



CodeIgniter



bootstrap

EBOOK

# TUTORIAL CODEIGNITER LENGKAP

STUDI KASUS SISTEM INFORMASI PERPUSTAKAAN

The image shows a screenshot of a web application titled "SISTEM INFORMASI PERPUSTAKAAN". It features a login form on the left with fields for Username, Password, and Login Sebagai (Admin), and a "Login" button. To the right is the main dashboard, which includes a "Selamat datang!" message, a brief description of the system, and four summary statistics: 5 Jumlah Buku, 2 Jumlah Anggota, 2 Jumlah Total Peminjaman, and 1 Jumlah Petugas. Below the dashboard is a cartoon illustration of a man with a mustache sitting at a desk with a typewriter, surrounded by books and a quill pen.

Diki Alfarabi Hadi, S.T.

WWW.MALASNGODING.COM

" Menuntut Ilmu adalah Taqwa,  
Menyampaikan ilmu adalah Ibadah,  
Mengulang-ulang ilmu adalah Zikir,  
Mencari Ilmu adalah Jihad ".

- Imam Al Ghazali -

# Tentang Penulis



Diki Alfarabi Hadi, seorang praktisi IT yang mengkhususkan diri di bidang *web development* dan *mobile app*, memulai petualangan dalam dunia pemrograman sejak tahun 2014 atau semenjak masih kuliah. Lebih senang dengan dunia pemrograman dibanding bidang IT yang lain. Senang berbagi pengetahuan dan mempelajari sesuatu yang baru khususnya bidang *web development*.

Memiliki pengalaman sebagai *programmer* di CV. Carikode Indonesia, sebagai *Back-end web developer* di PT. Bin Hasan Indonesia dalam pengembangan sistem *crowdfunding*, dan beberapa perusahaan lain. Pernah menjuarai beberapa perlombaan *web design* tingkat nasional bersama tim. Salah satunya adalah *web design competition* yang diadakan oleh PT. Pelindo III di Surabaya.

Saat ini penulis aktif berbagi tutorial pemrograman di [www.malasngoding.com](http://www.malasngoding.com). Untuk mengetahui informasi lebih lanjut tentang penulis, silahkan mengunjungi <https://www.malasngoding.com>. Atau mengirimkan email ke [malasngoding@gmail.com](mailto:malasngoding@gmail.com).

# Pengantar

Dengan menyebut nama Allah SWT yang Maha Pengasih lagi Maha Panyayang, Penulis panjatkan puja dan puji syukur atas kehadirat-Nya, yang telah melimpahkan rahmat, hidayah, dan inayah-Nya kepada penulis, sehingga penulis dapat menyelesaikan buku “ **Tutorial Codeigniter Lengkap Dengan Studi Kasus Membuat Sistem Informasi Perpustakaan** ” ini.

Penulis menyadari sepenuhnya masih ada kekurangan baik dari segi susunan kalimat maupun tata bahasa. Oleh karena itu, dengan tangan terbuka penulis menerima segala saran dan kritik dari pembaca agar menjadi lebih baik untuk ke depannya.

Akhir kata penulis berharap buku ini dapat memberikan manfaat maupun inspirasi terhadap pembaca.

# Daftar Isi

Tentang Penulis .....	1
Pengantar .....	2
Daftar Isi .....	3
Tutorial Codeigniter dasar .....	7
Dasar Bagian 1 : Berkenalan Dengan Framework Codeigniter.....	8
Pengertian MVC.....	9
Kelebihan Framework Codelgniter .....	9
Dasar Bagian 2 : Cara Instalasi Framework Codelgniter .....	11
Dasar Bagian 3 : Mengenal Uri Segment Pada Codeigniter .....	14
Dasar Bagian 4 : Pengenalan Controller Codeigniter.....	18
Pengenalan Controller.....	18
Cara Membuat Controller Pada Codelgniter .....	20
Dasar Bagian 5 : Pengenalan View Pada Codelgniter .....	23
Pengenalan View .....	23
Cara Membuat View Pada Codelgniter .....	23
Cara Parsing Data ke View Codelgniter .....	25
Dasar Bagian 6 : Membuat Template Web Sederhana (Templating) Pada Codeigniter .....	29
Membuat Template Web (Templating) Dengan Codeigniter.....	29
Setting Base Url pada Codelgniter .....	32
Membuat Halaman Template Dinamis Dengan Codeigniter.....	33
Kesimpulan .....	37
Dasar Bagian 7 : Form Validation Pada Codelgniter.....	38
Membuat Form Validation Pada Codelgniter .....	38
Dasar Bagian 8 : Upload File Dengan Codeigniter.....	42
Membuat Upload File Dengan Codeigniter .....	42
Dasar Bagian 9 : Membuat Download File Dengan Codeigniter .....	49
Membuat Download File Dengan Codeigniter .....	49
Membuat Download File Kita Tentukan Sendiri .....	50
Membuat Download File Yang Sudah Tersedia .....	50

Contoh Membuat Download File Dengan Codeigniter.....	50
Dasar Bagian 10 : Membuat Library Codeigniter Sendiri.....	53
Penjelasan.....	55
Dasar Bagian 11 : Menghilangkan Index.php Pada Codelgniter .....	56
Menghilangkan Index.php Pada Codelgniter.....	57
Dasar Bagian 12 : Mengenal Model, Koneksi Database Codeigniter dan MySQL .....	59
Membuat Koneksi Database Codeigniter dan MySQL .....	59
Pengenalan Model Pada Codelgniter.....	62
Dasar Bagian 13 : Membuat CRUD Dengan Codeigniter .....	66
Koneksi Database.....	66
Menampilkan Data Dari Database.....	68
Menginput Data Ke Database.....	71
Menghapus Data Dari Database.....	75
Mengupdate Data Pada Database .....	79
Dasar Bagian 14 : Membuat Pagination Dengan Codelgniter.....	85
Dasar Bagian 15 : Membuat Login Dengan Codeigniter.....	90
Persiapan Membuat Login Dengan Codeigniter.....	90
Menghubungkan Codeigniter dengan MySQL.....	92
Membuat Form Login Dengan Codeigniter.....	93
Tutorial Codeigniter Lanjutan     Studi Kasus Membuat Sistem Informasi Perpustakaan .....	100
Instalasi Codeigniter 3 .....	102
Persiapan Membuat Sistem Informasi Perpustakaan.....	104
Setting Base URL .....	104
Load Library, Helper, dan segala yang dibutuhkan.....	105
Menghilangkan Index.php dari URL codeigniter.....	106
Merancang & Membuat Database Perpustakaan, Serta Konfigurasi di Codeigniter .....	108
Alur sistem informasi perpustakaan.....	109
Mempersiapkan database dan table.....	109
Import database 'perpustakaan'.....	111

Konfigurasi & Koneksi Database CodeIgniter dengan Database .....	114
Membuat Model Yang Terstruktur dan Efisien Untuk CRUD .....	116
Integrasi Codeigniter dan Bootstrap .....	121
Integrasi CodeIgniter Dengan DataTables.....	123
Membuat Login (md5) Codeigniter Dengan Pesan Notifikasi .....	124
Persiapan Membuat Login Admin & Petugas (Multi User) .....	124
Membuat Login Admin dan Petugas (Multi User).....	125
Mempersiapkan Halaman Dashboard Admin .....	135
Membuat Logout Admin .....	143
Membuat Fitur Ganti Password Admin.....	145
Membuat CRUD Data Petugas Pada Admin.....	150
Menampilkan Data Petugas Dari Database.....	150
Input Data Petugas ke Database .....	152
Update Data Petugas di Database .....	156
Menghapus Data Petugas dari Database .....	160
Membuat Halaman Dashboard Petugas .....	163
Membuat Logout Petugas .....	168
Membuat Ganti Password Petugas .....	169
Membuat CRUD Anggota Pada Petugas .....	171
Menampilkan Data Anggota Dari Database .....	171
Input Data Anggota Ke Database .....	174
Update Data Anggota Di Database.....	177
Menghapus Data Anggota dari Database.....	181
Membuat Cetak Kartu Anggota .....	183
Membuat CRUD Data Buku Pada Petugas .....	186
Menampilkan data buku.....	186
Menginput Data Buku Ke Database.....	188
Membuat Update Data Buku Pada Database .....	191
Membuat Hapus Buku Dari Database .....	195
Membuat Proses Transaksi Peminjaman Buku Pada Petugas .....	196
Menampilkan Data Transaksi Peminjaman.....	196

Menginput Peminjaman Baru .....	200
Membuat Proses Batalkan Peminjaman .....	205
Membuat Proses Peminjaman Selesai .....	208
Membuat Filter Dan Cetak Laporan Peminjaman Buku.....	211
Melanjutkan Dashboard Admin.....	218
Menampilkan Data Anggota Pada Admin.....	218
Membuat Cetak Kartu Anggota Pada Admin .....	220
Menampilkan Data Buku Pada Halaman Admin .....	222
Menampilkan & Cetak Laporan Peminjaman Pada Admin.....	224
Membuat Ringkasan Pada Dashboard .....	230
Membuat Ringkasan Pada Dashboard Admin .....	230
Membuat Ringkasan Pada Dashboard Petugas.....	233
Penutup .....	235
Selesai.....	236

# **Tutorial Codeigniter dasar**

# **Dasar Bagian 1 : Berkenalan Dengan Framework Codeigniter**

Selamat datang di tutorial codeigniter dasar di ebook “Tutorial Lengkap Codeigniter Dengan Studi Kasus Membuat Sistem Informasi Perpustakaan”. Pada bagian pertama ini akan dijelaskan tentang pengertian framework codeigniter dan akan sedikit berkenalan dengan salah satu framework PHP yang populer ini.

Pada pembahasan ini sesuai dengan judul akan di jelaskan tentang pengertian dan cara menggunakan codeigniter. tentu anda pasti sudah sering mendengar tentang framework codeigniter. Baik itu dari teman, kampus, sekolah atau dari forum-forum pemrograman di internet. Tapi tidak sedikit juga dari teman-teman yang berkunjung ke [www.malasngoding.com](http://www.malasngoding.com) masih bertanya-tanya tentang apa sih framework codeigniter itu?

Oleh sebab itu, pada pembahasan bagian paling pertama ini kita akan sedikit berkenalan dengan codeigniter. Dan belajar bagaimana cara melakukan instalasi codeigniter. Kita akan mulai dengan sedikit pengertian dari *apa itu framework ?* dan *apa itu framework codeigniter ?*

Yuk di simak..

Framework adalah kumpulan intruksi-intruksi yang di kumpulkan dalam class dan function-function dengan fungsi masing-masing untuk memudahkan developer/programmer dalam pemanggilannya tanpa harus menuliskan syntax program yang sama secara berulang-ulang. hal ini memiliki kegunaan untuk menghemat waktu dan mencegah penulisan syntax secara berulang-ulang agar tercipta nya source code (kode program) yang bersih dan terstruktur.

Codeigniter adalah sebuah framework PHP yang bersifat open source dan menggunakan metode MVC (Model, View, Controller). codeigniter bersifat free alias tidak berbayar jika anda menggunakannya. framework codeigniter di buat dengan tujuan sama seperti framework lainnya yaitu untuk memudahkan developer atau programmer dalam membangun sebuah aplikasi berbasis web tanpa harus membuat nya dari awal.

Framework codeigniter juga termasuk salah satu framework PHP yang paling banyak di gunakan oleh web developer untuk membangun sebuah aplikasi berbasis website.

Pengembangan codeigniter oleh developer-nya juga sangat baik. Dan memiliki dokumentasi yang baik dan jelas sebagai panduan kita menggunakan framework codeigniter.

## **Pengertian MVC**

Pada pengertian codeigniter di atas tadi di jelaskan bahwa codeigniter menggunakan metode MVC. Apa itu MVC ? kita juga harus mengetahui apa itu MVC sebelum masuk dan lebih dalam mengoprek codeigniter.

MVC adalah teknik atau konsep yang memisahkan komponen utama menjadi tiga komponen yaitu model, view dan controller.

### **Model**

Model merupakan bagian penanganan yang berhubungan dengan pengolahan atau manipulasi database. seperti misalnya mengambil data dari database, menginput dan pengolahan database lainnya. semua intruksi atau fungsi yang berhubung dengan pengolahan database di letakkan di dalam model.

### **View**

View merupakan bagian yang menangani halaman user interface atau halaman yang muncul pada user(pada browser). tampilan dari user interface di kumpulkan pada view untuk memisahkannya dengan controller dan model sehingga memudahkan web designer dalam melakukan pengembangan tampilan halaman website.

### **Controller**

Controller merupakan kumpulan intruksi aksi yang menghubungkan model dan view, jadi user tidak akan berhubungan dengan model secara langsung, intinya data yang tersimpan di database (model) di ambil oleh controller dan kemudian controller pula yang menampilkan nya ke view. Jadi controller lah yang mengolah intruksi.

Dari penjelasan tentang model view dan controller di atas dapat di simpulkan bahwa controller sebagai penghubung view dan model. misalnya pada aplikasi yang menampilkan data dengan menggunakan metode konsep mvc, controller memanggil intruksi pada model yang mengambil data pada database, kemudian controller yang meneruskannya pada view untuk di tampilkan. jadi jelas sudah dan sangat mudah dalam pengembangan aplikasi dengan cara mvc ini karena web designer atau front-end developer tidak perlu lagi berhubungan dengan controller, dia hanya perlu berhubungan dengan view untuk mendesign tampilan aplikasi, karena back-end developer yang menangani bagian controller dan modelnya. jadi pembagian tugas pun menjadi mudah dan pengembangan aplikasi dapat di lakukan dengan cepat dan terstruktur.

## **Kelebihan Framework CodeIgniter**

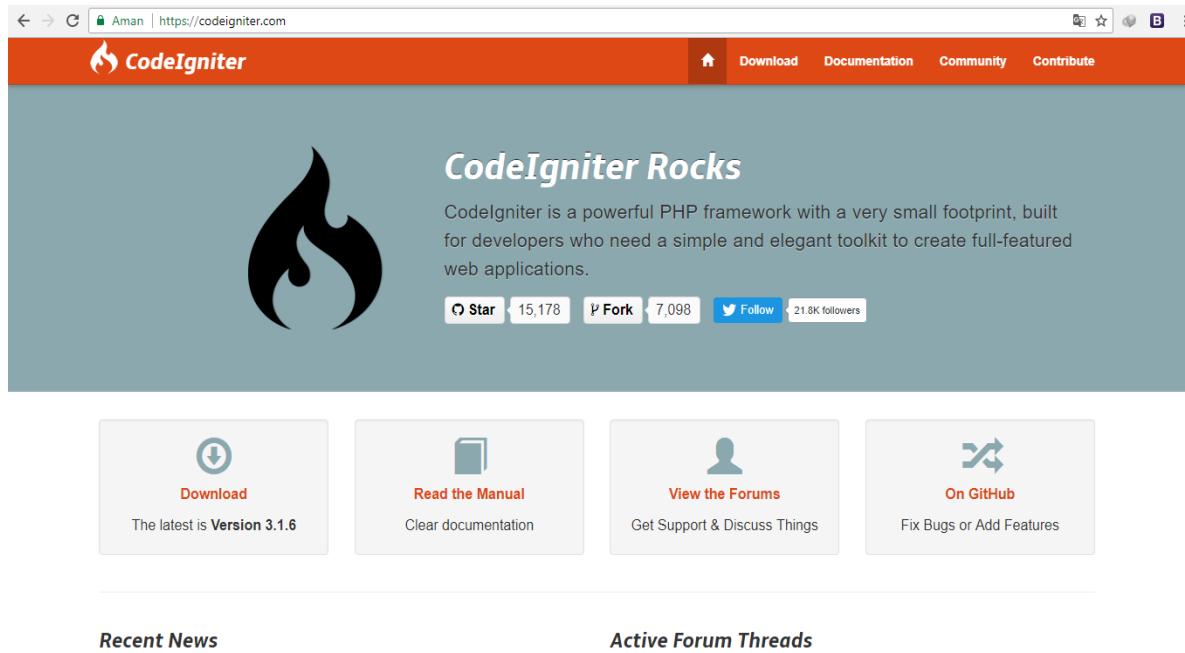
Adapun beberapa kelebihan dari codeigniter adalah :

- syntax yang terstruktur dan bersih
- kemudahan dalam menggunakannya
- codeigniter menyediakan fasilitas helper dan library yang dapat membantu developer dalam membuat pagination, session, manipulasi url dan lainnya secara cepat.yang akan kita pelajari pada tutorial codeigniter selanjutnya.
- keamanan yang sudah lumayan karena user atau pengakses aplikasi tidak berhubungan langsung dengan database.

Setelah selesai berkenalan dengan framework codeigniter, selanjutnya kita akan masuk ke tahap belajar cara menginstal framework codeigniter dan mulai menggunakannya.

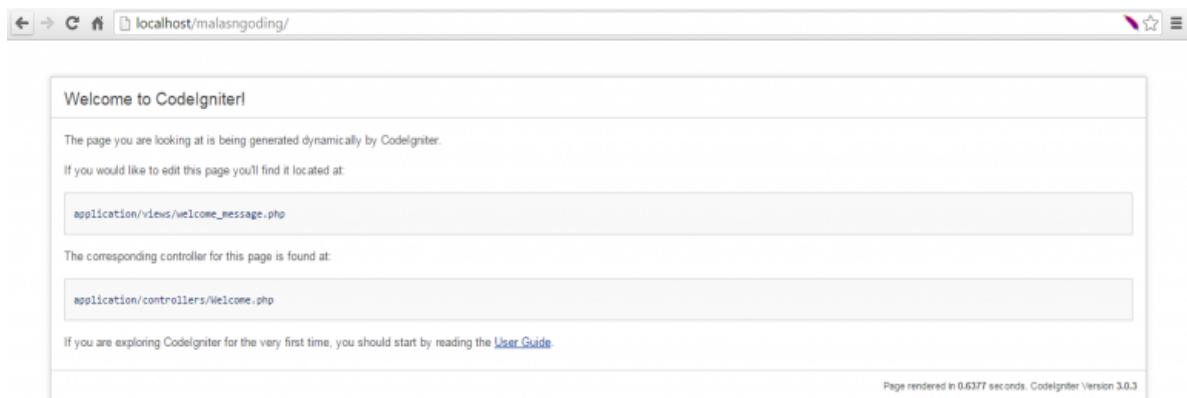
## Dasar Bagian 2 : Cara Instalasi Framework CodeIgniter

Instalasi Framework codeigniter caranya sangatlah mudah. anda dapat langsung mendownload CodeIgniter pada situs resminya di [www.codeigniter.com](https://codeigniter.com).



Setelah selesai mendownload framework codeigniter, kemudian ekstrak di folder htdocs (localhost). Ubah nama foldernya dengan nama project yang ingin teman-teman buat. Pada contoh ini saya mengubah nama folder nya dari "codeigniter.3.1.5" menjadi "malasngoding".

Terakhir tinggal langsung menjalankannya pada browser dengan alamat <http://localhost/malasngoding>

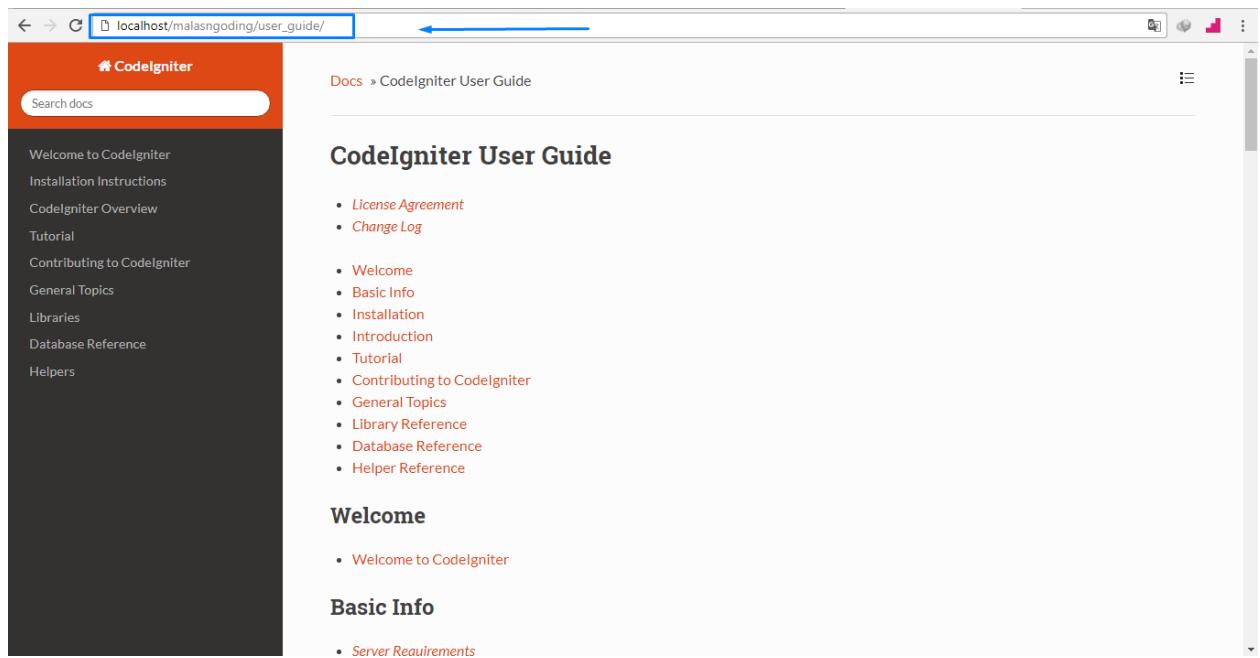


maka akan muncul tampilan seperti di atas yang menandakan bahwa codeigniter telah berhasil di install dan siap untuk di gunakan. tampilan di atas merupakan halaman default dasar dari codeigniter saat pertama kali di install.

#### Panduan Instalasi Codeigniter :

1. Download codeigniter di [www.codeigniter.com](http://www.codeigniter.com)
2. Ekstrak hasil download ke localhost (C:/XAMPP/htdocs )
3. Ubah nama folder nya dari Codeigniter 3.1.5 (tergantung versi yg kamu download) menjadi nama project yang mau di buat (malasngoding)
4. Jalankan pada browser “ http://localhost/nama\_folder\_project ”  
[\(http://localhost/malasngoding\)](http://localhost/malasngoding)
5. Selesai

Secara default Codeigniter menyediakan panduan cara penggunaan codeigniter pada folder codeigniter yang anda download tadi. yaitu terdapat pada folder *user\_guide*. dan untuk mempelajari panduan dari penggunaan codeigniter anda dapat langsung menjalankannya pada browser dengan alamat [http://localhost/malasngoding/user\\_guide/](http://localhost/malasngoding/user_guide/).



Segala cara penggunaan codeigniter bisa dibaca pada panduan yang telah di sediakan seperti pada gambar di atas.

## Dasar Bagian 3 : Mengenal Uri Segment Pada Codeigniter

Pada tutorial codeigniter ini saya akan menjelaskan tentang salah satu yang paling terpenting dalam konsep penggunaan codeigniter, yaitu **Uri Segment** pada codeigniter. jadi pada tutorial ini kita akan belajar tentang pengenalan dari uri segment codeigniter. apa pengertian uri segment pada codeigniter ?

URI adalah singkatan dari *Uniform Resource Identifier*. bisa kita bilang URI yang membantu kita dalam mengambil data melalui url codeigniter. cara penyebutan uri segment pada codeigniter sendiri misalnya segment 1, segment 2, segment 3 dan seterusnya. jika teman-teman perhatikan pada url codeigniter pada project codeigniter anda, pasti menjumpai index.php lalu di lanjutkan dengan nama class codeigniter anda kan. index.php di sini tidak berpengaruh dengan uri segment bahkan index.php dapat kita hilangkan untuk membuat url codeigniter kita lebih rapi.

Untuk tutorial menghilangkan index.php pada codeigniter akan saya bahas pada tutorial selanjutnya. kembali lagi ke penjelasan URI segment, yang dikatakan sebagai segment 1 pada codeigniter adalah nama class atau controller yang sedang anda jalankan. segment ke 2 adalah nama method atau function pada dari class/controller anda. dan segment 3 berupa data berbentuk id yang dikirim.

Jadi segment codeigniter dihitung dari setelah index.php pada codeigniter. dikatakan saja misalnya teman-teman memiliki url seperti contoh berikut ini

```
http://localhost/malasngoding/index.php/belajar/codeigniter/1
```

coba anda perhatikan pada contoh url di atas.

segment 1 = belajar

segment 2 = codeigniter

segment 3 = 1

untuk memberikan penjelasan yang lebih detail berikut ini adalah rumus cara mudah mengerti tentang url codeigniter

```
http://localhost/malasngoding/index.php/controller/method/id
```

data yang dikirimkan melalui url di codeigniter biasanya terletak pada segment 3. perhatikan contoh berikut ini.

buat sebuah controller baru dengan nama mobil.php

### **application/controllers/mobil.php**

```
<?php  
  
defined('BASEPATH') OR exit('No direct script access allowed');  
  
class Mobil extends CI_Controller {  
  
    public function warna(){  
        echo "Mobil itu berwarna " . $this->uri->segment('3');  
    }  
  
}
```

Coba anda perhatikan pada contoh di atas. saya membuat sebuah controller dengan nama mobil.php dan kemudian membuat method / function warna. pada function warna ini saya menampilkan isi dari URI segment ketiga ( `$this->uri->segment('3')` ).

Untuk mengakses uri segment bisa menggunakan syntax "`$this->uri->segment('urutan segment keberapa')`". maka hasilnya akan muncul.



nah, bisa anda perhatikan pada segment 3, segment 3 adalah "merah". maka hasil yang muncul adalah mobil itu berwarna merah. coba ubah merah menjadi biru



Mobil itu berwarna biru

perhatikan lagi contoh berikut ini dan semoga dapat memberi pemahaman lebih lagi tentang uri segment pada codeigniter.

### application/controllers/mobil.php

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Mobil extends CI_Controller {
    public function warna(){
        echo "Segment 1 adalah = " . $this->uri->segment('1') . "<br/>";
        echo "Segment 2 adalah = " . $this->uri->segment('2') . "<br/>";
```

```
        echo "Segment 3 adalah = " . $this->uri->segment('3') . "<br/>";
        echo "Segment 4 adalah = " . $this->uri->segment('4') . "<br/>";
        echo "Segment 5 adalah = " . $this->uri->segment('5') . "<br/>";
    }
}
```

kemudian akses alamat

<http://localhost/malasngoding/index.php/mobil/warna/hitam/merah/hijau>



Segment 1 adalah = mobil  
Segment 2 adalah = warna  
Segment 3 adalah = hitam  
Segment 4 adalah = merah  
Segment 5 adalah = hijau

Sekian tutorial tentang uri segment pada codeigniter.

## **Dasar Bagian 4 : Pengenalan Controller Codeigniter**

Pada tutorial codeigniter part 3 : Pengenalan Controller Codeigniter ini akan di jelaskan tentang cara membuat controller pada codeigniter, pengaturan dasar dan cara menggunakan controller pada codeigniter. seperti yang sudah di jelaskan pada tutorial sebelumnya tentang pengertian dan cara menggunakan codeigniter bahwa codeigniter menggunakan metode MVC untuk menciptaan kode atau syntax yang clean.

Di asumsikan untuk memahami konsep OOP terlebih dahulu sebelum mulai mempelajari codeigniter. karena codeigniter di bangun dengan menggunakan konsep OOP(Object Oriented Programming).

### **Pengenalan Controller**

Controller sesuai dengan artinya jika di artikan ke dalam bahasa indonesia yang berarti pengontrol atau pengatur, yang di maksud pengontrol atau pengatur di sini adalah controller yang berperan sebagai pengatur dari aksi pada aplikasi yang di bangun, seperti misalnya jika di codeigniter controller yang berperan paling penting dari mulai mengirimkan parameter, menangani inputan form(form handling), mengatur view dan model, mengaktifkan atau memanggil library dan helper codeigniter dan masih banyak lagi peran controller dalam membangun sebuah aplikasi dengan menggunakan framework php codeigniter.

Untuk memahami cara penggunaan controller codeigniter sekarang coba jalankan project Codeigniter yang sudah di install pada localhost, di sini nama folder project codeigniter saya adalah "malasngoding", sehingga untuk menjalankannya pada browser dapat di akses langsung dengan alamat <http://localhost/malasngoding/>.

Maka akan muncul halaman welcome codeigniter, ini karena pengaturan default controller yang di jalankan pertama kali adalah controller welcome. dapat di lihat pada application/controllers/welcome.php.

Controller ini memanggil view welcome\_message.php yang terletak pada folder view. view tidak kita jelaskan secara mendetail pada sub judul ini. melainkan akan kita bahas pada tutorial selanjutnya di ebook ini tentang cara membuat view pada codeigniter.

Secara default saat pertama kali menginstall codeigniter dan menjalankannya, controller codeigniter yang pertama kali di jalankan adalah controller welcome, saat anda menjalankannya pada browser maka yang pertama kali di jalankan adalah controller welcome. hal ini dapat di lihat pada pengaturan routes codeigniter yang menetapkan controller welcome sebagai controller default yang di jalankan. buka

project controller yang sudah anda letakkan pada localhost sesuai dengan tutorial sebelumnya di ebook ini. buka file pengaturan routes.php yang terletak di application/config/routes.php

### application/config/routes.php

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

dapat di lihat pada pengaturan routes codeigniter di atas, pada pengaturan default\_controller di setting controller “ welcome ”, terdapat juga pengaturan untuk menangani halaman 404 atau halaman yang di tampilkan jika tidak di temukannya data ada url. anda dapat mengatur halaman 404 anda dengan cara memasukkan controller yang ingin anda jadikan untuk menetapkan halaman 404 pada aplikasi anda.

Pada pengaturan \$route['translate\_uri\_dashes']=false adalah pengaturan untuk menetapkan nilai true atau false untuk izin penggunaan tanda “-” (dash) pada controller di url pada saat di jalankan.

controller default ini dapat di akses langsung dengan alamat

```
http://localhost/malasngoding
```



maka akan di munculkan controller default codeigniter yaitu controller welcome. atau bisa juga dengan mengakses controllernya dengan alamat

```
http://localhost/malasngoding/index.php/welcome
```

## Cara Membuat Controller Pada CodeIgniter

Jika sebelumnya kita menggunakan controller default bawaan saat pertama kali codeigniter di install, maka sekarang kita akan belajar membuat controller baru.

Untuk membuat controller baru pada codeigniter, Silahkan buat sebuah file baru pada application/controllers/. untuk contoh di sini kita akan membuat controller "belajar". Jadi kita harus membuat file belajar.php di application/controllers/.

Selanjutnya kita akan mendefinisikan class belajar dalam file belajar.php

### application/controllers/belajar.php

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

class Belajar extends CI_Controller {

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

    public function index(){
        echo "ini method index pada controller belajar";
    }

    public function halo(){
        echo "ini method halo pada controller belajar";
    }
}
```

## Penjelasan

Perhatikan pada controller yang kita buat di atas. pertama kali yang harus di lakukan adalah meng-extends controller baru ini dengan CI\_Controller.

```
class Belajar extends CI_Controller {
```

nama class harus di awali dengan huruf besar seperti contoh di atas. dan nama class harus sesuai dengan nama file controller yang di buat tadi. karena tadi saya membuat controller belajar.php maka penulisan class nya seperti atas.

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

syntax di atas berfungsi untuk mencegah akses langsung pada file controller.

kemudian anda dapat membuat function construct() untuk menjalankan fungsi yang anda inginkan pada saat controller di akses. biasanya pada fungsi construct() di letakkan fungsi untuk memanggil helper atau library (akan dipelajari pada sub selanjutnya).

kemudian lagi untuk contoh method pada controller ini saya membuat dua buah method yaitu index dan halo.

```
public function index(){
    echo "ini method index pada controller belajar | Cara membuat controller pada codeigniter";
}

public function halo(){
    echo "ini method halo pada controller belajar |Cara membuat controller pada codeigniter";
}
```

untuk menjalankan method index anda bisa mengaksesnya dengan alamat

```
localhost/malasngoding/index.php/belajar
```

untuk method index boleh tidak di tuliskan karena pada saat controller di akses, maka yang pertama di jalankan adalah method index

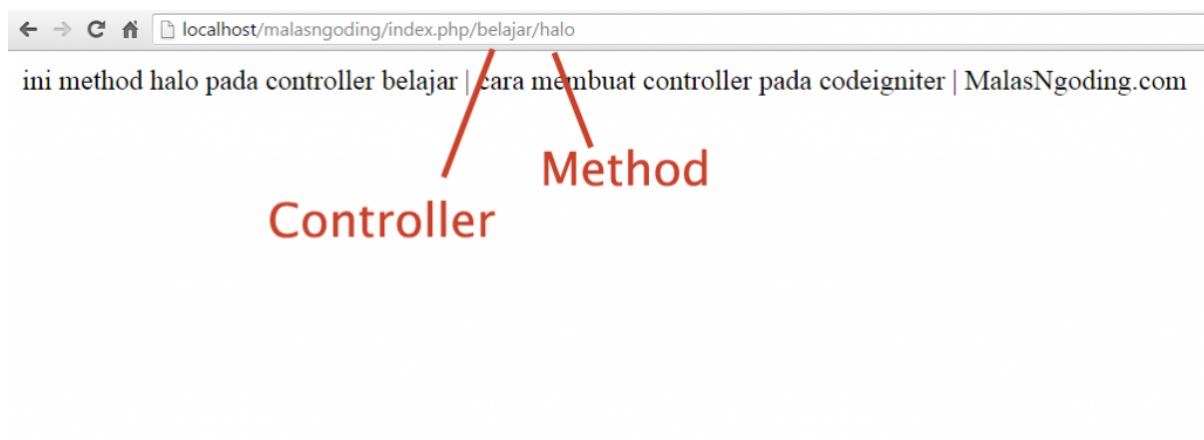
atau bisa juga

```
localhost/malasngoding/index.php/belajar/index
```



dan untuk mengakses function atau method halo dapat di akses dengan alamat

```
localhost/malasngoding/index.php/belajar/halo
```



Jika anda ingin menjadikan controller belajar ini sebagai controller default anda dapat melakukan setting nya di application/config/routes.php.

ubah pengaturan default\_controller menjadi belajar (nama controller yang ingin di set default).

```
$route['default_controller'] = 'belajar';
```

## Dasar Bagian 5 : Pengenalan View Pada CodeIgniter

Setelah mempelajari tentang cara membuat controller pada codeigniter dan pengertian serta cara menggunakan codeigniter pada tutorial sebelumnya, maka untuk kelanjutan dari tutorial codeigniter dasar kita akan membahas tentang cara membuat view pada codeigniter. seperti yang sudah di jelaskan bahwa view pada codeigniter berguna untuk meng-handle urusan tampilan dari aplikasi yang kita buat menggunakan codeigniter.

### Pengenalan View

View bertugas menampilkan user interface kepada user, sesuai dengan fungsi MVC yang memisahkan model, controller dan view sehingga memudahkan developer atau programmer dalam membuat pembaharuan serta dapat memudahkan developer bekerja dalam tim pada saat membangun aplikasi berbasis web menggunakan codeigniter.

sehingga web designer yang menangani tampilan user interface tidak perlu berurusan dengan back-end karena jatah kerja web designer ada pada view yang terletak pada folder view pada codeigniter. application/views/.

### Cara Membuat View Pada CodeIgniter

cara membuat view pada codeigniter caranya sangat mudah, anda hanya perlu membuat file baru pada folder view. Yaitu di *application/views/*.

buat view dengan nama file nya terserah anda. Di contoh ini, kita akan membuat view dengan nama view\_belajar.php

" application/views/view\_belajar.php "

langkah selanjutnya adalah membuat isi dari view.

application/views/view\_belajar.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Cara Membuat View Pada CodeIgniter | MalasNgoding.com</title>
</head>
<body>
    <h1>Cara Membuat View Pada CodeIgniter</h1>
    <h2>Ini adalah view view_belajar.php</h2>
    <h3>Ini adalah view yang di tampilkan pada controller belajar, method halo</h3>
</body>
</html>
```

dan untuk cara memanggil atau menampilkan view nya silahkan buka controller anda, di sini saya membuat controller belajar.php dan membuat method halo.

### **application/controllers/belajar.php**

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

class Belajar extends CI_Controller {

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

    }

    public function index(){
        echo "ini method index pada controller belajar | cara membuat
controller pada codeigniter | MalasNgoding.com";
    }

    public function halo(){
        $this->load->view('view_belajar');
    }

}
```

perhatikan pada method halo di atas. pada method halo ini saya menampilkan view view\_belajar.php

```
public function halo(){
    $this->load->view('view_belajar');
}
```

Teman-teman tidak perlu lagi menuliskan ekstensi '.php' pada saat memanggil view. syntax `$this->load->view()` dengan otomatis mengakses folder application/view codeigniter. dan coba kita jalankan method halo nya pada browser untuk menampilkan view view\_belajar.php

## Cara Membuat View Pada CodeIgniter

**Ini adalah view view\_belajar.php**

**Ini adalah view yang di tampilkan pada controller belajar, method halo**

dapat di perhatikan pada contoh di atas. view `view_belajar.php` pun muncul pada saat kita akses method halo.

Bagaimana cara membuat view di dalam subfolder. misalnya untuk sebagian view ada yang ingin anda kumpulkan pada sebuah folder untuk membuat view lebih rapi dan terstruktur. misalnya anda meletakkan view yang ingin anda panggil di dalam suatu folder. jika anda meletakkan view `view_belajar.php` di dalam folder `ngoding` di dalam view codeigniter.

`application/views/ngoding/view_belajar.php`

dan cara memanggil view `view_belajar.php` yang terletak di dalam folder `ngoding`, caranya anda hanya perlu memasukkan juga nama folder tempat view yang anda ingin tampilkan berada.

```
$this->load->view('ngoding/view_belajar');
```

## Cara Parsing Data ke View CodeIgniter

Untuk memarsing data dari controller ke view anda dapat memparsingnya dengan menggunakan bantuan array. jadi data yang akan di parsing kita masukkan ke array.

perhatikan contoh cara memarsing data ke view codeigniter berikut ini.

buka controller `belajar.php`

```
<?php
```

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

class Belajar extends CI_Controller {

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

    public function index(){
        echo "ini method index pada controller belajar | cara membuat controller pada codeigniter | MalasNgoding.com";
    }

    public function halo(){
        $data['nama_web'] = "MalasNgoding.com";
        $this->load->view('view_belajar',$data);
    }

}
```

## application/views/view\_belajar.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Cara Membuat View Pada CodeIgniter | MalasNgoding.com</title>
</head>
<body>
    <h1><?php echo $nama_web; ?></h1>
</body>
</html>
```

localhost/malasngoding/index.php/belajar/halo

# MalasNgoding.com

Perhatikan pada contoh di atas data di parsing pada degan memasukkan variabel ke dalam parameter kedua pada syntax \$this->load->view().

```
$data['nama_web'] = "MalasNgoding.com";
$this->load->view('view_belajar',$data);
```

dan dari view tinggal mengakses variabel \$nama\_web.

```
<h1><?php echo $nama_web; ?></h1>
```

kemudian bisa juga seperti cara berikut.

### **application/controllers/belajar.php**

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

class Belajar extends CI_Controller {

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

    }

    public function index(){
        echo "ini method index pada controller belajar | cara membuat
controller pada codeigniter | MalasNgoding.com";
    }

    public function halo(){
        $data = array(
            'judul' => "Cara Membuat View Pada CodeIgniter",
            'tutorial' => "CodeIgniter"
        );
        $this->load->view('view_belajar', $data);
    }

}
```

### **application/views/view\_belajar.php**

```
<!DOCTYPE html>
<html>
<head>
    <title>Cara Membuat View Pada CodeIgniter | MalasNgoding.com</title>
</head>
<body>
    <h2><?php echo $judul; ?></h2>
    <h3><?php echo $tutorial; ?></h3>
</body>
</html>
```



## Cara Membuat View Pada CodeIgniter

### CodeIgniter

Sekian tutorial cara membuat view pada codeigniter dan cara parsing data ke view codeigniter.

## Dasar Bagian 6 : Membuat Template Web Sederhana (Templating) Pada Codeigniter

Pada tutorial Codeigniter Dasar Bagian 5 ini, akan di jelaskan tentang membuat template website sederhana dengan menggunakan codeigniter. pada tutorial ini akan di jelaskan tentang cara menggunakan teknik multiple view pada codeigniter untuk membuat template website yang dinamis, sama seperti menggunakan include() pada bagian header dan footer pada penggunaan PHP native.

Tetapi di codeigniter kita membuat nya dengan men-load view codeigniter yang kita pisah-pisahkan sesuai keinginan, misalnya anda bisa memisahkan bagian header, footer, sidebar dan konten untuk mencegah pengulangan penulisan syntax dan memudahkan dalam hal memodifikasi template website.

### Membuat Template Web (Templating) Dengan Codeigniter

Buat sebuah controller yang menampilkan sebuah view. di sini saya membuat controller web.php dan sebuah view dengan nama v\_index.php.

#### application/controllers/web.php

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

class Web extends CI_Controller {

    function __construct(){
        parent::__construct();
        $this->load->helper('url');
    }

    public function index(){
        $data['judul'] = "Halaman depan";
        $this->load->view('v_index',$data);
    }
}
```

kemudian buat view v\_index.php

```
<!doctype html>
<html>
<head>
    <meta charset="UTF-8">
    <title>MalasNgoding | Membuat Template sederhana codeigniter</title>
    <link rel="stylesheet" type="text/css" href=<?php echo base_url() ?>assets/css/style.css">
</head>
<body>
```

```

<div id="wrapper">
    <header>
        <hgroup>
            <h1>MalasNgoding.com</h1>
            <h3>Membuat template sederhana codeigniter</h3>
        </hgroup>
        <nav>
            <ul>
                <li><a href="php echo base_url().'index.php/web' ?&gt;"Home</a></li>
                <li><a href="php echo base_url().'index.php/web/about' ?&gt;"About</a></li>
            </ul>
        </nav>
        <div class="clear"></div>
    </header>
    <section>
        <h1><?php echo $judul ?></h1>
        <p>
            do eiusmod
            minim veniam,
            commodo
            velit esse
            cupidatat non
            laborum.
            do eiusmod
            minim veniam,
            commodo
            velit esse
            cupidatat non
            laborum.
            </p>
            <p>
            do eiusmod
            minim veniam,
            commodo
            velit esse
            cupidatat non
            laborum.
            </p>
            <p>
            do eiusmod
            minim veniam,
            commodo
            velit esse
            cupidatat non
            laborum.
            </p>
        </section>
        <footer>
            <a href="http://www.malasngoding.com">MalasNgoding</a>
        </footer>
    </div>
</body>
</html>

```

oh iya jangan lupa untuk setting base\_url() codeigniter nya terlebih dulu agar kita bisa menghubungkan codeigniter dengan file css. file css nya saya buat pada folder project di dalam folder malasngoding/assets/css/style.css

```
body{  
    background: #eee;  
    color: #333;  
    font-family: sans-serif;  
    font-size:15px;  
}  
  
#wrapper{  
    background: #fff;  
    width: 1100px;  
    margin: 20px auto;  
}  
  
#wrapper header{  
    background: #232323;  
    padding: 20px;  
}  
  
#wrapper header hgroup{  
    float: left;  
    color: #fff;  
}  
#wrapper header nav{  
    float: right;  
    margin-top: 50px;  
}  
  
#wrapper header nav ul{  
    padding: 0;  
    margin: 0;  
}  
  
#wrapper header nav ul li{  
    float: left;  
    list-style: none;  
}  
  
#wrapper header nav ul li a{  
    padding: 15px;  
    color: #fff;  
    text-decoration: none;  
}  
.clear{  
    clear: both;  
}  
  
footer{  
    background: #232323;  
    padding: 20px;  
}  
  
footer a{  
    color: #fff;  
    text-decoration: none;  
}  
  
section{
```

```
    padding: 20px;  
}
```

## Setting Base Url pada Codeigniter

setting base\_url() buka pada file config.php di application/config/config.php

cari syntax berikut. Dan sesuai kan dengan letak project codeigniter kita.

```
$config['base_url'] = ' ';
```

Menjadi

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

setting dengan alamat project anda masing-masing. kemudian dapat di perhatikan pada contoh di atas bahwa untuk menghubungkan dengan file css , kita memerlukan bantuan base\_url().

```
<link rel="stylesheet" type="text/css" href="php echo base_url()<br/?>assets/css/style.css">
```

Hasil perintah di atas akan sama seperti

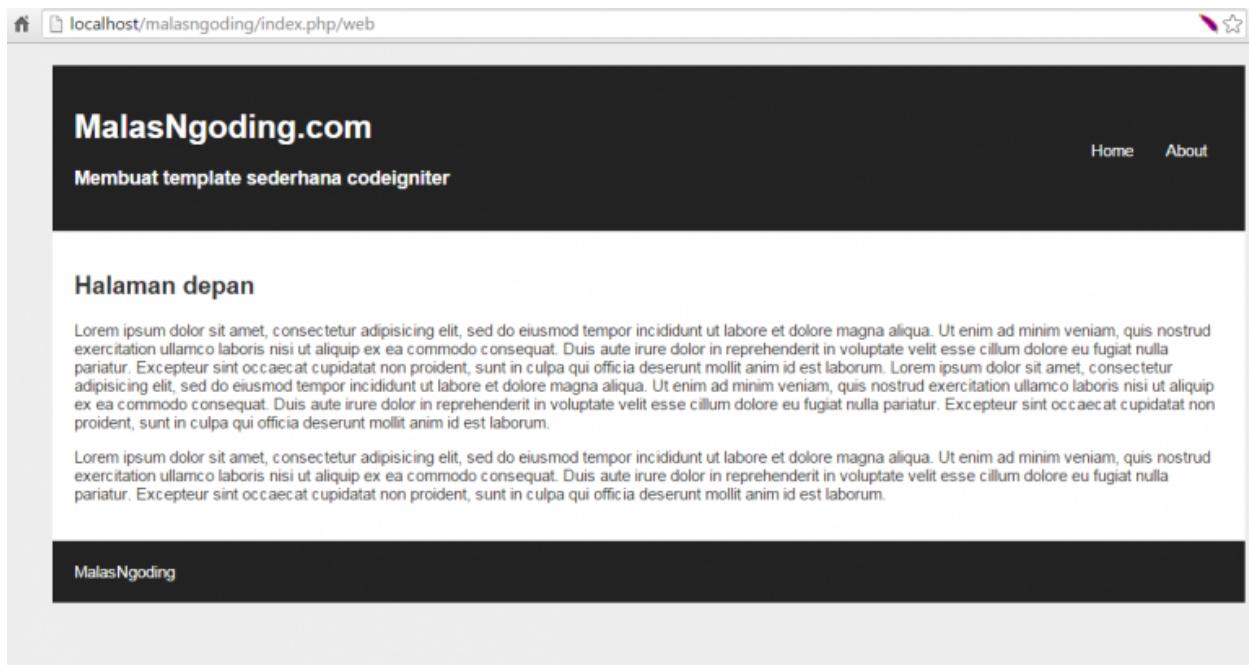
```
<link rel="stylesheet" type="text/css"  
href="http://localhost/malasngoding/assets/css/style.css">
```

Karena base\_url() nya sudah kita tentukan menjadi "localhost/malasngoding".

karena kita menggunakan base\_url() codeigniter, maka kita perlu mengaktifkan juga helper url seperti yang sudah kita buat pada controller web.php. silahkan perhatikan. Kita mengaktifkan helper url pada function construct().

jalankan controller web.php untuk melihat hasil dari template sederhana kita.

<http://localhost/malasngoding/index.php/web>



nah, tampilan web sederhana pun berhasil kita buat pada codeigniter. tapi halaman ini belum dinamis. kita masih perlu membuat bagian header dan footer secara berulang-ulang pada halaman lainnya.

## Membuat Halaman Template Dinamis Dengan Codeigniter

caranya kita harus memecah template ini menjadi beberapa bagian. yaitu header dan footer. sehingga template sederhana ini menjadi sebagai berikut.

### application/views/v\_header.php

```
<!doctype html>
<html>
<head>
    <meta charset="UTF-8">
    <title>MalasNgoding | Membuat Template sederhana codeigniter</title>
    <link rel="stylesheet" type="text/css" href=<?php echo base_url()>assets/css/style.css>
</head>
<body>
    <div id="wrapper">
        <header>
            <hgroup>
                <h1>MalasNgoding.com</h1>
                <h3>Membuat template sederhana codeigniter</h3>
            </hgroup>
            <nav>
                <ul>
                    <li><a href=<?php echo base_url().'index.php/web'>Home</a></li>
```

```

                <li><a href="<?php echo base_url().'index.php/web/about' ?>">About</a></li>
            </ul>
        </nav>
        <div class="clear"></div>
    </header>

```

### application/views/v\_footer.php

```

        <footer>
            <a href="http://www.malasngoding.com">MalasNgoding</a>
        </footer>
    </div>
</body>
</html>

```

### application/views/v\_index.php

```

<section>
<h1><?php echo $judul ?></h1>
<p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
    tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
    quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
    consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
    cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat
    non
    proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
    tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
    quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
    consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
    cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat
    non
    proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
</p>
<p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
    tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
    quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
    consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
    cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat
    non
    proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
</p>
</section>

```

sekarang template ini sudah menjadi tiga view dan cara memanggilnya dengan cara memanggil view secara berurut. dari v\_header, v\_index dan kemudian v\_footer.

### application/controllers/web.php

```

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

class Web extends CI_Controller {

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

```

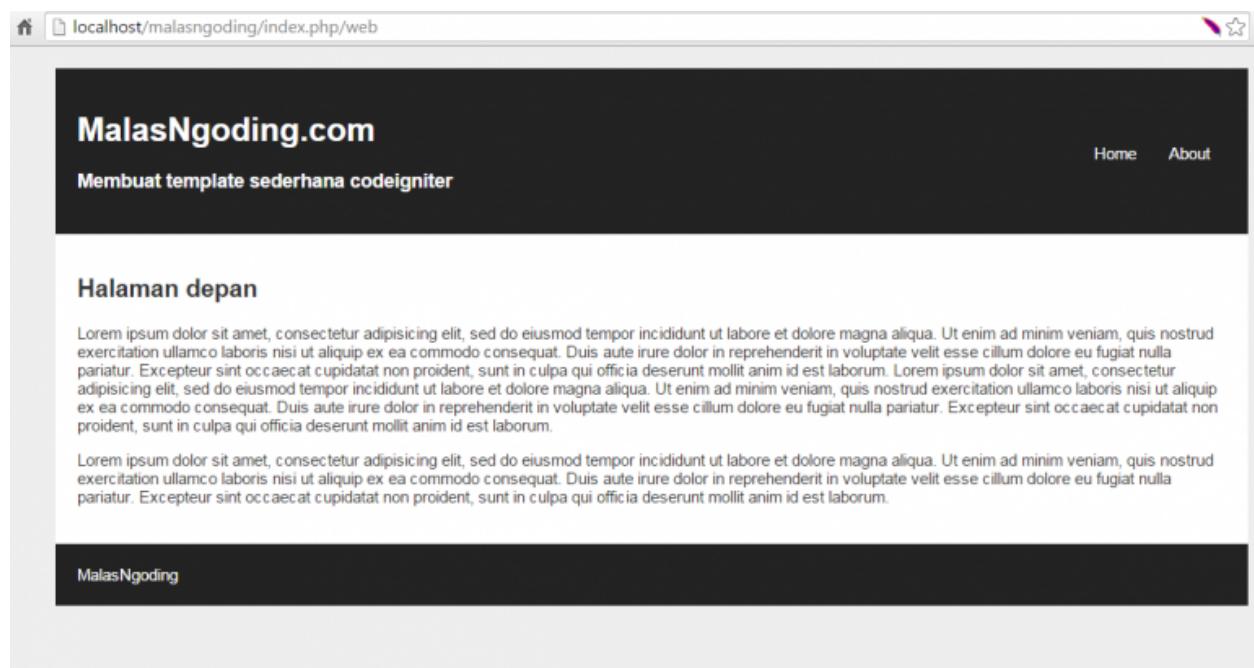
```

        $this->load->helper('url');
    }

    public function index(){
        $data['judul'] = "Halaman depan";
        $this->load->view('v_header',$data);
        $this->load->view('v_index',$data);
        $this->load->view('v_footer',$data);
    }
}

```

dan hasilnya juga akan sama, tetapi kelebihannya kita bisa dengan mudah membuat halaman lain dan tinggal memanggil v\_header dan v\_footer, hanya perlu mengganti v\_index untuk halaman lainnya.



perhatikan lagi pada hyperlink pada menu template di atas (v\_header.php) yang sudah kita setting untuk menuju method index dan method about.

```

<li><a href=<?php echo base_url().'index.php/web' ?>">Home</a></li>
<li><a href=<?php echo base_url().'index.php/web/about' ?>">About</a></li>

```

sekarang buat sebuah view lagi dengan nama v\_about.php

### **application/views/v\_about.php**

```

<section>
<h1><?php echo $judul ?></h1>
<p>
    halaman about yang bisa anda ubah sesuai keinginan. -
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
    tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,

```

```

        quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
        consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
        cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat
    non
        proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
        Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
        tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
        quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
        consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
        cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat
    non
        proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
    </p>
<p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
        tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
        quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
        consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
        cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat
    non
        proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
        Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
        tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
        quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
        consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
        cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat
    non
        proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
    </p>
<p>
        tutorial codeigniter malasngoding.com
    </p>
</section>

```

dan tambahkan lagi method about pada controller web untuk membuat halaman about.

### **Application/controllers/web.php**

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Web extends CI_Controller {

    function __construct(){
        parent::__construct();
        $this->load->helper('url');
    }

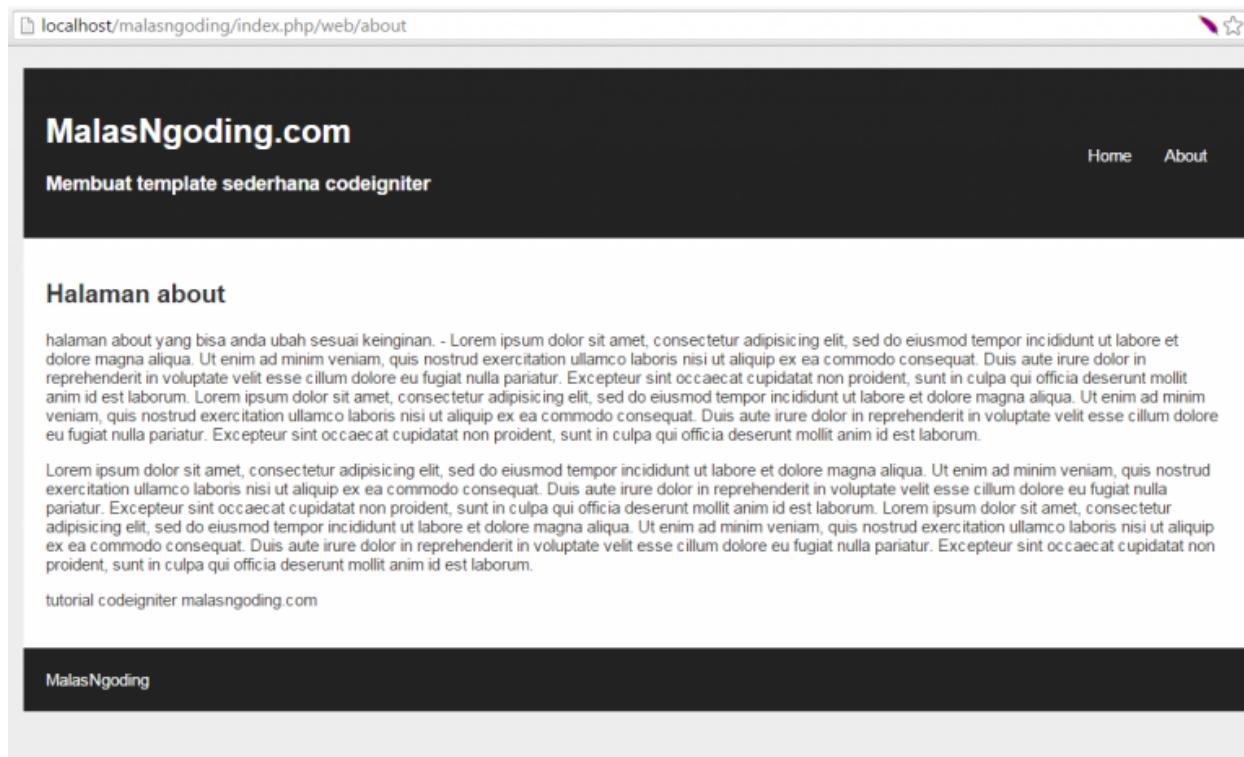
    public function index(){
        $data['judul'] = "Halaman depan";
        $this->load->view('v_header',$data);
        $this->load->view('v_index',$data);
        $this->load->view('v_footer',$data);
    }

    public function about(){
        $data['judul'] = "Halaman about";
        $this->load->view('v_header',$data);
        $this->load->view('v_about',$data);
        $this->load->view('v_footer',$data);
    }
}

```

Selanjutnya jalankan lagi pada browser,klik pada menu about atau bisa mengakses langsung

<http://localhost/malasngoding/index.php/web/about>



## Kesimpulan

Dengan membuat template web dengan multiple view atau memecahkan template menjadi beberapa bagian dan kemudian memanggilnya secara berurutan seperti contoh di atas yaitu dengan memanggil view header, index dan kemudian footer, maka akan memudahkan kita karena tidak perlu menuliskan syntax berulang-ulang misalnya tidak perlu lagi membuat header khusus untuk halaman index dan tidak perlu juga membuat header lagi untuk about.

intinya header dan footer dapat di gunakan secara bersama-sama dan hanya bagian konten nya saja yang berubah-ubah sesuai keinginan dengan cara memanggilnya pada method controller web.

## Dasar Bagian 7 : Form Validation Pada CodeIgniter

Pada tutorial CodeIgniter Bagian 6 ini, akan di jelaskan tentang membuat form validation pada codeigniter. terkadang kita memerlukan form validation dalam penginputan data, seperti misalnya kita menentukan form apa saja yang wajib diisi, form dengan format input tertentu, membuat konfirmasi password dan lainnya.

untuk membuat form validation pada codeigniter kita perlu memanggil atau membuka library form\_validation pada codeigniter.

berikut ini adalah cara membuat form validation pada codeigniter.

### Membuat Form Validation Pada CodeIgniter

Untuk contoh membuat form validation pada codeigniter, kita akan membuat sebuah controller dengan nama form.php

#### application/controllers/form.php

```
<?php

class Form extends CI_Controller{

    function __construct(){
        parent::__construct();
        $this->load->library('form_validation');
    }

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

    function aksi(){
        $this->form_validation->set_rules('nama','Nama','required');
        $this->form_validation->set_rules('email','Email','required');
        $this->form_validation->set_rules('konfir_email','Konfirmasi Email','required');

        if($this->form_validation->run() != false){
            echo "Form validation oke";
        }else{
            $this->load->view('v_form');
        }
    }
}
```

Perhatikan pada controller form.php di atas. pertama kita akan membuat dulu sebuah form pada view yang kita panggil pada method index agar di panggil pertama kali. di sini kita membuat view dengan nama v\_form.php.

oh ya untuk menggunakan library form validation codeigniter kita harus memanggil library form\_validation codeigniter terlebih dahulu. di sini saya memanggilnya pada function construct pada controller form. dapat di perhatikan pada construct() controller form di atas.

### application/views/v\_form.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Membuat form validation dengan Codeigniter | MalasNgoding.com</title>
</head>
<body>
    <h1>Membuat Form Validation dengan CodeIgniter</h1>
    <?php echo validation_errors(); ?>
    <?php echo form_open('form/aksi'); ?>
        <label>Nama</label><br/>
        <input type="text" name="nama"><br/>
        <label>Email</label><br/>
        <input type="text" name="email"><br/>
        <label>Konfirmasi Email</label><br/>
        <input type="text" name="konfir_email"><br/>
        <input type="submit" value="Simpan">
    </form>
</body>
</html>
```

Pada form di atas kita membuka form dengan function form codeigniter. perhatikan pada function form\_open() pada view di atas. kita menetapkan aksi dari form ke method aksi pada controller form.

```
<?php echo form_open('form/aksi'); ?>
```

dan saya juga membuat tiga buah form, yaitu nama, email dan konfir\_email. di sini saya ingin membuat form input tersebut wajib di isi, maka form validasi nya akan kita buat pada method aksi pada controller form.

```
function aksi(){
    $this->form_validation->set_rules('nama', 'Nama', 'required');
    $this->form_validation->set_rules('email', 'Email', 'required');
    $this->form_validation->set_rules('konfir_email', 'Konfirmasi
Email', 'required');

    if($this->form_validation->run() != false){
        echo "Form validation oke";
    }else{
        $this->load->view('v_form');
    }
}
```

untuk membuat form validation kita harus menentukan dulu form yang akan di berikan validasi. seperti contoh di atas saya membuat form validation pada form

input nama, email dan konfir\_email. function set\_rules di atas berarti kita menetapkan peraturan untuk form. cara penulisannya pada parameter pertama berikan nama form yang ingin di beri validasi, pada parameter kedua berikan kata yang di munculkan pada saat validasi. dan parameter ketiga isikan peraturan form. required berarti wajib, berarti form tersebut wajib di isi. sekarang kita jalankan form validation yang sudah kita buat dengan alamat <http://localhost/malasngoding/index.php/form>. lalu klik tombol submit jangan mengisi form nya maka akan muncul peringatan untuk mengisi form.



## Membuat Form Validation dengan CodeIgniter | MalasNgoding.com

The Nama field is required.

The Email field is required.

The Konfirmasi Email field is required.

Nama

Email

Konfirmasi Email

letak pesan peringatan form validationnya di tampilkan dengan fungsi berikut pada view form.

```
<?php echo validation_errors(); ?>
```

dan jika ingin menampilkan pesan dengan manual anda dapat menggunakan fungsi berikut

```
<?php echo form_error('nama'); ?>
<?php echo form_error('email'); ?>
<?php echo form_error('konfir_email'); ?>
```

Contoh full nya.

```
<!DOCTYPE html>
<html>
<head>
    <title>Membuat form validation dengan Codeigniter | MalasNgoding.com</title>
</head>
<body>
    <h1>Membuat Form Validation dengan CodeIgniter | MalasNgoding.com</h1>
    <?php echo validation_errors(); ?>
```

```
<?php echo form_open('form/aksi'); ?>
<label>Nama</label><br/>
<input type="text" name="nama"><br/>
<?php echo form_error('nama'); ?>
<label>Email</label><br/>
<input type="text" name="email"><br/>
<?php echo form_error('email'); ?>
<label>Konfirmasi Email</label><br/>
<input type="text" name="konfir_email"><br/>
<?php echo form_error('konfir_email'); ?>
<input type="submit" value="Simpan">
</form>
</body>
</html>
```

untuk selengkapnya tentang membuat form validation pada codeigniter anda dapat membacanya pada user\_guide yang sudah di sediakan oleh codeigniter pada project CI anda.

[http://localhost/malasngoding/user\\_guide/libraries/form\\_validation.html#the-form](http://localhost/malasngoding/user_guide/libraries/form_validation.html#the-form)

## **Dasar Bagian 8 : Upload File Dengan Codeigniter**

Membuat Upload File Dengan Codeigniter, itulah judul yang akan kita bahas pada tutorial codeigniter dasar bagian 7 ini. Sebenarnya tutorial upload file dengan codeigniter ini telah di bahas sebelumnya di situs belajar pemrograman tercinta kita, yaitu [www.malasngoding.com](http://www.malasngoding.com).

Pada panduan penggunaan codeigniter (user\_guide) juga telah di sediakan panduan tentang cara membuat upload file dengan menggunakan codeigniter. Teman-teman bisa mengaksesnya pada folder user\_guide di folder project codeigniter teman-teman.

Panduan upload file codeigniter:

[http://localhost/arca/user\\_guide/libraries/file\\_uploading.html](http://localhost/arca/user_guide/libraries/file_uploading.html)

codeigniter telah menyediakan library upload yang bisa kita gunakan dengan mudah. Biasanya untuk membuat upload file pada codeigniter kita bisa menggunakan library 'upload' dan helper 'form', serta helper 'url' untuk bantuan dalam membuat upload file dengan codeigniter.

Langsung saja masuk ke persiapan membuat upload file dengan codeigniter.

### **Membuat Upload File Dengan Codeigniter**

Untuk mulai membuat upload file dengan codeigniter, sediakan sebuah form pada view, dan buat juga sebuah controller. pada ebook tutorial codeigniter ini saya tidak menggabungkan semua controllernya, jadi setiap satu tutorial codeigniter saya pisah-pisah agar lebih memudahkan teman-teman yang baru belajar codeigniter untuk memahami syntax dan tidak tercampur.

di sini kita akan membuat controller baru dengan nama upload.php

**application/controllers/upload.php**

```
<?php  
  
class Upload extends CI_Controller{  
  
    function __construct(){  
        parent::__construct();  
        $this->load->helper(array('form', 'url'));  
    }  
  
    public function index(){
```

```

        $this->load->view('v_upload', array('error' => ' '));
    }

    public function aksi_upload(){
        $config['upload_path'] = './gambar/';
        $config['allowed_types'] = 'gif|jpg|png';
        $config['max_size'] = 100;
        $config['max_width'] = 1024;
        $config['max_height'] = 768;

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

        if ( ! $this->upload->do_upload('berkas')){
            $error = array('error' => $this->upload->display_errors());
            $this->load->view('v_upload', $error);
        }else{
            $data = array('upload_data' => $this->upload->data());
            $this->load->view('v_upload_sukses', $data);
        }
    }
}

```

bisa teman-teman perhatikan pada controller yang saya buat di atas, kita panggil dulu helper url dan helper form untuk membantu kita membuat upload file di codeigniter. helper form dan url nya saya panggil di function construct().

#### Ingat !

Helper url paling sering di gunakan saat membuat aplikasi dengan menggunakan codeigniter. Karena helper url berguna untuk menggunakan fungsi redirect() yang sudah disediakan oleh codeigniter. Yang berguna untuk membuat pengalihan halaman (hyperlink). Dan masih lagi kegunaan dari helper url yang akan anda jumpai saat membuat aplikasi dengan menggunakan framework codeigniter.

```

function __construct() {
    parent::__construct();
    $this->load->helper(array('form', 'url'));
}

```

kemudian pada function index nya saya tampilkan sebuah view v\_upload, pada view v\_upload ini ita akan membuat form upload nya.

```

public function index(){
    $this->load->view('v_upload', array('error' => ' '));
}

```

buat sebuah view dengan nama v\_upload sesuai dengan yang kita panggil pada method/function index di atas.

## Application/views/v\_upload.php

```
<html>
<head>
    <title>malasngoding.com</title>
</head>
<body>
    <center><h1>Membuat Upload File Dengan CodeIgniter |<br/>
MalasNgoding.com</h1></center>
    <?php echo $error;?>

    <?php echo form_open_multipart('upload/aksi_upload');?>

    <input type="file" name="berkas" />
    <br /><br />
    <input type="submit" value="upload" />
</form>

</body>
</html>
```

pada form upload di atas kita menggunakan function `form_open_multipart()`, function `form_open_multipart()` ini isinya sama dengan syntax berikut:

```
<form action="" enctype="multipart/form-data">
```

dan pada form nya kita beri nama “berkas”. dan kemudian aksi dari form tersebut kita arahkan ke **method aksi\_upload** pada **controller upload**.

```
public function aksi_upload(){
    $config['upload_path'] = './gambar/';
    $config['allowed_types'] = 'gif|jpg|png';
    $config['max_size'] = 100;
    $config['max_width'] = 1024;
    $config['max_height'] = 768;

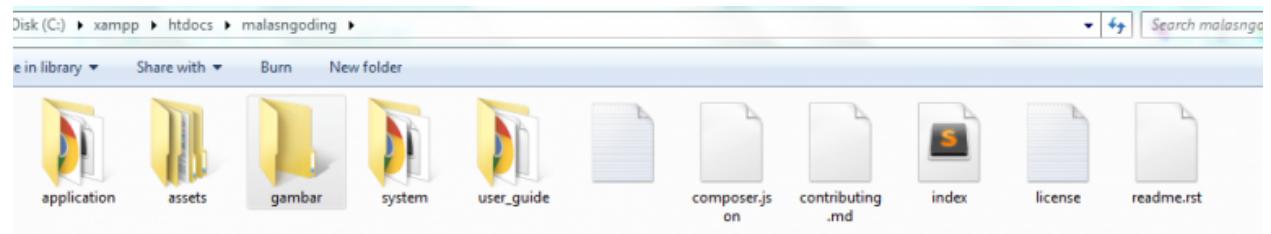
    $this->load->library('upload', $config);

    if ( ! $this->upload->do_upload('berkas')){
        $error = array('error' => $this->upload->display_errors());
        $this->load->view('v_upload', $error);
    }else{
        $data = array('upload_data' => $this->upload->data());
        $this->load->view('v_upload_sukses', $data);
    }
}
```

Yang paling penting ada di method `aksi_upload` ini. method ini akan kita jadikan sebagai pengatur dari proses upload file. terdapat variabel config dia atas, variabel ini berfungsi sebagai pengaturan upload file pada codeigniter. contohnya seperti berikut.

```
$config['upload_path']          = './gambar/';
$config['allowed_types']         = 'gif|jpg|png';
$config['max_size']              = 100;
$config['max_width']             = 1024;
$config['max_height']            = 768;
```

`$config['upload_path']` berfungsi untuk pengaturan kemana file akan di upload. di sini saya menetapkannya kedalam folder 'gambar' dalam directory root codeigniter (folder project kita). oleh karena itu kita buat dulu foldernya. Buat folder dengan nama "gambar". Jadi semua file yg di upload nantinya akan masuk ke dalam folder "gambar" ini.



kemudian lagi untuk membatasi ukuran file dan ekstensi file yang di perbolehkan untuk di upload bisa di setting pada

```
$config['allowed_types']      = 'gif|jpg|png'; // file yang di perbolehkan
$config['max_size']             = 100; // maksimal ukuran
$config['max_width']            = 1024; //lebar maksimal
$config['max_height']           = 768; //tinggi maksimal
```

setelah selesai membuat folder sebagai lokasi gambar yang di upload selanjutnya perhatikan syntax selanjutnya pada method aksi\_upload.

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

if ( ! $this->upload->do_upload('berkas')){
    $error = array('error' => $this->upload->display_errors());
    $this->load->view('v_upload', $error);
} else{
    $data = array('upload_data' => $this->upload->data());
    $this->load->view('v_upload_sukses', $data);
}
```

`$this->load->library('upload',$config)` berfungsi untuk memanggil library upload codeigniter dengan menggunakan pengaturan yang sudah di buat tadi pada variabel `$config`.

`$this->upload->do_upload()` berfungsi untuk melakukan aksi upload. di dalam parameternya berikan nama file upload tadi yang kita buat dengan nama 'berkas' seperti contoh di atas. dan kemudian dilakukan pengecekan, jika file tidak berhasil di upload atau jika file yang di upload tidak sesuai dengan pengaturan yang sudah kita buat maka pesan error kita masukkan dalam variabel `$error` dan parsing ke view `v_upload` untuk di tampilkan pesan errornya.

```
$error = array('error' => $this->upload->display_errors());
$this->load->view('v_upload', $error);
```

dan jika file berhasil di upload maka data file yang di upload kita masukkan kedalam variabel `$data` untuk kita parsing ke view `v_upload_sukses.php`.

### **application/views/v\_upload\_sukses.php**

```
<html>
<head>
    <title>malasngoding.com</title>
</head>
<body>

    <center><h1>Membuat      Upload      File      Dengan      CodeIgniter
MalasNgoding.com</h1></center>

    <ul>
        <?php foreach ($upload_data as $item => $value):?>
            <li><?php echo $item;?>: <?php echo $value;?></li>
        <?php endforeach; ?>
    </ul>

</body>
</html>
```

sekarang waktunya untuk mencoba apakah berhasil atau tidak.

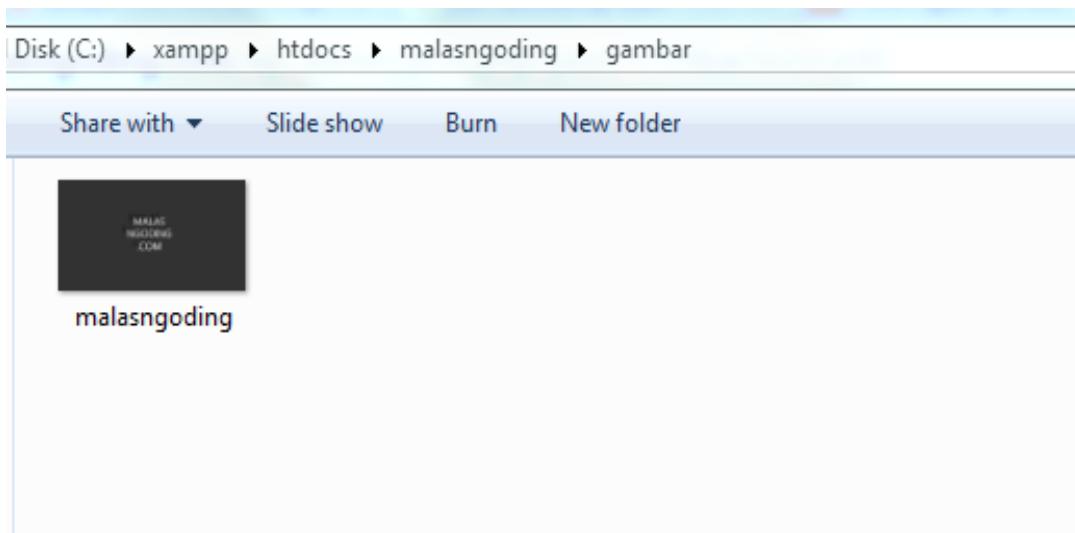


Pilih gambar yang ingin di upload dan submit (klik upload)

### Membuat Upload File Dengan CodeIgniter | MalasNgoding.com

- file\_name: malasngoding.png
- file\_type: image/png
- file\_path: C:/xampp/htdocs/malasngoding/gambar/
- full\_path: C:/xampp/htdocs/malasngoding/gambar/malasngoding.png
- raw\_name: malasngoding
- orig\_name: malasngoding.png
- client\_name: malasngoding.png
- file\_ext: png
- file\_size: 52.16
- is\_image: 1
- image\_width: 1366
- image\_height: 768
- image\_type: png
- image\_size\_str: width="1366" height="768"

file berhasil di upload. Yang muncul di pada gambar di atas adalah, informasi gambar yang di upload. Sesuai dengan perintah yang kita buat pada view **v\_upload\_sukses.php**. silahkan teman-teman sesuaikan sendiri pesan sukses nya.



Seperti yang tampil pada gambar di atas. Gambar pun berhasil di upload dan sudah masuk ke folder "gambar". Sesuai seperti intruksi yang kita buat pada method aksi\_upload.

## **Dasar Bagian 9 : Membuat Download File Dengan Codeigniter**

Setelah pada tutorial codeigniter sebelumnya saya telah menjelaskan tentang cara membuat upload file dengan codeigniter, maka tutorial ini saya akan menjelaskan dari kebalikannya yaitu Membuat Download File Dengan Codeigniter.

Silahkan di simak penjelasan berikut ini tentang cara membuat download file dengan codeigniter atau membuat link download dengan codeigniter.

Codeigniter telah menyediakan fasilitas untuk membuat download file yang sudah dikemas dalam function yang disediakan codeigniter, fasilitas ini dikemas oleh codeigniter ke dalam helper codeigniter. Yaitu helper download. sehingga kita tinggal memanggil helper download dan kemudian bisa langsung menggunakan nya untuk membuat link download atau membuat download file dengan codeigniter.

Berikut ini akan di jelaskan tentang cara membuat file download di codeigniter.

### **Membuat Download File Dengan Codeigniter**

Function download codeigniter telah dibungkus kedalam helper codeigniter, jadi untuk menggunakan fasilitas download di codeigniter sama caranya seperti menggunakan helper lainnya di codeigniter seperti pada tutorial sebelum-sebelumnya misalnya yang sudah saya jelaskan juga tentang penggunaan beberapa helper codeigniter. kita harus memanggil helper nya terlebih dulu.

Cara memanggil atau membuka helper 'download' pada codeigniter adalah dengan membuat syntax berikut.

```
$this->load->helper('download');
```

kemudian masuk ke tahap membuat download file dengan codeigniter nya. ada dua cara download file yang bisa kita buat, yaitu dengan cara menentukan sendiri isi file yang akan di download, atau membuat download file yang memang sudah tersedia. berikut ini adalah cara membuat download file yang isinya kita tentikan sendiri.

## Membuat Download File Kita Tentukan Sendiri

Di sini kita akan membuat sebuah download file dimana isi dari file yang di download tersebut kita tentukan sendiri. Perhatikan pada contoh berikut ini.

```
$isi = 'Here is some text!';
$nama_file = 'mytext.txt';
force_download($nama_file, $isi);
```

Syntax di atas adalah cara membuat download file yang isinya kita tentukan sendiri. dan function force\_download() adalah function yang sudah di sediakan oleh codeigniter untuk membuat aksi download. jadi untuk cara membuat download file yang kita tentukan sendiri isinya pada function force\_download(), masukkan isi file pada parameter pertama, dan masukkan nama dari file yang di download pada parameter kedua.

## Membuat Download File Yang Sudah Tersedia

Dan cara kedua adalah dengan membuat download file yang mana file nya sudah tersedia (tinggal download). Misalkan anda membuat website sekolah yang memiliki link download modul atau materi. Sebagai contohnya anggap saja segala modul atau materi yang akan di bagikan sudah tersedia di hosting/server, selanjutnya kita tinggal meletakkan lokasi dari file yang ingin di download tersebut.

Perhatikan contoh berikut.

```
force_download('/lokasi_file_malasngoding.txt', NULL);
```

Perhatikan pada syntax di atas, untuk membuat download file yang sudah ada, kita menggunakan function force\_download() juga. tetapi pada parameter pertamanya langsung masukkan lokasi file yang akan di download.

## Contoh Membuat Download File Dengan Codeigniter

Buat sebuah controller dengan nama download.php, dan view v\_download. sediakan juga file yang ingin kita buat sebagai file yang akan di download.

## application/controllers/download.php

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

class Download extends CI_Controller {

    function __construct(){
        parent::__construct();
        $this->load->helper(array('url','download'));
    }

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

    public function lakukan_download(){
        force_download('gambar/malasngoding.png',NULL);
    }

}
```

## application/views/v\_download.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Membuat download file dengan codeigniter | MalasNgoding.com</title>
</head>
<body>
    <h1>Membuat download file dengan codeigniter | MalasNgoding.com</h1>
    <br/>

    <a href="<?php echo base_url().'index.php/download/lakukan_download' ?>">Download file</a>
</body>
</html>
```

Gambar yang saya jadikan contoh untuk di download sudah saya sediakan pada directory root codeigniter dalam folder **gambar**. selanjutnya jalankan controller download.

*localhost/malasngoding/index.php/download*

Klik link download file dan kemudian file akan di download tanpa url berpindah alamat.

## Membuat download file dengan codeigniter | MalasNgoding.com

[Download file](#)



Sekianlah tutorial codeigniter part 9 : membuat download file dengan codeigniter.

## Dasar Bagian 10 : Membuat Library Codeigniter Sendiri

Pasti anda sudah mengenal apa itu library pada codeigniter karena pada tutorial sebelumnya sudah di jelaskan, dan di [www.malasngoding.com](http://www.malasngoding.com) juga telah di jelaskan berulang kali.

Kita juga di beri fasilitas oleh codeigniter untuk membuat library kita sendiri di codeigniter. apa kegunaan dari membuat library sendiri di codeigniter ? Terkadang kita banyak menuliskan syntax yang sama secara berulang-ulang. tetapi dengan membuat library sendiri, kita dapat dengan mudah memanggil function yang sudah kita buat sendiri dalam library yang kita buat secara berulang-ulang.

Langsung saja perhatikan penjelasan dan contoh berikut ini untuk membuat library sendiri di codeigniter.

Yang pertama kali yang harus di alakukan untuk cara membuat library sendiri di codeigniter adalah membuat file .php baru pada folder library yang sudah di sediakan oleh codeigniter, yaitu di application/libraries/.

Di sini saya membuat sebuah library dengan nama 'malasngoding'.

### application/libraries/malasngoding.php

```
<?php  
  
class Malasngoding{  
  
    function nama_saya(){  
        echo "Nama saya adalah malasngoding !";  
    }  
  
    function nama_kamu($nama){  
        echo "Nama kamu adalah ". $nama ." !";  
    }  
}
```

perhatikan pada syntax library malasngoding.php di atas. pertama kita deklarasikan terlebih dahulu class nya.

```
class Malasngoding{  
}
```

dan kemudian saya membuat dua buah method di sini dalam class malasngoding ini.

```
function nama_saya(){  
    echo "Nama saya adalah malasngoding !";  
}
```

```
function nama_kamu($nama){  
    echo "Nama kamu adalah ". $nama ." !";  
}
```

Sampai di tahap ini library buatan sendiri di codeigniter telah siap untuk di gunakan. library yang saya buat untuk contoh ini hanyalah library sederhana. nanti anda bisa mengembangkanya lagi sesuai dengan keinginan dan kebutuhan anda untuk memudahkan pekerjaan anda dalam membangun aplikasi menggunakan codeigniter.

Lalu langkah selanjut nya apa ? langkah selanjutnya adalah kita siap untuk memanggil dan menggunakan library “malasngoding” yang sudah di buat tadi. untuk mencoba library buatan sendiri ini siapkan sebuah controller untuk memanggil library ini. di sini saya membuat sebuah controller dengan nama ngoding.php

### **application/controllers/ngoding.php**

```
<?php  
defined('BASEPATH') OR exit('No direct script access allowed');  
  
class Ngoding extends CI_Controller {  
  
    function index(){  
        $this->load->library('malasngoding');  
        $this->malasngoding->nama_saya();  
        echo "<br/>";  
        $this->malasngoding->nama_kamu("Andi");  
    }  
}
```

Perhatikan, pada function index saya memanggil atau membuka terlebih dahulu library yang sudah kita buat tadi, yaitu library ‘malasngoding’.

```
$this->load->library('malasngoding');
```

Selanjutnya kita tinggal menggunakan function/method yang sudah kita buat tadi di dalam library malasngoding.

```
$this->malasngoding->nama_saya();  
echo "<br/>";  
$this->malasngoding->nama_kamu("Andi");
```

sekarang kita akan melihat hasilnya. silahkan akses halaman berikut.

*localhost/malasngoding/index.php/ngoding*



nah sampai di sini kita telah berhasil membuat dan menggunakan library yang kita buat sendiri.

## Penjelasan

Codeigniter menyediakan fasilitas untuk kita membuat library kita sendiri untuk memudahkan pekerjaan kita dengan menggunakan codeigniter. codeigniter juga telah menyediakan folder khusus sama seperti controller,model, view dan helper. untuk kita meletakkan library-library yang kita buat yaitu ada di application/libraries. Untuk membuat library kita buat file di folder libraries yang sudah di sediakan codeigniter, dan deklarasikan file library anda, dan kemudian berikan method atau function atau apapun yang anda perlukan.

dan selanjutnya untuk menggunakan library yang anda buat sendiri ini anda tinggal men-load atau membuka library yang anda buat tadi dengan syntax berikut

```
$this->load->library('di sini letakkan nama library yang anda buat')
```

dan library buatan sendiri pun selesai.

## **Dasar Bagian 11 : Menghilangkan Index.php Pada CodeIgniter**

Pada saat anda mengakses url/project CodeIgniter anda pasti anda melihat index.php pada url seperti misalnya

*"<http://localhost/malasngoding/belajar/index.php/pemrograman/>"*

index.php pada CodeIgniter dapat kita hilangkan dengan cara mudah sehingga anda tidak perlu lagi menggunakan index.php pada url CodeIgniter anda pada saat anda menjalankan project anda. jika index.php di hilangkan maka hasilnya akan seperti berikut ini

*"<http://localhost/malasngoding/belajar/pemrograman>"*

Jadi url CodeIgniter anda juga terlihat lebih rapi dan simpel dibandingkan dengan menggunakan index.php. silahkan anda simak penjelasan berikut ini untuk cara menghilangkan index.php pada CodeIgniter.

Untuk menghilangkan index.php pada url CodeIgniter yang pertama kali kita lakukan adalah mengubah pengaturan config "index page" CodeIgniter yang terletak pada file config.php di *application/config/config.php*.

Langkah selanjutnya agar url CodeIgniter dapat di akses tanpa menggunakan index.php adalah dengan cara membuat .htaccess pada folder root project CodeIgniter anda. anda bisa membaca dokumentasi dari CodeIgniter langsung pada user\_guide yang telah disediakan oleh CodeIgniter tentang cara penggunaan CodeIgniter dan cara menghilangkan index.php pada CodeIgniter dengan menggunakan .htaccess.

Dan untuk membaca user\_guide CodeIgniter sudah di jelaskan di atas pada tutorial CodeIgniter dasar. dan panduan dari user\_guide CodeIgniter tentang cara menghilangkan index.php pada CodeIgniter silahkan akses [http://localhost/malasngoding/user\\_guide/general/urls.html](http://localhost/malasngoding/user_guide/general/urls.html)

## Menghilangkan Index.php Pada CodeIgniter

Berikut adalah cara menghilangkan index.php pada url codeigniter. cara nya buka pengaturan "index\_page" codeigniter pada *application/config/config.php*, kemudian cari cari pengaturan index\_page seperti di bawah ini.

```
$config['index_page'] = 'index.php';
```

kemudian hapus index.php pada pengaturan sehingga menjadi

```
$config['index_page'] = '';
```

setelah menghapus index.php selanjutnya buat *htaccess* pada directory root codeigniter. cara nya buat file baru dengan nama ' .htaccess '. ingat tidak menggunakan ekstensi, karena .htaccess merupakan file yang tidak berekstensi. di awali dengan tanda titik '.', kemudian isi dengan perintah berikut

```
RewriteEngine On  
RewriteCond %{REQUEST_FILENAME} !-f  
RewriteCond %{REQUEST_FILENAME} !-d  
RewriteRule ^(.*)$ index.php/$1 [L]
```

simpan dan sekarang coba akses url anda. di sini untuk contoh, untuk mencoba apakah kita telah berhasil atau belum menghilangkan index.php pada codeigniter saya membuat controller baru dengan nama belajar.php. kemudian membuat sebuah method pemrograman.

### **application/controllers/belajar.php**

```
<?php  
defined('BASEPATH') OR exit('No direct script access allowed');  
  
class Belajar extends CI_Controller {  
  
    function pemrograman(){  
        echo "MENGHILANGKAN INDEX.PHP PADA CODEIGNITER | MALASNGODING.COM";  
    }  
}
```

jika biasa nya untuk mengakses method "pemrograman" dalam controller belajar di atas kita harus mengakses alamat

<http://localhost/malasngoding/index.php/belajar/pemrograman>

maka sekarang kita bisa langsung mengakses alamat berikut ini tanpa menggunakan index.php lagi.

<http://localhost/malasngoding/belajar/pemrograman>



Oke, sekianlah tutorial cara menghilangkan index.php pada codeigniter.

## Dasar Bagian 12 : Mengenal Model, Koneksi Database Codeigniter dan MySQL

Menghubungkan CodeIgniter Dengan Database MySQL, itulah yang akan saya jelaskan pada tutorial codeigniter dasar di bagian yang kesekian ini. Untuk menghubungkan CodeIgniter dengan database caranya sangat mudah, kita hanya perlu melakukan pengaturan pada file config/database.php.

Letak file untuk pengaturan database di CodeIgniter ada di :

*Application/config/database.php*

Pada file config database CodeIgniter ini kita harus memasukkan username dan password MySQL. Serta nama database yang akan digunakan.

Setelah menghubungkan CodeIgniter dengan database MySQL, selanjutnya untuk mencoba apakah database telah berhasil terkoneksi, kita akan langsung mencoba menampilkan data dari database. Ini di maksudkan untuk sekalian menjelaskan tentang model pada CodeIgniter. Karena pada tutorial sebelumnya telah dijelaskan tentang controller dan view pada CodeIgniter. Tapi model nya belum.

Baiklah langsung saja masuk ke cara menghubungkan CodeIgniter dengan database MySQL. Akan saya jelaskan sedikit tentang tutorial ini, untuk menghubungkan CodeIgniter dengan database yang pertama dilakukan adalah membuat dulu database nya. Kemudian menghubungkan CodeIgniter dengan database, kemudian lagi pada tutorial ini akan dijelaskan cara menampilkan data dari database sekalian untuk pengenalan model pada CodeIgniter. Silahkan simak penjelasan berikut ini untuk menghubungkan CodeIgniter dengan database MySQL.

## Membuat Koneksi Database CodeIgniter dan MySQL

Buat database dengan nama '**malasngoding**', dan di sini saya sertakan juga sebuah table dalam database "**malasngoding**". Table yang saya sertakan adalah sebuah table **user** yang berisi data nama, alamat dan pekerjaan.

Silahkan buat atau langsung import dari file SQL berikut.

**malasngoding.sql**

```
-- phpMyAdmin SQL Dump  
-- version 3.5.2.2
```

```

-- http://www.phpmyadmin.net
--
-- Host: 127.0.0.1
-- Generation Time: Jan 14, 2016 at 04:51 AM
-- Server version: 5.5.27
-- PHP Version: 5.4.7

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--

-- Database: `malasngoding`


-----
--



-- Table structure for table `user`


CREATE TABLE IF NOT EXISTS `user` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nama` varchar(50) NOT NULL,
  `alamat` text NOT NULL,
  `pekerjaan` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;

--



-- Dumping data for table `user`


INSERT INTO `user` (`id`, `nama`, `alamat`, `pekerjaan`) VALUES
(1, 'Andi', 'Surabaya', 'web programmer'),
(2, 'Budi', 'Bandung', 'Web Designer');

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Selanjutnya untuk menghubungkan codeigniter dengan database buka file config database codeigniter yang terletak di application/config/database.php

### **application/config/database.php**

Cari syntax berikut, lalu sesuaikan.

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

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

Pada file config database.php ini anda dapat melakukan pengaturan database anda dengan memasukkan hostname,username database, password dan nama database yang anda gunakan. pada tutorial ini username database adalah root, passwordnya kosong, host nya localhost dan nama database yang di gunakan adalah database dengan nama malasngoding.

```
'hostname' => 'localhost',
'username' => 'root',
'password' => '',
'database' => 'malasngoding',
```

Sehingga akan menjadi seperti berikut

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

Pada password dapat di kosongkan karena saya tidak menggunakan password. anda dapat menyesuaikan nya sendiri.

Sampai di sini kita telah selesai melakukan pengaturan untuk koneksi database codeigniter dengan database mysql. selanjutnya yang tidak kalah penting adalah memanggil library "database" codeigniter.

Karena untuk penggunaan yang berhubungan dengan database kita perlu membuka atau mengaktifkan library "database" milik si codeigniter ini.

Cara nya seperti memanggil library codeigniter yang lain. anda bisa memanggil library codeigniter dari function construct() pada controller, pada method atau pada autoload codeigniter. jadi kebetulan di sini saya akan membuka atau memanggil library database codeigniter pada config autoload codeigniter yang terletak pada application/config/autoload.php. jadi library "database" akan di load atau di jalankan secara otomatis saat project di jalankan.

### **application/config/autoload.php**

Cari pengaturan yang mengarah ke load library codeigniter seperti berikut

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

Kemudian tambahkan database. Dan aka menjadi seperti berikut.

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

Dan codeigniter pun berhasil di hubungkan dengan database mysql. bagaimana cara memeriksanya sudah terhubung atau belum. jadi pada tutorial ini kita akan menampilkan data dari table user yang sudah di buat tadi untuk sekalian berkenalan dengan model codeigniter.

## **Pengenalan Model Pada CodeIgniter**

Selanjutnya untuk menampilkan data dari table user tadi buat sebuah model dengan nama terserah anda.(model dibuat di dalam folder model di application/model). di sini saya membuat sebuah model dengan nama "m\_data.php".

### **application/models/m\_data.php**

```
<?php  
  
class M_data extends CI_Model{  
    function ambil_data(){  
        return $this->db->get('user');  
    }  
}
```

Perhatikan syntax di atas. saat membuat class model m\_data, penulisannya di anjurkan harus di awali dengan huruf besar. kemudian meng-extend model m\_data yang sudah di buat dengan CI\_Model. kemudian saya membuat sebuah function ambil\_data() pada model m\_data ini. function ambil\_data() ini saya buat untuk mengambil data pada table user.

Syntax `$this->db->get()` di gunakan untuk mengambil data dari database. dan nama table yang ingin di ambil data nya diletakkan dalam parameternya sehingga menjadi

```
$this->db->get('user');
```

Kemudian lagi perhatikan syntax return yang berfungsi untuk mengambilkan data yang di tangkap pada controller yang memanggil function ambil\_data() ini.

Sampai di sini model sudah jadi. langkah selanjutnya buat sebuah function pada controller anda. di sini saya membuat function user() pada controller belajar.php.

### **application/controllers/belajar.php**

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

class Belajar extends CI_Controller {

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

    function user(){
        $data['user'] = $this->m_data->ambil_data()->result();
        $this->load->view('v_user.php',$data);
    }
}
```

Perhatikan pada function user di atas. syntax

```
$this->m_data->ambil_data()
```

berfungsi untuk memanggil function ambil\_data() pada model m\_data. dan fungsi result() berguna untuk menjadikannya array. data yang di ambil dari fungsi ambil\_data pada model m\_data saya masukkan kedalam variabel untuk di parsing ke dalam view v\_user.php.

baca juga cara membuat view pada codeigniter dan memarsing data ke view codeigniter. cara memarsing data kedalam view codeigniter juga telah di jelaskan pada Dasar **Bagian 4 : Pengenalan View Pada Codeigniter**.

perhatikan juga pada controller belajar.php di atas. karena di sini kita akan menggunakan model m\_data. maka jangan lupa juga untuk memanggil model m\_data dengan syntax seperti berikut

```
$this->load->model('m_data')
```

Selanjutnya buat sebuah view dengan nama v\_user.php. pada view ini kita akan menampilkan data dari database yang kita parsing.

### application/views/v\_user.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Menghubungkan codeigniter dengan database mysql</title>
</head>
<body>
    <h1>Mengenal Model Pada Codeigniter | MalasNgoding.com</h1>
    <table border="1">
        <tr>
            <th>Nama</th>
            <th>Alamat</th>
            <th>Pekerjaan</th>
        </tr>
        <?php foreach($user as $u) { ?>
        <tr>
            <td><?php echo $u->nama ?></td>
            <td><?php echo $u->alamat ?></td>
            <td><?php echo $u->pekerjaan ?></td>
        </tr>
        <?php } ?>
    </table>
</body>
</html>
```

Perhatikan pada contoh di atas. kita menampilkan data dengan menggunakan `foreach()`. variabel \$user akan menjadi \$u. \$user adalah variabel yang di parsing dari controller tadi dan berisi data user dalam bentuk array. dan akses alamat controller user untuk melihat hasilnya.

<http://localhost/malasngoding/index.php/belajar/user>

## Mengenal Model Pada Codeigniter | MalasNgoding.com

Nama	Alamat	Pekerjaan
Andi	Surabaya	web programmer
Budi	Bandung	Web Designer

Sekian tutorial codeigniter cara menghubungkan codeigniter dengan database mysql.

## Dasar Bagian 13 : Membuat CRUD Dengan Codeigniter

Setelah pada tutorial codeigniter sebelum-sebelumnya kita telah belajar tentang pengertian dan cara menggunakan codeigniter, dan menggunakan library dan helper codeigniter. dan sampai pada tutorial codeigniter ini kita akan belajar tentang cara membuat crud dengan codeigniter. tentu anda sudah familiar dengan kata CRUD yang merupakan singkatan dari *Create, Read, Update dan Delete*.

CRUD maksudnya membuat input data ke database, menampilkan data dari database, mengedit mengupdate data pada database serta menghapus data pada database. CRUD sangat lah penting jika anda menggunakan operasi yang menggunakan database atau pengelolaan data pada database dengan codeigniter seperti misalnya menampilkan data dari database dengan codeigniter, menginput atau menambah data ke database, mengedit dan menghapus data.

Pada Tutorial Membuat CRUD dengan Codeigniter ini kita akan belajar tahap awal untuk membuat CRUD dengan codeigniter, mulai dari konfigurasi database dan mengambil data dari database.

silahkan anda simak contoh dan penjelasan berikut ini untuk membuat crud dengan codeigniter.

### Koneksi Database

baiklah, pertama-tama siapkan dulu database yang akan anda gunakan untuk belajar cara menampilkan data dari database dengan codeigniter. di sini saya membuat sebuah database baru dengan nama *malasngoding*. selanjutnya buat table user yang berisi column id, nama, alamat dan pekerjaan. silahkan import saja sql berikut ini untuk mempercepat.

#### user.sql

```
-- phpMyAdmin SQL Dump
-- version 3.5.2.2
-- http://www.phpmyadmin.net
--
-- Host: 127.0.0.1
-- Generation Time: Jan 20, 2016 at 10:20 AM
-- Server version: 5.5.27
-- PHP Version: 5.4.7

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";
```

```

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

-- 
-- Database: `malasngoding`
-- 

-- -----
-- 

-- 
-- Table structure for table `user`


CREATE TABLE IF NOT EXISTS `user` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nama` varchar(50) NOT NULL,
  `alamat` text NOT NULL,
  `pekerjaan` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;

-- 
-- Dumping data for table `user`
-- 


INSERT INTO `user` (`id`, `nama`, `alamat`, `pekerjaan`) VALUES
(1, 'Andi', 'Surabaya', 'web programmer'),
(2, 'Budi', 'Bandung', 'Web Designer');

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

kebetulan di sini saya telah memiliki beberapa record data, silahkan anda input dulu untuk dapat memastikan apakah data berhasil muncul nantinya. kemudian silahkan anda aktifkan library database karena kita akan menggunakan model codeigniter untuk membuat crud dengan codeigniter.

jika teman-teman belum membaca tutorial sebelum nya tentang cara menghubungkan codeigniter dengan database mysql. silahkan baca dulu pada pembahasan sebelumnya.

silahkan sesuaikan pengaturan database pada file config database.php codeigniter di

### **application/config/database.php**

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

```

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

```

sesuaikan dengan pengaturan database anda. nama database, username dan password mysql anda.

Sampai disini kita telah selesai melakukan konfigurasi untuk koneksi database codeigniter dan mysql.

## Menampilkan Data Dari Database

Kemudian lagi untuk menampilkan data dari database, kita akan membuat sebuah controller yang berisi parsing data yang kita ambil dari model ke view codeigniter. buat sebuah controller dengan nama crud.php

### application/controllers/crud.php

```

<?php

class Crud extends CI_Controller{

    function __construct(){
        parent::__construct();
        $this->load->model('m_data');
        $this->load->helper('url');
    }

    function index(){
        $data['user'] = $this->m_data->tampil_data()->result();
        $this->load->view('v_tampil',$data);
    }
}

```

pada controller crud.php ini pertama kita panggil atau buka dulu model m\_data, karena untuk operasi database nya akan kita buat pada model m\_data. kemudian pada function index kita menampilkan data dengan function tampil\_data yang kita buat dalam model m\_data untuk mengambil data dari database. kemudian memarsing nya ke view v\_tampil.

```

$data['user'] = $this->m_data->tampil_data()->result();
$this->load->view('v_tampil',$data);

```

oh ya jangan lupa untuk load library database. di sini saya load pada config autoload.php codeigniter.

### **application/config/autoload.php**

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

kemudian buat sebuah model dengan nama m\_data.php

### **application/models/m\_data.php**

```
<?php  
  
class M_data extends CI_Model{  
    function tampil_data(){  
        return $this->db->get('user');  
    }  
}
```

Syntax berikut

```
return $this->db->get('user');
```

berfungsi untuk mengambil data dari database, nama table yang akan kita ambil datanya kita tuliskan pada parameter seperti di atas. jadi data yang di ambil dari database ini kita kembalikan ke pemanggil fungsi ini nantinya dengan return.

Silahkan baca panduan operasi query database di user\_guide codeigniter di

```
http://localhost/malasngoding/user\_guide/database/examples.html
```

kemudian lagi buat sebuah view dengan nama v\_tampil.php

### **application/views/v\_tampil.php**

```
<!DOCTYPE html>  
<html>  
<head>  
    <title>Membuat CRUD dengan CodeIgniter | MalasNgoding.com</title>  
</head>  
<body>  
    <center><h1>Membuat CRUD dengan CodeIgniter | MalasNgoding.com</h1></center>  
    <center><?php echo anchor('crud/tambah','Tambah Data'); ?></center>  
    <table style="margin:20px auto;" border="1">  
        <tr>  
            <th>No</th>  
            <th>Nama</th>  
            <th>Alamat</th>
```

```

<th>Pekerjaan</th>
<th>Action</th>
</tr>
<?php
$no = 1;
foreach($user as $u) {
?>
<tr>
<td><?php echo $no++ ?></td>
<td><?php echo $u->nama ?></td>
<td><?php echo $u->alamat ?></td>
<td><?php echo $u->pekerjaan ?></td>
<td>
<?php echo anchor('crud/edit/'.$u->id,'Edit'); ?>
<?php echo anchor('crud/hapus/'.$u->id,'Hapus'); ?>
</td>
</tr>
<?php } ?>
</table>
</body>
</html>

```

Perhatikan function `anchor()` di atas. `anchor()` adalah function codeigniter yang berfungsi untuk membuat *hyperlink*. jadi pada parameter pertama di function `anchor()` letakkan link tujuan, dan pada parameter keduanya letakkan text yang akan di munculkan seperti yang terdapat pada contoh di atas.

Syarat untuk menggunakan function `anchor` ini anda harus men-load dulu helper url codeigniter seperti pada function `construct()` di atas.

Selanjutnya tinggal kita jalankan saya pada browser untuk melihat hasil contoh menampilkan data dari database dengan codeigniter.

```
http://localhost/malasngoding/crud
```

Perhatikan pada alamat url controller crud di atas. kenapa kita tidak lagi menggunakan index.php, karena saya telah menghilangkannya dengan membuat htaccess di codeigniter untuk dapat menjalankan controller codeigniter tanpa menggunakan index.php. tutorial cara menghilangkan index.php pada url codeigniter sudah saya bahas pada tutorial sebelumnya di [malasngoding.com](http://malasngoding.com).

The screenshot shows a browser window with the URL [localhost/malasngoding/crud](http://localhost/malasngoding/crud). The page title is "Membuat CRUD dengan CodeIgniter | MalasNgoding.com". Below the title is a blue link "Tambah Data". A table displays two rows of data:

No	Nama	Alamat	Pekerjaan	Action
1	Andi	Surabaya	web programmer	<a href="#">Edit</a> <a href="#">Hapus</a>
2	Budi	Bandung	Web Designer	<a href="#">Edit</a> <a href="#">Hapus</a>

sekian cara menampilkan data dari database dengan menggunakan CodeIgniter.

## Menginput Data Ke Database

Pada tutorial ini kita akan belajar cara menginput data ke database dengan CodeIgniter. Seperti pada PHP Native, pada tutorial ini kita akan menginput data ke database menggunakan form biasa. Silahkan simak contoh dan penjelasan berikut ini tentang cara membuat CRUD dengan CodeIgniter menginput data ke database.

Note :

Setting `base_url` CodeIgniter dengan nama project. Nama project CI saya pada tutorial ini adalah `malasngoding`. sehingga setting `base_url` dengan "`http://localhost/malasngoding/`"

### application/config/config.php

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

Pertama kita akan membuat dulu sebuah method pada controller `crud.php`. Di sini saya membuat method atau function `tambah()` yang berisi perintah untuk menampilkan `v_input`. View `v_input` akan saya jadikan sebagai form yang berguna sebagai form inputan, di mana data yang di input di sini akan masuk ke database.

### application/controllers/crud.php

```
<?php

class Crud extends CI_Controller{

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

```

        $this->load->helper('url');

    }

    function index(){
        $data['user'] = $this->m_data->tampil_data()->result();
        $this->load->view('v_tampil',$data);
    }

    function tambah(){
        $this->load->view('v_input');
    }

}

```

dan buat sebuah view yang kita jadikan sebagai form inputan data nya. saya buat view dengan nama v\_input.php

### **application/views/v\_input.php**

```

<!DOCTYPE html>
<html>
<head>
    <title>Membuat CRUD dengan CodeIgniter | MalasNgoding.com</title>
</head>
<body>
    <center>
        <h1>Membuat CRUD dengan CodeIgniter | MalasNgoding.com</h1>
        <h3>Tambah data baru</h3>
    </center>
    <form action=<?php echo base_url(). 'crud/tambah_aksi'; ?>" method="post">
        <table style="margin:20px auto;">
            <tr>
                <td>Nama</td>
                <td><input type="text" name="nama"></td>
            </tr>
            <tr>
                <td>Alamat</td>
                <td><input type="text" name="alamat"></td>
            </tr>
            <tr>
                <td>Pekerjaan</td>
                <td><input type="text" name="pekerjaan"></td>
            </tr>
            <tr>
                <td></td>
                <td><input type="submit" value="Tambah"></td>
            </tr>
        </table>
    </form>
</body>
</html>

```

dapat anda lihat di sini, pada form saya menentukan aksi dari form, saya arahkan ke method tambah\_aksi pada controller crud. oleh sebab itu kita buat lagi sebuah method atau function untuk meng-handle inputan pada form ini.

### **application/controllers/crud.php**

```

<?php

class Crud extends CI_Controller{

    function __construct(){
        parent::__construct();
        $this->load->model('m_data');
        $this->load->helper('url');

    }

    function index(){
        $data['user'] = $this->m_data->tampil_data()->result();
        $this->load->view('v_tampil',$data);
    }

    function tambah(){
        $this->load->view('v_input');
    }

    function tambah_aksi(){
        $nama = $this->input->post('nama');
        $alamat = $this->input->post('alamat');
        $pekerjaan = $this->input->post('pekerjaan');

        $data = array(
            'nama' => $nama,
            'alamat' => $alamat,
            'pekerjaan' => $pekerjaan
        );
        $this->m_data->input_data($data,'user');
        redirect('crud/index');
    }

}

```

coba anda perhatikan pada function tambah\_aksi tersebut. pertama kita menangkap inputan dari form dengan function `$this->input->post(' nama form input ')`. kemudian menjadikannya array

```

$data = array(
    'nama' => $nama,
    'alamat' => $alamat,
    'pekerjaan' => $pekerjaan
);

```

dan menginput data ke database dengan menggunakan model m\_data yang sudah saya jelaskan pada tutorial sebelumnya.

```
$this->m_data->input_data($data,'user');
```

jadi di sini pada parameter pertama nya saya input array data yang berisi data-data yang di input. dan pada parameter kedua nya saya beri nama dari table nya.(table

tujuan tempat menyimpan data inputan). dan kemudian mengalihkannya ke method index

```
redirect('crud/index');
```

sekarang buat sebuah function input\_data pada model m\_data.

### **application/models/m\_data.php**

```
<?php  
  
class M_data extends CI_Model{  
    function tampil_data(){  
        return $this->db->get('user');  
    }  
  
    function input_data($data,$table){  
        $this->db->insert($table,$data);  
    }  
}
```

Selesai, sekarang waktunya mencoba hasil dari tutorial input data ke database ini.

localhost/malasngoding/crud/tambah

**Membuat CRUD dengan CodeIgniter | MalasNgoding.com**

**Tambah data baru**

Nama	Rudi
Alamat	Semarang
Pekerjaan	Web Master
<input type="button" value="Tambah"/>	

dan klik tombol tambah

## Membuat CRUD dengan CodeIgniter | MalasNgoding.com

[Tambah Data](#)

No	Nama	Alamat	Pekerjaan	Action
1	Andi	Surabaya	web programmer	<a href="#">Edit</a> <a href="#">Hapus</a>
2	Budi	Bandung	Web Designer	<a href="#">Edit</a> <a href="#">Hapus</a>
3	Rudi	Semarang	Web Master	<a href="#">Edit</a> <a href="#">Hapus</a>

nah dapat anda lihat data yang barusan di input pun sudah berhasil tersimpan dan muncul pada table data user di atas. sekian lah tutorial membuat crud dengan codeigniter : input data ke database. selanjut nya akan kita lanjutkandengan membuat tombol/link hapus data dengan codeigniter.

## Menghapus Data Dari Database

Untuk kelanjutan dari tutorial membuat crud dengan codeigniter ini, kita akan membahas cara membuat tombol/link hapus data dari database. silahkn anda simak contoh dan penjelasan tentang cara membuat link/tombol untuk menghapus data pada database dengan codeigniter.

Untuk membuat hapus data caranya sangat mudah, sama seperti cara membuat crud dengan PHP native, hanya saja pada codeigniter kita menggunakan metode MVC sehingga membuat proses membuat crudnya agak sedikit berbeda, tetapi jangan khawatir, saya akan menjelaskan proses cara membuat crud dengan codeigniter nya secara jelas.

Seperti pada view v\_tampil.php yang sudah kita buat pada tutorial cara menampilkan data dari database untuk menampilkan data dalam bentuk table dan sudah berisi link edit dan hapus yang telah kita buat.

## Application/views/v\_tampil.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Membuat CRUD dengan CodeIgniter | MalasNgoding.com</title>
```

```

</head>
<body>
    <center><h1>Membuat CRUD dengan CodeIgniter | MalasNgoding.com</h1></center>
    <center><?php echo anchor('crud/tambah','Tambah Data'); ?></center>
    <table style="margin:20px auto;" border="1">
        <tr>
            <th>No</th>
            <th>Nama</th>
            <th>Alamat</th>
            <th>Pekerjaan</th>
            <th>Action</th>
        </tr>
        <?php
        $no = 1;
        foreach($user as $u) {
        ?>
        <tr>
            <td><?php echo $no++ ?></td>
            <td><?php echo $u->nama ?></td>
            <td><?php echo $u->alamat ?></td>
            <td><?php echo $u->pekerjaan ?></td>
            <td>
                <?php echo anchor('crud/edit/'.$u->id,'Edit'); ?>
                <?php echo anchor('crud/hapus/'.$u->id,'Hapus'); ?>
            </td>
        </tr>
        <?php } ?>
    </table>
</body>
</html>

```

bisa anda perhatikan di sini sudah terdapat link hapus yang kita buat dengan function hyperlink codeigniter yaitu anchor().

```
<?php echo anchor('crud/hapus/'.$u->id,'Hapus'); ?>
```

link ini tertuju pada method/function hapus di controller crud. dan berisi juga pengiriman data id data pada segment 3 nya. oleh sebab itu untuk membuat hapus datanya kita akan membuat method baru dengan nama hapus. kemudian kita akan tangkap data id nya dan memberika action untuk menghapus data pada database menurut data id melalui model codeigniter. buat sebuah method *hapus* pada controller crud.php

### **application/controllers/crud.php**

```

<?php
class Crud extends CI_Controller{
    function __construct(){
        parent::__construct();
        $this->load->model('m_data');
        $this->load->helper('url');
    }
}

```

```

function index() {
    $data['user'] = $this->m_data->tampil_data()->result();
    $this->load->view('v_tampil', $data);
}

function tambah() {
    $this->load->view('v_input');
}

function tambah_aksi() {
    $nama = $this->input->post('nama');
    $alamat = $this->input->post('alamat');
    $pekerjaan = $this->input->post('pekerjaan');

    $data = array(
        'nama' => $nama,
        'alamat' => $alamat,
        'pekerjaan' => $pekerjaan
    );
    $this->m_data->input_data($data, 'user');
    redirect('crud/index');
}

function hapus($id) {
    $where = array('id' => $id);
    $this->m_data->hapus_data($where, 'user');
    redirect('crud/index');
}

```

coba anda perhatikan di sini pada method hapus.

```

function hapus($id) {
    $where = array('id' => $id);
    $this->m_data->hapus_data($where, 'user');
    redirect('crud/index');
}

```

pada parameter function **hapus()** kita memberikan variabel \$id di sini yang berguna untuk menangkap data id yang di kirim melalui url dari link hapus tadi. kemudian saya jadikan array untuk kita kirimkan data array nya ke model m\_data . perhatikan juga pada

```
$this->m_data->hapus_data($where, 'user');
```

pada function hapus\_data di atas kita masukkan variabel array \$where yang berisi data id tadi, dan pada parameter kedua masukkan nama table. ini sesuai dengan logika anda, jika anda memiliki cara yang lebih mudah anda bisa menerapkannya agar tidak perlu menuliskan syntax program secara berulang-ulang.

karena di sini kita menggunakan function model hapus\_data pada model m\_data.php, maka kita buat kan lagi function hapus\_data pada model m\_data.php

## **application/models/m\_data.php**

```

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

```

sehingga

```

<?php
class M_data extends CI_Model{
    function tampil_data(){
        return $this->db->get('user');
    }

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

    function update_data(){
        return $this->db->get('user');
    }

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

```

pada function hapus\_data tersebut terdapat fungsi where yang berguna untuk menyeleksi query dan delete untuk menghapus record.

sekarang akan kita coba jalankan untuk melihat hasilnya.

<http://localhost/malasngoding/crud/>

No	Nama	Alamat	Pekerjaan	Action
1	Andi	Surabaya	web programmer	<a href="#">Edit</a> <a href="#">Hapus</a>
2	Budi	Bandung	Web Designer	<a href="#">Edit</a> <a href="#">Hapus</a>
3	Rudi	Semarang	Web Master	<a href="#">Edit</a> <a href="#">Hapus</a>

dan sekarang kita akan mencoba menghapus data record ketiga yaitu data user yang bernama *rudi* dengan nama klik pada link/tombol hapus. dan hasilnya data rudi pun akan terhapus

## Membuat CRUD dengan CodeIgniter | MalasNgoding.com

[Tambah Data](#)

No	Nama	Alamat	Pekerjaan	Action
1	Andi	Surabaya	web programmer	<a href="#">Edit</a> <a href="#">Hapus</a>
2	Budi	Bandung	Web Designer	<a href="#">Edit</a> <a href="#">Hapus</a>

dan data pun berhasil di hapus.

sekian Cara Membuat CRUD Dengan CodeIgniter Hapus Data. selanjutnya kita akan membahas tentang cara mengedit atau mengupdate data pada database.

## Mengupdate Data Pada Database

Pada bagian ini saya akan menjelaskan sambungan dari tutorial-tutorial sebelumnya tentang membuat crud dengan codeigniter. di mana pada tutorial sebelumnya tentang tutorial crud codeigniter kita telah belajar cara menampilkan data dari database dengan codeigniter, menginput data ke database dengan codeigniter, menghapus data dengan codeigniter dan pada kelanjutannya ini saya akan menjelaskan tentang cara mengupdate atau mengedit data pada database dengan codeigniter.

silahkan simak contoh dan penjelasan berikut ini.

untuk membuat edit data dan update data dengan codeigniter, kita perlu membuat halaman edit data dan beberapa method pada controller crud. seperti yang bisa anda lihat pada table data yang sudah saya jelaskan pada tutorial *menampilkan data dari database*. sudah terdapat link/tombol edit data yang mengarah ke method edit dan mengirimkan id pada segment 3 melalui url. jadi yang harus kita lakukan adalah membuat method edit dan model untuk untuk mengambil data sesuai dengan id yang ingin di edit menurut id.

buat sebuah method *edit* pada controller crud

### application/controllers/crud.php

```
function edit($id){  
    $where = array('id' => $id);  
    $data['user'] = $this->m_data->edit_data($where, 'user')->result();  
    $this->load->view('v_edit', $data);  
}
```

Sehingga controller crud menjadi seperti berikut.

```
<?php

class Crud extends CI_Controller{

    function __construct(){
        parent::__construct();
        $this->load->model('m_data');
        $this->load->helper('url');

    }

    function index(){
        $data['user'] = $this->m_data->tampil_data()->result();
        $this->load->view('v_tampil',$data);
    }

    function tambah(){
        $this->load->view('v_input');
    }

    function tambah_aksi(){
        $nama = $this->input->post('nama');
        $alamat = $this->input->post('alamat');
        $pekerjaan = $this->input->post('pekerjaan');

        $data = array(
            'nama' => $nama,
            'alamat' => $alamat,
            'pekerjaan' => $pekerjaan
        );
        $this->m_data->input_data($data,'user');
        redirect('crud/index');
    }

    function hapus($id){
        $where = array('id' => $id);
        $this->m_data->hapus_data($where,'user');
        redirect('crud/index');
    }

    function edit($id){
        $where = array('id' => $id);
        $data['user'] = $this->m_data->edit_data($where,'user')->result();
        $this->load->view('v_edit',$data);
    }

}
```

Perhatikan pada method edit di atas, kita menjadikan id menjadi array yang kemudian kita gunakan untuk mengambil data menurut id dengan menggunakan function edit\_data() pada model m\_data. oh ya fungsi result() di sini berguna untuk meng-generate hasil query menjadi array. dan kemudian kita tampilkan pada view v\_edit yang akan kita buat view nya sebentar lagi

buat function edit\_data pada model m\_data

## **application/model/m\_data.php**

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

dan buat sebuah view dengan nama v\_edit.php di mana view ini kita jadikan sebagai form yang menampilkan data yang akan di edit.

## **application/views/v\_edit.php**

```
<!DOCTYPE html>  
<html>  
<head>  
    <title>Membuat CRUD dengan CodeIgniter | MalasNgoding.com</title>  
</head>  
<body>  
    <center>  
        <h1>Membuat CRUD dengan CodeIgniter | MalasNgoding.com</h1>  
        <h3>Edit Data</h3>  
    </center>  
    <?php foreach($user as $u){ ?>  
    <form action="<?php echo base_url().'crud/update'; ?>" method="post">  
        <table style="margin:20px auto;">  
            <tr>  
                <td>Nama</td>  
                <td>  
                    <input type="hidden" name="id" value="<?php echo  
$u->id ?>">  
                    <input type="text" name="nama" value="<?php echo  
$u->nama ?>">  
                </td>  
            </tr>  
            <tr>  
                <td>Alamat</td>  
                <td><input type="text" name="alamat" value="<?php echo  
$u->alamat ?>"></td>  
            </tr>  
            <tr>  
                <td>Pekerjaan</td>  
                <td><input type="text" name="pekerjaan" value="<?php  
echo $u->pekerjaan ?>"></td>  
            </tr>  
            <tr>  
                <td></td>  
                <td><input type="submit" value="Simpan"></td>  
            </tr>  
        </table>  
    </form>  
    <?php } ?>  
</body>  
</html>
```

nah sampai di sini kita telah berhasil membuat edit data dalam bentuk form. tapi data yang di edit belum dapat di update karena kita belum membuat aksi untuk mengupdate data. untuk membuat aksi yang menghandle update data buat sebuah method lagi dengan nama update pada controller crud sesuai dengan action form edit yang kita arahkan pada method update.

```
<form action="<?php echo base_url() . 'crud/update'; ?>" method="post">
```

## application/controllers/crud.php

```
function update(){
    $id = $this->input->post('id');
    $nama = $this->input->post('nama');
    $alamat = $this->input->post('alamat');
    $pekerjaan = $this->input->post('pekerjaan');

    $data = array(
        'nama' => $nama,
        'alamat' => $alamat,
        'pekerjaan' => $pekerjaan
    );

    $where = array(
        'id' => $id
    );

    $this->m_data->update_data($where,$data,'user');
    redirect('crud/index');
}
```

di sini kita tangkap dulu data dari form edit.

```
$id = $this->input->post('id');
$nama = $this->input->post('nama');
$alamat = $this->input->post('alamat');
$pekerjaan = $this->input->post('pekerjaan');
```

kemudian masukkan data yang akan di update ke dalam variabel data

```
$data = array(
    'nama' => $nama,
    'alamat' => $alamat,
    'pekerjaan' => $pekerjaan
);
```

dan variabel where yang menjadi penentu data yang di update (id yang mana)

```
$where = array(
    'id' => $id
);
```

kemudian untuk menghandle update data pada database kita gunakan function update\_data() pada model m\_data yang akan kita buat sebentar lagi.

```
$this->m_data->update_data($where,$data,'user');
```

Selanjutnya untuk yg terakhir kita buat sebuah function lagi pada model dengan nama update\_data.

### **application/model/m\_data.php**

```
<?php

class M_data extends CI_Model{
    function tampil_data(){
        return $this->db->get('user');
    }

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

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

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

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

dan selesai. waktunya menjalankan pada browser untuk melihat hasil edit data pada database dengan codeigniter.

<http://localhost/malasngoding/crud>

dan klik link edit menurut data yang ingin di edit.

localhost/malasngoding/crud/edit/2

**Membuat CRUD dengan CodeIgniter | MalasNgoding.com**

Tambah data baru

Nama	Budi
Alamat	Bandung
Pekerjaan	Web Designer

Simpan

ubah data sesuai keinginan dan klik simpan

## Membuat CRUD dengan CodeIgniter | MalasNgoding.com

[Tambah Data](#)

No	Nama	Alamat	Pekerjaan	Action
1	Andi	Surabaya	web programmer	<a href="#">Edit</a> <a href="#">Hapus</a>
2	Santoso	Jakarta	Web Designer	<a href="#">Edit</a> <a href="#">Hapus</a>

dan data pun berhasil di update.

## Dasar Bagian 14 : Membuat Pagination Dengan CodeIgniter

Codeigniter telah menyediakan fasilitas untuk membuat pagination dengan sangat mudah. jadi dengan menggunakan codeigniter kita tidak perlu lagi capek-capek membuat pagination, kita hanya perlu membuka library pagination codeigniter dan melakukan beberapa settingan dan pagination pun siap. oke langsung saja ya kita masuk ke tutorial cara membuat pagination dengan codeigniter.

Tahap pertama yang harus di lakukan adalah setting base\_url dulu di *application/config/config.php*. sesuaikan dengan nama project ci kamu sendiri.

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

karena data yang kita gunakan atau yang akan kita tampilkan adalah data dari database oleh karena itu kita setting dulu database nya. untuk cara menghubungkan codeigniter dengan database mysql teman-teman bisa membaca tutorial sebelumnya tentang cara menghubungkan codeigniter dan database mysql dengan klik di sini.

### application/config/database.php

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

Silahkan setting username, password dan nama database anda. pada tutorial ini saya menggunakan database malasngoding.

dan berikut ini saya sertakan juga file sql malasngoding untuk contoh cara membuat pagination dengan codeigniter. silahkan anda import

## malasngoding.sql

```
-- phpMyAdmin SQL Dump
-- version 3.5.2.2
-- http://www.phpmyadmin.net
--
-- Host: 127.0.0.1
-- Generation Time: Jan 22, 2016 at 02:33 PM
-- Server version: 5.5.27
-- PHP Version: 5.4.7

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

-- Database: `malasngoding`
--


-- -----
-- Table structure for table `user`


CREATE TABLE IF NOT EXISTS `user` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nama` varchar(50) NOT NULL,
  `alamat` text NOT NULL,
  `pekerjaan` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=63 ;

-- Dumping data for table `user`


INSERT INTO `user` (`id`, `nama`, `alamat`, `pekerjaan`) VALUES
(1, 'Andi', 'Surabaya', 'web programmer'),
(2, 'Santoso', 'Jakarta', 'Web Designer'),
(6, 'Samsul', 'Sumedang', 'Pegawai'),
(7, 'Bob', 'jakarta', 'penyanyi'),
(8, 'marley', 'afrika', 'penyanyi'),
(9, 'Bob', 'jakarta', 'penyanyi'),
(10, 'Bob', 'jakarta', 'penyanyi'),
(11, 'Bob', 'jakarta', 'penyanyi'),
(12, 'Bob', 'jakarta', 'penyanyi'),
(13, 'Bob', 'jakarta', 'penyanyi'),
(14, 'Bob', 'jakarta', 'penyanyi'),
(15, 'Bob', 'jakarta', 'penyanyi'),
(16, 'Bob', 'jakarta', 'penyanyi'),
(17, 'Bob', 'jakarta', 'penyanyi'),
(18, 'marley', 'afrika', 'penyanyi'),
(19, 'Bob', 'jakarta', 'penyanyi'),
(20, 'Bob', 'jakarta', 'penyanyi'),
(21, 'Bob', 'jakarta', 'penyanyi'),
(22, 'Bob', 'jakarta', 'penyanyi'),
(23, 'Bob', 'jakarta', 'penyanyi'),
(24, 'Bob', 'jakarta', 'penyanyi'),
```

```

(25, 'Bob', 'jakarta', 'penyanyi'),
(26, 'Bob', 'jakarta', 'penyanyi'),
(27, 'Bob', 'jakarta', 'penyanyi'),
(28, 'Bob', 'jakarta', 'penyanyi'),
(29, 'Bob', 'jakarta', 'penyanyi'),
(30, 'Bob', 'jakarta', 'penyanyi'),
(31, 'Bob', 'jakarta', 'penyanyi'),
(32, 'marley', 'afrika', 'penyanyi'),
(33, 'Bob', 'jakarta', 'penyanyi'),
(34, 'Bob', 'jakarta', 'penyanyi'),
(35, 'Bob', 'jakarta', 'penyanyi'),
(36, 'Bob', 'jakarta', 'penyanyi'),
(37, 'Bob', 'jakarta', 'penyanyi'),
(38, 'Bob', 'jakarta', 'penyanyi'),
(39, 'Bob', 'jakarta', 'penyanyi'),
(40, 'Bob', 'jakarta', 'penyanyi'),
(41, 'Bob', 'jakarta', 'penyanyi'),
(42, 'Bob', 'jakarta', 'penyanyi'),
(43, 'Bob', 'jakarta', 'penyanyi'),
(44, 'Bob', 'jakarta', 'penyanyi'),
(45, 'Bob', 'jakarta', 'penyanyi'),
(46, 'Bob', 'jakarta', 'penyanyi'),
(47, 'marley', 'afrika', 'penyanyi'),
(48, 'Bob', 'jakarta', 'penyanyi'),
(49, 'Bob', 'jakarta', 'penyanyi'),
(50, 'Bob', 'jakarta', 'penyanyi'),
(51, 'Bob', 'jakarta', 'penyanyi'),
(52, 'Bob', 'jakarta', 'penyanyi'),
(53, 'Bob', 'jakarta', 'penyanyi'),
(54, 'Bob', 'jakarta', 'penyanyi'),
(55, 'Bob', 'jakarta', 'penyanyi'),
(56, 'Bob', 'jakarta', 'penyanyi'),
(57, 'Bob', 'jakarta', 'penyanyi'),
(58, 'Bob', 'jakarta', 'penyanyi'),
(59, 'Bob', 'jakarta', 'penyanyi'),
(60, 'Bob', 'jakarta', 'penyanyi'),
(61, 'Bob', 'jakarta', 'penyanyi'),
(62, 'marley', 'afrika', 'penyanyi');

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

nah pada database malasngoding, saya sudah menyertakan table user yang memiliki beberapa record sebagai contoh membuat pagination dengan codeigniter ini.

### **application/controllers/welcome.php**

```

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

class Welcome extends CI_Controller {

    function __construct(){
        parent::__construct();
        $this->load->helper(array('url'));
        $this->load->model('m_data');
    }

    public function index(){
        $this->load->database();
        $jumlah_data = $this->m_data->jumlah_data();
```

```

        $this->load->library('pagination');
        $config['base_url'] = base_url().'index.php/welcome/index/';
        $config['total_rows'] = $jumlah_data;
        $config['per_page'] = 10;
        $from = $this->uri->segment(3);
        $this->pagination->initialize($config);
        $data['user'] = $this->m_data->data($config['per_page'],$from);
        $this->load->view('v_data',$data);
    }
}

```

Pada method index ini lah kita akan membuat pagination codeigniter. dengan membuka library pagination codeigniter. dan menentukan pengaturan pagination codeigniter

```

$config['base_url'] = base_url().'index.php/welcome/index/';
$config['total_rows'] = $jumlah_data;
$config['per_page'] = 10;

```

Pada pengaturan per\_page maksud nya jumlah record yang di tampilkan per halaman. total\_rows untuk pengaturan jumlah dari seluruh record. dan base\_url adalah settingan url dari link pagination. variabel \$jumlah\_data berisi data jumlah data(perhatikan method jumlah pada model m\_data berikut)

### **application/models/m\_data.php**

```

<?php

class M_data extends CI_Model{
    function data($number,$offset){
        return $query = $this->db->get('user',$number,$offset)->result();
    }

    function jumlah_data(){
        return $this->db->get('user')->num_rows();
    }
}

```

pada model m\_data ini terdapat dua buah function yang saya buat, function data yang berguna untuk mengambil data user dari record ke berapa sampai record ke berapa sesuai data yang di kirim melalui parameter pada function. dan function jumlah\_data bertujuan untuk mengambil jumlah seluruh record.

### **application/views/v\_data.php**

```

<!DOCTYPE html>
<html>
<head>
    <title>Membuat Pagination Pada CodeIgniter | MalasNgoding.com</title>
</head>
<body>
<h1>Membuat Pagination Pada CodeIgniter | MalasNgoding.com</h1>
<table border="1">
    <tr>

```

```

<th>no</th>
<th>nama</th>
<th>alamat</th>
<th>pekerjaan</th>
</tr>
<?php
$no = $this->uri->segment('3') + 1;
foreach($user as $u) {
?>
<tr>
    <td><?php echo $no++; ?></td>
    <td><?php echo $u->nama ?></td>
    <td><?php echo $u->alamat ?></td>
    <td><?php echo $u->pekerjaan ?></td>
</tr>
<?php } ?>
</table>
<br/>
<?php
echo $this->pagination->create_links();
?>
</body>
</html>

```

dan akan kita lihat hasil nya

[http://localhost/malasngoding\\_paging/welcome](http://localhost/malasngoding_paging/welcome)



## Membuat Pagination Pada CodeIgniter | MalasNgoding.com

no	nama	alamat	pekerjaan
1	Andi	Surabaya	web programmer
2	Santoso	Jakarta	Web Designer
3	Samsul	Sumedang	Pegawai
4	Bob	jakarta	penyanyi
5	marley	afrika	penyanyi
6	Bob	jakarta	penyanyi
7	Bob	jakarta	penyanyi
8	Bob	jakarta	penyanyi
9	Bob	jakarta	penyanyi
10	Bob	jakarta	penyanyi

[1](#) [2](#) [3](#) [>](#) [Last](#) [>](#)

sekian tutorial cara membuat pagination pada codeigniter.

## Dasar Bagian 15 : Membuat Login Dengan Codeigniter

Pada tutorial codeigniter ini kita akan belajar cara membuat sistem login dengan codeigniter. kita akan membuat login yang sangat sederhana dulu. karena untuk pembelajaran agar mudah di pahami oleh teman-teman yang baru masuk ke pemrograman berorientasi object. khusus nya yang baru mempelajari framework codeigniter.

### Persiapan Membuat Login Dengan Codeigniter

Seperti halnya membuat login di php dasar yang telah saya bahas di [www.malasngoding.com](http://www.malasngoding.com), yang pertama kali yang harus kita lakukan adalah menyediakan table admin dulu. jadi username dan password nya di simpan di table ini. untuk mengikuti tutorial ini teman-teman buatkan dulu database dengan nama "malasngoding". kemudian buat table dengan nama admin. yang berisi 3 column. id, username, dan password. atau lebih cepatnya silahan teman-teman import saja sql berikut ini. jangan lupa buat database dengan nama "malasngoding" dulu. Dan import ke database 'malasngoding'.

#### **malasngoding.sql**

```
-- phpMyAdmin SQL Dump
-- version 3.5.2.2
-- http://www.phpmyadmin.net
--
-- Host: 127.0.0.1
-- Generation Time: Feb 26, 2016 at 09:42 AM
-- Server version: 5.5.27
-- PHP Version: 5.4.7

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Database: `malasngoding`
--

-----
-- Table structure for table `admin`
--

CREATE TABLE IF NOT EXISTS `admin` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
```

```

`username` varchar(255) NOT NULL,
`password` varchar(255) NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

-- 
-- Dumping data for table `admin`
-- 

INSERT INTO `admin` (`id`, `username`, `password`) VALUES
(1, 'malasngoding', '10406c1d7b7421b1a56f0d951e952a95');

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

	<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	<a href="#">Change</a>	<a href="#">Delete</a>	<a href="#">Export</a>
	<input checked="" type="checkbox"/>						

kita menggunakan md5 di sini untuk passwordnya. agar lebih aman. password aslinya 'malasngoding123'.

Username : malasngoding  
password : malasngoding123

Langkah selanjutnya adalah :

- Aktifkan library database
- Aktifkan library session
- Set encryption key session codeigniter
- Aktifkan helper url

Aktifkan library 'database' dan 'session' di file autoload codeigniter. dan aktifkan juga helper 'url'. kenapa kita harus mengaktifkan library database? karena kita akan menggunakan fungs-fungsi database yang telah di sediakan oleh codeigniter. dan kenapa kita harus mengaktifkan library session? karena kita akan memerlukan session di login. agar bisa mendeteksi apakah si admin atau user sudah login atau belum. kalau belum login maka tidak kita perbolehkan untuk mengakses halaman

admin. berarti nanti kita buat admin harus login dulu baru bisa mengakses halaman admin. jika tidak, admin akan di alihkan ke halaman login.

### **application/config/autoload.php**

isikan database dan session di autoload library.

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

aktifkan juga helper url.

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

Seperti point di atas tadi. karena kita menggunakan session di codeigniter. maka kita harus mengisikan dulu encryption key session nya. ada di application/config/config.php

```
$config['encryption_key'] = 'tutoriallogincodeigniterwwwmalasngodingcom';
```

Isikan terserah teman-teman. di sini saya mengisi nya dengan 'tutoriallogincodeigniterwwwmalasngodingcom'.

Nah sampai di sini saya rasa persiapan dasar nya sudah beres. database dan data login si admin sudah ada, helper dan library yang kita perlukan juga sudah kita aktifkan, dan encryption key session codeigniter juga sudah kita set.

## **Menghubungkan Codeigniter dengan MySQL**

Jangan lupa untuk menghubungkan codeigniter dengan database. baca tutorial sebelumnya tentang tutorial menghubungkan codeigniter dengan database mysql. tidak saya jelaskan lagi. teman-teman bisa melihat tutorial menghubungkan codeigniter dengan database mysql di tutorial sebelumnya yang sudah saya bahas.

### **Application/config/config.php**

```
'hostname' => 'localhost',
'username' => 'root',
'password' => '',
'database' => 'malasngoding',
```

## Membuat Form Login Dengan Codeigniter

Langkah selanjutnya buat sebuah controller. dengan nama login.php. di controller ini yang akan kita tugaskan untuk menampilkan form login dan melakukan verifikasi/authentikasi username dan password admin yang di masukkan. serta fungsi logout.

### application/controllers/login.php

```
<?php

class Login extends CI_Controller{

    function __construct(){
        parent::__construct();
        $this->load->model('m_login');

    }

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

    function aksi_login(){
        $username = $this->input->post('username');
        $password = $this->input->post('password');
        $where = array(
            'username' => $username,
            'password' => md5($password)
        );
        $cek = $this->m_login->cek_login("admin",$where)->num_rows();
        if($cek > 0){

            $data_session = array(
                'nama' => $username,
                'status' => "login"
            );

            $this->session->set_userdata($data_session);
            redirect(base_url("admin"));

        }else{
            echo "Username dan password salah !";
        }
    }

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

seperti yang telah kita ketahui. fungsi construct adalah fungsi yang pertama kali di jalankan pada saat class di jalankan. jadi di fungsi construct ini kita aktifkan model m\_login. mana model m\_login nya ? tenang.. sebentar lagi kita buat.

di fungsi index kita men-load atau membuat view yang bernama v\_login. di file view v\_login ini kita buat form loginnya. buat view baru dengan nama v\_login.php.

### **application/views/v\_login.php**

```
<!DOCTYPE html>
<html>
<head>
    <title>Membuat Login Dengan CodeIgniter | www.malasngoding.com</title>
</head>
<body>
    <h1>Membuat Login Dengan CodeIgniter <br/> www.malasngoding.com</h1>
    <form action="<?php echo base_url('login/aksi_login'); ?>\" method="post">

        <table>
            <tr>
                <td>Username</td>
                <td><input type="text" name="username"></td>
            </tr>
            <tr>
                <td>Password</td>
                <td><input type="password" name="password"></td>
            </tr>
            <tr>
                <td></td>
                <td><input type="submit" value="Login"></td>
            </tr>
        </table>
    </form>
</body>
</html>
```



## **Membuat Login Dengan CodeIgniter www.malasngoding.com**

Username

Password

pada form login di atas kita mengarahkan untuk aksi login nya ke fungsi aksi\_login di controller login. perhatikan lagi fungsi aksi\_login di controller login yang sudah kita buat tadi.

```
function aksi_login(){
    $username = $this->input->post('username');
    $password = $this->input->post('password');
    $where = array(
        'username' => $username,
        'password' => md5($password)
    );
    $cek = $this->m_login->cek_login("admin",$where)->num_rows();
    if($cek > 0){

        $data_session = array(
            'nama' => $username,
            'status' => "login"
        );

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

        redirect(base_url("admin"));
    }else{
        echo "Username dan password salah !";
    }
}
```

kita menangkap data username dan password yang di kirim lalu kita masukkan ke dalam array. agar kita kirimkan lagi ke model m\_login.

```
$username = $this->input->post('username');
$password = $this->input->post('password');
$where = array(
    'username' => $username,
    'password' => md5($password)
);
```

dan kita cek ketersediaan username dan password nya di model m\_login. fungsi num\_rows() berguna untuk menghitung jumlah record.

```
$cek = $this->m_login->cek_login("admin",$where)->num_rows();
```

buat sebuah model dengan nama m\_login.php

### **application/models/m\_login.php**

```
<?php

class M_login extends CI_Model{
    function cek_login($table,$where){
        return $this->db->get_where($table,$where);
    }
}
```

dan yang terakhir kita cek.

```

if($cek > 0){
    $data_session = array(
        'nama' => $username,
        'status' => "login"
    );

    $this->session->set_userdata($data_session);
    redirect(base_url("admin"));
} else{
    echo "Username dan password salah !";
}

```

jika username dan password di temukan atau benar, maka kita buat session nama berisi username yang di isi tadi. dan session status berisi 'login'. kemudian kita alihkan ke controller admin. jika username dan password tidak di temukan atau salah, maka halaman akan di tampilkan pesan "username dan password salah!".

buat sebuah controller baru lagi dengan nama admin.php.

### **application/controllers/admin.php**

```

<?php

class Admin extends CI_Controller{

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

        if($this->session->userdata('status') != "login"){
            redirect(base_url("login"));
        }
    }

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

```

di construct nya kita cek apakah session status. ini untuk mendeteksi apakah user atau admin sudah login atau belum. berarti jika admin atau user sudah berhasil login kan otomatis admin atau user tersebut sudah memiliki session login. (seperti yang saya jelaskan pada pembuatan session di atas tadi). jika tidak maka halaman akan di alihkan ke halaman login lagi. jadi apabila admin atau user belum berhasil login maka admin atau user tidak kita perbolehkan mengakses halaman admin.

untuk membuat halaman adminnya perhatikan pada fungsi index di controller admin.php, kita buat view untuk halaman adminnya. buat view baru lagi dengan nama v\_admin.php.

### **application/views/v\_admin.php**

```

<!DOCTYPE html>
<html>
<head>

```

```
<title>Membuat login dengan codeigniter | www.malasngoding.com</title>
</head>
<body>
    <h1>Login berhasil !</h1>
    <h2>Hai, <?php echo $this->session->userdata("nama"); ?></h2>
    <a href="php echo base_url('login/logout'); ?&gt;"&gt;Logout&lt;/a&gt;
&lt;/body&gt;
&lt;/html&gt;</pre
```

seperti yang teman-teman lihat juga di halaman admin ini. kita membuat link/tombol logout. jadi jika di klik akan di alihkan pada fungsi logout pada controller login.

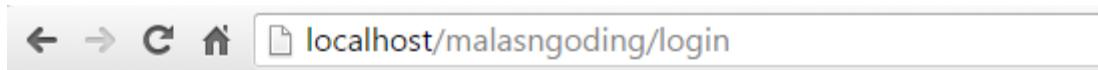
```
function logout(){
    $this->session->sess_destroy();
    redirect(base_url('login'));
}
```

Fungsi berikut ini

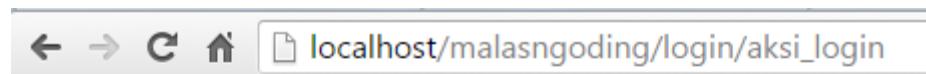
```
$this->session->sess_destroy();
```

berguna untuk menghapus semua session. dan login dengan codeigniter kita pun selesai. sekarang kita akan coba tes dulu.

kita coba dulu untuk memasukkan username dan password yang salah.



maka akan di tampilkan pesan username dan password salah.



Username dan password salah !

dan coba kita masukkan username dan password yang benar.



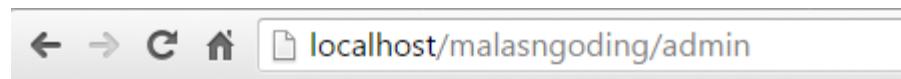
## Membuat Login Dengan CodeIgniter

### www.malasngoding.com

Username

Password

dan klik login.



Dan login pun berhasil kita buat.

# **Tutorial Codeigniter Lanjutan**

**Studi Kasus Membuat Sistem Informasi  
Perpustakaan**

Di tahap ini kita akan memulai membuat sebuah *project* aplikasi berbasis web yang akan kita buat dengan menggunakan Codeigniter. Akan dijelaskan secara berurutan secara bertahap. Teman-teman bisa mengikuti urutan yang harus dilakukan. Yang dimulai dari instalasi codeigniter 3 untuk memulai pembuatan sistem informasi perpustakaan berbasis web.

Sebenarnya aplikasi ini sudah dulu saya buat. Baru kemudian saya tulis tutorialnya di buku ini. Karena jika proses penulisan ebook ini dibuat berbarengan dengan pembuatan sistem informasi perpustakaan ini, akan sedikit ribet dan menyita banyak waktu. Oleh karena itu saya mempersiapkan terlebih dulu sistem informasinya, baru kemudian saya menulis panduan pembuatannya di dalam buku ini. Jadi jika ada urutan yang tidak sesuai saat teman-teman mengikuti tutorial ini, teman-teman bisa menanyakan langsung ke saya melalui kontak yang tertera di [www.malasngoding.com](http://www.malasngoding.com).

Pada bagian **tutorial codeigniter lanjutan** ini tidak lagi dijelaskan secara terperinci. Karena semua pembahasan tentang codeigniter dasar telah dijelaskan pada pembahasan sebelumnya atau di [www.malasngoding.com](http://www.malasngoding.com) maupun di bagian **tutorial codeigniter dasar** pada bagian atas. Pada ebook ini akan dijelaskan bagaimana cara membangun sistem informasi dengan menggunakan codeigniter secara bertahap seperti pada struktur judul dan halaman pada buku ini.

Dengan mengucap bismillah langsung saja kita mulai dari tahap instalasi codeigniter ke XAMPP.

Adapun beberapa aplikasi pendukung yang digunakan adalah sebagai berikut :

1. Text Editor Sublime
2. XAMPP 5.6.36
3. Web Browser Google Chrome / Mozilla Firefox

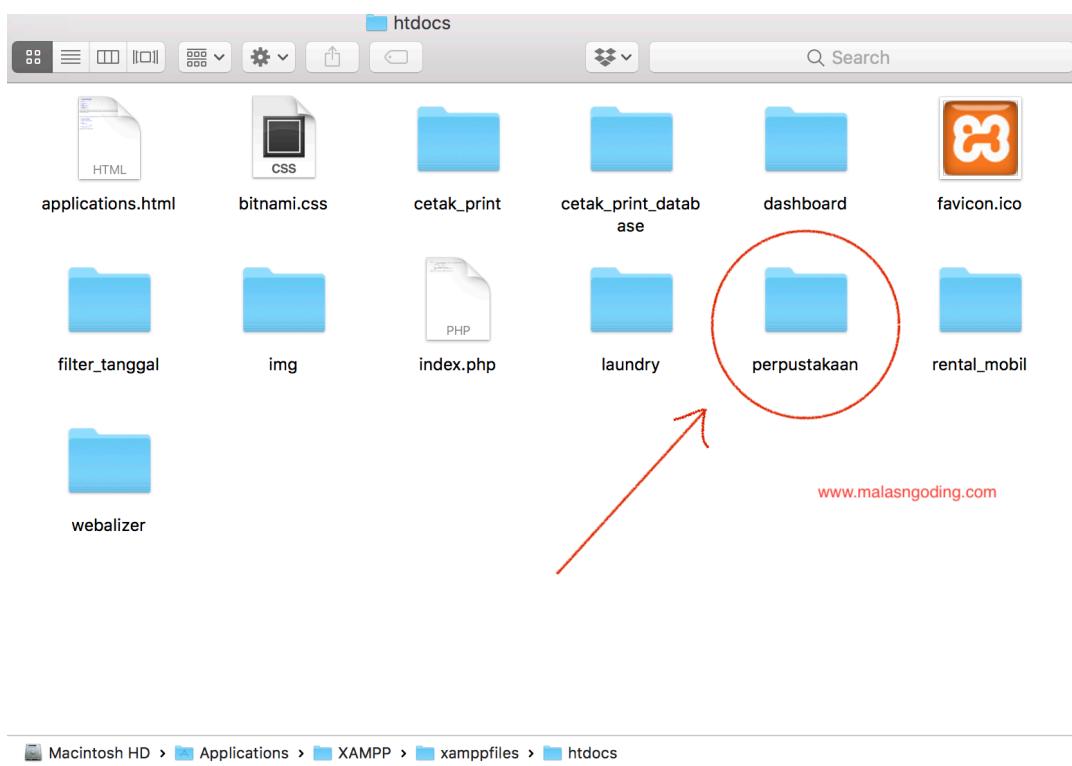
Semua aplikasi pendukung yang disebutkan di atas telah sertakan di dalam paket ebook ini (di dalam folder aplikasi pendukung).

## Instalasi Codeigniter 3

Tahap ini adalah tahap paling awal yang kita lakukan. Yaitu melakukan instalasi Codeigniter, dimana Codeigniter yang digunakan adalah codeigniter versi 3. Jika teman-teman ingin bereksperimen, boleh menggunakan codeigniter versi berapa saja. Hanya saja lebih baik menggunakan codeigniter 3 karena codeigniter 3 adalah versi terbaru pada saat ini. Dan versi codeigniter yang saya gunakan dalam pembuatan ebook ini.

Teman-teman bisa mendownload codeigniter langsung melalui situs resminya pada link berikut (<https://codeigniter.com/>). Atau teman-teman bisa menggunakan codeigniter 3 yang sudah saya sertakan di dalam paket ebook tutorial codeigniter lengkap dengan studi kasus membuat sistem informasi perpustakaan ini (dalam folder library).

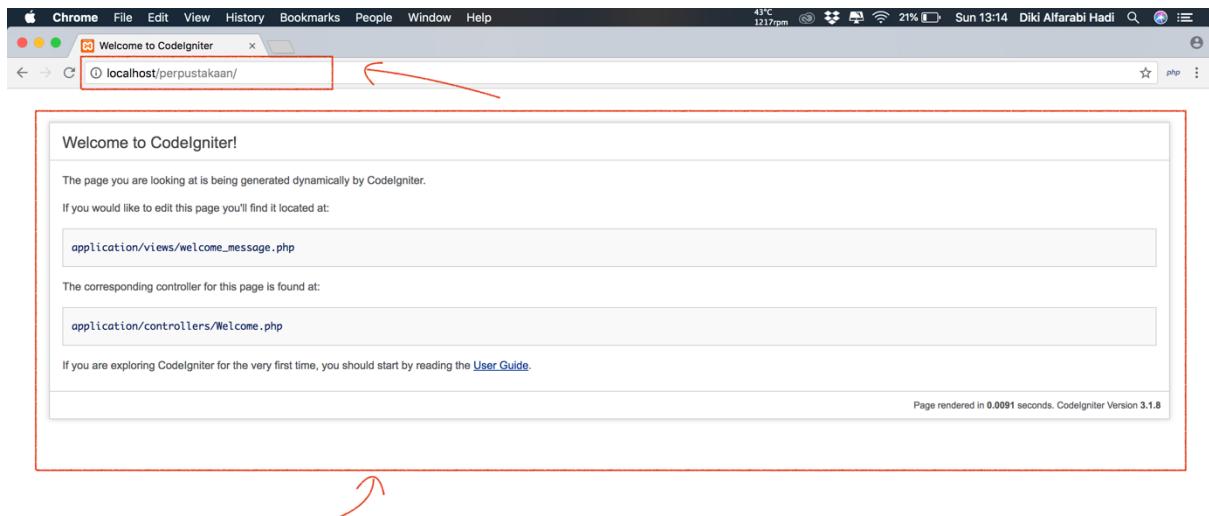
Setelah diekstrak, kemudian *copy* folder codeigniternya ke direktori localhost teman-teman (htdocs). Caranya juga sudah dijelaskan pada pembahasan tutorial codeigniter bagian dasar sebelumnya. Jika teman-teman menggunakan sistem operasi windows, direktori htdocs nya ada di **C:\XAMPP\htdocs**. Jika menggunakan macOS letak folder htdocs nya ada di **/application/XAMPP/htdocs/**. Setelah di paste, ubah nama folder nya menjadi **perpustakaan**. Seperti pada gambar berikut.



Kemudian aktifkan dulu XAMPP (apache dan mysql) nya. Baru kemudian buka web browser dan jalankan alamat berikut.

`http://localhost/perpustakaan`

maka akan muncul halaman default codeigniter. Dan menandakan bahwa instalasi codeigniter telah berhasil.



Sampai di sini maka instalasi Codeigniter pun selesai.

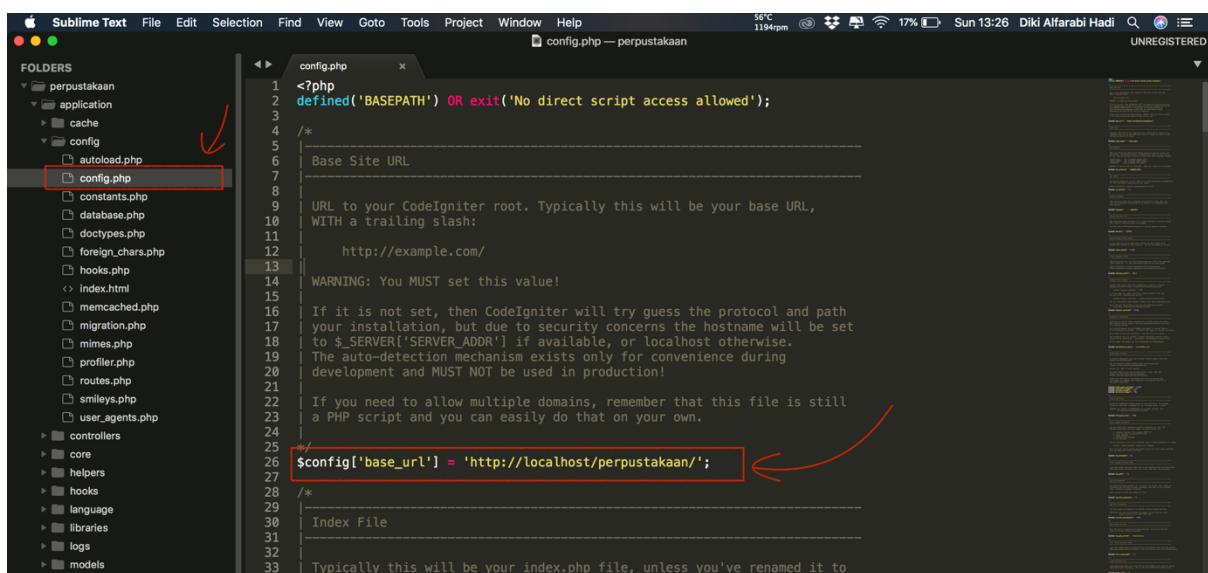
# Persiapan Membuat Sistem Informasi Perpustakaan

Diperlukan beberapa penyesuaian atau pengaturan pada codeigniter sebelum mulai membuat sistem informasi perpustakaan. Semua pengaturan atau penyesuaian yang kita lakukan di sini sesuai dengan kebutuhan kita untuk membuat sistem informasi perpustakaan dengan codeigniter. Di antaranya adalah :

## Setting Base URL

Pengaturan base url adalah pengaturan URL dasar dalam sebuah project aplikasi yang dibuat dengan menggunakan codeigniter. Pengaturan base url di codeigniter terdapat di dalam file config.php di dalam folder project codeigniter anda.

### perpustakaan/application/config/config.php



```
Sublime Text File Edit Selection Find View Goto Tools Project Window Help config.php — perpustakaan UNREGISTERED
FOLDERS
perpustakaan
└── application
    ├── cache
    └── config
        ├── autoload.php
        └── config.php
            ├── constants.php
            ├── database.php
            ├── doctypes.php
            ├── foreign_chars.php
            ├── hooks.php
            ├── index.html
            ├── memcached.php
            ├── migration.php
            ├── mimes.php
            ├── profiler.php
            ├── routes.php
            ├── smileys.php
            └── user_agents.php
        ├── controllers
        ├── core
        ├── helpers
        ├── hooks
        ├── language
        ├── libraries
        ├── logs
        └── models
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 
4 /*
5  * Base Site URL
6  */
7 
8 URL to your CodeIgniter root. Typically this will be your base URL,
9 WITH a trailing slash:
10 
11 http://example.com/
12 
13 WARNING: You MUST set this value!
14 
15 If it is not set, then CodeIgniter will try guess the protocol and path
16 your installation, but due to security concerns the hostname will be set
17 to $_SERVER['SERVER_ADDR'] if available, or localhost otherwise.
18 The auto-detection mechanism exists only for convenience during
19 development and MUST NOT be used in production!
20 
21 If you need to allow multiple domains, remember that this file is still
22 a PHP script and you can easily do that on your own.
23 
24 /*
25 */
26 $config['base_url'] = 'http://localhost/perpustakaan/';
27 
28 /*
29 Index File
30 
31 Typically this will be your index.php file, unless you've renamed it to
32 
33 
```

Sesuaikan dengan nama directory project kita. Dimana tadi folder project kita telah kita beri nama "perpustakaan". Maka sesuaikan pengaturan base\_url nya menjadi.

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

### INGAT !

Base url adalah fitur dari codeigniter untuk menetapkan url dasar. Dan base\_url akan sering kita gunakan dalam pengembangan aplikasi menggunakan codeigniter.

## Load Library, Helper, dan segala yang dibutuhkan

Seperti yang sudah dijelaskan pada tutorial codeigniter dasar dalam ebook ini maupun di [www.malasngoding.com](http://www.malasngoding.com). Bahwa codeigniter menyediakan banyak sekali library dan helper untuk membantu para developer/programmer dalam mengerjakan project. Yaitu dengan adanya helper dan library dari codeigniter.

Helper dan Library bisa dibilang seperti kita memiliki perpusataan coding dengan fungsi-fungsi yang berbeda dan memiliki kegunaan masing-masing. Kita tinggal memanggil dan menggunakannya saja. Kita juga bisa membuat helper atau library kita sendiri di codeigniter untuk memudahkan pekerjaan kita.

Di tahap ini kita akan men-load atau membuka beberapa library dan helper codeigniter yang kita butuhkan dalam pembuatan aplikasi rental mobil ini. Untuk menjalankan atau membuka library dan helper secara otomatis di codeigniter, kita bisa mengaturnya dalam file *autoload.php* dalam folder *application/config/autoload.php*.

### **perpustakaan/application/config/autoload.php**

Library yang akan kita load secara otomatis adalah *database*, *session* dan *form\_validation* .

Cari syntax

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

Kemudian tambahkan *database*, *session* dan *form\_validation*. Sehingga menjadi seperti berikut.

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

Dan helper yang akan kita load secara otomatis adalah *url* :

cari syntax

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

Kemudian tambahkan “*url*” (*karena kita hanya membutuhkan helper url disini*). Maka akan menjadi seperti berikut.

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

Seperti pada gambar berikut.

```
Sublime Text File Edit Selection Find Goto Tools Project Window Help autoload.php — perpustakaan
FOLDERS
perpustakaan
application
cache
config
    autoload.php
    config.php
    constants.php
    database.php
    doctypes.php
    foreign_chars.php
    hooks.php
    index.html
    memcached.php
    migration.php
    mimes.php
    profiler.php
    routes.php
    smileys.php
    user_agents.php
controllers
    core
    helpers
    hooks
    language
    libraries
    logs
    models
    third_party
    views
        .htaccess
        index.html
system
user_guide

config.php
50 | You can also supply an alternative library name to be assigned
51 | in the controller:
52 |
53 | $autoload['libraries'] = array('user_agent' => 'ua');
54 |
55 | $autoload['libraries'] = array('database', 'session', 'form_validation');
56 |
57 | -----
58 | Auto-load Drivers
59 |
60 | These classes are located in system/libraries/ or in your
61 | application/libraries/ directory, but are also placed inside their
62 | own subdirectory and they extend the CI_Driver_Library class. They
63 | offer multiple interchangeable driver options.
64 |
65 | -----
66 | Prototype:
67 |
68 | $autoload['drivers'] = array('cache');
69 |
70 | You can also supply an alternative property name to be assigned in
71 | the controller:
72 |
73 | $autoload['drivers'] = array('cache' => 'cch');
74 |
75 | -----
76 | Prototype:
77 |
78 | $autoload['helper'] = array('url', 'file');
79 |
80 | -----
81 | Prototype:
82 | $autoload['helper'] = array();
83 |
84 | -----
85 | Auto-load Helper Files
86 |
87 | Prototype:
88 |
89 | $autoload['helper'] = array('url', 'file');
90 |
91 | -----
92 | $autoload['helper'] = array('url');
93 |
94 | -----
95 | Auto-load Config files
```

Proses autoload library dan helper selesai.

## Menghilangkan Index.php dari URL codeigniter

Kita perlu menghilangkan index.php dari URL codeigniter di project sistem informasi perpustakaan ini. agar struktur url aplikasi kita terlihat bersih. Dan caranya pun sangat mudah.

Pertama, buka file **config.php**

**application/config/config.php**

Cari syntax

```
$config['index_page'] = 'index.php';
```

Lalu hapus index.php, sehingga menjadi

```
$config['index_page'] = '';
```

Perhatikan pada gambar berikut

```

13 | 
14 | WARNING: You MUST set this value!
15 | 
16 | If it is not set, then CodeIgniter will try guess the protocol and path
17 | your installation, but due to security concerns the hostname will be set
18 | to $_SERVER['SERVER_ADDR'] if available, or localhost otherwise,
19 | The auto-detection mechanism exists only for convenience during
20 | development and MUST NOT be used in production!
21 | 
22 | If you need to allow multiple domains, remember that this file is still
23 | a PHP script and you can easily do that on your own.
24 | 
25 | */
26 | $config['base_url'] = 'http://localhost/perpustakaan/';
27 | /*
28 | 
29 | Index File
30 | -----
31 | 
32 | 
33 | Typically this will be your index.php file, unless you've renamed it to
34 | something else. If you are using mod_rewrite to remove the page set this
35 | variable so that it is blank.
36 | 
37 | */
38 | $config['index_page'] = '';
39 | /*
40 | 
41 | URI PROTOCOL
42 | -----
43 | 
44 | 
45 | This item determines which server global should be used to retrieve the
46 | URI string. The default setting of 'REQUEST_URI' works for most servers.
47 | If your links do not seem to work, try one of the other delicious flavors:
48 | 
49 | 'REQUEST_URI'    Uses $_SERVER['REQUEST_URI']
50 | 'QUERY_STRING'   Uses $_SERVER['QUERY_STRING']
51 | 'PATH_INFO'      Uses $_SERVER['PATH_INFO']
52 |

```

Kedua, buatlah sebuah file **.htaccess** pada directory root project perpustakaan.

### .htaccess

```

RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php/$1 [L]

```

```

1 RewriteEngine On
2 RewriteCond %{REQUEST_FILENAME} !-f
3 RewriteCond %{REQUEST_FILENAME} !-d
4 RewriteRule ^(.*)$ index.php/$1 [L]

```

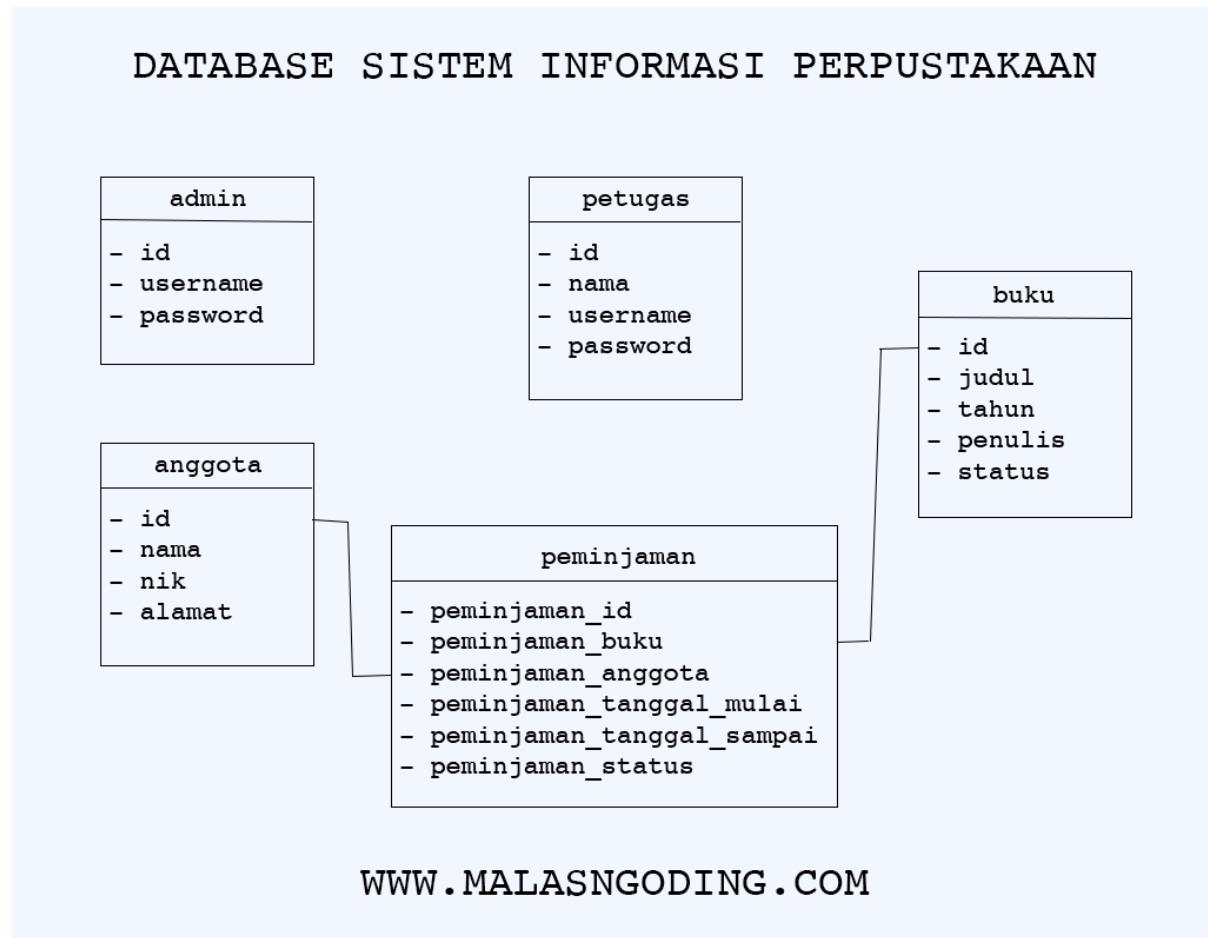
Simpan dan selesai.

Selanjutnya kita akan membuat database untuk sistem informasi perpustakaan yang akan kita bangun. Kemudian melakukan konfigurasi database pada codeigniter.

# Merancang & Membuat Database Perpustakaan, Serta Konfigurasi di Codeigniter

Sebelum mulai membuat sebuah sistem informasi perpustakaan ini, saya akan menguraikan dulu bentuk dan susunan database yang akan kita buat. Agar kita lebih paham dengan sistem informasi perpustakaan yang akan kita buat. Ke depannya teman-teman bisa menyuajikan dengan kebutuhan sistem informasi yang teman-teman buat. Perancangan database sederhana ini hanya sebagai simulasi database atas kebutuhan sistem informasi perpustakaan yang akan kita buat.

Berikut adalah gambaran struktur database yang akan kita buat untuk sistem informasi perpustakaan ini.



Gambar di atas merupakan gambaran struktur database yang akan kita buat dalam membuat sistem informasi perpustakaan sederhana ini. Terdapat 5 buah table yang kita butuhkan. Yaitu table *admin*, *petugas*, *buku*, *anggota* dan *peminjaman*. Yang mana kegunaan dari masing-masing table di atas akan saya jelaskan lebih rinci pada bagian selanjutnya.

## Alur sistem informasi perpustakaan

Sistem informasi perpustakaan yang akan kita buat memiliki 2 orang user atau pengguna yang bisa login ke sistem informasi perpustakaan ini. Yaitu *admin* dan *petugas*. Admin memiliki akses untuk mengelola data petugas. Seperti menambahkan petugas baru, mengedit data petugas dan menghapus petugas. Tapi admin tidak bisa mengelola data buku, anggota dan transaksi peminjaman. Jadi admin hanya bisa melihat data buku, melihat data anggota, melihat data peminjaman serta mencetak laporan peminjaman.

Karena yang mengelola data buku, anggota dan data peminjaman adalah si petugas. Petugas dapat mengelola data buku, mengelola data anggota serta mengelola data peminjaman seperti menginput peminjaman baru, membatalkan peminjaman dan mengkonfirmasi peminjaman yang di lakukan anggota bahwa peminjaman telah selesai. Petugas juga memiliki akses yang sama seperti admin dalam proses mencetak laporan peminjaman berdasarkan tanggal yang di tentukan.

## Mempersiapkan database dan table

Kita akan langsung saja mulai masuk ke tahap persiapan database dan table-table yang akan kita butuhkan untuk membangun sistem informasi perpustakaan ini. Berikut adalah tahap-tahap nya.

Buat database baru dengan nama “**perpustakaan**”. Dan kemudian ada 5 table yang akan kita buat dalam database perpustakaan ini. Di antaranya adalah :

1. admin
2. petugas
3. buku
4. anggota
5. peminjaman

### Tabel admin

Tabel admin akan kita jadikan sebagai tempat untuk menyimpan data admin seperti nama, username dan password.

Nama	Tipe	Auto_increment	Primary Key
<b>id</b>	Int(11)	✓	✓
<b>username</b>	Varchar(255)		
<b>password</b>	Varchar(255) – md5		

## Tabel petugas

Tabel petugas akan kita jadikan sebagai tempat menyimpan data petugas perpustakaan. Adapun data petugas yang kita simpan adalah data nama, username dan password.

Nama	Tipe	Auto_increment	Primary Key
<b>id</b>	Int(11)	✓	✓
<b>nama</b>	Varchar(30)		
<b>username</b>	Varchar(20)		
<b>password</b>	Varchar(30) – md5		

## Tabel anggota

Tabel anggota akan kita jadikan sebagai tempat menyimpan data anggota perpustakaan atau data orang yang meminjam buku. seperti data nama, nomor nik dan alamat.

Nama	Tipe	Auto_increment	Primary Key
<b>id</b>	Int(11)	✓	✓
<b>nama</b>	Varchar(255)		
<b>nik</b>	Int(11)		
<b>alamat</b>	text		

## Tabel buku

Tabel buku akan kita jadikan sebagai tempat menyimpan data buku yang ada pada perpustakaan. seperti data judul, tahun terbit, nama penulis dan status ketersediaan buku. Pada status buku kita memberi tipe int. berarti (1=tersedia, 2 = sedang dipinjam).

Nama	Tipe	Auto_increment	Primary Key
<b>id</b>	Int(11)	✓	✓
<b>judul</b>	Varchar(255)		
<b>tahun</b>	Int(11)		
<b>penulis</b>	Varchar(255)		
<b>status</b>	Int(11)		

## Tabel peminjaman

Tabel peminjaman kita gunakan untuk menyimpan data transaksi peminjaman, seperti buku yang di pinjam, anggota yang meminjam, tanggal anggota mulai meminjam buku, tanggal buku yang dipinjam harus dikembalikan, dan status peminjaman.

Nama	Tipe	Auto_increment	Primary Key
peminjaman_id	Int(11)	✓	✓
peminjaman_buku	Int(11)		
peminjaman_anggota	Int(11)		
peminjaman_tanggal_mulai	date		
peminjaman_tanggal_sampai	date		
peminjaman_status	int		

Pada kolumn peminjaman\_buku, data yang di simpan adalah id dari buku yang dipinjam oleh anggota. Begitu juga pada kolumn peminjaman\_anggota, data yang kita simpan adalah data id dari anggota yang meminjam.

Pada kolumn peminjaman\_status, data yang kita simpan adalah nilai 1 atau 2. Yang mana (1=sedang dipinjam, dan 2=selesai atau buku yang dipinjam telah dikembalikan).

## Import database ‘perpustakaan’

Ada alternatif lain untuk membuat table-table yang telah di sebutkan di atas. Jika teman-teman ingin menghemat waktu, teman-teman bisa langsung mengimport sql berikut.

Caranya silahkan buat dulu database dengan nama ‘perpustakaan’, kemudian klik pada menu *sql*. dan pastekan sql berikut. Lalu klik go.

### perpustakaan.sql

```
-- phpMyAdmin SQL Dump
-- version 4.8.0.1
-- https://www.phpmyadmin.net/
--
-- Host: localhost
-- Generation Time: Jun 18, 2018 at 02:15 PM
-- Server version: 10.1.32-MariaDB
-- PHP Version: 5.6.36

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `perpustakaan`
--
```

```

-- -----
-- 
-- Table structure for table `admin`
-- 

CREATE TABLE `admin` (
  `id` int(11) NOT NULL,
  `username` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- 
-- Table structure for table `anggota`
-- 

CREATE TABLE `anggota` (
  `id` int(11) NOT NULL,
  `nama` varchar(255) NOT NULL,
  `nik` int(11) NOT NULL,
  `alamat` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- 
-- Table structure for table `buku`
-- 

CREATE TABLE `buku` (
  `id` int(11) NOT NULL,
  `judul` varchar(255) NOT NULL,
  `tahun` year(4) NOT NULL,
  `penulis` varchar(255) NOT NULL,
  `status` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- 
-- Table structure for table `peminjaman`
-- 

CREATE TABLE `peminjaman` (
  `peminjaman_id` int(11) NOT NULL,
  `peminjaman_buku` int(11) NOT NULL,
  `peminjaman_anggota` int(11) NOT NULL,
  `peminjaman_tanggal_mulai` date NOT NULL,
  `peminjaman_tanggal_sampai` date NOT NULL,
  `peminjaman_status` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- 
-- Table structure for table `petugas`
-- 

CREATE TABLE `petugas` (
  `id` int(11) NOT NULL,
  `nama` varchar(255) NOT NULL,
  `username` varchar(255) NOT NULL,

```

```

    `password` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- 
-- Indexes for dumped tables
-- 

-- 
-- Indexes for table `admin`
-- 
ALTER TABLE `admin`
  ADD PRIMARY KEY (`id`);

-- 
-- Indexes for table `anggota`
-- 
ALTER TABLE `anggota`
  ADD PRIMARY KEY (`id`);

-- 
-- Indexes for table `buku`
-- 
ALTER TABLE `buku`
  ADD PRIMARY KEY (`id`);

-- 
-- Indexes for table `peminjaman`
-- 
ALTER TABLE `peminjaman`
  ADD PRIMARY KEY (`peminjaman_id`);

-- 
-- Indexes for table `petugas`
-- 
ALTER TABLE `petugas`
  ADD PRIMARY KEY (`id`);

-- 
-- AUTO_INCREMENT for dumped tables
-- 

-- 
-- AUTO_INCREMENT for table `admin`
-- 
ALTER TABLE `admin`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;

-- 
-- AUTO_INCREMENT for table `anggota`
-- 
ALTER TABLE `anggota`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;

-- 
-- AUTO_INCREMENT for table `buku`
-- 
ALTER TABLE `buku`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;

-- 
-- AUTO_INCREMENT for table `peminjaman`
-- 
ALTER TABLE `peminjaman`
  MODIFY `peminjaman_id` int(11) NOT NULL AUTO_INCREMENT;

```

```

-- 
-- AUTO_INCREMENT for table `petugas`
-- 

ALTER TABLE `petugas`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Maka hasilnya ke-5 table pun selesai kita buat seperti gambar berikut.

Table	Action	Rows	Type	Collation	Size	Overhead
admin	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	latin1_swedish_ci	16 KiB	-
anggota	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	latin1_swedish_ci	16 KiB	-
buku	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	latin1_swedish_ci	16 KiB	-
peminjaman	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	latin1_swedish_ci	16 KiB	-
petugas	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	latin1_swedish_ci	16 KiB	-
5 tables	<a href="#">Sum</a>	0	InnoDB	latin1_swedish_ci	80 KiB	0 B

## Konfigurasi & Koneksi Database CodeIgniter dengan Database

Setelah selesai membuat database. Dan table-table seperti yang sudah dijelaskan sebelumnya. Selanjutnya kita akan melakukan pengaturan atau konfigurasi database pada codeigniter. Untuk menghubungkan codeigniter dengan database “perpustakaan”.

Untuk melakukan konfigurasi database pada codeigniter, silahkan buka file *database.php* pada **application/config/database.php**.

Lihat pada gambar berikut:

```

    database.php
    58 |     'failover' => array(),
    59 |     'save_queries' => TRUE
    60 | }
    61 |
    62 | /**
    63 | * NOTE: Disabling this will also effectively disable both
    64 | * $this->db->last_query() and profiling of DB queries.
    65 | * When you run a query, with this setting set to TRUE (default),
    66 | * CodeIgniter will store the SQL statement for debugging purposes.
    67 | * However, this may cause high memory usage, especially if you run
    68 | * a lot of SQL queries ... disable this to avoid that problem.
    69 |
    70 | /**
    71 | * The $active_group variable lets you choose which connection group to
    72 | * make active. By default there is only one group (the 'default' group).
    73 | */
    74 | $active_group = 'default';
    75 | $query_builder = TRUE;
    76 |
    77 | $db['default'] = array(
    78 |     'dsn' => '',
    79 |     'hostname' => 'localhost',
    80 |     'username' => 'root', // username mysql
    81 |     'password' => '',
    82 |     'database' => 'perpustakaan', // nama database yang akan di gunakan
    83 |     'dbdriver' => 'mysqli',
    84 |     'dbprefix' => '',
    85 |     'pconnect' => FALSE,
    86 |     'db_debug' => (ENVIRONMENT !== 'production'),
    87 |     'cache_on' => FALSE,
    88 |     'cachedir' => '',
    89 |     'char_set' => 'utf8',
    90 |     'dbcollat' => 'utf8_general_ci',
    91 |     'swap_pre' => '',
    92 |     'encrypt' => FALSE,
    93 |     'compress' => FALSE,
    94 |     'stricton' => FALSE,
    95 |     'failover' => array(),
    96 |     'save_queries' => TRUE
    97 );

```

Sesuaikan dengan username dan password mysql teman-teman. Disini username dan password mysql saya masih default, yaitu username "root" dan password nya kosong. Jangan lupa untuk menambahkan nama database "perpustakaan" seperti pada gambar di atas. Karena database yang kita gunakan adalah database yang kita gunakan tadi, yaitu "perpustakaan".

```

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

```

Konfigurasi dan koneksi database codeigniter dengan database mysql pun selesai.

## Membuat Model Yang Terstruktur dan Efisien Untuk CRUD

Ini merupakan cara yang biasa saya lakukan setiap membuat aplikasi dengan codeigniter. Saya lebih senang membuat fungsi-fungsi yang bisa saya gunakan berulang kali, tanpa harus repot banyak-banyak membuat fungsi pada model. contohnya untuk insert data ke database, kita buat satu fungsi untuk insert data ke database. Dan bisa kita gunakan secara berulang-ulang kali. Begitu juga dengan fungsi edit, hapus dan update data pada database. Yuk simak caranya.

Buat sebuah file model dengan nama M\_data.php.

Ingat! Untuk penamaan file model di codeigniter, harus di awali dengan huruf besar. Contoh : M\_data.php

**Ingat!**  
Penamaan file *controller* dan *model* harus di awali dengan **huruf besar** di codeigniter 3 (disarankan).

Memang tidak masalah jika dibuat dengan huruf kecil, tetapi akan menyalahi aturan penggunaan codeigniter yang benar dan biasanya akan terjadi trouble saat di upload di hosting.

**perpustakaan/application/models/M\_data.php**

```
<?php

// WWW.MALASNGODING.COM | Author : Diki Alfarabi Hadi
// Model yang terstruktur. agar bisa digunakan berulang kali untuk membuat operasi
CRUD.
// Sehingga proses pembuatan CRUD menjadi lebih cepat dan efisien.

class M_data extends CI_Model{

    // FUNGSI CRUD
    // fungsi untuk mengambil data dari database
    function get_data($table){
        return $this->db->get($table);
    }

    // fungsi untuk menginput data ke database
    function insert_data($data,$table){
        $this->db->insert($table,$data);
    }

    // fungsi untuk mengedit data
    function edit_data($where,$table){
        return $this->db->get_where($table,$where);
    }

    // fungsi untuk mengupdate atau mengubah data di database
    function update_data($where,$data,$table){
        $this->db->where($where);
        $this->db->update($table,$data);
    }
}
```

```
}

// fungsi untuk menghapus data dari database
function delete_data($where,$table) {
    $this->db->delete($table,$where);
}
// AKHIR FUNGSI CRUD

}
?>
```

Dengan model seperti di atas, kita bisa menggunakannya secara berulang-ulang tanpa membuat 1 function model untuk 1 function pengelolaan database. Untuk menginput data ke database kita bisa menggunakan function *insert\_data()*, untuk menampilkan data atau mengambil data dari database kita bisa menggunakan function *get\_data()*. Untuk mengambil data tertentu bisa menggunakan *edit\_data()* , untuk menghapus data dari database kita bisa menggunakan function *delete\_data()*, dan untuk mengupdate data pada database kita bisa menggunakan function *update\_data()*.

Misalnya kita bisa menggunakan fungsi *insert\_data()* untuk menginput data ke table buku, kita juga bisa menggunakan fungsi *insert\_data()* untuk menginput data ke table anggota. Tanpa harus membuat masing-masing fungsi untuk menginput data ke table masing-masing. Begitu juga dengan fungsi *delete\_data()*, *edit\_data()*, *get\_data* dan *update\_data()*.

Kemudian kita harus mengatur autoload model di *application/config/autoload.php*. agar model *m\_data.php* bisa di load secara otomatis setiap controller/aplikasi di jalankan. Jadi tidak perlu lagi kita load dari controller secara manual.

### **application/config/autoload.php**

Cari syntax barikut

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

Dan ubah menjadi

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

Nah, sampai disini kita telah selesai melakukan semua autoload library, helper dan model yang kita butuhkan dalam pembuatan sistem informasi perpustakaan.

Jika teman-teman mengikuti panduan ini dengan benar, maka file *autoload.php* nya akan seperti berikut :

```

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

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

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

/*
| -----
| Auto-load Drivers

```

```

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

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

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

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

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

```

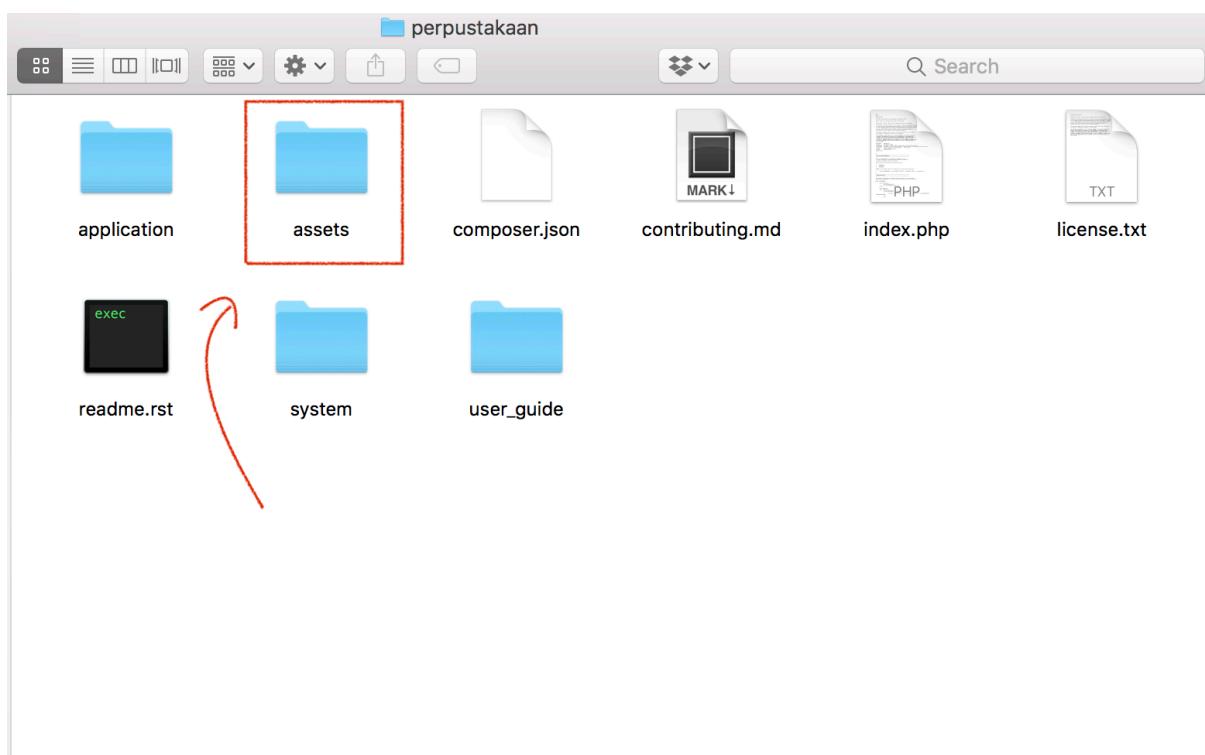
```
| in the controller:  
|  
|     $autoload['model'] = array('first_model' => 'first');  
*/  
$autoload['model'] = array('m_data');
```

Selesai, jangan lupa untuk di save. Selanjutnya kita akan melakukan instalasi Bootstrap pada codeigniter.

# Integrasi Codeigniter dan Bootstrap

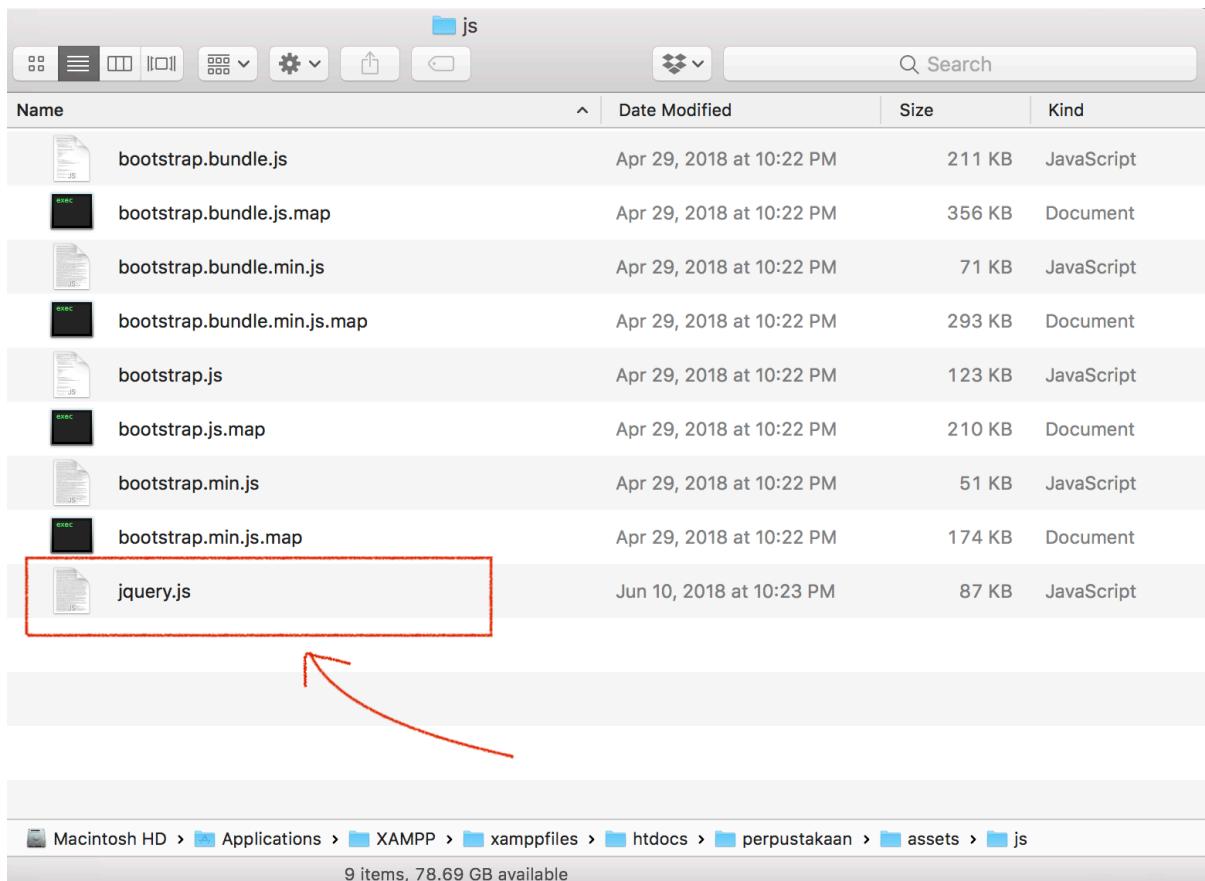
Setelah beberapa persiapan dan pengaturan codeigniter, selanjutnya kita masuk ke tahap integrasi bootstrap dengan codeigniter. Teman-teman bisa mendownload bootstrap dari situs resminya di <http://getbootstrap.com/>, atau bisa menggunakan bootstrap yang telah di sertakan di dalam folder “library” dalam paket ebook ini. Versi bootstrap yang akan kita gunakan dalam membangun sistem informasi perpustakaan ini adalah **bootstrap versi 4**. Atau versi bootstrap terbaru saat ini.

Ekstrak bootstrap ke directory root project “perpustakaan” ini. Lalu rename nama folder menjadi “**assets**”.



Oh ya untuk tutorial bootstrap juga udah lengkap di [www.malasngoding.com](http://www.malasngoding.com), silahkan di cek langsung. ☺

Download juga file **jquery.js** sebagai pendukung bootstrap. Karena javascript pada bootstrap juga di develop menggunakan jquery. dan letakkan jquery.js dalam folder “**assets/js**”. File jquery.js juga telah di sertakan di dalam folder “library” dalam paket ebook ini.



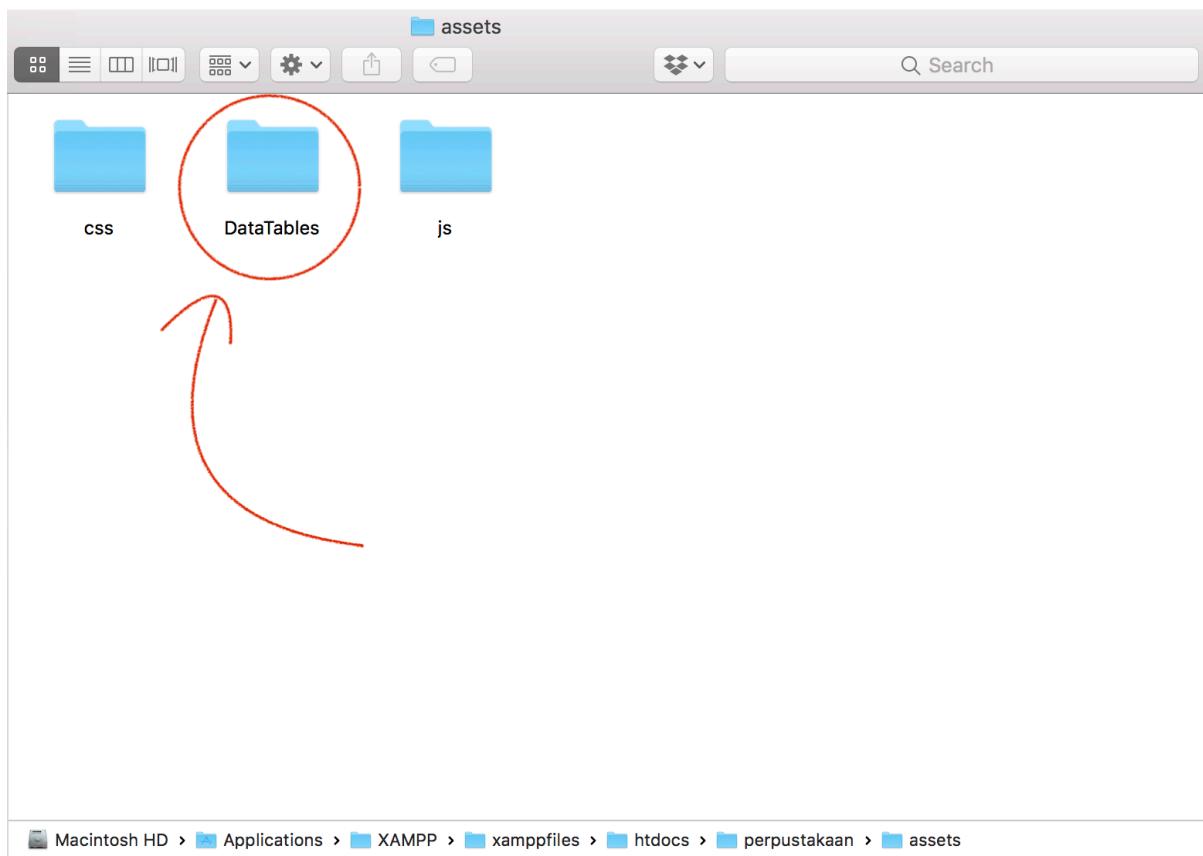
sampai di sini instalasi bootstrap telah selesai, dan sudah siap untuk digunakan. Kita tinggalkan sejenak dan kita akan beranjak ke instalasi atau integrasi codeigniter dengan datatables.

# Integrasi CodeIgniter Dengan DataTables

DataTable adalah plugin atau library jquery yang sangat berguna dalam membangun sebuah sistem atau aplikasi. Pasalnya kita tidak perlu lagi membuat pagination maupun form untuk pencarian data. Karena pagination dan form pencarian data serta sorting data sudah ditangani oleh plugin datatables ini.

Untuk cara instalasinya pun tidak sulit. Kita bisa langsung mendownload datatables dari situs resminya di <https://datatables.net/>. Atau teman-teman bisa menggunakan library datatables yang sudah saya sertakan dalam paket ebook ini, yaitu ada dalam folder "library".

Copy dan ekstrak dalam folder "assets".



Oke sampai di sini instalasi datatables pada codeigniter selesai. Tinggal selanjutnya saya jelaskan cara menghubungkan nya dan memanggil fungsi datatables pada tahap membuat halaman dashboard admin dan petugas.

Sebenarnya cara penggunaan datatables ini sangatlah mudah, kita tinggal menggunakan fungsi datatables untuk membuat data pada table menjadi mempunyai form pencarian data dan pagination secara otomatis. Silahkan perhatikan caranya pada bagian footer pada template admin dan petugas nantinya.

# Membuat Login (md5) Codeigniter Dengan Pesan Notifikasi

Setelah segala persiapan selesai, kita akan mulai dengan membuat login untuk admin dan petugas (multi user). Halaman login akan kita buat di halaman utama pada saat sistem informasi perpustakaan ini di akses.

Jadi pada saat sistem informasi ini pertama kali diakses, maka akan muncul halaman login untuk admin dan petugas. Oleh karena itu sekarang kita harus membuat controller dan view untuk admin. Kemudian kita tetapkan controller login ini sebagai controller default. Sehingga controller login lah yang di jalankan terlebih dulu pada saat sistem informasi ini dibuka atau diakses.

## Persiapan Membuat Login Admin & Petugas (Multi User)

Tahap persiapannya adalah mempersiapkan sebuah akun admin. Untuk membuat akun admin teman-teman tinggal langsung menginput data baru ke table admin. Di sini saya menginput data admin baru dengan menginput username "admin". Dan passwordnya "admin123". Dan pilih md5 pada kolumn function.

Perhatikan pada gambar berikut.

The screenshot shows the MySQL Workbench interface with the 'admin' table selected. The 'Insert' tab is active. A red box highlights the 'username' field containing 'admin' and the 'password' field containing 'admin123'. An arrow points from the 'Function' dropdown menu to the 'password' field, indicating it is set to MD5. Another arrow points from the 'Go' button to the bottom right.

Column	Type	Function	Null	Value
id	int(11)			
username	varchar(255)			admin
password	varchar(255)	MD5		admin123

Setelah mengisi data username, password serta memilih pilihan md5. Klik tombol go.

Maka data akun admin pun tersimpan.

Showing rows 0 - 0 (1 total, Query took 0.0010 seconds.)

```
SELECT * FROM `admin`
```

	<input type="checkbox"/> Show all	Number of rows:	25	Filter rows:	Search this table						
+ Options	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	<table border="1"> <tr> <th>id</th> <th>username</th> <th>password</th> </tr> <tr> <td>1</td> <td>admin</td> <td>0192023a7bbd73250516f069df18b500</td> </tr> </table>	id	username	password	1	admin	0192023a7bbd73250516f069df18b500
id	username	password									
1	admin	0192023a7bbd73250516f069df18b500									
<input type="checkbox"/>	<input type="checkbox"/> Check all	With selected:	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete						
				<input type="checkbox"/> Export							

Oke, akun admin sudah siap untuk kita gunakan untuk login nantinya. Sekarang kita akan membuat halaman login serta fungsi yang akan memproses login.

## Membuat Login Admin dan Petugas (Multi User)

Pada proses login yang akan kita buat sekarang, admin dan petugas dapat login melalui form login yang sama. Yang membedakannya hanya form pilihan “login sebagai” nya. Jika pada saat login form login sebagainya di pilih sebagai admin, maka username dan password yang di input akan di proses ke table admin. Begitu juga jika di pilih login sebagai petugas, maka data username dan password yang di input akan diproses atau disesuaikan dengan table petugas.

Login akan kita buat pada controller Login.php. oleh karena itu buat sebuah controller dengan nama **“Login.php”**. kemudian kita atur agar controller “Login.php” menjadi controller default. Yaitu controller yang pertama kali dijalankan jika aplikasi diakses. Untuk mengubah controller defaultnya silahkan teman-teman buka file routes.php yang ada di **application/config/routes.php**.

Cari syntax berikut.

```
$route['default_controller'] = 'welcome';
```

Lalu ubah menjadi

```
$route['default_controller'] = 'login';
```

Sampai di sini pengaturan controller telah selesai kita ubah. Yang awalnya controller defaultnya adalah controller bawaan codeigniter, yaitu "Welcome.php". telah menjadi controller "Login.php".

Kemudian pada file controller Login.php akan kita buat fungsi untuk menampilkan halaman login dan sebuah fungsi untuk menangani proses login dari username dan password yang di input pada halaman login. ketik syntax berikut pada controller Login.php.

### Application/controllers/Login.php

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

class Login extends CI_Controller {

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

    }

    // menampilkan halaman login
    public function index(){
        $this->load->view('v_login');
    }

    // validasi login
    function login_aksi(){
        $username = $this->input->post('username');
        $password = $this->input->post('password');
        $sebagai = $this->input->post('sebagai');

        $this->form_validation-
>set_rules('username','Username','required');
        $this->form_validation-
>set_rules('password','Password','required');

        if($this->form_validation->run() != false){
            $where = array(
                'username' => $username,
                'password' => md5($password)
            );

            if($sebagai == "admin"){
                $cek = $this->m_data->cek_login('admin',$where)-
>num_rows();
                $data = $this->m_data->cek_login('admin',$where)-
>row();

                if($cek > 0){
                    $data_session = array(
                        'id' => $data->id,
                        'username' => $data->username,
                        'status' => 'admin_login'
                    );
                }
            }
        }
    }
}
```

```

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

        redirect(base_url().'admin');
    }else{
        redirect(base_url().'login?alert=gagal');
    }

}else if($sebagai == "petugas"){
    $cek = $this->m_data->cek_login('petugas',$where)-
>num_rows();
    $data = $this->m_data->cek_login('petugas',$where)-
>row();

    if($cek > 0){
        $data_session = array(
            'id' => $data->id,
            'nama' => $data->nama,
            'username' => $data->username,
            'status' => 'petugas_login'
        );
        $this->session->set_userdata($data_session);

        redirect(base_url().'petugas');
    }else{
        redirect(base_url().'login?alert=gagal');
    }
}

}

```

Setelah teman-teman selesai menulis syntax di atas pada controller Login.php, akan sedikit saya jelaskan dulu, teman-teman pasti sudah tau apa itu function construct() jika teman-teman pernah belajar OOP, function construct seperti contoh di atas adalah function yang pertama kali akan di jalankan pada saat sebuah class di jalankan. Setelah function construct() di jalankan, selanjutnya yang di jalankan adalah function index(). Ini merupakan ketetapan dari codeigniter.

Perhatikan juga pada function index() di atas.

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

Pada function index() kita membuat perintah untuk membuka view v\_login.php. Syntax \$this->load->view() adalah perintah untuk menampilkan view (tampilan). Di sini kita memerintahkan untuk menampilkan view login (v\_login.php). ekstensi .php nya tidak perlu lagi kita tuliskan saat memanggil view. Cukup seperti perintah di atas.

**Ingat !**

Function index() adalah function yang pertama kali di jalankan setelah  
function construct() pada sebuah controller.

Pada saat controller login di jalankan, otomatis function index juga di jalankan, dan akan menampilkan halaman login (view v\_login.php).

Karena pada function index kita ingin menampilkan view v\_login.php, maka sekarang kita buat dulu view **v\_login.php** nya. Buat v\_login.php dalam folder views.

### **application/views/v\_login.php**

```
<!DOCTYPE html>
<html>
<head>
    <title>Login - Sistem Informasi Perpustakaan Malas Ngoding</title>
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url().'assets/css/bootstrap.css' ?>">
    <script type="text/javascript" src="<?php echo
base_url().'assets/js/jquery.js' ?>"></script>
    <script type="text/javascript" src="<?php echo
base_url().'assets/js/bootstrap.js' ?>"></script>
</head>
<body class="bg-dark">

    <div class="container">
        <br/><br/><br/><br/>

        <h3 class="font-weight-normal text-center text-white">SISTEM
INFORMASI</h3>
        <h2 class="font-weight-normal text-center text-white mb-
5"><b>PERPUSTAKAAN</b></h2>

        <div class="col-md-4 offset-md-4">
            <div class="card">
                <div class="card-body">
                    <?php
                    if(isset($_GET['alert'])) {
                        if($_GET['alert']=="gagal") {
                            echo "<div class='alert alert-danger font-weight-bold text-
center'>LOGIN GAGAL!</div>";
                        } else if($_GET['alert']=="belum_login") {
                            echo "<div class='alert alert-danger font-weight-bold text-
center'>SILAHKAN LOGIN TERLEBIH DULU!</div>";
                        }
                    }
                </div>
            </div>
        </div>
    </div>
</body>
```

```

        }else if($_GET['alert']=="logout") {
            echo "<div class='alert alert-success font-weight-bold text-center'>ANDA TELAH LOGOUT!</div>";
        }
    }
?>

<h4 class="font-weight-bold text-center mb-3 mt-3">LOGIN</h4>

<!-- validasi error -->
<?php echo validation_errors(); ?>

<form method="post" action=<?php echo
base_url().'login/login_aksi'; ?>>
    <div class="form-group">
        <label for="username">Username</label>
        <input name="username" type="text" class="form-control"
placeholder="Masukkan username">
    </div>
    <div class="form-group">
        <label for="password">Password</label>
        <input name="password" type="password" class="form-control"
placeholder="Masukkan Password">
    </div>
    <div class="form-group">
        <label for="sebagai">Login Sebagai :</label>
        <select name="sebagai" class="form-control">
            <option value="admin">Admin</option>
            <option value="petugas">Petugas</option>
        </select>
    </div>
    <button type="submit" class="btn btn-primary">Login</button>
</form>
</div>

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

</body>
</html>

```

Perhatikan pada tag `<head> ... </head>` di atas, Di sinilah terlihat salah satu kegunaan dari fungsi `base_url()` yang sudah dijelaskan di atas tadi.

```

<link rel="stylesheet" type="text/css" href=<?php echo
base_url().'assets/css/bootstrap.css' ?>>
<script type="text/javascript" src=<?php echo
base_url().'assets/js/jquery.js' ?>></script>
<script type="text/javascript" src=<?php echo
base_url().'assets/js/bootstrap.js' ?>></script>

```

Pada pengaturan `config['base_url']` sebelumnya kita telah menetapkannya dengan `http://localhost/perpustakaan/`. Jadi saat kita gunakan seperti contoh

menghubungkan template dengan bootstrap seperti di atas, maka hasilnya seperti menuju url bootstrap dan jquery yang telah kita instalasi pada project codeigniter ini seperti yang telah dijelaskan sebelumnya. Hasil urlnya menjadi seperti

```
<link rel="stylesheet" type="text/css" href="http://localhost/perpustakaan  
/assets/css/bootstrap.css' ?>">
```

Karena `base_url()` sudah sama dengan `http://localhost/perpustakaan/`.

Kemudian lagi perhatikan pada form yang sudah kita buat pada view `v_login.php` di atas.

```
<form method="post" action="<?php echo base_url().'login/login_aksi'; ?>">  
    <div class="form-group">  
        <label for="username">Username</label>  
        <input name="username" type="text" class="form-control"  
placeholder="Masukkan username">  
    </div>  
    <div class="form-group">  
        <label for="password">Password</label>  
        <input name="password" type="password" class="form-control"  
placeholder="Masukkan Password">  
    </div>  
    <div class="form-group">  
        <label for="sebagai">Login Sebagai :</label>  
        <select name="sebagai" class="form-control">  
            <option value="admin">Admin</option>  
            <option value="petugas">Petugas</option>  
        </select>  
    </div>  
    <button type="submit" class="btn btn-primary">Login</button>  
</form>
```

Action dari form ini telah kita atur ke function/method **login\_aksi** di controller **login**.

```
<?php echo base_url().'login/login_aksi'; ?>
```

Perhatikan method/function **login\_aksi** pada controller login yang telah kita buat. Karena function ini yang akan dieksekusi sebagai *form handler* dari form login kita tadi.

```
function login_aksi(){  
    $username = $this->input->post('username');  
    $password = $this->input->post('password');  
    $sebagai = $this->input->post('sebagai');  
  
    $this->form_validation->set_rules('username', 'Username', 'required');  
    $this->form_validation->set_rules('password', 'Password', 'required');
```

```

if($this->form_validation->run() != false){
    $where = array(
        'username' => $username,
        'password' => md5($password)
    );

    if($sebagai == "admin"){
        $cek = $this->m_data->cek_login('admin',$where)-
>num_rows();
        $data = $this->m_data->cek_login('admin',$where)->row();

        if($cek > 0){
            $data_session = array(
                'id' => $data->id,
                'username' => $data->username,
                'status' => 'admin_login'
            );

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

            redirect(base_url().'admin');
        }else{
            redirect(base_url().'login?alert=gagal');
        }
    }else if($sebagai == "petugas"){
        $cek = $this->m_data->cek_login('petugas',$where)-
>num_rows();
        $data = $this->m_data->cek_login('petugas',$where)-
>row();

        if($cek > 0){
            $data_session = array(
                'id' => $data->id,
                'nama' => $data->nama,
                'username' => $data->username,
                'status' => 'petugas_login'
            );

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

            redirect(base_url().'petugas');
        }else{
            redirect(base_url().'login?alert=gagal');
        }
    }
}else{
    $this->load->view('v_login');
}
}

```

Pertama kita menangkap data yang dikirim dari form. Dan kita tetapkan validasinya, bahwa username dan password harus wajib di isi (required).

```

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

$this->form_validation->set_rules('username','Username','required');
$this->form_validation->set_rules('password','Password','required');

```

Kemudian kita cek validasi nya.

```

if($this->form_validation->run() != false){
    $where = array(
        'username' => $username,
        'password' => md5($password)
    );

    if($sebagai == "admin"){
        $cek = $this->m_data->cek_login('admin',$where)->num_rows();
        $data = $this->m_data->cek_login('admin',$where)->row();

        if($cek > 0){
            $data_session = array(
                'id' => $data->id,
                'username' => $data->username,
                'status' => 'admin_login'
            );

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

            redirect(base_url().'admin');
        }else{
            redirect(base_url().'login?alert=gagal');
        }
    }

    else if($sebagai == "petugas"){
        $cek = $this->m_data->cek_login('petugas',$where)->num_rows();
        $data = $this->m_data->cek_login('petugas',$where)->row();

        if($cek > 0){
            $data_session = array(
                'id' => $data->id,
                'nama' => $data->nama,
                'username' => $data->username,
                'status' => 'petugas_login'
            );

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

            redirect(base_url().'petugas');
        }else{
            redirect(base_url().'login?alert=gagal');
        }
    }
}
else{
    $this->load->view('v_login');
}

```

Jika tidak ada masalah atau validasi sesuai, kita cek dulu form “sebagai” nya. Jika form **sebagai** yang dipilih adalah ‘admin’, maka table yang dicek adalah pada table ‘admin’. Jika form sebagai yang dipilih adalah ‘petugas’, maka table yang dicek adalah pada table ‘petugas’.

Jika username dan password yang diinput sesuai dengan username dan password pada tabel admin atau petugas (sesuai login sebagai apa). Jika sesuai maka kita buat session id, nama, username dan session status kita buat dengan ‘admin\_login’ untuk admin, dan ‘petugas\_login’ untuk petugas. Yang mana session id yang kita simpan adalah id admin atau petugas yang berhasil melakukan login.

Setelah kita buat sessionnya, maka kita alihkan halamannya ke controller admin (jika yang login adalah admin), dan halaman kita alihkan ke controller petugas (jika yang login adalah petugas).

```
redirect(base_url().'admin');
```

Atau

```
redirect(base_url().'petugas');
```

Jika login gagal maka kita alihkan kembali ke controller login sambil mengirim data alert melalui url untuk membuat pesan notifikasi bahwa login gagal.

```
redirect(base_url().'login?alert=gagal');
```

Perhatikan kembali pada view **v\_login.php** yang sudah kita buat.

Pada view v\_login.php telah kita buat pengecekan menggunakan *if..else* untuk membuat pesan notifikasi. Jika terdapat data alert yang di kirim dari url, dan jika isinya adalah “gagal”, maka akan di tampilkan pesan notifikasi “ LOGIN GAGAL! ”.

```
<?php
if(isset($_GET['alert'])){
    if($_GET['alert']=="gagal"){
        echo "<div class='alert alert-danger font-weight-bold text-center'>LOGIN GAGAL!</div>";

    }else if($_GET['alert']=="belum_login"){
        echo "<div class='alert alert-danger font-weight-bold text-center'>SILAHKAN LOGIN TERLEBIH DULU!</div>";

    }else if($_GET['alert']=="logout"){
        echo "<div class='alert alert-success font-weight-bold text-"
```

```
center'>ANDA TELAH LOGOUT!</div>";
    }
}
?>
```

Jika isi dari parameter *alert* adalah “*belum\_login*”, maka pesan notifikasi yang ditampilkan adalah pesan “SILAHKAN LOGIN TERLEBIH DULU!”. Dan jika pesannya adalah “*logout*”, maka akan ditampilkan notifikasi “ ANDA TELAH LOGOUT! ”.

Sampai di sini, sistem login kita selesai, tapi belum rampung 100%, karena kita belum mempersiapkan halaman admin dan halaman petugas. Oleh sebab itu kita akan membuat dan mempersiapkan halaman admin dan petugas dulu. Baru setelah itu kita coba apakah login sudah berjalan sebagai mana yang kita perintahkan atau belum.

# Mempersiapkan Halaman Dashboard Admin

Seperti yang telah kita buat pada pengecekan login sebelumnya, jika pada saat login, dipilih login sebagai admin dan login berhasil, kita mengalihkan halaman ke halaman admin (controller admin). Begitu juga jika login yang di pilih login sebagai petugas dan login berhasil, maka akan di alihkan ke halaman petugas (controller petugas).

Pada tahap sub judul ini kita akan membuat halaman dashboard untuk admin. halaman dashboard untuk admin akan kita buat dalam controller admin, sesuai dengan url yang dialihkan jika admin berhasil login (*redirect(base\_url().'admin')* ).

Buat sebuah controller baru dengan nama “Admin.php”.

## Application/controllers/Admin.php

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

class Admin extends CI_Controller {

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

        // cek session yang login, jika session status tidak sama dengan
        session admin_login,maka halaman akan di alihkan kembali ke halaman login.
        if($this->session->userdata('status')!="admin_login"){
            redirect(base_url().'login?alert=belum_login');
        }
    }

    function index(){
        $this->load->view('admin/v_header');
        $this->load->view('admin/v_index');
        $this->load->view('admin/v_footer');
    }

}
```

Pada function construct() kita membuat pengecekan jika si admin belum login, maka akan di alihkan kembali ke halaman login sambil mengirimkan data melalui url menggunakan GET. Ini bertujuan untuk melindungi controller admin dari pengguna yang belum melakukan login.

```
if($this->session->userdata('status') != "login"){
    redirect(base_url().'welcome?pesan=belumlogin');
}
```

Dan pada function index() di controller admin, kita membuat halaman dashboard untuk admin yang telah kita buat pada function index().

```

function index() {
    $this->load->view('admin/v_header');
    $this->load->view('admin/v_index');
    $this->load->view('admin/v_footer');
}

```

Pada function index() kita menampilkan view header, index dan footer. Kenapa kita banyak menampilkan view? Sebenarnya view yang kita tampilkan merupakan sebuah kesatuan, hanya saja di sini kita memisahkan menjadi beberapa bagian, yaitu view v\_header, view v\_index dan view v\_footer. Cara ini sudah di jelaskan pada sub judul sebelumnya pada bagian tutorial codeigniter dasar.

Disarankan untuk membaca tutorial membuat template web sederhana dengan codeigniter sebelum melanjutkan ke depan. Karena tutorial membuat tampilan atau template web sederhana dengan codeigniter sudah di jelaskan pada tutorial codeigniter dasar sebelumnya.

Buat sebuah folder dengan nama '**admin**' di dalam folder views. kemudian buat 3 buah view baru dengan nama v\_index.php, v\_header.php dan v\_footer.php di dalam folder admin.

### **Application/views/admin/v\_header.php**

```

<!DOCTYPE html>
<html>
<head>
    <title>Admin - Sistem Informasi Perpustakaan MALAS NGODING</title>
    <!-- css bootstrap -->
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url().'assets/css/bootstrap.css' ?>">

    <!-- css datatables -->
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url().'assets/DataTables/datatables.css' ?>">

    <!-- icon font awesome -->
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url().'assets/css/awesome/css/font-awesome.css' ?>">

    <!-- jquery dan bootstrap js -->
    <script type="text/javascript" src="<?php echo base_url().'assets/js/jquery.js'
?>"></script>
    <script type="text/javascript" src="<?php echo
base_url().'assets/js/bootstrap.js' ?>"></script>

    <!-- js datatables -->
    <script type="text/javascript" src="<?php echo
base_url().'assets/DataTables/datatables.js' ?>"></script>
</head>
<body>

    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <div class="container-fluid">
            <a class="navbar-brand" href="<?php echo base_url().'admin'; ?>">SI
Perpustakaan</a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-

```

```

target="#navbarText" aria-controls="navbarText" aria-expanded="false" aria-
label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarText">
    <ul class="navbar-nav mr-auto">
        <li class="nav-item">
            <a class="nav-link" href=<?php echo base_url().'admin'; ?>><i
class="fa fa-home"></i> Dashboard</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href=<?php echo base_url().'admin/petugas'; ?>><i
class="fa fa-user"></i> Petugas</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href=<?php echo base_url().'admin/anggota'; ?>><i
class="fa fa-users"></i> Anggota</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href=<?php echo base_url().'admin/buku'; ?>><i
class="fa fa-book"></i> Buku</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href=<?php echo base_url().'admin/peminjaman_laporan'; ?>><i class="fa fa-book"></i> Laporan
Peminjaman</a>
        </li>
        <li class="nav-item">
            <a href=<?php echo base_url().'admin/ganti_password' ?>" class="nav-
link"><i class="fa fa-lock"></i> Ganti Password</a>
        </li>
    </ul>

    <span class="navbar-text mr-3 text-center">
        Halo, <?php echo $this->session->userdata('username'); ?> [admin]
    </span>

    <a href=<?php echo base_url().'admin/logout' ?>" class="btn btn-outline-
light ml-1"><i class="fa fa-power-off"></i> KELUAR</a>

```

```

    </div>
    </div>
</nav>

<br/>
<br/>

```

## Application/views/admin/v\_index.php

```

<div class="container">
    <div class="jumbotron text-center">
        <div class="col-sm-8 mx-auto">
            <h1>Selamat datang!</h1>
            <p>Ini merupakan contoh sistem informasi perpustakaan hasil dari tutorial
<b>ebook tutorial codeigniter lengkap dengan studi kasus membuat sistem informasi
perpustakaan</b>.</p>
            <p>
                Anda telah login sebagai <b><?php echo $this->session-
>userdata('username'); ?></b> [admin].
            </p>
        </div>
    </div>
</div>

```

```

<div class="card">
    <div class="card-header">
        Dashboard
    </div>
    <div class="card-body">
        Halo
    </div>
</div>
</div>

```

## Application/views/admin/v\_footer.php

```

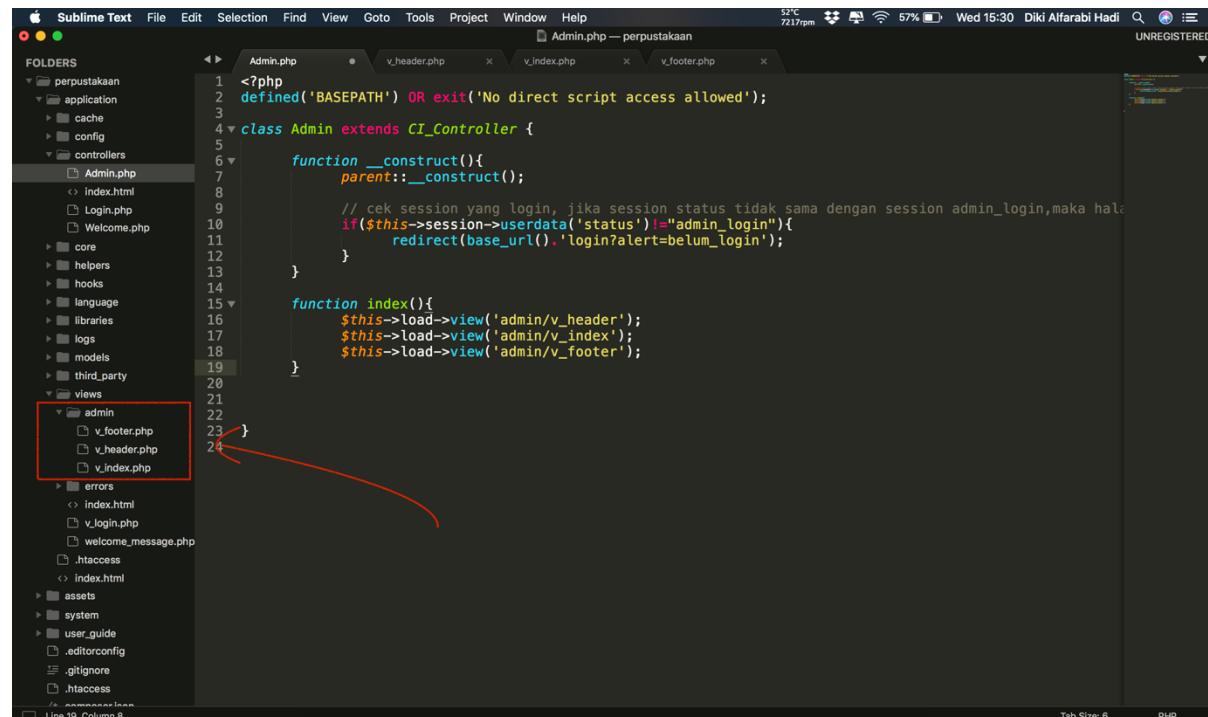
<script type="text/javascript">
    $(document).ready(function(){
        $('.table-datatable').DataTable();
    });
</script>

</body>
</html>

```

Fungsi js yang terdapat pada file view v\_footer.php di atas adalah fungsi untuk mengaktifkan plugin atau library DataTables yang telah dijelaskan sebelumnya.

Jika teman-teman mengikuti dengan benar, maka hasilnya seperti berikut.



The screenshot shows the Sublime Text interface with the Admin.php controller file open. In the sidebar, the views folder contains an admin folder which in turn contains v\_header.php, v\_index.php, and v\_footer.php. A red box highlights the v\_footer.php file. A red arrow points from the text 'Sampai di sini, login yang kita buat telah selesai. Silahkan buka controller login pada browser untuk melihat hasilnya.' to the v\_footer.php file in the sidebar.

```

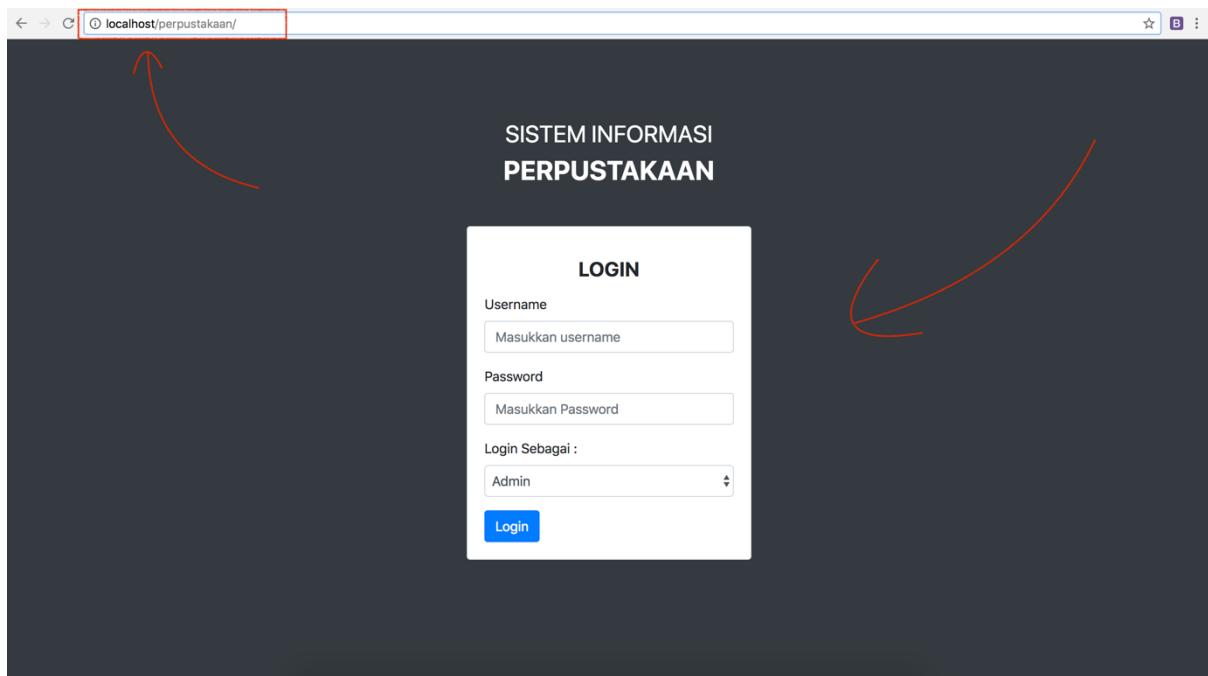
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Admin extends CI_Controller {
5
6     function __construct(){
7         parent::__construct();
8
9         // cek session yang login, jika session status tidak sama dengan session admin_login,maka hal
10        if($this->session->userdata('status') != "admin_login"){
11            redirect(base_url().'login?alert=belum_login');
12        }
13
14
15     function index(){
16         $this->load->view('admin/v_header');
17         $this->load->view('admin/v_index');
18         $this->load->view('admin/v_footer');
19     }
20
21
22
23
24

```

Sampai di sini, login yang kita buat telah selesai. Silahkan buka controller login pada browser untuk melihat hasilnya.

Akses <http://localhost/perpustakaan/>

Perlu di perhatikan di sini, saat kita mengakses <http://localhost/perpustakaan/> sama saja seperti kita mengakses <http://localhost/perpustakaan/login>. karena controller login telah kita set sebagai controller default pada tahap sebelumnya.

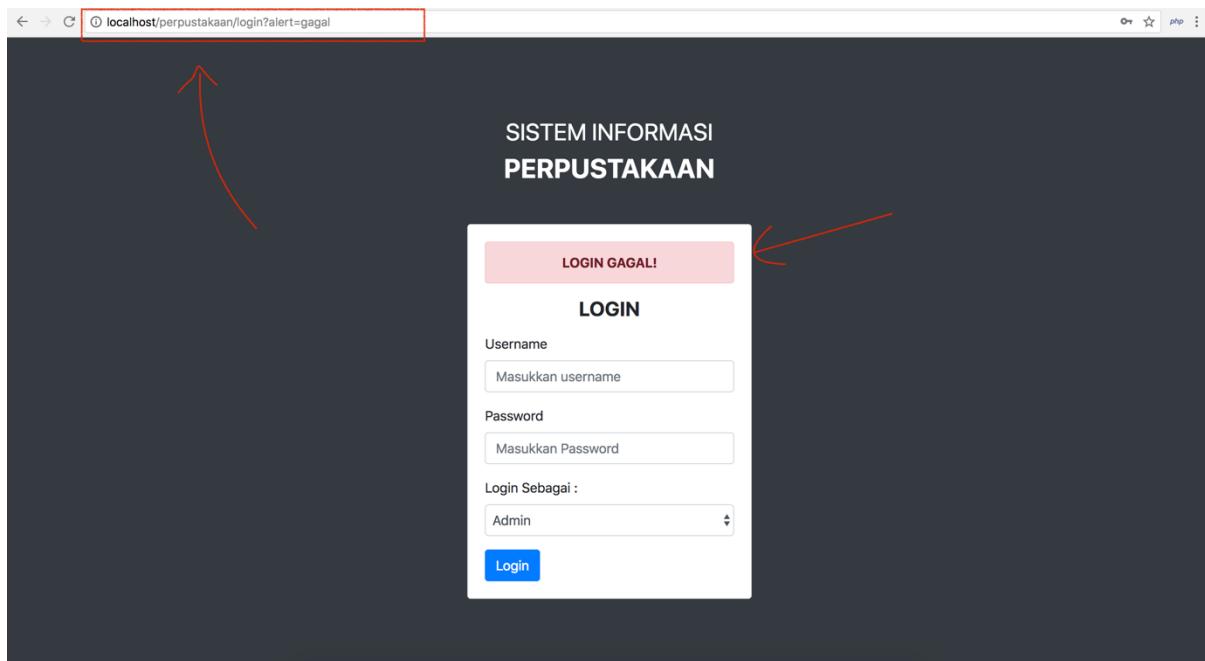


Jika teman-teman mengikuti panduan sebelumnya tentang membuat akun admin atau menginsert data username dan password admin ke tabel admin di atas, maka akun adminnya adalah :

Username : *admin*

Password : *admin123*

Pertama saya akan mencoba memasukkan username dan password yang salah. Untuk memastikan sistem login yang kita buat sudah berjalan sebagaimana mestinya. Di sini saya mencoba memasukkan username : *samsul*, dan passwordnya *samsul123*.

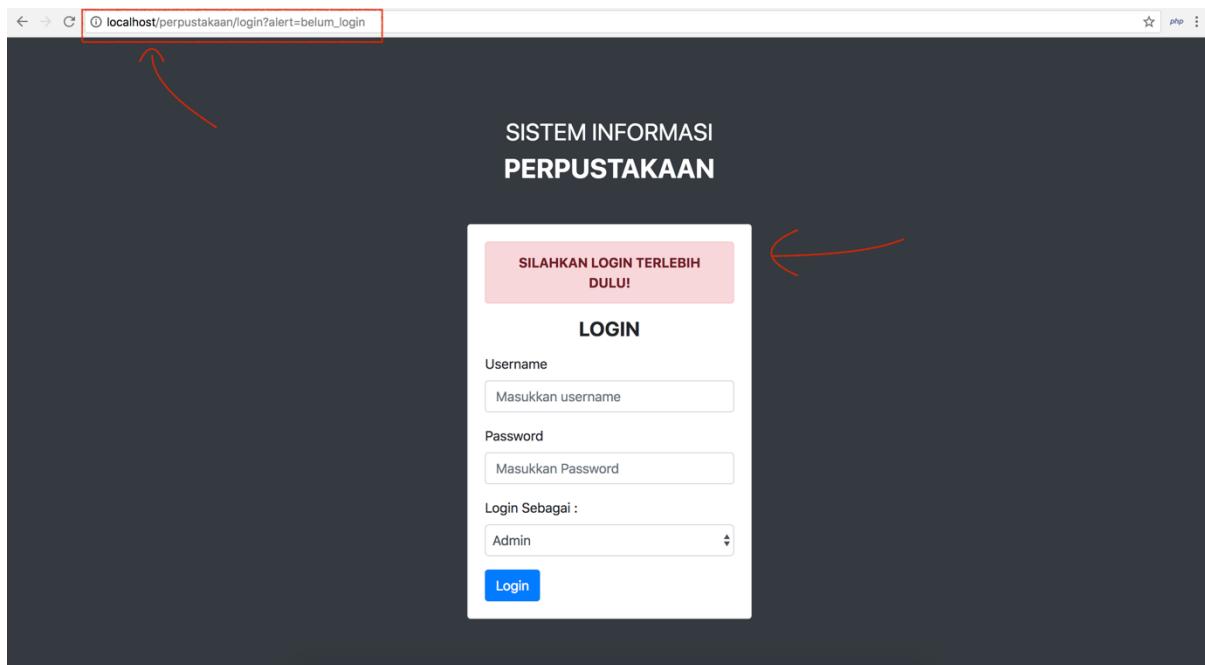


Seperti yang kita harapkan, pada saat username dan password yang di masukkan salah atau tidak sesuai, maka akan muncul pesan notifikasi bahwa “LOGIN GAGAL!”.

Dan coba untuk akses halaman admin tanpa login. Dengan langsung mengakses controller admin. Seharusnya jika belum login, maka tidak bisa mengakses halaman admin (controller admin). Coba kita akses langsung halaman admin dengan mengakses url berikut.

*http://localhost/perpustakaan/admin*

maka akan muncul pesan notifikasi yang mengharuskan kita untuk melakukan login terlebih dulu.



Setelah kita mencoba masuk langsung ke halaman admin tanpa login, dan mencoba memasukkan username password yang salah, sekarang kita akan mencoba memasukkan username dan password yang benar. Coba kita masukkan username = admin, dan password = admin123.

Maka hasilnya kita berhasil login dan masuk ke halaman dashboard admin.

SI Perpustakaan

Dashboard Petugas Anggota Buku Laporan Peminjaman Ganti Password

Halo, admin [admin]

KELUAR

Selamat datang!

Ini merupakan contoh sistem informasi perpustakaan hasil dari tutorial **ebook tutorial codeigniter lengkap dengan studi kasus membuat sistem informasi perpustakaan**.

Anda telah login sebagai **admin** [admin].

Dashboard

Halo

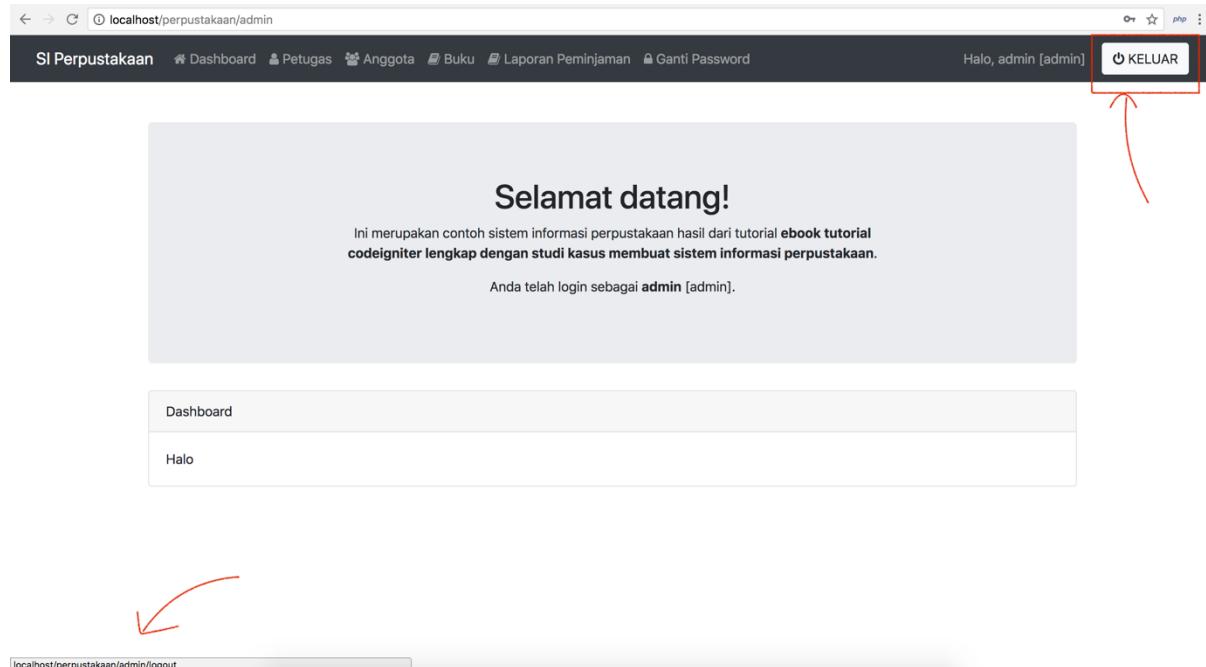
sampai di sini selesailah sudah proses login dan halaman untuk admin. Sedangkan halaman untuk petugas akan kita buat pada pembahasan selanjutnya setelah kita

membuat fungsi untuk mengelola data petugas oleh admin. Karena seperti pada penjelasan sebelumnya. Bahwa admin dapat mengelola data petugas. Jadi di sini setelah admin menambahkan data petugas, baru kita buat halaman untuk petugas. Silahkan teman-teman ikuti step-step pembahasan selanjutnya.

# Membuat Logout Admin

Setelah berhasil membuat login, selanjutnya kita akan membuat logout. Atau link keluar dari sistem. Perhatikan pada menu logout pada view header.php. kita telah membuat link menu "keluar".

```
<a href="=base_url('admin/logout') ?" class="btn btn-outline-light ml-1"><i class="fa fa-power-off"></i> KELUAR</a>
```



Link logout di atas, mengarahkan ke function/method **logout** di controller **admin**.

Oleh karena itu kita akan membuat function **logout** pada controller **admin**.

Tambahkan function logout berikut ini ke controller Admin.php.

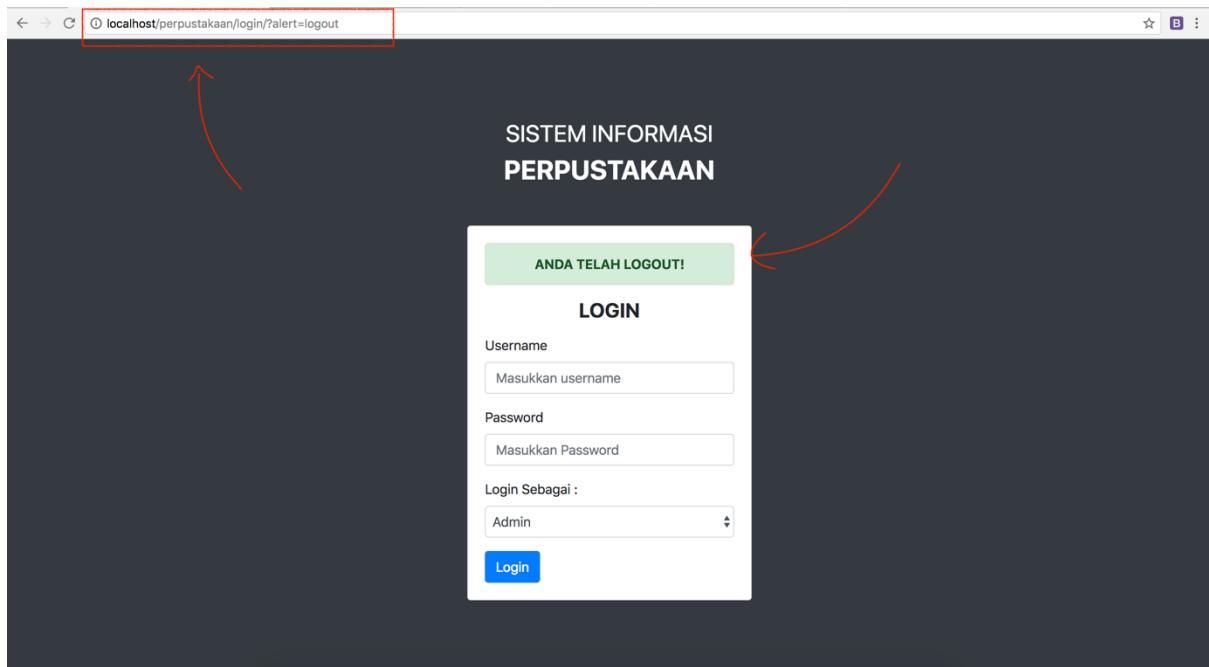
```
function logout(){
    $this->session->sess_destroy();
    redirect(base_url().'login/?alert=logout');
}
```

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Admin extends CI_Controller {
5
6     function __construct(){
7         parent::__construct();
8
9         // cek session yang login, jika session status tidak sama dengan session admin_login,maka halal
10        if($this->session->userdata('status')!="admin_login"){
11            redirect(base_url().'login?alert=belum_login');
12        }
13    }
14
15    function index(){
16        $this->load->view('admin/v_header');
17        $this->load->view('admin/v_index');
18        $this->load->view('admin/v_footer');
19    }
20
21    function logout(){
22        $this->session->sess_destroy();
23        redirect(base_url().'login?alert=logout');
24    }
25
26}
27
28

```

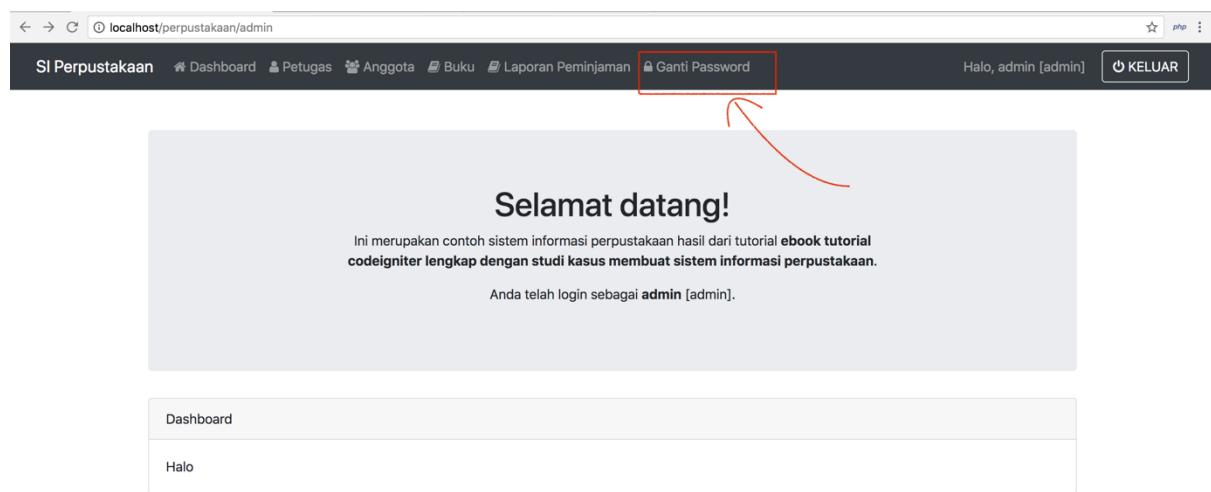
Syntax `$this->session->sess_destroy();` merupakan fungsi codeigniter untuk menghapus semua **session**. Setelah menghapus session, kita alihkan halaman ke halaman login sambil mengirim data alert melalui url untuk membuat pesan notifikasi logout. Sekarang coba klik pada menu "KELUAR". Maka halaman akan dialihkan kembali ke halaman login.



# Membuat Fitur Ganti Password Admin

Sampai pada tahap ini, kita akan membuat sebuah fitur untuk admin, yaitu fitur untuk mengganti password admin. Perhatikan pada menu ganti password di header admin. Kita telah membuatkan link menu untuk mengganti password. Dimana link tersebut sudah kita arahkan ke function **ganti\_password** di controller **Admin.php**.

```
<a href="php echo base_url().'admin/ganti_password' ?&gt;" class="nav-link"&gt;&lt;i class="fa fa-lock"&gt;&lt;/i&gt; Ganti Password&lt;/a&gt;</pre
```



Jadi kita akan membuat sebuah function baru lagi di controller admin, yaitu function **ganti\_password()**. Tambahkan function/syntax berikut pada controller Admin.php.

## Application/controllers/Admin.php

```
function ganti_password(){
    $this->load->view('admin/v_header');
    $this->load->view('admin/v_ganti_password');
    $this->load->view('admin/v_footer');
}
```

Seperti yang teman-teman lihat pada function di atas, kita menampilkan view **v\_header.php**, **view v\_ganti\_password.php** dan **view v\_footer.php**. pada **view v\_ganti\_password.php** akan kita buat form untuk mengganti password. Yuk kita buat sebuah file baru untuk view. Buat file baru dengan nama **v\_ganti\_password.php** dalam folder admin.

## Application/views/admin/v\_ganti\_password.php

```

<div class="container">

    <div class="row justify-content-md-center">
        <div class="col-md-6">
            <div class="card">
                <div class="card-header text-center">
                    <h4>Ganti Password</h4>
                </div>
                <div class="card-body">

                    <?php
                    if(isset($_GET['alert'])) {
                        if($_GET['alert']=="sukses") {
                            echo "<div class='alert alert-success'>Password berhasil
diganti.</div>";
                        }
                    }
                    ?>

                    <?php echo validation_errors(); ?>
                    <form method="post" action="<?php echo
base_url().'admin/ganti_password_aksi'; ?>">
                        <div class="form-group">
                            <label class="font-weight-bold" for="password_baru">Password
Baru</label>
                            <input type="password" class="form-control" name="password_baru"
placeholder="Masukkan password baru">
                        </div>
                        <div class="form-group">
                            <label class="font-weight-bold" for="password_ulang">Ulangi Password
Baru</label>
                            <input type="password" class="form-control" name="password_ulang"
placeholder="Ulangi password baru">
                        </div>

                        <input type="submit" class="btn btn-primary" value="Ubah Password">
                    </form>

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

```

Aksi dari form ini akan di jalankan pada function **ganti\_password\_aksi()** pada controller **admin**.

```
<form action="<?php echo base_url().'admin/ganti_password_aksi' ?>" method="post">
```

Jadi sekarang buat function **ganti\_password\_aksi()** pada controller **admin**.

## Application/controllers/Admin.php

```

function ganti_password_aksi(){
    $baru = $this->input->post('password_baru');
    $ulang = $this->input->post('password_ulang');

    $this->form_validation->set_rules('password_baru', 'Password

```

```

Baru', 'required|matches[password_ulang]');
    $this->form_validation->set_rules('password_ulang', 'Ulangi
Password', 'required');

    if($this->form_validation->run() !=false){
        $id = $this->session->userdata('id');

        $where = array('id' => $id);

        $data = array('password' => md5($baru));

        $this->m_data->update_data($where,$data,'admin');

        redirect(base_url().'admin/ganti_password/?alert=sukses');

    }else{
        $this->load->view('admin/v_header');
        $this->load->view('admin/v_ganti_password');
        $this->load->view('admin/v_footer');
    }
}

```

Syntax di atas adalah function yang kita buat untuk menangani form ganti password.

Perhatikan pada validation yang kita buat, kita menggunakan validation required dan matches. Seperti yang pernah di singgung sebelumnya tentang form validation required, required berarti kita menetapkan form tersebut menjadi wajib di isi atau tidak boleh kosong. Dan matches[] adalah form validation untuk memvalidasi kesamaan value. Di sini kita mewajibkan password\_baru harus sama dengan form password\_ulang.

```

$baru = $this->input->post('password_baru');
$ulang = $this->input->post('password_ulang');

$this->form_validation->set_rules('password_baru', 'Password
Baru', 'required|matches[password_ulang]');
$this->form_validation->set_rules('password_ulang', 'Ulangi Password', 'required');

```

Kemudian kita cek validasinya, jika sesuai dan tidak ada masalah, maka kita update data password si admin.

```

if($this->form_validation->run() !=false){
    $id = $this->session->userdata('id');

    $where = array('id' => $id);

    $data = array('password' => md5($baru));

    $this->m_data->update_data($where,$data,'admin');

    redirect(base_url().'admin/ganti_password/?alert=sukses');

}else{

```

```
        $this->load->view('admin/v_header');
        $this->load->view('admin/v_ganti_password');
        $this->load->view('admin/v_footer');
    }
```

Syntax `$this->session->userdata('id')` merupakan perintah untuk menampilkan session 'id' admin yang sedang login. Setelah data password admin di update, maka kemudian halaman akan di alihkan lagi ke function **ganti\_password()** sambil mengirimkan data alert melalui url.

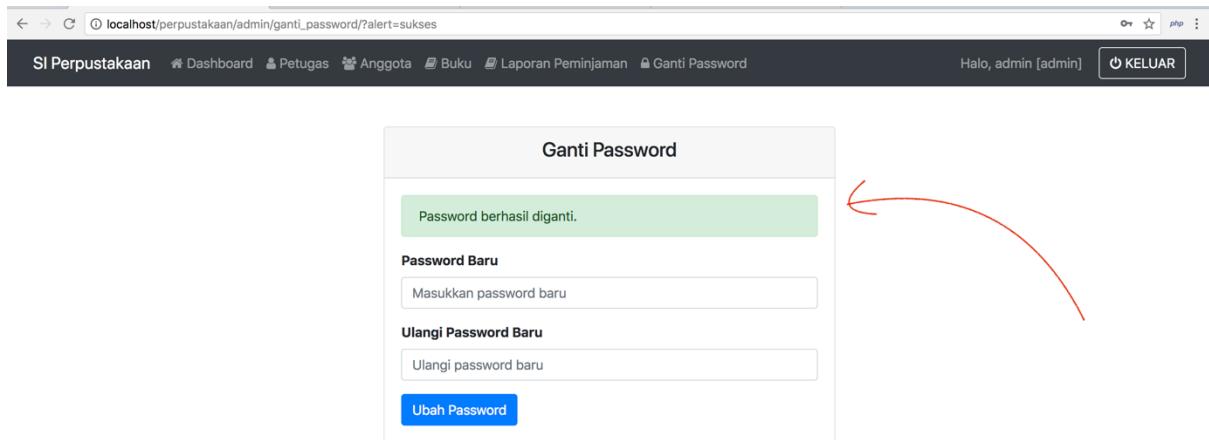
Sekarang kita akan mencoba mengganti password admin. Seperti yang di jelaskan tadi sebelumnya, kita membuat validasi password dimana password baru yang di ketikkan pada form password baru harus sama dengan yang diketikkan pada form ulang password. Ini berguna untuk memvalidasi dan mencegah admin salah memasukkan password barunya (karena kesalahan ketik).

Disini kita akan mencoba mengganti password admin dengan “**malasngoding123**”. Tapi kita coba dulu memasukkan password yang berbeda. Maka hasilnya akan muncul pesan notifikasi bahwa “password baru” dan “ulangi password baru” yang di input harus sama.

The screenshot shows a web application interface for changing a password. At the top, there's a navigation bar with links for Dashboard, Petugas, Anggota, Buku, Laporan Peminjaman, Ganti Password, and a user profile (Halo, admin [admin]). On the right, there's a 'KELUAR' (Logout) button. The main content area has a title 'Ganti Password'. Below it, a red arrow points to an error message: 'The Password Baru field does not match the Ulangi Password field.' There are two input fields: 'Password Baru' (containing 'Masukkan password baru') and 'Ulangi Password Baru' (containing 'Ulangi password baru'). At the bottom is a blue 'Ubah Password' button.

Coba kita masukkan password baru yang sesuai, pada form password baru kita ketik “**malasngoding123**”. Dan pada form ulangi password baru kita ketik

“malasngoding123”. Maka password admin pun berhasil di ganti. Dan muncul pesan notifikasi “Password berhasil diganti.”.



Sampai sini kita telah selesai membuat fitur ganti password untuk akun admin.

# Membuat CRUD Data Petugas Pada Admin

Cara membuat CRUD sudah di jelaskan pada tutorial codeigniter bagian dasar sebelumnya. Pada tahapan ini kita akan membuat CRUD untuk data-data petugas. Seperti perancangan database khusus nya pada tabel petugas yang sudah kita buat. data-data petugas yang akan kita simpan adalah data id, nama, username dan password.

## Menampilkan Data Petugas Dari Database

Data petugas akan kita tampilkan pada function/method petugas() pada controller Admin.php. sesuai dengan link yang sudah kita tentukan pada menu “Petugas”.

```
<a class="nav-link" href="=base_url('admin/petugas');&gt;&lt;i class="fa fa-user"&gt;&lt;/i&gt; Petugas&lt;/a&gt;</pre
```

Tambahkan sebuah function dengan petugas() pada controller Admin.php.

### Application/controllers/Admin.php

```
// CRUD petugas
function petugas(){
    // mengambil data dari database
    $data['petugas'] = $this->m_data->get_data('petugas')->result();
    $this->load->view('admin/v_header');
    $this->load->view('admin/v_petugas',$data);
    $this->load->view('admin/v_footer');
}
```

Pertama, kita akan mengambil data petugas dari database dengan menggunakan function **get\_data()** di model **m\_data**.

```
function get_data($table){
    return $this->db->get($table);
}
```

Dari data yang didapatkan dari hasil query di function get\_data() di atas kita return kembali untuk kita masukkan ke variabel array petugas.

```
$data['petugas'] = $this->m_data->get_data('petugas')->result();
$this->load->view('admin/v_header');
$this->load->view('admin/v_petugas',$data);
$this->load->view('admin/v_footer');
```

Lalu kita parsing datanya ke view mobil.php seperti pada syntax di atas. jadi di sini pada view v\_petugas.php lah kita akan menampilkan data mobilnya.

Buat view baru lagi di dalam folder admin dengan nama v\_petugas.php

### Application/views/admin/v\_petugas.php

```
<div class="container">
<div class="card">
    <div class="card-header text-center">
        <h4>Data Petugas</h4>
    </div>
    <div class="card-body">

        <a href=<?php echo base_url().'admin/petugas_tambah' ?>" class='btn btn-sm btn-success pull-right'><i class="fa fa-plus"></i> Petugas Baru</a>
        <br/>
        <br/>

        <table class="table table-bordered table-striped table-hover">
            <tr>
                <th width="1%">No</th>
                <th>Nama</th>
                <th>Username</th>
                <th width="16%">Opsi</th>
            </tr>
            <?php
            $no = 1;
            foreach($petugas as $p) {
                ?>
                <tr>
                    <td><?php echo $no++; ?></td>
                    <td><?php echo $p->nama; ?></td>
                    <td><?php echo $p->username; ?></td>
                    <td>
                        <a href=<?php echo base_url().'admin/petugas_edit/'.$p->id; ?>" class="btn btn-sm btn-warning"><i class="fa fa-wrench"></i> Edit</a>
                        <a href=<?php echo base_url().'admin/petugas_hapus/'.$p->id; ?>" class="btn btn-sm btn-danger"><i class="fa fa-trash"></i> Hapus</a>
                    </td>
                </tr>
                <?php
            }
            ?>
        </table>

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

Sekarang coba kita klik pada menu “Petugas”. Maka halaman akan diarahkan ke method/function petugas(). Dan data petugas pun berhasil di tampilkan.

The screenshot shows a web application interface for a library management system. At the top, there's a header with the title 'SI Perpustakaan' and links for Dashboard, Petugas, Anggota, Buku, Laporan Peminjaman, Ganti Password, and a log-in status for 'Halo, admin [admin]'. On the right, there's a 'KELUAR' button. Below the header, the main content area has a title 'Data Petugas' and a table with columns 'No', 'Nama', 'Username', and 'Opsi'. A green button labeled '+ Petugas Baru' is located at the top right of the table area. A red arrow points from the text above to this button.

Sampai di sini kita telah selesai menampilkan data petugas. Tapi karena kita masih belum memiliki data di tabel petugas, jadi masih kosong. Selanjutnya kita akan membuat form input data petugas, untuk menginput data petugas ke database.

## Input Data Petugas ke Database

Form input data petugas akan kita buat pada method/function **petugas\_tambah()** dalam controller **Admin.php**. seperti yang sudah kita arahkan pada tombol “Petugas Baru”.

```
<a href="php echo base_url().'admin/petugas_tambah' ?&gt;" class='btn btn-sm btn-success pull-right'&gt;&lt;i class="fa fa-plus"&gt;&lt;/i&gt; Petugas Baru&lt;/a&gt;</pre
```

Buat function **petugas\_tambah()** baru pada controller **Admin.php**.

### Application/controllers/Admin.php

```
function petugas_tambah(){
    $this->load->view('admin/v_header');
    $this->load->view('admin/v_petugas_tambah');
    $this->load->view('admin/v_footer');
}
```

Pada function **petugas\_tambah()** ini kita akan membuat form penginputan data petugas baru. Yaitu akan kita buat pada view **v\_petugas\_tambah.php**. seperti yang sudah kita tetapkan pada function **petugas\_tambah**.

## Application/views/admin/v\_petugas\_tambah.php

```
<div class="container">
    <div class="card">
        <div class="card-header text-center">
            <h4>Tambah Petugas Baru</h4>
        </div>
        <div class="card-body">
            <a href="php echo base_url().'admin/petugas' ?&gt;" class="btn btn-sm btn-light btn-outline-dark pull-right"&gt;&lt;i class="fa fa-arrow-left"&gt;&lt;/i&gt; Kembali&lt;/a&gt;
            &lt;br/&gt;
            &lt;br/&gt;

            &lt;form method="post" action="<?php echo base_url().'admin/petugas_tambah_aksi'; ?&gt;"&gt;
                &lt;div class="form-group"&gt;
                    &lt;label class="font-weight-bold" for="nama"&gt;Nama Lengkap&lt;/label&gt;
                    &lt;input type="text" class="form-control" name="nama" placeholder="Masukkan nama lengkap" required="required"&gt;
                &lt;/div&gt;
                &lt;div class="form-group"&gt;
                    &lt;label class="font-weight-bold" for="username"&gt;Username&lt;/label&gt;
                    &lt;input type="text" class="form-control" name="username" placeholder="Masukkan username" required="required"&gt;
                &lt;/div&gt;
                &lt;div class="form-group"&gt;
                    &lt;label class="font-weight-bold" for="password"&gt;Password&lt;/label&gt;
                    &lt;input type="password" class="form-control" name="password" placeholder="Masukkan password" required="required"&gt;
                &lt;/div&gt;

                &lt;input type="submit" class="btn btn-primary" value="Simpan"&gt;
            &lt;/form&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;</pre
```

Form tambah petugas ini akan di proses pada function/method **petugas\_tambah\_aksi()** di controller **Admin.php**.

```
<form method="post" action="php echo base_url().'admin/petugas_tambah_aksi'; ?&gt;"&gt;</pre
```

Jadi kita buat lagi function baru di controller **Admin.php**. yaitu function **petugas\_tambah\_aksi()**.

## Application/controllers/Admin.php

```
function petugas_tambah_aksi(){
    $nama = $this->input->post('nama');
    $username = $this->input->post('username');
    $password = $this->input->post('password');

    $data = array(
        'nama' => $nama,
        'username' => $username,
        'password' => md5($password)
    );
}
```

```

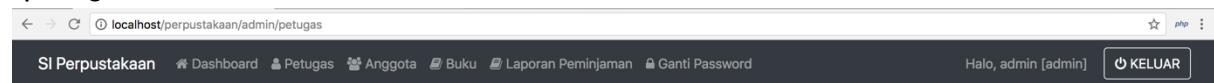
    // insert data ke database
    $this->m_data->insert_data($data,'petugas');

    // mengalihkan halaman ke halaman data petugas
    redirect(base_url().'admin/petugas');
}

```

Untuk penjelasannya sendiri sama saja seperti tutorial cara membuat CRUD dengan codeigniter yang telah di jelaskan pada tutorial codeigniter bagian dasar sebelumnya.

Sekarang kita akan mencoba menginput data petugas baru dengan klik pada tombol “petugas\_baru”.



Data Petugas			
No	Nama	Username	Opsi
			<a href="#">+ Petugas Baru</a>

localhost/perpustakaan/admin/petugas\_tambah

Kemudian akan di alihkan ke halaman tambah petugas baru. Lalu isikan data petugas baru.

Tambah Petugas Baru

Nama Lengkap  
Junaidi Alawi

Username  
junaidi

Password  
\*\*\*\*\*

**Simpan**

Disini saya coba menginput data petugas baru dengan nama “junaidi alawi”, username “junaidi”, dan passwordnya “petugas123”.

Dan klik simpan. Maka data petugas akan di simpan ke database. Dan otomatis muncul pada halaman petugas.

Data Petugas

No	Nama	Username	Opsi
1	Junaidi Alawi	junaidi	<b>Edit</b> <b>Hapus</b>

Oke, data petugas pun berhasil ditambahkan. Selanjutnya kita akan membuat fitur untuk mengedit data petugas.

## Update Data Petugas di Database

Untuk megredit data petugas, akan kita buat sebuah function baru lagi dengan nama **petugas\_edit()** di controller **Admin.php**. seperti yang telah kita arahkan pada tombol edit di data petugas.

```
<a href="<?php echo base_url().'admin/petugas_edit/'.$p->id; ?>" class="btn btn-sm btn-warning"><i class="fa fa-wrench"></i> Edit</a>
```

Buat function baru lagi dengan nama **petugas\_edit()** dalam controller **Admin.php**.

### Application/controllers/Admin.php

```
function petugas_edit($id){
    $where = array('id' => $id);
    // mengambil data dari database sesuai id
    $data['petugas'] = $this->m_data->edit_data($where, 'petugas')->result();
    $this->load->view('admin/v_header');
    $this->load->view('admin/v_petugas_edit', $data);
    $this->load->view('admin/v_footer');
}
```

Di sini kita menangkap data id petugas yang ingin diedit di dalam parameter function **petugas\_edit()**. Kemudian kita ambil data petugas yang memiliki id tersebut dari database, kemudian kita parsing ke view **petugas\_edit.php**. karena pada view **petugas\_edit.php** inilah akan kita buat form untuk mengedit data petugas.

### Application/views/admin/petugas\_edit.php

```
<div class="container">
    <div class="card">
        <div class="card-header text-center">
            <h4>Edit Petugas</h4>
        </div>
        <div class="card-body">
            <a href="<?php echo base_url().'admin/petugas' ?>" class='btn btn-sm btn-light btn-outline-dark pull-right'><i class="fa fa-arrow-left"></i> Kembali</a>
            <br/>
            <br/>

            <?php foreach($petugas as $p){ ?>
                <form method="post" action="<?php echo base_url().'admin/petugas_update'; ?>">
                    <div class="form-group">
                        <label class="font-weight-bold" for="nama">Nama Lengkap</label>
                        <input type="hidden" value="<?php echo $p->id; ?>" name="id">
                        <input type="text" class="form-control" name="nama" placeholder="Masukkan nama lengkap" required="required" value="<?php echo $p->nama; ?>">
                    </div>
                    <div class="form-group">
                        <label class="font-weight-bold" for="username">Username</label>
```

```

        <input type="text" class="form-control" name="username"
placeholder="Masukkan username" required="required" value="<?php echo $p->username;
?>">
    </div>
    <div class="form-group">
        <label class="font-weight-bold" for="password">Password</label>
        <input type="password" class="form-control" name="password"
placeholder="Masukkan password">
        <small class="form-text text-muted">Kosongkan jika tidak ingin mengubah
password.</small>
    </div>

    <input type="submit" class="btn btn-primary" value="Simpan">
</form>
<?php } ?>

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

```

Form edit data petugas ini akan di proses pada function **petugas\_update()** dalam controller **Admin.php** pada saat di submit.

```
<form method="post" action="<?php echo base_url().'admin/petugas_update'; ?>">
```

Jadi buat lagi sebuah function baru dengan nama **petugas\_update()** pada controller **Admin.php**.

### Application/controllers/Admin.php

```

function petugas_update(){
    $id = $this->input->post('id');
    $nama = $this->input->post('nama');
    $username = $this->input->post('username');
    $password = $this->input->post('password');

    $where = array(
        'id' => $id
    );

    // cek apakah form password di isi atau tidak
    if($password=="") {
        $data = array(
            'nama' => $nama,
            'username' => $username
        );

        // update data ke database
        $this->m_data->update_data($where,$data,'petugas');
    }else{
        $data = array(
            'nama' => $nama,
            'username' => $username,
            'password' => md5($password)
        );

        // update data ke database
        $this->m_data->update_data($where,$data,'petugas');
    }
}

```

```
}

// mengalihkan halaman ke halaman data petugas
redirect(base_url().'admin/petugas');
}
```

Penjelasan tentang update data di database dengan codeigniter, bisa teman-teman baca kembali pada pembahasan membuat crud di tutorial codeigniter bagian dasar sebelumnya. Karena intinya sama saja.

Sekarang kita akan mencoba mengubah/mengedit data petugas “Junaidi Alawi” ke “Moh Junaidi”.

Klik pada tombol edit.

The screenshot shows a web application titled "Data Petugas". At the top right, there is a green button labeled "+ Petugas Baru". Below it, a table has one row with data:

No	Nama	Username	Opsi
1	Junaidi Alawi	junaidi	Edit  Hapus

A red arrow originates from the text "Klik pada tombol edit." and points directly at the "Edit" button in the "Opsi" column of the first row.

Maka akan dialihkan ke halaman edit petugas.

localhost/perpustakaan/admin/petugas\_edit/1

SI Perpustakaan

Halo, admin [admin] KELUAR

Edit Petugas

Kembali

Nama Lengkap  
Junaidi Alawi

Username  
junaidi

Password  
Masukkan password  
Kosongkan jika tidak ingin mengubah password.

Simpan

Lalu ubah nama petugas dari “Junaidi Alawi” menjadi “Moh Junaidi”. Dan klik simpan.

localhost/perpustakaan/admin/petugas

SI Perpustakaan

Halo, admin [admin] KELUAR

Data Petugas

+ Petugas Baru

No	Nama	Username	Opsi
1	Moh Junaidi	junaidi	Edit  Hapus

Dan data pun berhasil diedit. Dan yang terakhir kita akan membuat hapus data dari database ketika tombol hapus diklik.

## Menghapus Data Petugas dari Database

Sesuai dengan link yang sudah kita tentukan pada tombol hapus di table data mobil seperti di atas.

```
<a href="php echo base_url().'admin/petugas_hapus/'.$p-&gt;id; ?&gt;"<br/class="btn btn-sm btn-danger"><i class="fa fa-trash"></i> Hapus</a>
```

Untuk aksi dari penghapusan data petugas kita akan membuatnya di function/method petugas\_hapus() dalam controller Admin.php.

Buat sebuah method baru dengan nama petugas\_hapus() di controller Admin.php.

### Application/controllers/Admin.php

```
function petugas_hapus($id){  
    $where = array(  
        'id' => $id  
    );  
  
    // menghapus data petugas dari database sesuai id  
    $this->m_data->delete_data($where,'petugas');  
  
    // mengalihkan halaman ke halaman data petugas  
    redirect(base_url().'admin/petugas');  
}  
// akhir CRUD petugas
```

Untuk lebih detail tentang menghapus data dari database menggunakan codeigniter, silahkan teman-teman baca kembali pada tutorial CRUD sebelumnya di buku ini. Yang sudah dijelaskan secara detail.

Sekarang coba klik tombol hapus pada mobil yang ingin dihapus.

localhost/perpustakaan/admin/petugas

SI Perpustakaan

Halo, admin [admin] KELUAR

Data Petugas

No	Nama	Username	Opsi
1	Moh Junaidi	junaidi	<a href="#">Edit</a> <a href="#">Hapus</a>

Dan petugas pun berhasil di hapus.

localhost/perpustakaan/admin/petugas

SI Perpustakaan

Halo, admin [admin] KELUAR

Data Petugas

No	Nama	Username	Opsi
----	------	----------	------

proses hapus data petugas pun berhasil dan selesai.

### Ingat !

CRUD adalah singkatan dari *Create, Read, Update and Delete*. Istilah CRUD sering di gunakan untuk gambaran proses input data ke database, menampilkan data ke database, mengedit data ke database dan menghapus data dari database.

Setelah selesai membuat CRUD Petugas, kita stop dulu sampai di sini tentang admin nya. Akan kita lanjut ke bagian petugas. Karena sampai di sini admin telah bisa mengelola data petugas. Tinggal selanjutnya kita buat login untuk petugas. Karena operasi yang lainnya hanya bisa di lakukan oleh petugas. Seperti mengelola data buku dan transaksi peminjaman (seperti yang telah dijelaskan pada perancangan sistem pada sub judul sebelumnya).

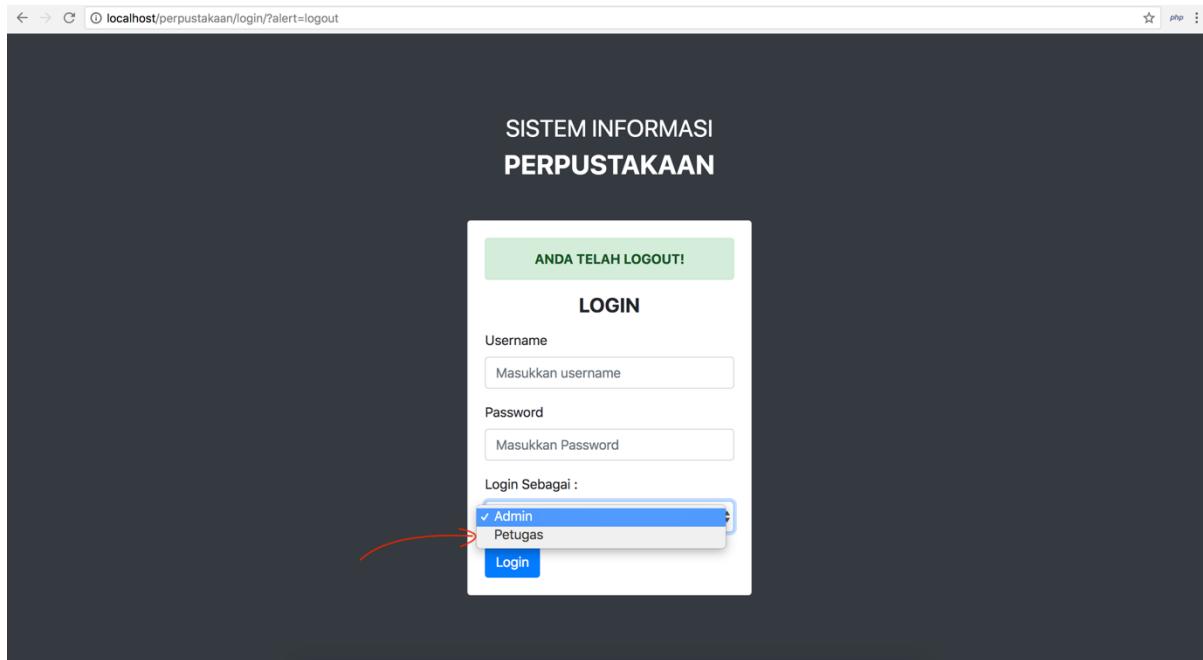
Oh ya jangan lupa untuk menambahkan lagi beberapa data petugas. Data petugas yang ditambahkan akan kita gunakan untuk login pada saat membuat halaman petugas. Di sini saya menambahkan lagi petugas dengan nama “*Jamaludin Ishak*”, username “*jamaludin*” dan passwordnya “*petugas123*”.

The screenshot shows a web browser window with the URL `localhost/perpustakaan/admin/petugas_tambah`. The page title is "SI Perpustakaan". The navigation menu includes Dashboard, Petugas, Anggota, Buku, Laporan Peminjaman, Ganti Password, and a log-in message "Halo, admin [admin]". There is also a "KELUAR" (Logout) button. The main content area is titled "Tambah Petugas Baru" (Add New Staff). It contains three input fields: "Nama Lengkap" (Full Name) with value "Jamaludin Ishak", "Username" with value "jamaludin", and "Password" with value "\*\*\*\*\*". A blue "Simpan" (Save) button is at the bottom. A red box highlights the input fields.

dan klik simpan.

# Membuat Halaman Dashboard Petugas

Pada saat membuat login untuk admin sebelumnya, kita telah membuat juga login untuk petugas.



Hanya saja pada saat petugas berhasil login, akan dialihkan ke controller **"Petugas.php"**. oleh karena itu kita akan membuat sebuah controller baru lagi dengan nama Petugas.php. dimana pada controller Petugas.php ini kita akan membuat halaman dashboard untuk petugas dan segala operasi yang bisa dilakukan oleh petugas pada sistem informasi perpustakaan ini.

Buat sebuah controller baru dengan nama **Petugas.php**.

## Application/controllers/Petugas.php

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

class Petugas extends CI_Controller {

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

        // cek session yang login, jika session status tidak sama dengan
        session petugas_login,maka halaman akan di alihkan kembali ke halaman login.
        if($this->session->userdata('status')!="petugas_login"){
            redirect(base_url().'login?alert=belum_login');
        }
    }

    function index(){
        $this->load->view('petugas/v_header');
        $this->load->view('petugas/v_index');
```

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

Pada function \_\_construct() tetap kita harus membuat pengecekan session login. Agar pengguna tidak dapat mengakses halaman dashboard petugas jika belum melakukan login. Karena bayangan saja jika aplikasi ini dapat diakses dengan bebas oleh semua orang tanpa harus melakukan login.

```
// cek session yang login, jika session status tidak sama dengan session
petugas_login,maka halaman akan di alihkan kembali ke halaman login.
if($this->session->userdata('status')!="petugas_login"){
    redirect(base_url().'login?alert=belum_login');
}
```

Sama seperti saat kita membuat halaman dashboard untuk admin. Kita membuat halaman dashboardnya pada function/method index(). Karena function index() yang pertama kali dijalankan jika kita mengakses sebuah controller.

Oleh sebab itu di controller petugas kita juga akan membuat halaman dashboard petugasnya pada function index.php seperti pada syntax di atas.

```
function index(){
    $this->load->view('petugas/v_header');
    $this->load->view('petugas/v_index');
    $this->load->view('petugas/v_footer');
}
```

Kemudian kita membuat view untuk petugas di dalam folder petugas. Semua view untuk petugas akan kita kumpulkan atau kita buat di dalam folder “petugas” dalam folder views. Agar struktur view project kita lebih rapi dan tersusun.

Buat sebuah folder dengan nama “**petugas**” di dalam folder “views”. Dan buat 3 buah view baru di dalam folder petugas. Yaitu v\_header.php, v\_index.php dan v\_footer.php.

### **Application/views/petugas/v\_header.php**

```
<!DOCTYPE html>
<html>
<head>
    <title>Petugas - Sistem Informasi Perpustakaan MALAS NGODING</title>
    <!-- css bootstrap -->
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url().'assets/css/bootstrap.css' ?>">

    <!-- css datatables -->
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url().'assets/DataTables/datatables.css' ?>">
```

```

<!-- icon font awesome -->
<link rel="stylesheet" type="text/css" href="php echo base_url().'assets/css/fontawesome.css' ?&gt;"&gt;

&lt;!-- jquery dan bootstrap js --&gt;
&lt;script type="text/javascript" src="<?php echo base_url().'assets/js/jquery.js' ?&gt;"&gt;&lt;/script&gt;
&lt;script type="text/javascript" src="<?php echo base_url().'assets/js/bootstrap.js' ?&gt;"&gt;&lt;/script&gt;

&lt;!-- js datatables --&gt;
&lt;script type="text/javascript" src="<?php echo base_url().'assets/DataTables/datatables.js' ?&gt;"&gt;&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;

&lt;nav class="navbar navbar-expand-lg navbar-dark bg-dark"&gt;
    &lt;div class="container-fluid"&gt;
        &lt;a class="navbar-brand" href="<?php echo base_url().'petugas'; ?&gt;"SI Perpustakaan</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarText" aria-controls="navbarText" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarText">
            <ul class="navbar-nav mr-auto">
                <li class="nav-item">
                    <a class="nav-link" href="php echo base_url().'petugas'; ?&gt;"<i class="fa fa-home"></i> Dashboard</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="php echo base_url().'petugas/anggota'; ?&gt;"<i class="fa fa-users"></i> Anggota</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="php echo base_url().'petugas/buku'; ?&gt;"<i class="fa fa-book"></i> Buku</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="php echo base_url().'petugas/peminjaman'; ?&gt;"<i class="fa fa-book"></i> Peminjaman</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="php echo base_url().'petugas/peminjaman_laporan'; ?&gt;"<i class="fa fa-book"></i> Laporan Peminjaman</a>
                </li>
                <li class="nav-item">
                    <a href="php echo base_url().'petugas/ganti_password' ?&gt;" class="nav-link"&gt;&lt;i class="fa fa-lock"&gt;&lt;/i&gt; Ganti Password&lt;/a&gt;
                &lt;/li&gt;
            &lt;/ul&gt;

            &lt;span class="navbar-text mr-3 text-center"&gt;
                Halo, &lt;?php echo $this-&gt;session-&gt;userdata('username'); ?&gt; [Petugas]
            &lt;/span&gt;

            &lt;a href="<?php echo base_url().'petugas/logout' ?&gt;" class="btn btn-outline-light ml-1"&gt;&lt;i class="fa fa-power-off"&gt;&lt;/i&gt; KELUAR&lt;/a&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/nav&gt;
</pre

```

```
<br/>
<br/>
```

## Application/views/petugas/v\_index.php

```
<div class="container">
    <div class="jumbotron text-center">
        <div class="col-sm-8 mx-auto">
            <h1>Selamat datang!</h1>
            <p>Ini merupakan contoh sistem informasi perpustakaan hasil dari tutorial
            <b>ebook tutorial codeigniter lengkap dengan studi kasus membuat sistem informasi
            perpustakaan</b>.</p>
            <p>
                Anda telah login sebagai <b><?php echo $this->session->userdata('nama') ;
            ?></b> [petugas].
            </p>
        </div>
    </div>

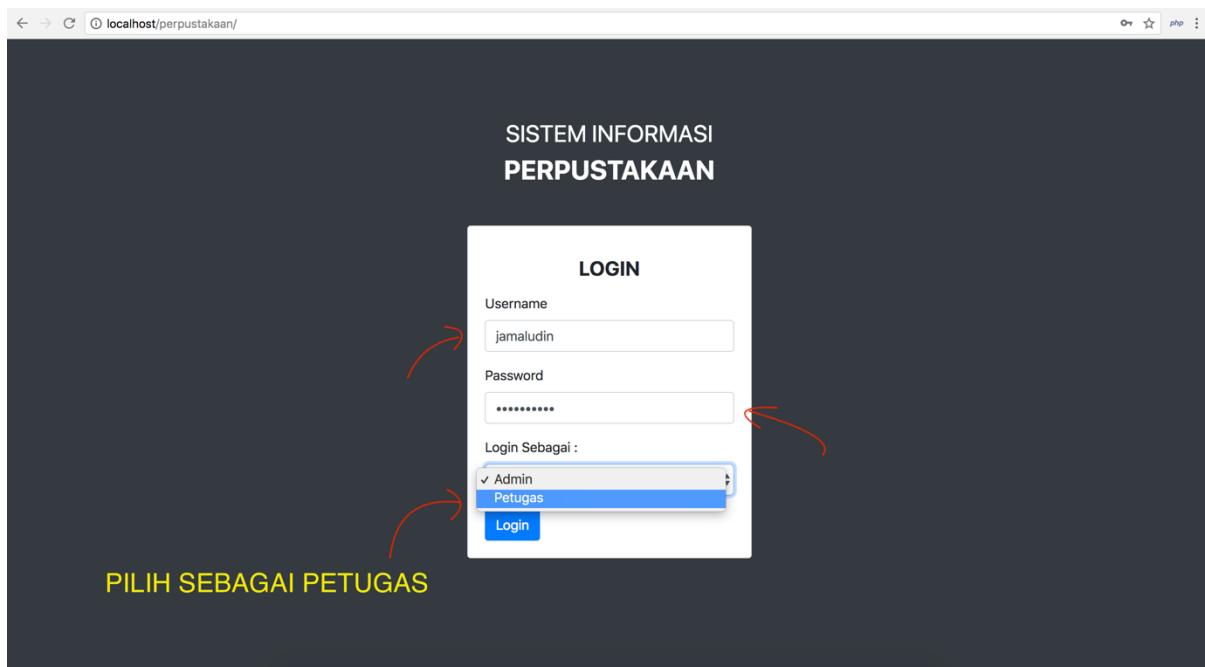
    <div class="card">
        <div class="card-header">
            Dashboard
        </div>
        <div class="card-body">
            Halo
        </div>
    </div>
</div>
```

## Application/views/petugas/v\_footer.php

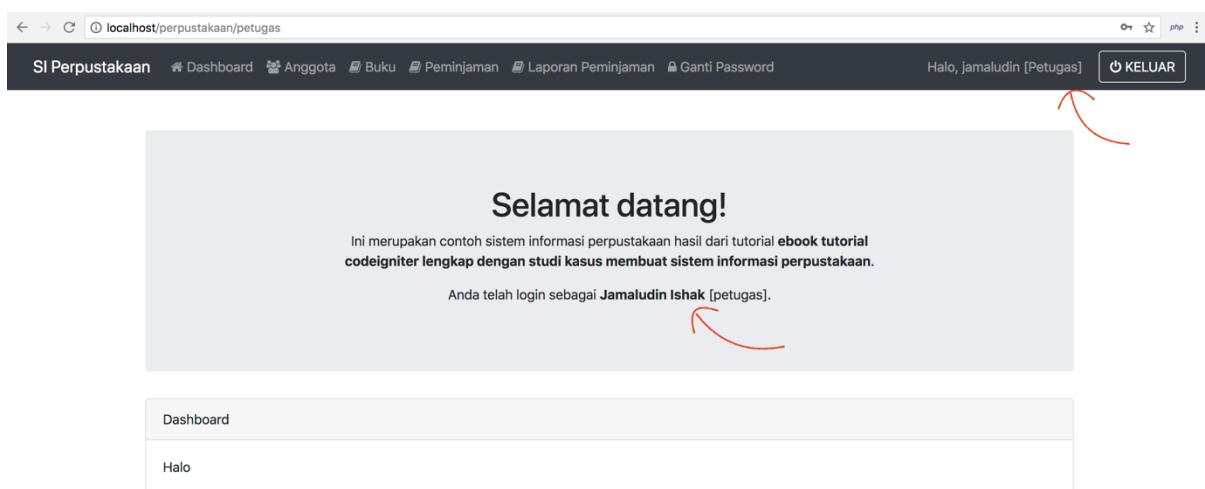
```
<script type="text/javascript">
    $(document).ready(function() {
        $('.table-datatable').DataTable();
    });
</script>

</body>
</html>
```

Sekarang langsung saja kita coba jalankan pada browser. Untuk melakukan login sebagai petugas dengan menggunakan akun petugas yang sudah kita buat melalui akun admin pada sub judul sebelumnya. Yaitu login sebagai petugas dengan username “jamaludin” dan password “petugas123”.



jangan lupa memilih login sebagai "Petugas". Dan klik login.



maka login pun berhasil. Kita telah berhasil sebagai petugas. Seperti yang terlihat pada gambar di atas. Selanjutnya kita akan membuat fungsi-fungsi yang ada pada menu petugas. Baru kemudian kita akan melanjutkan fungsi-fungsi yang ada pada admin yang sebelumnya telah kita tinggalkan sebentar.

## Membuat Logout Petugas

Seperti admin, petugas juga harus bisa logout dari sistem informasi perpustakaan yang akan kita buat ini. Caranya sama saja dengan membuat logout pada admin. Yaitu dengan membuat sebuah function logout pada controller.

Perhatikan pada link menu “KELUAR”. Saat diklik, maka akan kita alihkan ke function logout pada controller Petugas.php.

```
<a href="<?php echo base_url().'petugas/logout' ?>" class="btn btn-outline-light ml-1"><i class="fa fa-power-off"></i> KELUAR</a>
```

Buat sebuah function baru pada controller Petugas.php. yaitu function logout().

```
function logout(){
    $this->session->sess_destroy();
    redirect(base_url().'login/?alert=logout');
}
```

Pada saat tombol/link keluar diklik, maka session akan dihapus dengan perintah `$this->session->sess_destroy()`. Dan halaman akan dialihkan kembali ke halaman login.

# Membuat Ganti Password Petugas

Cara membuat ganti password pada petugas juga sama seperti yang telah dijelaskan sebelumnya pada cara membuat ganti password pada admin. Perhatikan pada menu ganti password yang telah kita buat sebelumnya. Jika diklik maka akan dialihkan ke function ganti\_password() pada controller petugas.

```
<a href="=base_url('petugas/ganti_password') ?&gt;" class="nav-link"&gt;&lt;i class="fa fa-lock"&gt;&lt;/i&gt; Ganti Password&lt;/a&gt;</pre
```

Jadi buat sebuah function baru dengan nama ganti\_password() pada controller Petugas.php. dan buat function ganti\_password\_aksi() juga sebagai function yang memproses form ganti password (sama seperti penjelasan sebelumnya).

## Application/controllers/Petugas.php

```
function ganti_password(){
    $this->load->view('petugas/v_header');
    $this->load->view('petugas/v_ganti_password');
    $this->load->view('petugas/v_footer');
}

function ganti_password_aksi(){
    $baru = $this->input->post('password_baru');
    $ulang = $this->input->post('password_ulang');

    $this->form_validation->set_rules('password_baru', 'Password Baru', 'required|matches[password_ulang]');
    $this->form_validation->set_rules('password_ulang', 'Ulangi Password', 'required');

    if($this->form_validation->run() != false){
        $id = $this->session->userdata('id');

        $where = array('id' => $id);

        $data = array('password' => md5($baru));

        $this->m_data->update_data($where, $data, 'petugas');

        redirect(base_url() . 'petugas/ganti_password/?alert=sukses');

    }else{
        $this->load->view('petugas/v_header');
        $this->load->view('petugas/v_ganti_password');
        $this->load->view('petugas/v_footer');
    }
}
```

Form ganti password nya akan kita buat pada view v\_ganti\_password dalam folder "petugas". Sekarang buat sebuah file view baru dengan nama v\_ganti\_password.php.

## Application/views/petugas/v\_ganti\_password.php

```

<div class="container">

    <div class="row justify-content-md-center">
        <div class="col-md-6">
            <div class="card">
                <div class="card-header text-center">
                    <h4>Ganti Password</h4>
                </div>
                <div class="card-body">

                    <?php
                    if(isset($_GET['alert'])) {
                        if($_GET['alert']=="sukses") {
                            echo      "<div      class='alert      alert-success'>Password      berhasil
diganti.</div>";
                        }
                    }
                    ?>

                    <?php echo validation_errors(); ?>
                    <form          method="post"          action="<?php
base_url().'petugas/ganti_password_aksi'; ?>">
                        <div class="form-group">
                            <label      class="font-weight-bold"      for="password_baru">Password
Baru</label>
                            <input      type="password"      class="form-control"      name="password_baru"
placeholder="Masukkan password baru">
                        </div>
                        <div class="form-group">
                            <label      class="font-weight-bold"      for="password_ulang">Ulangi Password
Baru</label>
                            <input      type="password"      class="form-control"      name="password_ulang"
placeholder="Ulangi password baru">
                        </div>

                        <input type="submit" class="btn btn-primary" value="Ubah Password">
                    </form>

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

```

Sampai disini selesai sudah pembuatan fitur ganti password untuk petugas. Teman-teman bisa mencoba nya langsung.

# Membuat CRUD Anggota Pada Petugas

Setelah selesai membuat halaman dashboard petugas, membuat logout dan ganti password untuk petugas, sekarang kita akan membuat CRUD data anggota. Sesuai dengan perancangan dan perencanaan sistem informasi perpustakaan yang telah kita buat sebelumnya. Bawa petugas dapat mengelola data anggota. Oleh sebab itu kita akan membuat CRUD data anggota pada halaman petugas.

## Menampilkan Data Anggota Dari Database

Untuk menampilkan data anggota dari database, kita akan membuat function atau method baru pada controller Petugas.php. function yang akan kita buat adalah **anggota()**.

### Application/controllers/Petugas.php

```
// crud anggota
function anggota(){
    // mengambil data dari database
    $data['anggota'] = $this->m_data->get_data('anggota')->result();
    $this->load->view('petugas/v_header');
    $this->load->view('petugas/v_anggota',$data);
    $this->load->view('petugas/v_footer');
}
```

Sama saja seperti cara *menampilkan data petugas dari database* yang telah kita bahas sebelumnya pada saat membuat CRUD data petugas pada halaman admin.

Disini pertama kita mengambil data kostumer dari table 'kostumer'.

```
$data['anggota'] = $this->m_data->get_data('anggota')->result();
```

Kemudian kita masukkan ke variabel array data. Dan kemudian kita parsing ke view v\_anggota.php.

Selanjutnya kita buat view baru lagi dengan nama v\_anggota.php pada folder petugas.

### Application/views/petugas/anggota.php

```
<div class="container">
<div class="card">
    <div class="card-header text-center">
        <h4>Data Anggota</h4>
    </div>
    <div class="card-body">
```

```

<a href="=base_url().'petugas/anggota_tambah' ?" class='btn btn-sm
btn-success pull-right'><i class="fa fa-plus"></i> Anggota Baru</a>
<br/>
<br/>

<div class="table-responsive">
    <table class="table table-bordered table-striped table-hover table-
datatable">
        <thead>
            <tr>
                <th width="1%">No</th>
                <th>Nama</th>
                <th>NIK</th>
                <th>Alamat</th>
                <th width="28%">Opsi</th>
            </tr>
        </thead>
        <tbody>
            <?php
            $no = 1;
            foreach($anggota as $a) {
                ?>
                <tr>
                    <td><?php echo $no++; ?></td>
                    <td><?php echo $a->nama; ?></td>
                    <td><?php echo $a->nik; ?></td>
                    <td><?php echo $a->alamat; ?></td>
                    <td>
                        <a target="_blank" href="=base_url().'petugas/anggota_kartu/'.$a-&gt;id; ?" class="btn btn-sm btn-primary"><i class="fa fa-address-card"></i> Cetak Kartu</a>
                        <a href="=base_url().'petugas/anggota_edit/'.$a-&gt;id; ?" class="btn btn-sm btn-warning"><i class="fa fa-wrench"></i> Edit</a>
                        <a href="=base_url().'petugas/anggota_hapus/'.$a-&gt;id; ?" class="btn btn-sm btn-danger"><i class="fa fa-trash"></i> Hapus</a>
                    </td>
                </tr>
            <?php
            }
            ?>
        </tbody>
    </table>
</div>

</div>
</div>

```

Data anggota yang kita parsing dari controller tadi kita tangkap dan menggunakan function `foreach()` untuk di tampilkan secara perulangan.

```

<table class="table table-bordered table-striped table-hover table-datatable">
    <thead>
        <tr>
            <th width="1%">No</th>
            <th>Nama</th>
            <th>NIK</th>
            <th>Alamat</th>
            <th width="28%">Opsi</th>
        </tr>
    </thead>
    <tbody>

```

```

<?php
$no = 1;
foreach($anggota as $a) {
?>
<tr>
<td><?php echo $no++; ?></td>
<td><?php echo $a->nama; ?></td>
<td><?php echo $a->nik; ?></td>
<td><?php echo $a->alamat; ?></td>
<td>
<a target="_blank" href="<?php echo
base_url().'petugas/anggota_kartu/'.$a->id; ?>" class="btn btn-sm btn-primary"><i
class="fa fa-address-card"></i> Cetak Kartu</a>
<a href="<?php echo base_url().'petugas/anggota_edit/'.$a->id; ?>" class="btn btn-sm btn-warning"><i class="fa fa-wrench"></i> Edit</a>
<a href="<?php echo base_url().'petugas/anggota_hapus/'.$a->id; ?>" class="btn btn-sm btn-danger"><i class="fa fa-trash"></i> Hapus</a>
</td>
</tr>
<?php
}
?>
</tbody>
</table>

```

Sekarang coba klik pada menu anggota. Maka data anggota pun akan muncul.

localhost/pertugasan/anggota

SI Perpustakaan

Dashboard Anggota Buku Peminjaman Laporan Peminjaman Ganti Password

Halo, jamaludin [Petugas] KELUAR

Data Anggota

Show 10 entries Search:

No	Nama	NIK	Alamat	Opsi
No data available in table				

Showing 0 to 0 of 0 entries Previous Next

+ Anggota Baru

Seperti yang terlihat pada gambar di atas, kita telah berhasil membuat tabel yang menampilkan data anggota dari database. Karena kita belum memiliki data anggota pada table anggota, maka data masih kosong. Selanjutnya kita akan membuat input data anggota ke database.

## Input Data Anggota Ke Database

Dan seperti yang telah teman-teman lihat pada syntax view v\_anggota.php di atas. kita telah membuat tombol untuk mengalihkan halaman ke method anggota\_tambah() di controller Petugas.php, karena pada method tersebut nantinya akan kita buat halaman untuk input data anggota ke database. Yaitu tombol “Anggota Baru”.

```
<a href=<?php echo base_url().'petugas/anggota_tambah' ?>" class='btn btn-sm btn-success pull-right'><i class="fa fa-plus"></i> Anggota Baru</a>
```

Maka dari itu, akan kita buat method baru yaitu *anggota\_tambah()* pada controller Petugas.php.

### Application/controllers/Petugas.php

```
function anggota_tambah(){
    $this->load->view('petugas/v_header');
    $this->load->view('petugas/v_anggota_tambah');
    $this->load->view('petugas/v_footer');
}
```

Pada method atau function *anggota\_tambah()* ini kita hanya menampilkan view v\_anggota\_tambah.php. pada view v\_anggota\_tambah.php ini kita akan membuat form input data anggota baru.

Buat view baru dalam folder petugas. Dengan nama v\_anggota\_tambah.php

### Application/views/petugas/v\_anggota\_tambah.php

```
<div class="container">
    <div class="card">
        <div class="card-header text-center">
            <h4>Tambah Anggota Baru</h4>
        </div>
        <div class="card-body">
            <a href=<?php echo base_url().'petugas/anggota' ?>" class='btn btn-sm btn-light btn-outline-dark pull-right'><i class="fa fa-arrow-left"></i> Kembali</a>
            <br/>
            <br/>

            <form method="post" action=<?php echo base_url().'petugas/anggota_tambah_aksi'; ?>>
                <div class="form-group">
                    <label class="font-weight-bold" for="nama">Nama Lengkap</label>
                    <input type="text" class="form-control" name="nama" placeholder="Masukkan nama lengkap" required="required">
                </div>
                <div class="form-group">
                    <label class="font-weight-bold" for="nik">nik</label>
                    <input type="number" class="form-control" name="nik" placeholder="Masukkan nik" required="required">
                </div>
            </form>
        </div>
    </div>
</div>
```

```

<div class="form-group">
    <label class="font-weight-bold" for="alamat">Alamat</label>
    <textarea class="form-control" name="alamat" required="required" placeholder="Masukkan alamat"></textarea>
</div>

    <input type="submit" class="btn btn-primary" value="Simpan">
</form>

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

```

The screenshot shows a web application interface. At the top, there's a navigation bar with links like 'Dashboard', 'Anggota', 'Buku', 'Peminjaman', 'Laporan Peminjaman', 'Ganti Password', and a user profile 'Halo, jamaludin [Petugas]'. Below the navigation is a button labeled 'KELUAR'. The main content area has a title 'Tambah Anggota Baru'. It contains three text input fields labeled 'Nama Lengkap', 'nik', and 'Alamat', each with a placeholder 'Masukkan ...'. A blue 'Simpan' button is at the bottom.

Form input data anggota ini akan di proses pada method `anggota_tambah_aksi()` sesuai dengan yang sudah kita setting pada form di atas.

```
<form method="post" action="php echo base_url() . 'petugas/anggota_tambah_aksi'; ?&gt;"&gt;</pre

```

Maka sekarang buat sebuah method baru lagi dengan nama `anggota_tambah_aksi()` pada controller Petugas.php.

### Application/controllers/Petugas.php

```

function anggota_tambah_aksi(){
    $nama = $this->input->post('nama');
    $nik = $this->input->post('nik');
    $alamat = $this->input->post('alamat');

    $data = array(
        'nama' => $nama,
        'nik' => $nik,
        'alamat' => $alamat
    )
}

```

```

);
// insert data ke database
$this->m_data->insert_data($data,'anggota');

// mengalihkan halaman ke halaman data anggota
redirect(base_url().'petugas/anggota');
}

```

Untuk proses aksi dari penginputan data anggota ini sama saja intinya seperti proses aksi penginputan data pada proses CRUD yang telah kita bahas berulang-ulang kali pada pembahasan sebelumnya. Setelah proses input data selesai, halaman akan langsung kita alihkan ke function anggota() di controller Petugas.php.

```
redirect(base_url().'petugas/anggota');
```

Sekarang akan kita coba menginput data anggota. Silahkan klik pada tombol “kostumer baru”. Lalu isi data kostumer dan klik “simpan”.

The screenshot shows a web application interface. At the top, there's a header bar with links for Dashboard, Anggota, Buku, Peminjaman, Laporan Peminjaman, Ganti Password, and a sign-in message "Halo, jamaludin [Petugas]". Below the header is a navigation link "localhost/perpusatakan/petugas/anggota\_tambah". The main content area has a title "Tambah Anggota Baru". It contains three input fields: "Nama Lengkap" with value "Rahmiati", "nik" with value "2832938293829", and "Alamat" with value "Jalan tembakau cipinang indah no.181". A blue "Simpan" button is located at the bottom left of the form. There's also a "Kembali" button at the top right of the form area.

Setelah di klik pada tombol “simpan”. Maka data akan tersimpan.

<p>Dan data anggota berhasil di simpan dan berhasil tampil pada halaman anggota.</p>	

## Update Data Anggota Di Database

Untuk membuat edit dan update data anggota, kita akan membuat function anggota\_edit() di controller Petugas.php. seperti yang sudah kita tetapkan pada tombol edit di tabel yang menampilkan data anggota seperti di atas.

```
<a href="php echo base_url().'petugas/anggota_edit/'.$a-&gt;id; ?&gt;" class="btn btn-sm btn-warning"&gt;&lt;i class="fa fa-wrench"&gt;&lt;/i&gt; Edit&lt;/a&gt;</pre

```

Buat sebuah method baru dengan nama *anggota\_edit()*.

### Application/controllers/Petugas.php

```
function anggota_edit($id){
    $where = array('id' => $id);
    // mengambil data dari database sesuai id
    $data['anggota'] = $this->m_data->edit_data($where, 'anggota')->result();
    $this->load->view('petugas/v_header');
    $this->load->view('petugas/v_anggota_edit', $data);
    $this->load->view('petugas/v_footer');
}
```

Di sini kita menangkap data id yang dikirimkan dari url.Kemudian kita ambil data anggota yang memiliki id sesuai dengan id yang kita tangkap. Kemudian kita parsing data nya ke view v\_anggota\_edit.php di folder petugas.

Buat view baru dengan nama **v\_anggota\_edit.php** di dalam folder petugas.

### Application/views/petugas/v\_anggota\_edit.php

```
<div class="container">
    <div class="card">
        <div class="card-header text-center">
            <h4>Edit Anggota</h4>
        </div>
        <div class="card-body">
            <a href="php echo base_url().'petugas/anggota' ?&gt;" class='btn btn-sm btn-light btn-outline-dark pull-right'&gt;&lt;i class="fa fa-arrow-left"&gt;&lt;/i&gt; Kembali&lt;/a&gt;
            &lt;br/&gt;
            &lt;br/&gt;

            &lt;?php foreach($anggota as $a){ ?&gt;
                &lt;form method="post" action="<?php echo base_url().'petugas/anggota_update'; ?&gt;"&gt;
                    &lt;div class="form-group"&gt;
                        &lt;label class="font-weight-bold" for="nama"&gt;Nama Lengkap&lt;/label&gt;
                        &lt;input type="hidden" value="<?php echo $a-&gt;id; ?&gt;" name="id"&gt;
                        &lt;input type="text" class="form-control" name="nama"
placeholder="Masukkan nama lengkap" required="required" value="<?php echo $a-&gt;nama; ?&gt;"&gt;
                    &lt;/div&gt;
                    &lt;div class="form-group"&gt;
                        &lt;label class="font-weight-bold" for="nik"&gt;nik&lt;/label&gt;
                        &lt;input type="number" class="form-control" name="nik"
placeholder="Masukkan nik" required="required" value="<?php echo $a-&gt;nik; ?&gt;"&gt;
                    &lt;/div&gt;
                    &lt;div class="form-group"&gt;
                        &lt;label class="font-weight-bold" for="alamat"&gt;Alamat&lt;/label&gt;
                        &lt;textarea class="form-control" name="alamat" required="required"
placeholder="Masukkan alamat"<?php echo $a->alamat; ?></textarea>
                    </div>

                    <input type="submit" class="btn btn-primary" value="Simpan">
                </form>
            <?php } ?>
        </div>
    </div>
</div>
```

Karena form edit kostumer ini akan di proses pada function *anggota\_update()* di controller Petugas.php.

```
<form method="post" action="php echo base_url().'petugas/anggota_update'; ?&gt;"&gt;</pre
```

maka kita buat sebuah function/method baru lagi di controller Petugas.php dengan nama *anggota\_update()*.

### Application/controllers/Petugas.php

```
function anggota_update(){
    $id = $this->input->post('id');
    $nama = $this->input->post('nama');
```

```

$nik = $this->input->post('nik');
$alamat = $this->input->post('alamat');

$where = array(
    'id' => $id
);

$data = array(
    'nama' => $nama,
    'nik' => $nik,
    'alamat' => $alamat
);

// update data ke database
$this->m_data->update_data($where,$data,'anggota');

// mengalihkan halaman ke halaman data anggota
redirect(base_url().'petugas/anggota');
}

```

Dan coba sekarang kita klik pada tombol edit untuk mengubah data anggota.

Maka kita akan dialihkan ke form edit anggota.

The screenshot shows a web application interface. At the top, there's a header bar with links for Dashboard, Anggota, Buku, Peminjaman, Laporan Peminjaman, Ganti Password, and a sign-out button labeled 'KELUAR'. Below the header, the main content area has a title 'Edit Anggota'. The form contains three input fields: 'Nama Lengkap' with value 'Rahmiati', 'nik' with value '2147483647', and 'Alamat' with value 'Jalan tembakau cipinang indah no.181'. A blue 'Simpan' button is located at the bottom left of the form.

Di sini kita akan mencoba mengubah data nik dan alamat dari anggota.

localhost/perpustakaan/petugas/anggota\_edit/

SI Perpustakaan

Halo, jamaludin [Petugas] KELUAR

Edit Anggota

Kembali

Nama Lengkap

Rahmiati

nik

1236447464746

Alamat

Jalan Merpati putih no.181, Jakarta

Simpan

Di sini saya mencoba merubah data nik dan alamat dari anggota yang bernama rahmiati. Lalu klik simpan.

localhost/perpustakaan/petugas/anggota

SI Perpustakaan

Halo, jamaludin [Petugas] KELUAR

Data Anggota

+ Anggota Baru

Show 10 entries

Search:

No	Nama	NIK	Alamat	Opsi
1	Rahmiati	2147483647	Jalan Merpati putih no.181, Jakarta	<a href="#">Cetak Kartu</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Showing 1 to 1 of 1 entries

Previous 1 Next

Dan data anggota pun berhasil di update.

## Menghapus Data Anggota dari Database

Dan terakhir untuk crud data anggota, kita akan membuat fungsi untuk hapus data kostumer.

Coba teman-teeman perhatikan pada tombol hapus pada table data anggota. Saat tombol hapus diklik maka akan di alihkan ke function anggota\_hapus() di controller Petugas sambil membawa id anggota yang akan di hapus.

```
<a href="php echo base_url().'petugas/anggota_hapus/'.$a-&gt;id; ?&gt;" class="btn btn-sm btn-danger"&gt;&lt;i class="fa fa-trash"&gt;&lt;/i&gt; Hapus&lt;/a&gt;</pre
```

Jadi aksi penghapusan data anggota ini akan kita buat di function/method *anggota\_hapus()* di controller Petugas.

Maka kita harus membuat method baru di controller admin. Yaitu method *anggota\_hapus()*.

### Application/controllers/Petugas.php

```
function anggota_hapus($id) {
    $where = array(
        'id' => $id
    );

    // menghapus data anggota dari database sesuai id
    $this->m_data->delete_data($where, 'anggota');

    // mengalihkan halaman ke halaman data anggota
    redirect(base_url().'petugas/anggota');
}
```

Di sini kita tanggap id anggota yang akan di hapus, kemudian kita masukkan ke variabel array where. Untuk kita seleksi data yang akan kita hapus. Setelah data di hapus maka halaman akan dialihkan ke method anggota() kembali.

Sebenarnya cara menghapus data anggota masih sama saja dengan cara menghapus data pada proses CRUD lainnya yang telah kita bahas sebelumnya.

Coba klik tombol hapus.

The screenshot shows a table with one row of data. The columns are labeled 'No', 'Nama', 'NIK', 'Alamat', and 'Opsi'. The first row contains the values: 1, Rahmiati, 2147483647, Jalan Merpati putih no.181, Jakarta, and a button labeled 'Hapus' with a red arrow pointing to it. Below the table, it says 'Showing 1 to 1 of 1 entries'.

No	Nama	NIK	Alamat	Opsi
1	Rahmiati	2147483647	Jalan Merpati putih no.181, Jakarta	<a href="#">Cetak Kartu</a> <a href="#">Edit</a> <a href="#">Hapus</a>

dan data anggota pun berhasil di hapus.

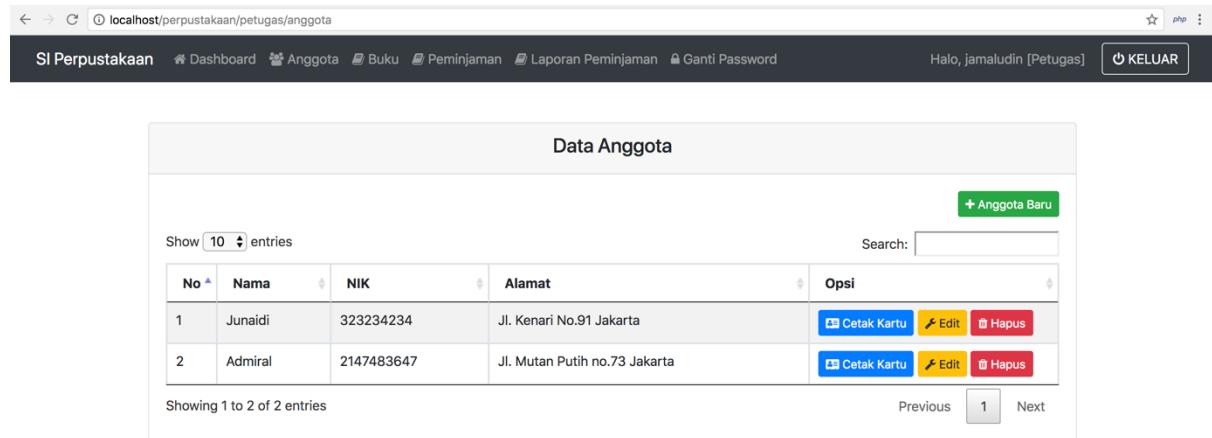
The screenshot shows an empty table with the same columns as before. It displays the message 'No data available in table'. Below the table, it says 'Showing 0 to 0 of 0 entries'.

No	Nama	NIK	Alamat	Opsi
----	------	-----	--------	------

Oke selesai sudah proses CRUD untuk anggota. Selanjutnya kita akan lanjut ke modul selanjutnya.

## Membuat Cetak Kartu Anggota

Sekarang coba kita input lagi beberapa data anggota. Untuk kita buatkan proses cetak kartu anggota. Di sini saya telah menginput lagi 2 data anggota.



No	Nama	NIK	Alamat	Opsi
1	Junaidi	323234234	Jl. Kenari No.91 Jakarta	<a href="#">Cetak Kartu</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	Admiral	2147483647	Jl. Mutan Putih no.73 Jakarta	<a href="#">Cetak Kartu</a> <a href="#">Edit</a> <a href="#">Hapus</a>

perhatikan pada link atau tombol “cetak kartu” pada data pelanggan yang telah kita buat di atas. Maka halaman akan dialihkan ke method atau function anggota\_kartu().

```
function anggota_kartu($id) {
    $where = array('id' => $id);
    // mengambil data dari database sesuai id
    $data['anggota'] = $this->m_data->edit_data($where, 'anggota')->result();
    $this->load->view('petugas/v_anggota_kartu',$data);
}
// akhir crud anggota
```

Caranya sama saja dengan membuat edit data. Pertama kita seleksi data anggota yang ber id sesuai dengan id pelanggan yang kita pilih untuk di cetak kartunya. Kamudian kita parsing ke view. Data akan kita tampilkan pada view *v\_anggota\_kartu.php* dalam folder petugas.

Buat sebuah view baru dalam folder petugas. Buat view baru dengan nama *v\_anggota\_kartu.php*.

### Application/views/petugas/v\_anggota\_kartu.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Cetak Kartu Anggota</title>
</head>
```

```

<body>

    <style type="text/css">
        .card{
            border: 1px solid #000;
            width: 450px;
        }

        .card-header{
            border-bottom: 1px solid #000;
            text-align: center;
            font-weight: bold;
            padding: 10px;
        }

        .card-body{
            padding: 20px;
        }
    </style>

    <div class="card">
        <div class="card-header">
            KARTU ANGGOTA PERPUSTAKAAN MALASNGODING
        </div>
        <div class="card-body">
            <div class="container">
                <table class="table table-borderless table-sm fs-2">
                    <?php
                    $no = 1;
                    foreach($anggota as $a) {
                        ?>
                        <tr>
                            <td width="14%">Nomor</td>
                            <td width="2%">:</td>
                            <td><?php echo 10000+$a->id; ?></td>
                        </tr>
                        <tr>
                            <td>Nama</td>
                            <td>:</td>
                            <td><?php echo $a->nama; ?></td>
                        </tr>
                        <tr>
                            <td>NIK</td>
                            <td>:</td>
                            <td><?php echo $a->nik; ?></td>
                        </tr>
                        <tr>
                            <td>Alamat</td>
                            <td>:</td>
                            <td><?php echo $a->alamat; ?></td>
                        </tr>
                        <?php
                    }
                    ?>
                </table>
            </div>
        </div>
    </div>

    <script type="text/javascript">
        window.print();
    </script>

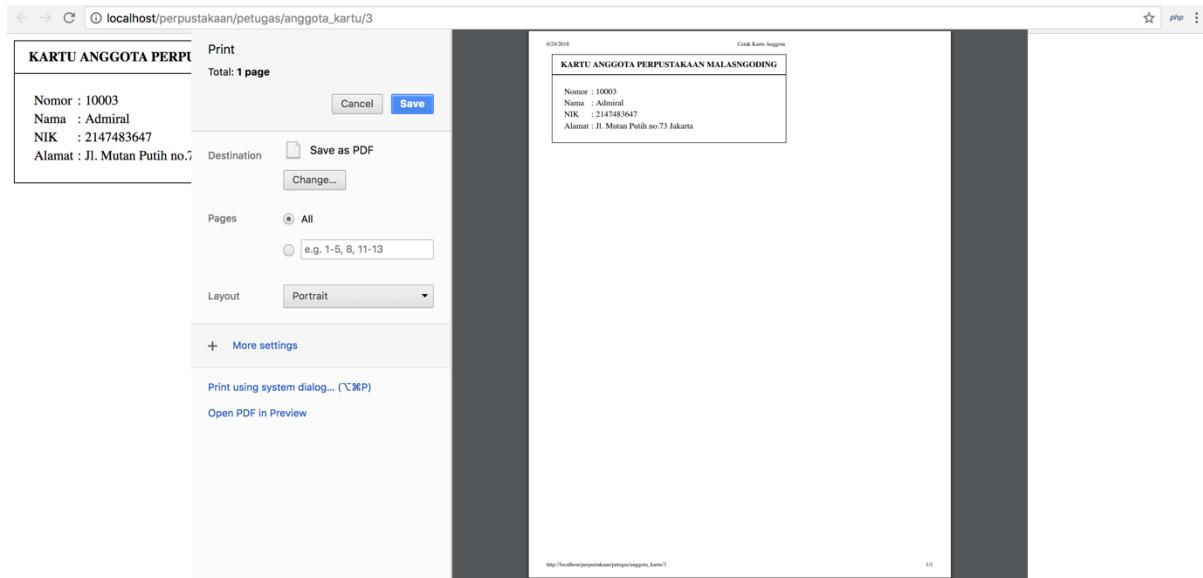
```

```
</html>
```

Perhatikan pada syntax di atas. Untuk membuat cetak halaman kita bisa menggunakan fungsi `window.print()` dari javascript.

```
<script type="text/javascript">
    window.print();
</script>
```

Coba lihat hasilnya. Klik pada tombol cetak kartu.



dan kartu anggota pun bisa dicetak langsung secara otomatis. Selesai.

# Membuat CRUD Data Buku Pada Petugas

Untuk membuat CRUD data buku, caranya juga sama saja dengan cara membuat CRUD data anggota yang telah kita bahas sebelumnya. Bedanya pada crud anggota, table yang di manipulasi adalah table “anggota”. Sedangkan pada CRUD buku, table yang akan kita manipulasi adalah table “buku”.

## Menampilkan data buku

Pada saat menu buku diklik, maka halaman akan dialihkan ke method buku() pada controller Petugas.php.

Oleh sebab itu sekarang kita akan membuat method buku() pada controller petugas.php. data buku akan kita tampilkan pada method buku().

### Application/controllers/Petugas.php

```
// crud buku
function buku() {
    // mengambil data dari database
    $data['buku'] = $this->m_data->get_data('buku')->result();
    $this->load->view('petugas/v_header');
    $this->load->view('petugas/v_buku',$data);
    $this->load->view('petugas/v_footer');
}
```

Kemudian buat sebuah view baru dengan nama *v\_buku.php* pada folder *petugas*.

### Application/views/petugas/v\_buku.php

```
<div class="container-fluid">
<div class="card">
    <div class="card-header text-center">
        <h4>Data Buku</h4>
    </div>
    <div class="card-body">

        <a href="php echo base_url().'petugas/buku_tambah' ?&gt;" class='btn btn-sm btn-success pull-right'&gt;&lt;i class="fa fa-plus"&gt;&lt;/i&gt; Buku Baru&lt;/a&gt;
        &lt;br/&gt;
        &lt;br/&gt;

        &lt;div class="table-responsive"&gt;
            &lt;table class="table table-bordered table-striped table-hover table-datatatable"&gt;
                &lt;thead&gt;
                    &lt;tr&gt;
                        &lt;th width="1%"&gt;No&lt;/th&gt;
                        &lt;th&gt;Judul Buku&lt;/th&gt;
                        &lt;th&gt;Tahun Terbit&lt;/th&gt;
                        &lt;th&gt;Penulis&lt;/th&gt;
                        &lt;th width="10%"&gt;Status&lt;/th&gt;
                        &lt;th width="13%"&gt;Opsi&lt;/th&gt;
                    &lt;/tr&gt;
                &lt;/thead&gt;</pre
```

```

</thead>
<tbody>
    <?php
    $no = 1;
    foreach($buku as $b) {
        ?>
        <tr>
            <td><?php echo $no++; ?></td>
            <td><?php echo $b->judul; ?></td>
            <td><?php echo $b->tahun; ?></td>
            <td><?php echo $b->penulis; ?></td>
            <td>
                <?php
                if($b->status == "1"){
                    echo "<span class='badge badge-success'>Tersedia</span>";
                }else if($b->status == "2"){
                    echo "<span class='badge badge-warning'>Sedang Dipinjam</span>";
                }
            ?>
            </td>
            <td>
                <a href=<?php echo base_url().'petugas/buku_edit/'.$b->id; ?>" class="btn btn-sm btn-warning"><i class="fa fa-wrench"></i> Edit</a>
                <a href=<?php echo base_url().'petugas/buku_hapus/'.$b->id; ?>" class="btn btn-sm btn-danger"><i class="fa fa-trash"></i> Hapus</a>
            </td>
        </tr>
        <?php
    }
    ?>
    </tbody>
    </table>
</div>

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

```

Kita lihat hasilnya, klik pada menu buku.

The screenshot shows a web application interface for a library system. At the top, there is a navigation bar with links: SI Perpustakaan, Dashboard, Anggota, Buku (which has a red arrow pointing to it), Peminjaman, Laporan Peminjaman, Ganti Password, and a sign-out button labeled 'KELUAR'. Below the navigation bar is a search bar with placeholder text 'Search:'. Underneath the search bar is a table header titled 'Data Buku' with columns: No, Judul Buku, Tahun Terbit, Penulis, Status, and Opsi. The table body contains the message 'No data available in table'. At the bottom of the table area, it says 'Showing 0 to 0 of 0 entries' and includes 'Previous' and 'Next' buttons.

selanjutnya kita akan membuat proses penginputan data buku.

## Menginput Data Buku Ke Database

Jika tombol “buku baru” di klik kita mengarahkan nya ke method *buku\_tambah()* pada controller Petugas.php.

### Application/controllers/Petugas.php

```
function buku_tambah() {
    $this->load->view('petugas/v_header');
    $this->load->view('petugas/v_buku_tambah');
    $this->load->view('petugas/v_footer');
}
```

Pada method *buku\_tambah()* kita menampilkan view *v\_buku\_tambah.php*. pada view ini lah kita buat form penginputan data buku.

Buat sebuah view baru dengan nama *v\_buku\_tambah.php* dalam folder petugas.

### Application/views/petugas/v\_buku\_tambah.php

```
<div class="container">
<div class="card">
    <div class="card-header text-center">
        <h4>Tambah Buku Baru</h4>
    </div>
    <div class="card-body">
        <a href="php echo base_url().'petugas/buku' ?&gt;" class='btn btn-sm btn-light
        btn-outline-dark pull-right'&gt;&lt;i class="fa fa-arrow-left"&gt;&lt;/i&gt; Kembali&lt;/a&gt;</pre

```

```

<br/>
<br/>

<form method="post" action="php echo base_url().'petugas/buku_tambah_aksi';
?&gt;&gt;&gt;
    &lt;div class="form-group"&gt;
        &lt;label class="font-weight-bold" for="judul"&gt;Judul Buku&lt;/label&gt;
        &lt;input type="text" class="form-control" name="judul"
placeholder="Masukkan judul buku" required="required"&gt;
    &lt;/div&gt;
    &lt;div class="form-group"&gt;
        &lt;label class="font-weight-bold" for="tahun"&gt;Tahun Terbit&lt;/label&gt;
        &lt;select class="form-control" name="tahun" required="required"&gt;
            &lt;option value=""&gt;- Pilih tahun&lt;/option&gt;
            &lt;?php for($stahun=date('Y');$stahun&gt;=1990;$stahun--) { ?&gt;
                &lt;option value="&lt;?php echo $stahun; ?&gt;"&gt;&lt;?php echo $stahun; ?&gt;&lt;/option&gt;
            &lt;?php } ?&gt;
        &lt;/select&gt;
    &lt;/div&gt;
    &lt;div class="form-group"&gt;
        &lt;label class="font-weight-bold" for="penulis"&gt;Penulis Buku&lt;/label&gt;
        &lt;input type="text" class="form-control" name="penulis"
placeholder="Masukkan nama penulis" required="required"&gt;
    &lt;/div&gt;
    &lt;input type="submit" class="btn btn-primary" value="Simpan"&gt;
&lt;/form&gt;

    &lt;/div&gt;
&lt;/div&gt;
&lt;/div&gt;
</pre

```

Form penginputan data buku di atas akan di proses pada method `buku_tambah_aksi()` dalam controller Petugas.

Oh ya pada form pemilihan tahun kita membuat tahunnya dengan menggunakan perulangan `for()`. Kita menampilkan dari tahun 1990 sampai tahun sekarang (2018).

```

<?php for($stahun=date('Y');$stahun>=1990;$stahun--) { ?>
    <option value="<?php echo $stahun; ?>"><?php echo $stahun; ?></option>
<?php } ?>

```

Jadi buat sebuah method baru lagi untuk menghandle penginputan data buku yang diinput. Buat method `buku_tambah_aksi()` pada controller Petugas.php.

### **Application/controllers/Petugas.php**

```

function buku_tambah_aksi(){
    $judul = $this->input->post('judul');
    $tahun = $this->input->post('tahun');
    $penulis = $this->input->post('penulis');

    $data = array(
        'judul' => $judul,
        'tahun' => $tahun,
        'penulis' => $penulis,
        'status' => 1
    );
}

```

```

    // insert data ke database
    $this->m_data->insert_data($data,'buku');

    // mengalihkan halaman ke halaman data buku
    redirect(base_url().'petugas/buku');
}

```

Perhatikan pada array data di atas, data status kita tetapkan dengan "1". Jadi setiap buku baru yang diinput, status nya menjadi "1". 1 berarti "tersedia". Dan 2 berarti "sedang dipinjam".

Sekarang akan kita coba jalankan pada browser. Klik tombol "Buku Baru". Dan input data buku baru.

localhost/perpustakaan/petugas/buku\_tambah

SI Perpustakaan

Halo, jamaludin [Petugas] KELUAR

Tambah Buku Baru

Judul Buku

Tahun Terbit

Penulis Buku

Simpan

setelah diisi, klik simpan.

<a href="#">←</a> <a href="#">→</a> <a href="#">C</a> <span>localhost/perpustakaan/petugas/buku</span>		<a href="#">php</a> <a href="#">⋮</a>												
SI Perpustakaan	<a href="#">Dashboard</a> <a href="#">Anggota</a> <a href="#">Buku</a> <a href="#">Peminjaman</a> <a href="#">Laporan Peminjaman</a> <a href="#">Ganti Password</a>	Halo, jamaludin [Petugas] <a href="#">KELUAR</a>												
<b>Data Buku</b>														
<div style="display: flex; justify-content: space-between;"> <span>Show <a href="#">10</a> entries</span> <span>Search: <input type="text"/></span> <span><a href="#">+ Buku Baru</a></span> </div>														
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>No</th> <th>Judul Buku</th> <th>Tahun Terbit</th> <th>Penulis</th> <th>Status</th> <th>Opsi</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Tutorial Codeigniter Lengkap Membangun Sistem Informasi Perpustakaan</td> <td>2018</td> <td>Diki Alfarabi Hadi</td> <td><span style="background-color: green; color: white; padding: 2px 5px;">Tersedia</span></td> <td style="text-align: center;"> <a href="#"> Edit</a> <a href="#"> Hapus</a> </td> </tr> </tbody> </table>			No	Judul Buku	Tahun Terbit	Penulis	Status	Opsi	1	Tutorial Codeigniter Lengkap Membangun Sistem Informasi Perpustakaan	2018	Diki Alfarabi Hadi	<span style="background-color: green; color: white; padding: 2px 5px;">Tersedia</span>	<a href="#"> Edit</a> <a href="#"> Hapus</a>
No	Judul Buku	Tahun Terbit	Penulis	Status	Opsi									
1	Tutorial Codeigniter Lengkap Membangun Sistem Informasi Perpustakaan	2018	Diki Alfarabi Hadi	<span style="background-color: green; color: white; padding: 2px 5px;">Tersedia</span>	<a href="#"> Edit</a> <a href="#"> Hapus</a>									
Showing 1 to 1 of 1 entries														
<a href="#">Previous</a> <span style="border: 1px solid #ccc; padding: 2px 5px; margin: 0 5px;">1</span> <a href="#">Next</a>														

nah data buku pun berhasil di simpan. Perhatikan pada kolumn status. Pada kolumn status kita telah membuat pengecekan pada view v\_buku.php. jika status = 1 berarti tampilkan "Tersedia". Jika status = 2, berarti tampilkan "Sedang dipinjam".

## Membuat Update Data Buku Pada Database

Pada saat tombol edit diklik, maka akan diarahkan ke method buku\_edit() pada controller Petugas.php. seperti yang telah kita buat sebelumnya.

Jadi sekarang kita buat lagi method baru dengan nama buku\_edit() pada controller Petugas.php. pada method ini kita akan membuat form untuk pengeditan data buku.

### Application/controllers/Petugas.php

```
function buku_edit($id){
    $where = array('id' => $id);
    // mengambil data dari database sesuai id
    $data['buku'] = $this->m_data->edit_data($where, 'buku')->result();
    $this->load->view('petugas/v_header');
    $this->load->view('petugas/v_buku_edit', $data);
    $this->load->view('petugas/v_footer');
}
```

Buat view baru lagi dalam folder "petugas" dengan nama v\_petugas\_edit.php.

## Application/views/petugas/v\_petugas\_edit.php

```
<div class="container">
    <div class="card">
        <div class="card-header text-center">
            <h4>Edit Buku</h4>
        </div>
        <div class="card-body">
            <a href="php echo base_url().'petugas/buku' ?" class="btn btn-sm btn-light
btn-outline-dark pull-right"><i class="fa fa-arrow-left"></i> Kembali</a>
            <br/>
            <br/>

            <?php foreach($buku as $b) { ?>
                <form method="post" action="php echo base_url().'petugas/buku_update';
?&gt;"&gt;
                    &lt;div class="form-group"&gt;
                        &lt;label class="font-weight-bold" for="judul"&gt;Judul Buku&lt;/label&gt;
                        &lt;input type="hidden" name="id" value="<?php echo $b-&gt;id; ?&gt;"&gt;
                        &lt;input type="text" class="form-control" name="judul"
placeholder="Masukkan judul buku" required="required" value="<?php echo $b-&gt;judul;
?&gt;"&gt;
                    &lt;/div&gt;
                    &lt;div class="form-group"&gt;
                        &lt;label class="font-weight-bold" for="tahun"&gt;Tahun Terbit&lt;/label&gt;
                        &lt;select class="form-control" name="tahun" required="required"&gt;
                            &lt;option value=""&gt;- Pilih tahun&lt;/option&gt;
                            &lt;?php for($tahun=date('Y');$tahun&gt;=1990;$tahun--) { ?&gt;
                                &lt;option &lt;?php if($tahun==$b-&gt;tahun){echo "selected='selected'" ;} ?&gt;
value="<?php echo $tahun; ?&gt;"&gt;&lt;?php echo $tahun; ?&gt;&lt;/option&gt;
                            &lt;?php } ?&gt;
                        &lt;/select&gt;
                    &lt;/div&gt;
                    &lt;div class="form-group"&gt;
                        &lt;label class="font-weight-bold" for="penulis"&gt;Penulis Buku&lt;/label&gt;
                        &lt;input type="text" class="form-control" name="penulis"
placeholder="Masukkan nama penulis" required="required" value="<?php echo $b-
&gt;penulis; ?&gt;"&gt;
                    &lt;/div&gt;
                    &lt;div class="form-group"&gt;
                        &lt;label class="font-weight-bold" for="status"&gt;Status Buku&lt;/label&gt;
                        &lt;select class="form-control" name="status" required="required"&gt;
                            &lt;option value=""&gt;- Pilih status&lt;/option&gt;
                            &lt;option &lt;?php if($b-&gt;status=="1"){echo "selected='selected'" ;} ?&gt;
value="1"&gt;Tersedia&lt;/option&gt;
                            &lt;option &lt;?php if($b-&gt;status=="2"){echo "selected='selected'" ;} ?&gt;
value="2"&gt;Sedang Dipinjam&lt;/option&gt;
                        &lt;/select&gt;
                    &lt;/div&gt;

                    &lt;input type="submit" class="btn btn-primary" value="Simpan"&gt;
                &lt;/form&gt;
            &lt;?php } ?&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;</pre
```

Form edit ini akan kita proses pada method `buku_update()`.

Buat method baru lagi dengan nama `buku_update()` dalam controller `Petugas.php`.

## Application/controllers/Petugas.php

```
function buku_update() {
    $id = $this->input->post('id');
    $judul = $this->input->post('judul');
    $stahun = $this->input->post('tahun');
    $penulis = $this->input->post('penulis');
    $status = $this->input->post('status');

    $where = array(
        'id' => $id
    );

    $data = array(
        'judul' => $judul,
        'tahun' => $stahun,
        'penulis' => $penulis,
        'status' => $status
    );

    // update data ke database
    $this->m_data->update_data($where,$data,'buku');

    // mengalihkan halaman ke halaman data buku
    redirect(base_url().'petugas/buku');
}
```

Sekarang coba edit data buku dengan cara klik tombol edit.

The screenshot shows a table titled "Data Buku" with columns: No, Judul Buku, Tahun Terbit, Penulis, Status, and Opsi. The first row contains the data: 1, Tutorial Codeigniter Lengkap Membangun Sistem Informasi Perpustakaan, 2018, Diki Alfarabi Hadi, Tersedia, and a row of buttons for Edit and Delete. A red arrow points from the "Edit" button in the last row to the explanatory text below.

No	Judul Buku	Tahun Terbit	Penulis	Status	Opsi
1	Tutorial Codeigniter Lengkap Membangun Sistem Informasi Perpustakaan	2018	Diki Alfarabi Hadi	Tersedia	Edit  Hapus

maka akan muncul form edit data buku.

**Edit Buku**

Judul Buku  
Tutorial Codeigniter Lengkap Membuat Sistem Informasi Perpustakaan

Tahun Terbit  
2018

Penulis Buku  
Diki Alfarabi Hadi

Status Buku  
- Pilih status  
✓ Tersedia  
Sedang Dipinjam

**Kembali** **Simpan**

Di sini saya mengubah judul buku dari “Tutorial Codeigniter Lengkap Membangun Sistem Informasi Perpustakaan” menjadi “Tutorial Codeigniter Lengkap Membuat Sistem Informasi Perpustakaan”. Dan disini kita juga bisa mengupdate status buku. Bisa megubah status buku menjadi tersedia atau sedang dipinjam.

Dan klik simpan.

**Data Buku**

Show 10 entries Search:

No	Judul Buku	Tahun Terbit	Penulis	Status	Opsi
1	Tutorial Codeigniter Lengkap Membuat Sistem Informasi Perpustakaan	2018	Diki Alfarabi Hadi	Tersedia	<b>Edit</b> <b>Hapus</b>

Showing 1 to 1 of 1 entries Previous 1 Next

dan selesai. Selanjutnya kita akan membuat fungsi untuk menghapus data buku dari database.

## Membuat Hapus Buku Dari Database

Pada tombol “Hapus” sudah kita buatkan juga link yang mengarah ke method *buku\_hapus()*.

Sekarang kita akan membuat fungsi *buku\_hapus()* pada controller Petugas.php.

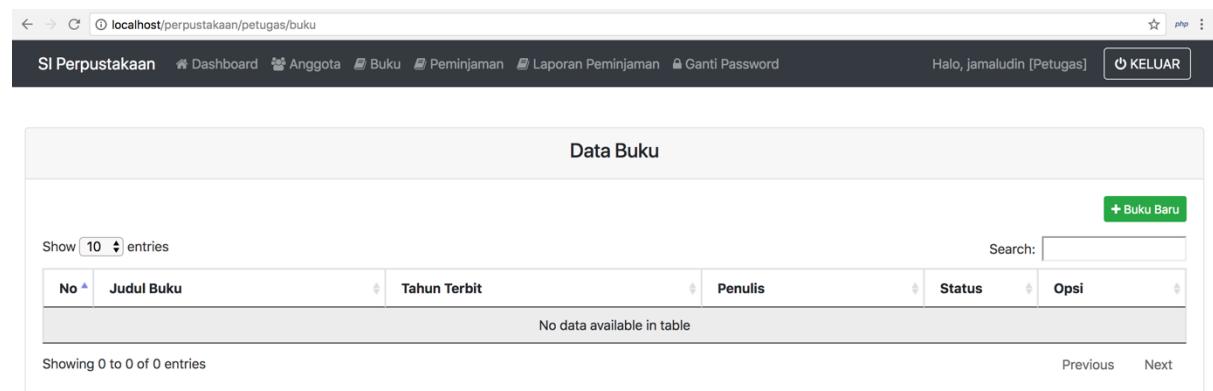
### Application/controllers/Petugas.php

```
function buku_hapus($id) {
    $where = array(
        'id' => $id
    );

    // menghapus data buku dari database sesuai id
    $this->m_data->delete_data($where, 'buku');

    // mengalihkan halaman ke halaman data buku
    redirect(base_url().'petugas/buku');
}
// akhir crud buku
```

Dan coba klik tombol hapus. Maka data buku akan dihapus dari database.



sampai di sini maka selesai lah proses CRUD untuk data buku.

# Membuat Proses Transaksi Peminjaman Buku Pada Petugas

Sampailah kita pada tahapan inti dari sistem informasi perpustakaan ini. Yaitu proses transaksi peminjaman buku. Saya jelaskan dulu sedikit tentang proses transaksi peminjaman yang akan kita buat.

Kembali perhatikan pada table peminjaman yang telah kita buat sebelumnya pada tahap perancangan.

The screenshot shows the MySQL Workbench interface with the following details:

- Server: localhost
- Database: perpustakaan
- Table: peminjaman
- Table structure view is selected.
- Columns listed:

  - # peminjaman\_id (Primary Key)
  - peminjaman\_buku
  - peminjaman\_anggota
  - peminjaman\_tanggal\_mulai
  - peminjaman\_tanggal\_sampai
  - peminjaman\_status

- Action buttons at the bottom: Check all, With selected, Browse, Change, Drop, Primary, Unique.

pada table peminjaman, data-data yang akan kita input diantaranya adalah menginput “id buku” yang ingin di pinjam pada kolumn “peminjaman\_buku”, “id anggota” yang ingin meminjam buku pada “peminjaman\_anggota”, kemudian menginput tanggal buku milai di pinjam, tanggal buku selesai dipinjam. Dan status peminjaman kita mengisi angka “2” untuk status “sedang dipinjam”, dan angka 1 untuk status “peminjaman selesai”.

## Menampilkan Data Transaksi Peminjaman

Oke langsung saja kita mulai. Perhatikan pada menu “peminjaman”. Jika diklik halaman akan dialihkan ke method *peminjaman()* pada controller Petugas.php.

Buat sebuah method, yaitu method *peminjaman()* pada controller Petugas.php.

### Application/controllers/Petugas.php

```
// proses transaksi_peminjaman
```

```

function peminjaman() {
    // mengambil data peminjaman buku dari database | dan mengurutkan data dari
    id peminjaman terbesar ke terkecil (desc)
    $data['peminjaman'] = $this->db->query("select * from
    peminjaman,buku,anggota where peminjaman.peminjaman_buku=buku.id and
    peminjaman.peminjaman_anggota=anggota.id order by peminjaman_id desc")->result();
    $this->load->view('petugas/v_header');
    $this->load->view('petugas/v_peminjaman',$data);
    $this->load->view('petugas/v_footer');
}

```

Pada method peminjaman ini kita mengambil data dari database dengan menggunakan query berikut.

```

select * from peminjaman,buku,anggota where peminjaman.peminjaman_buku=buku.id and
peminjaman.peminjaman_anggota=anggota.id order by peminjaman_id desc

```

Kemudian kita parsing ke view v\_peminjamana.php. query di atas iyalah query untuk menampilkan data dari 3 table yang berelasi sekaligus. Yaitu table peminjaman, buku dan anggota. Relasinya iyalah id anggota yang sesuai pada table "peminjaman" dengan table id anggota pada table "anggota". Dan id buku yang sesuai pada table "peminjaman" dengan id buku yang ada pada table "buku".

Karena data akan kita tampilkan pada view v\_peminjaman.php, maka akan kita buat view baru dengan nama v\_peminjaman.php. buat view dengan nama **v\_peminjaman.php** dalam folder *petugas*.

### **Application/views/petugas/v\_peminjaman.php**

```

<div class="container-fluid">
    <div class="card">
        <div class="card-header text-center">
            <h4>Data Peminjaman Buku</h4>
        </div>
        <div class="card-body">
            <a href="php echo base_url() . 'petugas/peminjaman_tambah' ?&gt;" class='btn btn-sm btn-success pull-right'&gt;&lt;i class="fa fa-plus"&gt;&lt;/i&gt; Peminjaman Baru&lt;/a&gt;
            &lt;br/&gt;
            &lt;br/&gt;

            &lt;div class="table-responsive"&gt;
                &lt;table class="table table-bordered table-striped table-hover table-datable"&gt;
                    &lt;thead&gt;
                        &lt;tr&gt;
                            &lt;th width="1%"&gt;No&lt;/th&gt;
                            &lt;th&gt;Buku&lt;/th&gt;
                            &lt;th&gt;Peminjam&lt;/th&gt;
                            &lt;th&gt;Mulai Pinjam&lt;/th&gt;
                            &lt;th&gt;Pinjam Sampai&lt;/th&gt;
                            &lt;th&gt;Status&lt;/th&gt;
                            &lt;th width="16%"&gt;Opsi&lt;/th&gt;
                        &lt;/tr&gt;
                    &lt;/thead&gt;
                    &lt;tbody&gt;
</pre

```

```

<?php
$no = 1;
foreach($peminjaman as $p) {
    ?
    <tr>
        <td><?php echo $no++; ?></td>
        <td><?php echo $p->judul; ?></td>
        <td><?php echo $p->nama; ?></td>
        <td><?php echo date('d-m-Y', strtotime($p->peminjaman_tanggal_mulai));
?></td>
        <td><?php echo date('d-m-Y', strtotime($p-
>peminjaman_tanggal_sampai)); ?></td>
        <td>
            <?php
            if($p->peminjaman_status == "1") {
                echo "<div class='badge badge-success'>Selesai</div>";
            } else if($p->peminjaman_status == "2") {
                echo "<div class='badge badge-warning'>Dipinjam</div>";
            }
            ?
        </td>
        <td class="text-center">

            <?php
            if($p->peminjaman_status=='1') {
                echo "-";
            } else if($p->peminjaman_status=='2') {
                ?
                <a href=<?php echo base_url().'petugas/peminjaman_selesai/'.$p-
>peminjaman_id; ?>" class="btn btn-sm btn-warning"><i class="fa fa-refresh"></i>
Selesai</a>
                <a href=<?php echo base_url().'petugas/peminjaman_batalkan/'.$p-
>peminjaman_id; ?>" class="btn btn-sm btn-danger"><i class="fa fa-close"></i>
Batalkan</a>
            <?php
            }
            ?
        </td>
    </tr>
    <?php
}
?
</tbody>
</table>
</div>

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

```

<a href="#">localhost/perpustakaan/petugas/peminjaman</a>																							
SI Perpustakaan		<a href="#">Dashboard</a>			<a href="#">Anggota</a>			<a href="#">Buku</a>															
<a href="#">Peminjaman</a>		<a href="#">Laporan Peminjaman</a>			<a href="#">Ganti Password</a>			Halo, jamaludin [Petugas]															
								<a href="#">KELUAR</a>															
<b>Data Peminjaman Buku</b>																							
<div style="display: flex; justify-content: space-between;"> <span>Show <b>10</b> entries</span> <span>Search: <input type="text"/></span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>No</th> <th>Buku</th> <th>Peminjam</th> <th>Mulai Pinjam</th> <th>Pinjam Sampai</th> <th>Status</th> <th>Opsi</th> </tr> </thead> <tbody> <tr> <td colspan="7" style="text-align: center;">No data available in table</td></tr> </tbody> </table>										No	Buku	Peminjam	Mulai Pinjam	Pinjam Sampai	Status	Opsi	No data available in table						
No	Buku	Peminjam	Mulai Pinjam	Pinjam Sampai	Status	Opsi																	
No data available in table																							
Showing 0 to 0 of 0 entries																							
Previous      Next																							

pada table peminjaman di atas, kita membuat pengecekan pada bagian status peminjaman.

```
<?php
if($p->peminjaman_status == "1") {
    echo "<div class='badge badge-success'>Selesai</div>";
} else if($p->peminjaman_status == "2") {
    echo "<div class='badge badge-warning'>Dipinjam</div>";
}
?>
```

Jika status peminjaman adalah 1, maka tampilkan “selesai”. Jika status peminjaman adalah 2, maka tampilkan “Dipinjam”.

Begini juga pada kolumn opsi, kita juga membuat pengecekan.

```
<?php
if($p->peminjaman_status=='1') {
    echo "-";
} else if($p->peminjaman_status=='2') {
?>
    <a href="<?php echo base_url().'petugas/peminjaman_selesai/'.$p->peminjaman_id;
?>" class="btn btn-sm btn-warning"><i class="fa fa-refresh"></i> Selesai</a>
    <a href="<?php echo base_url().'petugas/peminjaman_batalkan/'.$p->peminjaman_id;
?>" class="btn btn-sm btn-danger"><i class="fa fa-close"></i> Batalkan</a>
<?php
}
?>
```

Jika status peminjaman adalah 1, maka tampililka “-” saja. Namun jika status peminjamannya adalah 2 (sedang dipinjam) tampilkan kedua tombol yang kita buat, yaitu tombol “selesai” dan tombol “batalkan”.

oke selesai, tapi data masih kosong makanya tidak ada data yang muncul. Selanjutnya kita akan buat proses penginputan data peminjaman baru.

## Menginput Peminjaman Baru

Pada saat tombol “Peminjaman Baru” diklik, maka halaman akan dialihkan ke method *peminjaman\_tambah()* pada controller Petugas.php.

### Application/controllers/Petugas.php

```
function peminjaman_tambah() {
    // mengambil data buku yang berstatus 1 (tersedia) dari database
    $where = array('status'=>1);
    $data['buku'] = $this->m_data->edit_data($where, 'buku')->result();
    // mengambil data anggota dari database
    $data['anggota'] = $this->m_data->get_data('anggota')->result();
    $this->load->view('petugas/v_header');
    $this->load->view('petugas/v_peminjaman_tambah', $data);
    $this->load->view('petugas/v_footer');
}
```

Pada method di atas, data yang kita parsing adalah data buku dan data anggota. Kita parsing ke view *v\_peminjaman\_tambah.php*. jadi buat sebuah view baru lagi dengan nama *v\_peminjaman\_tambah.php* dalam folder *petugas*.

### Application/views/petugas/v\_peminjaman\_tambah.php

```
<div class="container">
<div class="card">
    <div class="card-header text-center">
        <h4>Proses Transaksi Peminjaman Buku</h4>
    </div>
    <div class="card-body">
        <a href="=base_url('petugas/peminjaman') ?&gt;" class='btn btn-sm btn-light btn-outline-dark pull-right'&gt;&lt;i class="fa fa-arrow-left"&gt;&lt;/i&gt; Kembali&lt;/a&gt;
        &lt;br/&gt;
        &lt;br/&gt;

        &lt;form method="post" action="<?=base_url('petugas/peminjaman_aksi'); ?&gt;"&gt;
            &lt;div class="form-group"&gt;
                &lt;label class="font-weight-bold" for="buku"&gt;Buku&lt;/label&gt;
                &lt;select name="buku" class="form-control"&gt;
                    &lt;option value=""&gt;- Pilih buku&lt;/option&gt;
                    &lt;?php foreach($buku as $b){ ?&gt;
                        &lt;option value="<?=base_url('petugas/peminjaman') . $b-&gt;id; ?&gt;"&gt;&lt;?php echo $b-&gt;judul. ' | ' . $b-&gt;tahun. ' | ' . $b-&gt;penulis; ?&gt;&lt;/option&gt;
                    &lt;?php } ?&gt;
                &lt;/select&gt;
            &lt;/div&gt;
            &lt;div class="form-group"&gt;</pre
```

```

<label class="font-weight-bold" for="anggota">Anggota</label>
<select name="anggota" class="form-control">
    <option value="">- Pilih anggota</option>
    <?php foreach($anggota as $a){ ?>
        <option value="<?php echo $a->id; ?>"><?php echo "Nama : ". $a->nama.
' | NIK : ' . $a->nik; ?></option>
    <?php } ?>
</select>
</div>
<div class="form-group">
    <label class="font-weight-bold" for="tanggal_mulai">Tanggal Mulai Pinjam</label>
    <input type="date" class="form-control" name="tanggal_mulai" placeholder="Masukkan tanggal mulai pinjam">
</div>
<div class="form-group">
    <label class="font-weight-bold" for="tanggal_sampai">Tanggal Pinjam Sampai</label>
    <input type="date" class="form-control" name="tanggal_sampai" placeholder="Masukkan tanggal pinjam sampai">
</div>

    <input type="submit" class="btn btn-primary" value="Simpan">
</form>

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

```

Seperti yang teman-teman lihat, di form penginputan data peminjaman baru di atas, kita menampilkan data buku dan data anggota pada form *select option (dropdown)*. Jadi nantinya petugas tinggal memilih anggota yang meminjam buku, dan memilih buku yang ingin dipinjam, kemudian memasukkan tanggal buku dipinjam, dan tanggal buku dikembalikan.

Form peminjaman di atas akan diproses pada method *peminjaman\_aksi()* pada controller Petugas.php.

```
<form method="post" action="<?php echo base_url() .'petugas/peminjaman_aksi'; ?>">
```

Maka buat sebuah method baru lagi dalam controller Petugas.php. yaitu method *peminjaman\_aksi()*.

## Application/controllers/Petugas.php

```

function peminjaman_aksi(){
    $buku = $this->input->post('buku');
    $anggota = $this->input->post('anggota');
    $tanggal_mulai = $this->input->post('tanggal_mulai');
    $tanggal_sampai = $this->input->post('tanggal_sampai');

    $data = array(
        'peminjaman_buku' => $buku,
        'peminjaman_anggota' => $anggota,

```

```

        'peminjaman_tanggal_mulai' => $tanggal_mulai,
        'peminjaman_tanggal_sampai' => $tanggal_sampai,
        'peminjaman_status' => 2
    );

    // insert data ke database
    $this->m_data->insert_data($data,'peminjaman');

    // mengubah status buku menjadi di pinjam (2)
    $w = array(
        'id' => $buku
    );
    $d = array(
        'status' => 2
    );
    $this->m_data->update_data($w,$d,'buku');

    // mengalihkan halaman ke halaman data peminjaman
    redirect(base_url().'petugas/peminjaman');
}

```

Pada method peminjaman\_aksi() di atas kita melakukan pemrosesan data peminjaman. Yaitu berupa data id anggota yang meminjam, id buku yang di pinjam, tanggal peminjaman di mulai, tanggal peminjaman berakhir dan status peminjaman. Untuk status peminjamannya kita menetapkan dengan angka 2. Yang berarti 2=sedang di pinjam, dan 1 = peminjaman selesai. Kita menginput data tersebut ke table “peminjaman”.

Tidak hanya itu, pada bagian selanjutnya, kita juga mengubah status buku yang dipinjam, dari status 1=tersedia ke 2=dipinjam.

```

// mengubah status buku menjadi di pinjam (2)
$w = array(
    'id' => $buku
);
$d = array(
    'status' => 2
);
$this->m_data->update_data($w,$d,'buku');

```

Setelah mengedit atau mengubah status buku yang dipinjam, selanjutnya kita alihkan halaman kembali ke halaman peminjaman (method peminjaman pada controller Petugas.php).

```

// mengalihkan halaman ke halaman data peminjaman
redirect(base_url().'petugas/peminjaman');

```

Sekarang akan kita coba eksekusi pada browser untuk melihat hasilnya. Silahkan teman-teman klik pada tombol “Peminjaman Baru”. Maka akan muncul form penginputan peminjaman baru.

Oh ya, jangan lupa untuk menambahkan beberapa data buku dan data anggota. Sebagai data yang akan dipilih nantinya pada form penambahan peminjaman baru ini.

localhost/perpustakaan/petugas/peminjaman\_tambah

SI Perpustakaan

Halo, jamaludin [Petugas]

KELUAR

Proses Transaksi Peminjaman Buku

Buku

Anggota

Tanggal Mulai Pinjam

Tanggal Pinjam Sampai

Simpan

dan isi data peminjamannya dengan cara memilih buku yang ingin dipinjam, pilih juga anggota yang akan melakukan peminjaman, pilih tanggal mulai peminjaman, pilih juga tanggal peminjamannya selesai.

Disini saya mengisi data dengan memilih buku yang di pinjam adalah buku “tutorial codeigniter lengkap membangun sistem informasi perpustakaan”. Memilih anggota yang bernama “admiral”, tanggal mulai peminjamannya tanggal “27/6/2018” sampai “28/6/2018”. Dan klik simpan.

Maka data peminjamanpun berhasil tersimpan dan tampil pada halaman data peminjaman buku.

localhost/perpustakaan/petugas/peminjaman						php	...
SI Perpustakaan						Halo, jamaludin [Petugas]	KELUAR
<b>Data Peminjaman Buku</b>							
<b>+ Peminjaman Baru</b>							
Show <input type="button" value="10"/> entries	Search:						
No	Buku	Peminjam	Mulai Pinjam	Pinjam Sampai	Status	Opsi	
1	Tutorial Codeigniter Lengkap Membangun Sistem Informasi Perpustakaan	Admiral	27-06-2018	28-06-2018	Dipinjam	Selesai	Batalkan
Showing 1 to 1 of 1 entries							
				Previous	1	Next	

seperti yang terlihat pada gambar di atas, data peminjaman pun berhasil tampil. Perhatikan, pada bagian statusnya adalah "Dipinjam", karena saat peminjaman baru diinput, maka nilai status yang kita input adalah "2". Seperti yang telah saya jelaskan sebelumnya. Selanjutnya pada saat membuat proses peminjaman "selesai", nilai statusnya lah yang akan kita ubah menjadi "1" yang berarti "Selesai".

Dan coba perhatikan pada data buku.

localhost/perpustakaan/petugas/buku						php	...
SI Perpustakaan						Halo, jamaludin [Petugas]	KELUAR
<b>Data Buku</b>							
<b>+ Buku Baru</b>							
Show <input type="button" value="10"/> entries	Search:						
No	Judul Buku	Tahun Terbit	Penulis	Status	Opsi		
1	Tutorial Codeigniter Lengkap Membangun Sistem Informasi Perpustakaan	2018	Diki Alfarabi Hadi	Sedang Dipinjam		Hapus	
2	Belajar cepat codeigniter dengan studi kasus membuat aplikasi rental mobil	2017	Diki Alfarabi Hadi	Tersedia		Hapus	
3	Belajar PHP dan MySQLi dari dasar sampai membuat sistem informasi laundry	2018	Diki Alfarabi Hadi	Tersedia		Hapus	
4	Panduan Belajar JQuery Dari Dasar Sampai Mahir Untuk Pemula	2016	Diki Alfarabi Hadi	Tersedia		Hapus	
5	Belajar HTML dan CSS dari Dasar Sampai Mahir	2016	Diki Alfarabi Hadi	Tersedia		Hapus	
Showing 1 to 5 of 5 entries							
				Previous	1	Next	

status dari buku yang di pinjam tadi pun berhasil berubah menjadi "Sedang Dipinjam". Karena pada saat proses penginputan data peminjaman sebelumnya kita telah mengubah status buku yang dipinjam menjadi "2".

## Membuat Proses Batalkan Peminjaman

Setelah sebelumnya kita selesai membuat proses menampilkan data peminjaman, membuat proses penginputan data peminjaman buku, selanjutnya pada sub judul ini kita akan membuat proses pembatalan peminjaman. Fitur ini perlu kita buat untuk mengatasi masalah jika petugas salah menginput data peminjaman, atau mungkin si anggota sendiri yang ingin membatalkan peminjaman buku.

Untuk membuat proses batalkan transaksi sudah kita buat sebuah tombol "Batalkan" pada data peminjaman. Jika diklik maka akan mengalihkan halaman menuju function/method *peminjaman\_batalkan()* pada controller Petugas.php. sambil mengirimkan data id peminjaman melalui url. Silahkan perhatikan pada syntax tombol batalkan yang telah kita buat pada view v\_peminjaman.php.

```
<a href="php echo base_url().'petugas/peminjaman_batalkan/'.$p-&gt;peminjaman_id; ?&gt;" class="btn btn-sm btn-danger"&gt;&lt;i class="fa fa-close"&gt;&lt;/i&gt; Batalkan&lt;/a&gt;</pre
```

Sekarang kita akan membuat method *peminjaman\_batalkan()* pada controller Petugas.php. untuk menangani proses pembatalan transaksi peminjaman.

### Application/controllers/Petugas.php

```
function peminjaman_batalkan($id){
    $where = array(
        'peminjaman_id' => $id
    );

    // mengambil data buku pada peminjaman ber id tersebut
    $data = $this->m_data->edit_data($where, 'peminjaman')->row();
    $buku = $data->peminjaman_buku;

    // mengembalikan status buku kembali ke tersedia (1)
    $w = array(
        'id' => $buku
    );
    $d = array(
        'status' => 1
    );
    $this->m_data->update_data($w, $d, 'buku');

    // menghapus data peminjaman dari database sesuai id
    $this->m_data->delete_data($where, 'peminjaman');

    // mengalihkan halaman ke halaman data buku
    redirect(base_url().'petugas/peminjaman');
}
```

Ini merupakan kebalikan dari fungsi input data peminjaman yang kita buat sebelumnya. Disini kita hanya menghapus data peminjaman yang ingin dibatalkan. Dan kemudian mengembalikan atau mengubah **status** buku yang dipilih dalam peminjaman tersebut kembali menjadi “1” yang berarti “Tersedia”.

Dan terakhir kita mengalihkan halaman kembali ke method peminjaman (halaman data peminjaman).

```
// mengalihkan halaman ke halaman data buku  
redirect(base_url().'petugas/peminjaman');
```

Sekarang coba klik pada tombol batalkan.

The screenshot shows a web browser window with the URL `localhost/perpustakaan/petugas/peminjaman`. The page title is "Data Peminjaman Buku". At the top right, there is a "KELUAR" button. The main content is a table with the following data:

No	Buku	Peminjam	Mulai Pinjam	Pinjam Sampai	Status	Opsi
1	Tutorial Codeigniter Lengkap Membangun Sistem Informasi Perpustakaan	Admiral	27-06-2018	28-06-2018	Dipinjam	<span style="background-color: yellow;">✖ Selesai</span> <span style="background-color: red;">✖ Batalkan</span>

Below the table, there is a message: "Showing 1 to 1 of 1 entries". At the bottom right of the table, there are navigation buttons: "Previous", a page number "1", and "Next". A red arrow points from the "Batalkan" button in the last row of the table to the text "maka data peminjaman pun akan dihapus." below it.

maka data peminjaman pun akan dihapus.

localhost/perpustakaan/petugas/peminjaman

SI Perpustakaan Dashboard Anggota Buku Peminjaman Laporan Peminjaman Ganti Password Halo, jamaludin [Petugas] KELUAR

### Data Peminjaman Buku

Show 10 entries Search:

No	Buku	Peminjam	Mulai Pinjam	Pinjam Sampai	Status	Opsi
No data available in table						

Showing 0 to 0 of 0 entries Previous Next

kemudian status buku yang dipinjam pada transaksi peminjaman yang di batalkan pun akan kembali menjadi “tersedia”.

localhost/perpustakaan/petugas/buku

SI Perpustakaan Dashboard Anggota Buku Peminjaman Laporan Peminjaman Ganti Password Halo, jamaludin [Petugas] KELUAR

### Data Buku

Show 10 entries Search:

No	Judul Buku	Tahun Terbit	Penulis	Status	Opsi
1	Tutorial Codeigniter Lengkap Membangun Sistem Informasi Perpustakaan	2018	Diki Alfarabi Hadi	<span style="background-color: green; color: white; padding: 2px;">Tersedia</span>	Edit  Hapus
2	Belajar cepat codeigniter dengan studi kasus membuat aplikasi rental mobil	2017	Diki Alfarabi Hadi	<span style="background-color: green; color: white; padding: 2px;">Tersedia</span>	Edit  Hapus
3	Belajar PHP dan MySQLi dari dasar sampai membuat sistem informasi laundry	2018	Diki Alfarabi Hadi	<span style="background-color: green; color: white; padding: 2px;">Tersedia</span>	Edit  Hapus
4	Panduan Belajar JQuery Dari Dasar Sampai Mahir Untuk Pemula	2016	Diki Alfarabi Hadi	<span style="background-color: green; color: white; padding: 2px;">Tersedia</span>	Edit  Hapus
5	Belajar HTML dan CSS dari Dasar Sampai Mahir	2016	Diki Alfarabi Hadi	<span style="background-color: green; color: white; padding: 2px;">Tersedia</span>	Edit  Hapus

Showing 1 to 5 of 5 entries Previous 1 Next

sampai disini proses batalkan peminjaman telah berhasil dan selesai. Selanjutnya kita akan membuat proses penyelesaian peminjaman saat tombol selesai di klik.

## Membuat Proses Peminjaman Selesai

Setelah sebelumnya kita selesai membuat proses menampilkan data peminjaman, membuat proses penginputan data peminjaman buku, dan kita juga telah membuat pembatalan peminjaman. selanjutnya pada sub judul ini kita akan membuat proses selesai pada transaksi peminjaman buku.

Perhatikan pada tombol “selesai” yang telah kita buat pada data peminjaman buku. Di sini saya telah menambahkan beberapa data peminjaman, untuk contoh data yang akan diproses.

The screenshot shows a web browser window with the URL `localhost/perpustakaan/petugas/peminjaman`. The page title is "Data Peminjaman Buku". At the top, there is a navigation bar with links for Dashboard, Anggota, Buku, Peminjaman, Laporan Peminjaman, Ganti Password, and a user profile "Halo, jamaludin [Petugas]". There is also a "KELUAR" button. Below the navigation, there is a search bar and a green button labeled "+ Peminjaman Baru". The main content is a table with the following data:

No	Buku	Peminjam	Mulai Pinjam	Pinjam Sampai	Status	Opsi
1	Panduan Belajar JQuery Dari Dasar Sampai Mahir Untuk Pemula	Junaidl	26-06-2018	30-06-2018	Dipinjam	<span>C Selesai</span> <span>X Batalkan</span>
2	Tutorial Codeigniter Lengkap Membangun Sistem Informasi Perpustakaan	Admiral	26-06-2018	28-06-2018	Dipinjam	<span>C Selesai</span> <span>X Batalkan</span>

Below the table, it says "Showing 1 to 2 of 2 entries". At the bottom right, there are buttons for "Previous", "1", and "Next". A red arrow points from the text in the previous paragraph to the "C Selesai" button in the second row of the table.

pada tombol “selesai” telah kita buat akan dialihkan ke function `peminjaman_selesai()` pada controller Petugas.php sambil mengirimkan id peminjaman yang akan di proses (selesai).

Buat sebuah function baru lagi pada controller **Petugas.php**, yaitu function `peminjaman_selesai()`.

### Application/controllers/Petugas.php

```
function peminjaman_selesai($id) {
    $where = array(
        'peminjaman_id' => $id
    );

    // mengambil data buku pada peminjaman ber id tersebut
    $data = $this->m_data->edit_data($where, 'peminjaman')->row();
    $buku = $data->peminjaman_buku;

    // mengembalikan status buku kembali ke tersedia (1)
    $w = array(
        'id' => $buku
    )
}
```

```

);
$d = array(
    'status' => 1
);
$this->m_data->update_data($w,$d,'buku');

// mengubah status peminjaman menjadi selesai (1)
$this->m_data-
>update_data($where,array('peminjaman_status'=>1),'peminjaman');

// mengalihkan halaman ke halaman data buku
redirect(base_url().'petugas/peminjaman');
}

```

Perhatikan pada method yang kita buat di atas. Pada method peminjaman\_selesai() kita mengupdate **status** buku yang dipinjam menjadi 1 (tersedia).

```

$where = array(
    'peminjaman_id' => $id
);

// mengambil data buku pada peminjaman ber id tersebut
$data = $this->m_data->edit_data($where,'peminjaman')->row();
$buku = $data->peminjaman_buku;

// mengembalikan status buku kembali ke tersedia (1)
$w = array(
    'id' => $buku
);
$d = array(
    'status' => 1
);
$this->m_data->update_data($w,$d,'buku');

```

Dan kemudian kita juga mengupdate **status** peminjaman menjadi 1 (selesai).

```

// mengubah status peminjaman menjadi selesai (1)
$this->m_data->update_data($where,array('peminjaman_status'=>1),'peminjaman');

```

Dan terakhir kita alihkan halaman kembali ke halaman peminjaman.

```

// mengalihkan halaman ke halaman data buku
redirect(base_url().'petugas/peminjaman');

```

Sekarang mari kita coba klik “selesai” pada transaksi peminjaman buku nomor 2, yaitu peminjaman yang di lakukan oleh anggota yang bernama “admiral”.

Data Peminjaman Buku						
No	Buku	Peminjam	Mulai Pinjam	Pinjam Sampai	Status	Opsi
1	Panduan Belajar JQuery Dari Dasar Sampai Mahir Untuk Pemula	Junaidi	26-06-2018	30-06-2018	Dipinjam	<span>Selesai</span> <span>Batalkan</span>
2	Tutorial Codeigniter Lengkap Membangun Sistem Informasi Perpustakaan	Admiral	26-06-2018	28-06-2018	Dipinjam	<span>Selesai</span> <span>Batalkan</span>

Show 10 entries Search:

Showing 1 to 2 of 2 entries Previous 1 Next

localhost/perpustakaan/petugas/peminjaman\_selesai/2

maka status data peminjaman akan menjadi seperti berikut.

Data Peminjaman Buku						
No	Buku	Peminjam	Mulai Pinjam	Pinjam Sampai	Status	Opsi
1	Panduan Belajar JQuery Dari Dasar Sampai Mahir Untuk Pemula	Junaidi	26-06-2018	30-06-2018	Dipinjam	<span>Selesai</span> <span>Batalkan</span>
2	Tutorial Codeigniter Lengkap Membangun Sistem Informasi Perpustakaan	Admiral	26-06-2018	28-06-2018	Selesai	-

Show 10 entries Search:

Showing 1 to 2 of 2 entries Previous 1 Next

jika proses peminjaman telah selesai, maka tombol "batalkan" dan tombol "selesai" tidak lagi muncul seperti pengecekan yang telah kita buat pada view v\_peminjaman.php.

## Membuat Filter Dan Cetak Laporan Peminjaman Buku

Setelah selesai membuat proses peminjaman buku, selanjutnya kita akan membuat filter laporan peminjaman buku berdasarkan tanggal dan membuat cetak laporannya.

Perhatikan pada menu "Laporan Peminjaman". Pada menu tersebut sudah kita buat link yang mengarahkan halaman ke method *peminjaman\_laporan()* pada controller Petugas.php.

```
<a class="nav-link" href="=base_url('petugas/peminjaman_laporan')?"&gt;&lt;i class="fa fa-book"&gt;&lt;/i&gt; Laporan Peminjaman&lt;/a&gt;</pre
```

Jadi sekarang buat sebuah method baru lagi dengan nama *peminjaman\_laporan()* pada controller Petugas.php. pada method *peminjaman\_laporan()* ini akan kita buat form untuk memfilter tanggal berapa sampai tanggal berapa transaksi peminjaman ingin dicetak.

### Application/controllers/Petugas.php

```
function peminjaman_laporan(){
    if(isset($_GET['tanggal_mulai']) && isset($_GET['tanggal_sampai'])){
        $mulai = $this->input->get('tanggal_mulai');
        $sampai = $this->input->get('tanggal_sampai');
        // mengambil data peminjaman berdasarkan tanggal mulai sampai tanggal sampai
        $data['peminjaman'] = $this->db->query("select * from peminjaman,buku,anggota where peminjaman.peminjaman_buku=buku.id and peminjaman.peminjaman_anggota=anggota.id and date(peminjaman_tanggal_mulai) >= '$mulai' and date(peminjaman_tanggal_mulai) <= '$sampai' order by peminjaman_id desc")->result();
    }else{
        // mengambil data peminjaman buku dari database | dan mengurutkan data dari id peminjaman terbesar ke terkecil (desc)
        $data['peminjaman'] = $this->db->query("select * from peminjaman,buku,anggota where peminjaman.peminjaman_buku=buku.id and peminjaman.peminjaman_anggota=anggota.id order by peminjaman_id desc")->result();
    }
    $this->load->view('petugas/v_header');
    $this->load->view('petugas/v_peminjaman_laporan',$data);
    $this->load->view('petugas/v_footer');
}
```

Perhatikan pada syntax di atas. Kita membuat pengecekan jika terdapat data GET tanggal\_mulai dan GET tanggal\_sampai, maka query yang kita jalankan adalah query yang memfilter data peminjaman berdasarkan tanggal. Jika tidak maka tampilkan semua data peminjaman.

Selanjutnya data yang kita dapat dari query tersebut kita parsing ke view v\_peminjaman\_laporan.php. karena kita akan menampilkan datanya dan membuat form filter tanggalnya pada view v\_peminjaman\_laporan.php tersebut.

Buat sebuah view baru dalam folder “petugas” dengan nama v\_peminjaman\_laporan.php.

### Application/views/petugas/v\_peminjaman\_laporan.php

```
<div class="container-fluid">
    <div class="card">
        <div class="card-header text-center">
            <h4>Laporan Peminjaman Buku</h4>
        </div>
        <div class="card-body">

            <br/>

            <div class="row">
                <div class="col-md-4">
                    <div class="card">
                        <div class="card-header text-center">
                            <h6>Filter Berdasarkan Tanggal</h6>
                        </div>
                        <div class="card-body">

                            <form method="get" action="">
                                <div class="form-group">
                                    <label class="font-weight-bold" for="tanggal_mulai">Tanggal Mulai Pinjam</label>
                                    <input type="date" class="form-control" name="tanggal_mulai" placeholder="Masukkan tanggal mulai pinjam">
                                </div>
                                <div class="form-group">
                                    <label class="font-weight-bold" for="tanggal_sampai">Tanggal Pinjam Sampai</label>
                                    <input type="date" class="form-control" name="tanggal_sampai" placeholder="Masukkan tanggal pinjam sampai">
                                </div>
                                <input type="submit" class="btn btn-primary" value="Filter">
                            </form>

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

        <br/>
        <?php
        // membuat tombol cetak jika data sudah di filter
        if(isset($_GET['tanggal_mulai']) && isset($_GET['tanggal_sampai'])){
            $mulai = $_GET['tanggal_mulai'];
            $sampai = $_GET['tanggal_sampai'];
        ?>
            <a class='btn btn-primary' target="_blank" href='<?php echo base_url().'petugas/peminjaman_cetak/?tanggal_mulai='.$mulai.'&tanggal_sampai='.$sa mpai ?>'><i class='fa fa-print'></i> CETAK</a>
        <?php
        }
        ?>
        <br/>
```

```

<br/>

<div class="table-responsive">
    <table class="table table-bordered table-striped table-hover table-
datatable">
        <thead>
            <tr>
                <th width="1%">No</th>
                <th>Buku</th>
                <th>Peminjam</th>
                <th>Mulai Pinjam</th>
                <th>Pinjam Sampai</th>
                <th>Status</th>
            </tr>
        </thead>
        <tbody>
            <?php
            $no = 1;
            foreach($peminjaman as $p) {
                ?>
                <tr>
                    <td><?php echo $no++; ?></td>
                    <td><?php echo $p->judul; ?></td>
                    <td><?php echo $p->nama; ?></td>
                    <td><?php echo date('d-m-Y', strtotime($p->peminjaman_tanggal_mulai)); ?></td>
                    <td><?php echo date('d-m-Y', strtotime($p->peminjaman_tanggal_sampai)); ?></td>
                    <td>
                        <?php
                            if($p->peminjaman_status == "1"){
                                echo "<div class='badge badge-success'>Selesai</div>";
                            }else if($p->peminjaman_status == "2"){
                                echo "<div class='badge badge-warning'>Dipinjam</div>";
                            }
                        ?>
                    </td>
                </tr>
            <?php
            }
            ?>
        </tbody>
    </table>
</div>

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

```

Pada view ini kita membuat sebuah form untuk memfilter data peminjaman. Yaitu form penginputan tanggal dari dan tanggal sampai. Yang maksudnya ingin memfilter data dari tanggal berapa dan sampai berapa.

```

<form method="get" action="">
    <div class="form-group">
        <label class="font-weight-bold" for="tanggal_mulai">Tanggal Mulai Pinjam</label>
        <input type="date" class="form-control" name="tanggal_mulai" placeholder="Masukkan tanggal mulai pinjam">
    </div>
    <div class="form-group">

```

```

        <label class="font-weight-bold" for="tanggal_sampai">Tanggal Pinjam Sampai</label>
        <input type="date" class="form-control" name="tanggal_sampai"
placeholder="Masukkan tanggal pinjam sampai">
</div>
<input type="submit" class="btn btn-primary" value="Filter">
</form>

```

Kita juga membuat pengecekan jika tanggal sudah dipilih, maka akan kita tampilkan tombol cetaknya.

```

<?php
// membuat tombol cetak jika data sudah di filter
if(isset($_GET['tanggal_mulai']) && isset($_GET['tanggal_sampai'])){
    $mulai = $_GET['tanggal_mulai'];
    $sampai = $_GET['tanggal_sampai'];
?
<a class='btn btn-primary' target="_blank" href='<?php echo
base_url().'petugas/peminjaman_cetak/?tanggal_mulai='.$mulai.'&tanggal_sampai='.$sa
mpai ?>'><i class='fa fa-print'></i> CETAK</a>
<?php
}
?>

```

Pada tombol cetak kita arahkan ke method `peminjaman_cetak()` pada controller `Petugas.php`. sambil mengirimkan data `tanggal_mulai` dan `tanggal_sampai`.

```

<?php
echo
base_url().'petugas/peminjaman_cetak/?tanggal_mulai='.$mulai.'&tanggal_sampai='.$sa
mpai;
?>

```

Nanti pada method `peminjaman_cetak()` akan kita tangkap data tanggal yang dikirim, dan langsung kita cetak pada viewnya.

Buat sebuah method baru lagi dengan nama `peminjaman_cetak()` pada controller `Petugas.php`.

### Application/controllers/Petugas.php

```

function peminjaman_cetak(){
    if(isset($_GET['tanggal_mulai']) && isset($_GET['tanggal_sampai'])){
        $mulai = $this->input->get('tanggal_mulai');
        $sampai = $this->input->get('tanggal_sampai');
        // mengambil data peminjaman berdasarkan tanggal mulai sampai tanggal
        sampai
        $data['peminjaman'] = $this->db->query("select * from
peminjaman,buku,anggota where peminjaman.peminjaman_buku=buku.id and
peminjaman.peminjaman_anggota=anggota.id and date(peminjaman_tanggal_mulai) >=
'$mulai' and date(peminjaman_tanggal_mulai) <= '$sampai' order by peminjaman_id
desc")->result();
        $this->load->view('petugas/v_peminjaman_cetak',$data);
    }else{

```

```

        redirect(base_url().'petugas/peminjaman');
    }
}

```

Pada method peminjaman\_cetak() inilah kita seleksi data peminjaman berdasarkan tanggal yang dipilih dan kemudian langsung kita tampilkan pada view `v_peminjaman_cetak.php`. pada view `v_peminjaman_cetak.php` ini akan kita tampilkan data peminjaman yang sudah kita filter berdasarkan tanggal.

Buat sebuah view baru dengan nama `v_peminjaman_cetak.php` dalam folder **petugas**.

### **Application/views/petugas/v\_peminjaman\_cetak.php**

```

<!DOCTYPE html>
<html>
<head>
    <title>Cetak Laporan Peminjaman Buku</title>
</head>
<body>

    <style type="text/css">
        table{
            border-collapse: collapse;
        }

        table th,td{
            border: 1px solid #000;
        }
    </style>

    <center>
        <h3>Laporan Peminjaman Buku</h3>
    </center>

    <table>
        <tr>
            <th width="1%">No</th>
            <th>Buku</th>
            <th>Peminjam</th>
            <th>Mulai Pinjam</th>
            <th>Pinjam Sampai</th>
            <th>Status</th>
        </tr>
        <?php
        $no = 1;
        foreach($peminjaman as $p) {
            ?>
            <tr>
                <td><?php echo $no++; ?></td>
                <td><?php echo $p->judul; ?></td>
                <td><?php echo $p->nama; ?></td>
                <td><?php echo date('d-m-Y', strtotime($p->peminjaman_tanggal_mulai)); ?></td>
                <td><?php echo date('d-m-Y', strtotime($p->peminjaman_tanggal_sampai)); ?></td>
            ?></tr>
            <td>
                <center>
                    <?php
                    if($p->peminjaman_status == "1") {

```

```

        echo "<div class='badge badge-success'>Selesai</div>";
    }else if($p->peminjaman_status == "2") {
        echo "<div class='badge badge-warning'>Dipinjam</div>";
    }
    ?>
</center>
</td>
</tr>
<?php
}
?>
</table>

<script type="text/javascript">
    window.print();
</script>

</body>
</html>

```

pada view ini kita menampilkan data yang telah di filter, kemudian kita cetak dengan fungsi javascript `window.print()`.

Sekarang akan kita coba jalankan pada browser.

No	Buku	Peminjam	Mulai Pinjam	Pinjam Sampai	Status
1	Panduan Belajar JQuery Dari Dasar Sampai Mahir Untuk Pemula	Junaidi	26-06-2018	30-06-2018	Dipinjam
2	Tutorial Codeigniter Lengkap Membangun Sistem Informasi Perpustakaan	Admiral	26-06-2018	28-06-2018	Selesai

masukkan data tanggal yang ingin difilter. Disini saya memasukkan tanggal 1/6/2018 sampai tanggal 30/6/2018. Dan klik filter.

Laporan Peminjaman Buku

Filter Berdasarkan Tanggal

Tanggal Mulai Pinjam  
mm/dd/yyyy

Tanggal Pinjam Sampai  
mm/dd/yyyy

**Filter**

**CETAK**

Show 10 entries Search: [ ]

No	Buku	Peminjam	Mulai Pinjam	Pinjam Sampai	Status
1	Panduan Belajar JQuery Dari Dasar Sampai Mahir Untuk Pemula	Junaidi	26-06-2018	30-06-2018	Dipinjam
2	Tutorial Codeigniter Lengkap Membangun Sistem Informasi Perpustakaan	Admiral	26-06-2018	28-06-2018	Selesai

Showing 1 to 2 of 2 entries Previous 1 Next

maka data pada tanggal yang di filter pun berhasil tampil. Silahkan klik cetak untuk melihat hasil data yang dicetaknya.

Print Total: 1 page

Cancel Save

Destination  Save as PDF Change...

Pages  All  e.g. 1-5, 8, 11-13

Layout Portrait More settings

Print using system dialog... (⌘P) Open PDF in Preview

Laporan Peminjaman Buku

No	Buku	Peminjam	Mulai Pinjam	Pinjam Sampai	Status
1	Panduan Belajar JQuery Dari Dasar Sampai Mahir Untuk Pemula	Junaidi	26-06-2018	30-06-2018	Dipinjam
2	Tutorial Codeigniter Lengkap Membangun Sistem Informasi Perpustakaan	Admiral	26-06-2018	28-06-2018	Selesai

nah, data peminjaman pun berhasil dicetak.

Sampai di sini, semua fungsi pada petugas sudah selesai kita buat, selanjutnya kita akan melanjutkan ke bagian halaman dashboard admin. Untuk melanjutkan fungsi-fungsi pada admin yang sempat kita tinggalkan.

# Melanjutkan Dashboard Admin

Sebelum kita membuat dashboard petugas dan fungsi-fungsi yang bisa dilakukan oleh petugas, kita telah terlebih dulu membuat dashboard admin dan membuat fungsi CRUD data petugas pada admin. Sekarang kita akan melanjutkan membuat fungsi-fungsi yang bisa dilakukan oleh admin.

Seperti yang telah dibahas pada saat merancang sistem informasi perpustakaan ini. Admin dapat mengelola data petugas (CRUD), dan kemudian admin hanya dapat melihat data buku, melihat data anggota, dan melihat data peminjaman dan laporan. Sedangkan petugas dapat mengelola data anggota, mengelola data buku dan mengelola peminjaman buku.

## Menampilkan Data Anggota Pada Admin

Pada sub judul ini kita akan menyelesaikan fungsi-fungsi yang bisa dilakukan oleh admin. Sebelumnya kita telah membuat CRUD petugas pada halaman admin. Sekarang kita akan melanjutkannya dengan membuat fungsi untuk menampilkan data anggota, buku, dan data peminjaman buku.

Silahkan login kembali sebagai admin. Pada menu anggota kita telah membuat link yang mengarahkan halaman ke fungsi anggota() pada controller Admin.php.

```
<a class="nav-link" href="=base_url('admin/anggota')?"&gt;&lt;i class="fa fa-users"&gt;&lt;/i&gt; Anggota&lt;/a&gt;</pre
```

Buat sebuah method baru dengan nama *anggota()* pada controller Admin.php.

### Application/controllers/Admin.php

```
// anggota
function anggota(){
    // mengambil data dari database
    $data['anggota'] = $this->m_data->get_data('anggota')->result();
    $this->load->view('admin/v_header');
    $this->load->view('admin/v_anggota',$data);
    $this->load->view('admin/v_footer');
}
```

Pada method *anggota()* di atas, kita akan menampilkan data anggota melalui view *v\_anggota.php* yang terletak dalam folder “admin”. Saya rasa teman-teman telah familiar dengan syntax di atas karena sudah berulang-ulang kali dibuat pada fungsi sebelumnya dan sudah berulang-ulang kali dijelaskan pada sub judul sebelumnya.

Buat sebuah view baru lagi dalam folder “admin” dengan nama *v\_anggota.php*.

## Application/views/admin/v\_anggota.php

```
<div class="container">
    <div class="card">
        <div class="card-header text-center">
            <h4>Data Anggota</h4>
        </div>
        <div class="card-body">

            <div class="table-responsive">
                <table class="table table-bordered table-striped table-hover table-
datatable">
                    <thead>
                        <tr>
                            <th width="1%">No</th>
                            <th>Nama</th>
                            <th>NIK</th>
                            <th>Alamat</th>
                            <th width="10%">Opsi</th>
                        </tr>
                    </thead>
                    <tbody>
                        <?php
                            $no = 1;
                            foreach($anggota as $a) {
                                ?>
                                <tr>
                                    <td><?php echo $no++; ?></td>
                                    <td><?php echo $a->nama; ?></td>
                                    <td><?php echo $a->nik; ?></td>
                                    <td><?php echo $a->alamat; ?></td>
                                    <td>
                                        <a href="php echo base_url().'admin/anggota_kartu/'.$a-&gt;id; ?&gt;" class="btn btn-sm btn-primary"&gt;&lt;i class="fa fa-address-card" style="font-size: 1.5em; color: #007bff; margin-right: 5px;" Cetak Kartu</a>
                                    </td>
                                </tr>
                            <?php
                            }
                            ?>
                        </tbody>
                    </table>
                </div>
            </div>
        </div>
    </div>
```

Sekarang kita akan melihat hasilnya dengan klik pada menu anggota.

The screenshot shows a web-based library management system. At the top, there's a navigation bar with links for Dashboard, Petugas, Anggota, Buku, Laporan Peminjaman, Ganti Password, and a sign-out button ('KELUAR'). Below the navigation is a table titled 'Data Anggota' showing two records:

No	Nama	NIK	Alamat	Opsi
1	Junaidi	323234234	Jl. Kenari No.91 Jakarta	<button>Cetak Kartu</button>
2	Admiral	2147483647	Jl. Mutan Putih no.73 Jakarta	<button>Cetak Kartu</button>

Below the table, it says 'Showing 1 to 2 of 2 entries'. There are also 'Previous' and 'Next' buttons.

kita tidak lagi membuat tombol edit dan hapus pada data anggota. Karena sesuai perancangan, admin hanya dapat melihat data anggota dan mencetak kartu anggota.

## Membuat Cetak Kartu Anggota Pada Admin

Pada data anggota, sudah kita buat tombol “Cetak Kartu”. Pada saat tombol cetak kartu diklik akan dialihkan ke method anggota\_kartu() sambil mengirimkan data id anggota yang ingin di cetak kartunya.

Buat method anggota\_kartu() pada controller Admin.php.

### Application/controllers/Admin.php

```
function anggota_kartu($id){
    $where = array('id' => $id);
    // mengambil data dari database sesuai id
    $data['anggota'] = $this->m_data->edit_data($where, 'anggota')->result();
    $this->load->view('admin/v_anggota_kartu', $data);
}
// akhir anggota
```

Seperti pada saat membuat cetak kartu anggota pada halaman petugas, kita ambil data anggota yang memiliki id yang di pilih, kemudian kita parsing ke view v\_anggota\_kartu.php dalam folder “admin”.

### Application/views/admin/v\_anggota\_kartu.php

```
<!DOCTYPE html>
```

```

<html>
<head>
    <title>Cetak Kartu Anggota</title>
</head>
<body>

    <style type="text/css">
        .card{
            border: 1px solid #000;
            width: 450px;
        }

        .card-header{
            border-bottom: 1px solid #000;
            text-align: center;
            font-weight: bold;
            padding: 10px;
        }

        .card-body{
            padding: 20px;
        }
    </style>

    <div class="card">
        <div class="card-header">
            KARTU ANGGOTA PERPUSTAKAAN MALASNGODING
        </div>
        <div class="card-body">
            <div class="container">
                <table class="table table-borderless table-sm fs-2">
                    <?php
                    $no = 1;
                    foreach($anggota as $a) {
                        ?>
                        <tr>
                            <td width="14%">Nomor</td>
                            <td width="2%">:</td>
                            <td><?php echo 10000+$a->id; ?></td>
                        </tr>
                        <tr>
                            <td>Nama</td>
                            <td>:</td>
                            <td><?php echo $a->nama; ?></td>
                        </tr>
                        <tr>
                            <td>NIK</td>
                            <td>:</td>
                            <td><?php echo $a->nik; ?></td>
                        </tr>
                        <tr>
                            <td>Alamat</td>
                            <td>:</td>
                            <td><?php echo $a->alamat; ?></td>
                        </tr>
                        <?php
                    }
                    ?>
                </table>
            </div>
        </div>
    </div>

    <script type="text/javascript">

```

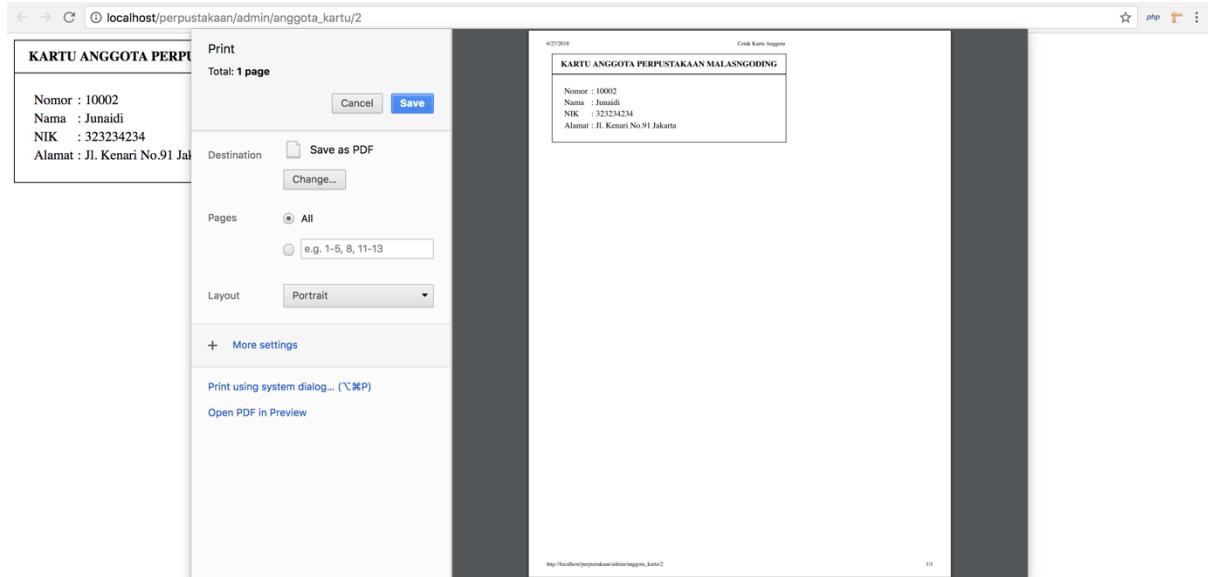
```

        window.print();
    </script>

</body>
</html>

```

Sekarang coba klik pada tombol “Cetak Kartu”. Maka kartu anggota akan dicetak.



selanjutnya kita akan membuat fungsi untuk menampilkan data buku pada halaman admin.

## Menampilkan Data Buku Pada Halaman Admin

Untuk menampilkan data buku pada halaman admin, caranya sama saja seperti menampilkan data seperti yang telah dijelaskan pada sub judul-sub judul sebelumnya.

Pada menu buku di halaman admin kita akan menampilkan data buku pada saat diklik, buat sebuah method *buku()* pada controller Admin.php.

### Application/controllers/Admin.php

```

// buku
function buku() {
    // mengambil data dari database
    $data['buku'] = $this->m_data->get_data('buku')->result();
    $this->load->view('admin/v_header');
    $this->load->view('admin/v_buku',$data);
    $this->load->view('admin/v_footer');
}
// akhir buku

```

Data buku akan kita tampilkan pada view v\_buku.php dalam folder admin seperti pada syntax di atas.

Buat view baru lagi dengan nama *v\_buku.php* dalam folder admin.

### **Application/views/admin/v\_buku.php**

```
<div class="container-fluid">
<div class="card">
    <div class="card-header text-center">
        <h4>Data Buku</h4>
    </div>
    <div class="card-body">

        <div class="table-responsive">
            <table class="table table-bordered table-striped table-hover table-
datatable">
                <thead>
                    <tr>
                        <th width="1%">No</th>
                        <th>Judul Buku</th>
                        <th>Tahun Terbit</th>
                        <th>Penulis</th>
                        <th width="10%">Status</th>
                    </tr>
                </thead>
                <tbody>
                    <?php
                    $no = 1;
                    foreach($buku as $b) {
                        ?>
                        <tr>
                            <td><?php echo $no++; ?></td>
                            <td><?php echo $b->judul; ?></td>
                            <td><?php echo $b->tahun; ?></td>
                            <td><?php echo $b->penulis; ?></td>
                            <td>
                                <?php
                                if($b->status == "1") {
                                    echo "<span class='badge badge-success'>Tersedia</span>";
                                } else if($b->status == "2") {
                                    echo "<span class='badge badge-warning'>Sedang
Dipinjam</span>";
                                }
                                ?>
                            </td>
                        </tr>
                    <?php
                }
                ?>
            </tbody>
        </table>
    </div>
</div>
</div>
```

Sekarang coba klik pada menu “Buku”.

The screenshot shows a web application interface for a library system. At the top, there is a header bar with links for Dashboard, Petugas, Anggota, Buku, Laporan Peminjaman, Ganti Password, and a sign-out button labeled "KELUAR". Below the header, the title "Data Buku" is displayed. A table lists five books with columns for No, Judul Buku, Tahun Terbit, Penulis, and Status. The books are: 1. Tutorial Codeigniter Lengkap Membangun Sistem Informasi Perpustakaan (2018, Diki Alfarabi Hadi, Tersedia), 2. Belajar cepat codeigniter dengan studi kasus membuat aplikasi rental mobil (2017, Diki Alfarabi Hadi, Tersedia), 3. Belajar PHP dan MySQLi dari dasar sampai membuat sistem informasi laundry (2018, Diki Alfarabi Hadi, Tersedia), 4. Panduan Belajar JQuery Dari Dasar Sampai Mahir Untuk Pemula (2016, Diki Alfarabi Hadi, Sedang Dipinjam), and 5. Belajar HTML dan CSS dari Dasar Sampai Mahir (2016, Diki Alfarabi Hadi, Tersedia). The status column uses color-coded boxes: green for 'Tersedia' and yellow for 'Sedang Dipinjam'. At the bottom of the table, it says "Showing 1 to 5 of 5 entries" and has navigation buttons for Previous, Next, and page number 1.

data buku telah berhasil ditampilkan sekalian dengan status bukunya.

## Menampilkan & Cetak Laporan Peminjaman Pada Admin

Seperti yang teman-teman lihat pada menu “Peminjaman”, halaman akan dialihkan ke method *peminjaman\_laporan()* pada controller Admin.php.

Buat method *peminjaman\_laporan()* pada controller Admin.php

### Application/controllers/Admin.php

```
// peminjaman
function peminjaman_laporan(){
    if(isset($_GET['tanggal_mulai']) && isset($_GET['tanggal_sampai'])){
        $mulai = $this->input->get('tanggal_mulai');
        $sampai = $this->input->get('tanggal_sampai');
        // mengambil data peminjaman berdasarkan tanggal mulai sampai tanggal
        // sampai
        $data['peminjaman'] = $this->db->query("select * from
        peminjaman,buku,anggota where peminjaman.peminjaman_buku=buku.id and
        peminjaman.peminjaman_anggota=anggota.id and date(peminjaman_tanggal_mulai) >=
        '$mulai' and date(peminjaman_tanggal_mulai) <= '$sampai' order by peminjaman_id
        desc")->result();
    }else{
        // mengambil data peminjaman buku dari database | dan mengurutkan data
        // dari id peminjaman terbesar ke terkecil (desc)
        $data['peminjaman'] = $this->db->query("select * from
        peminjaman,buku,anggota where peminjaman.peminjaman_buku=buku.id and
        peminjaman.peminjaman_anggota=anggota.id order by peminjaman_id desc")->result();
    }
}
```

```

    }
    $this->load->view('admin/v_header');
    $this->load->view('admin/v_peminjaman_laporan',$data);
    $this->load->view('admin/v_footer');
}

```

Masih sama seperti penjelasan pada saat membuat laporan peminjaman pada petugas.untuk pengecekannya juga sama, jika di submit data tanggal, maka query yang di jalankan adalah query yang memfilter data peminjaman berdasarkan tanggal, jika tidak maka query yang di gunakan adalah query yang menampilkan semua data peminjaman.

Laporan peminjaman akan kita buat pada view *v\_peminjaman\_laporan.php* seperti pada syntax di atas.

Buat view baru dengan nama *v\_peminjaman\_laporan.php* dalam folder admin.

### **Application/views/admin/v\_peminjaman\_laporan.php**

```

<div class="container-fluid">
    <div class="card">
        <div class="card-header text-center">
            <h4>Laporan Peminjaman Buku</h4>
        </div>
        <div class="card-body">

            <br/>

            <div class="row">
                <div class="col-md-4">
                    <div class="card">
                        <div class="card-header text-center">
                            <h6>Filter Berdasarkan Tanggal</h6>
                        </div>
                        <div class="card-body">

                            <form method="get" action="">
                                <div class="form-group">
                                    <label class="font-weight-bold" for="tanggal_mulai">Tanggal Mulai Pinjam</label>
                                    <input type="date" class="form-control" name="tanggal_mulai" placeholder="Masukkan tanggal mulai pinjam">
                                </div>
                                <div class="form-group">
                                    <label class="font-weight-bold" for="tanggal_sampai">Tanggal Pinjam Sampai</label>
                                    <input type="date" class="form-control" name="tanggal_sampai" placeholder="Masukkan tanggal pinjam sampai">
                                </div>
                                <input type="submit" class="btn btn-primary" value="Filter">
                            </form>

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

```

```

<?php
// membuat tombol cetak jika data sudah di filter
if(isset($_GET['tanggal_mulai']) && isset($_GET['tanggal_sampai'])){
    $mulai = $_GET['tanggal_mulai'];
    $sampai = $_GET['tanggal_sampai'];
    ?>
    <a class='btn btn-primary' target="_blank" href='<?php echo base_url().'admin/peminjaman_cetak/?tanggal_mulai='.$mulai.'&tanggal_sampai='.$sampai ?>'><i class='fa fa-print'></i> CETAK</a>
    <?php
}
?>
<br/>
<br/>

<div class="table-responsive">
    <table class="table table-bordered table-striped table-hover table-datatatable">
        <thead>
            <tr>
                <th width="1%">No</th>
                <th>Buku</th>
                <th>Peminjam</th>
                <th>Mulai Pinjam</th>
                <th>Pinjam Sampai</th>
                <th>Status</th>
            </tr>
        </thead>
        <tbody>
            <?php
            $no = 1;
            foreach($peminjaman as $p) {
                ?>
                <tr>
                    <td><?php echo $no++; ?></td>
                    <td><?php echo $p->judul; ?></td>
                    <td><?php echo $p->nama; ?></td>
                    <td><?php echo date('d-m-Y', strtotime($p->peminjaman_tanggal_mulai)); ?></td>
                    <td><?php echo date('d-m-Y', strtotime($p->peminjaman_tanggal_sampai)); ?></td>
                    <td>
                        <?php
                        if($p->peminjaman_status == "1"){
                            echo "<div class='badge badge-success'>Selesai</div>";
                        }else if($p->peminjaman_status == "2"){
                            echo "<div class='badge badge-warning'>Dipinjam</div>";
                        }
                        ?>
                    </td>
                </tr>
            <?php
            }
            ?>
        </tbody>
    </table>
</div>

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

```

Kemudian buat method untuk menghandle cetak data peminjaman. Seperti pada tombol "Cetak" di atas. Akan dialihkan ke method *peminjaman\_cetak()*.

```
<a class='btn btn-primary' target='_blank' href='<?php echo base_url().'admin/peminjaman_cetak/?tanggal_mulai='.$mulai.'&tanggal_sampai='.$sampai ?>'><i class='fa fa-print'></i> CETAK</a>
```

Buat method *peminjaman\_cetak()* pada controller Admin.php.

### Application/controllers/Admin.php

```
function peminjaman_cetak(){
    if(isset($_GET['tanggal_mulai']) && isset($_GET['tanggal_sampai'])){
        $mulai = $this->input->get('tanggal_mulai');
        $sampai = $this->input->get('tanggal_sampai');
        // mengambil data peminjaman berdasarkan tanggal mulai sampai tanggal sampai
        $data['peminjaman'] = $this->db->query("select * from peminjaman,buku,anggota where peminjaman.peminjaman_buku=buku.id and peminjaman.peminjaman_anggota=anggota.id and date(peminjaman_tanggal_mulai) >= '$mulai' and date(peminjaman_tanggal_mulai) <= '$sampai' order by peminjaman_id desc")->result();
        $this->load->view('admin/v_peminjaman_cetak',$data);
    }else{
        redirect(base_url().'admin/peminjaman');
    }
}
// akhir peminjaman
```

Cetak laporannya akan kita buat pada view *v\_peminjaman\_cetak.php* pada folder "admin".

Buat view baru lagi dengan nama *v\_peminjaman\_cetak.php* pada folder admin.

### Application/views/admin/v\_peminjaman\_cetak.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Cetak Laporan Peminjaman Buku</title>
</head>
<body>

    <style type="text/css">
        table{
            border-collapse: collapse;
        }

        table th,td{
            border: 1px solid #000;
        }
    </style>

    <center>
        <h3>Laporan Peminjaman Buku</h3>
    </center>
```

```


| No                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Buku | Peminjam | Mulai Pinjam | Pinjam Sampai | Status |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|----------|--------------|---------------|--------|
| <?php<br>\$no = 1;<br>foreach(\$peminjaman as \$p) {<br>?><br><tr><br><td><?php echo \$no++; ?></td><br><td><?php echo \$p->judul; ?></td><br><td><?php echo \$p->nama; ?></td><br><td><?php echo date('d-m-Y', strtotime(\$p->peminjaman_tanggal_mulai)); ?></td><br><td><?php echo date('d-m-Y', strtotime(\$p->peminjaman_tanggal_sampai)); ?></td><br></tr><br><td><br><center><br><?php<br>if(\$p->peminjaman_status == "1") {<br>echo "<div class='badge badge-success'>Selesai</div>";<br>} else if(\$p->peminjaman_status == "2") {<br>echo "<div class='badge badge-warning'>Dipinjam</div>";<br>}<br>?><br></center><br></td><br></tr><br><?php<br>}<br>?><br></table><br><br><script type="text/javascript"><br>window.print();<br></script><br><br></body><br></html> |      |          |              |               |        |


```

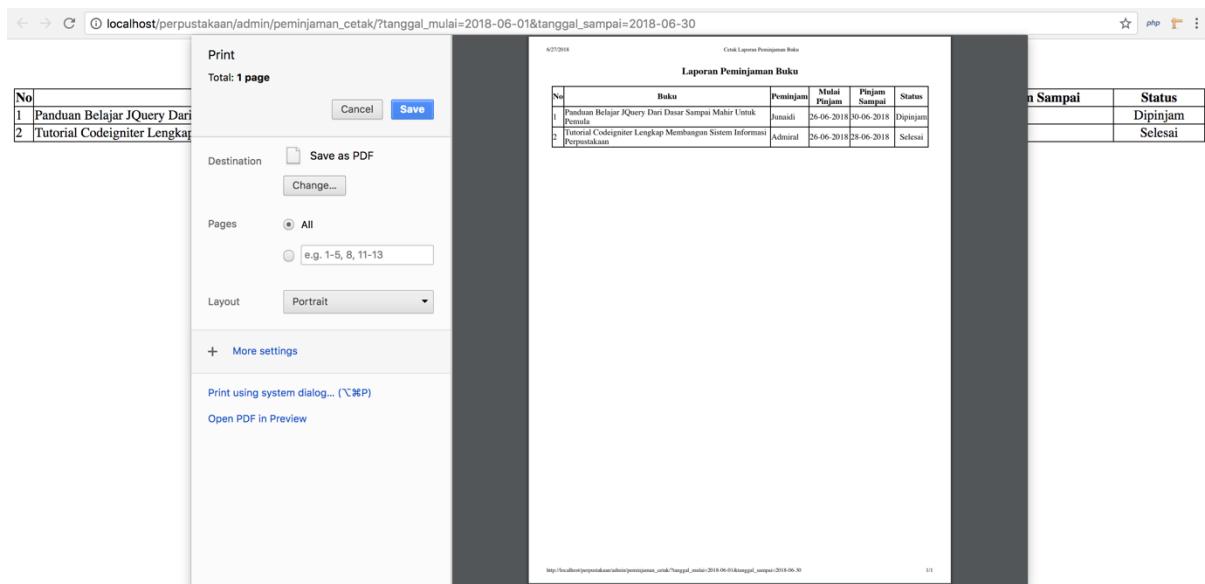
Oke selesai. Sekarang coba klik pada menu “Peminjaman”.

The screenshot shows a web application interface for a library management system. At the top, there's a navigation bar with links for Dashboard, Petugas, Anggota, Buku, Laporan Peminjaman, Ganti Password, and a sign-out link. Below the navigation is a title 'Laporan Peminjaman Buku'. A sidebar on the left contains a 'Filter Berdasarkan Tanggal' section with two date input fields: 'Tanggal Mulai Pinjam' (06/01/2018) and 'Tanggal Pinjam Sampai' (06/30/2018). A red arrow points from the text 'dan klik filter.' to the 'Filter' button. Another red arrow points from the text 'Maka data akan tampil sesuai dengan tanggal yang ditentukan.' to the 'CETAK' button. Below the filter section is a table with columns: No, Buku, Peminjam, Mulai Pinjam, Pinjam Sampai, and Status. The table shows two entries: 'Panduan Belajar JQuery Dari Dasar Sampai Mahir Untuk Pemula' by Junaidi and 'Tutorial Codeigniter Lengkap Membangun Sistem Informasi Perpustakaan' by Admiral. The 'Status' column indicates 'Dipinjam' for the first entry and 'Selesai' for the second.

dan klik filter. Maka data akan tampil sesuai dengan tanggal yang ditentukan. Dan akan muncul tombol cetak.

This screenshot shows the same application after applying the date filters. The 'Filter' button has been clicked, and now a 'CETAK' button is visible below the filter section. A red arrow points from the text 'akan muncul tombol cetak.' to this 'CETAK' button. The rest of the interface is identical to the previous screenshot, including the table of book lending records.

klik "Cetak".



cetak data peminjaman pun selesai kita buat.

## Membuat Ringkasan Pada Dashboard

Pada tahap ini kita akan membuat ringkasan jumlah data. Misalnya jumlah data buku, jumlah anggota, jumlah peminjaman dan jumlah petugas. Kita akan membuat ringkasan ini pada halaman dashboard admin dan halaman dashboard petugas.

### Membuat Ringkasan Pada Dashboard Admin

Kita mulai dengan memodifikasi halaman dashboard admin. Buka file `v_index.php` yang terletak pada folder admin.

#### Application/views/admin/v\_index.php

```
<div class="container">
    <div class="jumbotron text-center">
        <div class="col-sm-8 mx-auto">
            <h1>Selamat datang!</h1>
            <p>Ini merupakan contoh sistem informasi perpustakaan hasil dari tutorial <b>ebook tutorial codeigniter lengkap dengan studi kasus membuat sistem informasi perpustakaan</b>.</p>
            <p>Anda telah login sebagai <b><?php echo $this->session-</b><?><?php echo userdata('username'); ?></b> [admin].</p>
        </div>
    </div>
</div>
```

```

<div class="row">
    <div class="col-md-3">
        <div class="card bg-primary text-white">
            <div class="card-body">
                <h1>
                    <?php echo $this->m_data->get_data('buku')->num_rows(); ?>
                    <div class="pull-right">

                        <i class="fa fa-book"></i>
                    </div>
                </h1>
                Jumlah Buku
            </div>
        </div>
    <div class="col-md-3">
        <div class="card bg-danger text-white">
            <div class="card-body">
                <h1>
                    <?php echo $this->m_data->get_data('anggota')->num_rows(); ?>
                    <div class="pull-right">

                        <i class="fa fa-users"></i>
                    </div>
                </h1>
                Jumlah Anggota
            </div>
        </div>
    </div>
    <div class="col-md-3">
        <div class="card bg-warning text-white">
            <div class="card-body">
                <h1>
                    <?php echo $this->m_data->get_data('peminjaman')->num_rows(); ?>
                    <div class="pull-right">

                        <i class="fa fa-book"></i>
                    </div>
                </h1>
                Jumlah Total Peminjaman
            </div>
        </div>
    </div>
    <div class="col-md-3">
        <div class="card bg-primary text-white">
            <div class="card-body">
                <h1>
                    <?php echo $this->m_data->get_data('petugas')->num_rows(); ?>
                    <div class="pull-right">

                        <i class="fa fa-user"></i>
                    </div>
                </h1>
                Jumlah Petugas
            </div>
        </div>
    </div>
</div>

```

Setelah di save, sekarang coba kita login sebagai admin untuk melihat hasilnya.

The screenshot shows a web browser window with the URL 'localhost/perpustakaan/admin'. The page title is 'SI Perpustakaan'. The top navigation bar includes links for Dashboard, Petugas, Anggota, Buku, Laporan Peminjaman, and Ganti Password. On the right, it shows 'Halo, admin [admin]' and a 'KELUAR' button. The main content area features a large 'Selamat datang!' message, a note about the system being a tutorial example, and a message confirming login as 'admin [admin]'. Below this are four summary cards: 'Jumlah Buku' (5 books, blue card), 'Jumlah Anggota' (2 members, red card), 'Jumlah Total Peminjaman' (2 loans, yellow card), and 'Jumlah Petugas' (1 staff, blue card).

nah, ringkasan data pun berhasil tampil pada halaman dashboard admin. Akan saya jelaskan sedikit. Silahkan perhatikan pada element card pada syntax di atas. Di sana kita membuat card untuk masing-masing jumlah sata. Seperti misalnya.

```
<div class="card bg-primary text-white">
  <div class="card-body">
    <h1>
      <?php echo $this->m_data->get_data('buku')->num_rows(); ?>
      <div class="pull-right">
        <i class="fa fa-book"></i>
      </div>
    </h1>
    Jumlah Buku
  </div>
</div>
```

Pada card jumlah buku di atas kita menghitung jumlah data yang ada pada table buku.

```
<?php echo $this->m_data->get_data('buku')->num_rows(); ?>
```

Beginu juga dengan jumlah yang lain seperti jumlah anggota, jumlah peminjaman dan jumlah petugas.

## Membuat Ringkasan Pada Dashboard Petugas

Setelah selesai membuat ringkasan jumlah pada halaman admin, selanjutnya kita akan membuat ringkasan pada halaman petugas. Caranya sama saja dengan membuat ringkasan jumlah pada halaman admin.

Buka file v\_index.php pada folder petugas.

### Application/views/petugas/v\_index.php

```
<div class="container">
    <div class="jumbotron text-center">
        <div class="col-sm-8 mx-auto">
            <h1>Selamat datang!</h1>
            <p>Ini merupakan contoh sistem informasi perpustakaan hasil dari tutorial <b>ebook tutorial codeigniter lengkap dengan studi kasus membuat sistem informasi perpustakaan</b>.</p>
            <p>
                Anda telah login sebagai <b><?php echo $this->session->userdata('nama'); ?></b> [petugas].
            </p>
        </div>
    </div>

    <div class="row">
        <div class="col-md-3">
            <div class="card bg-primary text-white">
                <div class="card-body">
                    <h1>
                        <?php echo $this->m_data->get_data('buku')->num_rows(); ?>
                        <div class="pull-right">
                            <i class="fa fa-book"></i>
                        </div>
                    </h1>
                    Jumlah Buku
                </div>
            </div>
        </div>
        <div class="col-md-3">
            <div class="card bg-danger text-white">
                <div class="card-body">
                    <h1>
                        <?php echo $this->m_data->get_data('anggota')->num_rows(); ?>
                        <div class="pull-right">
                            <i class="fa fa-users"></i>
                        </div>
                    </h1>
                    Jumlah Anggota
                </div>
            </div>
        </div>
        <div class="col-md-3">
            <div class="card bg-warning text-white">
                <div class="card-body">
                    <h1>
                        <?php echo $this->m_data->get_data('peminjaman')->num_rows(); ?>
                        <div class="pull-right">
                            <i class="fa fa-book"></i>
                        </div>
                    </h1>
                </div>
            </div>
        </div>
    </div>
</div>
```

```

        </div>
    </h1>
    Jumlah Total Peminjaman
</div>
</div>
<div class="col-md-3">
    <div class="card bg-primary text-white">
        <div class="card-body">
            <h1>
                <?php echo $this->m_data->get_data('petugas')->num_rows(); ?>
                <div class="pull-right">

                    <i class="fa fa-user"></i>
                </div>
            </h1>
            Jumlah Petugas
        </div>
    </div>
</div>

</div>
</div>

```

Setelah disimpan, coba login sebagai petugas untuk melihat hasilnya.

Selamat datang!

Ini merupakan contoh sistem informasi perpustakaan hasil dari tutorial **ebook tutorial codeigniter lengkap dengan studi kasus membuat sistem informasi perpustakaan**.

Anda telah login sebagai **Jamaludin Ishak** [petugas].

5		Jumlah Buku
2		Jumlah Anggota
2		Jumlah Total Peminjaman
1		Jumlah Petugas

selesai.

## **Penutup**

Selesai sudah pembahasan cara pembuatan sistem informasi perpustakaan sederhana dalam ebook “Tutorial Codeigniter Lengkap Dengan Studi Kasus Membuat Sistem Informasi Perpustakaan” ini. Kita telah belajar Codeigniter dari sangat dasar sampai membuat sistem informasi sederhana yang siap pakai.

Di mulai dari membuat login multi user yang melibatkan admin dan petugas sebagai pengguna. Membuat hak aksesnya masing-masing seperti perancangan yang telah kita bahas pada bab sebelumnya. Yaitu admin dapat mengelola data petugas seperti menambah petugas, mengubah dan menghapus petugas, dan admin hanya dapat melihat semua data seperti data buku, anggota dan laporan peminjaman tanpa bisa mengubah-ubahnya. Sesuai perancangan yang kita buat, hanya petugas yang dapat mengelola data peminjaman buku, dimulai dari mengelola data buku, data anggota sampai dengan mengelola data peminjaman seperti menginput peminjaman baru, membatalkan peminjaman serta mengkonfirmasi peminjaman jika sudah selesai.

Diharapkan ebook ini dapat bermanfaat bagi teman-teman yang sedang mempelajari codeigniter dan bootstrap. Teman-teman bisa mengembangkan lagi sistem informasi perpustakaan ini sesuai dengan kebutuhan dengan berbekal materi-materi dan penjelasan dari ebook ini. Karena dalam ebook ini sudah sangat lengkap dijelaskan segala proses yang dibutuhkan dalam membuat sistem informasi atau aplikasi, tinggal teman-teman sendiri yang bereksperimen dengan menambah fitur atau mengembangkannya lagi.

# **Selesai**

## Tutorial Rekomendasi Lainnya

Kami merekomendasikan beberapa tutorial lanjutan yang bisa menunjang proses pembelajaran teman-teman tentang PHP maupun codeigniter.



**Belajar PHP & MySQLi dari dasar sampai membuat sistem informasi laundry**

**Belajar cepat Codeigniter dengan studi kasus membuat aplikasi rental mobil**

Untuk lebih lengkapnya silahkan kunjungi website [www.malasngoding.com](http://www.malasngoding.com) atau bisa mengirimkan email ke malasngoding@gmail.com.