

Praktikum WEB

Studi Kasus Membuat Aplikasi SIAKAD Sederhana

```
298     $dst['offset_w'] = 0;  
299     $dst['offset_h'] = 0;  
300     unset($_TMP['image']);  
301 }  
302  
303 $dst['image'] = imagecreatetruecolor($dst['width'], $dst['height']);  
304 imagealphablending($dst['image'], FALSE);  
305 imagesavealpha($dst['image'], TRUE);  
306  
307 $transparent = imagecolorallocatealpha($dst['image'], 255, 255, 255, 0);  
308 imagefilledrectangle($dst['image'], 0, 0, $dst['width'], $dst['height'], $transparent);
```

```
324  
325  
326  
327 $bgcolor = imagecolorallocate($dst['image'], $params['r'], $params['g'], $params['b']);  
328 imagefill($dst['image'], 0, 0, $bgcolor);  
329 imagecopyresampled($dst['image'], $currimg, $dst_off_w, $dst_off_h, 0, 0, $dst['width'], $dst['height'], $currimg, $dst_off_w, $dst_off_h, $currimg['width'], $currimg['height']);  
330 $dst['width'] = $params['height'];  
331 $dst['height'] = $params['width'];  
332  
333 }  
334  
335  
336 $dst['type'] = $info[2];  
337  
338 switch ($dst['type'])  
339 {  
340     case 1:  
341         imagetruecolortopalette($src['image'], FALSE, 256);  
342         imagegif($dst['image'], $dst['file']);  
343         break;
```

Daftar Isi

BAB 1 PERSIAPAN.....	1
1.1 Chrome Web Browser (Google Chrome)	1
1.2 Text Editor dan IDE.....	1
1.3 XAMPP.....	4
BAB 2 CODEIGNITER.....	5
2.1 Apa itu CodeIgniter	5
2.2 Struktur Folder CodeIgniter	6
2.3 Model View Controller	7
2.4 Memulai Konfigurasi dan Membuat Say Hello !	9
BAB 3 Membuat CRUD CodeIgniter	15
3.1 Membuat Database	15
3.2 Menampilkan Data Menggunakan MVC.....	18
3.3 Simpan Data	23
3.4 Edit Data.....	25
3.5 Hapus Data	27
BAB 4 Studi Kasus Membuat Aplikasi SIAKAD Sederhana	29
4.1 Konfigurasi CodeIgniter.....	31
4.2 Membuat Login	33
4.3 Membuat Home	38
4.4 Membuat Jurusan CRUD	45
4.4.1 Membuat Tampil Data	45
4.4.2 Membuat Tambah Data	49
4.4.3 Mengedit dan Menghapus Data Jurusan	52
4.4.4 Membuat Laporan Dalam Bentuk PDF.....	54
4.4.5 Membuat Laporan Dalam Bentuk Excel.....	56
4.4.6 Membuat Chart (Grafik).....	57
4.5 Membuat Matakuliah CRUD	60
4.5.1 Membuat Tampil Data	60
4.5.2 Membuat Tambah Data	63

4.5.3 Mengedit dan Menghapus Data Matakuliah	67
BAB 5 Integrasi CRUD Generator Harviacode	68

BAB 1

PERSIAPAN

Sebelum kita mempelajari bagaimana membuat web menggunakan PHP dan Framework CodeIgniter, ada beberapa persiapan yang perlu dilakukan. Seperti penginstallan aplikasi-aplikasi yang dibutuhkan dan aplikasi-aplikasi pendukung dalam mempelajarinya.

1.1 Chrome Web Browser (Google Chrome)



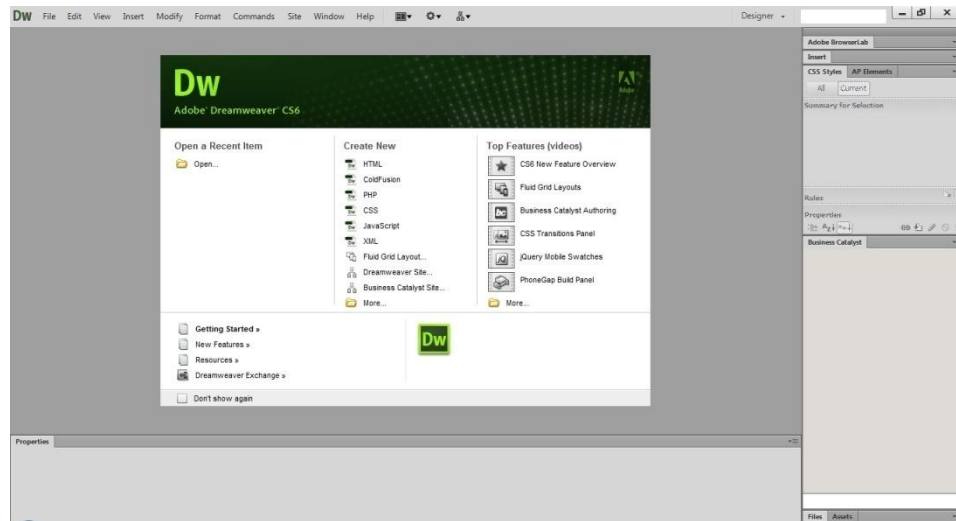
Web browser adalah perangkat utama yang akan kita gunakan untuk menampilkan halaman web yang pada dasarnya terbuat dari HTML dan CSS.

Google Chrome adalah web browser besutan Google yang memiliki beberapa keunggulan. Chrome bisa dikatakan browser yang sangat ringan, cepat dan kaya akan aplikasi/Add-on. Selain itu Chrome menggunakan engine3 Webkit, engine browser yang rata-rata mendukung fitur-fitur terbaru dari teknologi HTML dan CSS. Anda dapat mendownloadnya di <http://google.com/chrome>.

1.2 Text Editor dan IDE

Anda bebas menggunakan text editor apapun dalam praktikum ini

- **Dreamweaver**



Adobe Dreamweaver adalah program yang digunakan untuk membuat atau menyunting halaman web. Software Dreamweaver dikeluarkan oleh Adobe System. Aplikasi ini banyak digunakan oleh para programmer, desainer dan developer web dikarenakan kemudahan dalam penggunaannya, kelengkapan fiturnya dan juga dukungannya terhadap teknologi terkini. Adobe Dreamweaver menyediakan fitur editor WYSIWYG (*What You See is What You Get*) atau dalam bahasa kesehariannya disebut Design View. Maksudnya adalah, tampilan hasil akhir web kita nanti akan sama dengan tampilan pada saat proses perancangan halaman web.

- Sublime

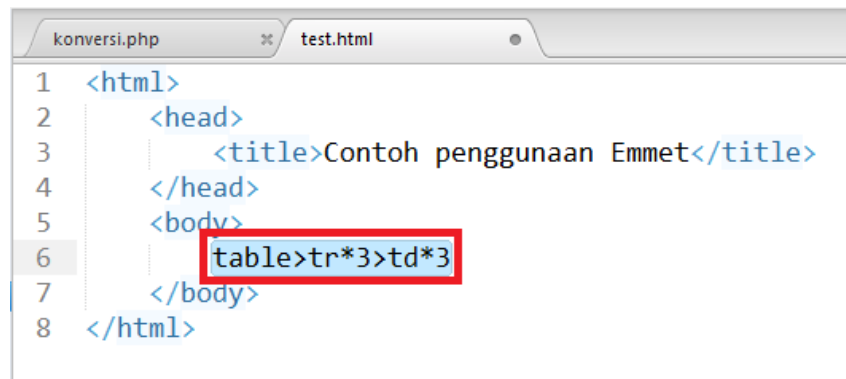


Sublime adalah text editor multiplatform yang kaya akan plugin untuk membantu developer dalam editing script, jika anda menggunakan sublime sebagai teks editor berikut ini plugin yang dapat ditambahkan agar text editor menjadi lebih handal.

1) Emmet (Zend Coding)

Emmet yang sebelumnya dikenal dengan Zen Coding merupakan toolkit yang digunakan untuk web development. Toolkit di sini bisa diartikan sebagai plugin atau add-on yang dipasang pada editor kode program. Dengan Emmet ini, anda dapat menulis HTML dan CSS secara cepat. Bagaimana penggunaannya ?

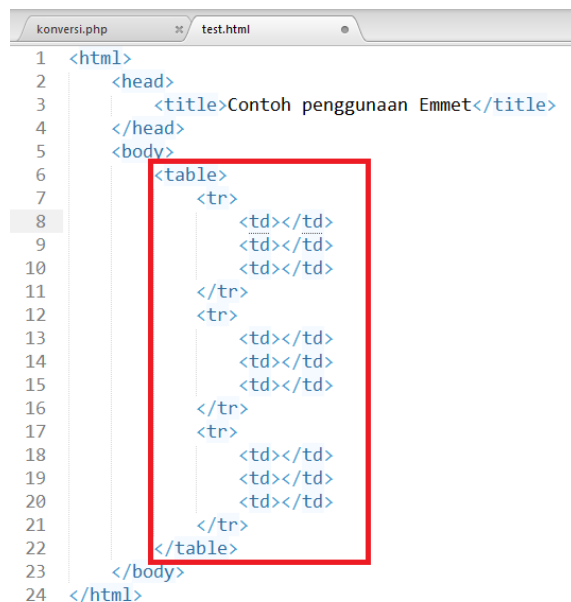
Sebagai contoh, misalkan saya ingin membuat tabel yang terdiri dari tiga baris dan tiga kolom. Dengan menggunakan Emmet, saya cukup menuliskan kode berikut :



```
1 <html>
2   <head>
3     <title>Contoh penggunaan Emmet</title>
4   </head>
5   <body>
6     <table>tr*3>td*3
7   </body>
8 </html>
```

The screenshot shows a code editor with two tabs: 'konversi.php' and 'test.html'. The 'test.html' tab is active, displaying the code above. Line 6, containing the Emmet shorthand '<table>tr*3>td*3', is highlighted with a red rectangular box.

Kemudian tekan tombol **Tab**



```
1 <html>
2   <head>
3     <title>Contoh penggunaan Emmet</title>
4   </head>
5   <body>
6     <table>
7       <tr>
8         <td></td>
9         <td></td>
10        <td></td>
11      </tr>
12      <tr>
13        <td></td>
14        <td></td>
15        <td></td>
16      </tr>
17      <tr>
18        <td></td>
19        <td></td>
20        <td></td>
21      </tr>
22    </table>
23  </body>
24 </html>
```

The screenshot shows the same code editor as before, but now the Emmet shorthand has been expanded into a full HTML table structure. The expanded code, from line 6 to line 22, is enclosed in a red rectangular box. The table consists of three rows and three columns, with each cell containing the text '</td>'.

Diatas adalah hasil dari Emmet yang kita gunakan.

2) All Auto Complete

Sejatinya, sublime menyediakan layanan auto complete berdasarkan kata yang pernah diketikkan dalam file yang sama. Tetapi, sublime kini menyediakan *tools*

dengan nama All AutoComplete yang merupakan plugin tambahan yang berguna untuk melengkapi teks secara otomatis dengan menampilkan berbagai kemungkinan sesuai kata yang kita ketik.

- **PHPStorm**

PHPStorm adalah IDE (*Integrated Development Environment*) yang sangat handal dan sangat baik dalam development aplikasi PHP. PHP Storm sudah terintegrasi dengan Emmet dan banyak fitur seperti Git yang sudah siap dikonfigurasi di dalamnya.

1.3 XAMPP



Fungsi **XAMPP** sendiri adalah **sebagai server** yang berdiri sendiri (localhost), yang terdiri beberapa program antara lain : **Apache HTTP Server, MySQL database**, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman **PHP dan Perl**. Nama XAMPP sendiri merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah untuk digunakan yang dapat menampilkan halaman web yang dinamis.

BAB 2

CODEIGNITER

2.1 Apa itu CodeIgniter

CodeIgniter adalah sebuah web application framework yang bersifat open source digunakan untuk membangun aplikasi php dinamis. Tujuan utama pengembangan CodeIgniter adalah untuk membantu developer untuk mengerjakan aplikasi lebih cepat daripada menulis semua code dari awal. CodeIgniter menyediakan berbagai macam library yang dapat mempermudah dalam pengembangan. CodeIgniter diperkenalkan kepada public pada tanggal 28 februari 2006. CodeIgniter sendiri dibangun menggunakan konsep Model-View-Controller development pattern. CodeIgniter sendiri merupakan salah satu framework tercepat dibandingkan dengan framework lainnya. Pada acara frOSCon (August 2008), pembuat php Rasmus Lerdorf mengatakan dia menyukai codeigniter karena dia lebih ringan dan cepat dibandingkan framework lainnya ("*because it is faster, lighter and the least like a framework.*")

Kenapa Menggunakan Framework?

Salah satu alasan mengapa orang menggunakan framework terutama dalam membangun sebuah aplikasi adalah kemudahan yang ditawarkan. Didalam sebuah framework biasanya sudah tersedia struktur aplikasi yang baik, standard coding (1), best practice (2) dan design pattern (3), dan common function (4). Dengan menggunakan framework kita dapat langsung fokus kepada business process yang dihadapi tanpa harus berfikir banyak masalah struktur aplikasi, standar coding dan lain-lain.

2.2 Struktur Folder CodeIgniter

Name	Date Modified	Size	Kind
application	Mar 15, 2011 1:54 PM	--	Folder
cache	Mar 15, 2011 1:54 PM	--	Folder
config	Mar 15, 2011 1:54 PM	--	Folder
controllers	Mar 15, 2011 1:54 PM	--	Folder
core	Mar 15, 2011 1:54 PM	--	Folder
errors	Mar 15, 2011 1:54 PM	--	Folder
helpers	Mar 15, 2011 1:54 PM	--	Folder
hooks	Mar 15, 2011 1:54 PM	--	Folder
language	Mar 15, 2011 1:54 PM	--	Folder
libraries	Mar 15, 2011 1:54 PM	--	Folder
logs	Mar 15, 2011 1:54 PM	--	Folder
migrations	Mar 15, 2011 1:54 PM	--	Folder
models	Mar 15, 2011 1:54 PM	--	Folder
third_party	Mar 15, 2011 1:54 PM	--	Folder
views	Mar 15, 2011 1:54 PM	--	Folder
index.html	Mar 15, 2011 1:53 PM	4 KB	HTML ...ument
system	Mar 15, 2011 1:54 PM	--	Folder
core	Mar 15, 2011 1:54 PM	--	Folder
database	Mar 15, 2011 1:54 PM	--	Folder
fonts	Mar 15, 2011 1:54 PM	--	Folder
helpers	Mar 15, 2011 1:54 PM	--	Folder
language	Mar 15, 2011 1:54 PM	--	Folder
libraries	Mar 15, 2011 1:54 PM	--	Folder
index.html	Mar 15, 2011 1:53 PM	4 KB	HTML ...ument
user_guide	Mar 15, 2011 1:54 PM	--	Folder
index.php	Mar 15, 2011 1:53 PM	8 KB	PHP
license.txt	Mar 15, 2011 1:53 PM	4 KB	Plain Text

Folder **application** : disinilah aplikasi yang akan kita bangun diletakkan.

- **Folder config** - tempat menyimpan semua file konfigurasi yang ada di dalam aplikasi, mulai dari database, router dan autoload aplikasi.
- **Folder controllers** - tempat menyimpan semua file controller.
- **Folder errors** - tempat menyimpan semua template error aplikasi.
- **Folder helpers** - tempat menyimpan helper-helper yang bukan berasal dari CI.
- **Folder hooks** - tempat menyimpan hook yang digunakan untuk mengubah alur fungsi dari core CI.
- **Folder language** - tempat menyimpan bahasa-bahasa yang akan di gunakan.
- **Folder libraries** - tempat menyimpan semua library buatan kita sendiri.
- **Folder models** - tempat menyimpan semua model.
- **Folder views** - tempat menyimpan semua file view aplikasi.

Folder **system** menyimpan semua file baik itu file aplikasi yang dibuat maupun core framework-nya.

- Folder **cache** tempat menyimpan semua cache yang dibuat caching library.
- Folder **database** tempat menyimpan semua driver database drivers dan class yang akan digunakan.
- Folder **fonts** tempat menyimpan semua font yang digunakan image manipulation library.
- Folder **helpers** tempat menyimpan semua helper core CI.
- Folder **language** tempat menyimpan semua language core CI.
- Folder **libraries** tempat menyimpan semua library core CI
- Folder **logs** tempat menyimpan semua logs generated oleh CI.
- Folder **plugin** tempat menyimpan semua plugin core CI.
- Folder **scaffolding** tempat menyimpan semua file yang berfungsi sebagai scaffolding.

Folder **user_guide**

- berisi userguide/manual penggunaan CI.

File index.php file yang akan handle semua request yang dilakukan oleh client.

2.3 Model View Controller

MVC adalah konsep dasar yang harus diketahui sebelum mengenal CodeIgniter. MVC adalah singkatan dari Model View Controller. MVC sebenarnya adalah sebuah pattern/teknik pemrograman yang memisahkan bisnis logic (alur pikir), data logic (penyimpanan data) dan presentation logic (antarmuka aplikasi) atau secara sederhana adalah memisahkan antara desain, data dan proses. Adapun komponen-komponen MVC antara lain:

1. Model

Model berhubungan dengan data dan interaksi ke database atau webservice. Model juga merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, file XML maupun webservice. Biasanya di dalam

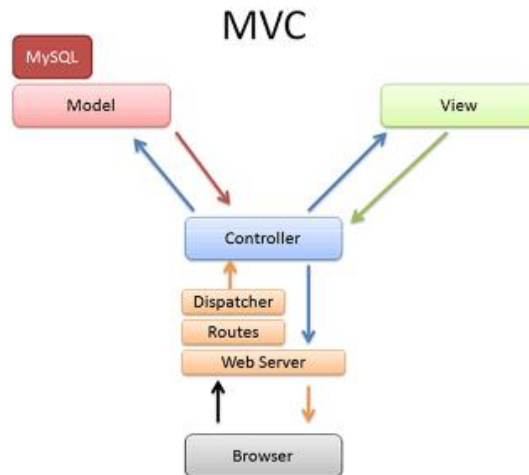
model akan berisi class dan fungsi untuk mengambil, melakukan update dan menghapus data website. Sebuah aplikasi web biasanya menggunakan basis data dalam menyimpan data, maka pada bagian Model biasanya akan berhubungan dengan perintah-perintah query SQL.

2. View

View berhubungan dengan segala sesuatu yang akan ditampilkan ke end-user. Bisa berupa halaman web, rss, javascript dan lain-lain. Kita harus menghindari adanya logika atau pemrosesan data di view. Di dalam view hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. View dapat dikatakan sebagai halaman website yang dibuat dengan menggunakan HTML dan bantuan CSS atau JavaScript. Di dalam view jangan pernah ada kode untuk melakukan koneksi ke basisdata. View hanya dikhususkan untuk menampilkan data-data hasil dari model dan controller

3. Controller

Controller bertindak sebagai penghubung data dan view. Di dalam Controller inilah terdapat class-class dan fungsi-fungsi yang memproses permintaan dari View ke dalam struktur data di dalam Model. Controller juga tidak boleh berisi kode untuk mengakses basis data karena tugas mengakses data telah diserahkan kepada model. Tugas controller adalah menyediakan berbagai variabel yang akan ditampilkan di view, memanggil model untuk melakukan akses ke basis data, menyediakan penanganan kesalahan/error, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap input.



Jika dipetakan, alur kerja CodeIgniter akan tampak seperti gambar 3. Browser berinteraksi melalui controller. Controller-lah yang akan menerima dan membalas semua request dari browser. Untuk data maka controller akan meminta ke Model dan untuk UI/template akan meminta ke View. Jadi “Otak” dari aplikasi ada di controller, “Muka” aplikasi ada di view dan “Data” ada di model.

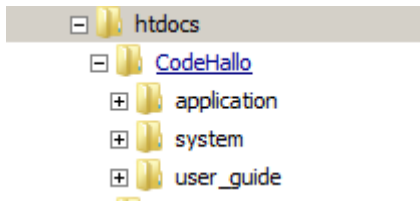
Ketika browser meminta sebuah halaman web maka router akan mencari controller mana yang harus menangani request tersebut. Setelah itu barulah si controller menggunakan model untuk mengakses data dan View untuk menampilkan data tersebut.

2.4 Memulai Konfigurasi dan Membuat Say Hello !

Sebelum kita membuat aplikasi dengan CodeIgniter tentunya kita harus memiliki file CodeIgniter yang bisa anda download di alamat <https://www.codeigniter.com/download>

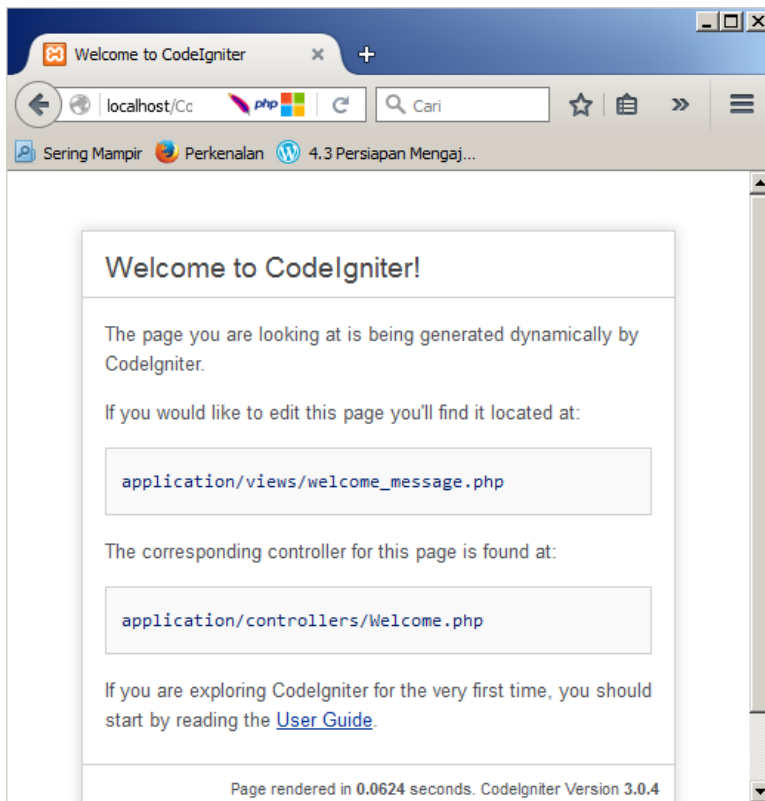
Setelah anda download Extract file tersebut ke folder **c:/xampp/htdocs** jika menggunakan xampp dan berinama project kita dengan nama **CodeHallo**.

Sehingga direktori yang kita punya adalah :

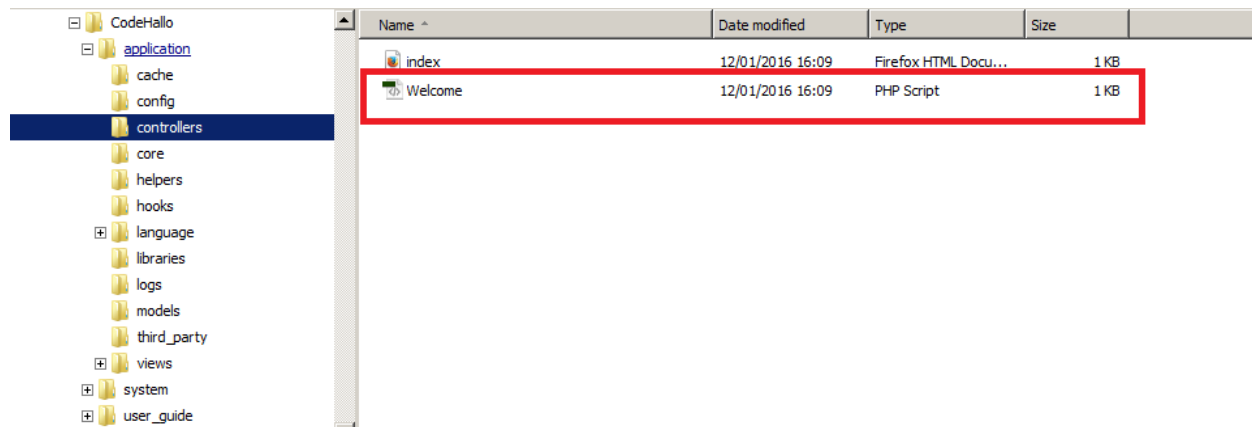


Nah sekarang coba buka browser dan ketikkan alamat aplikasi kita di **localhost/CodeHallo**

Maka akan muncul welcome message CodeIgniter seperti gambar berikut :



Welcome message ini adalah default dari CodeIgniter. File ini tersimpan di Folder ***CodeHallo/application/controllers/Welcome.php***

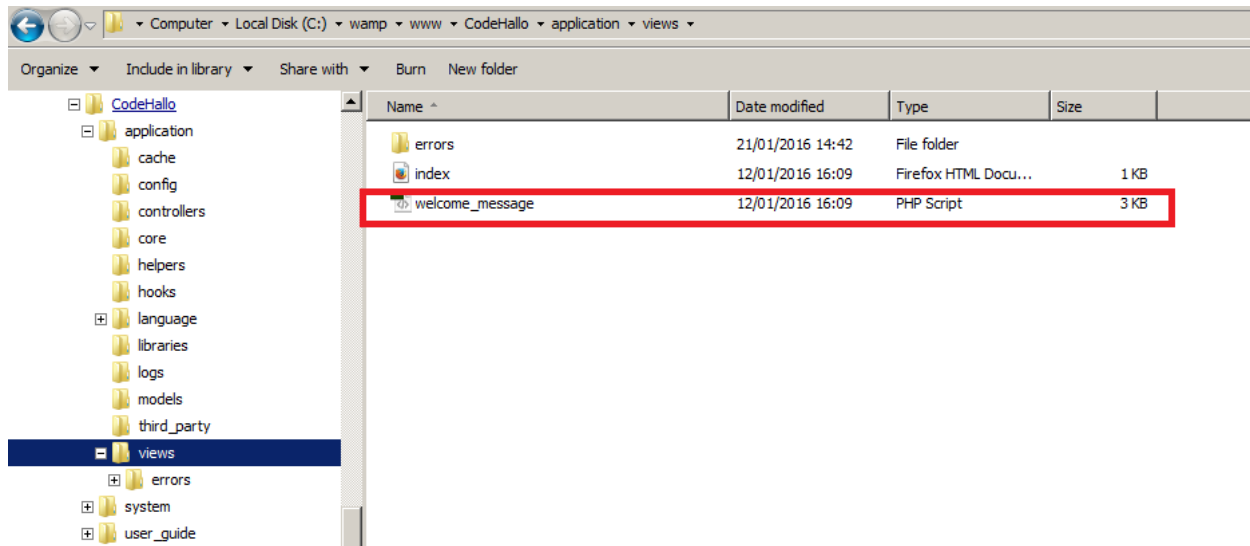


Berikut ini adalah isi dari file **welcome.php** (Buka dengan SublimeText atau editor Favoritmu)

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access
   allowed');
3.
4. class Welcome extends CI_Controller {
5.
6.     public function index()
7.     {
8.         $this->load->view('welcome message');
9.     }
10. }
```

- Baris 1 adalah tag pembuka pada script php.
- Baris 2 adalah **basepath** yang wajib ada pada setiap file **Controller CodeIgniter**
- Baris 4 adalah **nama Kelas** dan merupakan extends dari **CI_Controller**
- Baris 6 adalah **Function index()** yang akan memanggil file **welcome_message.php** pada folder views.



Mari kita lihat isi dari file **welcome_message.php** yang berada pada direktori **CodeHallo/application/views/welcome_message.php**

```

Welcome.php x welcome_message.php x
3  <?><!DOCTYPE html>
4  <html Lang="en">
5  <head>
67 </head>
68 <body>
69
70 <div id="container">
71   <h1>Welcome to CodeIgniter!</h1>
72
73   <div id="body">
74     <p>The page you are looking at is being generated dynamically by CodeIgniter.</p>
75
76     <p>If you would like to edit this page you'll find it located at:</p>
77     <code>application/views/welcome_message.php</code>
78
79     <p>The corresponding controller for this page is found at:</p>
80     <code>application/controllers/Welcome.php</code>
81
82     <p>If you are exploring CodeIgniter for the very first time, you should start by reading the <a href="user_guide/">
      User Guide</a>.</p>
83   </div>
84
85   <p class="footer">Page rendered in <strong>{elapsed_time}</strong> seconds. <?php echo (ENVIRONMENT === 'development')
      ? 'CodeIgniter Version <strong>' . CI_VERSION . '</strong>' : '' ?></p>
86 </div>
87 </body>
88 </html>
89

```

Isi dari file **welcome_message.php** adalah script html yang ditampilkan ketika kita membuka alamat **localhost/CodeHallo**.

Nah sekarang kita akan menambahkan Function **Hallo** pada File Controller **Welcome.php**

```

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

```

```

3.
4. class Welcome extends CI_Controller {
5.
6.     public function index()
7.     {
8.         $this->load->view('welcome_message');
9.     }
10.
11.     public function hallo()
12.     {
13.         $this->load->view('tampilkan_hallo');
14.     }
15. }

```

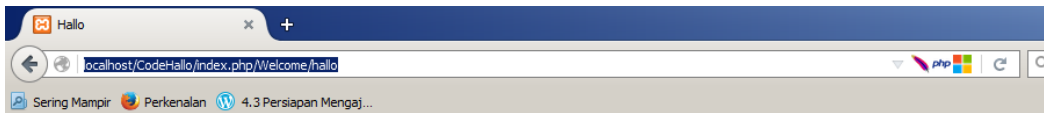
Kita menambahkan **function hallo()** yang akan memanggil file **tampilkan_hallo** pada folder **views**. Sekarang tambahkan atau buat file **tampilkan_hallo.php** pada folder **Views** :

```

1. <?php
2.     echo "<h2> Hallo Selamat belajar Codeigniter </h2>";
3. ?>

```

Simpan file **tampilkan_hallo.php** dan buka browser dengan alamat seperti berikut **localhost/CodeHallo/index.php/Welcome/hallo** :



Hallo Selamat Belajar CodeIgniter

Berikut penjelasan url yang kita akses :

`http://localhost/CodeHallo/index.php/Welcome/hallo`

The diagram illustrates the components of the URL `http://localhost/CodeHallo/index.php/Welcome/hallo`. Vertical lines connect each part of the URL to its corresponding label below:

- `http://localhost` is labeled "alamat local".
- `/CodeHallo` is labeled "Nama Project".
- `/index.php` is labeled "file index.php".
- `/Welcome` is labeled "Controller Welcome".
- `/hallo` is labeled "Function yg Diakses".

Untuk saat ini kita hanya menggunakan Controller dan View. Seperti yang kita bahas sebelumnya Kontroller bertugas menyediakan berbagai variabel yang akan ditampilkan di view, menyediakan penanganan kesalahan/error, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap input.

BAB 3

Membuat CRUD CodeIgniter

Sebelum membuat project yang besar kita akan membuat aplikasi CRUD terlebih dahulu, apa itu CRUD ??? CRUD adalah singkatan dari **Create, Read, Update, Delete** karena sejatinya untuk membuat aplikasi Sistem Informasi tidak bisa terlepas dari CRUD. Sebelum itu mari kita buat Databasenya terlebih dahulu.

3.1 Membuat Database

Buat database dengan nama **db_universitas** dan table dengan nama **tb_mahasiswa** berikut ini struktur tabelnya :

Nama Field	Tipe Data	Panjang	Keterangan
Npm	Varchar	8	Primary key
Nama	Varchar	50	
Alamat	Teks		

Tambahkan 2 record pada table **tb_mahasiswa** :

```
SELECT * FROM `tb_mahasiswa` LIMIT 50 (0.000 s) Edit
```

<input type="checkbox"/> Modify	npm	nama	alamat
<input type="checkbox"/> edit	11263074	Yenni Fatya Suci	Batulicin
<input type="checkbox"/> edit	11630452	Yogi Mahendri	Wanasari, Catur

Setelah kita menambahkan 2 buah record, berikutnya kita akan membuat konfigurasi pada CodeIgniter agar dapat berkomunikasi dengan database :

- Buka file **CodeHallo/application/config/database.php**

```
1. ....
2.
3. $db['default'] = array(
4.     'dsn' => '',
5.     'hostname' => 'localhost',
6.     'username' => 'root',
7.     'password' => '',
```

```

8.     'database' => 'db_universitas',
9.     'dbdriver' => 'mysqli',
10.    'dbprefix' => '',
11. ....

```

Hostname = **localhost**, username= **root**, database = **db_universitas**, jika pada database anda menggunakan password isikan passwordnya sesuai dengan database yang kalian miliki.

Catatan :

Kita bisa melihat bahwa konfigurasi yang kita buat semuanya disimpan kedalam array dengan variable bernama **\$db**. ☺

Sekarang kita akan menampilkan datanya pada browser.

1. Buat file baru dengan nama **crud.php** pada direktori **/application/controllers/crud.php** dan buat kelasnya seperti berikut :

```

2. <?php
3. defined('BASEPATH') OR exit('No direct script access allowed');
4.
5. class Crud extends CI_Controller
6. {
7.     //code ditulis disini
8. }

```

Lalu tambahkan coding untuk menampilkan data seperti berikut :

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed');
3.
4. class Crud extends CI_Controller
5. {
6.     public function index()
7.     {
8.         $data = $this->db->query('select*from tb_mahasiswa');
9.         foreach ($data->result_array() as $d)
10.        {
11.            echo "NPM = ".$d['npm']."<br>";
12.            echo "Nama Mahasiswa = ".$d['nama']."<br>";
13.            echo "Alamat = ".$d['alamat']."<br><hr>";
14.        }
15.    }
16. }

```

- Baris **8** adalah metode untuk melakukan query database lalu disimpan pada variable ***\$data***
- Baris **9** data tersebut di ***foreach*** kan lalu dibuat alias dan disimpan pada variable ***\$d***
- Baris **10** data tersebut ditampilkan menggunakan echo dan memanggil data menggunakan variable ***\$d['fieldname']***. Tag html ***</br>*** adalah break agar data ditampilkan tidak 1 baris.
- Jalankan pada browser dengan alamat ***localhost/CodeHallo/index.php/crud***

A PHP Error was encountered

Severity: Notice

Message: Undefined property: Crud::\$db

Filename: controllers/crud.php

Line Number: 8

Backtrace:

File: C:\wamp\www\CodeHallo\application\controllers\crud.php

Line: 8

Function: _error_handler

File: C:\wamp\www\CodeHallo\index.php

Line: 292

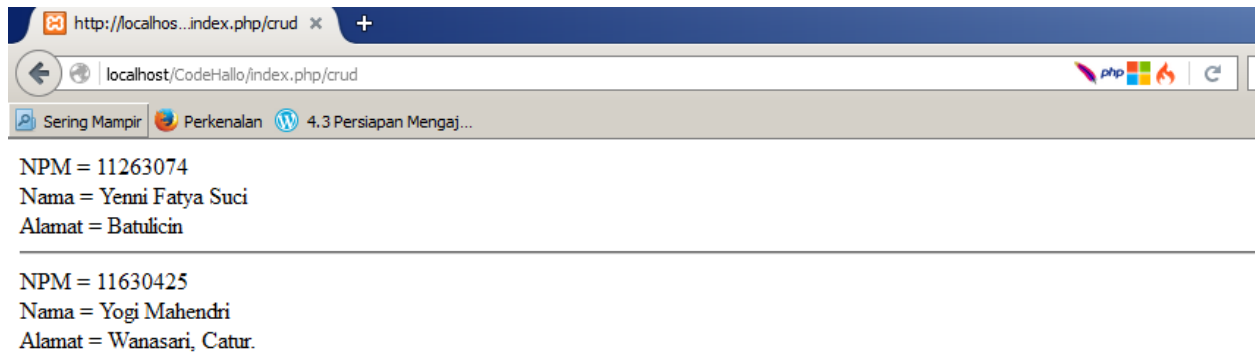
Function: require_once

- Bila error seperti ini terjadi, ini disebabkan oleh variable ***\$db*** pada file ***database.php*** belum dikenali saat kita memanggil ***\$db*** pada controller.
- Solusinya adalah buka file ***/applications/config/autoload.php*** pada bagian autoload libraries tambahkan seperti berikut .

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

- Simpan dan refresh browser.
- Dengan begini variable ***\$db*** sudah dapat dikenali saat kita panggil, ini berlaku untuk library-library yang lain jika ingin menggunakannya maka harus dimuat (di definisikan)

pada **autoload.php** terlebih dahulu. Berikut tampilan setelah ditambahkan library pada autoload.



3.2 Menampilkan Data Menggunakan MVC

Sebelumnya kita hanya menggunakan Controller, untuk membuat query dan sekaligus menampilkan datanya, padahal konsep MVC sudah menjelaskan bahwa kita harus memisahkan antara **model** (database query), **view** (Menampilkan data), **Controller** (Memproses data, parsing, dll).

Oleh karena itu kita akan mencoba menampilkan data menggunakan konsep MVC yakni memisahkan antara Model View dan Controller.

- Tambahkan file baru, beri nama **/application/models/model_crud.php** dan simpan pada folder **models**.

```
1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed');
3.
4. class Model_crud extends CI_Model
5. {
6.     //penulisan code disini
7. }
```

- Kita lihat bahwa script yang kita buat hampir sama dengan controller tapi yang membedakan adalah Extends **CI_Model** dan **CI_Controller**. Tambahkan kodenya menjadi seperti berikut

```

1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed');
3.
4. class Model_crud extends CI_Model
5. {
6.     public function AmbilDataMahasiswa($where="")
7.     {
8.         $data = $this->db->query('select*from tb_mahasiswa
9.         '.$where);
10.        return $data->result_array();
11.    }

```

- Baris 6 kita tambahkan parameter **\$where** ini digunakan acuan pada saat nanti kita mengedit atau menghapus data.
- Baris 8 bisa kita lihat bahwa query database disimpan pada variable **\$data** dan **return** dalam **array**.

Berikutnya buat file baru dengan nama **view_crud.php** dan simpan pada direktori **/applications/views/view_crud.php** kita akan menampilkan datanya menggunakan table.

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Data Mahasiswa</title>
</head>
<body>
    <a href="#">Tambah Data</a>
    <table border="1" style="border-collapse: collapse"; width="50%" >
        <tr style="background: grey">
            <th>NPM</th>
            <th>Nama</th>
            <th>Alamat</th>
            <th>Aksi</th>
        </tr>
        <tr>
            <td>11630019</td>
            <td>Adutya Darmawan</td>
            <td>Veteran Km.3</td>
            <td> <a href="#">Edit</a> | <a href="#">Hapus</a> </td>
        </tr>
    </table>
</body>
</html>

```

- Sekarang kita kembali ke file **application/controllers/crud.php** dan ubah menjadi seperti berikut :

```
1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed');
3.
4. class Crud extends CI_Controller
5. {
6.     public function index()
7.     {
8.         $data=$this->model_crud->AmbilDataMahasiswa();
9.         $this->load->view('tampilkan_data', array('data'
=> $data));
10.     }
11. }
```

- Baris ke **8** **\$data** menyimpan hasil query dari **model_crud.php** dan menampungnya pada **\$data** lalu melemparkan datanya ke **tampilkan_data.php** menggunakan array.
- Sekarang coba buka browser dan akan tampil data error seperti berikut :

A PHP Error was encountered

Severity: Notice

Message: Undefined property: Crud::\$model_crud

Filename: controllers/crud.php

Line Number: 8

Backtrace:

File: C:\xampp\htdocs\Codehallo\application\controllers\crud.php

Line: 8

Function: _error_handler

File: C:\xampp\htdocs\Codehallo\index.php

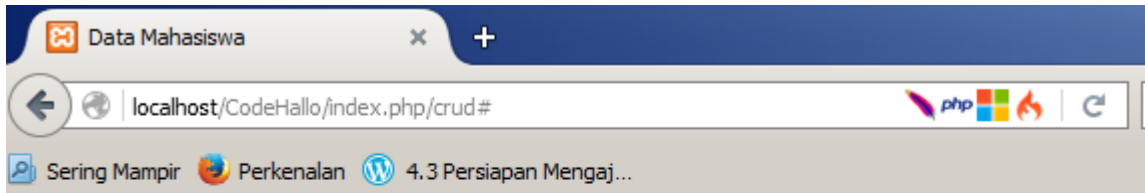
Line: 315

Function: require_once

- Lihat pesan error yang muncul pada browser, ini terjadi karena **model_crud.php** belum dikenali pada aplikasi tersebut. Caranya buka file **/application/config/autoload.php** pada baris **135** ubah menjadi seperti berikut :

```
$autoload['model'] = array('model_crud');
```

- Pada baris **135** kita menambahkan **'model_crud'** agar dapat dikenali oleh controller.
- Refresh Browser



Tambah Data

NPM	Nama	Alamat	Aksi
11630019	Aditya Darmawan	Veteran Km. 3	Edit Hapus

- Data yang ditampilkan pada browser adalah data yang kita tulis pada table html, data ini tidak ada dalam database. Ini kita gunakan untuk membuktikan bahwa semua Model, View, Controller yang kita buat tidak mengalami Error. Kali ini kita akan mencoba mengambil datanya dari variable **\$data** pada controller.
- Ubah file **/application/views/tampilkan_data.php** menjadi seperti berikut :

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.     <meta charset="UTF-8">
5.     <title>Data Mahasiswa</title>
6. </head>
7. <body>
8.     <a href="<?php echo base_url(). 'index.php/crud/add_data';
9.     ?>">Tambah Data</a>
10.     <table border="1" style="border-collapse: collapse; width: 50%" >
11.         <tr style="background: grey;">
12.             <th>NPM</th>
13.             <th>Nama</th>
14.             <th>Alamat</th>
15.             <th>Aksi</th>
16.         </tr>
17.         <?php foreach ($data as $d) { ?>
18.             <tr>
19.                 <td><?php echo $d['npm']; ?></td>
20.                 <td><?php echo $d['nama']; ?></td>
21.                 <td><?php echo $d['alamat']; ?></td>
22.                 <td>
```

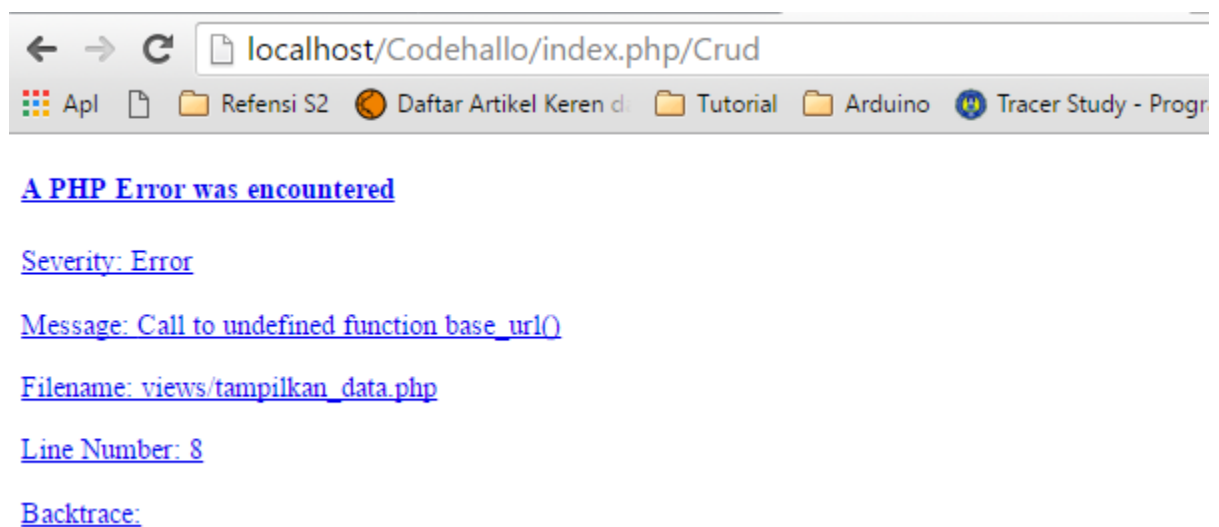


```

23.         <a href="<?php echo
    base_url()."index.php/crud/edit_data/" . $d['npm']; ?>">Edit</a> |
24.         <a href="<?php echo
    base_url()."index.php/crud/delete_data/" . $d['npm']; ?>">Hapus</a>
    </td>
25.     </tr>
26.
27.     <?php } ?>
28. </table>
29. </body>
30. </html>

```

Kemudian jalankan pada browser, bila terjadi error seperti dibawah ini,



Pada file **/application/config/config.php** tambahkan syntax `base_url()` nya seperti berikut

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

Lalu tambahkan pula url pada file **/application/config/autoload.php** pada helper seperti berikut

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

Refresh dan jalankan kembali



Tambah Data

NPM	Nama	Alamat	Aksi
11263074	Yenni Fatya Suci	Batulicin	Edit Hapus
11630425	Yogi Mahendri	Wanasari, Catur.	Edit Hapus

- Kita sudah berhasil menampilkan data yang ada pada database ke browser menggunakan konsep **MVC**. Berikutnya kita akan menambahkan fungsi edit, hapus, dan simpan.

3.3 Simpan Data

- Terlebih dahulu buat modelnya dengan nama function **MasukkanData()** pada file **/application/models/model_crud.php**

```

1. public function MasukkanData($tableName, $data)
2. {
3.     $hasil = $this->db->insert($tableName, $data);
4.     return $hasil;
5. }

```

- Berikutnya tambahkan 2 function pada file **/application/controllers/crud.php** seperti berikut

```

1. public function tambah_data()
2. {
3.     $this->load->view('form_tambah_data');
4. }
5.
6. public function simpanData()
7. {
8.     $npm = $_POST['npm'];
9.     $nama = $_POST['nama'];
10.    $alamat = $_POST['alamat'];
11.
12.    $data_insert = array(
13.        'npm' => $npm,
14.        'nama' => $nama,
15.        'alamat' => $alamat,
16.    );
17.

```

```

18.     $hasil = $this->model_crud->MasukkanData('tb_mahasiswa',
    $data_insert);
19.
20.     if ($hasil>=1) {
21.         $this->session->set_flashdata('pesan', "Tambah Data
    Sukses");
22.         redirect('crud/index');
23.     }else{
24.         echo "<h2>Menambahkan data gagal</h2>";
25.     }
26. }

```

- Pada baris 1-4 adalah function yang digunakan untuk memanggil **form_tambah_data.php** pada **application/views/form_tambah_data.php**
- Baris **8-10** adalah inisialisasi variable dan pada baris **12** data dibungkus kedalam array dan ditampung pada variable **\$hasil**
- Baris **20** adalah pengecekan hasil query insert jika data berhasil di masukkan ke dalam database maka akan return jika hasil besar dari sama dengan 1 maka tampilkan flash data **Tambah Data Sukses**. Dan akan di redirect ke halaman index.
- Baris **23** jika sebaliknya hasil sama dengan 0 maka tampilkan pesan **Insert Data gagal**.

Berikut adalah isi dari **form_tambah_data.php**

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Tambah Data</title>
</head>
<body>
    <form method="POST" action="<?php echo
base_url(). "index.php/Crud/simpanData"; ?>">
        <table>
            <tr>
                <td>NPM</td>
                <td><input type="text" name="npm"></td>
            </tr>
            <tr>
                <td>Nama</td>
                <td><input type="text" name="nama"></td>
            </tr>
            <tr>
                <td>Alamat</td>
                <td><textarea name="alamat"></textarea></td>
            </tr>
        </table>
    </form>

```

```

                <td></td>
                <td><input type="submit" name="btnSubmit"
value="Simpan"></td>
            </tr>
        </table>
    </form>
</body>
</html>

```

Jalankan aplikasinya dengan mengakses alamat **localhost/CodeHallo/index.php/crud**

3.4 Edit Data

Untuk edit data terlebih dahulu kita buat modelnya, buka **/application/models/model_crud.php** dan tambahkan function **UpdateData()** seperti berikut :

```

1. public function MasukkanData($tableName, $data)
2. {
3. $hasil = $this->db->insert($tableName, $data);
4. return $hasil;
5. }
6.
7. public function updateData($tableName, $data, $id)
8. {
9.     $hasil = $this->db->update($tableName, $data, $id);
10.    return $hasil;
11. }

```

- Pada baris **7** kita membuat function dengan nama **UpdateData()** yang mana menggunakan 3 parameter yaitu **\$tablename** (nama table), **\$data** (untuk menampung query), **\$id** (parameter seleksi).
- Setelah kita menambahkan model, saatnya kita menambahkan controllernya, buka file **/application/controllers/crud.php** tambahkan fungsi **edit_data()** seperti berikut :

```

1. public function edit_data($npm)
2. {
3.     $hasil=$this->model_crud->AmbilDataMahasiswa("where npm
= '$npm'");
4.     $data = array(
5.         "npm" => $hasil[0]['npm'],
6.         "nama" => $hasil[0]['nama'],
7.         "alamat" => $hasil[0]['alamat'],
8.     );
9.     $this->load->view('form edit data', $data);

```

```
10.         }
```

- Tambahkan function **UpdateData()** untuk mengambil perubahan data dan mengirimnya ke database seperti berikut:

```
1.     public function UpdateData()  
2.     {  
3.         $npm = $_POST['npm'];  
4.         $nama = $_POST['nama'];  
5.         $alamat = $_POST['alamat'];  
6.  
7.         $data_update = array(  
8.             'nama' => $nama,  
9.             'alamat' => $alamat,  
10.        );  
11.        $where = array('npm' => $npm);  
12.        $hasil = $this->model_crud->  
>updateData('tb_mahasiswa', $data_update, $where);  
13.        if ($hasil>=1)  
14.        {  
15.            $this->session->set_flashdata('pesan', "Tambah  
Data Sukses");  
16.            redirect('crud/index');  
17.        }else{  
18.            echo "<h2>Menambahkan data gagal</h2>";  
19.        }  
20.    }
```

- Tambahkan file **form_edit_data.php** pada folder **/application/views/form_edit_data.php** seperti berikut :

```
1. <!DOCTYPE html>  
2. <html lang="en">  
3. <head>  
4.     <meta charset="UTF-8">  
5.     <title>Tambah Data</title>  
6. </head>  
7. <body>  
8.     <form method="POST" action="<?php echo  
base_url(). "index.php/Crud/UpdateData"; ?>">  
9.         <table>  
10.            <tr>  
11.                <td>NPM</td>  
12.                <td><input type="text" name="npm"  
value="<?php echo $npm; ?>"></td>  
13.            </tr>  
14.            <tr>  
15.                <td>Nama</td>  
16.                <td><input type="text" name="nama"  
value="<?php echo $nama; ?>"></td>
```

```

17.         </tr>
18.     </tr>
19.         <td>Alamat</td>
20.         <td><textarea name="alamat"> <?php echo
    $alamat; ?></textarea></td>
21.     </tr>
22. </tr>
23.         <td></td>
24.         <td><input type="submit" name="btnSubmit"
    value="Simpan"></td>
25.     </tr>
26. </table>
27. </form>
28. </body>
29. </html>

```

3.5 Hapus Data

- Seperti biasa tambahkan modelnya terlebih dahulu pada file */application/models/model_crud.php* seperti berikut:

```

1.     public function hapusData($tableName, $id)
2.     {
3.         $hasil = $this->db->delete($tableName, $id);
4.         return $hasil;
5.     }

```

- Lalu tambahkan *function DeleteData()* pada controller *crud.php*

```

1.     public function DeleteData($npm)
2.     {
3.         $ambilData = array('npm' => $npm);
4.         $hasil = $this->model_crud->hapusData('tb_mahasiswa',
    $ambilData);
5.         if ($hasil>=1)
6.         {
7.             redirect('crud/index');
8.             $this->session->set_flashdata('pesan', "Hapus Data
    Sukses");
9.         }else{
10.             echo "<h2>Menambahkan data gagal</h2>";
11.         }
12.     }

```

Tambahkan coding action edit dan delete pada *application/views/tampilkan_data.php* seperti berikut :

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Data Mahasiswa</title>
</head>
<body>
    <a href="<?php echo base_url()."index.php/crud/tambah_data"; ?>">Tambah
Data</a>
    <table border="1" style="border-collapse:collapse; width:50%" >
        <tr style="background:grey;">
            <th>NPM</th>
            <th>Nama</th>
            <th>Alamat</th>
            <th>Aksi</th>
        </tr>
        <?php foreach ($data as $d) { ?>

        <tr>
            <td><?php echo $d['npm']; ?></td>
            <td><?php echo $d['nama']; ?></td>
            <td><?php echo $d['alamat']; ?></td>
            <td>
                <a href="<?php echo
base_url()."index.php/crud/edit_data/" . $d['npm']; ?>">Edit</a>
|
                <a href="<?php echo
base_url()."index.php/crud/DeleteData/" . $d['npm'];
?>">Hapus</a> </td>
            </tr>

            <?php } ?>
        </table>
</body>
</html>

```

Cobalah untuk mengupdate data, insert, ataupun delete data, bila error periksa kembali controller, konfigurasi, helper maupun library, apakah sudah terpasang dengan benar. 😊

BAB 4

Studi Kasus Membuat Aplikasi SIAKAD Sederhana

Pada bab ini kita akan membuat aplikasi siakad dan integrasi template menggunakan bootstrap, sebelum memulai ada baiknya anda menyediakan file-file seperti berikut :

1. Codeigniter Versi 2.2 atau yang lebih tinggi (<https://github.com/bcit-ci/CodeIgniter/tree/2.2-stable>)
2. Ace Template (<https://github.com/bopoda/ace>)
3. Rancangan Database dengan nama *siakad_db*

Berikut ini rancangan tabelnya,:

- 1) Tabel tbl_admin

Nama Field	Tipe Data	Panjang	Keterangan
Id_username	Int	3	PK, AI
Username	Varchar	50	Not null
Password	Varchar	150	Not null (MD5)
Nama_lengkap	Varchar	50	Not null

- 2) Tabel tbl_prodi

Nama Field	Tipe Data	Panjang	Keterangan
Kd_prodi	Varchar	10	Pk
Prodi	Varchar	50	Not null
Singkat	Varchar	10	Not null
Ketua_prodi	Varchar	60	Not null
Nik	Varchar	20	Not null
Akreditasi	Varchar	2	Not null
Tgl_insert	Datetime		Not null
Tgl_update	Datetime		Not null

- 3) Tabel tbl_mata_kuliah

Nama Field	Tipe Data	Panjang	Keterangan
------------	-----------	---------	------------

Kd_mk	Varchar	10	PK
Kd_prodi	Varchar	10	
Nama_mk	Varchar	50	
Sks	Int	6	
Smt	enum	'1','2','3','4','5','6','7','8'	
Aktif	Enum	'Ya','Tidak'	
Tgl_insert	Datetime		
Tgl_update	Datetime		

4) Tabel tbl_mahasiswa








Nama Field	Tipe Data	Panjang	Keterangan
Th_akademik	Varchar	15	
Npm	Varchar	8	pk
Kd_prodi	Varchar	15	
Nama_mhs	Varchar	50	
Jenis_kelamin	Integer	1	
Tempat_lahir	Varchar	50	
Tanggal_lahir	Date		
Alamat	Varchar	150	
Kota	Varchar	50	
Hp	Varchar	30	
Email	Varchar	50	
Nama_ayah	Varchar	50	
Nama_ibu	Varchar	100	
Alamat_ortu	Varchar	150	
Hp_ortu	Varchar	15	
Password	Varchar	50	
Status	Enum	'Aktif','Tidak'	
Tgl_insert	Datetime		
Tgl_update	Datetime		

5) Tabel tbl_dosen

Nama Field	Tipe Data	Panjang	Keterangan
Kd_dosen	Varchar	10	Pk
Kd_prodi	Varchar	10	
Nidn	Varchar	20	
Nama_dosen	Varchar	70	
Jenis_kelamin	Integer	1	
Tempat_lahir	Varchar	50	
Tanggal_lahir	Date		
Alamat	Varchar	70	
Hp	Varchar	15	
Pendidikan	Varchar	30	
Email	Varchar	50	
Prodi	Varchar	50	
Password	Varchar	50	
Tgl_insert	Datetime		
Status	Enum	'Aktif','Tidak'	
Tgl_update	Datetime		
Tgl_masuk	Date		

4.1 Konfigurasi CodeIgniter

Extract codeigniter ke folder **htdocs** bernama dengan **siakadtutorial** jangan lupa kopikan folder asset yang ada pada **ace template** ke root direktori codeigniter sehingga struktur folder aplikasi kita akan menjadi seperti berikut :

Name ^	Date modified	Type	Size	
 application	31/10/2015 11:14	File folder		
 assets	05/02/2015 10:21	File folder		
 system	31/10/2015 11:14	File folder		
 .travis.yml	31/10/2015 11:14	Text Document	1 KB	
 index	31/10/2015 11:14	YML File	1 KB	
 license	31/10/2015 11:14	PHP File	7 KB	
 README.md	31/10/2015 11:14	Text Document	3 KB	
		MD File	1 KB	

Buka folder project anda menggunakan editor favoritmu, kita akan mengkonfigurasi file ***application/config/autoload.php*** terlebih dahulu seperti berikut :

- 1) Tambahkan autoload library ***session*** dan ***database***

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

- 2) Tambahkan autoload helper ***url*** dan ***form***

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

Berikutnya tambahkan konfigurasi ***base_url*** dan ***encryption_key*** pada file ***/application/config/config.php*** seperti berikut :

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

.....

$config['encryption_key'] =
'edtrtybnmklojhuytrdcfgvhjkjhuygyvfcx';

.....
```

Encryption_key digunakan untuk mengenkripsi session atau cookies yang disimpan oleh browser.

Berikutnya yang perlu kita konfigurasi adalah pada file ***/application/config/database.php***

```

.....
$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => 'db_siakad',
    'database' => '',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
.....

```

Dan terakhir kita akan mengkonfigurasi bagian ***application/config/routes.php*** ubah di bagian ***default_controller*** menjadi ***'LoginController'*** seperti berikut :

```

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

```

Nah sekarang untuk tahapan konfigurasi sudah selesai berikutnya kita akan membuat controllernya untuk menghandle request dari user user.

4.2 Membuat Login

Buat file baru dengan nama ***Login.php*** dan simpan pada folder ***application/controllers/***

Login.php seperti berikut :

```

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

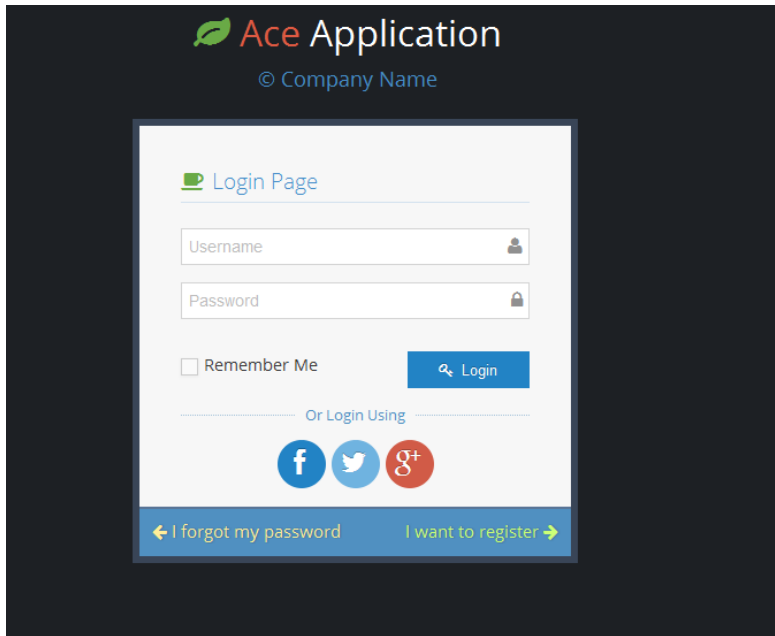
class Login extends CI_Controller {

    public function index()
    {
        $this->load->view('tampilan_login');
    }
}

```

Berikutnya kita buat file baru untuk viewnya dengan nama **tampilan_login.php** dan simpan pada direktori **/application/views/tampilan_login.php** pada kali ini kita akan mengambil dari **ace template**.

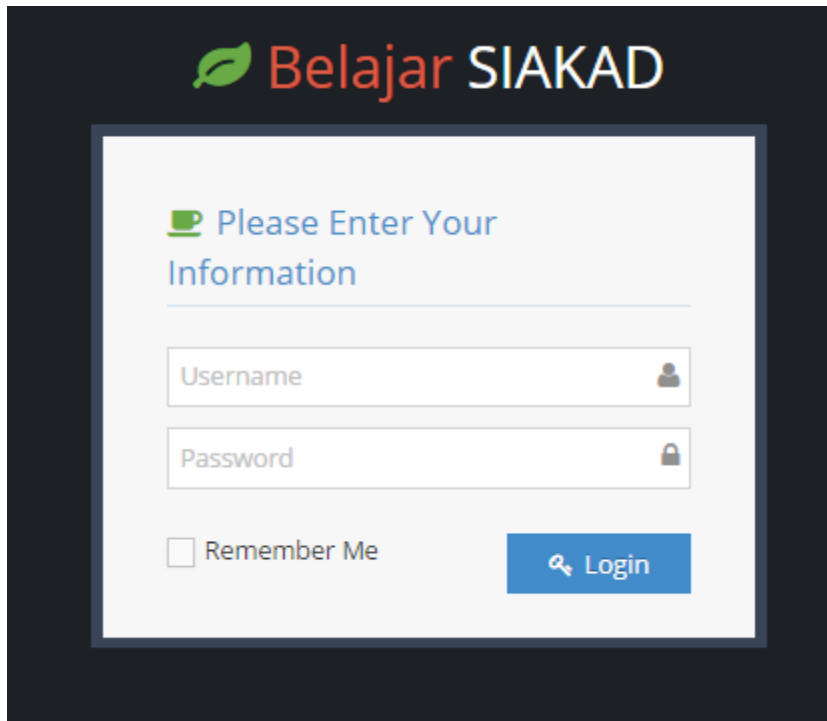
- Buka ace template dan buka file **login.html** copykan **semua code** yang ada di dalamnya ke **/application/views/tampilan_login.php**
- Lalu buka **localhost/siakadtutorial/** dan akan menghasilkan tampilan seperti berikut :



- Ini artinya anda berhasil memanggil **tampilan_login.php** dari controller.
- Berikutnya kita akan menambahkan **base_url** di setiap file css yang ada pada folder **asset**. Ini tujuannya agar suatu saat kita hosting tampilan tidak hancur. Buka file **tampilan_login.php** dan tambahkan script seperti berikut :

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

- Tambahkan pula untuk css yang lain, juga pada javascript
- Hilangkan pula bagian yang tidak diperlukan sehingga anda mendapatkan tampilan seperti berikut.



Jika kita klik tombol **login** belum terjadi respon apapun, ini karena kita belum memberikan login action pada form login tersebut, oleh karenanya tambahkan action seperti berikut pada form login :

```
<form method="POST" action="<?php echo base_url();
?>index.php/login/getlogin">
<fieldset>
<label>
    <span class="block input-icon input-icon-right">
        <input id="username" name="username" type="text" class="span12"
placeholder="Username" required=true />
        <i class="icon-user"></i>
    </span>
</label>
<label>
    <span class="block input-icon input-icon-right">
        <input id="password" name="password" type="password"
class="span12" placeholder="Password" required=true />
        <i class="icon-lock"></i>
    </span>
</label>
<div class="space"></div>
<div class="clearfix">
    <label class="inline">
        <input type="checkbox" />
        <span class="lbl"> Remember Me</span>
    </label>
```

```

        <button type="submit" class="width-35 pull-right btn btn-small btn-
primary">
            <i class="icon-key"></i>
            Login
        </button>
</div>
<div class="space-4"></div>
</fieldset>
</form>

```

- Pada bagian action terlihat kita akan mengakses controller **login** dan function **getlogin**.
- Untuk mengecek inputan user kita menggunakan **required=true** menandakan inputan tidak boleh kosong.
- Pada input **username** dan **password** kita menambahkan **id**, dan **name** yang digunakan untuk **post** data ataupun mengambil data.

Bisa anda coba menjalankan form login. Lihat adakah perubahannya dari sebelumnya ?

Nah sekarang saatnya menambahkan function **getLogin()** pada file **/application/controllers/login.php**

```

public function getlogin()
{
    $u = $this->input->post('username');
    $p = $this->input->post('password');
    $this->load->model('model_login');
    $this->model_login->getlogin($u,$p);
}

```

Pada baris ke 5 kita memanggil model_login dan pada baris berikutnya kita memanggil function **getlogin()** yang ada pada **model_login**.

Berikutnya kita buat **model_login.php** dan simpan pada folder **/application/models/model_login.php** berikut ini adalah scriptnya :

```

public function getlogin($u,$p)
{
    $pwd = md5($p);
    $this->db->where('username', $u);
    $this->db->where('password', $pwd);
    $query = $this->db->get('admin');
}

```

```

        if($query->num_rows()>0)
        {
            foreach ($query->result() as $row)
            {
                $sess = array('username' => $row -> username,
'nama_lengkap' => $row->nama_lengkap);
                $this->session->set_userdata($sess);
                redirect('home');
            }
        }
        else
        {
            $this->session->set_flashdata('info','maaf username
atau password salah');
            redirect('login');
        }
    }
}

```

- Function **getLogin()** menggunakan 2 parameter yaitu **\$u** dan **\$p**
- Pada baris 7 password dienkripsi menggunakan **md5**
- Baris 8-10 adalah **query** yang digunakan pada table **admins** menggunakan active record.
- Baris 11 adalah pengecekan menggunakan **num_rows()** jika **besar dari nol** maka data **ada(tersedia)** pada database, jika data sama dengan nol maka data tidak ada pada database.
- Pada baris 13 jika data ditemukan artinya besar dari sama dengan **nol** maka **username**, dan nama lengkap akan disimpan di **session**.
- Tambahkan coding berikut pada file **application/views/tampilan_login.php** dibawah tag <i>

```

<span class="block input-icon input-icon-right">
<input id="password" name="password" type="password" class="span12"
placeholder="Password" required=true />
<i class="icon-lock"></i>
<font color="red">
    <?php
        $info = $this->session->flashdata('info');
        if(!empty($info))
        {

```



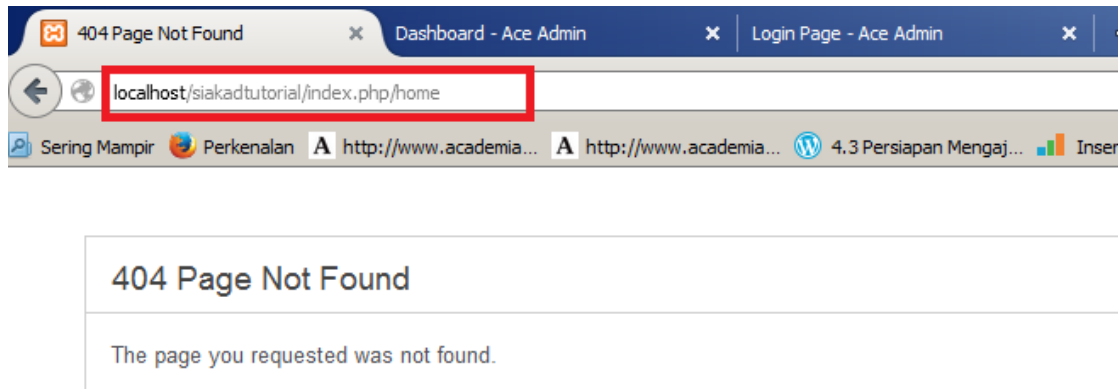
```

        echo $info;
    }
    ?>
</font>
</span>

```

Coba anda jalankan dan apa yang terjadi jika kondisi benar dan salah.

Jika benar maka akan menampilkan **404 Page Not Found**



Perhatikan pada url diatas, ia memanggil kontroller **home** sedangkan kita belum membuatnya, maka wajar jika menampilkan pesan **error 404 Page Not Found**. Halaman tidak ditemukan, hehe :P .

4.3 Membuat Home

Buat file baru dengan nama **home.php** dan simpan pada folder **application/controllers/home.php** seperti berikut :

```

1. <?php if ( ! defined('BASEPATH')) exit('No direct script
   access allowed');
2.
3. class Home extends CI_Controller {
4.
5.     public function index() {
6.         $this->load->view('tampilan_home');
7.     }
8. }

```

Berikutnya kita buat file baru untuk viewnya dengan nama **tampilan_home.php** dan simpan pada direktori **/application/views/tampilan_home.php** pada kali ini kita akan mengambil dari **ace template**.

- Buka ace template dan buka file **index.html** copykan **semua code** yang ada di dalamnya ke **/application/views/tampilan_home.php**

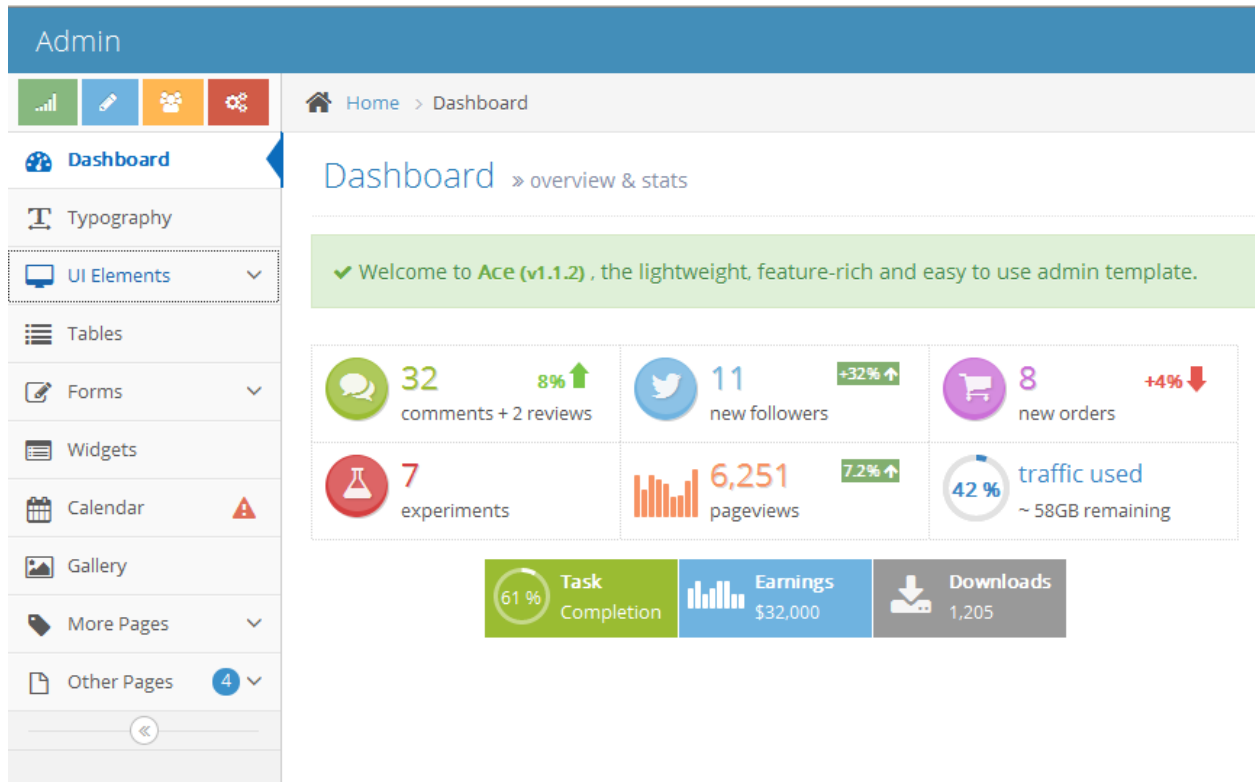
Lalu buka **localhost/siakadtutorial/index.php/home** dan akan menghasilkan tampilan seperti berikut :

[Admin](#)

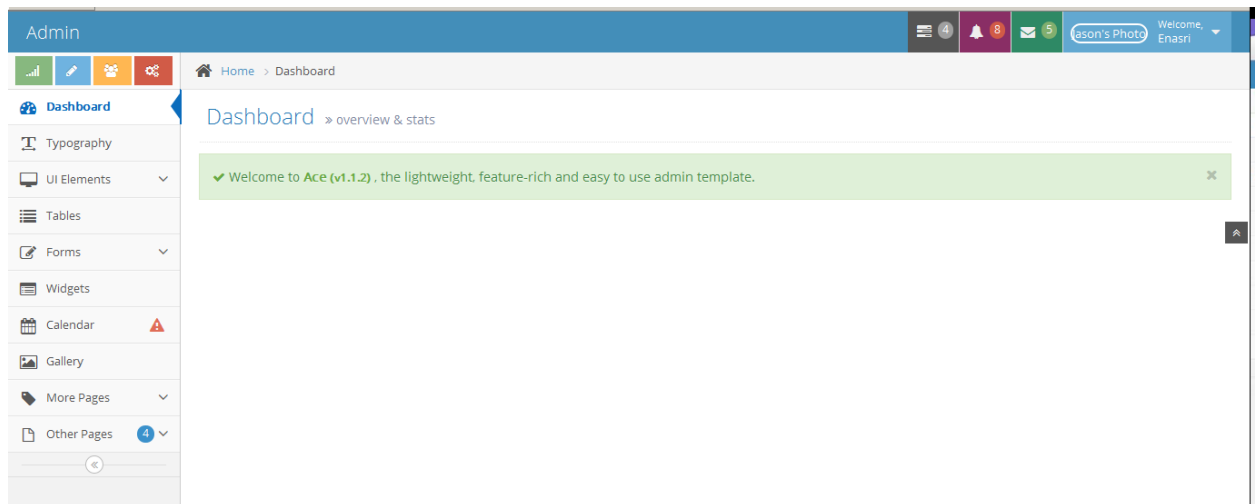
- [4](#)
 - 4 Tasks to complete
 - [Software Update 65%](#)
 - [Hardware Upgrade 35%](#)
 - [Unit Testing 15%](#)
 - [Bug Fixes 90%](#)
 - [See tasks with details](#)
- [8](#)
 - 8 Notifications
 - [New Comments +12](#)
 - [Bob just signed up as an editor ...](#)
 - [New Orders +8](#)
 - [Followers +11](#)
 - [See all notifications](#)
- [5](#)
 - 5 Messages
 - [Alex's Avatar Alex: Ciao sociis natoque penatibus et auctor ... a moment ago](#)
 - [Susan's Avatar Susan: Vestibulum id ligula porta felis euismod ... 20 minutes ago](#)
 - [Bob's Avatar Bob: Nullam quis risus eget urna mollis ornare ... 3:15 pm](#)
 - [See all messages](#)
- [Jason's Photo Welcome, Enasri](#)

Bagaimana dengan tampilannya ? hancur bukan ? Masih ingatkah kita pada poin 4.2 bahwa kita harus menambahkan **base_url()** disetiap link asset ?

Jika sudah anda tambahkan **base_url()** pada setiap link asset maka akan muncul tampilan seperti berikut :



Kemudian bersihkan static content yang ada pada, sehingga menghasilkan tampilan seperti berikut :



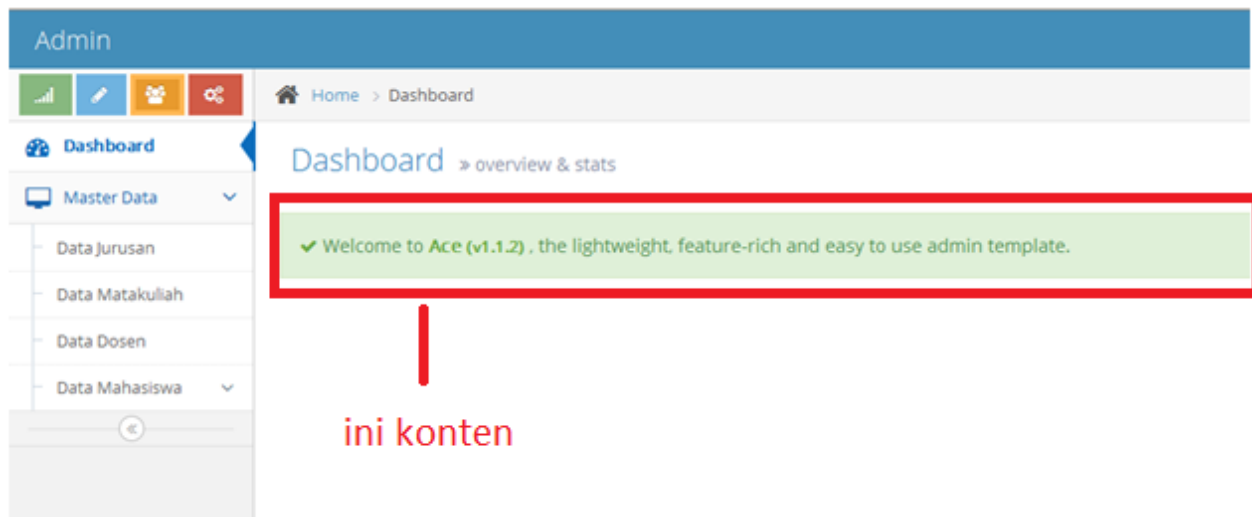
Nah sekarang bila kita lihat file **tampilan_home.php** menampung seluruh konten baik menu, judul aplikasi, dll. Kita akan mencoba memisahkan antara **menu** dan tampilan **konten** itu sendiri. Agar suatu saat bila ada kesalahan atau maintenance koding lebih mudah untuk kita mendebug aplikasi kita karena kode lebih rapih dan tidak menjadi 1 file.

Cari code berikut lalu cut `<ul class="nav nav-list">` sampai `<!-- /.nav-list -->`

- Buat file baru dengan nama **menu_navigasi.php** dan simpan pada folder views, pastekan coding yang tadi anda cut.
- Setelah itu pada **tampilan_home.php** masukkan code berikut di tempat anda cut code sebelumnya.

```
<!-- nav list -->
<?php $this->load->view('menu_navigasi'); ?>
<!-- end nav list -->
```

Lalu kembali pada file **menu_navigasi.php** sesuaikan menunya menjadi seperti berikut ini. Dan hilangkan static konten yang lain.



Nah setelah tadi kita memisahkan Menu, saatnya kita memisahkan konten dengan home, buat file baru dengan nama **tampilan_konten.php** dan simpan pada folder **views**. Selanjutnya, buka **tampilan_home.php** dan cut coding berikut pindahkan ke **tampilan_konten.php**

```
<!-- PAGE CONTENT BEGINS -->
isi yg disini di cut
<!-- PAGE CONTENT ENDS -->
```

Berikut isi **tampilan_konten.php**

```
<div class="alert alert-block alert-success">
  <button type="button" class="close" data-
dismiss="alert">
    <i class="ace-icon fa fa-times"></i>
  </button>

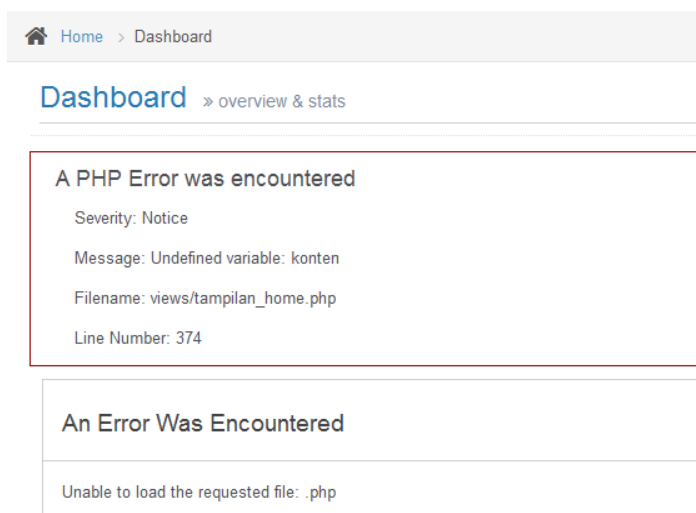
  <i class="ace-icon fa fa-check green"></i>

  Selamat Datang di Aplikasi
  <strong class="green">
    SIAKAD
    <small>(v1.3.3)</small>
  </strong>, Silahkan anda download di source di github
</div>
```

Tambahkan coding berikut pada **tampilan_home.php** untuk memanggil konten yang tadi kita cut.

```
<!-- PAGE CONTENT BEGINS -->
<?php $this->load->view($konten); ?>
<!-- PAGE CONTENT ENDS -->
```

- Berbeda saat kita memanggil **tampilan_menu.php** kali ini kita memanggil variable konten. Tujuannya adalah agar konten dapat menjadi dinamis sesuai dengan data yang dipilih.
- Coba jalankan apa yang terjadi ? Error ?



- Error ini terjadi karena saat kita memanggil **\$konten** variable tersebut belum tersedia pada controller oleh karenanya mari kita tambahkan variabelnya pada file **/application/controllers/home.php** ganti seperti berikut

```
public function index() {
    $this->load->view('tampilan_home');
    $isi['konten'] = 'tampilan_konten';
    $isi['judul'] = 'Home';
    $isi['sub_judul'] = '';
    $this->load->view('tampilan_home', $isi);
}
```

- Kembali ke **tampilan_home.php** yang ada pada folder **/application/views/tampilan_home.php** ubah seperti berikut :

```
<div class="page-content">
  <div class="page-header">
    <h1>
      <?php echo $judul; ?>
      <small>
        <i class="ace-icon fa fa-angle-double-
right"></i>
        <?php echo $sub_judul; ?>
      </small>
    </h1>
  </div><!-- /.page-header -->

  <div class="row">
    <div class="col-xs-12">
      <!-- PAGE CONTENT BEGINS -->
      <?php $this->load->view($konten); ?>
      <!-- PAGE CONTENT ENDS -->
    </div><!-- /.col -->
  </div><!-- /.row -->
</div><!-- /.page-content -->
```

- Refresh browser jalankan.

Agar tidak mengganggu hapus bagian **<!-- /.nav-search -->** dan juga hapus bagian **<!-- /.ace-settings-container -->**

Saat ini aplikasi masih belum secure, mengapa belum secure ??? karena saat kita akses controller langsung misalnya **localhost/siakadtutorial/index.php/home** maka kita akan diarahkan ke halaman home tanpa melalui login, ini sangat berbahaya jika kita menghosting aplikasi kita. Oleh karenanya mari kita buat securitynya terlebih dahulu :

- Buat file model baru dengan nama **model_security.php** simpan pada **/application/models/ model_security.php**

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

class Model_security extends CI_model
{
    public function getSecurity()
    {
        $username = $this->session->userdata('username');
        if(empty($username))
        {
            $this->session->sess_destroy();
            redirect('login');
        }
    }
}
```

Jangan lupa menambahkan autoload model_security pada file autoload, agar kita bisa memanggilnya tanpa menuliskannya pada controller. Caranya buka file **/application/config/autoload.php** dan tambahkan seperti berikut :

```
$autoload['model'] = array('model_security');
```

Sekarang kita sisipkan `model_security` dan ***logout()*** pada controller home buka file */application/controllers/home.php*

```
class Home extends CI_Controller {

    public function index() {
        $this->model_security->getSecurity();
        $isi['konten'] = 'tampilan_konten';
        $isi['judul'] = 'Home';
        $isi['sub_judul'] = '';
        $this->load->view('tampilan_home', $isi);
    }

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

}
```

4.4 Membuat Jurusan CRUD

Setelah kita tadi membuat home, berikutnya kita akan menampilkan data jurusan beserta operasinya.

4.4.1 Membuat Tampil Data

- Tambahkan file baru pada *application/controllers/jurusan.php* dengan nama ***jurusan.php*** seperti berikut :

```
class Jurusan extends CI_Controller {

    public function index() {
```



```

        $this->model_security->getSecurity();
        $isi['konten']      = 'jurusan/tampilan_jurusan';
        $isi['judul']      = 'Master';
        $isi['sub_judul']   = 'Jurusan';
        $isi['data']       = $this->db->get('prodi');
        $this->load->view('tampilan_home', $isi);
    }
}

```

- Berikutnya tambahkan file baru pada folder **views** dengan nama

/application/views/jurusan/tampilan_jurusan.php

```

<table id="sample-table-2" class="table table-striped table-bordered table-hover">
<thead>
    <tr>
        <th>No</th>
        <th>Kode</th>
        <th>Jurusan</th>
        <th>Singkatan</th>
        <th>Ketua Jurusan</th>
        <th>NIK</th>
        <th>Akreditasi</th>
        <th>Aksi</th>
    </tr>
</thead>
<tbody>
    <tr>
        <?php
            $no = 1;
            foreach ($data->result() as $row) {
                ?>
                <td><?php echo $no++; ?></td>
                <td><?php echo $row->kd_prodi ?></td>
                <td><?php echo $row->prodi ?></td>
                <td><?php echo $row->singkat ?></td>
                <td><?php echo $row->ketua_prodi ?></td>
                <td><?php echo $row->nik ?></td>
                <td><?php echo $row->akreditasi ?></td>
                <td>
                    <a class="btn btn-mini btn-info" href="<?php echo base_url();
?>index.php/jurusan/edit/<?php echo $row->kd_prodi; ?>"><i class="icon-edit bigger-
120"></i></a> ||



```

```

                <a class="btn btn-mini btn-danger" href="<?php echo base_url();
?>index.php/jurusan/hapus/<?php echo $row->kd_prodi; ?>" onclick="return confirm('anda
yakin mau menghapus'); "><i class="icon-trash bigger-120"></i></a>
            </td>
        </tr>
    </tbody>
</table>

```

- Jalankan pada browsermu, apakah data pada table prodi sudah muncul ?

No	Kode	Jurusan	Singkatan	Ketua Jurusan	NIK	Akreditasi	Aksi
1	023	teknik metalurgi	TM	aganaga	97988768	A	 
2	12	ngt	ngt	ngt	ngt	ng	 
3	34	adasdasd	asdasdas	asdasdasd	asdasd	as	 
4	55-201	Teknik Informatika	TI	Deddy Rusdiansyah	4846	A	 
5	57-201	Sistem Informasi	SI	Danish Putra Pramudia	6856	B	 
6	BN	BN	BN	BN	BN	BN	 

- Nah sekarang kita akan menambahkan datatable untuk keperluan pencarian dan paging, buka **ace template** pilih **tables.html** copykan javascript pada line **2864 – 2872** dan pastekan ke file **/application/views/jurusan/tampil_jurusan.php** sebelum script **<table>**

```

tampilan_home.php x tampil_jurusan.php x
1 <script type="text/javascript">
2     $(function() {
3
4         var oTable1 = $('#sample-table-2').dataTable( {
5             "aoColumns": [
6                 null, null, null, null, null, null, null,
7                 { "bSortable": false }
8             ] } );
9
10    })
11 </script>

```

- Pada **application/views/tampilan_home.php** tambahkan script berikut sebelum tag **</head>** :

```

<script type="text/javascript">
    window.jQuery || document.write("<script src='<?php echo
    base_url();?>assets/js/jquery-2.0.3.min.js'>"+ "</script>");
</script>

<script type="text/javascript">
    if("ontouchend" in document) document.write("<script src='<?php echo
    base_url();?>assets/js/jquery.mobile.custom.min.js'>"+ "</script>");
</script>

<script src="<?php echo base_url();?>assets/js/bootstrap.min.js"></script>

<script src="<?php echo base_url();?>assets/js/jquery.dataTables.min.js"></script>
<script src="<?php echo base_url();?>assets/js/jquery.dataTables.bootstrap.js"></script>
<script src="<?php echo base_url();?>assets/js/ace-elements.min.js"></script>
<script src="<?php echo base_url();?>assets/js/ace.min.js"></script>

```

- Lalu **hapus** javascript yang ada pada bagian bawah ***application/views/tampilan_home.php***. Yang dimulai dari **<!--inline scripts related to this page-->**
- Maka bila di jalankan akan muncul pagging dan pencariannya...

Tambah Data

Display 10 records Search:

No	Kode	Jurusan	Singkatan	Ketua Jurusan	NIK	Akreditasi	Aksi
1	023	teknik metalurgi	TM	aganaga	97988768	A	
2	12	ngt	ngt	ngt	ngt	ng	
3	34	adasdasd	asdasdas	asdasdasd	asdasd	as	
4	55-201	Teknik Infomatika	TI	Deddy Rusdiansyah	4846	A	
5	57-201	Sistem Informasi	SI	Danish Putra Pramudia	6856	B	
6	BN	BN	BN	BN	BN	BN	

Showing 1 to 6 of 6 entries

- Bila anda lihat, kita lupa menambahkan tombol untuk menambah data, mari kita tambahkan tombol tambah datanya pada ***/application/views/jurusan/tampil_jurusan.php*** :

```

13 <p>
14 <a href="<?php echo base_url();?>index.php/jurusan/tambah" class="btn btn-primary btn-small">tambah data</a>
15 </p>
16 <table id="sample-table-2" class="table table-striped table-bordered table-hover">

```

4.4.2 Membuat Tambah Data

- Tambahkan script berikut pada file */application/controllers/jurusan.php* dibawah function *index()*

```
public function tambah()
{
    $this->model_security->getsecurity();
    $isi['konten'] = 'jurusan/form_tambahjurusan';
    $isi['judul'] = 'Master';
    $isi['sub_judul'] = 'Tambah Jurusan';
    $isi['kode'] = "";
    $isi['jurusan'] = "";
    $isi['singkat'] = "";
    $isi['ketua'] = "";
    $isi['nik'] = "";
    $isi['akreditasi'] = "";
    $this->load->view('tampilan_home',$isi);
}
```

- Setelah itu tambahkan file baru dengan nama *form_tambahjurusan.php* pada folder */application/jurusan/form_tambahjurusan.php*

```
<form class="form-horizontal" method="POST" action="<?php echo
base_url();?>index.php/jurusan/simpan" onsubmit="return cekform();">

    <div class="control-group">
        <label class="control-label">Kode Jurusan</label>
        <div class="controls">
            <input type="text" name="kode" id="kode" placeholder="kode"
class="span3" required=true value="<?php echo $kode; ?>">
        </div>
    </div>

    <div class="control-group">
        <label class="control-label">Jurusan</label>
        <div class="controls">
            <input type="text" name="jurusan" id="jurusan"
placeholder="jurusan" class="span3" required=true value="<?php echo $jurusan; ?>">
        </div>
    </div>

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

```

        <label class="control-label">Singkatan</label>
        <div class="controls">
            <input type="text" name="singkat" id="singkat"
placeholder="singkatan" class="span3" required=true value="<?php echo $singkat; ?>">
        </div>
    </div>

    <div class="control-group">
        <label class="control-label">Ketua Jurusan</label>
        <div class="controls">
            <input type="text" name="ketua" id="ketua" placeholder="ketua
jurusan" class="span3" required=true value="<?php echo $ketua; ?>">
        </div>
    </div>

    <div class="control-group">
        <label class="control-label">NIK</label>
        <div class="controls">
            <input type="text" name="nik" id="nik" placeholder="nik"
class="span3" required=true value="<?php echo $nik; ?>">
        </div>
    </div>

    <div class="control-group">
        <label class="control-label">Akreditasi</label>
        <div class="controls">
            <input type="text" name="akreditasi" id="akreditasi"
placeholder="akreditasi" required=true class="span3" value="<?php echo $akreditasi; ?>">
        </div>
    </div>

    <div class="control-group">
        <hr>
        <label for=""></label>
        <div class="controls">
            <button type="submit" class="btn btn-primary btn-small">Simpan</button>
            <a href="<?php echo base_url();?>index.php/jurusan" class="btn btn-
primary btn-small">Tutup</a>
        </div>
    </div>
</form>

```

- Untuk validasi inputannya kita buat menggunakan html5 yaitu **required=true** seperti pada form Login.
- Berikutnya buat **file baru** dengan nama **model_jurusan.php** dengan script seperti berikut. simpan pada **/application/models/model_jurusan.php**

```
class Model_jurusan extends CI_model
{
    public function getData($key)
    {
        $this->db->where('kd_prodi', $key);
        $hasil = $this->db->get('prodi');
        return $hasil;
    }

    public function getInsert($data)
    {
        $this->db->insert('prodi', $data);
    }
}
```

pada controller **jurusan.php** tambahkan **function simpan()** untuk menghandle post dari user.

```
public function simpan()
{
    $this->model_security->getSecurity();

    $key = $this->input->post('kode');
    $data['kd_prodi'] = $this->input->post('kode');
    $data['prodi'] = $this->input->post('jurusan');
    $data['singkat'] = $this->input->post('singkat');
    $data['ketua_prodi'] = $this->input->post('ketua');
    $data['nik'] = $this->input->post('nik');
    $data['akreditasi'] = $this->input->post('akreditasi');

    $this->load->model('model_jurusan');
    $query = $this->model_jurusan->getData($key);
    if($query->num_rows()>0)
    {
        $this->model_jurusan->getUpdate($key,$data);
        $this->session->set_flashdata('info','data sukses diupdate');
    } else
```

```

    {
        $this->model_jurusan->getInsert($data);
        $this->session->set_flashdata('info','data sukses simpan');
    }
}

```

Tambahkan link pada menu jurusan, buka **menu_navigasi.php** pada link jurusan ubah menjadi seperti berikut :

```

<li>
    <a href="<?php echo base_url(); ?>index.php/jurusan">
        <i class="icon-list"></i>
        <span class="menu-text"> Jurusan </span>
    </a>
</li>

```

Sekarang coba anda tambahkan datanya.

4.4.3 Mengedit dan Menghapus Data Jurusan

Untuk mengedit dan mendelete kita perlu membuat function **getUpdate()** dan **getDelete()**

Tambahkan script pada **model_jurusan.php** seperti berikut

```

public function getUpdate($key,$data)
{
    $this->db->where('kd_prodi',$key);
    $this->db->update('prodi',$data);
}

public function getDelete($key)
{
    $this->db->where('kd_prodi',$key);
    $this->db->delete('prodi');
}

```

Tambahkan pula **function edit()** dan **function hapus()** pada file **/application/controllers/jurusan.php** untuk mengedit data jurusan ,

```

public function edit()
{
    $this->model_security->getSecurity();
    $isi['konten'] = 'jurusan/edit_jurusan';
    $isi['judul']   = 'master';
    $isi['sub_judul']   = 'edit jurusan';

    $key = $this->uri->segment(3);
    $this->db->where('kd_prodi',$key);
    $query = $this->db->get('prodi');
    if($query->num_rows()>0)
    {
        foreach ($query->result() as $row)
        {
            $isi['kode']           = $row->kd_prodi;
            $isi['jurusan']        = $row->prodi;
            $isi['singkat']        = $row->singkat;
            $isi['ketua']          = $row->ketua_prodi;
            $isi['nik']            = $row->nik;
            $isi['akreditasi']      = $row->akreditasi;
        }
    }
    else
    {
        $isi['kode']           = "";
        $isi['jurusan']        = "";
        $isi['singkat']        = "";
        $isi['ketua']          = "";
        $isi['nik']            = "";
        $isi['akreditasi']      = "";
    }

    $this->load->view('tampilan_home',$isi);
}

```

```

public function hapus()
{
    $this->model_security->getSecurity();
    $this->load->model('model_jurusan');

    $key = $this->uri->segment(3);
    $this->db->where('kd_prodi', $key);
    $query = $this->db->get('prodi');

```



```

        if($query->num_rows()>0)
        {
            $this->model_jurusan->getDelete($key);
        }
        redirect('jurusan','refresh');
    }
}

```

Berikutnya kita tambahkan file baru dengan nama **edit_jurusan.php** dan simpan pada **/application/views/jurusan/edit_jurusan.php**, untuk isi dari **edit_jurusan.php** hampir sama hanya berbeda pada bagian **kode jurusan**, selebihnya semuanya sama.

```

<div class="control-group">
    <label class="control-label">Kode Jurusan</label>
    <div class="controls">
        <input readonly type="text" name="kode" id="kode"
placeholder="kode" class="span3" required=true value="<?php echo $kode; ?>">
    </div>
</div>

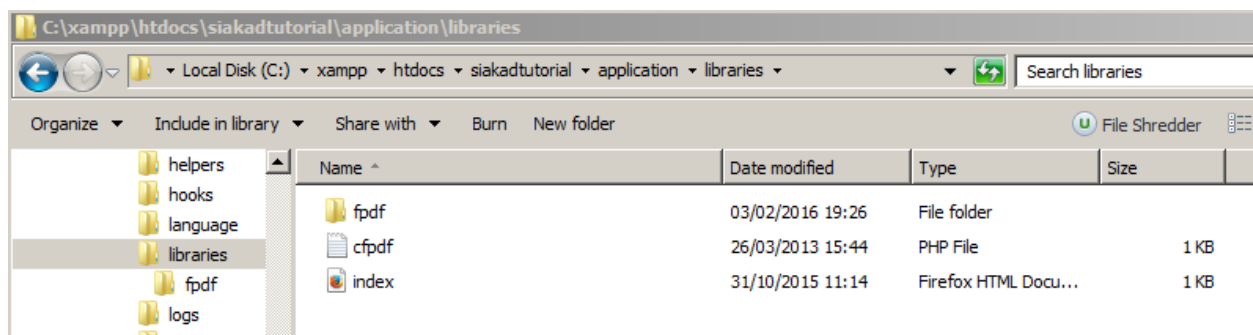
```

Perhatikan coding yang digaris bawahi. Untuk validasi lainnya anda bisa tambahkan sendiri.



4.4.4 Membuat Laporan Dalam Bentuk PDF

Untuk membuat laporan dalam bentuk PDF maka anda terlebih dahulu harus mempunyai library FPDF anda bisa download disini https://github.com/Muharir17/Library_FPDF bisa anda clone atau anda download dalam bentuk .zip. Jika sudah extract Library FPDF ke folder **/application/Libraries/**



Tambahkan function berikut pada **/application/controllers/jurusan.php**

```
public function pdf()
{
    $this->load->library('fpdf');
    $pdf = new FPDF('L', 'mm', 'A4');
    $pdf->AddPage();
    $pdf->SetFont('Arial', 'B', 'L');

    $pdf->Cell(10,7, 'No', 1, 0, 'C');
    $pdf->Cell(15,7, 'Kode', 1, 0, 'C');
    $pdf->Cell(40,7, 'Jurusan', 1, 0, 'C');
    $pdf->Cell(27,7, 'Singkatan', 1, 0, 'C');
    $pdf->Cell(50,7, 'Ketua', 1, 0, 'C');
    $pdf->Cell(50,7, 'Nik', 1, 0, 'C');
    $pdf->Cell(27,7, 'Akreditasi', 1, 1, 'C');

    // tampilkan dari database
    $pdf->SetFont('Arial', '', 'L');
    $query = $this->db->get('prodi');
    $no=1;
    foreach ($query->result() as $q)
    {
        $pdf->Cell(10,7,$no,1,0, 'C');
        $pdf->Cell(15,7,$q->kd_prodi,1,0);
        $pdf->Cell(40,7,$q->prodi,1,0);
        $pdf->Cell(27,7,$q->singkat,1,0);
        $pdf->Cell(50,7,$q->ketua_prodi,1,0);
        $pdf->Cell(50,7,$q->nik,1,0);
        $pdf->Cell(27,7,$q->akreditasi,1,1);

        $no++;
    }

    // end
    $pdf->Output();
}
```

Jika sudah untuk membuka laporan anda cukup membuka url **localhost/siakadtutorial/index.php/jurusan/pdf** maka akan menghasilkan file pdf seperti berikut :

dari 1 - + Perbesaran Otomatis

No	Kode	Jurusan	Singkatan	Ketua	Nik	Akreditasi
1	005	Ekonomi Syariah	FEKON	Mukhlis., SE	73453645365	B
2	34	adasdasd	asdasdas	asdasdasd	asdasd	as
3	55-201	Teknik Informatika	TI	Deddy Rusdiansyah	4846	A
4	57-201	Sistem Informasi	SI	Danish Putra Pramudia	6856	B
5	BN	BN	BN	BN	BN	BN
6	k03	Sistem Informasi	SI	Nina., M.Kom	19876535453636	c

Tugas anda berikutnya tinggal tambahkan menunya pada bilah samping.

4.4.5 Membuat Laporan Dalam Bentuk Excel

Berbeda dengan sebelumnya saat kita membuat laporan dalam bentuk excel kita tidak memerlukan library tambahan function berikut pada ***/application/controllers/jurusan.php***

```
public function excel()
{
    header("Content-type=application/vnd.ms-excel");
    header("content-disposition:attachment;filename=laporantransaksi.xls");

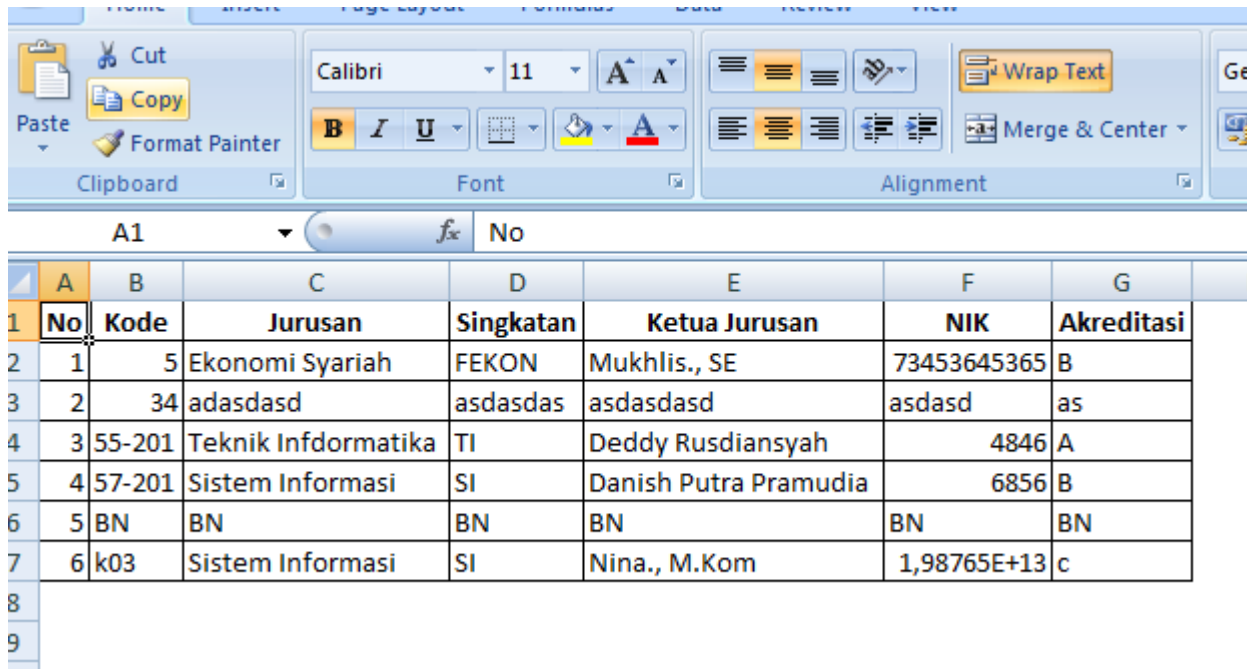
    $isi['konten'] = 'jurusan/laporan_excel';

    $this->model_security->getsecurity();
    $isi['konten'] = 'jurusan/lap_excel';
    $isi['judul'] = 'Master';
    $isi['sub_judul'] = 'Jurusan';
    $isi['data'] = $this->db->get('prodi');
    $this->load->view('jurusan/lap_excel',$isi);
}
```

Berikutnya buatlah file view baru dengan nama lap_excel.php dan simpan pada folder ***/application/views/jurusan/lap_excel.php***

```
1 <table id="sample-table-2" class="table table-striped table-bordered table-hover" border="1">
2     <thead>
3         <tr>
4             <th>No</th>
5             <th>Kode</th>
6             <th>Jurusan</th>
7             <th>Singkatan</th>
8             <th>Ketua Jurusan</th>
9             <th>NIK</th>
10            <th>Akreditasi</th>
11        </tr>
12    </thead>
13    <tbody>
14        <tr>
15            <?php
16            $no = 1;
17            foreach ($data->result() as $row) {
18                >>
19                <td><?php echo $no++; ?></td>
20                <td><?php echo $row->kd_prodi ;?></td>
21                <td><?php echo $row->prodi ;?></td>
22                <td><?php echo $row->singkat ;?></td>
23                <td><?php echo $row->ketua_prodi ;?></td>
24                <td><?php echo $row->nik ;?></td>
25                <td><?php echo $row->akreditasi ;?></td>
26            </tr>
27        <?php ?>
28    </tbody>
29 </table>
```

Jika sudah untuk membuka laporan anda cukup membuka url ***localhost/siakadtutorial/index.php/jurusan/excel*** maka akan menghasilkan file pdf seperti berikut :



No	Kode	Jurusan	Singkatan	Ketua Jurusan	NIK	Akreditasi
1	5	Ekonomi Syariah	FEKON	Mukhlis., SE	73453645365	B
2	34	adasdasd	asdasdas	asdasdasd	asdasd	as
3	55-201	Teknik Infidormatika	TI	Deddy Rusdiansyah	4846	A
4	57-201	Sistem Informasi	SI	Danish Putra Pramudia	6856	B
5	BN	BN	BN	BN	BN	BN
6	k03	Sistem Informasi	SI	Nina., M.Kom	1,98765E+13	c

Tugas anda berikutnya tinggal tambahkan menunya pada bilah samping.

4.4.6 Membuat Chart (Grafik)

Untuk membuat grafik ini anda harus membuat table dengan nama ***performance*** untuk strukturnya seperti berikut :

Nama Field	Tipe Data	Panjang	Keterangan
Performance_id	Int	10	Primary, Auto Increment
Tahun	Smallint	4	
Fti	Int	10	
Fekon	Double		
kesmas	double		

Tambahkan file controller baru dengan nama ***ChartController.php*** pada folder ***/application/controllers/ChartController.php***

```
jurusan.php  chartcontroller.php x
1  <?php
2
3  if (!defined('BASEPATH')) exit('No direct script access allowed');
4
5
6  class ChartController extends CI_Controller {
7
8      function __construct() {
9          parent::__construct();
10         $this->load->model('chartmodel', 'chart');
11     }
12
13     public function index() {
14         $this->model_security->getsecuriry();
15         $isi['konten'] = 'chart';
16         $isi['judul'] = 'Master';
17         $isi['sub_judul'] = 'Statistik';
18         $this->load->view('tampilan_home', $isi);
19
20         $results = $this->chart->get_chart_data();
21         $data['chart_data'] = $results['chart_data'];
22         $data['min_year'] = $results['min_year'];
23         $data['max_year'] = $results['max_year'];
24         $this->load->view('chart', $data);
25     }
26
27 }
```

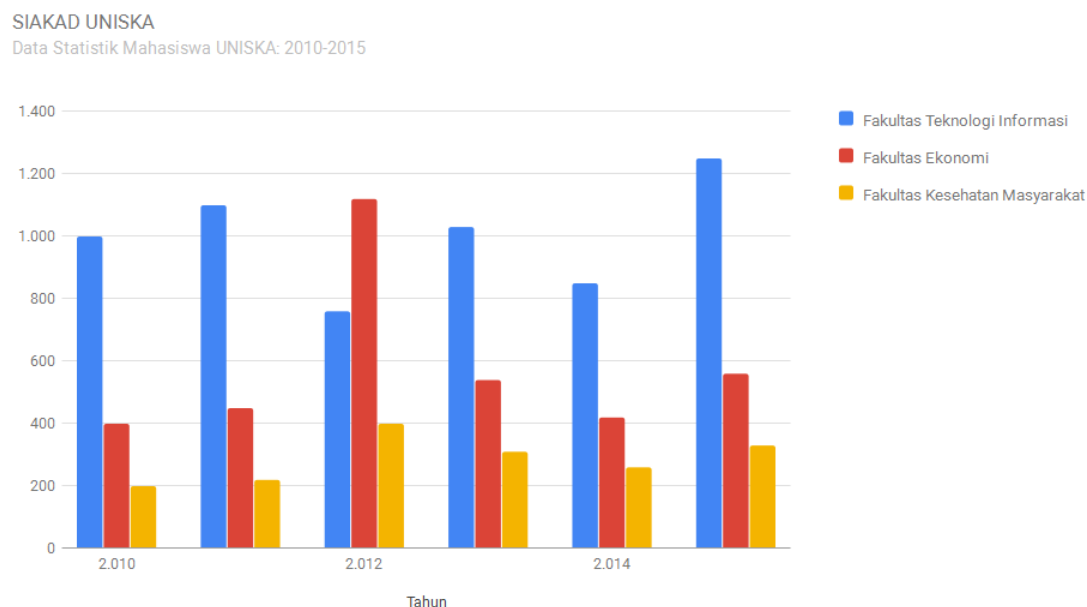
Berikutnya kita akan membuat viewnya simpan pada folder **application/views/chart.php** sebagai berikut , setelah anda selesai membuat viewnya untuk mencoba hasilnya anda dapat mengetik url seperti ini **localhost/siakadtutorial/index.php /ChartController** .

```

1 <script type="text/javascript" src="https://www.google.com/jsapi"></script>
2 <script type="text/javascript">
3   google.load("visualization", "1.1", {packages: ["bar"]});
4   google.setOnLoadCallback(drawChart);
5   function drawChart() {
6     var data = google.visualization.arrayToDataTable([
7       ['Tahun', 'Fakultas Teknologi Informasi',
8        'Fakultas Ekonomi', 'Fakultas Kesehatan Masyarakat'],
9
10    <?php
11      foreach ($chart_data as $data) {
12        echo '[' . $data->performance_year . ',' .
13        $data->fti . ',' . $data->fekon . ',' . $data->kesmas . '],';
14      }
15    <?>
16    ]]);
17
18    var options = {
19      chart: {
20        title: 'SIKAD UNISKA',
21        subtitle: 'Data Statistik Mahasiswa UNISKA:
22        <?php echo $min_year.'-'. $max_year;?>',
23      }
24    };
25
26    var chart = new google.charts.Bar(document.getElementById('columnchart_material'));
27    chart.draw(data, options);
28  }
29 </script>
30 </head>
31 <body>
32   <div id="columnchart_material" style="width: 900px; height: 500px;"></div>
33 </body>

```

Hasilnya seperti berikut



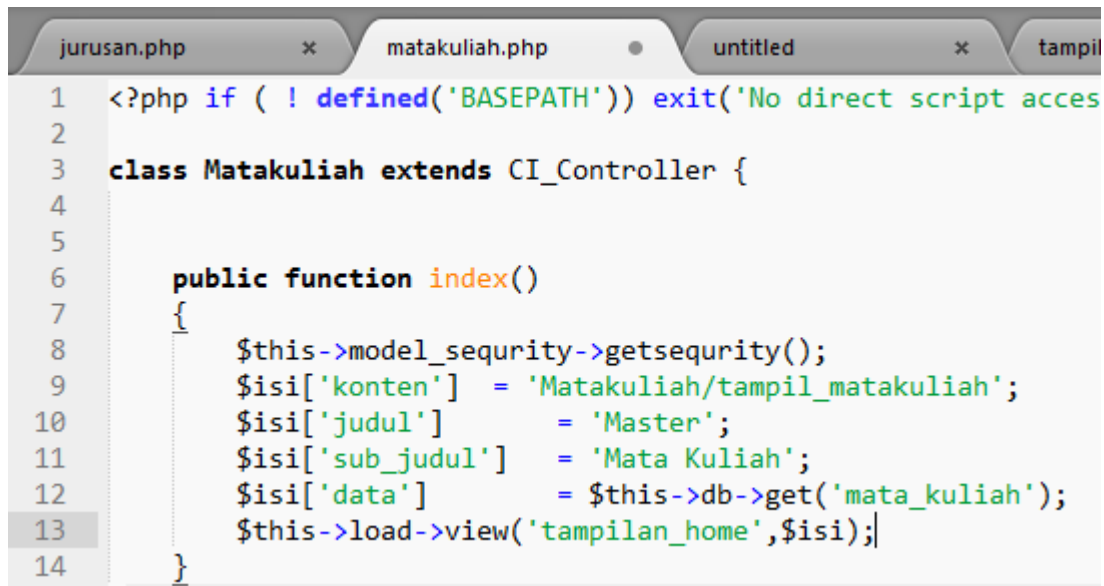
4.5 Membuat Matakuliah CRUD

Setelah kita tadi membuat Matakuliah CRUD, berikutnya kita akan menampilkan data jurusan beserta operasinya.

4.5.1 Membuat Tampil Data

Tambahkan file controller baru dengan nama **matakuliah.php** simpan pada folder **controllers**.

Berikut isi dari scriptnya :

The image shows a code editor with four tabs: 'jurusan.php', 'matakuliah.php', 'untitled', and 'tampil'. The 'matakuliah.php' tab is active, displaying PHP code for a controller. The code starts with a security check using 'defined('BASEPATH')' and 'exit()'. It then defines a 'Matakuliah' class extending 'CI_Controller'. Inside the class, there is an 'index()' method that sets view data for 'konten', 'judul', 'sub_judul', and 'data', and then loads the 'tampilan_home' view.

```
1 <?php if ( ! defined('BASEPATH')) exit('No direct script acces
2
3 class Matakuliah extends CI_Controller {
4
5
6     public function index()
7     {
8         $this->model_security->getsecurity();
9         $isi['konten'] = 'Matakuliah/tampil_matakuliah';
10        $isi['judul']   = 'Master';
11        $isi['sub_judul'] = 'Mata Kuliah';
12        $isi['data']     = $this->db->get('mata_kuliah');
13        $this->load->view('tampilan_home',$isi);
14    }
```

Lalu buat folder dengan nama **matakuliah** pada folder **views** tambahkan file dengan nama **application/views/matakuliah/tampil_matakuliah.php**

```

jurusan.php x matakuliah.php • untitled x tampil_matakuliah.php x
1 <script type="text/javascript">
2     $(function() {
3
4         var oTable1 = $('#sample-table-2').dataTable( {
5             "aoColumns": [
6                 null, null, null, null, null, null,
7                 { "bSortable": false }
8             ] } );
9
10    })
11 </script>
12 <p>
13 <a href="<?php echo base_url();?>index.php/matakuliah/tambah" class="btn btn-primary btn-small">tambah
14 data</a>
15 </p>
16 <table id="sample-table-2" class="table table-striped table-bordered table-hover">
17     <thead>
18         <tr>
19             <th>no</th>
20             <th>nama jurusan</th>
21             <th>kode matakuliah</th>
22             <th>nama matkul</th>
23             <th>sks</th>
24             <th>semester</th>
25             <th>aktif</th>
26             <th>aksi</th>
27     </thead>

```

```

jurusan.php x matakuliah.php • untitled x tampil_matakuliah.php x
28 <tbody>
29     <tr>
30         <?php
31         $no = 1;
32         foreach ($data->result() as $row) {
33             $jurusan = $this->Model_matakuliah->tampilmatkul($row->kd_prodi);
34             <?>
35             <td><?php echo $no++; |?></td>
36             <td><?php echo $jurusan ; ?></td>
37             <td><?php echo $row->kd_mk ; ?></td>
38             <td><?php echo $row->nama_mk ; ?></td>
39             <td><?php echo $row->sks ; ?></td>
40             <td><?php echo $row->smt; ?></td>
41             <td><?php echo $row->aktif; ?></td>
42
43             <td>
44                 <a href="">edit</a>
45                 <a href="">delete</a>
46             </td>
47         </tr>
48         <?php }?>
49     </tbody>
50 </table>
51

```

Berikutnya tambahkan modelnya dengan nama **model_matakuliah.php** dan simpan pada **/application/models/model_matakuliah.php** berikut scriptnya :


```

jurusan.php x matakuliah.php x tampil_matakuliah.php x model_matakuliah.php
1 <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3 class Model_matakuliah extends CI_Model{
4
5     public function tampilmatkul($key)
6     {
7         $this->db->where('kd_prodi',$key);
8         $query = $this->db->get('prodi');
9         if($query->num_rows()>0)
10        {
11            foreach ($query->result() as $row)
12            {
13                $hasil = $row->prodi;
14            }
15        }
16        else
17        {
18            $hasil = '';
19        }
20        return $hasil;
21    }
22
23    public function getlistjurusan()
24    {
25        return $this->db->get('prodi');
26    }
27 }

```

Jalankan, Bila muncul error seperti berikut ini dikarenakan **model_matakuliah** belum terload otomatis.

tambah data

A PHP Error was encountered

Severity: Notice

Message: Undefined property: CI_Loader::\$model_matakuliah

Filename: matakuliah/tampil_matakuliah.php

Line Number: 33

Fatal error: Call to a member function tampilmatkul() on a non-object in C:\xampp\htdocs\siakadtutorial\application\views\matakuliah\tampil_matakuliah.php on line 33

no	nama jurusan	kode matakuliah	nama matkul	sks	semester	aktif	aksi

Untuk mengatasinya buka file **autoload.php** pada folder config dan tambahkan modelnya seperti berikut :

```

111
112 $autoload['model'] = array('model_security','model_matakuliah');
113

```

Dan pada **/application/controllers/matakuliah.php** load modelnya dengan menambahkan script berikut :

```
jurusan.php x matakuliah.php • tampil_matakuliah.php x model_mata
1 <?php if ( ! defined('BASEPATH')) exit('No direct script access all
2
3 class Matakuliah extends CI_Controller {
4
5     public function index()
6     {
7         $this->load->model('Model_matakuliah');
8         $this->model_security->getsecurity();
9         $isi['konten'] = 'Matakuliah/tampil_matakuliah';
10        $isi['judul']   = 'Master';
11        $isi['sub_judul'] = 'Mata Kuliah';
12        $isi['data']     = $this->db->get('mata_kuliah');
13        $this->load->view('tampilan_home',$isi);
14    }
15 }
```

4.5.2 Membuat Tambah Data

Bila pada CRUD jurusan kita parsing data menggunakan php, kali ini kita akan menggunakan JQuery Ajax untuk tambah data .

Buka **/application/controllers/matakuliah.php** tambahkan **function tambah()** untuk tambah datanya

```
jurusan.php x matakuliah.php x tampil_matakuliah.php x mode
15
16     public function tambah()
17     {
18         $this->load->model('Model_matakuliah');
19         $this->model_security->getsecurity();
20         $isi['konten'] = 'Matakuliah/form_tambahmatakuliah';
21         $isi['judul']   = 'master';
22         $isi['sub_judul'] = 'matakuliah';
23         $this->load->view('tampilan_home',$isi);
24     }
25 }
```

Berikutnya kita buat file baru dengan nama **form_tambahmatakuliah.php** simpan pada folder **/views/matakuliah/form_tambahmatakuliah.php**

```

1 <script type="text/javascript">
2   $(document).ready(function(){
3
4
5     function getDataMK()
6     {
7       var jurusan = $("#jurusan").val();
8       //di pancing apakah javascript jalan
9       //alert(jurusan);
10
11       $.ajax({
12         type : 'post',
13         url : '<?php echo site_url();?>/matakuliah/getDataMK',
14         data : 'jurusan='+jurusan, //jangan ada titik //variable yg di kirim
15         success : function(data){
16           $("#view_data").html(data); // data di tampung di div view data
17         }
18       });
19     }
20
21     $("#jurusan").change(function(){ //jika dipilih jurusan akan menampilkan data mata kuliah berdasarkan jurusan
22       //memanggil fungsi getDataMK
23       getDataMK();
24       //dilanjutkan dengan membuat class getDataMk di controller
25     });
26

```

```

27     $("#simpan").click(function(){
28       var string = $("#my-form").serialize();
29       //alert(string);
30       $.ajax({
31         type : 'POST',
32         url : '<?php echo site_url();?>/matakuliah/simpan',
33         data : string,
34         success : function(data){
35           alert(data);
36         }
37       });
38     });
39   });
40 </script>

```



```

jurusan.php x matakuliah.php x view_datamatakuliah.php x form_tambahmatakuliah.php x
1 <script type="text/javascript">
2 $(document).ready(function(){
3     //alert('Apakah JQuery berjalan dengan normal');
4 });
5 $(function() {
6     //javascript untuk memanggil plugin data table , samakan dengan nama ID table
7     var oTable1 = $('#sample-table-2').dataTable( {
8         "aoColumns": [ // script dibawah harus di samakan dengan jumlah kolom di table, kalo tidak sama
9             // dengan kolom table akan error sorting,paging dan searching
10             null,null,null, null,null, null,null,
11             { "bSortable": false } //script ini tidak membuat sortiran
12         ] } );
13     })
14 </script>
15 <div class="row-fluid">
16     <h3 class="header smaller lighter blue">Data Mata Kuliah</h3>
17     <table id="sample-table-2" class="table table-striped table-bordered table-hover">
18         <thead>
19             <tr>
20                 <th>No</th>
21                 <th>Jurusan</th>
22                 <th>Kode</th>
23                 <th>Nama Mata Kuliah</th>
24                 <th>SKS</th>
25                 <th>SMT</th>
26                 <th>Status</th>
27                 <th class="hidden-480">Aksi</th>
28             </tr>
29         </thead>
30
31         <tbody>
32             <?php
33             // $this->Load->model('m_matakuliah');
34             // untuk menampilkan data dari table, diambil dari variable table yg ada di controller jurusan
35             // data di parshing / lempar dari controller
36             $no=1;
37             foreach($data->result() as $row){
38                 // mengambil nama jurusan pada model matakuliah
39                 $jurusan = $this->model_matakuliah->tampilmatkul($row->kd_prodi);
40                 <?>
41                 <tr>
42                     <td><?php echo $no++;?></td>
43                     <td><?php echo $jurusan;?></td>
44                     <td><?php echo $row->kd_mk;?></td>
45                     <td><?php echo $row->nama_mk;?></td>
46                     <td><?php echo $row->sks;?></td>
47                     <td><?php echo $row->smt;?></td>
48                     <td><?php echo $row->aktif;?></td>
49                     <td class="td-actions">
50                         <div class="hidden-phone visible-desktop action-buttons">
51                             <!-- edit data berdasarkan kd_prodi -->
52                             <!-- proses edit menggunakan javascript, salah ambil file -->
53                             <a class="green" href="javascript:void(0)" onclick="editData('<?php echo $row->kd_mk;?>')">
54                                 <i class="icon-pencil bigger-130"></i>
55                             </a>
56                             <!-- buat juga fungsi hapusData pada javascript -->
57                             <a class="red" href="javascript:void(0);" onclick="hapusData('<?php echo $row->kd_mk;?>')">
58                                 <!-- sedikit penambahan javascript untuk konfirmasi -->
59                                 <i class="icon-trash bigger-130"></i>
60                             </a>
61                         </div>

```

```

61 <!-- script yg di bawah ini hanya untuk -->
62 <div class="hidden-desktop visible-phone">
63 <div class="inline position-relative">
64 <button class="btn btn-minier btn-yellow dropdown-toggle" data-toggle="dropdown">
65 <i class="icon-caret-down icon-only bigger-120"></i>
66 </button>
67
68 <ul class="dropdown-menu dropdown-icon-only dropdown-yellow pull-right dropdown-caret
69 dropdown-close">
70
71 <li>
72 <a href="#" class="tooltip-success" data-rel="tooltip" title="Edit">
73 <span class="green">
74 <i class="icon-edit bigger-120"></i>
75 </span>
76 </a>
77 </li>
78
79 <li>
80 <a href="#" class="tooltip-error" data-rel="tooltip" title="Delete">
81 <span class="red">
82 <i class="icon-trash bigger-120"></i>
83 </span>
84 </a>
85 </li>
86 </ul>
87 </div>
88 </div>
89 </td>
90 </tr>
91 <?php } ?>
92 </tbody>
93 </table>
94 </div>

```

Jangan lupa untuk mengubah link pada **tampilan_menu.php** menjadi seperti berikut :

```

<li>
  <a href="<?php echo base_url();?>index.php/matakuliah">
    <i class="icon-double-angle-right"></i>
    Data Matakuliah
  </a>
</li>

```

4.5.3 Mengedit dan Menghapus Data Matakuliah

Untuk edit dan hapusnya silahkan anda buat sendiri sebagai bahan pembelajaran.

BAB 5

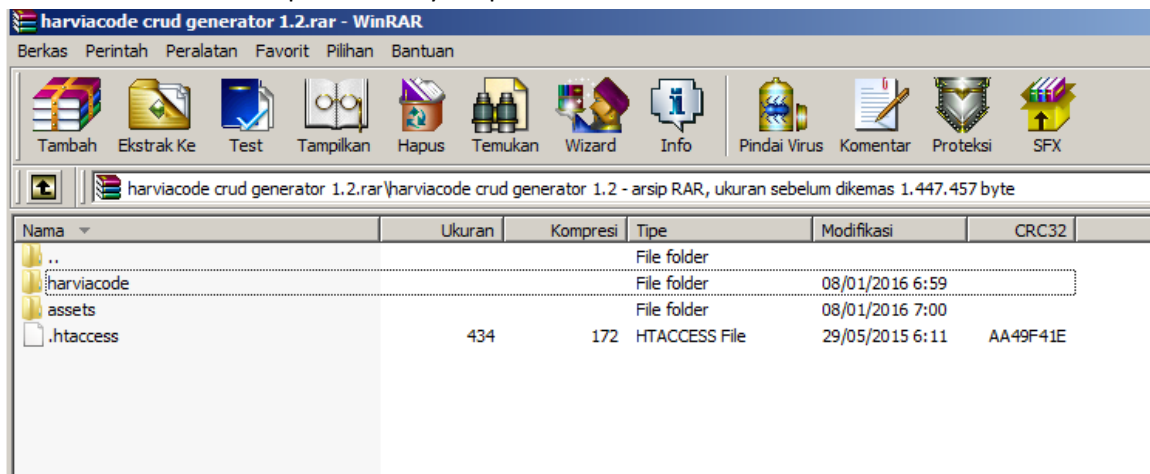
Integrasi CRUD Generator Harviacode

5.1 Apa itu Harviacode CRUD generator ?

CRUD generator adalah library dari pihak ke 3 untuk membuat CRUD lebih mudah hanya dengan menginputkan nama table dan kita sudah bisa insert, update, delete hanya dengan 1 kali klik, dengan begini proses membuat crud yang memakan waktu bisa menjadi lebih singkat. Bahkan dengan menggunakan Harviacode kita sudah dibuatkan laporan berupa export ke excel dan word document selain itu fitur seperti pagging dan data tables sudah tersedia.

5.2 Bagaimana cara menggunakannya ?

- Pertama download terlebih dahulu library nya pada alamat berikut
<http://harviacode.com/2015/05/08/codeigniter-crud-generator/>
- Setelah anda dapatkan filenya seperti berikut



- Extract ketiganya ke folder root project kita sehingga struktur folder project kita menjadi seperti berikut :

Local Disk (C:) > xampp > htdocs > siakadtutorial

Share with ▾ Burn New folder

Name ^	Date modified	Type	Size
application	31/10/2015 11:14	File folder	
assets	01/02/2016 11:05	File folder	
harviacode	08/01/2016 6:59	File folder	
system	31/10/2015 11:14	File folder	
.htaccess	31/10/2015 11:14	Text Document	1 KB
.travis.yml	29/05/2015 6:11	HTACCESS File	1 KB
index	31/10/2015 11:14	YML File	1 KB
license	31/10/2015 11:14	PHP File	7 KB
README.md	31/10/2015 11:14	Text Document	3 KB
		MD File	1 KB

Setelah itu kita konfigurasi terlebih dahulu

- Buka folder **harviacode** -> **lib** -> **config.php**

```

1  <?php
2  //configure your database setting here
3  $hostname = 'localhost';
4  $username = 'root';
5  $password = '';
6  $database = 'db_siakad';
7
8  //configure target folder here, default is application folder CI
9  $target = "../application/";
10 // $target = "output/";
11 ?>

```

- Lalu anda buka link berikut ini **localhost/siakadtutorial/harviacode** maka akan muncul halaman seperti berikut

☒ Codeigniter 2 ☐ Codeigniter 3

☒ Regular Table ☐ Datatables

☐ Export Excel

☐ Export Word

☐ Create ../application/config/pagination.php

Custom Controller Name

Custom Model Name

Codeigniter CRUD Generator 1.2 by harviacode.com

About :

Codeigniter CRUD Generator is a simple tool to auto generate model, controller and view from generator will make a complete CRUD operation, pagination, search, form*, form validation, export bootstrap 3 style. You still need to modify the result code for more customization.

* generate textarea and text input only

Please visit and like harviacode.com for more info and PHP tutorials.

How to use this CRUD Generator :

- Simply put 'harviacode' folder, 'asset' folder and .htaccess file into your codeigniter root
- Change database configuration in harviacode/lib/config.php.
- Open <http://localhost/codeigniter/harviacode>.
- Write your table name, choose codeigniter version and push generate button.
- That steps will generate this following files :
 - ../application/models/tablename_model.php
 - ../application/controllers/tablename.php
 - ../application/views/tablename_list.php
 - ../application/views/tablename_form.php
 - ../application/views/tablename_read.php
 - ../application/config/pagination.php

Important :

- On application/config/autoload.php, load **database library**, **session library** and **url helper**
- On application/config/config.php, set **\$config['index_page'] = ''**, **\$config['url_suffix'] = ''**
- On application/config/database.php, set **hostname**, **username**, **password** and **database**

Tinggal anda sesuaikan table apa yang akan anda buat CRUD nya . CodeIgniter berapa yang anda pakai dan apakah anda ingin menggunakan **table biasa** atau menggunakan **dataTable**, **export word** dan **excel**.

Contoh table dosen. Maka akan menjadi seperti berikut :

dosen

☒ Codeigniter 2

☐ Codeigniter 3

☐ Regular Table

☒ Datatables

☒ Export Excel

☒ Export Word

☒ Create ../application/config/pagination.php

Custom Controller Name

dosen

Custom Model Name

dosen_model

Generate

Pilih generate. Maka secara otomatis Harviacode sudah membuatkan Model View dan Controllernya.

Bila sudah selesai anda cukup mengetikkan alamat berikut ini :

Data Dosen = ***localhost/siakadtutorial/index.php/dosen***

Read Dosen = ***localhost/siakadtutorial/index.php/dosen/read***

Create Dosen = ***localhost/siakadtutorial/index.php/dosen/create***

Edit Dosen = ***localhost/siakadtutorial/index.php/dosen/edit***

Delete Dosen = ***localhost/siakadtutorial/index.php/dosen/delete***

Bila sudah tinggal anda sesuaikan dengan template yang anda gunakan.

Catatan

- Library FPDF dapat di download https://github.com/Muharir17/Library_FPDF
- Script CodeHallo dapat di download <https://github.com/Muharir17/CodeHallo>
- Script SiakadTutorial dapat di download <https://github.com/Muharir17/siakadtutorial>
- Untuk menampilkan grafik anda harus terkoneksi dengan internet karena kita menggunakan Google API .