeigniter.

1. Struktur Direktori Pada Folder Application

Direktori application merupakan tempat file-file dari aplikasi yang akan dibuat. Berikut juga model MVC juga terdapat pada direktori ini. Kemudian jika ingin menambahkan fitur-fitur untuk aplikasi juga di simpan pada direktori ini, seperti template css javascrip, template HTML, dan file untuk eksport data, juga harus di simpan pada direktori ini. File-file tersebut di simpan pada folder atau subdirektori yang telah di sediakan oleh codeigniter itu sendiri.

Adapun daftar sub durektori yang terdapat pada direktori applikasi seperti pada gambar 16 berikut:



Gambar 16 Subdirektori Application

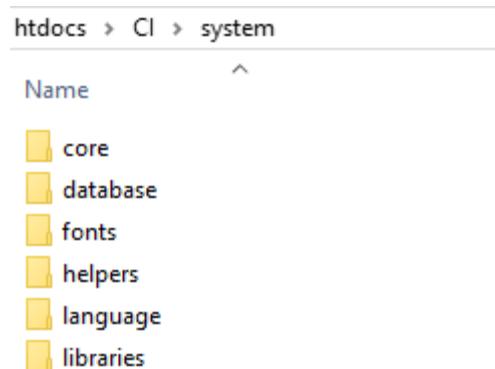
Adapun penjelasan dari direktori pada gambar 16 tersebut sebagai berikut:

- ❖ chace, digunakan untuk menyimpan halaman-halaman yang telah dibuka sebelumnya kemudian di sembunyikan (chaced)
- ❖ config, berisikan file-file konfigurasi yang digunakan untuk aplikasi yang dibuat.
- ❖ controller, berisi file-file controller dari aplikasi.
- ❖ core, digunakan untuk menempatkan daftar file kelas dasar (base class) yang nantinya diturunkan pada class-class yang digunakan oleh aplikasi
- ❖ helpers, digunakan untuk menyimpan atau menempatkan file-file helper atau pustaka buatan sendiri yang di definisikan sendiri
- ❖ hooks, berisi file pendukung aplikasi. Sebagai contoh, jika kita ingin memanggil suatu fungsi yang tersimpan di dalam file tertentu sebelum atau sesudah file controller dipanggil, maka dapat menempatkan file yang akan di eksekusi tersebut didalam sub direktori ini
- ❖ language, dalam direktori ini dapat mendefinisikan nilai konstanta-konstanta tertentu dalam bahasa yang diinginkan.
- ❖ libraries, berisi daftar file library (pustaka dalam bentuk kelas yang di definisikan sendiri)
- ❖ logs, digunakan oleh codeigniter untuk menyimpan logs (catatan) catatan yang secara otomatis ketika terjadi kesalahan.
- ❖ models, berisi daftar file model yang perlukan oleh aplikasi.
- ❖ third_party, digunakan untuk menyimpan plugin yang dikembangkan oleh pihak ketiga.

- ❖ views, berisi file view yang digunakan oleh aplikasi.

2. Struktur Direktori Pada Folder System

Pada direktori ini berisikan file-file yang telah di sediakan oleh codeigniter yang telah di kelasifikasikan berdasarkan fungsinya masing-masing, adapun sub kategori yang berada pada direktori system seperti pada gambar 17 berikut:



Gambar 17 sub direktori system

Adapun penjelasan dari direktori pada gambar 17 tersebut sebagai berikut:

- ❖ core, berisikan file-file inti berupa class-class yang di gunakan oleh codeigniter, seperti CI_Controller, CI_Model dan lain-lain
- ❖ database, berisikan file daftar file driver yang digunakan untuk mengakses database.
- ❖ fonts, berisikan daftar file font
- ❖ helpers, berisi daftar file helper standar yang di sediakan oleh codeigniter.

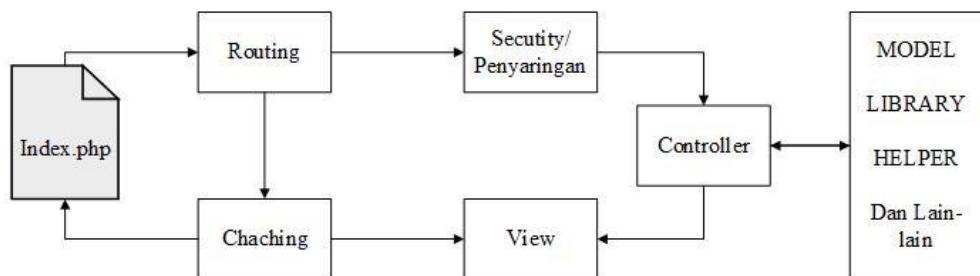
- ❖ language, berisi file-file bahasa (untuk keperluan translasi bahasa)
- ❖ libraries, berisi daftar file daftar pustaka kelas standar yang disediakan oleh codeigniter

3. Direktori user_guide

Direktori ini berisikan file-file dokumentasi penggunaan codeigniter dengan format file HTML direktori ini dapat tidak ikut sertakan dalam pembuatan aplikasi. Atau di cut keluar dari direktori temapt di codegniter. Karena direktori ini tidak berpengaruh pada kedua direktori sebelumnya.

Alur Aplikasi CodeIgniter

Adapun alur dari aplikasi yang ditulis menggunakan codeigniter digambarkan seperti pada gambar 18 berikut:



Gambar 18 Alur Aplikasi CodeIgniter

Alur pada gambar 18 tersebut dapat di jelaskan seperti berikut:

1. File index.php atau yang sering di sebut dengan entry script berperan sebagai controller depan, yang akan menginisialisasi daftar file yang dibutuhkan untuk menjalankan projek codeigniter. Dimana user

melakukan perintah aplikasi ke server web melalui index.php, dengan format Unified Resource Identification (URI) seperti berikut :

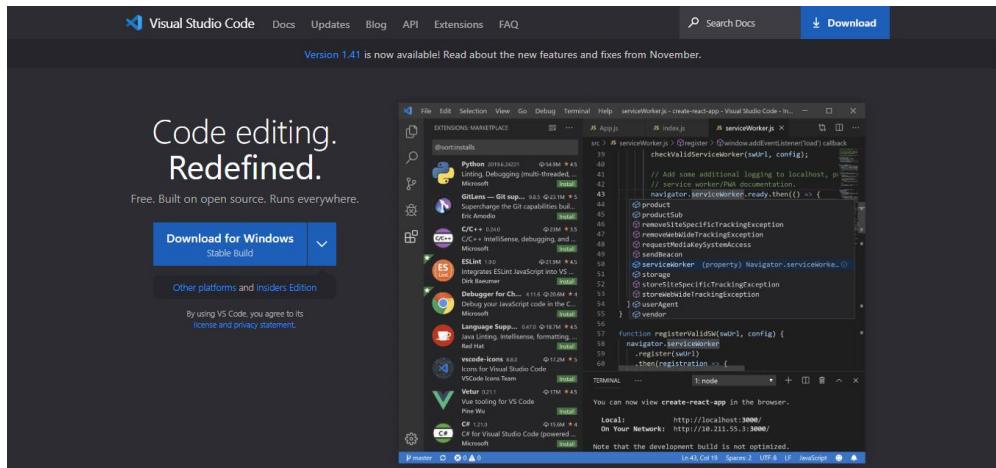
<http://namahost/index.php/kelas-controller/metode-controller>

2. Permintaan yang dikirim oleh user berbentuk URI akan ditangkap oleh router, dan router akan menentukan controller dan metode mana yang harus dipanggil
3. Jika ternyata halaman yang diminta oleh user telah di sembunyikan (chaced), halaman tersebut akan diambil dari cache dan langsung disajikan kedalam web browser.
4. Sebelum controller yang diminta oleh user di eksekusi atau dimuat, permintaan tersebut atau semua data yang dikirim oleh user akan disaring terlebih dahulu untuk keperluan pengamanan.
5. Controller akan memenuhi model, library, helper, dan file-file yang diperlukan untuk memenuhi permintaan user
6. Controller akan memuat view untuk di sajikan ke web browser jika mode caching diaktifkan, maka view akan di caching terlebih dahulu sebelum ditampilkan, dengan demikian jika ada permintaan yang sama maka halaman tersebut tinggal di ambil melalui cache.

Contoh Implementasi MVC

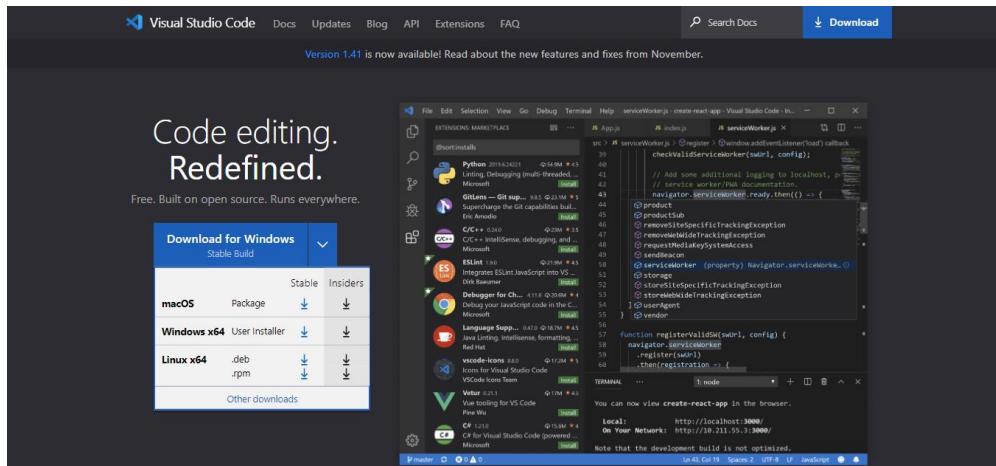
Pada contoh kali ini hal-hal yang diperlukan yaitu XAMPP yang telah di install kemudian visual studio code yang digunakan untuk teks editor pada praktik di pembahasan buku ini. Adapun langkah-langkah dalam menginstal visual studio code adalah sebagai berikut:

- Download terlebih dahulu visual studio code pada website resminya pada alamat berikut <https://code.visualstudio.com/> yang tampilannya seperti pada gambar 19 berikut:



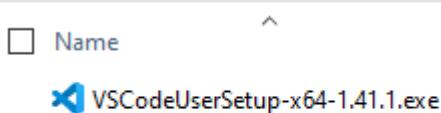
Gambar 19 Halaman Website resmi Visual Code

- Setelah itu klik tombol yang ada panah kebawah maka akan muncul tampilan seperti pada gambar 20 berikut.



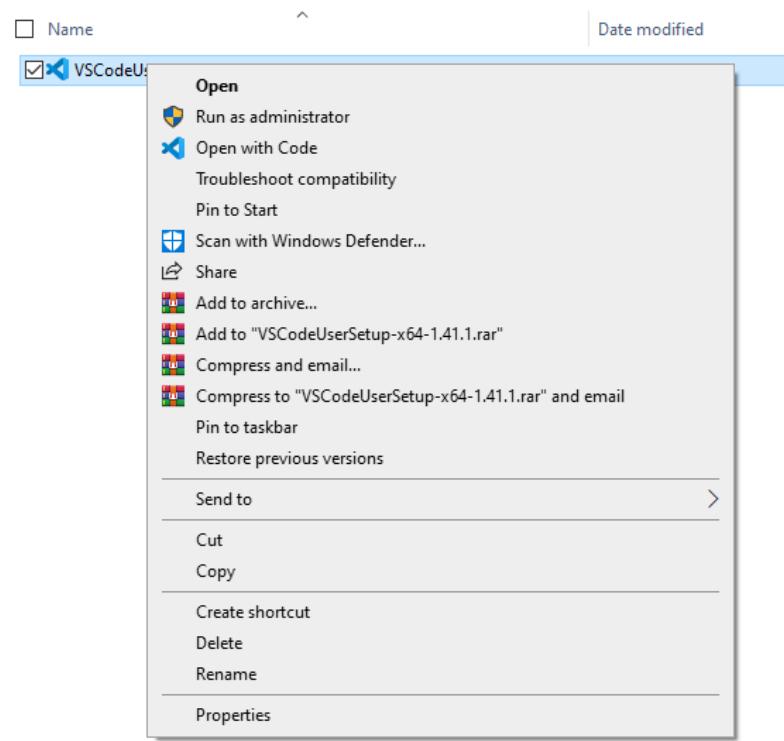
Gambar 20 Opsi memilih instalasi

3. Pilih sesuai operasi sistem karena tutorial kali ini menggunakan windows sehingga memilih visual studio code for windows



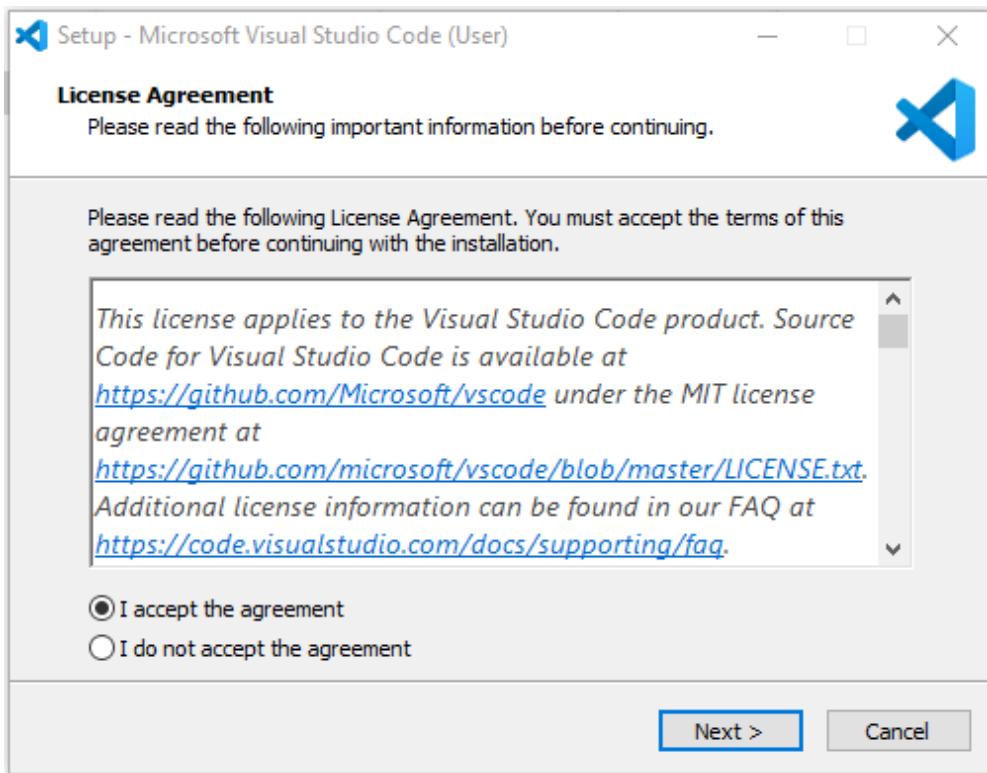
Gambar 21 Hasil Download

4. Pada gambar 21 merupakan file hasil download visual studio code



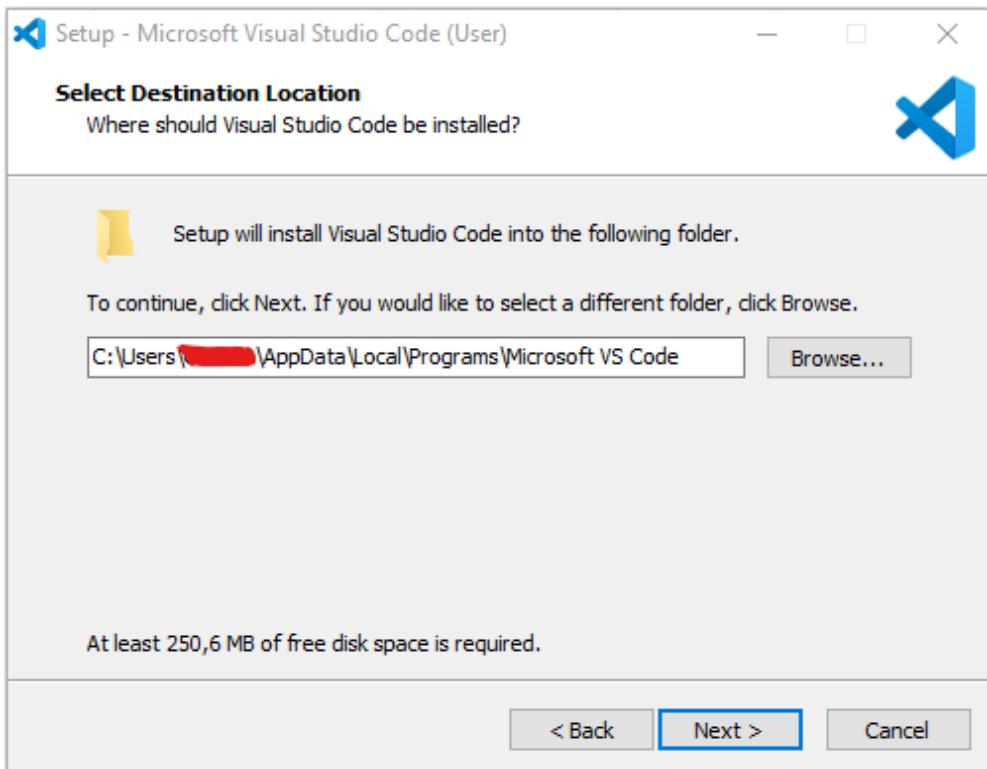
Gambar 22

5. Pada Gambar 22 merupakan peroses awal untuk instalasi visual code dengan cara klik kanan pilih Run as administrator, jika munul notifikasi berupa pop up pilih yes untuk melanjutkan proses.



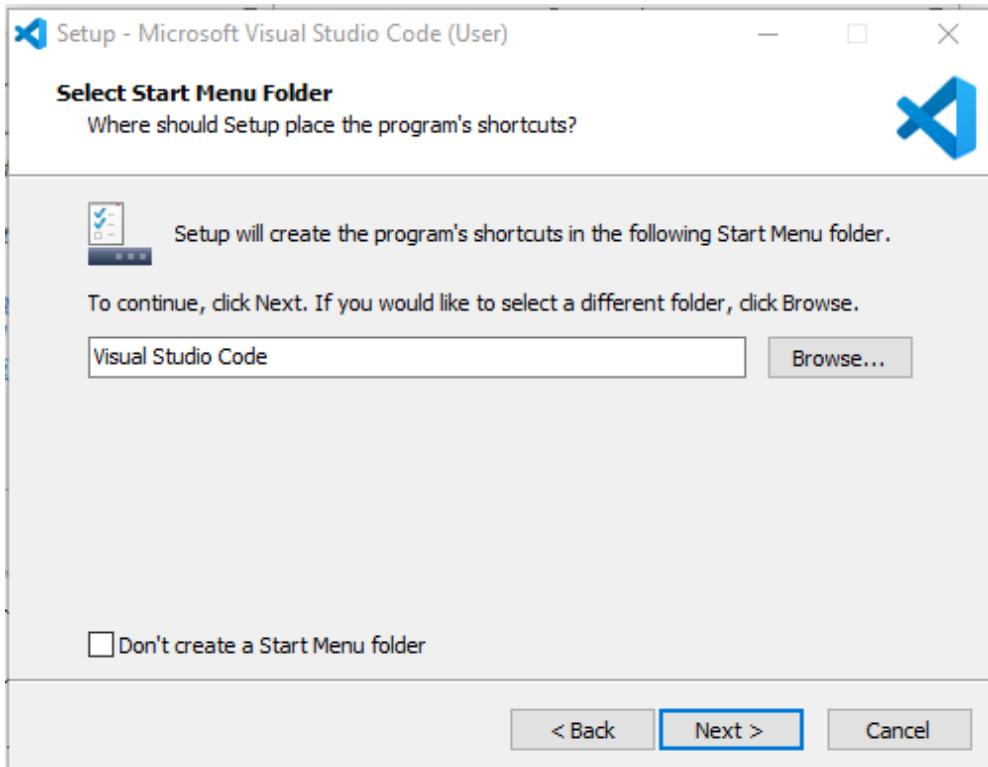
Gambar 23 License Agreement

Pada gambar 23 merupakan pilihan lisensi atau ketentuan dari visual studio code pada tampilan ini pilih I accept the agreement kemudian klik tombol **Next**, untuk melanjutkan peroses instalasi.



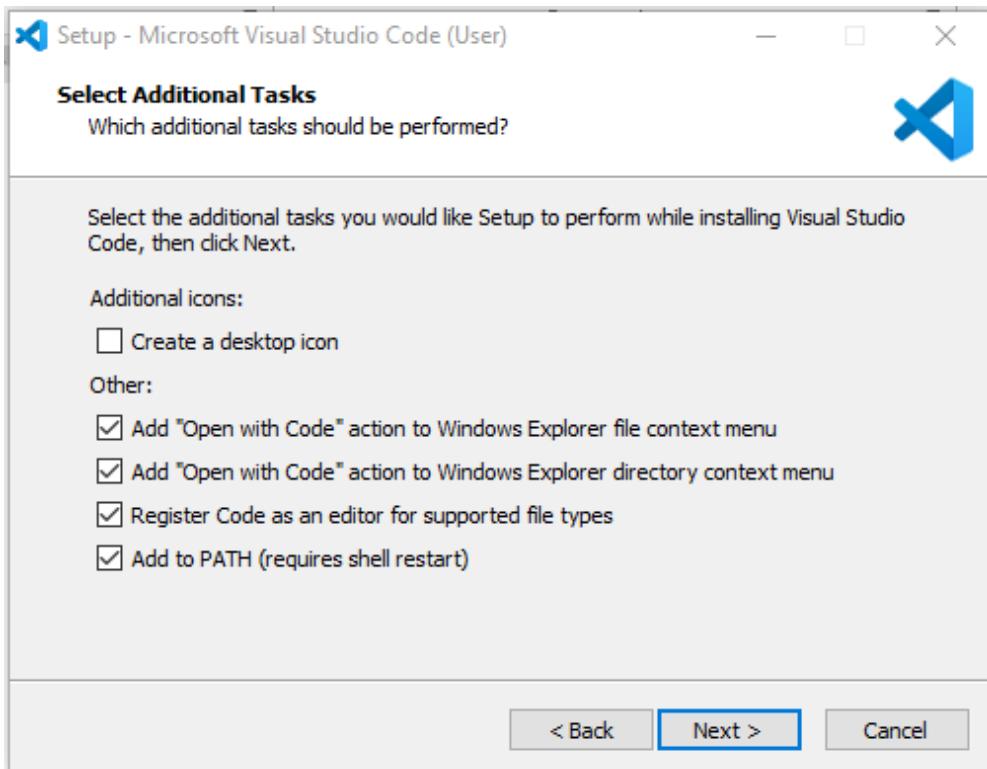
Gambar 24 memilih direktori installasi

Pada gambar 24 merupakan opsi untuk tempat di installnya aplikasi visual studio code secara default visual studio code akan terinstal di direktori C:\User\nama_user\AppData\Local\Programs\Microsoft VS Code, jika ingin memilih opsi lain dapat memilih tombol Browse... kemudian memilih folder tempat di installnya visual studio code, jika telah selesai klik tombol **Next** untuk melanjutkan proses install



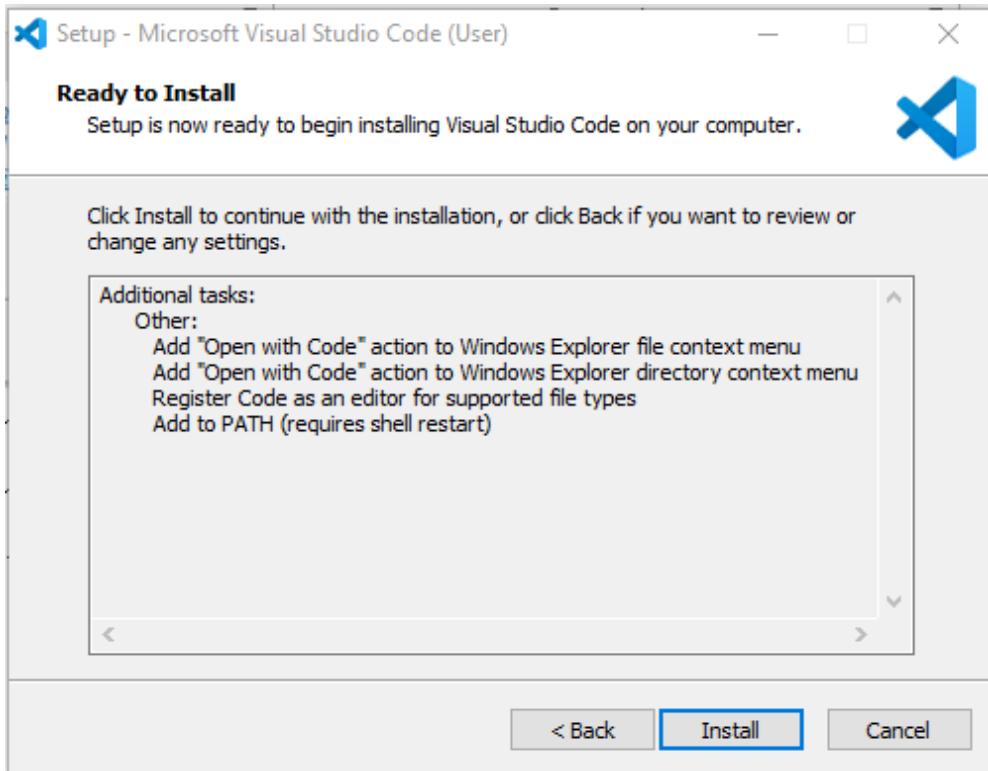
Gambar 25 memilih Start Menu

Pada gambar 25 merupakan opsi untuk memilih start menu, jika telah memilih start menu maka klik **Next** untuk melanjutkan proses install



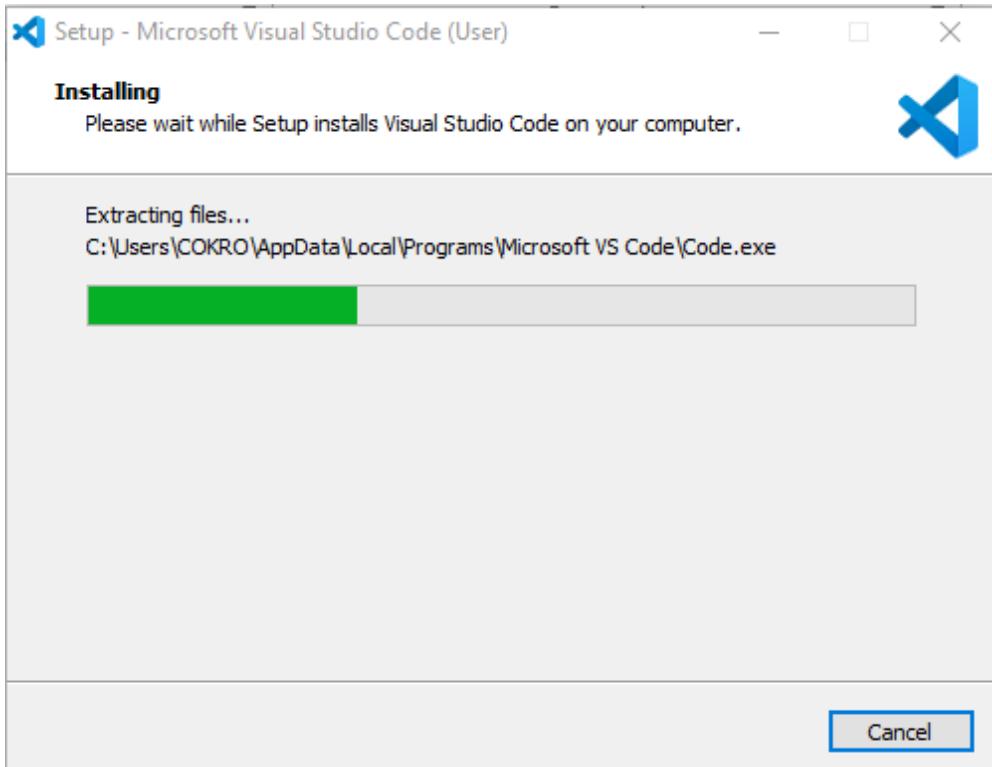
Gambar 26 menambahkan taks

Pada gambar 26 merupakan opsi untuk menambahkan taks untuk rekomendasi ceklis seperti pada gambar 26 agar dalam penggunaan visual code menjadi mudah pengaplikasiannya pada folder maupun file yang akan dilakukan proses pemerograman, jika telas selesai klik **Next** untuk melanjutkan proses install.



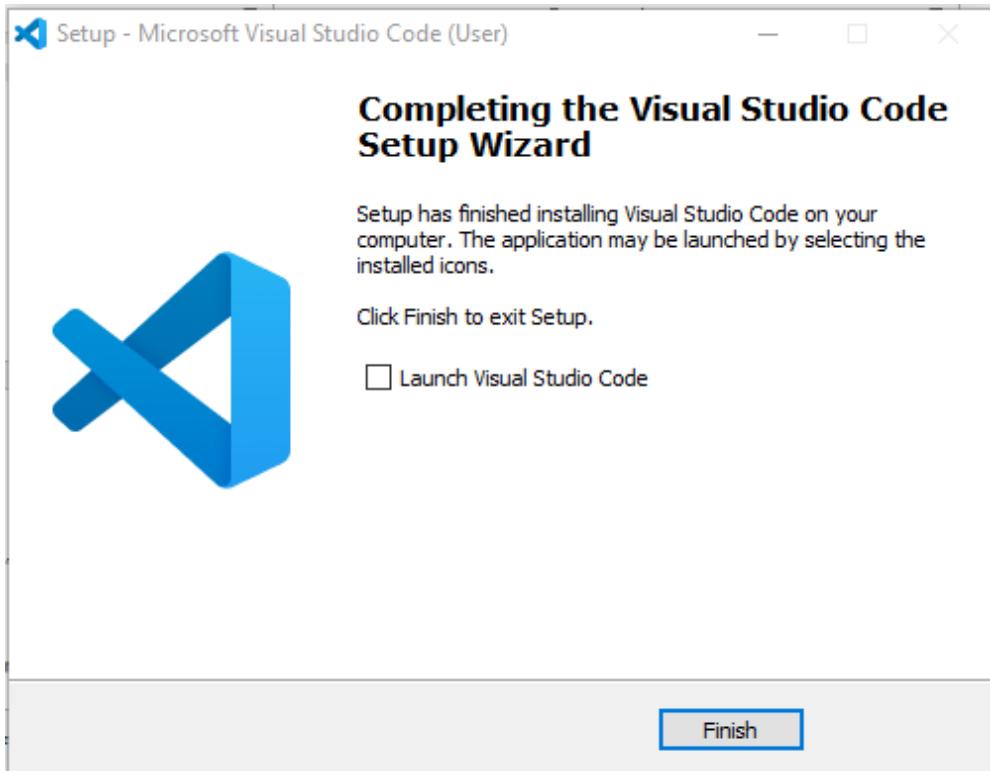
Gambar 27 setup siap untuk di instal

Pada gambar 27 menunjukan bahwa visual studio code telah siap untuk di install, klik Install untuk melanjutkan proses install visual studio code



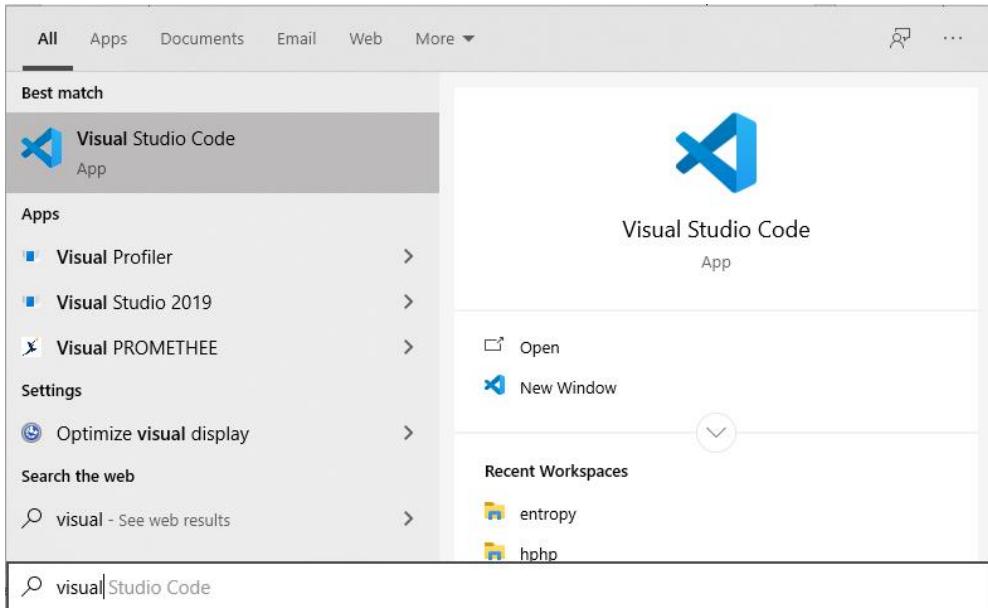
Gambar 28 proses install

Pada gambar 28 merupakan peroses install aplikasi, pada proses ini tunggu sekitar 5 menit jika telah selesai maka akan muncul gambar seperti pada gambar 29 berikut.



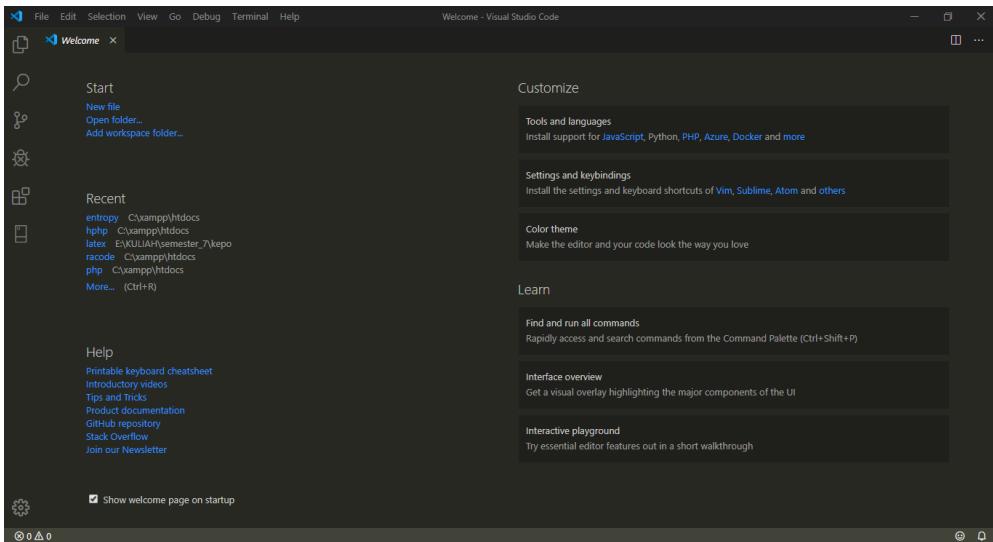
Gambar 29 Instalasi Visual studio selesai

Setelah muncul seperti gambar 29 kemudian klik finish untuk mengakhiri proses instalasi visual studio code.



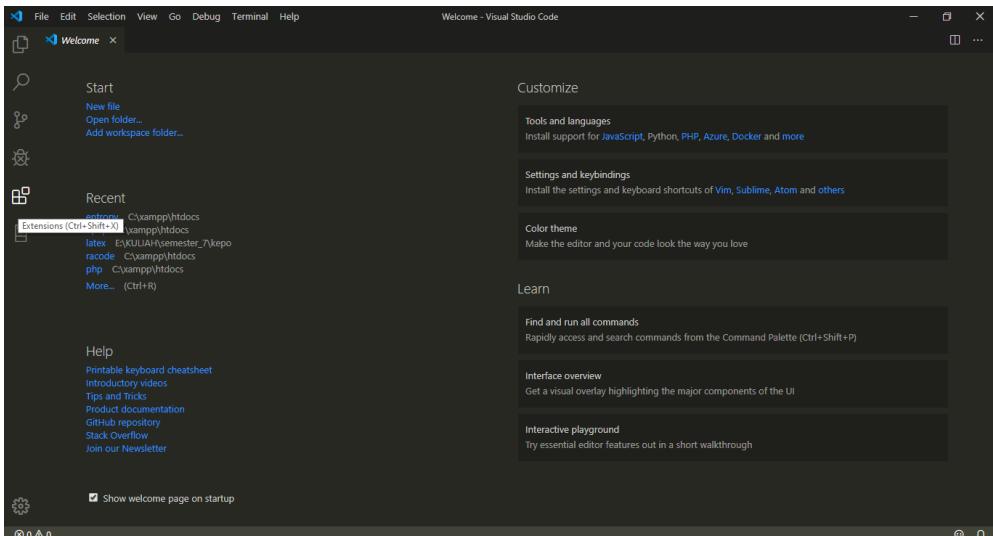
Gambar 30 Mencari Visual Studio Code

Untuk dapat menjalankan visual studio code dapat dengan cara klik icon search yang berada di dekat icon windows yang berada pada task bar kemudian tekan dan ketik visual maka muncul visual tudio code dan klik open untuk menjalankan visual studio code.



Gambar 31 tampilan visual studio stelah di install

Pada gambar 31 merupakan tampilan awal visual studio code jika telah selesai di install.



Gambar 32 Memilih Ekstensi

Pada gambar 32 merupakan menu untuk memilih ekstensi yang dapat di terapkan pada visual studio code

Agar mempermudah peroses coding alangkah baiknya ekstensi visual studio code yang dibutuhkan di install. Adapun ekstensi yang di gunakan pada projek yang di bahas pada buku ini adalah sebagai berikut

1. Auto Rename Tag

Digunakan untuk merename atau mengganti nama tag pembuka dan penutup secara bersamaan, digunakan untuk HTML dan CSS

Contoh untuk mengganti nama dari

```
<div> ... </div> menjadi <td>...</td>
```

2. Indent-rainbow

Untuk memberi tanda garis berupa warna, sehingga dapat di ketahui tag pembuka dan tag penutup dari suatu codingan. Contoh seperti pada gambar 33 berikut.

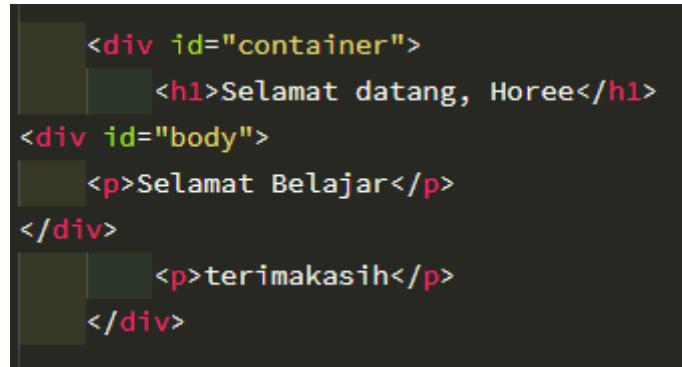


```
<div id="container">
  <h1>Selamat datang, Horee</h1>
  <div id="body">
    <p>Selamat Belajar</p>
  </div>
  <p>terimakasih</p>
</div>
```

Gambar 33

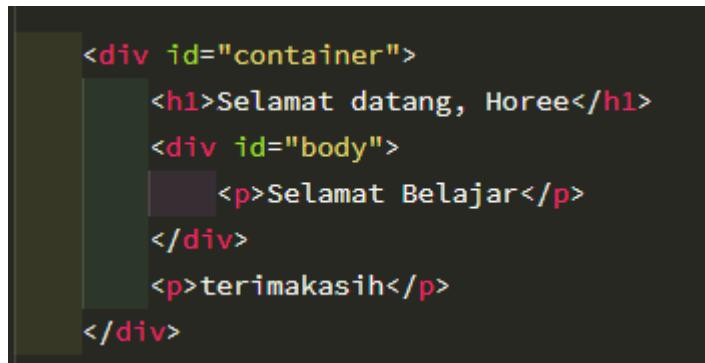
3. Beautify

Digunakan untuk merapihkan code sehingga menjadi lebih teratur sebagai contoh seperti pada gambar 34 untuk codingan yang tidak rapih sehingga menghasilkan code yang rapih seperti pada gambar 35



```
<div id="container">
    <h1>Selamat datang, Horee</h1>
    <div id="body">
        <p>Selamat Belajar</p>
    </div>
    <p>terimakasih</p>
</div>
```

Gambar 34 Sebelum Menggunakan Beautify



```
<div id="container">
    <h1>Selamat datang, Horee</h1>
    <div id="body">
        <p>Selamat Belajar</p>
    </div>
    <p>terimakasih</p>
</div>
```

Gambar 35 Setelah Menggunakan Beautify

4. IntelliSense for CSS class names HTML

digunakan untuk mengkoreksi tag-tag dari CSS dan HTML

5. Material Icon Theme

Digunakan untuk memberikan icon pada folder atau file sesuai dengan fungsinya masing masing, sebagai contoh seperti pada gambar 36 berikut:

6. Monokai Theme

Bagi yang biasa menggunakan teks editor sublime text 3 dianjurkan untuk menggunakan ekstensi ini agar tampilan code menjadi seperti sublime text 3

7. PHP intellisense for codeigniter

Digunakan untuk mengkoreksi atau secara otomatis merekomendasikan code yang akan ditulis sesuai dengan librari codeigniter.

Sebagai alternatif untuk menerapkan ekstensi pada visual studio code bisa dengan cara memasukan code tersebut pada settings.json yang ada pada visual studio code berikut merupakan code yang harus di sesuaikan pada settings.json.

```
{  
  "workbench.colorTheme": "Monokai",  
  "workbench.iconTheme": "material-icon-theme",  
  "explorer.openEditors.visible": 0,  
  "editor.minimap.enabled": false,  
  "editor.lineHeight": 23,  
  "editor.fontFamily": "'Source Code Pro',Consolas, 'Courier New', monospace",  
  "terminal.integrated.shell.windows":  
    "C:\\\\WINDOWS\\\\System32\\\\WindowsPowerShell\\\\v1.0\\\\powershell.exe",  
  "php.suggest.basic": false,  
  "editor.formatOnSave": true,  
  "liveServer.settings.donotShowInfoMsg": true,  
  "files.autoSave": "afterDelay"  
}
```

Contoh MVC sederhana

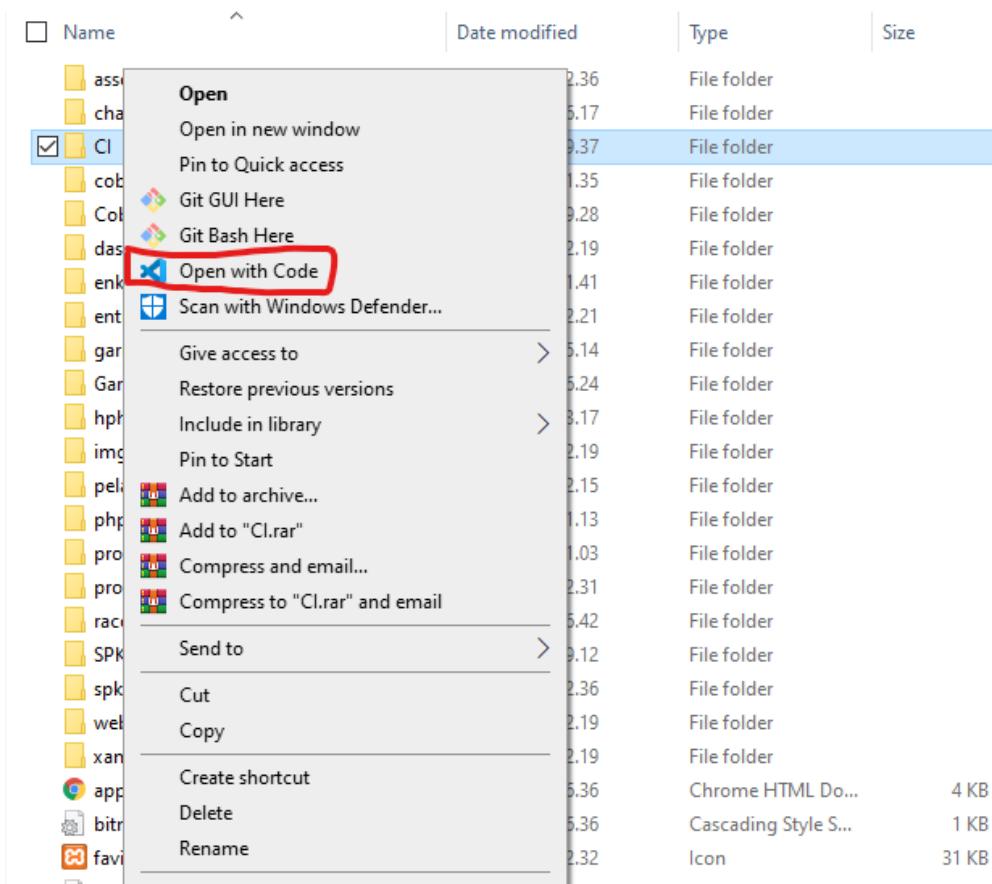
Jika aplikasi menggunakan model MVC merupakan aplikasi yang lengkap, karena pada dasarnya jika menggunakan framework codeigniter maka harus menggunakan model MVC ini, untuk urutan contoh MVC adalah sebagai berikut:

1. Buat terlebih dahulu model untuk menyajikan data
2. Buat controller untuk mengambil data dari model dan mengirimkan pada view
3. Buat view untuk menampilkan data yang telah dikirim oleh controller.

Untuk peroses dalam aplikasi tersebut adalah sebagai berikut:

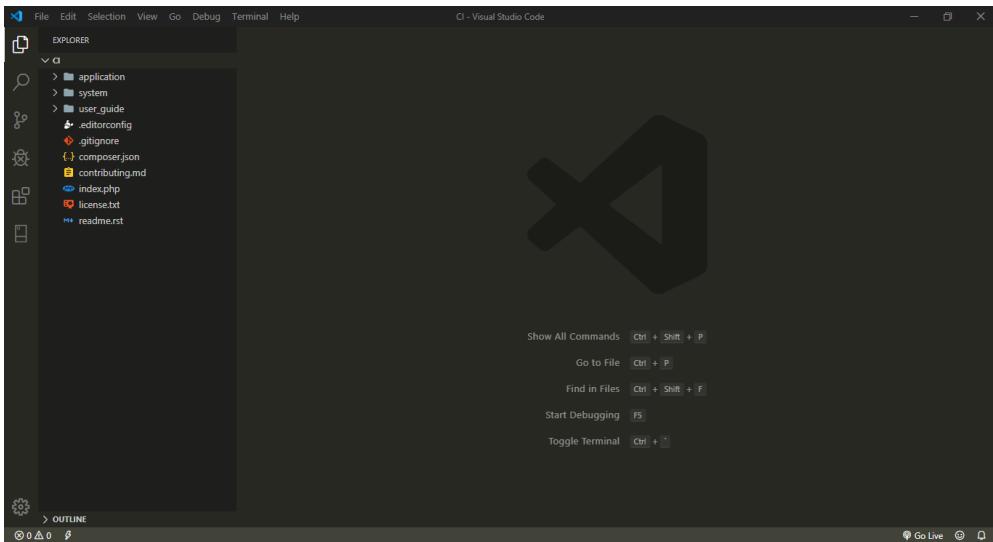
1. Controller akan mengambil data yang terdapat pada model yang berkaitan dengan controller tersebut dengan parameter nama fungsi atau nama variabel
2. Kemudian pada controller ada perintah untuk menampilkan view dimana pada view tersebut akan mengambil data dari controller yang telah diambil dari model.
3. Maka view akan memperoses data yang dikirimkan oleh controller sehingga dapat ditampilkan hasil keluarannya
4. Terakhir controller akan memperoses hasil yang ditampilkan oleh view ke layar web browser.

Agar dapat memulai contoh MVC pada codeigniter buka folder codeigniter menggunakan visual studio code seperti pada gambar 36 berikut:



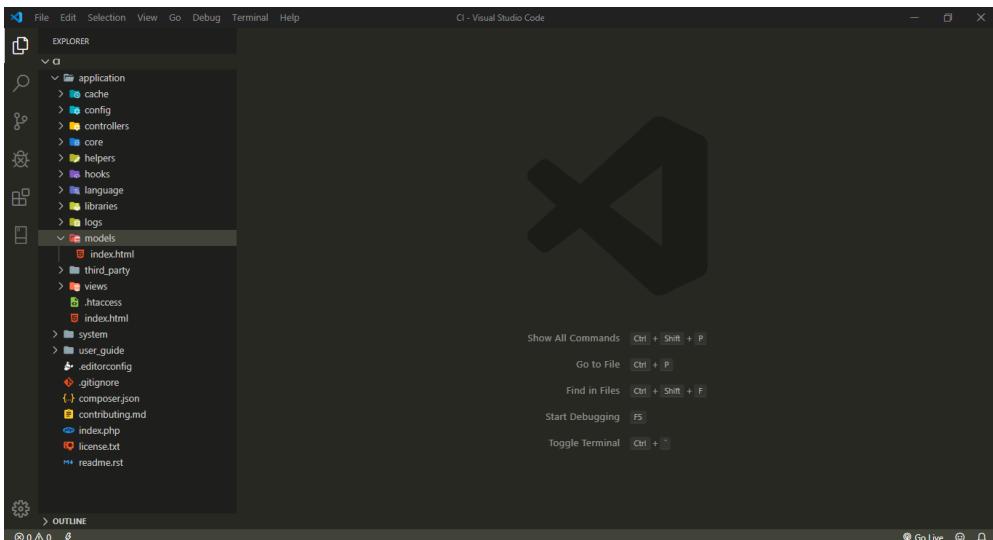
Gambar 36 membuka folder ci

Pada gambar 36 merupakan cara untuk membuka folder codeigniter menggunakan visual studio code untuk hasilnya seperti pada gambar 37 berikut:



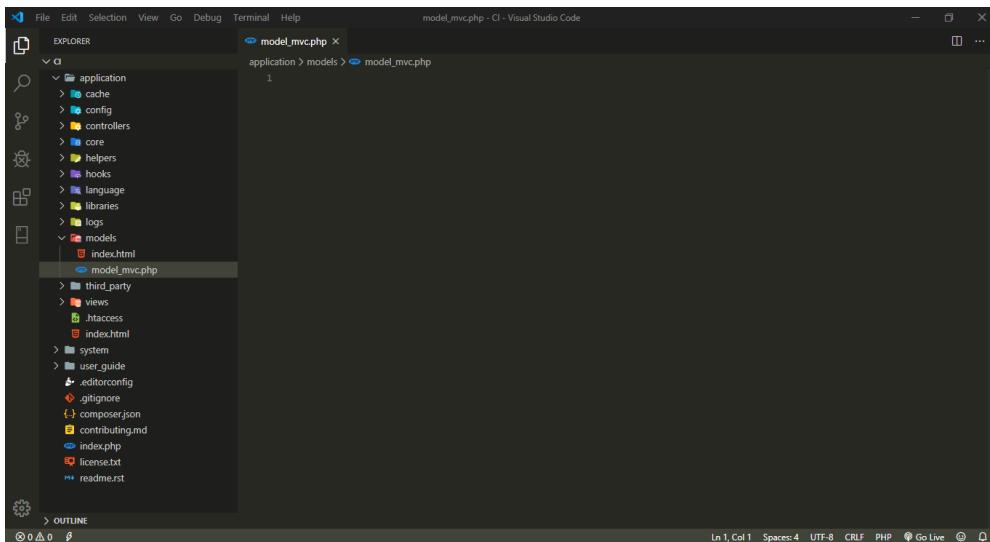
Gambar 37 folder ci di buka dengan visual studio code

Pada gambar 37 tampilan direktori code yang di buka menggunakan visual studio code. Setelah tampilannya seperti pada gambar 37 klik direktori application kemudian pilih sub direktori model sehingga tampilannya seperti pada gambar



Gambar 38

Setelah muncul seperti pada gambar 38 kemudian klik kanan pada sub direktori models kemudian pilih new file dan beri nama model_mvc.php sehingga hasilnya seperti pada gambar 39 berikut.



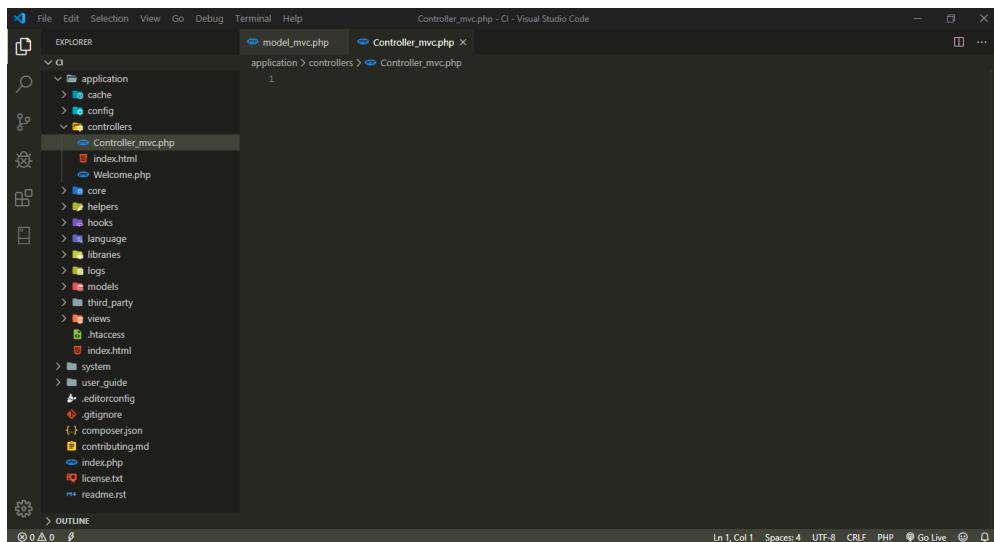
Gambar 39 file pada models

Jika telah muncul tampilan seperti pada gambar 39 maka masukan codingan berikut pada file model_mvc.php

```
<?php  
class Model_mvc extends CI_Model  
{  
    // membuat variabel atau properti dengan nama $str dengan tipe data  
    string  
    public $str = 'Mencoba CodeIgniter';  
}
```

Setelah memasukan codingan tersebut buat Controller dengan nama Controller_mvc.php dengan cara klik kanan sub direktori controllers

kemudian buat file baru dengan memilih new file kemudian berinama Controller_mvc.php. maka hasilnya seperti pada gambar 40 berikut



Gambar 40

Jika telah muncul tampilan seperti pada gambar 40 maka masukan codingan berikut pada file Controller_mvc.php

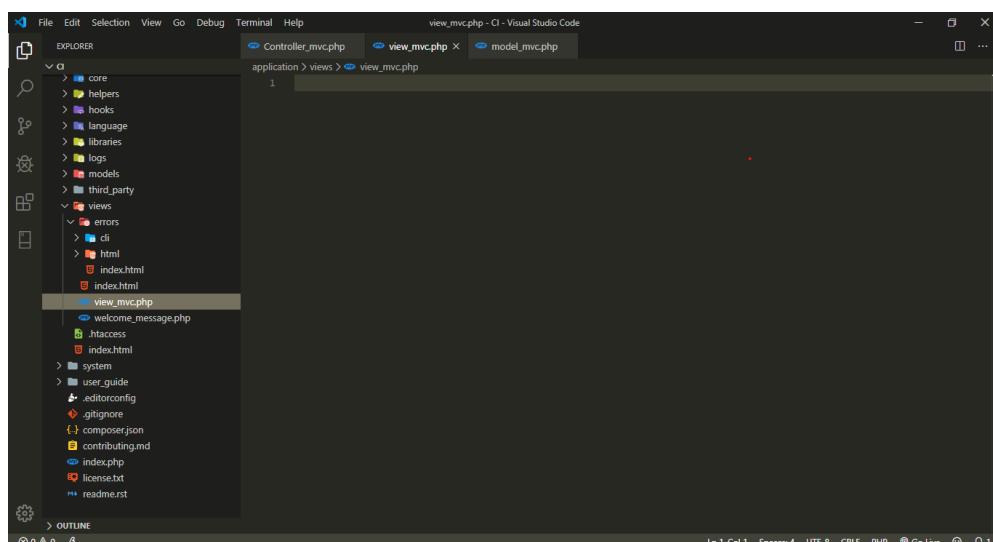
```
<?php  
  
class Controller_mvc extends CI_Controller  
{  
    public function index()  
    {  
        // memanggil atau memuat 'model_mvc'  
        $this->load->model('model_mvc');  
        // mengambil objek dari kelas model_mvc'  
        // yang dimasukan ke variabel $data_model  
        $data_model = $this->model_mvc;  
        // mengambil data yang terdapat pada model
```

```

$string = $data_model->str;
// membuat inisial data yang di kirim ke view
$data['data'] = $string;
// menampilkan dan mengirimkan data ke view
$this->load->view('view_mvc', $data);
}
}

```

Setelah file controller di buat dan di isikan codingan tersebut maka selanjutnya buat tampilan atau view yang bertujuan untuk di tampilkan pada web browser dengan nama view_mvc.php, untuk caranya yaitu pilih sub direktori views kemudian klik kanan pilih new file kemudian berinama view_mvc yang hasilnya seperti pada gambar 41 berikut



Gambar 40

Jika tempilannya telah sama atau mirip seperti pada gambar 40 maka masukan code berikut pada file view_mvc.php. walaupun ekstensi pada file tersebut merupakan php tapi isi code didalamnya merupakan tag HTML dikarenakan di gunakan untuk tampilan sehingga bisa lebih menarik.

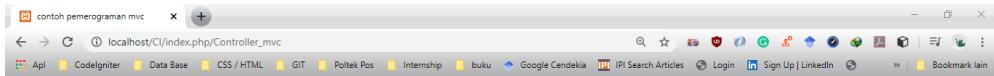
```
<html>

<head>
    <title>
        contoh pemrograman mvc
    </title>

<body>
    <h3>
        <?php echo $data; ?>
    </h3>
</body>
</head>

</html>
```

Kemudian untuk menjalankan hasil dari codingan controller model dan view tersebut nyalakan terlebih dahulu xampp yaitu dengan menyalakan xampp control panel dan memilih start pada apache dan mysql, setelah itu buka web browser kemudian isikan alamat tersebut http://localhost/CI/index.php/Controller_mvc maka hasilnya seperti pada gambar 41 berikut



Mencoba CodeIgniter

Gambar 41

Bada bagian file view dapat juga di kombinasikan dengan css sebagai contoh berikut merupakan code css yang di sisipkan pada view pada codeigniter

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

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

<head>
    <meta charset="utf-8">
    <title> contoh pemerograman mvc </title>

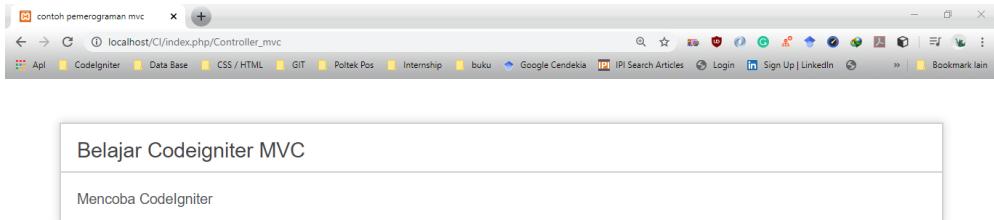
    <style type="text/css">
        ::selection {
            background-color: #E13300;
        }
    </style>
</head>
<body>
    <h1>Hello World</h1>
</body>
</html>
```

```
color: white;  
}  
  
::-moz-selection {  
    background-color: #E13300;  
    color: white;  
}  
  
body {  
    background-color: #fff;  
    margin: 40px;  
    font: 13px/20px normal Helvetica, Arial, sans-serif;  
    color: #4F5155;  
}  
  
a {  
    color: #003399;  
    background-color: transparent;  
    font-weight: normal;  
}  
  
h1 {  
    color: #444;  
    background-color: transparent;  
    border-bottom: 1px solid #D0D0D0;  
    font-size: 19px;  
    font-weight: normal;  
    margin: 0 0 14px 0;
```

```
padding: 14px 15px 10px 15px;  
}  
  
code {  
    font-family: Consolas, Monaco, Courier New, Courier,  
monospace;  
    font-size: 12px;  
    background-color: #f9f9f9;  
    border: 1px solid #D0D0D0;  
    color: #002166;  
    display: block;  
    margin: 14px 0 14px 0;  
    padding: 12px 10px 12px 10px;  
}  
  
#body {  
    margin: 0 15px 0 15px;  
}  
  
p.footer {  
    text-align: right;  
    font-size: 11px;  
    border-top: 1px solid #D0D0D0;  
    line-height: 32px;  
    padding: 0 10px 0 10px;  
    margin: 20px 0 0 0;  
}
```

```
#container {  
    margin: 10px;  
    border: 1px solid #D0D0D0;  
    box-shadow: 0 0 8px #D0D0D0;  
}  
</style>  
</head>  
  
<body>  
  
<div id="container">  
    <h1>Belajar Codeigniter MVC</h1>  
    <div id="body">  
        <p><?php echo $data ?></p>  
    </div>  
</div>  
  
</body>  
  
</html>
```

Untuk hasilnya seperti apda gambar 42 berikut ini



Gambar 42

Penjelasan Mengirim data MVC

Pada konsep mvc dikarenakan menggunakan class sehingga konsep turunan dari kelas pasti digunakan atau secara intinya konsep OOP sangat digunakan. Untuk itu berikut penjelasan cara mengirimkan data menggunakan konsep MVC pada codeigniter

Pada code model di contoh implementasi MVC terdapat code berikut:

```
public $str = 'Mencoba CodeIgniter'
```

Merupakan objek sebagai variabel str dengan isi data string (Mencoba CodeIgniter) yang mana objek tersebut berstatus public yang berarti dapat digunakan oleh class lain, sehingga jika data tersebut akan di ambil atau digunakan pada controller harus mendekralasikan terlebih dahulu class dari model seperti pada code Controller_mvc terdapat code berikut

```
$this->load->model('model_mvc');
```

Code tersebut berarti memanggil atau memuat model dari folder model dengan nama class dan file ‘model_mvc’ code tersebut dapat di sisipkan pada setiap fungsi pada class yang berada pada controller contoh seperti code tersebut

```
public function get_data()
{
    $this->load->model('model_mvc');
    // code yang berkaitan dengan model
}
```

Selain code tersebut jika model yang sama di gunakan digunakan pada setiap fungsi atau method dapat di tuliskan pada konstruktor class seperti pada code berikut

```
function __construct()
{
    parent::__construct();
    $this->load->model('model_mvc');
}
```

Setelah meload atau memanggil model maka setiap fungsi dan objek yang berstatus public dapat di gunakan pada controller, pada control implementasi MVC tersebut yaitu menggunakan objek str untuk menggunakan data di dalamnya. Untuk dapat menggunakan atau memanggil data pada objek dapat dilakukan dengan cara seperti pada code berikut:

```
$data_model = $this->model_mvc;  
  
$string = $data_model->str;
```

Pada variabel \$data_model yang memuat model_mvc kemudian untuk mengambil fungsi yang berada pada model dapat menggunakan variabel baru pada contoh terbut yaitu \$string dengan isi variabel \$data_model kemudian merujuk pada str yang merupakan objek yang terdapat pada model.

Selain menggunakan code tersebut dapat dilakukan seperti code tersebut sehingga menjadi lebih sederhana.

```
$string = $this->model_mvc->str
```

Coding tersebut intinya sama yaitu mengambil data pada objek str yang terdapat pada file model_mvc. Kemudian untuk mengirimkan data tersebut pada view harus menggunakan array asosiatif atau data harus berupa objek pada code implementasi MVC menggunakan code berikut

```
$data['data'] = $string;
```

Variabel \$string merupakan data yang di kirim ke view yang berisikan data objek str dari model, code untuk mengirim data juga dapat dituliskan seperti berikut:

```
$data = array('data' => $string);
```

Atau bisa juga sebagai berikut

```
$data = ['data' => $string]
```

Code tersebut dapat diimplementasikan pada php 5.4 atau versi di atasnya.

Untuk mengirimkan data tersebut pada view dilakukan pada saat memuat view yaitu dengan menjadikan variabel \$data menjadi parameter seperti pada code berikut

```
$this->load->view('view_mvc', $data);
```

Berdasarkan code tersebut maka code tersebut juga dapat di tuliskan seperti berikut:

```
$this->load->view('view_mvc', array('data' => $string));
```

Atau

```
$this->load->view('view_mvc', ['data' => $string]);
```

data merupakan kunci dari array asosiatif yang digunakan untuk memanggil data pada view yaitu dengan cara merubahnya menjadi variabel yaitu dengan menambahkan tanda seperti berikut(\$) sebagai contoh pada view dapat dipanggil seperti berikut

```
<?php echo $data ?>
```

Atau

```
<?= $data ?>
```

BAB 2

Metode Entropy

Pada Bab Ini akan membahas mengenai metode entropy

Kelebihan dan kekurangan metode entropy

Penjelasan mengenai cara penggunaan metode entropy

Jenis data yang bisa diolah menggunakan metode entropy

Metode – metode yang digunakan untuk melakukan pembobotan keriteria serta perbedaannya dengan metode entropy

Metode Entropy

Metode entropy merupakan metode yang di gunakan untuk menentukan tingkat kepentingan dari keriteria atau pembobotan untuk keriteria [1] selain itu metode ini juga dapat di gunakan untuk menentukan tingkat kepentingan awal atau bobot awal dari keriteria [2]. Sehingga walaupun di perhitungan awal bobot dari nilai entropy kecil pada suatu keriteria milasikan dikarenakan fariasi data yang kecil pada keriteria tersebut, namun jika keriteria tersebut di anggap penting oleh pengambil keputusan maka dia dapat memberikan bobot yang tinggi pada criteria tersebut, kedua bobot tersebut kemudian dapat di kalkulasikan sehingga mendapatkan nilai entropy akhir[4] Lalu metode ini dapat menyelidiki keserasian dalam diskriminasi diantara sekumpulan data. Nilai-nilai alternatif pada kriteria tertentun digambarkan dalam secision matrix (DM). dengan menggunakan metode entropy dengan variasi nilai tertinggi akan mendapatkan nilai tertinggi[3].

Metode entropy yang dibahas pada buku ini yaitu metode entropy shannons atau shannons entropy pada dasarnya metode ini sama hanya saja metode shannons entropy memiliki persamaan seperti berikut :

$$h_i = -h_0 \sum_{j=1}^m p_{ij} \cdot \ln p_{ij}, i = 1, \dots, n,$$

Pada tahun 1948 Claude E. Shannon memperkenalkan entropy informasi atau sering di sebut dengan Entropy shannon[5]. Metode ini juga dapat di gunakan untuk mengevaluasi bobot pada dasar subjektif dan objektif bobot[6]

Ketentuan data yang di gunakan untuk metode entropy ini yaitu:

- ❖ Data dapat berupa data kualitatif
- ❖ Data juga dapat berupa data kuantitatif
- ❖ Data-data tersebut harus dapat terukur
- ❖ Satuan untuk setiap keriteria boleh berbeda

Kelebihan dan Kekurangan Entropy

Adapun kelebihan dan kekurangan dari metode ini yaitu:

Kelebihan dari metode ini diantaranya

Dapat membobotkan data baik itu data kualitatif atau data kuantitatif selama data tersebut dapat terukur

Memberikan bobot awal untuk pengambilan keputusan

Kekurangan dari metode ini:

Hasil bobot bisa sangat kecil maupun sangat besar tergantung pada data nominal data yang di gunakan atau fariasi data yang kecil atau besar

Tahapan Penggunaan Metode Entropy

1. Normalisasi terlebih dahulu data setiap keriteria, menggunakan rumus berikut:

$$P_{ij} = \frac{X_{ij}}{\sum_{j=1}^m X_{ij}}, \quad j = 1, \dots, m, \quad i = 1, \dots, n$$

Dimana:

P_{ij} merupakan data nilai yang telah di normalisasi

X_{ij} merupakan nilai pada satu kolom dan baris

$\sum_{j=1}^m X_{ij}$ merupakan nilai total dari satu kolom data dan satu baris
data dari data ke 1 – n

m yaitu jumlah dari alternatif

Untuk lebih jelasnya dapat melihat tabel tersebut

alternatif	Kriteria 1	Kriteria 2	Kriteria 2
1	X_{ij}	X_{ij}	X_{ij}
2	X_{ij}	X_{ij}	X_{ij}
3	X_{ij}	X_{ij}	X_{ij}

$\sum_{j=1}^m X_{ij}$ dari keriteria 1 = $X_{ij} + X_{ij} + X_{ij}$

Sehingga P_{ij} dari alternatif 1 dan kriteria satu seperti berikut

$$\frac{X_{ij}}{\sum_{j=1}^m X_{ij}}$$

- Setelah menormalisasi data tersebut lakukan perhitungan entropy menggunakan persaman berikut:

$$h_i = -h_0 \sum_{j=1}^m p_{ij} \cdot \ln p_{ij}, i = 1, \dots, n,$$

Dimana:

h_i merupakan nilai entropy awal

$-h_0$ merupakan nilai dari koefisien, untuk mendapatkan nilai tersebut dapat dilakukan dengan cara $(\ln m)^{-1}$ diaman m merupakan jumlah alternatif \ln merupakan log atau nilai dari logaritma

$\sum_{j=1}^m p_{ij} \cdot \ln p_{ij}$ merupakan nilai total dari perhitungan $P_{ij} * \ln p_{ij}$ untuk tahapan perhitungannya dapat dilakukan seperti berikut:

Pertama cari nilai dari h_0 dengan cara 1 dibagi $(\ln m)$ atau seperti berikut:

$$h_0 = \frac{1}{(\ln m)}$$

Setelah hasil h_0 ditemukan lanjutkan dengan mencari nilai $P_{ij} * \ln p_{ij}$ terlebih dahulu yaitu seperti contoh pada tabel berikut:

alternatif	Kriteria 1	Kriteria 2	Kriteria 2
1	$P_{ij} * \ln p_{ij}$	$P_{ij} * \ln p_{ij}$	$P_{ij} * \ln p_{ij}$
2	$P_{ij} * \ln p_{ij}$	$P_{ij} * \ln p_{ij}$	$P_{ij} * \ln p_{ij}$
3	$P_{ij} * \ln p_{ij}$	$P_{ij} * \ln p_{ij}$	$P_{ij} * \ln p_{ij}$

Setelah nilai pada setiap baris dan kolom di temukan jumlahkan setiap baris dan satu kolom secara vertikal contoh untuk nilai total alternatif 1 – 3 pada kriteria 1 seperti berikut

$$\sum_{j=1}^m p_{ij} \cdot \ln p_{ij} = P_{ij} * \ln p_{ij} + P_{ij} * \ln p_{ij} + P_{ij} * \ln p_{ij}$$

Kemudian setelah nilai nya di temukan kalikan dengan nilai h_0 namun harus pada keadaan yang negatif dikarenakan pada rumus entropy nilai h_0 bernilai negatif

3. Cari nilai bobot entropy akhir dengan menggunakan rumus berikut:

$$W_i = \frac{d_i}{\sum_{s=1}^n d_i}, i = 1, \dots, n$$

Dimana:

W_i merupakan nilai bobot entropy untuk setiap kriteria

i merupakan keriteria ke n

d_i merupakan nilai dari $1 - h_i$

$\sum_{s=1}^n d_i$ merupakan nilai total dari d_i

Untuk lebih jelasnya terdapat pada penjelasan berikut:

Setiap nilai h_i dari semua kriteria menjadi pengurang dari 1 (satu) satu merupakan nilai bawaan dari rumus bobot entropy sehingga dihasilkan nilai dari d_i

Setelah itu cari nilai total dengan cara menambahkan nilai setiap keriteria contoh seperti $d_1 + d_2 + \dots + d_n$

Kemudian untuk mencari nilai bobot entropy = W_i bagi setiap nilai d_i dari setiap kriteria dengan nilai total dari d_i

Metode Lain Untuk Pembobotan

Untuk membobotkan kriteria juga dapat dilakukan menggunakan metode lain selain metode entropy salah satunya yaitu dengan menggunakan metode AHP (*Analitical Hirarki Process*)

Jurnal jurnal yang di gunakan pada bab ini

IMPLEMENTASI METODE ENTROPY DAN TECHNIQUE FOR ORDER PREFERENCE BY SIMILARITY TO IDEAL SOLUTION (TOPSIS) DALAM PEMILIHAN BIRO PERJALANAN UMROH

PENERAPAN METODE ENTROPY DAN METODE PROMETHEE DALAM MERANGKING KUALITAS GETAH KARET

Pemilihan Supplier Untuk Industri Makanan Menggunakan Metode Promethee, Journal Of Logistics and Supply Chain Management,

USULAN PENENTUAN PRIORITAS SUPPLIER BAHAN BAKU PLATE STEEL DENGAN METODE PROMETHEE DI PT DIRGANTARA INDONESIA (PERSERO)

A new qualitative acoustic emission parameter based on Shannon's entropy for damage monitoring

Application of Integrated Shannon's Entropy and VIKOR Techniques in Prioritization of Flood Risk in the Shemshak Watershed, Iran

BAB 3

IMPLEMENTASI ENTROPY

Pada BAB ini akan dibahas tentang

- ❖ Implementasi dari metode entropy menggunakan data-data yang berbeda beda
- ❖ Menggunakan tiga contoh perhitungan entropy
- ❖ Menggunakan data-data yang masih bisa terukur

Perhitungan Shannon's Entropy

Buat terlebih dahulu data seperti berikut contoh :

1 tabel untuk menghitung bobot pada handphone

Alternatif	Harga	Kamera	Batrei	Memori
Handphone 1	1000	10 MP	2000 mAh	16GB
Handphone 2	2000	10 MP	3500 mAh	32GB
Handphone 3	1500	13MP	2000 mAh	32GB

Pertama normalisasi data yang terdapat pada tabel tersebut dengan cara mencari nilai maksimum dari data tersebut:

Alternatif	Harga	Kamera	Batrei	Memori
Handphone 1	1000	10	2000	16
Handphone 2	2000	10	3500	32
Handphone 3	1500	13	2000	32
Niali total	4500	33	7500	80

Setelah nilai total untuk setiap keriteria di dapatkan maka lanjutkan normalisasi data dengan menjadikan nilai total tersebut sebagai pembagi untuk setiap data yang terdapat pada baris keriteria masing-masing seperti pada tabel berikut

Alternatif	Harga	Kamera	Batrei	Memori
Handphone 1	1000/4500	10/33	2000/7500	16/80
Handphone 2	2000/4500	10/33	3500/7500	32/80

Handphone 3	1500/4500	13/33	2000/7500	32/80
-------------	-----------	-------	-----------	-------

Setelah data di bagi oleh nilai total masing masing keriteria maka mendapatkan hasil, untuk hasilnya sebagai berikut

Alternatif	Harga	Kamera	Batrei	Memori
Handphone 1	0.222	0.303	0.267	0.2
Handphone 2	0.444	0.303	0.467	0.4
Handphone 3	0.333	0.393	0.267	0.4

setelah data di normalisasi dan menghasilkan data seperti pada tabel tersebut maka lanjutkan ke proses persamaan entropy, pertama cari terlebih dahulu nilai konsistensi pada data tersebut menggunakan rumus berikut:

$$h_0 = \frac{1}{(\ln m)}$$

Dimana diketahui $m = 3$ karena pada data tersebut terdapat 3(tiga) alternatif

Sehingga hasilnya seperti berikut

$$h_0 = \frac{1}{(\ln 3)} = 0,910239227$$

Setelah itu kemudian cari nilai $P_{ij} * \ln p_{ij}$ untuk setiap kolom pada data yang terdapat pada tabel tersebut sehingga seperti pada tabel tersebut:

Alternatif	Harga	Kamera	Batrei	Memori
Handphone 1	$(0.222) * \ln (0.222)$	$(0.303) * \ln (0.303)$	$(0.267) * \ln (0.267)$	$(0.2) * \ln (0.2)$

Handphone 2	$(0.444) * \ln(0.444)$	$(0.303) * \ln(0.303)$	$(0.467) * \ln(0.467)$	$(0.4) * \ln(0.4)$
Handphone 3	$(0.333) * \ln(0.333)$	$(0.393) * \ln(0.393)$	$(0.267) * \ln(0.267)$	$(0.4) * \ln(0.4)$

Sehingga hasilnya seperti berikut

Alternatif	Harga	Kamera	Batrei	Memori
Handphone 1	- 0,334127293	-0,361788809	- 0,352575268	- 0,321887582
Handphone 2	-0,36050	-0,361788809	- 0,355585952	- 0,366516293
Handphone 3	- 0,366171059	-0,367040647	- 0,352575268	- 0,366516293

Selanjutnya cari nilai total untuk setiap keriteria dari data tersebut, berikut merupakan nilai total untuk setiap keriteria:

Harga	Kamera	Batrei	Memori
-1,06079559	-1,090618266	-1,060736487	-1,054920168

setelah mendapatkan nilai total kalikan nilai total tersebut dengan nilai konsistensi nilai konsistensi tersebut harus minus dikarenakan ketentuan dari rumus entropy berikut merupakan peroses dan hasil kali nilai total dengan nilai konsistensi:

Harga	Kamera	Batrei	Memori
-0,910239227	-0,910239227	-0,910239227	-0,910239227

*-1,06079559	*-1,090618266	*-1,060736487	* -1,054920168
--------------	---------------	---------------	----------------

Sedangkan untuk hasilnya seperti berikut:

Harga	Kamera	Batrei	Memori
0,965577757	0,992723527	0,96552396	0,960229718

Setelah itu hasil tersebut jadikan pengurang dari 1 (satu) nilai satu merupakan bawaan dari rumus

Untuk caranya seperti berikut :

Harga	Kamera	Batrei	Memori
1-0,965577757	1-0,992723527	1-0,96552396	1-0,960229718

Maka hasilnya seperti berikut:

Harga	Kamera	Batrei	Memori
0,034422	0,007276	0,034476	0,03977

Setelah mendapatkan nilai hasil pengurangan untuk setiap keriteria kemudian jumlahkan setiap nilai tersebut atau cari nilai totalnya, untuk nilai totalnya sebagai berikut:

0,115945

Setelah di temukan nilai total hasil penjumlahan nilai dari semua keriteria, nilai total tersebut di jadikan pembagi untuk setiap keriteria seperti berikut:

Harga	Kamera	Batrei	Memori
0,034422 /	0,007276	0,034476	0,03977

0,115945	/ 0,115945	/ 0,115945	/ 0,115945
----------	------------	------------	------------

Untuk hasil bobot setiap keriteri adalah sebagai berikut.

Harga	Kamera	Batrei	Memori
0,296884138	0,06275795	0,29734813	0,343009782

Yang mana jika keseruhan bobot tersebut dibobotkan akan mendapatkan nilai total 1 (satu)

Dari contoh berikut nilai total bobot akan tetap satu walaupun keriteria lebih dari 4 misalkan 5 atau 6 maka dari itu sebagai pembuktian berikut merupakan contoh kedua penggunaan metode entropy untuk pembobotan.

Pada contoh data berikut merupakan contoh data penentuan bobot dari 5(lima) alternatif yaitu yang di misalkan sebagai handphone yang masing-masing memiliki lima keriteria diantaranya terdiri dari Harga (satuan Dolar), kamera depan (tolak ukur pixcel), kamera belakang (tolak ukur pixcel), RAM (tolak ukur gigabyte (GB)) kapasitas batarai (torakukur mAh), dan Memory penyimpanan (tolak ukur gigabyte (GB)). Untuk lebih jelasnya berikut merupakan contoh data untuk menentukan bobot criteria.

Alternatif	Harga (\$)	Kamera Depan	Kamera Belakang	RAM	Kapasitas Batrai
Handphone 1	300	5	24	2	64
Handphone 2	250	5	13	2	32
Handphone 3	330	13	24	3	64
Handphone 4	210	5	8	2	32
Handphone 5	190	2	5	2	16

Langkah pertama yaitu normalisasi data seperti pada contoh pertama yaitu cari nilai total dari setiap keriteria seperti pada tabel data berikut:

Alternatif	Harga (\$)	Kamera Depan	Kamera Belakang	RAM	Kapasitas Batrai
Handphone 1	300	5	24	2	64
Handphone 2	250	5	13	2	32
Handphone 3	330	13	24	3	64
Handphone 4	210	5	8	2	32
Handphone 5	190	2	5	2	16
Nilai Total	1280	30	74	11	208

Setelah nilai total untuk setiap keriteria di dapatkan maka lanjutkan normalisasi data dengan menjadikan nilai total tersebut sebagai pembagi untuk setiap data yang terdapat pada baris keriteria masing-masing seperti pada tabel berikut

Alternatif	Harga (\$)	Kamera Depan	Kamera Belakang	RAM	Kapasitas Batrai
Handphone 1	300/1280	5/30	24/74	2/74	64/74
Handphone 2	250/1280	5/30	13/74	2/74	32/74
Handphone 3	330/1280	13/30	24/74	3/74	64/74
Handphone 4	210/1280	5/30	8/74	2/74	32/74
Handphone 5	190/1280	2/30	5/74	2/74	16/74

Setelah data di bagi oleh nilai total masing masing keriteria maka mendapatkan hasil, untuk hasilnya sebagai berikut

Alternatif	Harga (\$)	Kamera Depan	Kamera Belakang	RAM	Kapasitas Batrai
Handphone 1	0,234375	0,166667	0,324324	0,181818	0,307692
Handphone 2	0,195313	0,166667	0,175676	0,181818	0,153846
Handphone 3	0,257813	0,433333	0,324324	0,272727	0,307692
Handphone 4	0,164063	0,166667	0,108108	0,181818	0,153846
Handphone 5	0,148438	0,066667	0,067568	0,181818	0,076923

setelah data di normalisasi dan menghasilkan data seperti pada tabel tersebut maka lanjutkan ke proses persamaan entropy, pertama cari terlebih dahulu nilai konsistensi pada data tersebut menggunakan rumus berikut:

$$h_0 = \frac{1}{(\ln m)}$$

Dimana diketahui $m = 5$ karena pada data tersebut terdapat 5(lima) alternatif

Sehingga hasilnya seperti berikut

$$0,621334935$$

Setelah itu kemudian cari nilai $P_{ij} * \ln p_{ij}$ untuk setiap kolom pada data yang terdapat pada tabel tersebut sehingga seperti pada tabel tersebut:

Alternatif	Harga (\$)	Kamera Depan	Kamera Belakang	RAM	Kapasitas Batrai
Hand-	(0,234375)	(0,166667)	(0,324324)	(0,181818)	(0,307692)

phone 1	* ln (0,234375)	* ln (0,166667)	* ln (0,324324)	* ln (0,181818)	* ln (0,307692)
Hand- phone 2	(0,195313) * ln (0,195313)	(0,166667) * ln (0,166667)	(0,175676) * ln (0,175676)	(0,181818) * ln (0,181818)	(0,153846) * ln (0,153846)
Hand- phone 3	(0,257813) * ln (0,257813)	(0,433333) * ln (0,433333)	(0,324324) * ln (0,324324)	(0,272727) * ln (0,272727)	(0,307692) * ln (0,307692)
Hand- phone 4	(0,164063) * ln (0,257813)	(0,166667) * ln (0,166667)	(0,108108) * ln (0,108108)	(0,181818) * ln (0,181818)	(0,153846) * ln (0,153846)
Hand- phone 5	(0,148438) * ln (0,148438)	(0,066667) * ln (0,066667)	(0,067568) * ln (0,067568)	(0,181818) * ln (0,181818)	(0,076923) * ln (0,076923)

Lalu untuk hasilnya sebagai berikut

Alternatif	Harga (\$)	Kamera Depan	Kamera Belakang	RAM	Kapasitas Batrai
Handphone 1	-0,34004	-0,29863	-0,36519	-0,30995	-0,362663
Handphone 2	-0,31898	-0,29863	-0,30552	-0,30995	-0,28797
Handphone 3	-0,34947	-0,36237	-0,36519	-0,35435	-0,362663
Handphone 4	-0,29654	-0,29863	-0,2405	-0,30995	-0,28797
Handphone 5	-0,28316	-0,18054	-0,18207	-0,30995	-0,197304

Selanjutnya cari nilai total untuk setiap keriteria dari data tersebut, berikut merupakan nilai total untuk setiap keriteria:

Harga (\$)	Kamera Depan	Kamera Belakang	RAM	Kapasitas Batrai
-1,58819	-1,43879	-1,45848	-1,59417	-1,498569

setelah mendapatkan nilai total kalikan nilai total tersebut dengan nilai konsistensi nilai konsistensi tersebut harus minus dikarenakan ketentuan dari rumus entropy berikut merupakan peroses dan hasil kali nilai total dengan nilai konsistensi:

Harga (\$)	Kamera Depan	Kamera Belakang	RAM	Kapasitas Batrai
-0,62133 *-1,58819	-0,62133 *-1,43879	-0,62133 *-1,45848	-0,62133 *-1,59417	-0,62133 *-1,498569

Sehingga mendapatkan hasil seperti berikut:

Harga (\$)	Kamera Depan	Kamera Belakang	RAM	Kapasitas Batrai
0,986796	0,893971	0,906202	0,990511	0,931113

Setelah hasil perkalian dengan nilai konsistensi di dapatkan hasil tersebut di jadikan sebagai nilai pengurang dari 1 (satu), nilai 1 (satu) merupakan ketentuan dari rumus entropy, untuk peroses pengurangannya sebagai berikut:

Harga (\$)	Kamera Depan	Kamera Belakang	RAM	Kapasitas Batrai
1-0,986796	1-0,893971	1-0,906202	1-0,990511	1-0,931113

Berikut merupakan hasil dari pengurangan tersebut:

Harga (\$)	Kamera Depan	Kamera Belakang	RAM	Kapasitas Batrai
0,013204	0,106029	0,093798	0,009489	0,068887

Jika telah mendapatkan hasil pengurangan, di lanjutkan dengan mencari nilai total dari semua keriteria yaitu dengan menambahkan hasil pengurangan tersebut dari keriteria harga samapai keriteria kapas batrai. Berikut merupakan hasil totalnya

0,291406393

Setelah di temukan nilai total hasil penjumlahan nilai dari semu keriteria, nilai total tersebut di jadikan pembagi untuk setiap keriteria seperti berikut:

Harga (\$)	Kamera Depan	Kamera Belakang	RAM	Kapasitas Batrai
0,013204/ 0,291406	0,106029/ 0,291406	0,093798/ 0,291406	0,009489/ 0,291406	0,068887/ 0,291406

Untuk hasil bobot setiap keriteri adalah sebagai berikut.

Harga (\$)	Kamera Depan	Kamera Belakang	RAM	Kapasitas Batrai
0,04531	0,363853	0,321882	0,032561	0,236394

Dari hasil pembobotan tersebut jika di totalkan nilainya akan berjumlah 1, hal ini membuktikan walaupun jumlah keriteria ditambah menjadi banyak maka nilai total akan tetap 1 (satu) yang mana nilai satu akan terbagi sesuai banyaknya kriteria.

Dari kedua contoh tersebut dapat dilihat perbedaan yang cukup signifikan yang terdapat pada kriteria harga yang pada contoh ke satu memiliki bobot yang dominan sedangkan pada contoh yang kedua bobot untuk keriteria harga menjadi sangat kecil, begitu pula untuk keriteria kamera pada contoh yang pertama memiliki nilai bobot yang cukup kecil sedangkan pada contoh yang kedua memiliki nilai yang cukup dominan baik itu kriteria kamera depan maupun kamera belakang.

Hah ini membuktikan bahwa tingkat fariasi data yang terdapat pada setiap kriteria sangat berpengaruh untuk lebih jelasnya dapat dilihat pada tabel data untuk contoh ke 1 berikut

Alternatif	Harga	Kamera	Batrei	Memori
Handphone 1	1000	10 MP	2000 mAh	16GB
Handphone 2	2000	10 MP	3500 mAh	32GB
Handphone 3	1500	13MP	2000 mAh	32GB

Yang di maksud tingkat fariasi data yang terdapat keriteria yaitu perubahan data atau jenis data yang terdapat pada keriteria misalkan pada keriteria Harga pada tebel tersebut ternyata nilai untuk keriteria tersebut memiliki pola yaitu kelipatan dari 5 begitu pula pada keriteria batrai juga memiliki pola dan juga kriteria memori. Sedangkan kenapa suatu keriteria bisa memiliki bobot yang kecil di karenakan data pada keriteria tersebut acak seperti pada kriteria kamera pada tabel tersebut.

Selain kedua hal tersebut hal yang dapat memperbesar bobot kriteria secara signifikan yaitu data yang acak tetapi memiliki jarak yang sangat jauh sepeerti pada keriteria kamera ada nilai 2 dan 24 maka keriteria tersebut kemungkinan memiliki bobot yang sangat besar.

Kemudian bagaimana cara mengatasi hal tersebut, agar pembagian bobot bisa sesuai dan tidak terlalu membingungkan bagi pengambil keputusan bisa dilakukan cara mengkalsifikasikan data data tersebut, lantas bagaimana cara mengkalsifikasikan data tersebut misalkan data yang terdapat pada stu keriteria ternyata memiliki pola yaitu data paling kecil merupakan 1 dan data paling besar merupakan 50 bisa di klasifikasikan menjadi 5 data lain dengan nilai 1 – 5 atau bisa di klasifikasikan datanya dari 1 sampai 9 untuk contohnya seperti berikut.

Misalkan data yang terdapat pada keriteria ke satu memiliki nilai antara 1 sampai 50 maka di bagi menjadi nilai tersebut misalkan menjadi 5

klasifikasi atau bisa di sebut sub keriteria misalkan pembagiannya seperti berikut:

- ❖ Jika nilai pada kriteria1 diantara 1 sampai 10 maka memiliki nilai atau bobot 1 (satu)
- ❖ Jika nilai pada kriteria1 diantara 11 sampai 20 maka memiliki nilai atau bobot 2 (dua)
- ❖ Jika nilai pada kriteria1 diantara 21 sampai 30 maka memiliki nilai atau bobot 3 (dua)
- ❖ Jika nilai pada kriteria1 diantara 31 sampai 40 maka memiliki nilai atau bobot 4 (dua)
- ❖ Jika nilai pada kriteria1 diantara 41 sampai 50 maka memiliki nilai atau bobot 5 (dua)

Atau untuk lebih sederhananya seperti pada tabel berikut:

Aturan	Bobot atau nilai
$1 \leq X \leq 10$	1
$11 \leq X \leq 20$	2
$21 \leq X \leq 30$	3
$31 \leq X \leq 40$	4
$41 \leq X \leq 50$	5

Cara klasifikasi ini dapat di terapkan untuk mencari entropy namun untuk penggunaanya harus disesuaikan dengan keadaan dan keperluan pengambil keputusan sehingga dapat menghasilkan bobot yang sesuai untuk keriteria, kemudian untuk nilai 1 sampai 50 pada contoh tersebut hanya perumpamaan agar mendapat gambaran untuk memecahkan permasalahan yang mirip seperti kasus tersebut.

Untuk memperjelas cara mengklasifikasikan data atau membuat sub keriteria maka dibuat contoh ke 3 (tiga), berikut merupakan contoh data yang digunakan untuk contoh ke 3 (tiga)

BAB 4

IMPLEMENTASI METODE ENTROPY PADA CODEIGNITER

Pada bab ini akan membahas

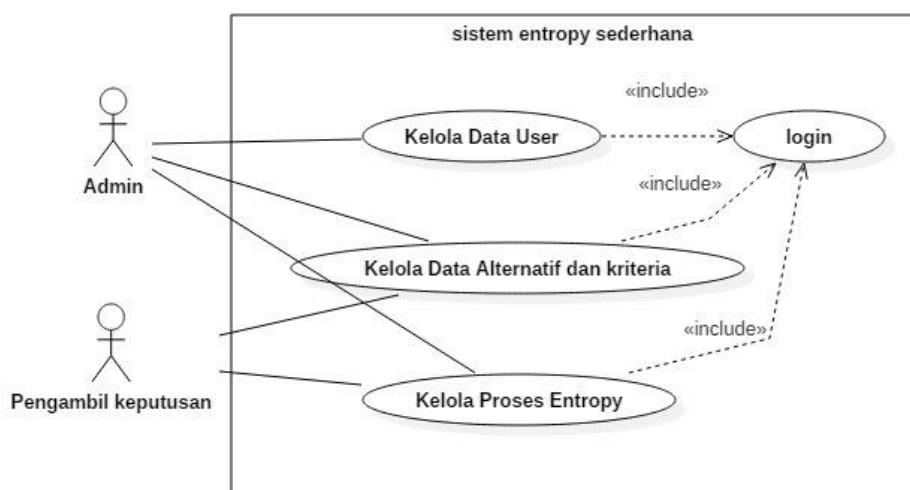
Pada bab ini akan membahas implementasi metode entropy pada sistem dengan meliputi perancangan sistem terdiri dari usecase diagram class diagram kemudian perancangan database dari sistem serta codinga implemtasi teori pada sistem

Perancangan Sistem

Adapun perancangan sistem pada buku ini yang di bahas yaitu perancangan usecase diagram class diagram dan perancangan basis data

Use Case Diagram

Pada perancangan sistem ini di butuhkan usecase diagram seperti pada gambar ..., dengan tujuan agar mengetahui peran dari aktor yang terlibat pada sistem, adapun aktor yang terdapat pada sistem ini yaitu aktor admin dan aktor pengambil keputusan



Gambar

Class Diagram

Kemudian setelah membuat usecase diagram dilanjutkan dengan membuat kelas diagram yang bertujuan untuk menunjukan class apasaja dan method apa saja yang digunakan pada sistem

Gambar

Perancangan Basisdata

Setelah class diagram di buat dilanjutkan dengan membuat perancangan basis data sistem adapun untuk basis data sistem bernama db_sistem.sql dengan ketentuan tabel-tabel yang terdapat pada basis data tersebut adalah sebagai berikut

Tabel user

<i>Field Name</i>	Tipe Data	<i>Field Size</i>	Keterangan
<i>user_id</i>	<i>Int</i>	11	<i>id user (primary key)</i>
<i>user_name</i>	<i>varchar</i>	20	<i>username user</i>
<i>user_email</i>	<i>varchar</i>	60	<i>Email user</i>
<i>user_password</i>	<i>varchar</i>	60	<i>Password user</i>
<i>user_level</i>	<i>varchar</i>	5	<i>Level user</i>
<i>status</i>	<i>Int</i>	1	<i>Staus user</i>

Tabel Data alternatif dan kriteria

<i>Field Name</i>	Tipe Data	<i>Field Size</i>	Keterangan
<i>alternatif_id</i>	<i>Int</i>	11	<i>id user (primary key)</i>
<i>alternatif_name</i>	<i>varchar</i>	20	<i>username user</i>
<i>Kriteria_1</i>	<i>Int</i>	11	<i>Email user</i>
<i>Kriteria_1</i>	<i>Int</i>	11	<i>Password user</i>
<i>Kriteria_1</i>	<i>Int</i>	11	<i>Level user</i>
<i>Kriteria_1</i>	<i>Int</i>	11	<i>Status user</i>
<i>Tipe_alternatif</i>	<i>varchar</i>	10	

Tabel Data Bobot Entropy

<i>Field Name</i>	Tipe Data	<i>Field Size</i>	Keterangan
<i>bobot_id</i>	<i>Int</i>	11	<i>id user (primary key)</i>
<i>Bobot_kriteria_1</i>	<i>varchar</i>	20	<i>username user</i>
<i>Bobot_kriteria_2</i>	<i>varchar</i>	60	<i>Email user</i>
<i>Bobot_kriteria_3</i>	<i>varchar</i>	60	<i>Password user</i>
<i>Bobot_kriteria_4</i>	<i>varchar</i>	3	<i>Level user</i>
<i>Tipe_alternatif</i>	<i>varchar</i>	10	<i>Status user</i>
<i>Id_user</i>			

Persiapan Implementasi Entropy

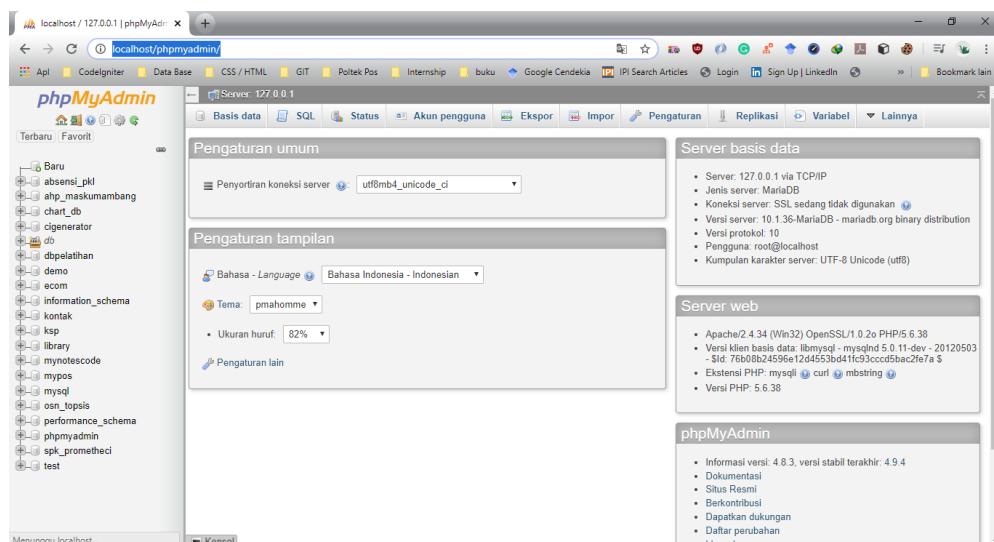
Untuk persiapan implementasi entropy diantaranya yaitu :

1. Codeigniter versi 3

2. Web server local (yang terinstall pada computer)
3. code editor (direkomendasikan menggunakan visual studio code dengan ketentuan seperti pada bab 1 buku ini)

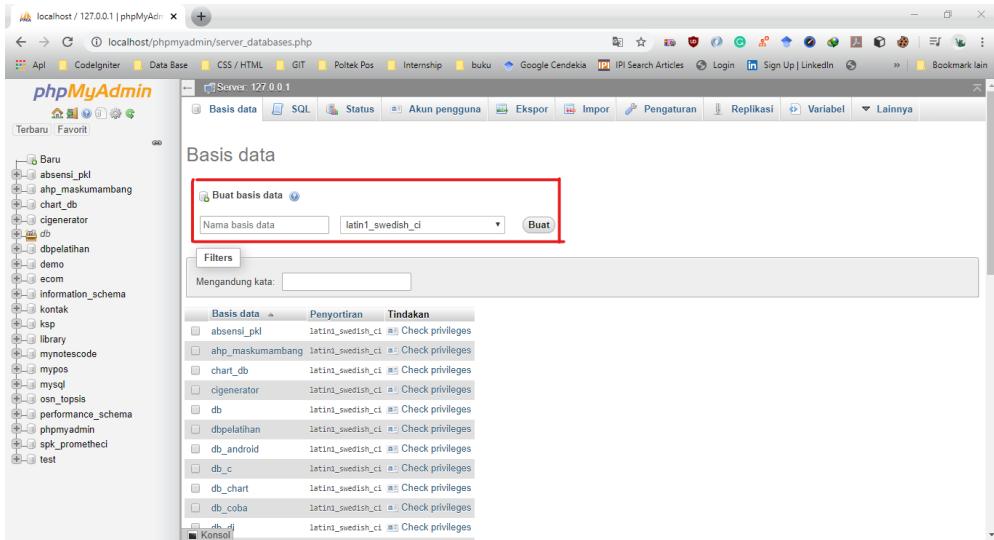
adapun tahapan tahapan implementasi ini yaitu buat terlebih dahulu databas sistem database tersebut di buat di phpmyadmin menggunakan mariadb, berikut merupakan langkah langkah untuk membuat database sistem.

1. nyalakan terlebih dahulu xampp dengan menyalakan dengan cara klik tombol start pada apache dan mysql.
2. masuk kehalaman phpmyadmin dengan cara memasukan alamat berikut <http://localhost/phpmyadmin/> maka tampilannya seperti berikut:



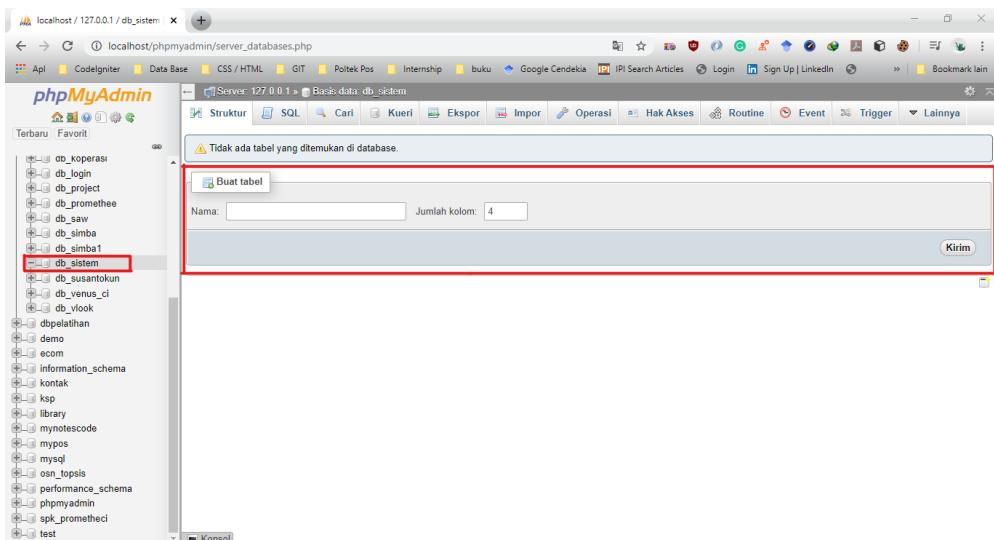
Gambar

Jika phpmyadmin di password isikan terlebih dahulu password dari phpmyadmin tersebut, selanjutnya jika menggunakan bahasa indonesia pilih menu Basis data sehingga tampilannya seperti berikut:



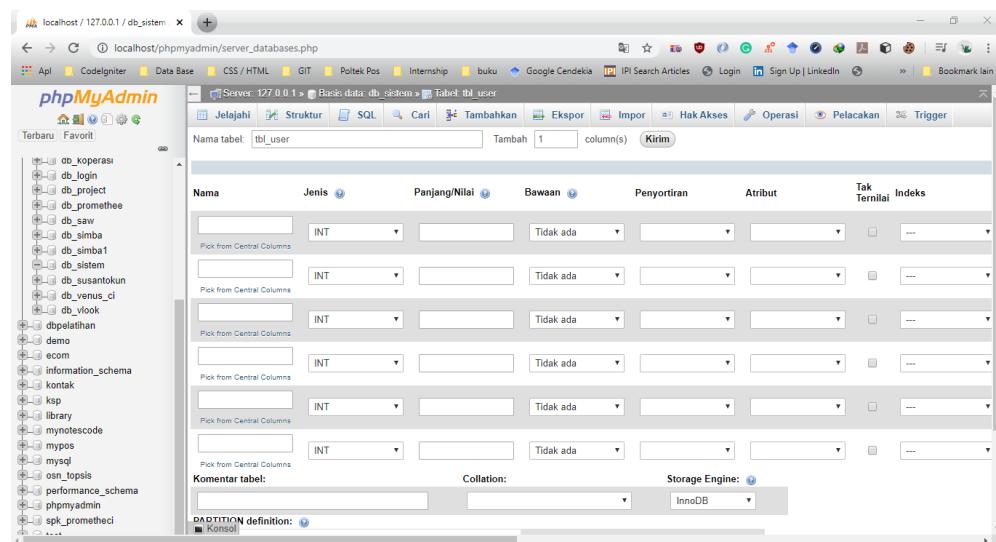
Gambar

Kemudian jika sudah tampil seperti pada gambar... buat basis data dengan mengisikan teks boks nama basis data dengan nama db_sistem kemudian klik tombol buat, untuk hasilnya maka akan seperti pada gambar ... berikut.



Gambar

Selanjutnya buat tabel pada basis data tersebut dengan nama `tbl_user` untuk ketentuan nama field dan tipe data dapat mengikuti ketentuan seperti pada rancangan data base untuk tabel user, untuk membuat tabel pada phpmyadmin isikan nama database pada teksbox seperti pada gambar ... tersebut kemudian pilih jumlah kolom atau field setelah seuai klik tombol kirim maka hasilnya seperti pada gambar ... berikut



Gambar ...

Setelah itu isi sesuai ketentuan yang ada pada tabel perancangan basis data sehingga hasilnya seperti pada gambar berikut ini

The screenshot shows the phpMyAdmin interface for creating a new table named 'tbl_user'. The table structure is defined as follows:

Nama	Jenis	Panjang/Nilai	Bawaan	Penyortiran	Atribut	Tak Ternilai	Indeks
user_id	INT		Tidak ada				PRIMARY
user_name	VARCHAR	30	Tidak ada				PRIMARY
user_email	VARCHAR	60	Tidak ada				
user_password	VARCHAR	60	Tidak ada				
user_level	VARCHAR	5	Tidak ada				
status	INT	1	Tidak ada				

Below the table structure, there are fields for 'Komentar tabel:', 'Collation:', and 'Storage Engine: InnoDB'. The 'PADA DEFINITION' section is also visible.

Gambar

Jika sudah seperti tampilan pada gambar selanjutnya skrok atau gulung tampilan ke kanan dan pilih A I (auto increment) setelah itu gulung tampilan ke bawah kemudian klik tombol simpan kemudian tunggu proses pembuatan tabel, jika telah selesai membuat tabel user maka hasilnya seperti pada gambar berikut:

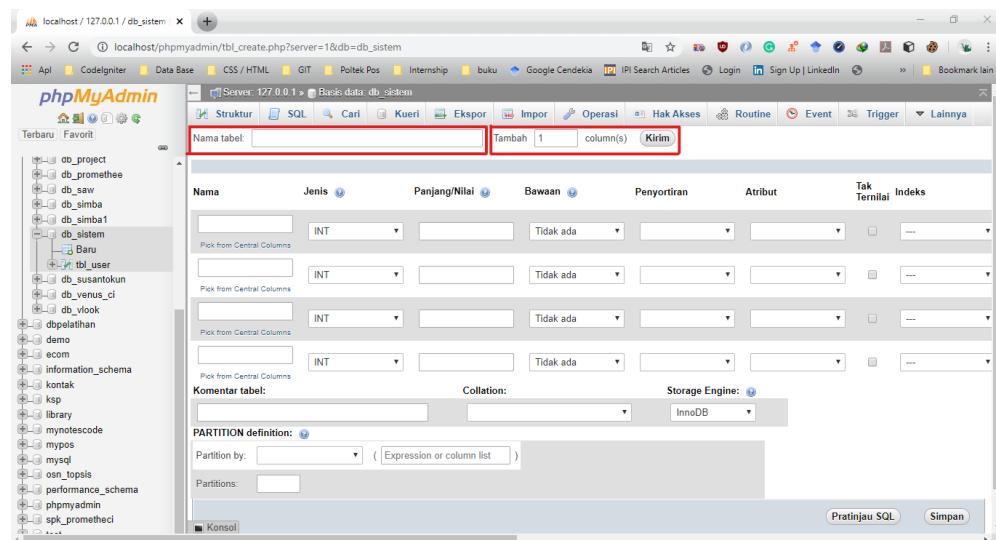
The screenshot shows the 'Struktur tabel' (Table Structure) page for the 'tbl_user' table. The table has the following structure:

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	user_id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
2	user_name	varchar(30)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
3	user_email	varchar(60)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
4	user_password	varchar(60)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
5	user_level	varchar(5)	latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
6	status	int(1)			Tidak	Tidak ada			Ubah Hapus Lainnya

Below the table structure, there are buttons for 'Pilih Semua', 'Dengan pilihan:', 'Jelajahi', 'Ubah', 'Hapus', 'Utama', 'Unik', 'Indeks', 'Teks penuh', and 'Add to central columns'. There are also buttons for 'Cetak', 'Usulkan struktur tabel', 'Lacak tabel', 'Move columns', and 'Normalisasi'. The 'Tambahkan' button is set to '1 kolom' and 'setelah status'. The 'Indeks' and 'Partitions' sections are also visible at the bottom.

Gambar

Jika tampilan sudah seperti pada gambar tersebut maka dilanjutkan dengan membuat tabel baru dengan cara klik menu baru yang terdapat pada sub db_sistem yang terdapat pada bagian kiri seperti pada gambar ... tersebut setelah di klik maka akan muncul tampilan seperti berikut:



Gambar

Kemudian jika telah muncul seperti pada gambar ... berikut berikan nama tabel pada teks box dengan nama `tbl_alternatif`, lalu jika jumlah kolom kurang bisa di tambahkan sesuai dengan kebutuhan kolom pada tabel, dikarenakan pada tabel ini membutuhkan tujuh kolom sehingga pada teks box tambah angka 1 di ubah menjadi 3 kemudian klik tombol kirim di sebelah kanannya maka kolom akan bertambah seperti pada gambar berikut:

The screenshot shows the phpMyAdmin interface for creating a new table named 'tbl_alternatif'. The table structure is defined as follows:

Nama	Jenis	Panjang/Nilai	Bawaan	Penyortiran	Atribut	Tak Ternilai	Indeks
alternatif	INT		Tidak ada				

Gambar

Kemudian jika tampilan sudah seperti pada gambar ... isi data setiap kolom seperti ketentuan pada tabel perancangan basis data jika sudah di isi maka tampilannya akan seperti pada gambar ... berikut:

The screenshot shows the phpMyAdmin interface for creating a new table named 'tbl_alternatif' with populated data. The table structure is defined as follows:

Nama	Jenis	Panjang/Nilai	Bawaan	Penyortiran	Atribut	Tak Ternilai	Indeks
alternatif_id	INT		Tidak ada				PRIMARY
alternatif_nama	VARCHAR	30	Tidak ada				PRIMARY
kriteria_1	INT		Tidak ada				
kriteria_2	INT		Tidak ada				
kriteria_3	INT		Tidak ada				
kriteria_4	INT		Tidak ada				
tipe_alternatif	VARCHAR	10	Tidak ada				

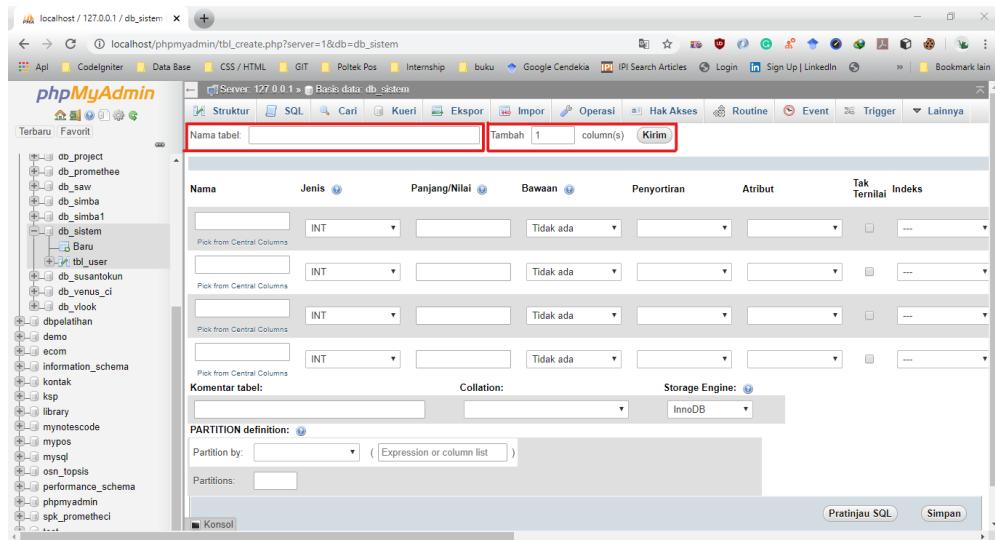
Gambar

Jika sudah seperti gambar ... maka selanjutnya gulung tampilan tersebut ke sebelah kanan kemudian celis check box A_I (autoincrement) untuk primary key tabel, setelah itu gulung tabel tersebut ke bawah kemudian klik tombol simpan, kemudian tunggu pembuatan tabel oleh sistem jika sudah selesai maka tampilannya akan seperti pada gambar berikut.

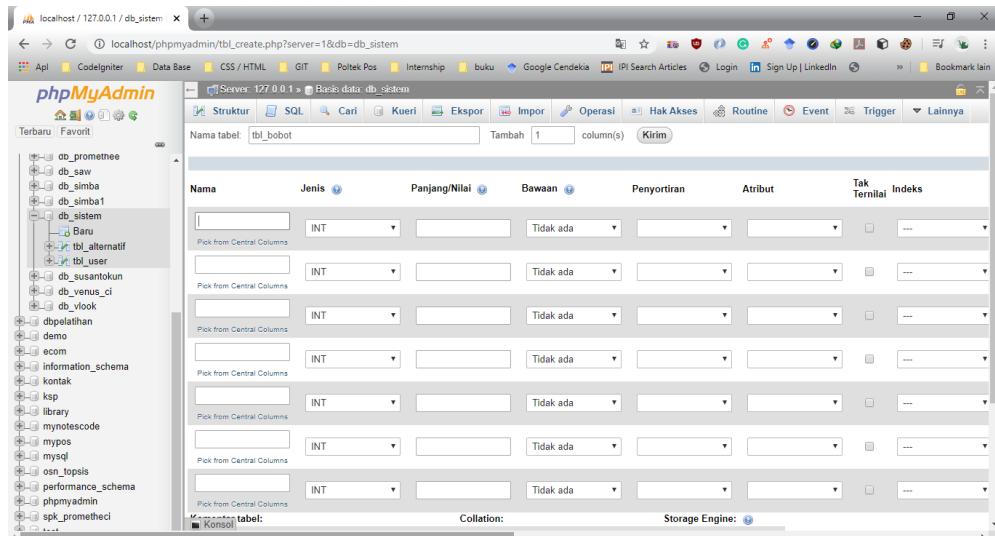
#	Nama	Jenis	Penyortiran	Atribut	Tak.Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	alternatif_id	int(11)			Tidak	Tidak ada	AUTO_INCREMENT	Ubah Hapus Lainnya	
2	alternatif_nama	varchar(30)	latin1_swedish_ci		Tidak	Tidak ada		Ubah Hapus Lainnya	
3	kriteria_1	int(11)			Tidak	Tidak ada		Ubah Hapus Lainnya	
4	kriteria_2	int(11)			Tidak	Tidak ada		Ubah Hapus Lainnya	
5	kriteria_3	int(11)			Tidak	Tidak ada		Ubah Hapus Lainnya	
6	kriteria_4	int(11)			Tidak	Tidak ada		Ubah Hapus Lainnya	
7	tipe_alternatif	varchar(10)	latin1_swedish_ci		Tidak	Tidak ada		Ubah Hapus Lainnya	

Gambar

Jika tabel alternatif telah dibuat maka selanjutnya buat tabel data bobot entropy dengan cara menekan sub database baru pada basis data db_sistem sehingga hasilnya seperti pada gambar berikut.

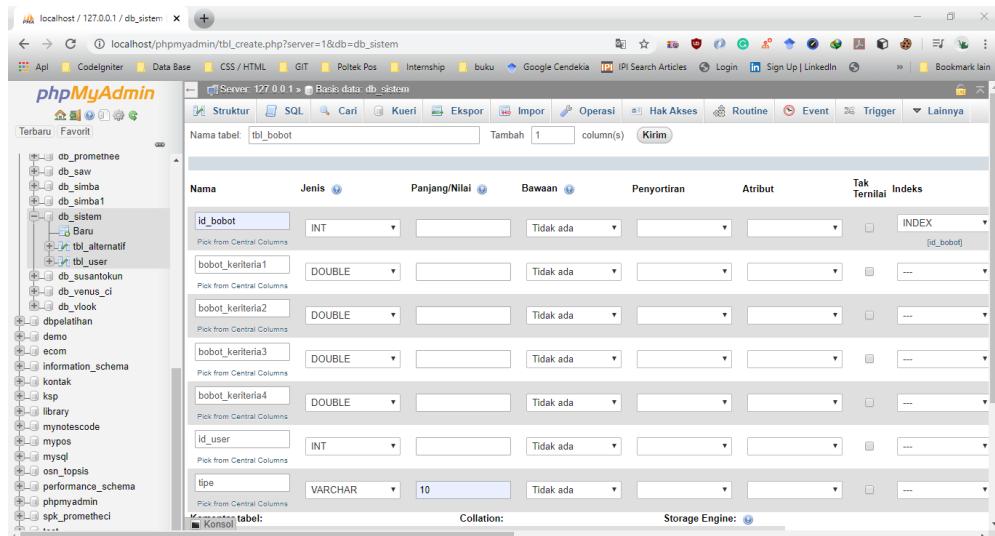


Krmudian tambah kolom tabel menjadi tujuh kolom caranya sama seperti memambahkan kolom pada tabael alternatif sehingga hasilnya seperti berikut:



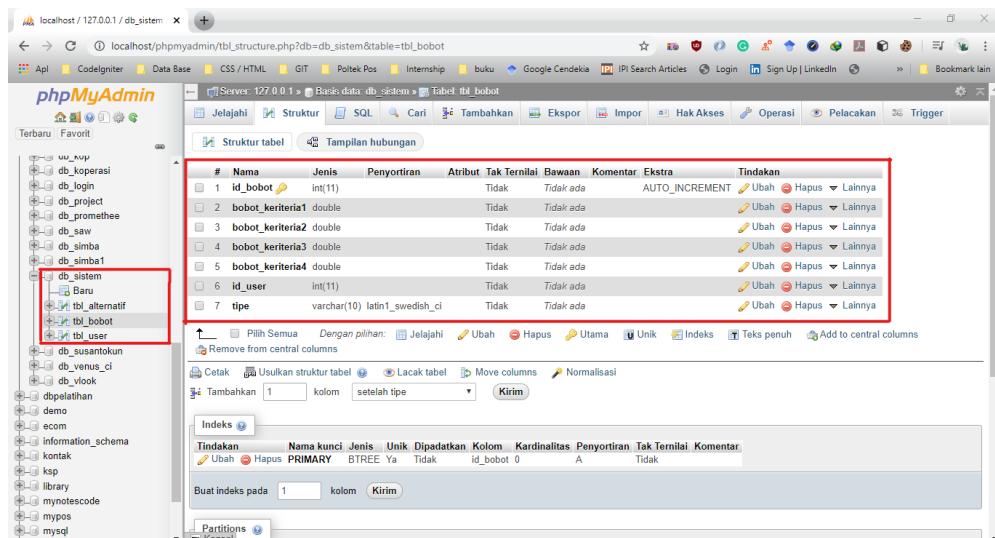
Gambar

Setelah itu isi kolom tersebut sesuai ketentuan pada rancangan bsis data sehingga hasilnya seperti berikut:



Gambar

Jika sudah seperti gambar ... maka selanjutnya gulung tampilan tersebut ke sebelah kanan kemudian celis check box A_I (autoincrement) untuk primary key tabel, setelah itu gulung tabel tersebut ke bawah kemudian klik tombol simpan, kemudian tunggu paembuatan tabel oleh sistem jika sudah selesai maka tampilannya akan seperti pada gambar berikut.



Gambar

Setelah selesai membuat database dilanjutkan dengan instalasi codeigniter yaitu dengan cara download terlebih dahulu codeigniter pada situs resminya. Hal ini dapat mengikuti langkah langkah pada bab1 namun untuk nama projeknya di ganti menjadi projekentropy sehingga pada htdoc tampilannya seperti berikut.

This PC > Local Disk (C:) > xampp > htdocs			
Name	Date modified	Type	Size
Coba_laravel	23/01/2020 19.28	File folder	
dashboard	18/12/2019 12.19	File folder	
enkripsi	19/12/2019 11.41	File folder	
entropy	26/01/2020 22.21	File folder	
garuda_crud_generator	19/01/2020 15.14	File folder	
Garuda-CRUD-Generator	19/01/2020 16.24	File folder	
hphp	20/01/2020 23.17	File folder	
img	18/12/2019 12.19	File folder	
mynpos	04/02/2020 23.27	File folder	
pelatihan	10/01/2020 02.15	File folder	
php	15/01/2020 11.13	File folder	
<input checked="" type="checkbox"/> projekentropy	07/02/2020 22.54	File folder	
promethee	15/01/2020 11.03	File folder	

Gambar

Setelah itu setting configurasi pada codeigniter yaitu pada file config.php kemudian setelah setting config dilanjutkan ke setting auto load yaitu pada file autoload.php setelah itu terakhir yaitu setting koneksi database yaitu pada file database.php, untuk lebih jelasnya berikut merupakan code yang diubah ayau dilakukan konfigurasi

Tempat direktori :

C:\xampp\htdocs\projekentropy\application\config\autoload.php

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

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

Tempat direktori :

C:\xampp\htdocs\projekentropy\application\config\config.php

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

Tempat direktori :

C:\xampp\htdocs\projekentropy\application\config\database.php

```
$active_group = 'default';
```

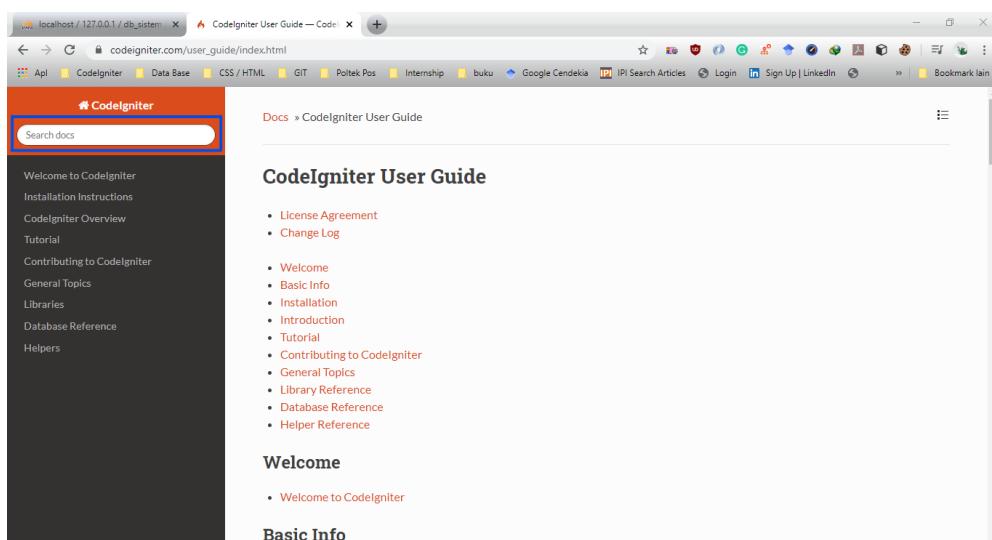
```
$query_builder = TRUE;
```

```
$db['default'] = array(
```

```
    'dsn'      => "",  
    'hostname' => 'localhost',  
    'username' => 'root',  
    'password' => "",  
    'database' => 'db_sistem',  
    '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
```

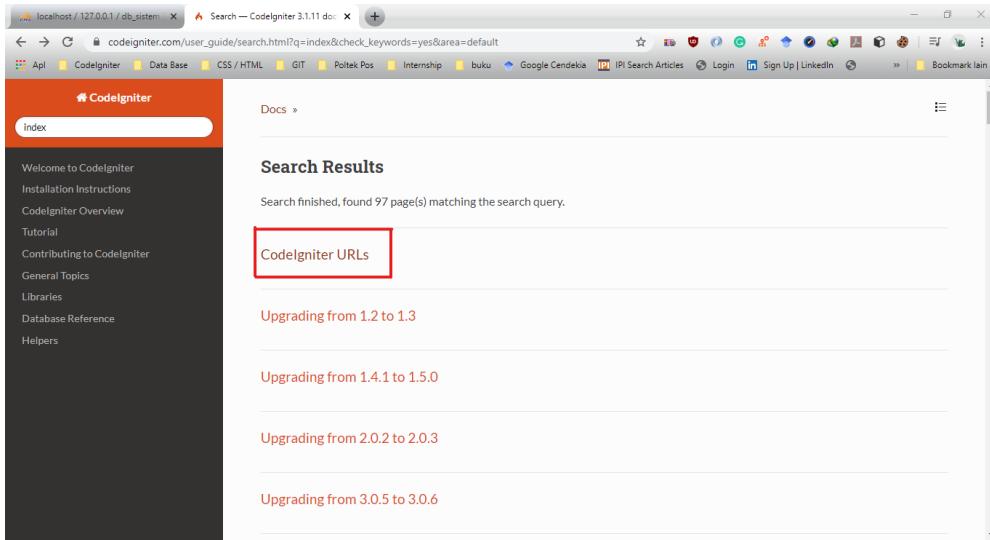
```
);
```

Kemudian pada direktori utama codeigniter akan di tambahkan file bernama .htaccess yang berguna untuk menghilangkan penulisan index.php pada alamat codeigniter, misalkan yang awalnya <http://codeigniter/index.php/controller/> menjadi <http://codeigniter/controller/> saja tetapi masih menghasilkan tampilan yang sama. Untuk menerapkan file .htaccess dapat dilakukan dengan cara mengakses dokumentasi codeigniter baik online maupun bawaan dari codeigniter itu sendiri untuk tampilan dari dokumentasi codeigniter sebagai berikut:

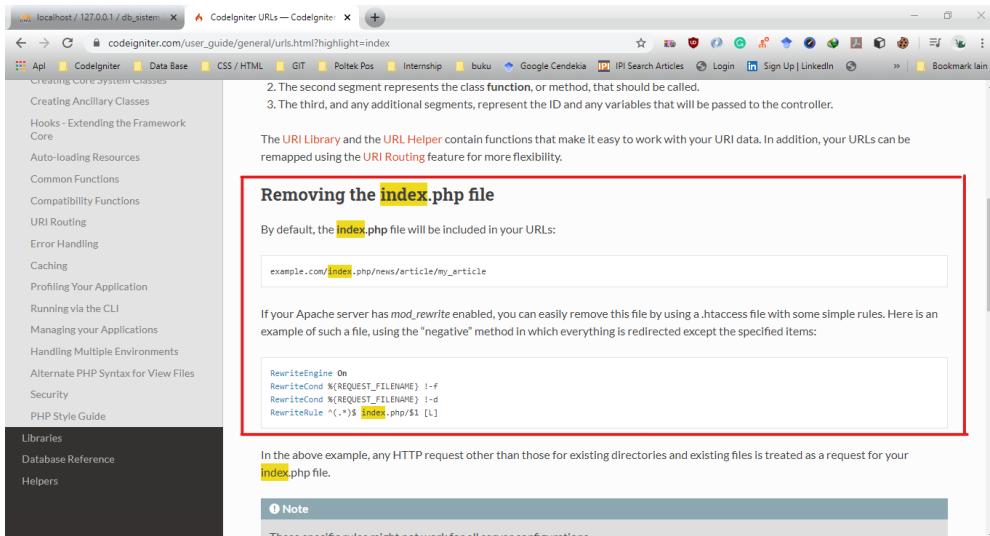


Gambar

Setelah mendapatkan tampilan seperti pada gambar... pada browser kemudian pada menu pencharian ketik index kemudian tekan enter, sehingga akan muncul tampilan seperti pada gambar berikut



Gambar



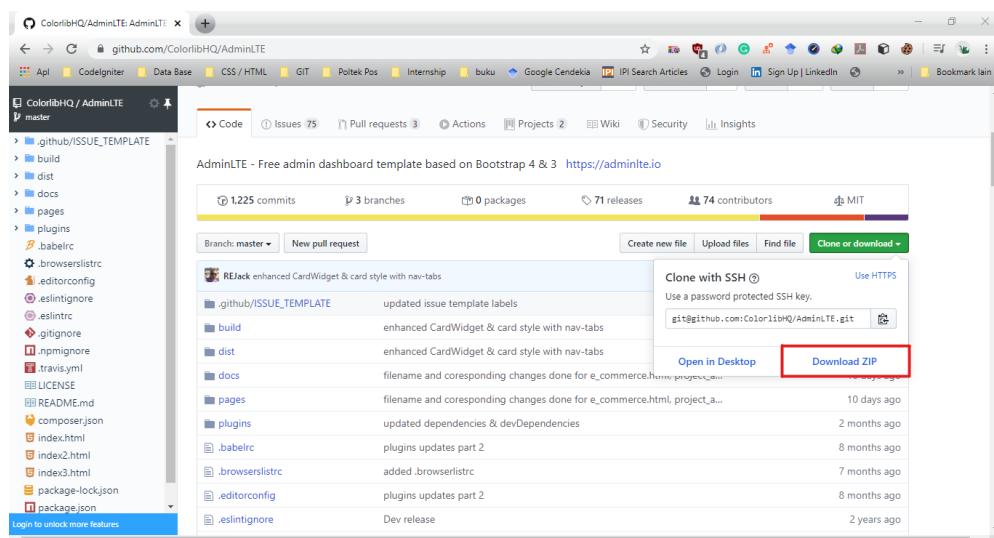
Gambar

Pada gambar berikut pilih CodeIgniter URLs kemudian gulung ke bawah dan carai codingan removing index.php setelah itu copy code tersebut dan masukan pada file .htaccess pada direktori utama codeigniter kemudian save, maka index.php pada codeigniter telah di hilangkan,

Penggunaan template

Pada impelemtasi ini menggunakan template Admin LTERyang dapat di download pada website resminya <https://adminlte.io/> atau pada alamat github berikut <https://github.com/ColorlibHQ/AdminLTE>

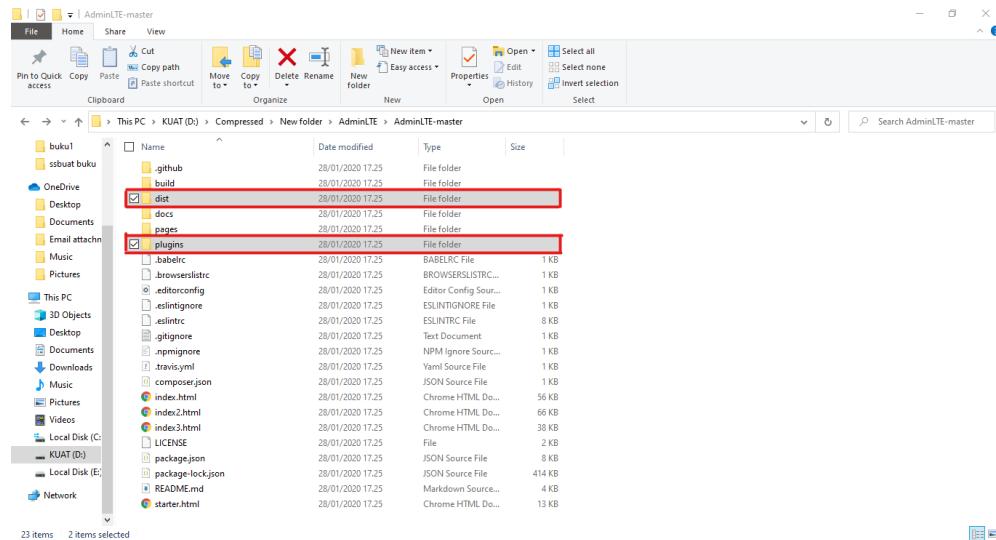
Jika menggunakan github alangkah baiknya menggunakan git scm sebagai tools untuk mendownload file atau projek dari github namun jika git scm tidak terinstall pada computer maka alternatifnya yaitu mendownload file zip dari github untuk cara download nya seperti berikut.



Gambar

Setelah file tempalate di unduh ekstrak terlebih dahulu file tersebut kemudian filih folder dist dan folder plugins kemudian buka folder dist lalu copy semu folder yang terdapat pada folder tersebut, kemudian untuk folder plugins bisa buka folder tersebut kemudian pilih data css dan js yang di gunakan atau kalau tidak mau susah bisa copy langsung folder plugins kemudian hasil copy folder tersebut paste ke folder resources yang terdapat

pada direktori projek_entropy. Untuk lebih jelasnya tempat kedua folder itu terdapat pada direktori utama AdminLTE-master seperti pada gambar berikut:



Gambar

Setelah di pastekan pada folder resource maka hasilnya seperti pada gambar berikut:

Name	Date modified	Type	Size
css	08/02/2020 01.15	File folder	
img	08/02/2020 01.26	File folder	
js	08/02/2020 01.15	File folder	
plugins	08/02/2020 01.16	File folder	

Gambar

Setelah itu buat folder layout pada folder views pada subdirektori application pada direktori projek_entropy kemudian isi dengan file main.php, lalu setelah file main.php telah di buat buka file index.html yang terdapat

pada direktori AdminLTE-master menggunakan visual studio code kemudian copy semua code yang terdapat pada file tersebut lalu pindahkan ke file main.php yang terdapat pada projek_entropy. Setelah code tersebut di pindahkan ada beberapa code yang harus diubah agar template tersebut dapat dijalankan menggunakan codeigniter berikut merupakan code yang harus di ubah

Code awal pada template

Bagian header

```
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>AdminLTE 3 | Dashboard</title>
    <!-- Tell the browser to be responsive to screen width -->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- Font Awesome -->
    <link rel="stylesheet" href="plugins/fontawesome-free/css/all.min.css">
    <!-- Ionicons -->
    <link rel="stylesheet"
        href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css"
    >
    <!-- Tempusdominus Bootstrap 4 -->
    <link rel="stylesheet" href="plugins/tempusdominus-bootstrap-
4/css/tempusdominus-bootstrap-4.min.css">
    <!-- iCheck -->
    <link rel="stylesheet" href="plugins/icheck-bootstrap/icheck-
bootstrap.min.css">
    <!-- JQVMap -->
```

```

<link rel="stylesheet" href="plugins/jqvmap/jqvmap.min.css">
<!-- Theme style -->
<link rel="stylesheet" href="dist/css/adminlte.min.css">
<!-- overlayScrollbars -->
<link rel="stylesheet"
      href="plugins/overlayScrollbars/css/OverlayScrollbars.min.css">
<!-- Daterange picker -->
<link rel="stylesheet" href="plugins/daterangepicker/daterangepicker.css">
<!-- summernote -->
<link rel="stylesheet" href="plugins/summernote/summernote-bs4.css">
<!-- Google Font: Source Sans Pro -->
<link
      href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700" rel="stylesheet">
</head>

```

Bahian footer

```

<script src="plugins/jquery/jquery.min.js"></script>
<!-- jQuery UI 1.11.4 -->
<script src="plugins/jquery-ui/jquery-ui.min.js"></script>
<!-- Resolve conflict in jQuery UI tooltip with Bootstrap tooltip -->
<script>
  $.widget.bridge('uibutton', $.ui.button)
</script>
<!-- Bootstrap 4 -->
<script src="plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
<!-- ChartJS -->
<script src="plugins/chart.js/Chart.min.js"></script>

```

```
<!-- Sparkline -->
<script src="plugins/sparklines/sparkline.js"></script>
<!-- JQVMap -->
<script src="plugins/jqvmap/jquery.vmap.min.js"></script>
<script src="plugins/jqvmap/maps/jquery.vmap.usa.js"></script>
<!-- jQuery Knob Chart -->
<script src="plugins/jquery-knob/jquery.knob.min.js"></script>
<!-- daterangepicker -->
<script src="plugins/moment/moment.min.js"></script>
<script src="plugins/daterangepicker/daterangepicker.js"></script>
<!-- Tempusdominus Bootstrap 4 -->
<script src="plugins/tempusdominus-bootstrap-4/js/tempusdominus-
bootstrap-4.min.js"></script>
<!-- Summernote -->
<script src="plugins/summernote/summernote-bs4.min.js"></script>
<!-- overlayScrollbars -->
<script
src="plugins/overlayScrollbars/js/jquery.overlayScrollbars.min.js"></script>
<!-- AdminLTE App -->
<script src="dist/js/adminlte.js"></script>
<!-- AdminLTE dashboard demo (This is only for demo purposes) -->
<script src="dist/js/pages/dashboard.js"></script>
<!-- AdminLTE for demo purposes -->
<script src="dist/js/demo.js"></script>
```

Code yang di ubah dan di gunakan pada codeigniter

Pada header menjadi:

```
<head>

<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title>AdminLTE 3 | Dashboard</title>
<!-- Tell the browser to be responsive to screen width -->
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- Font Awesome -->
<link rel="stylesheet" href=<?php echo
site_url('resources/plugins/fontawesome-free/css/all.min.css'); ?>">
<!-- Ionicons -->
<link rel="stylesheet"
href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css"
>
<!-- Tempusdominus Bbootstrap 4 -->
<link rel="stylesheet" href=<?php echo
site_url('resources/plugins/tempusdominus-bootstrap-4/css/tempusdominus-
bootstrap-4.min.css'); ?>">
<!-- iCheck -->
<link rel="stylesheet" href=<?php echo
site_url('resources/plugins/icheck-bootstrap/icheck-bootstrap.min.css'); ?>">
<!-- JQVMap -->
<link rel="stylesheet" href=<?php echo
site_url('resources/plugins/jqvmap/jqvmap.min.css'); ?>">
<!-- Theme style -->
<link rel="stylesheet" href=<?php echo
site_url('resources/css/adminlte.min.css'); ?>">
<!-- overlayScrollbars -->
```

```

<link rel="stylesheet" href="<?php echo
site_url('resources/plugins/overlayScrollbars/css/OverlayScrollbars.min.css');
?>">

<!-- Daterange picker -->

<link rel="stylesheet" href="<?php echo
site_url('resources/plugins/daterangepicker/daterangepicker.css'); ?>">

<!-- summernote -->

<link rel="stylesheet" href="<?php echo
site_url('resources/plugins/summernote/summernote-bs4.css'); ?>">

<!-- Google Font: Source Sans Pro -->

<link
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,4
00i,700" rel="stylesheet">

</head>

```

Pada bagian javascript atau bagian akhir template Menjadi:

```

<script src="<?php echo site_url('resources/plugins/jquery/jquery.min.js');
?>"></script>

<!-- jQuery UI 1.11.4 -->

<script src="<?php echo site_url('resources/plugins/jquery-ui/jquery-
ui.min.js'); ?>"></script>

<!-- Resolve conflict in jQuery UI tooltip with Bootstrap tooltip -->

<script>
    $.widget.bridge('uibutton', $.ui.button)
</script>

<!-- Bootstrap 4 -->

<script src="<?php echo
site_url('resources/plugins/bootstrap/js/bootstrap.bundle.min.js');
?>"></script>

<!-- ChartJS -->

<script src="<?php echo site_url('resources/plugins/chart.js/Chart.min.js');
```

```
?>"></script>

<!-- Sparkline -->

<script src="php echo site_url('resources/plugins/sparklines/sparkline.js'); ?&gt;""></script>

<!-- JQVMap -->

<script src="php echo site_url('resources/plugins/jqvmap/jquery.vmap.min.js'); ?&gt;""></script>

<script src="php echo site_url('resources/plugins/jqvmap/maps/jquery.vmap.usa.js'); ?&gt;""></script>

<!-- jQuery Knob Chart -->

<script src="php echo site_url('resources/plugins/jquery-knob/jquery.knob.min.js'); ?&gt;""></script>

<!-- daterangepicker -->

<script src="php echo site_url('resources/plugins/moment/moment.min.js'); ?&gt;""></script>

<script src="php echo site_url('resources/plugins/daterangepicker/daterangepicker.js'); ?&gt;""></script>

<!-- Tempusdominus Bootstrap 4 -->

<script src="php echo site_url('resources/plugins/tempusdominus-bootstrap-4/js/tempusdominus-bootstrap-4.min.js'); ?&gt;""></script>

<!-- Summernote -->

<script src="php echo site_url('resources/plugins/summernote/summernote-bs4.min.js'); ?&gt;""></script>

<!-- overlayScrollbars -->

<script src="php echo site_url('resources/plugins/overlayScrollbars/js/jquery.overlayScrollbars.min.js'); ?&gt;""></script>

<!-- AdminLTE App -->

<script src="php echo site_url('resources/js/adminlte.js'); ?&gt;""></script>
```

```
<!-- AdminLTE dashboard demo (This is only for demo purposes) -->
<script src=<?php echo site_url('resources/js/pages/dashboard.js'); ?>></script>
<!-- AdminLTE for demo purposes -->
<script src=<?php echo site_url('resources/js/demo.js'); ?>></script>
```

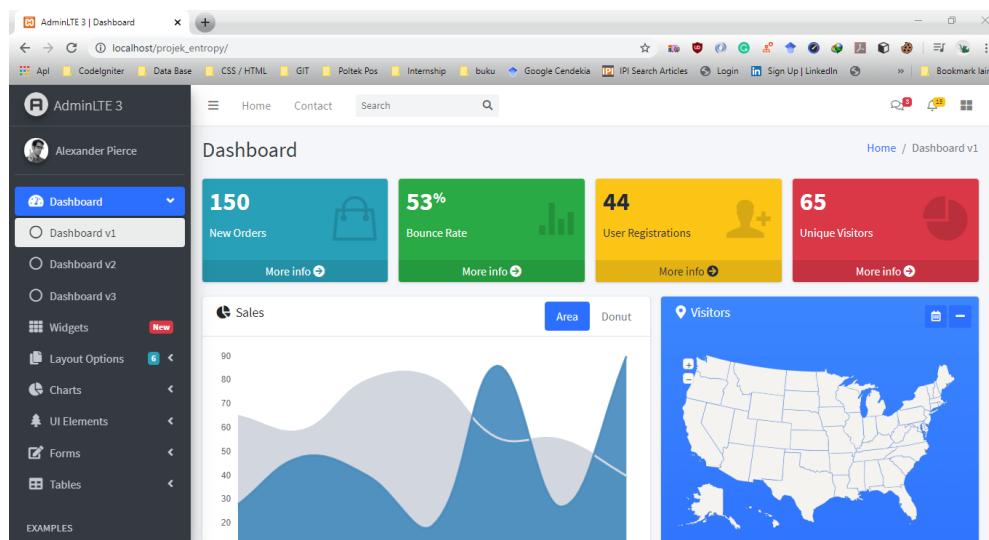
Kemudian code untuk gambar yang semula

```
img src="dist/img/nama gambar dan ekstensinya"
```

Menjadi

```
img src=<?php echo site_url('resources/img/nama gambar dan ekstensinya); ?>"
```

Sehingga hasil dari penggunaan template pada codeigniter seperti pada gambar berikut:



Gambar

Untuk fitur pada template tersebut di sesuaikan dengan kebutuhan sistem sehingga fitur-fitur yang terdapat pada gambar ... tersebut belum tentu di gunakan semua untuk projek ini.

Implementasi Program

Setelah membuat database, setting konfigurasi codeigniter kemudian penerapan template pada codeigniter, dilanjutkan dengan implementasi program, dimana perogram ini disisipkan metode atau algoritma entropy pada bagian sub modulnya, akan tetapi dikarenakan data untuk dilakukannya metode entropy harus di olah terlebi dahulu maka pada program ini dilengkapi dengan:

Fitu CRUD (ceate, read, update, dan delete)

Fitur template yang telah di bahas pada sub bab sebelumnya

Fitur login session multi user

Terdiri dari dua user (user admin dan user pengambil keputusan)

Fitur entropy

Terdapat enkripsi data untuk password

Dalam implementasi Program ini perlu di buat file-file dan folder terlebi dahulu pada subdirektori codeigniter yang terdapat pada controller, model, view, serta library.

Berikut merupakan file PHP yang dibuat pada sub direktori Controller

Local Disk (C:) > xampp > htdocs > projek_entropy > application > controllers

<input type="checkbox"/>	Name	Date modified	Type	Size
	Dashboard.php	10/02/2020 01.01	PHP Source File	1 KB
	index.html	19/09/2019 19.08	Chrome HTML Do...	1 KB
	Login.php	10/02/2020 01.00	PHP Source File	2 KB
	Proses_entropy.php	09/02/2020 23.36	PHP Source File	3 KB
	Tbl_alternatif.php	07/02/2020 15.50	PHP Source File	5 KB
	Tbl_bobot.php	09/02/2020 23.39	PHP Source File	5 KB
	Tbl_user.php	10/02/2020 01.55	PHP Source File	4 KB

Catatan : kosongkan terlebi dahulu file-file tersebut

Kemudian setelah membuat file PHP pada direktori application buat file PHP pada sub direktori models dengan nama file seperti pada gambar berikut

Local Disk (C:) > xampp > htdocs > projek_entropy > application > models

<input type="checkbox"/>	Name	Date modified	Type	Size
	Entropy_model.php	08/02/2020 23.02	PHP Source File	1 KB
	index.html	19/09/2019 19.08	Chrome HTML Do...	1 KB
	Login_model.php	08/02/2020 03.43	PHP Source File	1 KB
	Tbl_alternatif_model.php	07/02/2020 15.50	PHP Source File	2 KB
	Tbl_bobot_model.php	09/02/2020 23.37	PHP Source File	2 KB
	Tbl_user_model.php	07/02/2020 15.50	PHP Source File	2 KB

Catatan : kosongkan terlebi dahulu file-file tersebut

Kemudian setelah membuat file PHP pada direktori models buat file PHP pada sub direktori libraries dengan nama file seperti pada gambar berikut

Local Disk (C:) > xampp > htdocs > projek_entropy > application > libraries

<input type="checkbox"/>	Name	Date modified	Type	Size
	enkripsi.php	10/02/2020 00.30	PHP Source File	1 KB
	index.html	19/09/2019 19.08	Chrome HTML Do...	1 KB

Catatan : kosongkan terlebih dahulu file-file tersebut

Kemudian setelah itu pada sub direktori view buat terlebih dahulu 5 folder untuk memisahkan view dari setiap controller selain itu penempatan file juga akan menjadi lebih tertata. Adapun 5 folder itu yaitu:

Folder layout

Folder entropy

Folder tbl_alternatif

Folder tbl_bobot

Folder tbl_user

Selain kelima folder tersebut pada subdirektori views buat dua file PHP yaitu file dashboard.php dan login.php

Untuk lebihjelasnya dapat dilihat pada gambar berikut:

Local Disk (C:) > xampp > htdocs > projek_entropy > application > views				
<input type="checkbox"/> Name	Date modified	Type	Size	
entropy	08/02/2020 23.49	File folder		
errors	08/02/2020 01.15	File folder		
layouts	08/02/2020 01.15	File folder		
tbl_alternatif	08/02/2020 04.29	File folder		
tbl_bobot	09/02/2020 23.20	File folder		
tbl_user	08/02/2020 04.29	File folder		
dashboard.php	10/02/2020 00.59	PHP Source File	4 KB	
index.html	19/09/2019 19.08	Chrome HTML Do...	1 KB	
login.php	09/02/2020 20.52	PHP Source File	7 KB	

Untuk folder errors dan file index.html merupakan bawaan dari codeigniter, kemudian setelah kelima folder tersebut telah dibuat didalam

folder tersebut buat file PHP untuk folder entropy buat dua file yaitu hasil.php dan index.php

Jelasnya seperti pada gambar tersebut

Local Disk (C:) > xampp > htdocs > projek_entropy > application > views > entropy				
<input type="checkbox"/> Name	Date modified	Type	Size	
hasil.php	09/02/2020 23.44	PHP Source File	6 KB	
index.php	09/02/2020 23.49	PHP Source File	2 KB	

Kemudian untuk folder layouts buat satu file php yakni main.php

Jelasnya seperti gambar berikut

Local Disk (C:) > xampp > htdocs > projek_entropy > application > views > layouts				
<input type="checkbox"/> Name	Date modified	Type	Size	
main.php	10/02/2020 01.53	PHP Source File	12 KB	

Untuk folder tbl_alternatif buat tiga file php terdiri dari add.php , edit.php dan index.php, jelasnya seperti pada gambar berikut:

Local Disk (C:) > xampp > htdocs > projek_entropy > application > views > tbl_alternatif				
<input type="checkbox"/> Name	Date modified	Type	Size	
add.php	09/02/2020 23.50	PHP Source File	3 KB	
edit.php	09/02/2020 23.52	PHP Source File	4 KB	
index.php	09/02/2020 23.51	PHP Source File	4 KB	

Untuk folder tbl_bobot buat tiga file php terdiri dari add.php , edit.php dan index.php, jelasnya seperti pada gambar berikut:

Local Disk (C:) > xampp > htdocs > projek_entropy > application > views > tbl_bobot

Name	Date modified	Type	Size
add.php	10/02/2020 00.04	PHP Source File	3 KB
edit.php	10/02/2020 00.01	PHP Source File	4 KB
index.php	10/02/2020 00.17	PHP Source File	4 KB

Untuk folder tbl_user buat tiga file php terdiri dari add.php , edit.php dan index.php, jelasnya seperti pada gambar berikut:

Local Disk (C:) > xampp > htdocs > projek_entropy > application > views > tbl_user

Name	Date modified	Type	Size
add.php	09/02/2020 23.57	PHP Source File	3 KB
edit.php	10/02/2020 00.44	PHP Source File	3 KB
index.php	10/02/2020 00.49	PHP Source File	4 KB

Catatan : untuk semua file tersebut di kosongkan terlebih dahulu

Tahap selanjutnya yaitu memulai coding untuk tampilan dashboard, buka file Dashboard.php pada controller kemudian masukan codeberikut.

```
<?php  
defined('BASEPATH') or exit('No direct script access allowed');  
  
class Dashboard extends CI_Controller  
{  
    public function index()  
    {  
        if ($this->session->userdata('user_level') === 'admin') {  
            $data['_view'] = 'dashboard';
```

```
        $this->load->view('layouts/main', $data);

    } elseif ($this->session->userdata('user_level') === 'user') {

        $data['_view'] = 'dashboard';

        $this->load->view('layouts/main', $data);

    } else {

        echo "Access Denied";

        redirect('');

    }

}
```

Pada code tersebut nama class harus sama dengan nama file karena pada file bernama Dashboard maka class yang terdapat pada folder tersebut bernama Dashboard. Kemudian pada code tersebut bertujuan untuk menampilkan halaman dashboard untuk setiap user. Sebagai pembeda tampilan yang di tampilkan yaitu menggunakan parameter session, parameter tersebut di buat terlebih dahulu pada file login, kemudian parameter tersebut memiliki nilai berupa string yaitu admin dan user kemudian di bandingkan dengan nilai string pada code jika sesuai maka akan menampilkan dashboard sesuai levelnya.

Selanjutnya buka folder main.php pada folder layouts kemudian isikan code tersebut

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```

<title>AdminLTE 3 | Dashboard</title>

<!-- Tell the browser to be responsive to screen width -->

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

<!-- Font Awesome -->

<link rel="stylesheet" href="php echo site_url('resources/plugins/fontawesome-free/css/all.min.css'); ?&gt;"&gt;

&lt!-- Ionicons --&gt;

&lt;link rel="stylesheet" href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css" &gt;

&lt!-- Tempusdominus Bootstrap 4 --&gt;

&lt;link rel="stylesheet" href="<?php echo site_url('resources/plugins/tempusdominus-bootstrap-4/css/tempusdominus-bootstrap-4.min.css'); ?&gt;"&gt;

&lt!-- iCheck --&gt;

&lt;link rel="stylesheet" href="<?php echo site_url('resources/plugins/icheck-bootstrap/icheck-bootstrap.min.css'); ?&gt;"&gt;

&lt!-- JQVMap --&gt;

&lt;link rel="stylesheet" href="<?php echo site_url('resources/plugins/jqvmap/jqvmap.min.css'); ?&gt;"&gt;

&lt!-- Theme style --&gt;

&lt;link rel="stylesheet" href="<?php echo site_url('resources/css/adminlte.min.css'); ?&gt;"&gt;

&lt!-- overlayScrollbars --&gt;

&lt;link rel="stylesheet" href="<?php echo site_url('resources/plugins/overlayScrollbars/css/OverlayScrollbars.min.css'); ?&gt;"&gt;

&lt!-- Daterangepicker --&gt;

&lt;link rel="stylesheet" href="<?php echo site_url('resources/plugins/daterangepicker/daterangepicker.css'); ?&gt;"&gt;
</pre

```

```

<!-- summernote -->

<link rel="stylesheet" href="php echo site_url('resources/plugins/summernote/summernote-bs4.css'); ?&gt;"" />

<link rel="stylesheet" href="php echo site_url('resources/plugins/datatables-bs4/css/dataTables.bootstrap4.css'); ?&gt;"" />

<!-- Google Font: Source Sans Pro -->

<link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700" rel="stylesheet">

</head>

<body class="hold-transition sidebar-mini layout-fixed">

<div class="wrapper">

    <!-- Navbar -->

    <nav class="main-header navbar navbar-expand navbar-white navbar-light">

        <!-- Left navbar links -->

        <ul class="navbar-nav">
            <li class="nav-item">
                <a class="nav-link" data-widget="pushmenu" href="#"><i class="fas fa-bars"></i></a>
            </li>
        </ul>

        <!-- Right navbar links -->

        <ul class="navbar-nav ml-auto">
            <li class="nav-item">
                <a class="nav-link" data-widget="control-sidebar" data-

```

```

slide="true" href="#">>

    <i class="fas fa-th-large"></i>
</a>
</li>
</ul>
</nav>

<!-- /.navbar -->
<!-- Main Sidebar Container -->
<aside class="main-sidebar sidebar-dark-primary elevation-4">
    <!-- Brand Logo -->
    <a href="index3.html" class="brand-link">
        
        <span class="brand-text font-weight-light">Aplikasi Entropy</span>
    </a>

    <!-- Sidebar -->
    <div class="sidebar">
        <!-- Sidebar user panel (optional) -->
        <div class="user-panel mt-3 pb-3 mb-3 d-flex">
            <div class="image">
                
            </div>
            <div class="info">
                <a href="#" class="d-block"><?= $this->session->

```

```
>userdata('user_name'); ?></a>
    </div>
</div>
<?php if ($this->session->userdata('user_level') === 'admin') { ?>
<nav class="mt-2">
    <ul class="nav nav-pills nav-sidebar flex-column" data-
widget="treeview" role="menu" data-accordion="false">

        <a href="<?= site_url('welcome/index'); ?>" class="nav-
link active">
            <i class="nav-icon fas fa-tachometer-alt"></i>
            <p>
                Dashboard
            </p>
        </a>

        <li class="nav-item">
            <a href="<?php echo site_url('tbl_user'); ?>" class="nav-
link">
                <i class="nav-icon fas fa-user-circle"></i>
                <p>
                    Data User
                </p>
            </a>
        </li>
        <li class="nav-item has-treeview">
            <a href="<?php echo site_url('tbl_alternatif'); ?>" class="nav-link">
```

```
<i class="nav-icon fas fa-file-alt"></i>
<p>
    Data Alternatif
</p>
</a>
</li>
<li class="nav-item has-treeview">
    <a href=<?php echo site_url('proses_entropy'); ?>">
        <i class="nav-icon fas fa-edit"></i>
        <p>
            Proses Entropy
        </p>
        </a>
    </li>
    <li class="nav-item has-treeview">
        <a href=<?php echo site_url('login/logout'); ?>">
            <i class="nav-icon fas fa-sign-out-alt"></i>
            <p>
                Sign Out
            </p>
            </a>
        </li>
    </ul>
</nav>
<?php } elseif ($this->session->userdata('user_level') === 'user') {
```

```

?>

<nav class="mt-2">

    <ul class="nav nav-pills nav-sidebar flex-column" data-
widget="treeview" role="menu" data-accordion="false">

        <a href="= site_url('welcome/index'); ?" class="nav-
link active">

            <i class="nav-icon fas fa-tachometer-alt"></i>

            <p>
                Dashboard
            </p>

            </a>

        <li class="nav-item has-treeview">

            <a href="= site_url('tbl_alternatif'); ?" class="nav-link">

                <i class="nav-icon fas fa-file-alt"></i>

                <p>
                    Data Alternatif
                </p>

                </a>

            </li>

            <li class="nav-item has-treeview">

                <a href="= site_url('proses_entropy'); ?" class="nav-link">

                    <i class="nav-icon fas fa-edit"></i>

                    <p>
                        Proses Entropy
                    </p>
                </a>
            </li>
        </ul>
    </nav>

```

```
</a>
</li>
<li class="nav-item has-treeview">
    <a href="php echo site_url('login/logout'); ?&gt;"&gt;
        &lt;i class="nav-icon fas fa-sign-out-alt"&gt;&lt;/i&gt;
        &lt;p&gt;
            Sign Out
        &lt;/p&gt;
        &lt;/a&gt;
    &lt;/li&gt;
&lt;/ul&gt;
&lt;/nav&gt;
&lt;?php } ?&gt;
&lt;!-- Sidebar Menu --&gt;

&lt;!-- /.sidebar-menu --&gt;
&lt;/div&gt;
&lt;!-- /.sidebar --&gt;
&lt;/aside&gt;
&lt;!-- Content Wrapper. Contains page content --&gt;
&lt;div class="content-wrapper"&gt;
    &lt;!-- Content Header (Page header) --&gt;
    &lt;?php
        if (isset($_view) &amp;&amp; $_view)
            $this-&gt;load-&gt;view($_view);
    ?&gt;</pre
```

```
<!-- /.content -->
</div>

<!-- /.content-wrapper -->
<footer class="main-footer">
    <strong>Copyright © 2020</strong>
    All rights reserved.

    <div class="float-right d-none d-sm-inline-block">
        <b>Version</b> 1.0
    </div>
</footer>

<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>

<!-- ./wrapper -->

<!-- jQuery -->
<script src=<?php echo site_url('resources/plugins/jquery/jquery.min.js'); ?>"></script>

<!-- jQuery UI 1.11.4 -->
<script src=<?php echo site_url('resources/plugins/jquery-ui/jquery-ui.min.js'); ?>"></script>

<!-- Resolve conflict in jQuery UI tooltip with Bootstrap tooltip -->
<script>
```

```

$.widget.bridge('uibutton', $.ui.button)
</script>

<!-- Bootstrap 4 -->
<script src="php echo site_url('resources/plugins/bootstrap/js/bootstrap.bundle.min.js'); ?&gt;"&gt;&lt;/script&gt;

&lt;!-- ChartJS --&gt;
&lt;script src="<?php echo site_url('resources/plugins/chart.js/Chart.min.js'); ?&gt;"&gt;&lt;/script&gt;

&lt;!-- Sparkline --&gt;
&lt;script src="<?php echo site_url('resources/plugins/sparklines/sparkline.js'); ?&gt;"&gt;&lt;/script&gt;

&lt;!-- JQVMap --&gt;
&lt;script src="<?php echo site_url('resources/plugins/jqvmap/jquery.vmap.min.js'); ?&gt;"&gt;&lt;/script&gt;
&lt;script src="<?php echo site_url('resources/plugins/jqvmap/maps/jquery.vmap.usa.js'); ?&gt;"&gt;&lt;/script&gt;

&lt;!-- jQuery Knob Chart --&gt;
&lt;script src="<?php echo site_url('resources/plugins/jquery-knob/jquery.knob.min.js'); ?&gt;"&gt;&lt;/script&gt;

&lt;!-- daterangepicker --&gt;
&lt;script src="<?php echo site_url('resources/plugins/moment/moment.min.js'); ?&gt;"&gt;&lt;/script&gt;
&lt;script src="<?php echo site_url('resources/plugins/daterangepicker/daterangepicker.js'); ?&gt;"&gt;&lt;/script&gt;

&lt;!-- Tempusdominus Bootstrap 4 --&gt;
&lt;script src="<?php echo site_url('resources/plugins/tempusdominus-bootstrap-4/js/tempusdominus-bootstrap-4.min.js'); ?&gt;"&gt;&lt;/script&gt;

&lt;!-- Summernote --&gt;
&lt;script src="<?php echo</pre

```

```
site_url('resources/plugins/summernote/summernote-bs4.min.js');
?>"></script>

<!-- overlayScrollbars -->

<script src="php echo site_url('resources/plugins/overlayScrollbars/js/jquery.overlayScrollbars.min.js'); ?&gt;"&gt;&lt;/script&gt;

&lt;!-- AdminLTE App --&gt;

&lt;script src="<?php echo site_url('resources/js/adminlte.js'); ?&gt;"&gt;&lt;/script&gt;

&lt;!-- AdminLTE dashboard demo (This is only for demo purposes) --&gt;

&lt;script src="<?php echo site_url('resources/js/pages/dashboard.js'); ?&gt;"&gt;&lt;/script&gt;

&lt;!-- AdminLTE for demo purposes --&gt;

&lt;script src="<?php echo site_url('resources/js/demo.js'); ?&gt;"&gt;&lt;/script&gt;

&lt;script src="<?php echo site_url('resources/plugins/datatables/jquery.dataTables.js'); ?&gt;"&gt;&lt;/script&gt;

&lt;script src="<?php echo site_url('resources/plugins/datatables-bs4/js/dataTables.bootstrap4.js'); ?&gt;"&gt;&lt;/script&gt;

&lt;script&gt;
$(function() {
    $("#example1").DataTable();
    $('#example2').DataTable({
        "paging": true,
        "lengthChange": false,
        "searching": false,
        "ordering": true,
        "info": true,</pre
```

```
        "autoWidth": false,  
    });  
});  
</script>  
</body>  
  
</html>
```

Code tersebut merupakan dasar untuk penggunaan template, pada file tersebut didalamnya terdapat link yang mengambil atau memanggil library css dan java script yang di gunakan pada projek ini, sehingga file ini merupakan dasar untuk tampilan keseluruhan projek ini, jika file ini tidakjalan maka template yang di gunakan tidak akan berjalan dengan semestinya atau bahkan tidak jalan sama sekali.

Pada code tersebut juga dilakukan pembagian navbar menggunakan session, hal ini dilakukan dikarenakan tampilan untuk user admin dan user pengambil keputusan berbeda, hal yang sama juga dilakukan pada file dashboard pada subdirektori views konten di dalamnya di bagi menggunakan klausia if dengan parameter session user level di karenakan tampilan dashboard untuk user admin dan user pengambil keputusan berbeda, berikut merupakan code untuk tampilan dashboard untuk user admin dan user pengambil keputusan.

```
<div class="content-header">
    <div class="container-fluid">
        <div class="row mb-2">
            <div class="col-sm-6">
                <h1 class="m-0 text-dark">Dashboard</h1>
            </div><!-- /.col -->
        </div><!-- /.row -->
    </div><!-- /.container-fluid -->
</div>

<!-- Main content -->
<section class="content">
    <div class="container-fluid">
        <!-- Small boxes (Stat box) -->
        <?php if ($this->session->userdata('user_level') === 'admin') { ?>
            <div class="row">
                <div class="col-lg-4 col-6">
                    <!-- small box -->
                    <div class="small-box bg-info">
                        <div class="inner">
                            <h3></h3>
                            <p>Data Alternatif</p>
                        </div>
                        <div class="icon">
```

```
<i class="ion ion-document"></i>

</div>

<a href="#"><?=base_url('tbl_alternatif') ?>" class="small-box-footer">More info <i class="fas fa-arrow-circle-right"></i></a>

</div>

</div>

<!-- ./col -->

<div class="col-lg-4 col-6">

<!-- small box -->

<div class="small-box bg-success">

<div class="inner">

<h3></h3>

<p>Data Bobot</p>

</div>

<div class="icon">

<i class="ion ion-stats-bars"></i>

</div>

<a href="#"><?=base_url('tbl_bobot') ?>" class="small-box-footer">More info <i class="fas fa-arrow-circle-right"></i></a>

</div>

</div>

<!-- ./col -->

<div class="col-lg-4 col-6">
```

```

<!-- small box -->
<div class="small-box bg-warning">
    <div class="inner">
        <h3></h3>

        <p>Data User</p>
    </div>
    <div class="icon">
        <i class="ion ion-person-add"></i>
    </div>
    <a href="#"><?=base_url('tbl_user'); ?>" class="small-box-footer">More info <i class="fas fa-arrow-circle-right"></i></a>
    </div>
</div>
<!-- ./col -->
</div>
<?php } elseif ($this->session->userdata('user_level') === 'user') { ?>
<div class="row">
    <div class="col-lg-6 col-6">
        <!-- small box -->
        <div class="small-box bg-info">
            <div class="inner">
                <h3></h3>

                <p>Data Alternatif</p>
            </div>
        </div>
    </div>
</div>

```

```
</div>
<div class="icon">
    <i class="ion ion-document"></i>
</div>
<a href="#"><?=base_url('tbl_alternatif') ?>" class="small-box-footer">More info <i class="fas fa-arrow-circle-right"></i></a>
</div>
</div>
<!-- ./col -->
<div class="col-lg-6 col-6">
    <!-- small box -->
    <div class="small-box bg-success">
        <div class="inner">
            <h3></h3>

            <p>Data Bobot</p>
        </div>
        <div class="icon">
            <i class="ion ion-stats-bars"></i>
        </div>
        <a href="#"><?=base_url('tbl_bobot') ?>" class="small-box-footer">More info <i class="fas fa-arrow-circle-right"></i></a>
    </div>
</div>
```

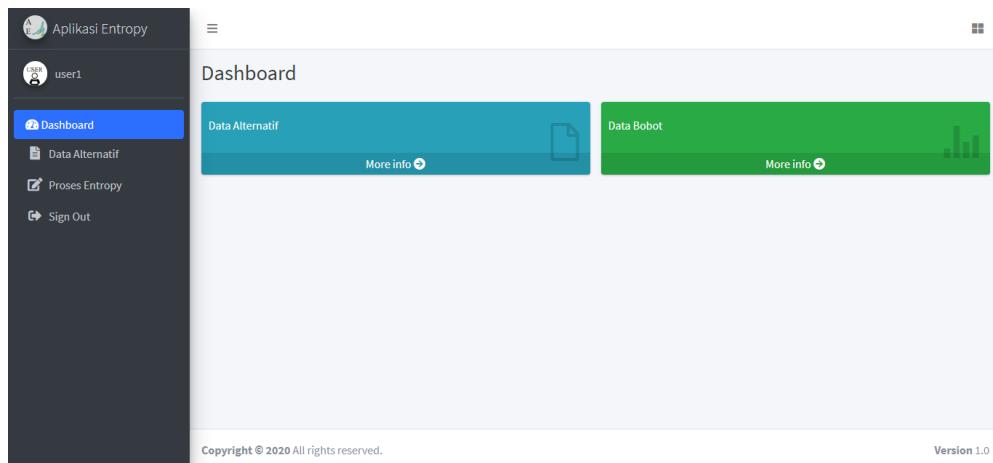
```

<!-- ./col -->
</div>

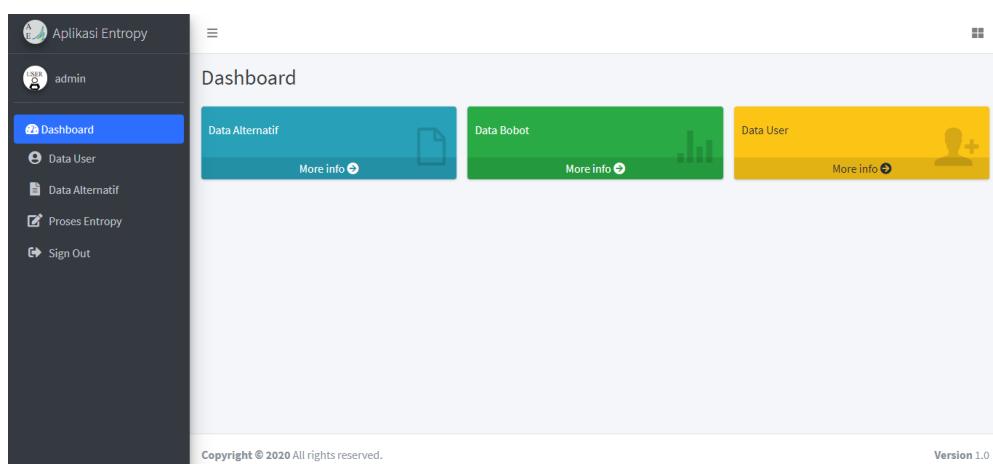
<?php } ?>
</div><!-- /.container-fluid -->
</section>

```

Jika code tersebut telah di pindahkan ke file dashboard pada sub direktory views maka hasil jadinya seperti berikut:



Tampilan dashboard untuk user pemilih keputusan



Tampilan untuk user admin

Kedua tampilan tersebut akan muncul jika user telah melakukan login pada sistem, maka dari itu buat terlebih dahulu fungsi login untuk sistem, buka file Login pada controller kemudian masukan code berikut:

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

class Login extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->model('login_model');
        $this->load->library('enkripsi');
    }

    public function index()
    {
        $this->form_validation->set_rules('username', 'username', 'required');
        $this->form_validation->set_rules('password', 'Password',
'required|min_length[4]');
        if ($this->form_validation->run() == FALSE) {

            $this->load->view('login');
        } else {
            $username = $this->input->post('username', TRUE);
            $password = $this->enkripsi->encryptIt($this->input->post('password', TRUE));
            $validasi = $this->login_model->validate($username, $password);
            if ($validasi->num_rows() > 0) {
                $data = $validasi->row();
                $id = $data->user_id;
                $level = $data->user_level;
                $user_name = $data->user_name;
                $sesiondata = array(
                    'user_id' => $id,
                    'user_level' => $level,
                    'user_name' => $user_name,
                    'logged_in' => TRUE
                )
            }
        }
    }
}
```

```

);
$this->session->set_userdata($sesiondata);

if ($level === 'admin') {
    redirect('dashboard');
} elseif ($level === 'user') {
    redirect('dashboard');
}
} else {
    echo $this->session->set_flashdata('msg', 'Username or Password
is Wrong');
    redirect("");
}
}

function logout()
{
    $this->session->sess_destroy();
    redirect('Login');
}
}

```

Pada code tersebut terdapat fungsi construktor, fungsi index dan fungsi logout yang sedangkan method untuk login itu sendiri terdapat pada fungsi index, pada fungsi construktor terdapat code untuk meload atau memanggil class yang terdapat pada model dan class yang terdapat pada library kedua class tersebut di simpan pada construktor agar dapat digunakan pada semua fungsi yang terdapat pada class tersebut, kemudian untuk method logout atau keluar dari sistem terdapat pada fungsi logout.

Pada fungsi index itu sendiri merupakan fungsi yang menangkapmasukan dari view login kemudian membandingkan data yang dikirimkan ke model yang bersangkutan lalu pada model data tersebut ditangkap oleh fungsi yang bersangkutan kemudian di bandingkan dengan data yang terdapat pada database jika data ada maka dikirim lagi ke controller kemudian pada controller data tersebut dijadikan parameter untuk membuat session, kemudian data session tersebut di jadikan parameter untuk

perbandingan user level jika user level tersebut sesuai dan sama dengan perbandingnya maka fungsi indeks akan menjalankan fungsi index yang terdapat pada file Dashboard pada controller, setelah code Login controller di buat buka file Login_model.php pada sub direktori models kemudian isikan code berikut:

```
<?php  
class Login_model extends CI_Model  
{  
  
    function validate($email, $password)  
    {  
        $this->db->where('user_name', $email);  
        $this->db->where('user_password', $password);  
        $result = $this->db->get('tbl_user');  
        return $result;  
    }  
}
```

Fungsi atau method validate merupakan method yang menangkap data yang di kirimkan oleh fungsi index yang terdapat pada file Login, setelah itu buka file enkripsi yang terdapat pada subdirektori Libraries kemudian pindahkan code berikut pada file enkripsi.php

```
<?php  
class Enkripsi  
{  
    public function encryptIt($password)  
    {
```

```

$cryptKey = '1212';
$qEncoded = base64_encode(mcrypt_encrypt(MCRYPT_RIJNDAEL_256,
md5($cryptKey), $password, MCRYPT_MODE_CBC,
md5(md5($cryptKey))));

return ($qEncoded);
}

public function decryptIt($password)
{
$cryptKey = '1212';

$qDecoded = rtrim(mcrypt_decrypt(MCRYPT_RIJNDAEL_256,
md5($cryptKey), base64_decode($password), MCRYPT_MODE_CBC,
md5(md5($cryptKey))), "\0");

return ($qDecoded);
}
}

```

Pada code enkripsi tersebut terdapat dua fungsi yaitu fungsi encryptIt dan fingsi decryptIt, dimana fungsi encryptIt digunakan untuk mengenkripsi data password sehingga password yang di simpan pada database telah di enkripsi sehingga berbentuk kode kemudian untuk mengurai kode tersebut (dekrripsi) menggunakan fungsi decryptIt. Kedua fungsi atau kelas ini di gunakan agar user dapat melakukan login, kemudian fungsi ini juga di gunakan untuk mengelola password dari user.

Setelah code pada file Login.php, Login_model.php, dan Enkripsi.php dibuat, buka file login yang terdapat pada view kemudian masukan code berikut yang bertujuan untuk membuat tampilan form login pada sistem ini berikut merupakan code untuk view login.php

```

<!DOCTYPE html>
<html>

<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title>AdminLTE 3 | Validation Form</title>
<!-- Tell the browser to be responsive to screen width -->
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- Font Awesome -->
<link rel="stylesheet" href="<?php echo site_url('resources/plugins/fontawesome-free/css/all.min.css'); ?>">
<!-- Ionicons -->
<link href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css" rel="stylesheet">
<!-- Theme style -->
<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhO f23Q9Ifjh" crossorigin="anonymous">
<!-- Google Font: Source Sans Pro -->
<link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700" rel="stylesheet">
</head>

<body class="hold-transition sidebar-mini">
<div class="wrapper">
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
<!-- Content Header (Page header) -->
<section class="content-header">
<div class="container-fluid">
<div class="row mb-2">
<div class="col-sm-6 offset-lg-3">
<center>
<h1>Sistem Entropy</h1>
</center>
</div>
</div>

```

```

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

<!-- Main content -->

<section class="content">
  <div class="container-fluid">
    <div class="row">
      <!-- left column -->
      <div class="col-lg-6 offset-lg-3">
        <!-- jquery validation -->
        <div class="card card-primary">
          <div class="card-header">
            <h3 class="card-title">Login Sistem</h3>
          </div>
          <!-- /.card-header -->
          <!-- form start -->
          <form      role="form"      id="quickForm"      action="">
method="post">
            <div class="card-body">
              <div class="form-group">
                <label
for="exampleInputUsername1">username</label>
                <input      type="text"      name="username"
class="form-control"      id="exampleInputUsername1"      placeholder="Enter
username">
                </div>
                <div class="form-group">
                  <label
for="exampleInputPassword1">Password</label>
                  <input      type="password"      name="password"
class="form-control"                      id="exampleInputPassword1"
placeholder="Password">
                  </div>
                </div>
                <!-- /.card-body -->
                <div class="card-footer">
                  <button class="btn btn-primary" value="Login"
name="kirim" type="submit">Sign in
                  <i class="nav-icon fas fa-sign-in-alt"></i>
                </button>
              </div>
            </div>
          </form>
        </div>
      </div>
    </div>
  </div>
</section>

```

```

        </form>
    </div>
    <!-- /.card -->
</div>
<!-- .col (left) -->
<!-- right column -->
<div class="col-md-6">

    </div>
    <!-- .col (right) -->
</div>
<!-- /.row -->
</div><!-- /.container-fluid -->
</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->
</div>
<!-- ./wrapper -->

<!-- jQuery -->
<script src=<?php echo site_url('resources/plugins/jquery/jquery.min.js'); ?>"></script>
<!-- Bootstrap 4 -->
<script src=<?php echo site_url('resources/plugins/bootstrap/js/bootstrap.bundle.min.js'); ?>"></script>
<!-- jquery-validation -->
<!-- jquery-validation -->
<script src=<?php echo site_url('resources/plugins/jquery-validation/jquery-validate.min.js'); ?>"></script>
<script src=<?php echo site_url('resources/plugins/jquery-validation/additional-methods.min.js'); ?>"></script>
<script type="text/javascript">
$(document).ready(function() {
    $.validator.setDefaults({
        submitHandler: function() {
            alert("Form successful submitted!");
        }
    });
    $('#quickForm').validate({
        rules: {

```

```

username: {
    required: true,
    username: true,
},
password: {
    required: true,
    minlength: 5
},
messages: {
    username: {
        username: "Please enter a userame address",
        email: "Please enter a vaild userame address"
    },
    password: {
        required: "Please provide a password",
        minlength: "Your password must be at least 5 characters
long"
    },
},
errorElement: 'span',
errorPlacement: function(error, element) {
    error.addClass('invalid-feedback');
    element.closest('.form-group').append(error);
},
highlight: function(element, errorClass, validClass) {
    $(element).addClass('is-invalid');
},
unhighlight: function(element, errorClass, validClass) {
    $(element).removeClass('is-invalid');
}
});
});
</script>
</body>

</html>

```

Untuk code login mrnggunakan tag html lengkap dikarenakan base css dari adminLTE yang kurang mendukung tampilan login sehingga untuk tampilan login mengambil base dari css bootstrap secara langsung, namun

untuk javascript pada tampilan login tetap menggunakan base dari adminLTE kemudian untuk tampilan login sendiri memiliki fitur JQuery sehingga ketika data kurang atau tidak di isi pada form login akan memunculkan peringatan. Berikut merupakan tampilan halaman Login pada sistem ini

Sistem Entropy

Login Sistem

username	<input type="text" value="Enter username"/>
Password	<input type="password" value="Password"/>
<input type="button" value="Sign in ➔"/>	

Login Sistem

Setelah membuat menu login dan dashboard untuk ke dua userl level dilanjutkan ke peroses CRUD data alternatif, kenapa harus CRUD data alternatif ?. itu karena data yang akan di bobotkan merupakan data kriteria yang terdapat pada data alternatif sehingga data alternatif yang dahulu diolah atau di siapkan. Maka dari itu untuk menyiapkan data alternatif perlu proses crud untuk data alternatif untuk melakukan peroses crud tentunya di butuhkan controller, model, dan view berikut merupakan code controller untuk file Tbl_alternatif.php

```
<?php  
class Tbl_alternatif extends CI_Controller{  
    function __construct()  
    {  
        parent::__construct();  
        $this->load->model('Tbl_alternatif_model');  
    }  
}
```

Pada file controller Tbl_alternatif terdapat fungsi __construct yang merupakan konstruktor dari class tersebut, kemudian pada fungsi tersebut terdapat parent::__construct(); yang merupakan induk dari code yang terdapat pada class tersebut maksud parren di sini jika ada model, helper atau library dan lain sebagainya di deklarasikan pada fungsi tersebut berarti fungsi atau class yang telah di deklarasikan tersebut dapat di gunakan pada fungsi lainnya yang terdapat pada class tersebut.

```
function index()  
{  
    $data['tbl_alternatif'] = $this->Tbl_alternatif_model-  
>get_all_tbl_alternatif();  
  
    $data['_view'] = 'tbl_alternatif/index';  
    $this->load->view('layouts/main',$data);  
}
```

Kemudian setelah fungsi atau method __construct dilanjutkan dengan fungsi index, fungsi ini merupakan fungsi yang pertama dijalankan jika controller dengan class Tbl_alternatif di panggil pada alamat tanpa di sertai fungsi yang lainnya, maka otomatis fungsi index yang akan jalan.

Pada fungsi index tersebut terdapat variabel data yang di dalamnya terdapat array assosiatif yang merupakan parameter untuk mengirim data pada view kemudian di sertai dengan data dari model Tbl_alternatif_model

yang memanggil fungsi `get_all_tbl_alternatif` yang terdapat pada model tersebut. Kemudian untuk array assosiatif memiliki nilai 'tbl_alternatif/index' yang merupakan nama folder dan file dari view yang akan ditampilkan. Untuk hasil tampilan dari code tersebut sebagai berikut:

Alternatif Id	Alternatif Nama	Kriteria 1	Kriteria 2	Kriteria 3	Kriteria 4	Tipe Alternatif	Actions
1	Handphone1	1500	20	12	32	AA10	<button>Edit</button> <button>Delete</button>
3	Handphone2	1000	25	7	16	AA10	<button>Edit</button> <button>Delete</button>

Showing 1 to 2 of 2 entries

Copyright © 2020 All rights reserved. Version 1.0

```
function add()
{
    $this->load->library('form_validation');

    $this->form_validation-
>set_rules('alternatif_nama','Alternatif Nama','required|max_length[30]');
    $this->form_validation->set_rules('kriteria_1','Kriteria
1','required|integer');
    $this->form_validation->set_rules('kriteria_2','Kriteria
2','required|integer');
    $this->form_validation->set_rules('kriteria_3','Kriteria
3','required|integer');
    $this->form_validation->set_rules('kriteria_4','Kriteria
4','required|integer');
    $this->form_validation->set_rules('tipe_alternatif','Tipe
Alternatif','required|max_length[10]');

    if($this->form_validation->run())
    {
        $params = array(
            'Alternatif Nama' => $this->input->post('Alternatif Nama'),
            'Kriteria 1' => $this->input->post('kriteria_1'),
            'Kriteria 2' => $this->input->post('kriteria_2'),
            'Kriteria 3' => $this->input->post('kriteria_3'),
            'Kriteria 4' => $this->input->post('kriteria_4'),
            'Tipe Alternatif' => $this->input->post('tipe_alternatif')
        );
        $this->model->insert($params);
        $this->session->setFlashdata('pesan', 'Data berhasil
ditambahkan');
        redirect('Alternatif');
    }
}
```

```

'alternatif_nama' => $this->input-
>post('alternatif_nama'),
'kriteria_1' => $this->input->post('kriteria_1'),
'kriteria_2' => $this->input->post('kriteria_2'),
'kriteria_3' => $this->input->post('kriteria_3'),
'kriteria_4' => $this->input->post('kriteria_4'),
'tipe_alternatif' => $this->input-
>post('tipe_alternatif'),
);

$tbl_alternatif_id = $this->Tbl_alternatif_model-
>add_tbl_alternatif($params);
redirect('tbl_alternatif/index');
}
else
{
$data['_view'] = 'tbl_alternatif/add';
$this->load->view('layouts/main',$data);
}
}
}

```

Setelah fungsi index pada class tersebut dilanjutkan dengan fungsi add yang di gunakan untuk menambahkan data, pada fungsi ini menggunakan klausa if dan else sehingga jika tidak ada data yang dikirim pada fungsi tersebut maka akan menjalankan telebih dahulu view dari form untuk menambahkan data. Kemudian jika data yang di kirim melalui form tambah data yang di tujuhan pada fungsi add tersebut kemudian sesuai dengan aturan form validation maka data akan diterima menggunakan method post dan di tampung pada stau variabel yang berisikan array kemudian variabel tersebut dijadikan parameter untuk menjalankan fungsi add_tbl_alternatif yang terdapat pada model Tbl_alternatif_model, lalu jika dtu tersebut berhasil di simpan maka akan pindah ke halaman index dari controller tersebut .

```

function edit($alternatif_id)
{
    // check if the tbl_alternatif exists before trying to edit it
    $data['tbl_alternatif'] = $this->Tbl_alternatif_model-
>get_tbl_alternatif($alternatif_id);

    if(isset($data['tbl_alternatif']['alternatif_id']))
    {
        $this->load->library('form_validation');

        $this->form_validation-
>set_rules('alternatif_nama','Alternatif Nama','required|max_length[30]');
        $this->form_validation->set_rules('kriteria_1','Kriteria
1','required|integer');
        $this->form_validation->set_rules('kriteria_2','Kriteria
2','required|integer');
        $this->form_validation->set_rules('kriteria_3','Kriteria
3','required|integer');
        $this->form_validation->set_rules('kriteria_4','Kriteria
4','required|integer');
        $this->form_validation-
>set_rules('tipe_alternatif','Tipe Alternatif','required|max_length[10]');

        if($this->form_validation->run())
        {
            $params = array(
                'alternatif_nama' => $this->input-
>post('alternatif_nama'),
                'kriteria_1' => $this->input-
>post('kriteria_1'),
                'kriteria_2' => $this->input-
>post('kriteria_2'),
                'kriteria_3' => $this->input-
>post('kriteria_3'),
                'kriteria_4' => $this->input-
>post('kriteria_4'),
                'tipe_alternatif' => $this->input-
>post('tipe_alternatif'),
            );
        }

        $this->Tbl_alternatif_model-
>update_tbl_alternatif($alternatif_id,$params);
    }
}

```

```

        redirect('tbl_alternatif/index');
    }
else
{
    $data['_view'] = 'tbl_alternatif/edit';
    $this->load->view('layouts/main',$data);
}
else
    show_error('The tbl_alternatif you are trying to edit does not exist.');
}

```

Setelah fungsi add dilanjutkan dengan fungsi edit, pada fungsi ini pertama menangkap parameter berupa id dari tabel, biasnya parameter tersebut di sisipkan pada tombol, kemudian parameter tersebut digunakan untuk mengecek apakah tabel dengan parameter id tersebut ada atau tidak jika tidak ada maka akan muncul error, kemudian jika data tersebut ada maka data tersebut akan dikirim ke form sehingga pada form yang akan di edit terdapat data yang akan di edit. Kemudian jika data yang di edit dikirim melalui form akan dilakukan form validation seperti pada fungsi add hanya saja ada parameter id dan variabel yang menampung data sebagai parameter untuk menjalankan fungsi edit yang terdapat pada model.

```

function remove($alternatif_id)
{
    $tbl_alternatif = $this->Tbl_alternatif_model-
>get_tbl_alternatif($alternatif_id);

    // check if the tbl_alternatif exists before trying to delete it
    if(isset($tbl_alternatif['alternatif_id']))
    {
        $this->Tbl_alternatif_model->delete_tbl_alternatif($alternatif_id);
        redirect('tbl_alternatif/index');
    }
    else
        show_error('The tbl_alternatif you are trying to delete does not
exist.');
}

```

```
}
```

```
}
```

Setelah fungsi edit dilanjutkan dengan fungsi remove yang digunakan untuk menghapus data, fungsi ini menggunakan parameter id yang di kirim melalui tombol yang berkaitan dengan data yang akan di hapus.

Untuk lebih jelasnya berikut merupakan code lengkap pada file Tbl_alternatif yang terdapat pada controller.

```
<?php
class Tbl_alternatif extends CI_Controller{
    function __construct()
    {
        parent::__construct();
        $this->load->model('Tbl_alternatif_model');
    }

    function index()
    {
        $data['tbl_alternatif'] = $this->Tbl_alternatif_model-
>get_all_tbl_alternatif();

        $data['_view'] = 'tbl_alternatif/index';
        $this->load->view('layouts/main',$data);
    }

    function add()
    {
        $this->load->library('form_validation');

        $this->form_validation-
>set_rules('alternatif_nama','Alternatif Nama','required|max_length[30]');
        $this->form_validation->set_rules('kriteria_1','Kriteria
1','required|integer');
        $this->form_validation->set_rules('kriteria_2','Kriteria
2','required|integer');
        $this->form_validation->set_rules('kriteria_3','Kriteria
3','required|integer');
    }
}
```

```

        $this->form_validation->set_rules('kriteria_4','Kriteria
4','required|integer');
        $this->form_validation->set_rules('tipe_alternatif','Tipe
Alternatif','required|max_length[10]');

        if($this->form_validation->run())
{
    $params = array(
        'alternatif_nama' => $this->input-
>post('alternatif_nama'),
        'kriteria_1' => $this->input->post('kriteria_1'),
        'kriteria_2' => $this->input->post('kriteria_2'),
        'kriteria_3' => $this->input->post('kriteria_3'),
        'kriteria_4' => $this->input->post('kriteria_4'),
        'tipe_alternatif' => $this->input-
>post('tipe_alternatif'),
    );
}

$tbl_alternatif_id = $this->Tbl_alternatif_model-
>add_tbl_alternatif($params);
redirect('tbl_alternatif/index');

}
else
{
    $data['_view'] = 'tbl_alternatif/add';
    $this->load->view('layouts/main',$data);
}
}

function edit($alternatif_id)
{
    // check if the tbl_alternatif exists before trying to edit it
    $data['tbl_alternatif'] = $this->Tbl_alternatif_model-
>get_tbl_alternatif($alternatif_id);

    if(isset($data['tbl_alternatif']['alternatif_id']))
    {
        $this->load->library('form_validation');

        $this->form_validation-
>set_rules('alternatif_nama','Alternatif Nama','required|max_length[30]');
        $this->form_validation->set_rules('kriteria_1','Kriteria

```

```

1','required|integer');
                $this->form_validation->set_rules('kriteria_2','Kriteria
2','required|integer');
                $this->form_validation->set_rules('kriteria_3','Kriteria
3','required|integer');
                $this->form_validation->set_rules('kriteria_4','Kriteria
4','required|integer');
                $this->form_validation-
>set_rules('tipe_alternatif','Tipe Alternatif','required|max_length[10]');

                if($this->form_validation->run())
{
    $params = array(
        'alternatif_nama' => $this->input-
>post('alternatif_nama'),
        'kriteria_1' => $this->input-
>post('kriteria_1'),
        'kriteria_2' => $this->input-
>post('kriteria_2'),
        'kriteria_3' => $this->input-
>post('kriteria_3'),
        'kriteria_4' => $this->input-
>post('kriteria_4'),
        'tipe_alternatif' => $this->input-
>post('tipe_alternatif'),
    );
}

        $this->Tbl_alternatif_model-
>update_tbl_alternatif($alternatif_id,$params);
        redirect('tbl_alternatif/index');
    }
    else
    {
        $data['_view'] = 'tbl_alternatif/edit';
        $this->load->view('layouts/main',$data);
    }
}
else
    show_error('The tbl_alternatif you are trying to edit does not exist.');
}

function remove($alternatif_id)

```

```

{
    $tbl_alternatif = $this->Tbl_alternatif_model-
>get_tbl_alternatif($alternatif_id);

    // check if the tbl_alternatif exists before trying to delete it
    if(isset($tbl_alternatif['alternatif_id']))
    {
        $this->Tbl_alternatif_model->delete_tbl_alternatif($alternatif_id);
        redirect('tbl_alternatif/index');
    }
    else
        show_error('The tbl_alternatif you are trying to delete does not
exist.');
}
}

```

Setelah file Tbl_alternatif.php telah siap, selanjutnya buka file Tbl_alternatif_model kemudian isikan code berikut:

```

<?php
class Tbl_alternatif_model extends CI_Model
{
    function __construct()
    {
        parent::__construct();
    }

    function get_tbl_alternatif($alternatif_id)
    {
        $tbl_alternatif = $this->db->query("
            SELECT
                *
            FROM
                `tbl_alternatif`

            WHERE
                `alternatif_id` = ?
            ",array($alternatif_id))->row_array();
    }
}

```

```
        return $tbl_alternatif;
    }
```

Pada file tersebut buat class dengan nama file yang sama kemudian buat fungsi konstruktor, selanjutnya buat fungsi get_tbl_alternatif yang digunakan untuk mengambil data alternatif berdasarkan parameter id alternatif, data id alternatif tersebut di kirimkan melalui fungsi edit yang terdapat pada controller.

```
function get_all_tbl_alternatif()
{
    $tbl_alternatif = $this->db->query("
        SELECT
            *
        FROM
            `tbl_alternatif`
        WHERE
            1 = 1
        ORDER BY `alternatif_id` DESC
    ")->result_array();
    return $tbl_alternatif;
}
```

Kemudian dilanjutkan dengan membuat fungsi get_all_tbl_alternatif yang digunakan untuk memanggil atau mengambil semua data yang terdapat pada tabel alternatif pada basis data, fungsi ini di gunakan pada fungsi index pada controller.

```
function add_tbl_alternatif($params)
{
    $this->db->insert('tbl_alternatif',$params);
    return $this->db->insert_id();
}
function update_tbl_alternatif($alternatif_id,$params)
{
```

```

        $this->db->where('alternatif_id',$alternatif_id);
        return $this->db->update('tbl_alternatif',$params);
    }

    function delete_tbl_alternatif($alternatif_id)
    {
        return $this->db-
>delete('tbl_alternatif',array('alternatif_id'=>$alternatif_id));
    }
}

```

Selanjutnya dilanjutkan dengan membuat fungsi add_tbl_alternatif , update_tbl_alternatif , dan delete_tbl_alternatif, yang mana fungsi add_tbl_alternatif digunakan untuk menambahkan data ke database kemudian fungsi ini terhubung dengan fungsi add pada controller kemudian untuk fungsi update_tbl_alternatif terhubung dengan fungsi edit pada controller dengan fungsi tersebut data alternatif dapat diperbarui kemudian untuk fungsi delete_tbl_alternatif digunakan untuk menghapus data dan terhubung dengan fungsi remove pada controller.

Briuk merupakan code lengkap pada file Tbl_alternatif_model:

```

<?php
class Tbl_alternatif_model extends CI_Model
{
    function __construct()
    {
        parent::__construct();
    }
    function get_tbl_alternatif($alternatif_id)
    {
        $tbl_alternatif = $this->db->query("
            SELECT
            *
            FROM
            `tbl_alternatif`
            WHERE
            `alternatif_id` = ?
        ");
    }
}

```

```

        ","array($alternatif_id))->row_array();
        return $tbl_alternatif;
    }
    function get_all_tbl_alternatif()
    {
        $tbl_alternatif = $this->db->query(
            "SELECT
                *
            FROM
                `tbl_alternatif`
            WHERE
                1 = 1
            ORDER BY `alternatif_id` DESC
        ")->result_array();
        return $tbl_alternatif;
    }
    function add_tbl_alternatif($params)
    {
        $this->db->insert('tbl_alternatif',$params);
        return $this->db->insert_id();
    }
    function update_tbl_alternatif($alternatif_id,$params)
    {
        $this->db->where('alternatif_id',$alternatif_id);
        return $this->db->update('tbl_alternatif',$params);
    }

    function delete_tbl_alternatif($alternatif_id)
    {
        return $this->db-
>delete('tbl_alternatif',array('alternatif_id'=>$alternatif_id));
    }
}

```

Setelah membuat dan memindahkan code tersebut pada model dilanjutkan pada file index.php yang terdapat pada direktori view folder tbl_alternatif pada file index.php tersebut isi dengan code berikut. Code tersebut merupakan tampilan data alternatif berupa tabel, dimana data tersebut di kirim melalui parameter array asosiatif yang terdapat pada fungsi index yang terdapat pada file Tbl_alternatif.

Berikut merupakan data code untuk file index.php pada view tbl_alternatif.

```
<div class="content-header">

<div class="container-fluid">

<div class="row mb-2">

<div class="col-sm-6">

<h1 class="m-0 text-dark">Alternatif</h1>

</div><!-- /.col -->

</div><!-- /.row -->

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

</div>

<section class="content">

<div class="container-fluid">

<div class="card">

<div class="card-header">

<h3 class="card-title">Data Alternatif dan Kriteria</h3>

<div class="float-right">

<a href=<?php echo site_url('tbl_alternatif/add'); ?>">

class="btn btn-success btn-sm">Add</a>
```

```
</div>

</div>

<!-- /.card-header -->


```

```

<td><?php echo $t['alternatif_id']; ?></td>

<td><?php echo $t['alternatif_nama']; ?></td>

<td><?php echo $t['kriteria_1']; ?></td>

<td><?php echo $t['kriteria_2']; ?></td>

<td><?php echo $t['kriteria_3']; ?></td>

<td><?php echo $t['kriteria_4']; ?></td>

<td><?php echo $t['tipe_alternatif']; ?></td>

<td>

    <a href=<?php echo site_url('tbl_alternatif/edit/' .
$t['alternatif_id']); ?>" class="btn btn-info btn-xs"><span class="fa fa-pencil"></span> Edit</a>

    <a href=<?php echo site_url('tbl_alternatif/remove/' .
$t['alternatif_id']); ?>" class="btn btn-danger btn-xs"><span class="fa fa-trash"></span> Delete</a>

</td>

</tr>

<?php } ?>

</tbody>

<tfoot>

<tr>

    <th>Alternatif Id</th>

```

```
<th>Alternatif Nama</th>
<th>Kriteria 1</th>
<th>Kriteria 2</th>
<th>Kriteria 3</th>
<th>Kriteria 4</th>
<th>Tipe Alternatif</th>
<th>Actions</th>
</tr>
</tfoot>
</table>
</div>
<!-- /.card-body -->
</div><!-- /.container-fluid -->
</section>
```

Untuk hasil dari code tersebut sebagai berikut:

Selanjutnya yaitu pada file add.php pada folder tbl_alternatif isikan code sebagai berikut, code tersebut digunakan untuk membuat tampilan form input untuk data alternatif, yang terdiri dari form – form kriteria label, nama alternatif, dan tipe dari alternatif tersebut.

```
<div class="content-header">
    <div class="container-fluid">
        <div class="row mb-2">
            <div class="col-sm-6">
                <h3 class="m-0 text-dark">Form tambah Data Alternatif</h3>
            </div><!-- /.col -->
        </div><!-- /.row -->
    </div><!-- /.container-fluid -->
</div>
```

```
<section class="content">

<div class="container-fluid">

    <div class="card card-primary">

        <div class="card-header">

            <h3 class="card-title">Menambah Data
Alternatif</h3>

        </div>

        <!-- /.card-header -->

        <!-- form start -->

        <?php echo form_open('tbl_alternatif/add'); ?>

        <div class="card-body">

            <label for="alternatif_nama" class="control-
label"><span class="text-danger">*</span>Alternatif Nama</label>

            <div class="form-group">

                <input type="text"
name="alternatif_nama" value="<?php echo $this->input-
>post('alternatif_nama'); ?>" class="form-control" id="alternatif_nama" />

                <span class="text-danger"><?php echo
form_error('alternatif_nama'); ?></span>

            </div>

            <label for="kriteria_1" class="control-
```

```
label"><span class="text-danger">*</span>Kriteria 1</label>

    <div class="form-group">

        <input type="text" name="kriteria_1"
value=<?php echo $this->input->post('kriteria_1'); ?>" class="form-
control" id="kriteria_1" />

        <span class="text-danger"><?php echo
form_error('kriteria_1'); ?></span>

    </div>

    <label for="kriteria_2" class="control-
label"><span class="text-danger">*</span>Kriteria 2</label>

    <div class="form-group">

        <input type="text" name="kriteria_2"
value=<?php echo $this->input->post('kriteria_2'); ?>" class="form-
control" id="kriteria_2" />

        <span class="text-danger"><?php echo
form_error('kriteria_2'); ?></span>

    </div>

    <label for="kriteria_3" class="control-
label"><span class="text-danger">*</span>Kriteria 3</label>

    <div class="form-group">

        <input type="text" name="kriteria_3"
value=<?php echo $this->input->post('kriteria_3'); ?>" class="form-
control" id="kriteria_3" />
```

```
<span class="text-danger"><?php echo  
form_error('kriteria_3'); ?></span>  
  
        </div>  
  
        <label for="kriteria_4" class="control-  
label"><span class="text-danger">*</span>Kriteria 4</label>  
  
        <div class="form-group">  
  
            <input type="text" name="kriteria_4"  
value="<?php echo $this->input->post('kriteria_4'); ?>" class="form-  
control" id="kriteria_4" />  
  
            <span class="text-danger"><?php echo  
form_error('kriteria_4'); ?></span>  
  
        </div>  
  
        <label for="tipe_alternatif" class="control-  
label"><span class="text-danger">*</span>Tipe Alternatif</label>  
  
        <div class="form-group">  
  
            <input type="text"  
name="tipe_alternatif" value="<?php echo $this->input-  
>post('tipe_alternatif'); ?>" class="form-control" id="tipe_alternatif" />  
  
            <span class="text-danger"><?php echo  
form_error('tipe_alternatif'); ?></span>  
  
        </div>  
  
    </div>
```

```

<div class="card-footer">

    <button type="submit" class="btn btn-primary">Submit</button>

</div>

<?php echo form_close(); ?>

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

</section>

```

Setelah di isikan code tersebut maka tampilan dari form input atau add data alternatif adalah sebagai berikut:

Setelah membuat tampilan form untuk menambah data alternatif berikutnya yaitu membuka file edit.php pada folder tbl_alternatif, pada dasarnya untuk tampilan sangat mirip hanyasaja beda pada form input, untuk

form edit telah disisipkan nilai dari data yang akan diedit, sehingga pada form edit form inputnya sudah terdapat data yang akan diedit.

Berikut merupakan code untuk form edit yang terdapat pada file edit.php.

```
<div class="content-header">

    <div class="container-fluid">

        <div class="row mb-2">

            <div class="col-sm-6">

                <h3 class="m-0 text-dark">Form Edit Data
                Alternatif</h3>

            </div><!-- /.col -->

        </div><!-- /.row -->

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

</div>

<section class="content">

    <div class="container-fluid">

        <div class="card card-primary">

            <div class="card-header">

                <h3 class="card-title">Mengedit Data
                Alternatif</h3>

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

```
</div>

<!-- /.card-header -->

<!-- form start -->

<?php echo form_open('tbl_alternatif/edit' .
$tbl_alternatif['alternatif_id']); ?>

<div class="card-body">

    <label for="alternatif_nama" class="control-
label"><span class="text-danger">*</span>Alternatif Nama</label>

    <div class="form-group">

        <input type="text"
name="alternatif_nama" value=<?php echo ($this->input-
>post('alternatif_nama') ? $this->input->post('alternatif_nama') :
$tbl_alternatif['alternatif_nama']); ?>" class="form-control"
id="alternatif_nama" />

        <span class="text-danger"><?php echo
form_error('alternatif_nama'); ?></span>

    </div>

    <label for="kriteria_1" class="control-
label"><span class="text-danger">*</span>Kriteria 1</label>
```

```
<div class="form-group">

    <input type="text" name="kriteria_1"
value=<?php echo ($this->input->post('kriteria_1') ? $this->input-
>post('kriteria_1') : $tbl_alternatif['kriteria_1']); ?>" class="form-control"
id="kriteria_1" />

    <span class="text-danger"><?php echo
form_error('kriteria_1'); ?></span>

</div>

<label for="kriteria_2" class="control-
label"><span class="text-danger">*</span>Kriteria 2</label>

<div class="form-group">

    <input type="text" name="kriteria_2"
value=<?php echo ($this->input->post('kriteria_2') ? $this->input-
>post('kriteria_2') : $tbl_alternatif['kriteria_2']); ?>" class="form-control"
id="kriteria_2" />

    <span class="text-danger"><?php echo
form_error('kriteria_2'); ?></span>

</div>

<label for="kriteria_3" class="control-
```

```
label"><span class="text-danger">*</span>Kriteria 3</label>

        <div class="form-group">

            <input type="text" name="kriteria_3"
value=<?php echo ($this->input->post('kriteria_3') ? $this->input-
>post('kriteria_3') : $tbl_alternatif['kriteria_3']); ?>" class="form-control"
id="kriteria_3" />

            <span class="text-danger"><?php echo
form_error('kriteria_3'); ?></span>

        </div>

        <label for="kriteria_4" class="control-
label"><span class="text-danger">*</span>Kriteria 4</label>

        <div class="form-group">

            <input type="text" name="kriteria_4"
value=<?php echo ($this->input->post('kriteria_4') ? $this->input-
>post('kriteria_4') : $tbl_alternatif['kriteria_4']); ?>" class="form-control"
id="kriteria_4" />

            <span class="text-danger"><?php echo
form_error('kriteria_4'); ?></span>

        </div>
```

```
<label for="tipe_alternatif" class="control-label"><span class="text-danger">*</span>Tipe Alternatif</label>

<div class="form-group">

    <input type="text"
name="tipe_alternatif" value=<?php echo ($this->input->post('tipe_alternatif') ? $this->input->post('tipe_alternatif') :
Tbl_alternatif['tipe_alternatif']); ?>" class="form-control"
id="tipe_alternatif" />

    <span class="text-danger"><?php echo
form_error('tipe_alternatif'); ?></span>

</div>

</div>

<div class="card-footer">

    <button type="submit" class="btn btn-primary">Submit</button>

</div>

<?php echo form_close(); ?>

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

```
</section>
```

Untuk Hasilnya sebagai berikut:

The screenshot shows a web application interface. On the left is a sidebar with a logo, the title 'Aplikasi Entropy', and a user icon labeled 'admin'. Below these are four menu items: 'Dashboard' (selected), 'Data User', 'Data Alternatif', and 'Proses Entropy'. At the bottom of the sidebar are 'Sign Out' and a help icon. The main content area has a header 'Form Edit Data User' and a sub-header 'Mengedit Data User'. It contains five input fields with validation asterisks: 'User Password' (value: admin), 'User Name' (value: admin), 'User Email' (value: admin@admin.co), 'User Level' (value: admin), and 'Status' (value: 1). A blue 'Submit' button is at the bottom.

Code Controller User

```
<?php  
  
class Tbl_user extends CI_Controller  
  
{  
  
    function __construct()  
    {  
  
        parent::__construct();  
  
        $this->load->model('Tbl_user_model');  
  
        $this->load->library('enkripsi');  
  
    }  
}
```

```
function index()

{
    $data['tbl_user'] = $this->Tbl_user_model->get_all_tbl_user();

    $data['_view'] = 'tbl_user/index';

    $this->load->view('layouts/main', $data);

}

function add()
{
    $this->load->library('form_validation');

    $this->form_validation->set_rules('user_password', 'User Password',
'required|max_length[60]');

    $this->form_validation->set_rules('user_name', 'User Name',
'required|max_length[30]');

    $this->form_validation->set_rules('user_email', 'User Email',
'required|max_length[60]|valid_email');

    $this->form_validation->set_rules('user_level', 'User Level',
'required|max_length[5]');

    $this->form_validation->set_rules('status', 'Status', 'required|integer');
```

```
if ($this->form_validation->run()) {  
  
    $params = array(  
  
        'user_password' => $this->enkripsi->encryptIt($this->input->post('user_password')),  
  
        'user_name' => $this->input->post('user_name'),  
  
        'user_email' => $this->input->post('user_email'),  
  
        'user_level' => $this->input->post('user_level'),  
  
        'status' => $this->input->post('status'),  
  
    );  
  
  
  
    $tbl_user_id = $this->Tbl_user_model->add_tbl_user($params);  
  
    redirect('tbl_user/index');  
  
} else {  
  
    $data['_view'] = 'tbl_user/add';  
  
    $this->load->view('layouts/main', $data);  
  
}  
  
}  
  
  
  
function edit($user_id)
```

```
{  
  
    $data['tbl_user'] = $this->Tbl_user_model->get_tbl_user($user_id);  
  
    if (isset($data['tbl_user']['user_id'])) {  
  
        $this->load->library('form_validation');  
  
        $this->form_validation->set_rules('user_password', 'User Password',  
            'required|max_length[60]');  
  
        $this->form_validation->set_rules('user_name', 'User Name',  
            'required|max_length[30]');  
  
        $this->form_validation->set_rules('user_email', 'User Email',  
            'required|max_length[60]|valid_email');  
  
        $this->form_validation->set_rules('user_level', 'User Level',  
            'required|max_length[5]');  
  
        $this->form_validation->set_rules('status', 'Status', 'required|integer');  
  
        if ($this->form_validation->run()) {  
  
            $params = array(  
                'user_password' => $this->enkripsi->encryptIt($this->input->post('user_password')),  
                'user_name' => $this->input->post('user_name'),
```

```
'user_email' => $this->input->post('user_email'),  
  
'user_level' => $this->input->post('user_level'),  
  
'status' => $this->input->post('status'),  
  
);  
  
  
  
$this->Tbl_user_model->update_tbl_user($user_id, $params);  
  
redirect('tbl_user/index');  
  
} else {  
  
    $data['_view'] = 'tbl_user/edit';  
  
    $this->load->view('layouts/main', $data);  
  
}  
  
} else  
  
    show_error('The tbl_user you are trying to edit does not exist.');
```

}


```
function remove($user_id)  
  
{  
  
    $tbl_user = $this->Tbl_user_model->get_tbl_user($user_id);  
  
  
  
    if (isset($tbl_user['user_id'])) {
```

```
$this->Tbl_user_model->delete_tbl_user($user_id);

redirect('tbl_user/index');

} else

show_error('The tbl_user you are trying to delete does not exist.');

}

}
```

Code Model User

```
<?php

class Tbl_user_model extends CI_Model

{

function __construct()

{

parent::__construct();

}

}

function get_tbl_user($user_id)

{

$tbl_user = $this->db->query("

SELECT
```

```
*  
  
FROM  
  
`tbl_user`  
  
  
WHERE  
  
`user_id` = ?  
  
", array($user_id))->row_array();  
  
  
  
return $tbl_user;  
  
}  
  
function get_all_tbl_user()  
  
{  
  
$tbl_user = $this->db->query("SELECT  
*  
  
  
FROM  
  
`tbl_user`
```

```
WHERE  
      1 = 1  
  
      ORDER BY `user_id` DESC  
      ")>result_array();  
  
      return $tbl_user;  
  }  
  
  function add_tbl_user($params)  
  {  
      $this->db->insert('tbl_user', $params);  
      return $this->db->insert_id();  
  }  
  
  function update_tbl_user($user_id, $params)  
  {  
      $this->db->where('user_id', $user_id);  
      return $this->db->update('tbl_user', $params);  
  }  
  
  function delete_tbl_user($user_id)  
  {
```

```
        return $this->db->delete('tbl_user', array('user_id' => $user_id));  
    }  
}
```

Code View User index

```
<div class="content-header">  
  
    <div class="container-fluid">  
  
        <div class="row mb-2">  
  
            <div class="col-sm-6">  
  
                <h1 class="m-0 text-dark">User</h1>  
  
            </div><!-- /.col -->  
  
        </div><!-- /.row -->  
  
    </div><!-- /.container-fluid -->  
  
</div>  
  
  
  
<section class="content">  
  
    <div class="container-fluid">  
  
        <div class="card">  
  
            <div class="card-header">  
  
                <h3 class="card-title">Data User</h3>
```

```
<div class="float-right">

    <a href="<?php echo site_url('tbl_user/add'); ?>" class="btn btn-
success btn-sm">Add</a>

</div>

</div>

<!-- /.card-header -->

<div class="card-body">

    <table id="example2" class="table table-bordered table-striped">

        <thead>

            <tr>

                <th>User Id</th>

                <th>User Password</th>

                <th>User Name</th>

                <th>User Email</th>

                <th>User Level</th>

                <th>Status</th>

                <th>Actions</th>

            </tr>

        </thead>

        <tbody>
```

```
<?php foreach ($tbl_user as $t) { ?>

<tr>

<td><?php echo $t['user_id']; ?></td>

<td><?php echo $this->enkripsi-
>decryptIt($t['user_password']); ?></td>

<td><?php echo $t['user_name']; ?></td>

<td><?php echo $t['user_email']; ?></td>

<td><?php echo $t['user_level']; ?></td>

<td><?php if ($t['status'] === '1') {

    echo 'aktif';

} else {

    echo 'tidak aktif';

}

?></td>

<td>

<a href=<?php echo site_url('tbl_user/edit/' .
$t['user_id']); ?>" class="btn btn-info btn-xs"><span class="fa fa-
pencil"></span> Edit</a>

<a href=<?php echo site_url('tbl_user/remove/' .
$t['user_id']); ?>" class="btn btn-danger btn-xs"><span class="fa fa-
```

```
trash"></span> Delete</a>

        </td>

    </tr>

<?php } ?>

</tbody>

<tfoot>

    <tr>

        <th>User Id</th>

        <th>User Password</th>

        <th>User Name</th>

        <th>User Email</th>

        <th>User Level</th>

        <th>Status</th>

        <th>Actions</th>

    </tr>

</tfoot>

</table>

</div>

<!-- /.card-body -->

</div>
```

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

Code View User add

```
<div class="content-header">  
  <div class="container-fluid">  
    <div class="row mb-2">  
      <div class="col-sm-6">  
        <h1 class="m-0 text-dark">Form Tambah  
        Data User</h1>  
      </div><!-- /.col -->  
    </div><!-- /.row -->  
  </div><!-- /.container-fluid -->  
</div>  
  
<section class="content">  
  <div class="container-fluid">  
    <div class="card card-primary">  
      <div class="card-header">  
        <h3 class="card-title">Menambah Data
```

```
User</h3>

        </div>

        <!-- /.card-header -->

        <!-- form start -->

        <?php echo form_open('tbl_user/add'); ?>

        <div class="card-body">

            <label for="user_password" class="control-
label"><span class="text-danger">*</span>User Password</label>

            <div class="form-group">

                <input type="password"
name="user_password" value="<?php echo $this->input-
>post('user_password'); ?>" class="form-control" id="user_password" />

                <span class="text-danger"><?php echo
form_error('user_password'); ?></span>

            </div>

            <label for="user_name" class="control-
label"><span class="text-danger">*</span>User Name</label>

            <div class="form-group">

                <input type="text" name="user_name"
value="<?php echo $this->input->post('user_name'); ?>" class="form-
control" id="user_name" />

                <span class="text-danger"><?php echo
```

```
form_error('user_name'); ?></span>

        </div>

        <label for="user_email" class="control-
label"><span class="text-danger">*</span>User Email</label>

        <div class="form-group">

            <input type="text" name="user_email"
value=<?php echo $this->input->post('user_email'); ?>" class="form-
control" id="user_email" />

            <span class="text-danger"><?php echo
form_error('user_email'); ?></span>

        </div>

        <label for="user_level" class="control-
label"><span class="text-danger">*</span>User Level</label>

        <div class="form-group">

            <input type="text" name="user_level"
value=<?php echo $this->input->post('user_level'); ?>" class="form-
control" id="user_level" />

            <span class="text-danger"><?php echo
form_error('user_level'); ?></span>

        </div>

        <label for="status" class="control-
label"><span class="text-danger">*</span>Status</label>
```

```

<div class="form-group">

    <input type="text" name="status"
value=<?php echo $this->input->post('status'); ?>" class="form-control"
id="status" />

    <span class="text-danger"><?php echo
form_error('status'); ?></span>

</div>

</div>

<div class="card-footer">

    <button type="submit" class="btn btn-
primary">Submit</button>

</div>

<?php echo form_close(); ?>

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

</section>

```

Code View User Edit

```

<div class="content-header">

    <div class="container-fluid">

```

```
<div class="row mb-2">

    <div class="col-sm-6">

        <h3 class="m-0 text-dark">Form Edit Data
User</h3>

    </div><!-- /.col -->

</div><!-- /.row -->

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

</div>

<section class="content">

<div class="container-fluid">

    <div class="card card-primary">

        <div class="card-header">

            <h3 class="card-title">Mengedit Data
User</h3>

        </div>

        <!-- /.card-header -->

        <!-- form start -->

        <?php echo form_open('tbl_user/edit/' .
$tbl_user['user_id']); ?>

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

```
<label for="user_password" class="control-label"><span class="text-danger">*</span>User Password</label>

<div class="form-group">

    <input type="text"
name="user_password" value="<?php echo ($this->input->post('user_password') ? $this->input->post('user_password') : $this->enkripsi->decryptIt($tbl_user['user_password'])); ?>" class="form-control"
id="user_password" />

    <span class="text-danger"><?php echo
form_error('user_password'); ?></span>

</div>

<label for="user_name" class="control-label"><span class="text-danger">*</span>User Name</label>

<div class="form-group">

    <input type="text" name="user_name"
value="<?php echo ($this->input->post('user_name') ? $this->input->post('user_name') : $tbl_user['user_name']); ?>" class="form-control"
id="user_name" />

    <span class="text-danger"><?php echo
form_error('user_name'); ?></span>

</div>

<label for="user_email" class="control-label"><span class="text-danger">*</span>User Email</label>
```

```
<div class="form-group">

    <input type="text" name="user_email"
value=<?php echo ($this->input->post('user_email') ? $this->input-
>post('user_email') : $tbl_user['user_email']); ?>" class="form-control"
id="user_email" />

    <span class="text-danger"><?php echo
form_error('user_email'); ?></span>

</div>

    <label for="user_level" class="control-
label"><span class="text-danger">*</span>User Level</label>

    <div class="form-group">

        <input type="text" name="user_level"
value=<?php echo ($this->input->post('user_level') ? $this->input-
>post('user_level') : $tbl_user['user_level']); ?>" class="form-control"
id="user_level" />

        <span class="text-danger"><?php echo
form_error('user_level'); ?></span>

    </div>

    <label for="status" class="control-
label"><span class="text-danger">*</span>Status</label>

    <div class="form-group">

        <input type="text" name="status"
value=<?php echo ($this->input->post('status') ? $this->input->post('status')
```

```
: $tbl_user['status']); ?>" class="form-control" id="status" />

        <span class="text-danger"><?php echo
form_error('status'); ?></span>

    </div>

</div>

<div class="card-footer">

    <button type="submit" class="btn btn-
primary">Submit</button>

    </div>

    <?php echo form_close(); ?>

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

</section>
```

Code controller tbl bobot

```
<?php

class Tbl_bobot extends CI_Controller

{

    function __construct()

    {
```

```
parent::__construct();

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

}

function index()

{

    $data['tbl_bobot'] = $this->Tbl_bobot_model->get_all_tbl_bobot();

    $data['_view'] = 'tbl_bobot/index';

    $this->load->view('layouts/main', $data);

}

function add()

{

    $this->load->library('form_validation');

    $this->form_validation->set_rules('bobot_keriteria1', 'Bobot Keriteria1', 'required');

    $this->form_validation->set_rules('bobot_keriteria2', 'Bobot Keriteria2', 'required');

    $this->form_validation->set_rules('bobot_keriteria3', 'Bobot Keriteria3', 'required');

    $this->form_validation->set_rules('bobot_keriteria4', 'Bobot Keriteria4',
```

```
'required');

$this->form_validation->set_rules('id_user', 'Id User', 'required');

$this->form_validation->set_rules('tipe', 'Tipe',
'required|max_length[10]');

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

$params = array(
    'bobot_keriteria1' => $this->input->post('bobot_keriteria1'),
    'bobot_keriteria2' => $this->input->post('bobot_keriteria2'),
    'bobot_keriteria3' => $this->input->post('bobot_keriteria3'),
    'bobot_keriteria4' => $this->input->post('bobot_keriteria4'),
    'id_user' => $this->input->post('id_user'),
    'tipe' => $this->input->post('tipe'),
);

$tbl_bobot_id = $this->Tbl_bobot_model->add_tbl_bobot($params);

redirect('tbl_bobot/index');

} else {

$data['_view'] = 'tbl_bobot/add';

$this->load->view('layouts/main', $data);
```

```
}

}

function edit($id_bobot)

{

    // check if the tbl_bobot exists before trying to edit it

    $data['tbl_bobot'] = $this->Tbl_bobot_model-
>get_tbl_bobot($id_bobot);

if (isset($data['tbl_bobot']['id_bobot'])) {

    $this->load->library('form_validation');

    $this->form_validation->set_rules('bobot_keriteria1', 'Bobot
Keriteria1', 'required');

    $this->form_validation->set_rules("bobot_keriteria2", 'Bobot
Keriteria2', 'required');

    $this->form_validation->set_rules('bobot_keriteria3', 'Bobot
Keriteria3', 'required');

    $this->form_validation->set_rules('bobot_keriteria4', 'Bobot
Keriteria4', 'required');

    $this->form_validation->set_rules('id_user', 'Id User', 'required');

    $this->form_validation->set_rules('tipe', 'Tipe',
```

```
'required|max_length[10]');

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

    $params = array(
        'bobot_keriteria1' => $this->input->post('bobot_keriteria1'),
        'bobot_keriteria2' => $this->input->post('bobot_keriteria2'),
        'bobot_keriteria3' => $this->input->post('bobot_keriteria3'),
        'bobot_keriteria4' => $this->input->post('bobot_keriteria4'),
        'id_user' => $this->input->post('id_user'),
        'tipe' => $this->input->post('tipe'),
    );

    $this->Tbl_bobot_model->update_tbl_bobot($id_bobot, $params);
    redirect('tbl_bobot/index');

} else {

    $data['_view'] = 'tbl_bobot/edit';
    $this->load->view('layouts/main', $data);

}

} else

show_error('The tbl_bobot you are trying to edit does not exist.');
```

```
}

function remove($id_bobot)

{
    $tbl_bobot = $this->Tbl_bobot_model->get_tbl_bobot($id_bobot);

    // check if the tbl_bobot exists before trying to delete it

    if (isset($tbl_bobot['id_bobot'])) {

        $this->Tbl_bobot_model->delete_tbl_bobot($id_bobot);

        redirect('tbl_bobot/index');

    } else

        show_error('The tbl_bobot you are trying to delete does not exist.');

    }

}
```

Code model tbl bobot

```
<?php

class Tbl_bobot_model extends CI_Model

{

    function __construct()

    {
```

```
parent::__construct();

}

function get_tbl_bobot($id_bobot)
{
    return $this->db->get_where('tbl_bobot',array('id_bobot'=>$id_bobot))->row_array();
}

function get_all_tbl_bobot()
{
    $this->db->order_by('id_bobot', 'desc');

    return $this->db->get('tbl_bobot')->result_array();
}

function add_tbl_bobot($params)
{
    $this->db->insert('tbl_bobot',$params);

    return $this->db->insert_id();
}
```

```
function update_tbl_bobot($id_bobot,$params)
{
    $this->db->where('id_bobot',$id_bobot);
    return $this->db->update('tbl_bobot',$params);
}

function delete_tbl_bobot($id_bobot)
{
    return $this->db->delete('tbl_bobot',array('id_bobot'=>$id_bobot));
}
```

Code view indx bobot

```
<div class="content-header">
<div class="container-fluid">
<div class="row mb-2">
<div class="col-sm-6">
<h1 class="m-0 text-dark">Bobot</h1>
</div><!-- /.col -->
```

```
</div><!-- /.row -->

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

</div>

<section class="content">

<div class="container-fluid">

<div class="card">

<div class="card-header">

<h3 class="card-title">Data Bobot Menggunakan Entropy</h3>

<div class="float-right">

<?php if ($this->session->userdata('user_level') === 'admin') {

?>

<a href="php echo site_url('tbl_bobot/add'); ?&gt;" class="btn btn-success btn-sm"&gt;Add&lt;/a&gt;

&lt;?php } ?&gt;

&lt;/div&gt;

&lt;/div&gt;

&lt;!-- /.card-header --&gt;

&lt;div class="card-body"&gt;

&lt;table id="example2" class="table table-bordered table-striped"&gt;</pre
```

```
<thead>

<tr>

<th>Id Bobot</th>

<th>Bobot Keriteria1</th>

<th>Bobot Keriteria2</th>

<th>Bobot Keriteria3</th>

<th>Bobot Keriteria4</th>

<th>Id User</th>

<th>Tipe</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

<?php foreach ($tbl_bobot as $t) { ?>

<tr>

<td><?php echo $t['id_bobot']; ?></td>

<td><?php echo $t['bobot_keriteria1']; ?></td>

<td><?php echo $t['bobot_keriteria2']; ?></td>

<td><?php echo $t['bobot_keriteria3']; ?></td>

<td><?php echo $t['bobot_keriteria4']; ?></td>
```

```

<td><?php echo $t['id_user']; ?></td>

<td><?php echo $t['tipe']; ?></td>

<td>

    <a href=<?php echo site_url('tbl_bobot/edit/' .
$t['id_bobot']); ?>" class="btn btn-info btn-xs"><span class="fa fa-
pencil"></span> Edit</a>

    <a href=<?php echo site_url('tbl_bobot/remove/' .
$t['id_bobot']); ?>" class="btn btn-danger btn-xs"><span class="fa fa-
trash"></span> Delete</a>

</td>

</tr>

<?php } ?>

</tbody>

<tfoot>

<tr>

    <th>Id Bobot</th>

    <th>Bobot Keriterial</th>

    <th>Bobot Keriteria2</th>

    <th>Bobot Keriteria3</th>

    <th>Bobot Keriteria4</th>

    <th>Id User</th>

```

```
<th>Tipe</th>
<th>Actions</th>
</tr>
</tfoot>
</table>
</div>
<!-- /.card-body -->
</div>
</div><!-- /.container-fluid -->
</section>
```

Code view add bobot

```
<div class="content-header">
  <div class="container-fluid">
    <div class="row mb-2">
      <div class="col-sm-6">
        <h3 class="m-0 text-dark">Form tumpah Data
        Bobot</h3>
        </div><!-- /.col -->
    </div><!-- /.row -->
```

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

</div>

<section class="content">
  <div class="container-fluid">
    <div class="card card-primary">
      <div class="card-header">
        <h3 class="card-title">Menambah Data
        Bobot</h3>
      </div>
      <!-- /.card-header -->
      <!-- form start -->
      <?php echo form_open('tbl_bobot/add'); ?>
      <div class="card-body">
        <label for="bobot_keriteria1" class="control-label"><span class="text-danger">*</span>Bobot Keriteria1</label>
        <div class="form-group">
          <input type="text"
            name="bobot_keriteria1" value="<?php echo $this->input-
            >post('bobot_keriteria1'); ?>" class="form-control" id="bobot_keriteria1" />
          <span class="text-danger"><?php echo
```

```
form_error('bobot_keriteria1'); ?></span>

        </div>

        <label for="bobot_keriteria2" class="control-label"><span class="text-danger">*</span>Bobot Keriteria2</label>

        <div class="form-group">

            <input type="text"
name="bobot_keriteria2" value="<?php echo $this->input-
>post('bobot_keriteria2'); ?>" class="form-control" id="bobot_keriteria2" />

            <span class="text-danger"><?php echo
form_error('bobot_keriteria2'); ?></span>

        </div>

        <label for="bobot_keriteria3" class="control-label"><span class="text-danger">*</span>Bobot Keriteria3</label>

        <div class="form-group">

            <input type="text"
name="bobot_keriteria3" value="<?php echo $this->input-
>post('bobot_keriteria3'); ?>" class="form-control" id="bobot_keriteria3" />

            <span class="text-danger"><?php echo
form_error('bobot_keriteria3'); ?></span>

        </div>

        <label for="bobot_keriteria4" class="control-label"><span class="text-danger">*</span>Bobot Keriteria4</label>
```

```
<div class="form-group">

    <input type="text"
name="bobot_keriteria4" value="<?php echo $this->input-
>post('bobot_keriteria4'); ?>" class="form-control" id="bobot_keriteria4" />

        <span class="text-danger"><?php echo
form_error('bobot_keriteria4'); ?></span>

    </div>

    <label for="id_user" class="control-
label"><span class="text-danger">*</span>Id User</label>

    <div class="form-group">

        <input type="text" name="id_user"
value="<?php echo $this->input->post('id_user'); ?>" class="form-control"
id="id_user" />

        <span class="text-danger"><?php echo
form_error('id_user'); ?></span>

    </div>

    <label for="tipe" class="control-label"><span
class="text-danger">*</span>Tipe</label>

    <div class="form-group">

        <input type="text" name="tipe"
value="<?php echo $this->input->post('tipe'); ?>" class="form-control"
id="tipe" />

        <span class="text-danger"><?php echo
```

```
form_error('tipe'); ?></span>

</div>

</div>

<div class="card-footer">

    <button type="submit" class="btn btn-primary">Submit</button>

</div>

<?php echo form_close(); ?>

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

</section>
```

Code view edit bobot

```
<div class="content-header">

<div class="container-fluid">

    <div class="row mb-2">

        <div class="col-sm-6">

            <h3 class="m-0 text-dark">Form Edit Data

Bobot</h3>

        </div><!-- /.col -->
```

```
</div><!-- /.row -->

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

</div>

<section class="content">

<div class="container-fluid">

    <div class="card card-primary">

        <div class="card-header">

            <h3 class="card-title">Mengedit Data Bobot</h3>

        </div>

        <!-- /.card-header -->

        <!-- form start -->

        <?php echo form_open('tbl_bobot/edit/' .
$tbl_bobot['id_bobot']); ?>

        <div class="card-body">

            <div class="col-md-6">

                <label for="bobot_keriteria1" class="control-label"><span class="text-danger">*</span>Bobot Keriteria1</label>
```

```
<div class="form-group">

    <input type="text"
name="bobot_keriteria1" value="<?php echo ($this->input-
>post('bobot_keriteria1') ? $this->input->post('bobot_keriteria1') :
$tbl_bobot['bobot_keriteria1']); ?>" class="form-control"
id="bobot_keriteria1" />

    <span class="text-
danger"><?php echo form_error('bobot_keriteria1'); ?></span>

</div>

</div>

<div class="col-md-6">

    <label for="bobot_keriteria2"
class="control-label"><span class="text-danger">*</span>Bobot
Keriteria2</label>

    <div class="form-group">

        <input type="text"
name="bobot_keriteria2" value="<?php echo ($this->input-
>post('bobot_keriteria2') ? $this->input->post('bobot_keriteria2') :
$tbl_bobot['bobot_keriteria2']); ?>" class="form-control"
id="bobot_keriteria2" />

        <span class="text-
danger"><?php echo form_error('bobot_keriteria2'); ?></span>

    </div>
```

```
</div>

<div class="col-md-6">

    <label for="bobot_keriteria3"
class="control-label"><span class="text-danger">*</span>Bobot
Keriteria3</label>

    <div class="form-group">

        <input type="text"
name="bobot_keriteria3" value="<?php echo ($this->input-
>post('bobot_keriteria3') ? $this->input->post('bobot_keriteria3') :
$tbl_bobot['bobot_keriteria3']); ?>" class="form-control"
id="bobot_keriteria3" />

        <span class="text-
danger"><?php echo form_error('bobot_keriteria3'); ?></span>

    </div>

    </div>

    <div class="col-md-6">

        <label for="bobot_keriteria4"
class="control-label"><span class="text-danger">*</span>Bobot
Keriteria4</label>

        <div class="form-group">

            <input type="text"
name="bobot_keriteria4" value="<?php echo ($this->input-
>post('bobot_keriteria4') ? $this->input->post('bobot_keriteria4') :
```

```
$tbl_bobot['bobot_keriteria4']); ?>" class="form-control"
id="bobot_keriteria4" />

    <span class="text-
danger"><?php echo form_error('bobot_keriteria4'); ?></span>

</div>

</div>

<div class="col-md-6">

    <label for="id_user" class="control-
label"><span class="text-danger">*</span>Id User</label>

    <div class="form-group">

        <input type="text"
name="id_user" value="<?php echo ($this->input->post('id_user') ? $this-
>input->post('id_user') : $tbl_bobot['id_user']); ?>" class="form-control"
id="id_user" />

        <span class="text-
danger"><?php echo form_error('id_user'); ?></span>

    </div>

</div>

<div class="col-md-6">

    <label for="tipe" class="control-
label"><span class="text-danger">*</span>Tipe</label>

    <div class="form-group">
```

```
<input type="text" name="tipe"
value=<?php echo ($this->input->post('tipe') ? $this->input->post('tipe') :
$tbl_bobot['tipe']); ?>" class="form-control" id="tipe" />

<span class="text-
danger"><?php echo form_error('tipe'); ?></span>

</div>

</div>

</div>

<div class="card-footer">

    <button type="submit" class="btn btn-
primary">Submit</button>

    </div>

    <?php echo form_close(); ?>

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

</section>
```

Penerapan Metode Pada Sistem

Pada file Proses_entropy.php yang terdapat pada controller isikan code berikut yang merupakan logika untuk metode entropy.

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

class Proses_entropy extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->model('Entropy_model');
    }

    public function index()
    {
        $data['_view'] = 'entropy/index';
        $this->load->view('layouts/main', $data);
    }
}
```

Pada code tersebut buat terlebih dahulu dua fungsi terdiri dari fungsi konstruktor yang menjalankan model Entropy_model kemudian fungsi index yang digunakan untuk menampilkan form pencarian data entropy berdasarkan tipe, tipe tersebut menjadikan parameter untuk di mulainya perhitungan entropy.

```
public function proses()
{
    $where = $this->input->post('parameter');
    $data_criteria = $this->Entropy_model->get_criteria($where);
    $data_sum = $this->Entropy_model->sumdata($where);
    if ($data_sum->num_rows() > 0) {

        $total = $data_sum->row();
        $criteria1 = $total->c1;
        $criteria2 = $total->c2;
        $criteria3 = $total->c3;
    }
}
```

```

$criteria4 = $total->c4;
}

$cr1 = 0;
$cr2 = 0;
$cr3 = 0;
$cr4 = 0;
$totalAlternatif = count($data_criteria);
$N = (-1 / log($totalAlternatif));

$hasil = array();
foreach ($data_criteria as $row) {
    $hasil[] = array(
        $cr1 += $row->kriteria_1 / $criteria1 * log($row->kriteria_1 /
$criteria1),
        $cr2 += $row->kriteria_2 / $criteria2 * log($row->kriteria_2 /
$criteria2),
        $cr3 += $row->kriteria_3 / $criteria3 * log($row->kriteria_3 /
$criteria3),
        $cr4 += $row->kriteria_4 / $criteria4 * log($row->kriteria_4 /
$criteria4),
    );
}
$nilaitotal_ej = ((1 - ($N * $cr1)) + (1 - ($N * $cr2)) + (1 - ($N * $cr3)) +
(1 - ($N * $cr4)));

$w_c1 = ((1 - ($N * $cr1)) / $nilaitotal_ej);
$w_c2 = ((1 - ($N * $cr2)) / $nilaitotal_ej);
$w_c3 = ((1 - ($N * $cr3)) / $nilaitotal_ej);
$w_c4 = ((1 - ($N * $cr4)) / $nilaitotal_ej);
$total = (($w_c1) + ($w_c2) + ($w_c3) + ($w_c4));
// print_r($total);
$data = array(
    'c1' => $w_c1,
    'c2' => $w_c2,
    'c3' => $w_c3,
    'c4' => $w_c4,
    'total' => $total,
    'tipe' => $this->input->post('parameter'),
    'id_user' => $this->session->userdata('user_id'),
    '_view' => 'entropy/hasil'
);

```

```
        $this->load->view('layouts/main', $data);
    }
}
```

Pada code tersebut merupakan logika dari entropy dimana fungsi ini menerima tipe dari alternatif untuk memulai proses entropy, kemudian dengan paramater tersebut maka data setiap kriteria dari semua alternatif yang bersangkutan dengan alternatif tersebut diambil begitupula dengan data nilai total dari semua alternatif yang bersangkutan dengan tipe alternatif maka akan di ambil datanya. dimana data total tersebut menjadi pembagi bagi data-data yang lainnya, kemudian setelah di temukan hasilnya maka dikali dengan konsistensi data yang didapatkan dari menghitung jumlah alternatif berdasarkan keriteria.

Kemudian di jadikan nilai pengurang dari 1 (satu) yang merupakan ketentuan dari rumus entropy setelah itu di cari nilai nilai totalnya, kemudian nilai total tersebut menjadi pembagi untuk pengurangan dari 1 (satu) maka di temukan nilai bobot untuk setiap kriteria. Setelah itu data yang terdapat pada variabel tersebut di kirimkan ke view dengan di tampung pada sebuah variabel berisikan array.

Untuk lebih jelasnya berikut merupakan code untuk controller Proses_entropy

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

class Proses_entropy extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->model('Entropy_model');
    }
}
```

```

public function index()
{
    $data['_view'] = 'entropy/index';
    $this->load->view('layouts/main', $data);
}

public function proses()
{
    $where = $this->input->post('parameter');
    $data_criteria = $this->Entropy_model->get_criteria($where);
    $data_sum = $this->Entropy_model->sumdata($where);
    if ($data_sum->num_rows() > 0) {

        $total = $data_sum->row();
        $criteria1 = $total->c1;
        $criteria2 = $total->c2;
        $criteria3 = $total->c3;
        $criteria4 = $total->c4;
    }

    $cr1 = 0;
    $cr2 = 0;
    $cr3 = 0;
    $cr4 = 0;
    $totalAlternatif = count($data_criteria);
    $N = (-1 / log($totalAlternatif));

    $hasil = array();
    foreach ($data_criteria as $row) {
        $hasil[] = array(
            $cr1 += $row->kriteria_1 / $criteria1 * log($row->kriteria_1 /
$criteria1),
            $cr2 += $row->kriteria_2 / $criteria2 * log($row->kriteria_2 /
$criteria2),
            $cr3 += $row->kriteria_3 / $criteria3 * log($row->kriteria_3 /
$criteria3),
            $cr4 += $row->kriteria_4 / $criteria4 * log($row->kriteria_4 /
$criteria4),
        );
    }
    $nilaitotal_ej = ((1 - ($N * $cr1)) + (1 - ($N * $cr2)) + (1 - ($N * $cr3)))
}

```

```

+ (1 - ($N * $cr4));

$w_c1 = ((1 - ($N * $cr1)) / $nilaitotal_ej);
$w_c2 = ((1 - ($N * $cr2)) / $nilaitotal_ej);
$w_c3 = ((1 - ($N * $cr3)) / $nilaitotal_ej);
$w_c4 = ((1 - ($N * $cr4)) / $nilaitotal_ej);
$total = (($w_c1) + ($w_c2) + ($w_c3) + ($w_c4));
// print_r($total);
$data = array(
    'c1' => $w_c1,
    'c2' => $w_c2,
    'c3' => $w_c3,
    'c4' => $w_c4,
    'total' => $total,
    'tipe' => $this->input->post('parameter'),
    'id_user' => $this->session->userdata('user_id'),
    '_view' => 'entropy/hasil'
);
$this->load->view('layouts/main', $data);
}
}

```

Setelah controller dibuat langkah selanjutnya buka file Entropy_model yag terdapat pada folder models kemudian isikan code model berikut:

```

<?php

class Entropy_model extends CI_Model

{
    public function get_criteria($where)

    {
        $this->db->select('*');

        $query = $this->db->get_where('tbl_alternatif', array('tipe_alternatif' =>

```

```

$where));

    return $query->result();

}

public function sumdata($where)

{
    $this->db->select_sum('kriteria_1', 'c1');

    $this->db->select_sum('kriteria_2', 'c2');

    $this->db->select_sum('kriteria_3', 'c3');

    $this->db->select_sum('kriteria_4', 'c4');

    $query = $this->db->get_where('tbl_alternatif', array('tipe_alternatif' =>
$where));

    return $query;

}
}

```

Pada controller tersebut berisikan dua fungsi yaitu fungsi untuk mengambil data berdasarkan parameter, kemudian fungsi untuk mengambil data atau nilai total dari setiap baris keriteria berdasarkan parameter

Jika model telah di buat selanjutnya buka file index.php yang terdapat pada folder entropy pada sub direktori views. Pada file tersebut isikan code berikut:

```
<div class="content-header">
```

```

<div class="container-fluid">
    <div class="row mb-2">
        <div class="col-sm-6">
            <h1 class="m-0 text-dark">Proses Entropy</h1>
        </div><!-- /.col -->
    </div><!-- /.row -->
</div><!-- /.container-fluid -->
</div>

<section class="content">
    <div class="container-fluid">
        <div class="card card-primary">
            <div class="card-header">
                <h3 class="card-title"></h3>
            </div>
            <!-- /.card-header -->
            <!-- form start -->
            <?php echo form_open('proses_entropy/proses'); ?>
            <div class="card-body">
                <label for="parameter" class="control-label"><span class="text-danger">*</span>Tipe Alternatif pada tabel alternatif</label>
                <div class="form-group">
                    <input type="text" name="parameter" value="<?php echo $this->input->post('parameter'); ?>" class="form-control" id="parameter" required />
                    <span class="text-danger"><?php echo form_error('parameter'); ?></span>
                </div>
            </div>
            <div class="card-footer">
                <button type="submit" class="btn btn-primary">Submit</button>
            </div>
            <?php echo form_close(); ?>
        </div><!-- /.card -->
    </div><!-- /.container-fluid -->
</section>

```

Code tersebut merupakan form untuk mengirim parameter berupa tipe alternatif yang di gunakan sebagai batasan untuk melakukan perhitungan

entropy. Baik itu untuk mengambil data kriteria maupun data nilai total dari kriteria.

Untuk tampilan form parameter seperti berikut:



Selanjutnya masukan code berikut pada file hasil.php yang terdapat pada sub direktori view pada folder entropy, code tersebut bertujuan untuk menampilkan data hasil perhitungan entropy kemudian menyimpannya ke database sistem.

```
<div id="container-fluid">
    <h1>Entropy</h1>
    <div class="container-fluid">
        <div class="row">
            <div class="col-md-12">
                <div class="card">
                    <!-- /.card-header -->
                    <div class="card-body">
                        <table class="table table-bordered">
```

```

<thead>
    <tr>
        <th style="width: 10px">#</th>
        <th>Nama Kriteria</th>
        <th>Nilai Bobot masing-masing kriteria</th>
    </tr>
</thead>
<?php echo form_open('tbl_bobot/add'); ?>
<tbody>
    <tr>
        <td>1.</td>
        <td>Keriteria 1</td>
        <td>
            <div class="col-md-12">
                <div class="form-group">
                    <input type="text" name="bobot_keriteria1"
value="<?= $c1 ?>" class="form-control" id="bobot_keriteria1" readonly/>
                    <span class="text-danger"><?php echo
form_error('bobot_keriteria1'); ?></span>
                </div>
            </div>
        </td>
    </tr>
    <tr>
        <td>2.</td>
        <td>Keriteria 2</td>
        <td>

```

```
<div class="col-md-12">
    <div class="form-group">
        <input type="text" name="bobot_keriteria2"
value="<?=$c2 ?>" class="form-control" id="bobot_keriteria2" readonly/>
        <span class="text-danger"><?php echo
form_error('bobot_keriteria2'); ?></span>
    </div>
</div>
</td>
</tr>
<tr>
    <td>3.</td>
    <td>Keriteria 3</td>
    <td>
        <div class="col-md-12">
            <div class="form-group">
                <input type="text" name="bobot_keriteria3"
value="<?=$c3 ?>" class="form-control" id="bobot_keriteria3" readonly/>
                <span class="text-danger"><?php echo
form_error('bobot_keriteria3'); ?></span>
            </div>
        </div>
        </td>
    </tr>
    <tr>
        <td>4.</td>
        <td>Keriteria 4</td>
        <td>
```

```

<div class="col-md-12">
    <div class="form-group">
        <input type="text" name="bobot_keriteria4"
value="<?=$c4 ?>" class="form-control" id="bobot_keriteria4" readonly/>
        <span class="text-danger"><?php echo
form_error('bobot_keriteria4'); ?></span>
    </div>
</div>
<div class="col-md-12">
    <div class="form-group">
        <input type="text" name="id_user"
value="<?=$id_user ?>" class="form-control" id="id_user" hidden />
        <span class="text-danger"><?php echo
form_error('id_user'); ?></span>
    </div>
</div>
<div class="col-md-12">
    <div class="form-group">
        <input type="text" name="tipe" value="<?=
$tipe ?>" class="form-control" id="tipe" hidden />
        <span class="text-danger"><?php echo
form_error('tipe'); ?></span>
    </div>
</div>
</td>
</tr>
<tr>
<td>4.</td>

```

```

<th scope="col">Aksi</th>
<th scope="col"><input type="submit" name="submit" value="Simpan Data Entropy" class="btn btn-success"></th>
</tr>
</tbody>
<?php echo form_close(); ?>
</table>
</div>

```

Untuk tampilan code tersebut adalah sebagai berikut:

The screenshot shows a web application interface titled 'Entropy'. On the left is a dark sidebar menu with the following items:

- Dashboard** (selected)
- Data User
- Data Alternatif
- Proses Entropy
- Sign Out

The main content area has a title 'Entropy' and displays a table with the following data:

#	Nama Kriteria	Nilai Bobot masing-masing kriteria
1.	Keriteria 1	0.17065249438406
2.	Keriteria 2	0.052424227613369
3.	Keriteria 3	0.29694720074987
4.	Keriteria 4	0.4799760772527
	Aksi	Simpan Data Entropy

At the bottom of the page, there is a footer bar with the text 'Copyright © 2020 All rights reserved.' and 'Version 1.0'.