



CodeIgniter



Bootstrap

EBOOK

BELAJAR CEPAT CODEIGNITER

Studi kasus membuat aplikasi rental mobil

- + FULL SOURCE CODE
- + Fitur denda keterlambatan otomatis
- + Live Search & Pagination (DataTable)
- + md5 Login Security
- + Export Laporan (Print, PDF)



Diki Alfarabi Hadi, S.T.

WWW.MALASNGODING.COM

*“Ilmu itu bukan yang di hafal,
tetapi yang memberi manfaat”*

- Imam Syafi'i -

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. Untuk mengetahui informasi lebih lanjut tentang penulis, silahkan mengunjungi <https://www.malasngoding.com>. Atau mengirimkan email ke 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 "**Belajar Cepat Codeigniter Dengan Studi Kasus Membuat Aplikasi Rental Mobil**" 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	6
Dasar Bagian 1 : Berkenalan Dengan Framework Codeigniter	7
Pengertian MVC	8
Kelebihan Framework Codeigniter	9
Dasar Bagian 2 : Cara Instalasi Framework Codeigniter	10
Dasar Bagian 3 : Mengenal Uri Segment Pada Codeigniter	13
Dasar Bagian 4 : Pengenalan Controller Codeigniter.....	17
Pengenalan Controller	17
Cara Membuat Controller Pada Codeigniter.....	19
Dasar Bagian 5 : Pengenalan View Pada Codeigniter.....	23
Pengenalan View.....	23
Cara Membuat View Pada Codeigniter	23
Cara Parsing Data ke View Codeigniter	26
Dasar Bagian 6 : Membuat Template Web Sederhana (Templating) Pada Codeigniter.....	29
Membuat Template Web (Templating) Dengan Codeigniter	29
Setting Base Url pada Codeigniter	32
Membuat Halaman Template Dinamis Dengan Codeigniter	34
Kesimpulan	39
Dasar Bagian 7 : Form Validation Pada Codeigniter.....	40
Membuat Form Validation Pada Codeigniter	40
Dasar Bagian 8 : Upload File Dengan Codeigniter	45
Membuat Upload File Dengan Codeigniter	45
Dasar Bagian 9 : Membuat Download File Dengan Codeigniter.....	53

Membuat Download File Dengan Codeigniter	53
Membuat Download File Kita Tentukan Sendiri	54
Membuat Download File Yang Sudah Tersedia	54
Contoh Membuat Download File Dengan Codeigniter	55
Dasar Bagian 10 : Membuat Library Codeigniter.....	57
Penjelasan.....	59
Dasar Bagian 11 : Menghilangkan Index.php Pada Codelgniter	61
Menghilangkan Index.php Pada Codelgniter.....	62
Dasar Bagian 12 : Mengenal Model, Koneksi Database Codeigniter dan MySQL ..	64
Membuat Koneksi Database Codeigniter dan MySQL	64
Pengenalan Model Pada Codelgniter.....	68
Dasar Bagian 13 : Membuat CRUD Dengan Codeigniter	72
Koneksi Database	72
Menampilkan Data Dari Database.....	74
Menginput Data Ke Database	78
Menghapus Data Dari Database	82
Mengupdate Data Pada Database	87
Dasar Bagian 14 : Membuat Pagination Dengan Codelgniter	93
Dasar Bagian 15 : Membuat Login Dengan Codeigniter	99
Persiapan Membuat Login Dengan Codeigniter.....	99
Menghubungkan Codeigniter dengan MySQL.....	102
Membuat Form Login Dengan Codeigniter.....	102
Tutorial Codeigniter Lanjutan Studi Kasus Membuat Aplikasi Rental Mobil	110
Instalasi Codeigniter 3.....	112
Persiapan Membuat Aplikasi Rental Mobil	113
Setting Base URL.....	113
Load Library, Helper, dan segala yang dibutuhkan	114
Merancang & Membuat Database Aplikasi Rental Mobil dan Konfigurasi database di Codeigniter	116

Konfigurasi database pada codeigniter.....	122
Membuat Model Yang Terstruktur dan Efisien Untuk CRUD	124
Menghilangkan index.php dari URL Codeigniter.....	129
Integrasi Codeigniter dan Bootstrap.....	131
Membuat login dengan codeigniter, md5 & Pesan Notifikasi	132
Menyiapkan Halaman Dashboard Admin	141
Membuat Logout	153
Membuat Fitur Ganti Password Admin	154
Membuat CRUD Data Mobil	160
Menampilkan Data Mobil dari database.....	160
Input Data Mobil ke Database	163
Update Data Mobil di Database.....	167
Menghapus Data Mobil dari Database	171
Membuat CRUD Data Kostumer	174
Menampilkan Data Kostumer dari database	174
Input Data Kostumer Ke Database.....	177
Update Data Kostumer Di Database	181
Menghapus Data Kostumer dari Database.....	186
Membuat Proses Transaksi Rental Mobil	188
Menampilkan Data Transaksi Rental	189
Menginput Data Transaksi Rental Mobil	193
Membuat Batalkan Transaksi	203
Membuat Transaksi Selesai (Konfirmasi)	204
Membuat Filter dan Cetak Laporan Rental Mobil	215
Menampilkan Data Transaksi Berdasarkan Filter Tanggal.....	220
Membuat Print Laporan.....	221
Membuat Cetak Laporan Ke PDF Menggunakan DOMPDF	227
Selesai.....	235

Tutorial Codeigniter dasar

Dasar Bagian 1 : Berkenalan Dengan Framework Codeigniter

Selamat datang di tutorial codeigniter dasar di ebook “ Belajar Cepat CodeIgniter Lengkap Dengan Studi Kasus Membuat Aplikasi Rental Mobil + FULL SOURCE CODE ”. 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 dijelaskan 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 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 berhubungan 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>

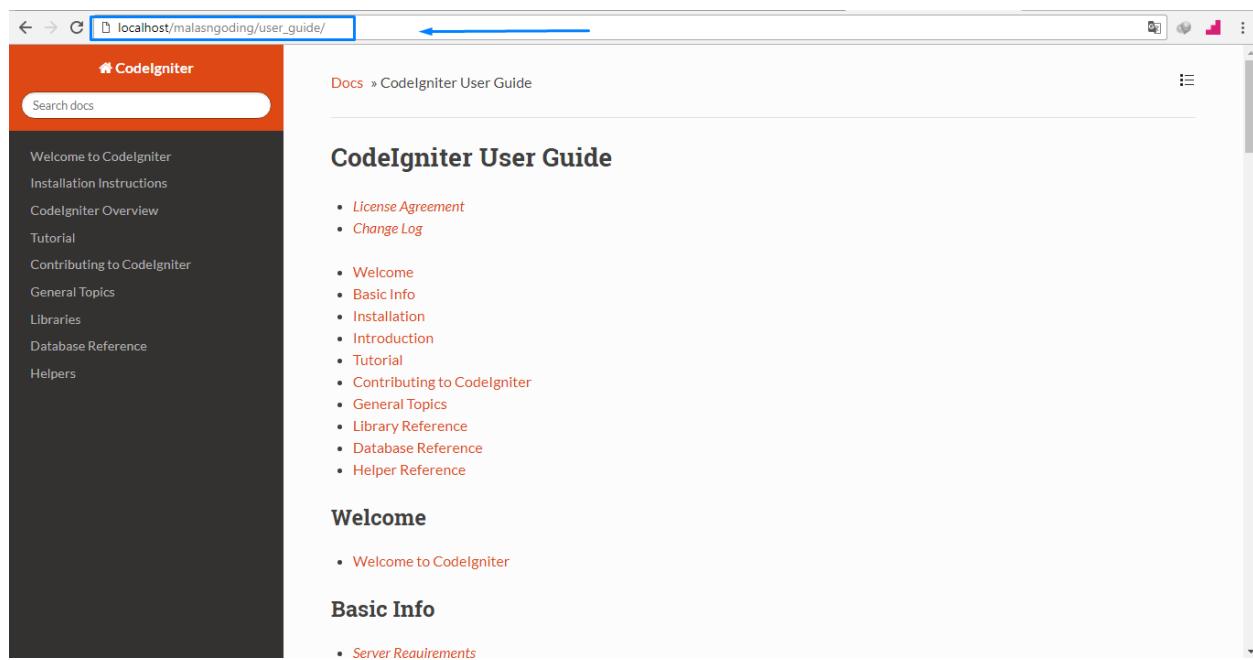


maka akan muncul tampilan seperti di atas yang menandakan bahwa codeigniter telah berhasil di install dan siap untuk di gunakan. tampilan di atas merupakan halaman default dasar dari codeigniter saat pertama kali di install.

Panduan Instalasi Codeigniter :

1. Download codeigniter di www.codeigniter.com
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>)

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/.



Segala cara penggunaan codeigniter bisa di baca 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 di katakan 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 di kirim.

jadi segment codeigniter di hitung dari setelah index.php pada codeigniter. katakan 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 di kirimkan 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 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.



Mobil itu berwarna merah

Segment 1

Segment 2

Segment 3

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 jalanakan 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 dilakukan 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";  
}
```

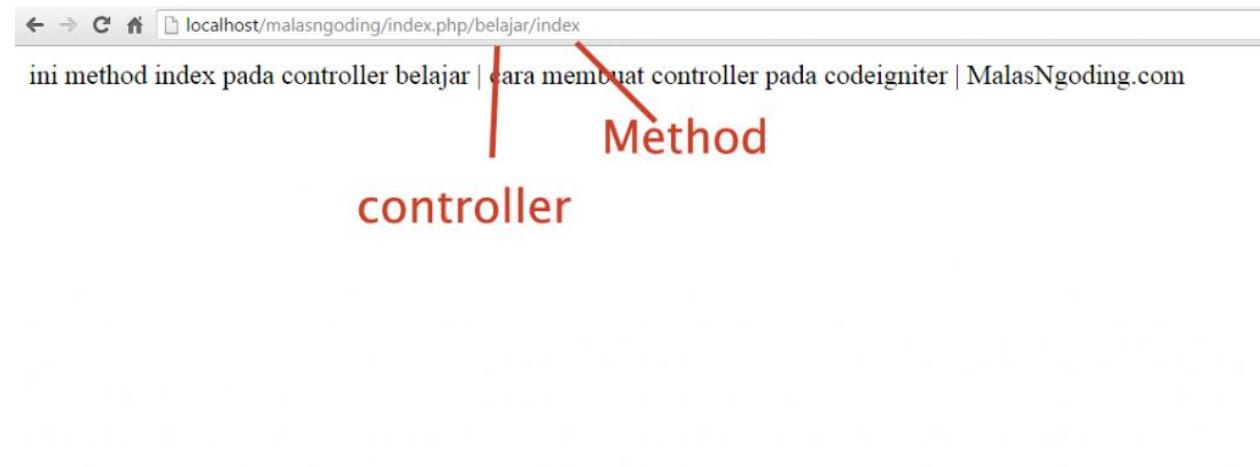
untuk menjalankan method index anda bisa mengaksesnya dengan alamat

```
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

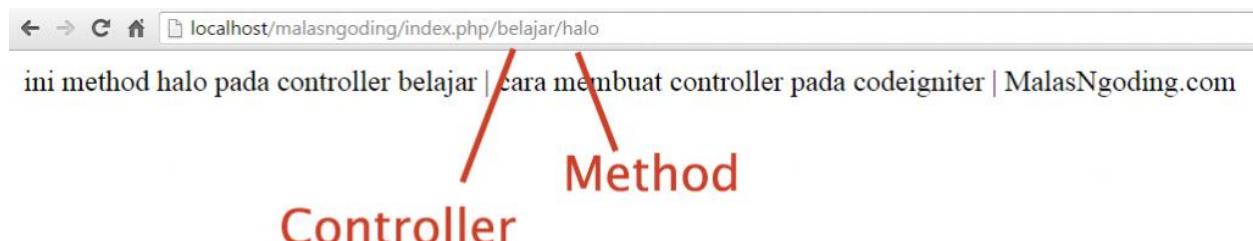
atau bisa juga

```
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/view/*.

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/view/*.

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

“ *application/view/view_belajar.php*”

langkah selanjutnya adalah membuat isi dari view.

application/view/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/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(){
        $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');
```

anda 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/view/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/view/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>
```

MalasNgoding.com

Perhatikan pada contoh di atas data di parsing pada dengan 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/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 = array(
        'judul' => "Cara Membuat View Pada CodeIgniter",
        'tutorial' => "CodeIgniter"
    );
    $this->load->view('view_belajar', $data);
}

}
```

application/view/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>
```

localhost/malasngoding/index.php/belajar/halo

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/controller/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() ?&gt;assets/css/style.css"&gt;
&lt;/head&gt;
&lt;body&gt;
    &lt;div id="wrapper"&gt;
        &lt;header&gt;
            &lt;hgroup&gt;
                &lt;h1&gt;MalasNgoding.com&lt;/h1&gt;
                &lt;h3&gt;Membuat template sederhana
codeigniter&lt;/h3&gt;
            &lt;/hgroup&gt;
            &lt;nav&gt;
                &lt;ul&gt;
                    &lt;li&gt;&lt;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>
                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.
            <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.
            </p>
        </section>
    </div>

```

```

        </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>
        <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 sesuaikan 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

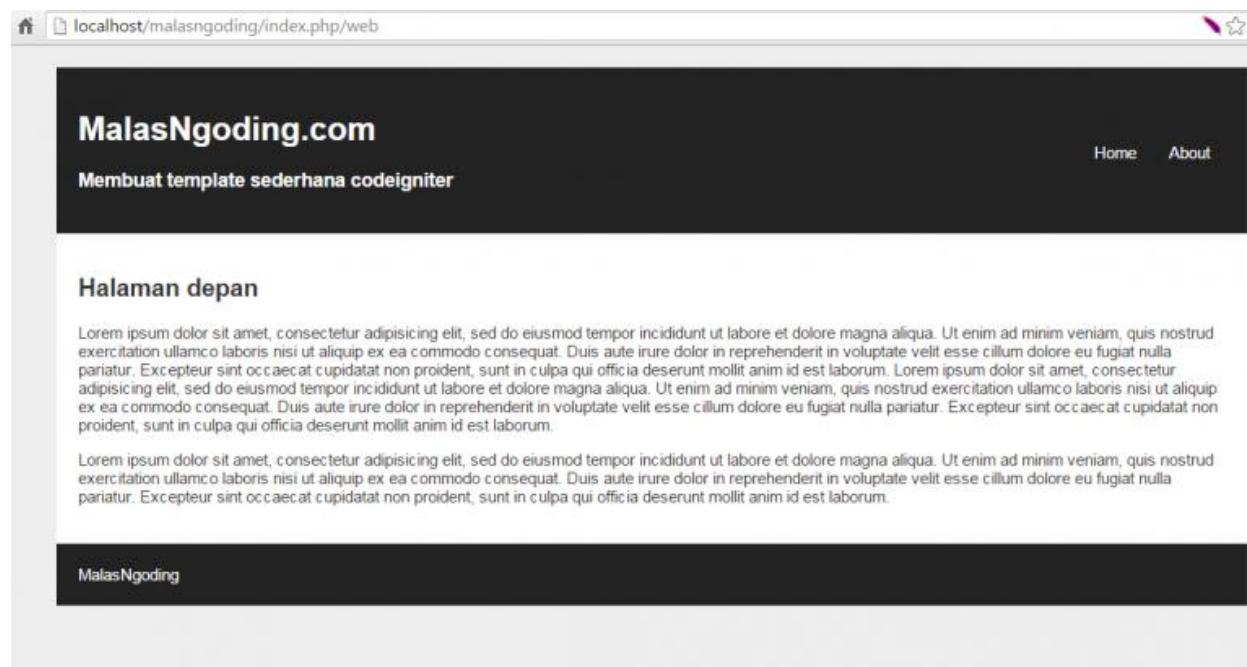
```
<link rel="stylesheet" type="text/css"  
href="http://localhost/malasngoding/assets/css/style.css">
```

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/view/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/view/v_footer.php

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

```
</html>
```

application/view/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/controller/web.php

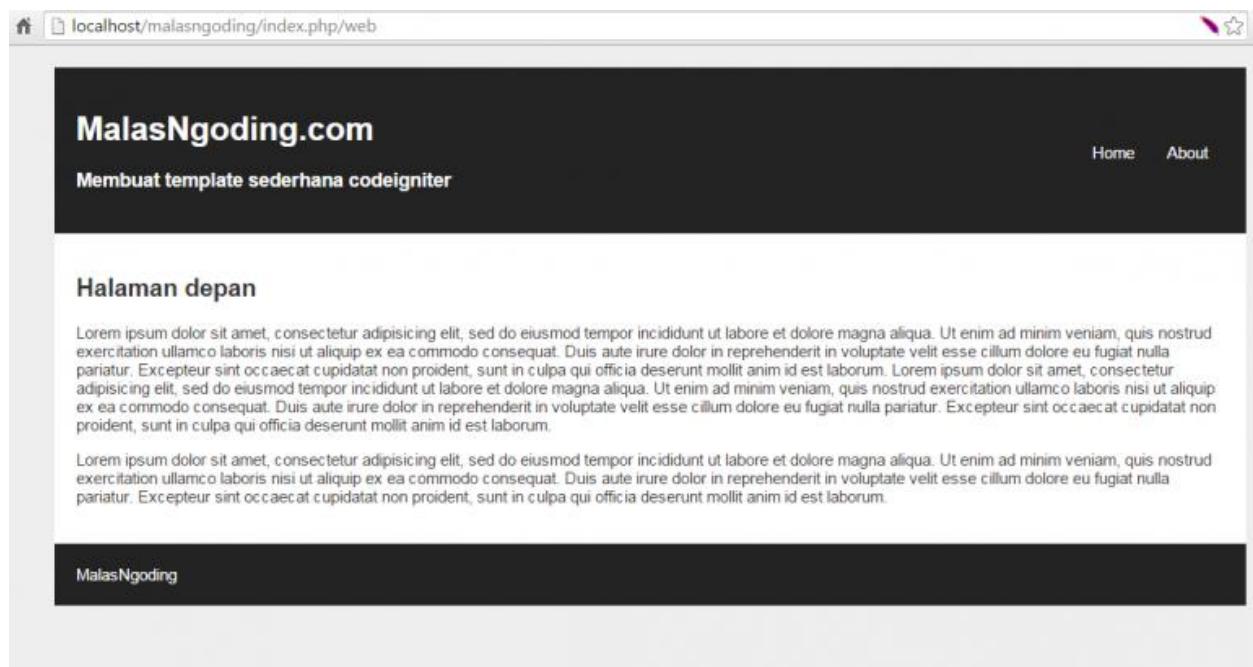
```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

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/view/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/controller/web.php

```

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

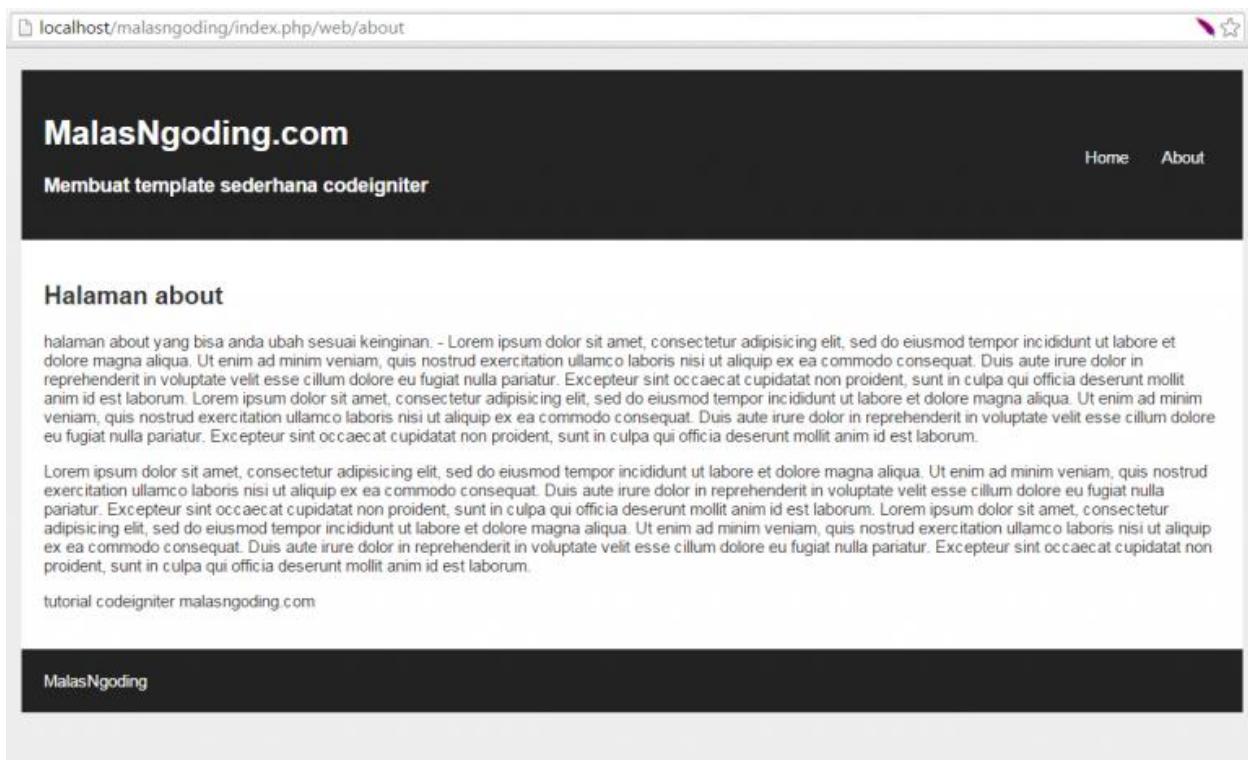
    function __construct(){
        parent::__construct();
        $this->load->helper('url');
    }

    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/controller/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/view/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

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.

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

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/controller/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 kita 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/view/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);
}
}
}

```

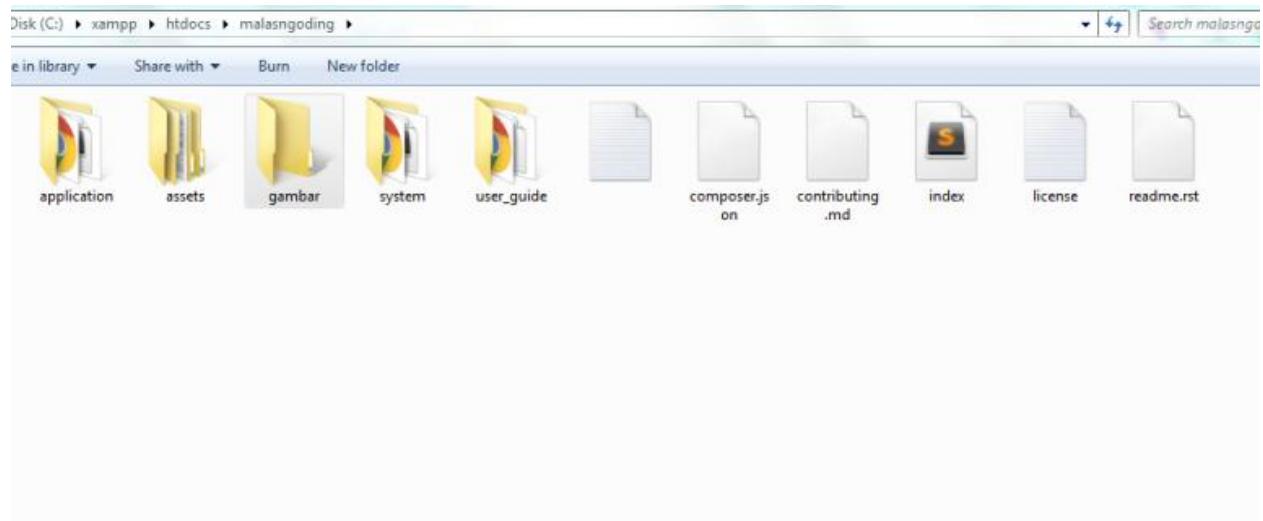
Yang paling penting ada di method aksi_upload ini. method ini akan kita jadikan sebagai pengatur dari proses upload file. terdapat variabel config dia atas, variabel ini berfungsi sebagai pengaturan upload file pada codeigniter. contohnya seperti berikut.

```

$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/view/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.

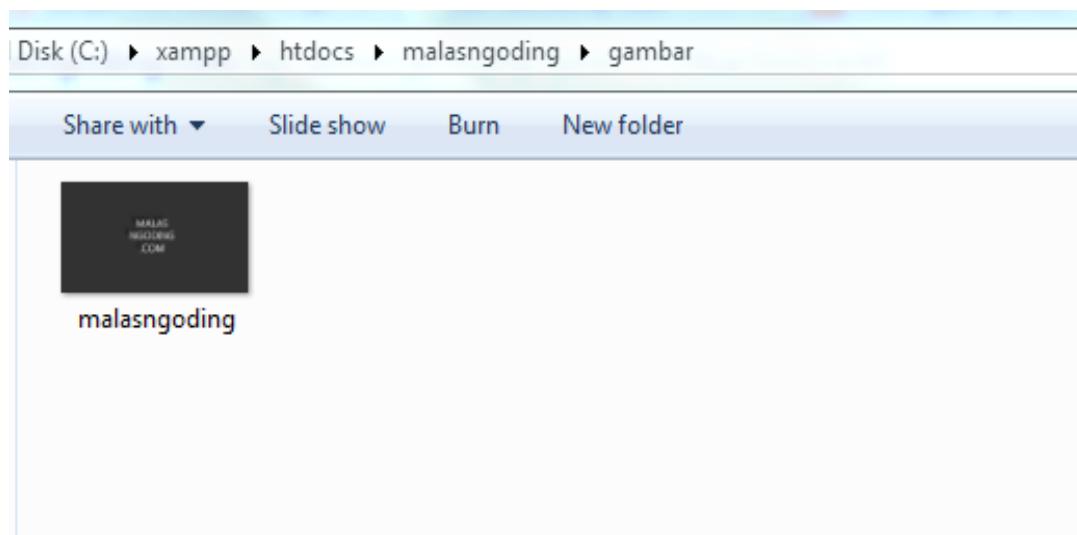


Pilih gambar yang ingin di upload dan submit (klik upload)

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/controller/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/view/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>

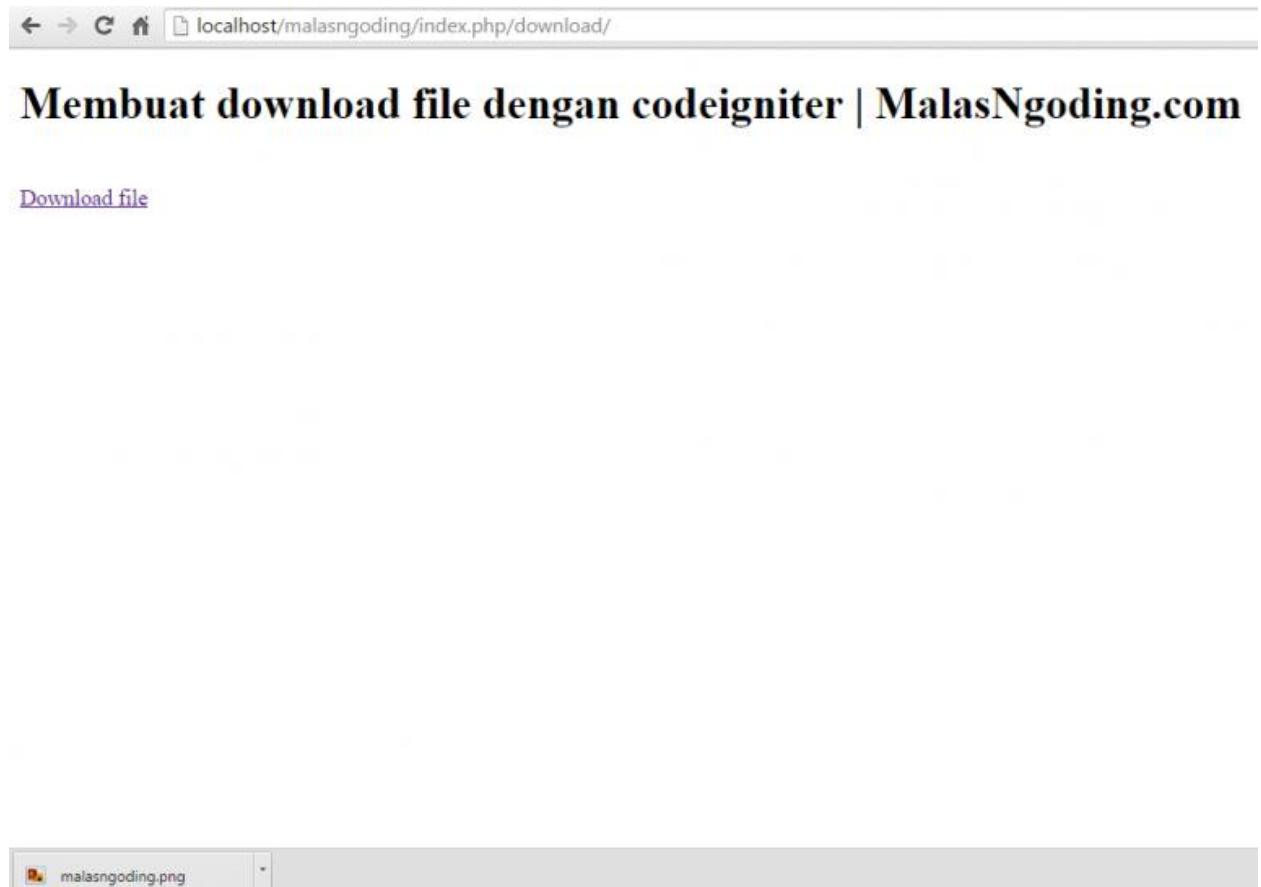
    <br/>

    <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.



terima kasih dan 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 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/controller/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 tingal 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 di bandingkan 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*.

Dan langkah selanjut nya 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 di sediakan 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

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 di jelaskan 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 di lakukan adalah membuat dulu database nya. kemudian menghubungkan codeigniter dengan database, kemudian lagi pada tutorial ini akan di jelaskan *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/controller/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/view/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

```
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/view/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 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 saja 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.



Membuat CRUD dengan CodeIgniter | MalasNgoding.com

Tambah Data

No	Nama	Alamat	Pekerjaan	Action
1	Andi	Surabaya	web programmer	Edit Hapus
2	Budi	Bandung	Web Designer	Edit Hapus

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/view/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	Edit Hapus
2	Budi	Bandung	Web Designer	Edit Hapus
3	Rudi	Semarang	Web Master	Edit Hapus

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 lanjutkan dengan 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/view/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	Edit Hapus
2	Budi	Bandung	Web Designer	Edit Hapus
3	Rudi	Semarang	Web Master	Edit Hapus

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

No	Nama	Alamat	Pekerjaan	Action
1	Andi	Surabaya	web programmer	Edit Hapus
2	Budi	Bandung	Web Designer	Edit Hapus

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/view/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 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
<input type="button" value="Simpan"/>	

ubah data sesuai keinginan dan klik simpan

localhost/malasngoding/crud/index

Membuat CRUD dengan CodeIgniter | MalasNgoding.com

[Tambah Data](#)

No	Nama	Alamat	Pekerjaan	Action
1	Andi	Surabaya	web programmer	Edit Hapus
2	Santoso	Jakarta	Web Designer	Edit Hapus

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>
```

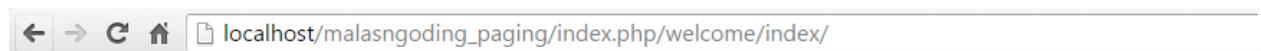
```

<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



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, 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 */;
```

+ Options						
	← T →	▼	id	username	password	
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	malasngoding	10406c1d7b7421b1a56f0d951e952a95
Check All / Uncheck All With selected:  Change  Delete  Export						
Show :	Start row:	0	Number of rows:	30	Headers every 100 rows	

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>
        </table>
    </form>
</body>

```

```

        </tr>
        <tr>
            <td></td>
            <td><input type="submit" value="Login"></td>
        </tr>
    </table>
</form>
</body>
</html>

```



Membuat Login Dengan CodeIgniter

www.malasngoding.com

Username	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Login"/>	

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'); ?>">Logout</a>  
</body>  
</html>
```

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.



Username	andi
Password
<input type="button" value="Login"/>	

maka akan ditampilkan 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.

Login berhasil !

Hai, malasngoding

[Logout](#)

Dan login pun berhasil kita buat.

Tutorial Codeigniter Lanjutan

Studi Kasus Membuat Aplikasi Rental Mobil

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 aplikasi rental mobil berbasis web.

Sebenarnya aplikasi ini sudah duluan saya buat. Baru kemudian saya tulis tutorialnya di buku ini. Karena jika proses penulisan ebook ini dengan proses pembuatan aplikasi rental mobil ini, akan sedikit ribet dan menyita banyak waktu. Oleh karena itu saya mempersiapkan terlebih dulu aplikasinya, 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.

Pada ebook ini tidak lagi dijelaskan secara terperinci. Karena semua pembahasan tentang codeigniter dasar telah di jelaskan pada pembahasan sebelumnya atau di www.malasngoding.com. Pada ebook ini akan di jelaskan bagaimana cara membangun aplikasi 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 v3.2.2
3. Web Browser Google Chrome

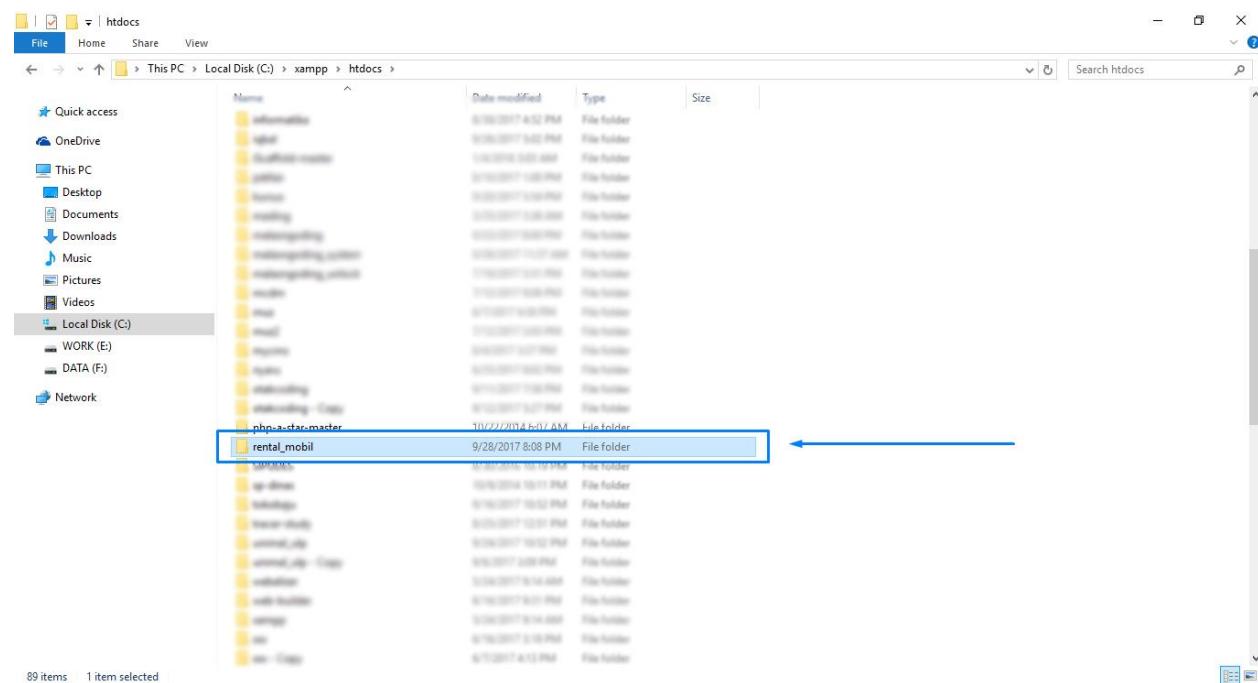
Semua aplikasi pendukung yang disebutkan di atas telah sertakan di dalam paket ebook ini (di dalam folder aplikasi pendukung).

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 dan saat ini digunakan dalam tutorial pembuatan ebook tutorial membuat aplikasi rental mobil dengan codeigniter dan bootstrap ini.

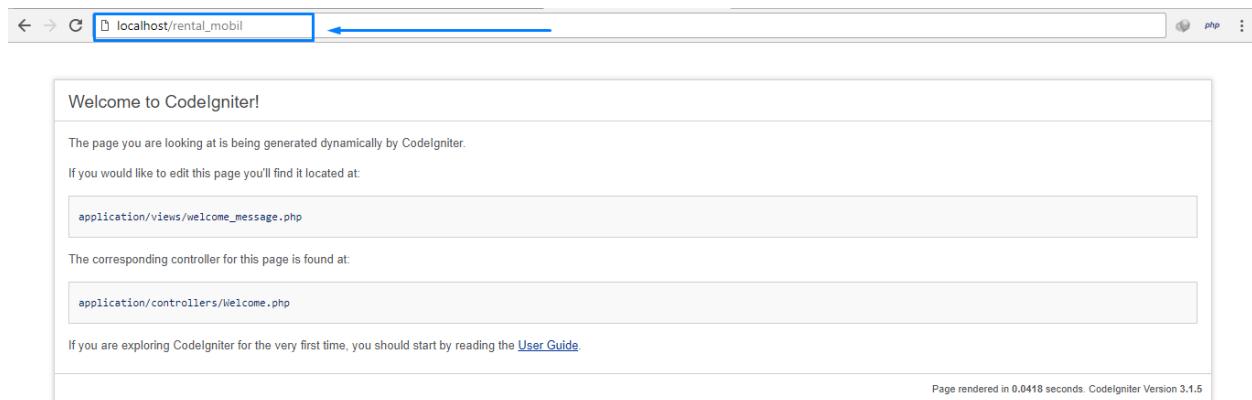
Silahkan teman-teman download codeigniter dari situs resminya (<https://codeigniter.com/>) . codeigniter 3 juga telah di sertakan di dalam ebook ini.

kemudian copy project codeigniter nya ke direktori localhost. (C:\XAMPP\htdocs) dan ubah nama folder nya menjadi “rental_mobil”. Seperti pada gambar berikut.



Kemudian buka web browser dan jalankan.

http://localhost/rental_mobil



Sampai di sini maka instalasi Codeigniter pun selesai.

Persiapan Membuat Aplikasi Rental Mobil

Diperlukan beberapa penyesuaian atau pengaturan pada codeigniter sebelum mulai kita gunakan untuk membuat aplikasi rental mobil. Semua pengaturan atau penyesuaian yang kita lakukan di sini sesuai dengan kebutuhan kita untuk membuat aplikasi rental mobil 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.

rental_mobil/application/config/config.php

The screenshot shows the Sublime Text editor with the file 'config.php' open. The file path is 'C:\xampp\htdocs\rental_mobil\application\config'. The code in the file is as follows:

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

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

/*
Index File
Typically this will be your index.php file, unless you've renamed it to
something else. If you are using mod rewrite to remove the page, set this
to ''.
*/
```

Sesuaikan dengan nama directory project kita. Dimana tadi folder project aplikasi kita kita beri nama "rental_mobil". Maka sesuaikan pengaturan base_url nya menjadi.

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

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. 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 di bilang seperti kita memiliki perpusatakan 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*.

Rental_mobil/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('');
```

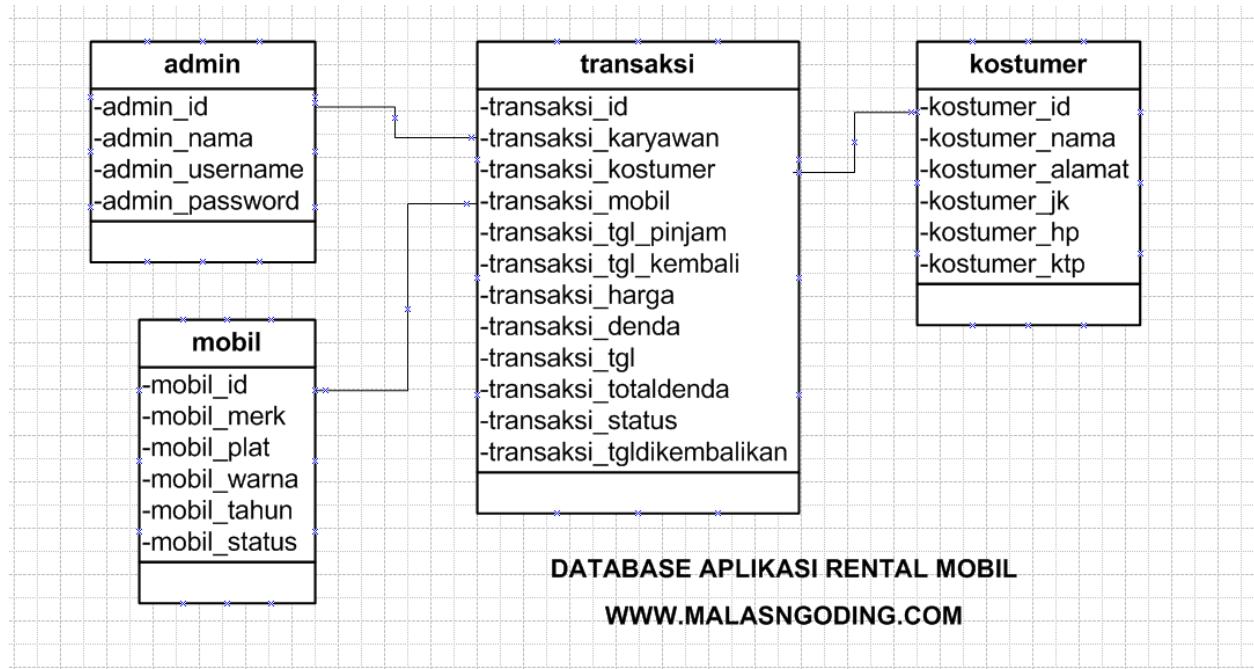
Kemudian tambahkan “*url*” (*karena kita hanya membutuhkan helper url disini*). Maka akan menjadi seperti berikut.

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

Proses autoload library dan helper selesai. Selanjutnya kita akan membuat database untuk aplikasi rental mobil dan melakukan konfigurasi database pada codeigniter.

Merancang & Membuat Database Aplikasi Rental Mobil dan Konfigurasi database di Codeigniter

Berikut adalah gambaran struktur database yang akan kita buat untuk aplikasi rental mobil ini.



Buat sebuah database baru dengan nama “rental_mobil”.

Ada 4 buah tabel yang akan kita buat dalam database. Yaitu :

- admin
- mobil
- transaksi
- kostumer

Tabel admin

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

Nama	Tipe	Auto_increment	Primary Key
admin_id	Int(11)	✓	✓
admin_nama	Varchar(255)		
admin_username	Varchar(255)		

admin_password	Varchar(255)		
----------------	--------------	--	--

Tabel mobil

Tabel mobil akan kita jadikan sebagai tempat menyimpan data mobil seperti merk mobil, no. plat, warna, tahun pembuatan dan status ketersediaan mobil.

Nama	Tipe	Auto_increment	Primary Key
mobil_id	Int(11)	✓	✓
mobil_merk	Varchar(30)		
mobil_plat	Varchar(20)		
mobil_warna	Varchar(30)		
mobil_tahun	Int(11)		
mobil_status	Int(11)		

Tabel kostumer

Tabel kostumer akan kita jadikan sebagai tempat menyimpan data kostumer atau data orang yang menyewa mobil. seperti nama, alamat, jenis kelamin, no. hp dan no.ktp.

Nama	Tipe	Auto_increment	Primary Key
kostumer_id	Int(11)	✓	✓
kostumer_nama	Varchar(255)		
kostumer_alamat	text		
kostumer_jk	Varchar(10)		
kostumer_hp	Varchar(20)		
kostumer_ktp	Varchar(50)		

Tabel transaksi

Tabel transaksi kita gunakan untuk menyimpan data transaksi, seperti karyawan/admin yg melakukan atau melayani kostumer saat melakukan transaksi, kostumer, mobil yang di sewa, tanggal mulai di rental, tanggal batas pengembalian, harga rental, harga denda per hari, tanggal transaksi, total denda jika terlambat di kembalikan dan tanggal saat mobil di kembalikan oleh kostumer.

Nama	Tipe	Auto_increment	Primary Key
transaksi_id	Int(11)	✓	✓
transaksi_karyawan	Int(11)		

transaksi_kostumer	Int(11)		
transaksi_mobil	Int(11)		
transaksi_tgl_pinjam	date		
transaksi_tgl_kembali	date		
transaksi_harga	Int(11)		
transaksi_denda	Int(11)		
transaksi_tgl	date		
transaksi_totaldenda	Int(11)		
transaksi_status	Int(11)		
transaksi_tgldikembalikan	date		

Atau untuk menghemat waktu teman-teman bisa copas langsung sql berikut ke database yang telah teman-teman buat (database : rental_mobil).

rental_mobil.sql

```
-- phpMyAdmin SQL Dump
-- version 4.5.1
-- http://www.phpmyadmin.net
--
-- Host: 127.0.0.1
-- Generation Time: 29 Sep 2017 pada 08.56
-- Versi Server: 10.1.16-MariaDB
-- PHP Version: 5.5.38

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 utf8mb4 */;

--
-- Database: `rental_mobil`
--

-----
-- Struktur dari tabel `admin`


CREATE TABLE `admin` (
  `admin_id` int(11) NOT NULL,
  `admin_nama` varchar(255) NOT NULL,
  `admin_username` varchar(255) NOT NULL,
```

```

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

-- 
-- Dumping data untuk tabel `admin`
-- 

INSERT INTO `admin` (`admin_id`, `admin_nama`, `admin_username`, `admin_password`) VALUES
(1, 'Diki Alfarabi Hadi', 'diki', 'dffaa4c60a250f19dc4a79b1d05c8d53');

-- 
-- Struktur dari tabel `kostumer`
-- 

CREATE TABLE `kostumer` (
`kostumer_id` int(11) NOT NULL,
`kostumer_nama` varchar(255) NOT NULL,
`kostumer_alamat` text NOT NULL,
`kostumer_jk` varchar(10) NOT NULL,
`kostumer_hp` varchar(20) NOT NULL,
`kostumer_ktp` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- 
-- Dumping data untuk tabel `kostumer`
-- 

INSERT INTO `kostumer` (`kostumer_id`, `kostumer_nama`, `kostumer_alamat`, `kostumer_jk`, `kostumer_hp`, `kostumer_ktp`)
VALUES
(2, 'Samsul Alami', 'Krukuh', 'P', '833637363', 'qwertyuiu'),
(3, 'Jamulidin S.E', 'Kuta Blang', 'L', '856567567', '234234234234'),
(4, 'Imam Faluti', 'Krung Mane', 'L', '2147483647', '34234234324');

-- 
-- Struktur dari tabel `mobil`
-- 

CREATE TABLE `mobil` (
`mobil_id` int(11) NOT NULL,
`mobil_merk` varchar(30) NOT NULL,
`mobil_plat` varchar(20) NOT NULL,
`mobil_warna` varchar(30) NOT NULL,
`mobil_tahun` int(11) NOT NULL,
`mobil_status` int(11) NOT NULL COMMENT '1 = tersedia, 2 = di pinjam'

```

```

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- 
-- Dumping data untuk tabel `mobil`
-- 

INSERT INTO `mobil` (`mobil_id`, `mobil_merk`, `mobil_plat`, `mobil_warna`, `mobil_tahun`, `mobil_status`) VALUES
(2, 'Toyota Avanza Veloz', 'B 9879 UK', 'Hitam', 2016, 1),
(3, 'Honda Jazz', 'B 2793 UI', 'Silver', 2002, 2),
(4, 'Kijang Inova', 'B 3242 JJ', 'Putih', 2012, 1);

-- 
-- Struktur dari tabel `transaksi`
-- 

CREATE TABLE `transaksi` (
`transaksi_id` int(11) NOT NULL,
`transaksi_karyawan` int(11) NOT NULL,
`transaksi_kostumer` int(11) NOT NULL,
`transaksi_mobil` int(11) NOT NULL,
`transaksi_tgl_pinjam` date NOT NULL,
`transaksi_tgl_kembali` date NOT NULL,
`transaksi_harga` int(11) NOT NULL,
`transaksi_denda` int(11) NOT NULL,
`transaksi_tgl` date NOT NULL,
`transaksi_totaldenda` int(11) NOT NULL,
`transaksi_status` int(11) NOT NULL,
`transaksi_tgldikembalikan` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- 
-- Dumping data untuk tabel `transaksi`
-- 

INSERT INTO `transaksi` (`transaksi_id`, `transaksi_karyawan`, `transaksi_kostumer`, `transaksi_mobil`, `transaksi_tgl_pinjam`, `transaksi_tgl_kembali`, `transaksi_harga`, `transaksi_denda`, `transaksi_tgl`, `transaksi_totaldenda`, `transaksi_status`, `transaksi_tgldikembalikan`) VALUES
(6, 1, 2, '2017-09-01', '2017-09-09', 200000, 100000, '2017-09-28', 2100000, 1, '2017-09-30'),
(7, 1, 3, '2017-09-02', '2017-09-04', 400000, 100000, '2017-09-15', 200000, 1, '2017-09-06'),
(8, 1, 2, 3, '2017-09-08', '2017-09-30', 200000, 100000, '2017-09-28', 0, 0, '0000-00-00');

-- 
-- Indexes for dumped tables
-- 

```

```

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

-- 
-- Indexes for table `kostumer` 
-- 
ALTER TABLE `kostumer` 
  ADD PRIMARY KEY (`kostumer_id`);

-- 
-- Indexes for table `mobil` 
-- 
ALTER TABLE `mobil` 
  ADD PRIMARY KEY (`mobil_id`);

-- 
-- Indexes for table `transaksi` 
-- 
ALTER TABLE `transaksi` 
  ADD PRIMARY KEY (`transaksi_id`);

-- 
-- AUTO_INCREMENT for dumped tables 
-- 

-- 
-- AUTO_INCREMENT for table `admin` 
-- 
ALTER TABLE `admin` 
  MODIFY `admin_id` int(11) NOT NULL AUTO_INCREMENT, 
AUTO_INCREMENT=2;
-- 
-- AUTO_INCREMENT for table `kostumer` 
-- 
ALTER TABLE `kostumer` 
  MODIFY `kostumer_id` int(11) NOT NULL AUTO_INCREMENT, 
AUTO_INCREMENT=5;
-- 
-- AUTO_INCREMENT for table `mobil` 
-- 
ALTER TABLE `mobil` 
  MODIFY `mobil_id` int(11) NOT NULL AUTO_INCREMENT, 
AUTO_INCREMENT=5;
-- 
-- AUTO_INCREMENT for table `transaksi` 
-- 
ALTER TABLE `transaksi` 
  MODIFY `transaksi_id` int(11) NOT NULL AUTO_INCREMENT, 
AUTO_INCREMENT=9;
/*!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 */;
```

Tabel	Jumlah	Baris	Jenis	Penyortiran	Ukuran	Beban
admin	10	1	InnoDB	latin1_swedish_ci	16 KB	-
kostumer	3	3	InnoDB	latin1_swedish_ci	16 KB	-
mobil	3	3	InnoDB	latin1_swedish_ci	16 KB	-
transaksi	3	3	InnoDB	latin1_swedish_ci	16 KB	-

Konfigurasi database pada codeigniter

Setelah selesai membuat database. Dan table-table seperti yang sudah di jelaskan di atas. Selanjutnya kita akan melakukan pengaturan atau konfigurasi database pada codeigniter. Untuk menghubungkan codeigniter dengan database mysql.

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

Lihat pada gambar berikut:

```

C:\xampp\htdocs\rental_mobil\application\config\database.php (rental_mobil) - Sublime Text (UNREGISTERED)
File Edit Selection Find Goto Tools Project Preferences Help
FOLDERS
rental_mobil
  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
  smilies.php
  user_agents.php
  controllers
  core
  helpers
  hooks
  language
  libraries
  logs
  models
  third_party
  views
Line 1, Column 1
Tab Size 5 PHP
database.php x
67 | The $active_group variable lets you choose which connection group to
68 | make active. By default there is only one group (the 'default' group).
69 |
70 | The $query_builder variables lets you determine whether or not to load
71 | the query builder class.
72 */
73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77   'dsn' => '',
78   'hostname' => 'localhost',
79   'username' => 'root',
80   'password' => '',
81   'database' => 'rental_mobil',
82   'dbdriver' => 'mysqli',
83   'dbprefix' => '',
84   'pconnect' => FALSE,
85   'db_debug' => (ENVIRONMENT !== 'production'),
86   'cache_on' => FALSE,
87   'cachedir' => '',
88   'char_set' => 'utf8',
89   'dbcollat' => 'utf8_general_ci',
90   'swap_pre' => '',
91   'encrypt' => FALSE,
92   'compress' => FALSE,
93   'stricton' => FALSE,
94   'failover' => array(),
95   'save_queries' => TRUE
96 );
97

```

Sesuaikan dengan username dan password mysql teman-teman. Jangan lupa untuk menambahkan nama database “rental_mobil” seperti pada gambar di atas.

```

$db['default'] = array(
  'dsn' => '',
  'hostname' => 'localhost',
  'username' => 'root',
  'password' => '',
  'database' => 'rental_mobil',
  '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
);

```

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_rental.php.

Ingat! Untuk penamaan file model di codeigniter, harus di awali dengan huruf besar.

Contoh : M_rental.php

Ingat!

Penamaan Controller dan Model harus di awali dengan **huruf besar** di codeigniter 3

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

rental_mobil/application/models/M_rental.php

```
<?php

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

class M_rental extends CI_Model{

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

    function get_data($table){
        return $this->db->get($table);
    }

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

```

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

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

```

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()*.

Kemudian kita harus mengatur autoload model di application/config/autoload.php. agar model *m_rental* di load secara otomatis. Jadi tidak perlu lagi kita load dari controller.

Application/config/autoload.php

Cari syntax barikut

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

Dan ubah menjadi

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

Nah, sampai disini kita telah selesai melakukan semua autoload library, helper dan model yang kita butuhkan dalam pembuatan aplikasi rental mobil.

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_rental');

```

Menghilangkan index.php dari URL Codeigniter

Kita perlu menghilangkan index.php dari URL codeigniter di project aplikasi rental mobil kita, agar struktur url aplikasi kita terlihat bersih. Dan caranya pun sangat mudah.

Pertama, buka file **config.php**

Application/config/config.php

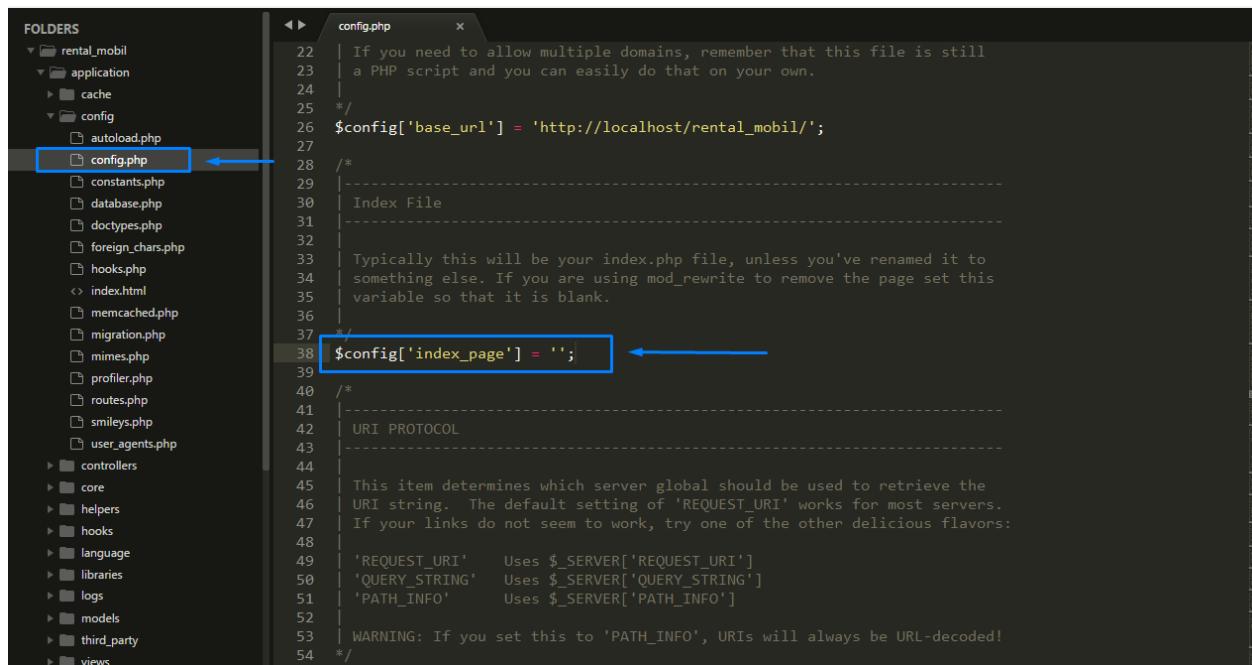
Cari syntax

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

Lalu hapus index.php, sehingga menjadi

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

Perhatikan pada gambar berikut

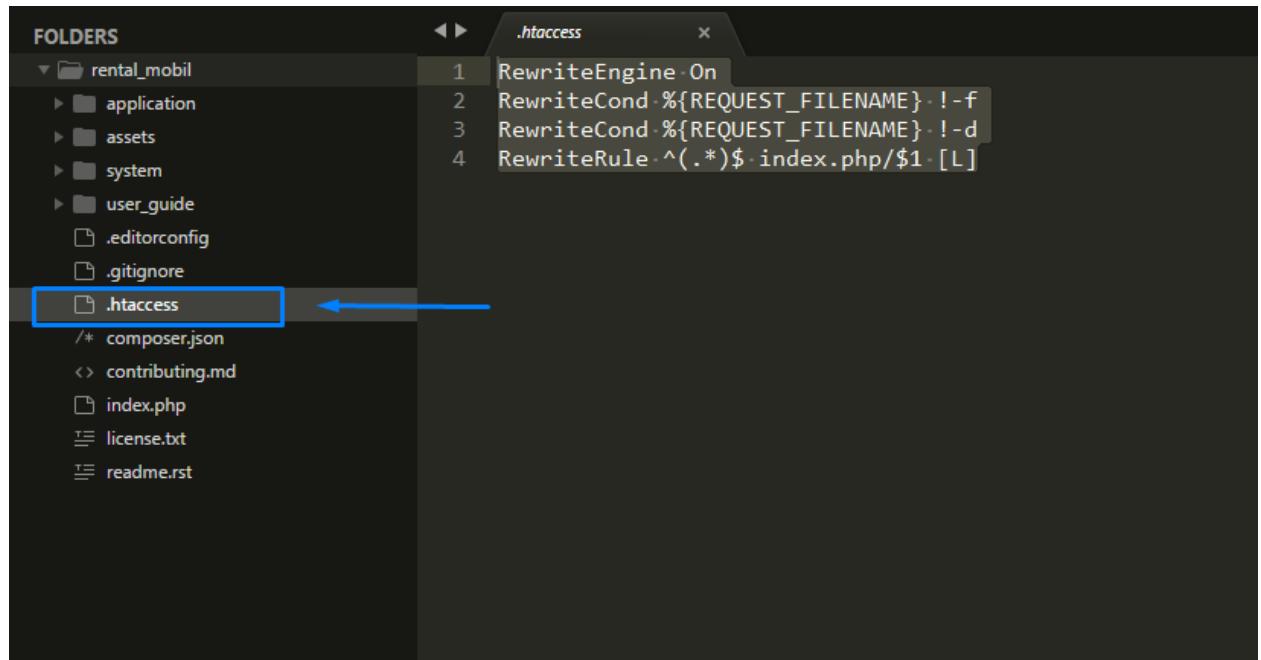


Kedua, buatlah sebuah file .htaccess pada directory root project rental_mobil.

.htaccess

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
```

```
RewriteRule ^(.*)$ index.php/$1 [L]
```



The screenshot shows a code editor interface with a sidebar labeled "FOLDERS" containing project files like "rental_mobil", "application", "assets", "system", "user_guide", ".editorconfig", ".gitignore", and ".htaccess". The main panel is titled ".htaccess" and contains the following code:

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

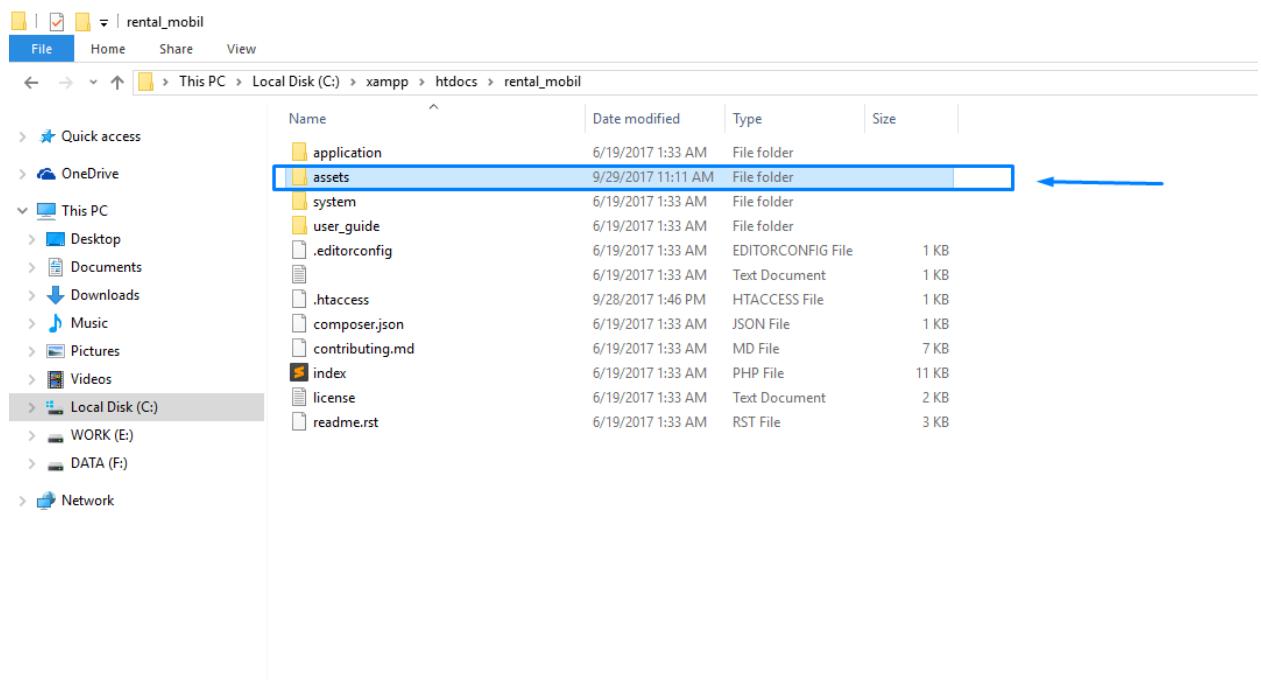
A blue double-headed arrow points from the ".htaccess" file in the sidebar to the ".htaccess" file in the main panel, indicating they are the same file.

Simpan dan selesai.

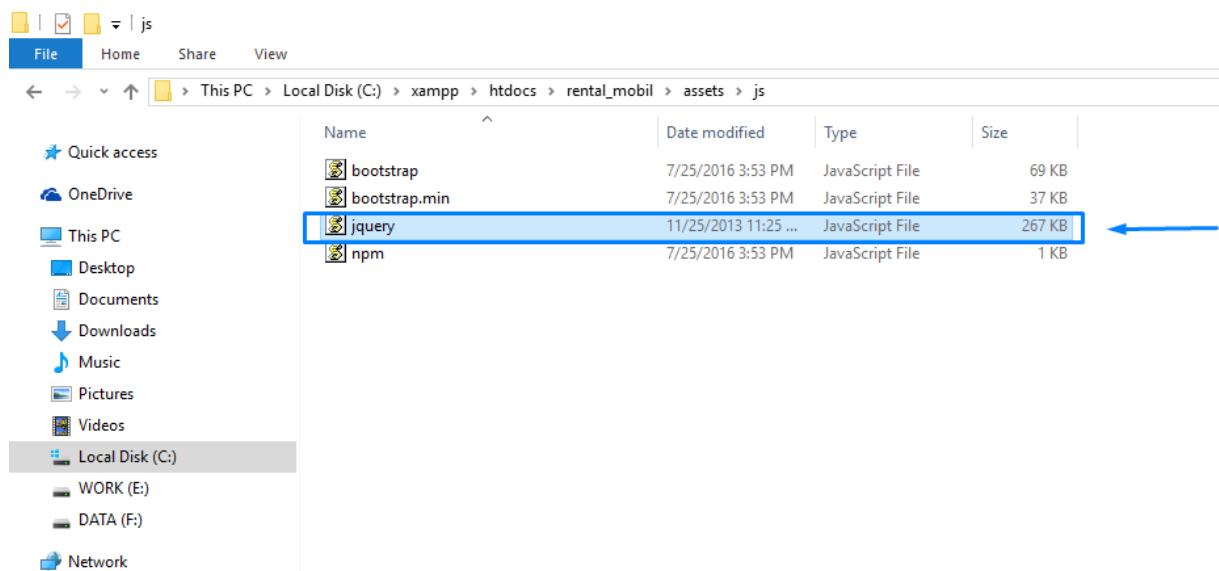
Integrasi Codeigniter dan Bootstrap

Teman-teman bisa mendownload bootstrap dari situs resminya di <http://getbootstrap.com/>, atau bisa menggunakan bootstrap yang telah di sertakan di dalam folder ebook ini. ekstrak bootstrap ke directory root project rental_mobil ini. Lalu rename nama folder menjadi “**assets**”.

Oh ya untuk tutorial bootstrap juga udah lengkap di www.malasngoding.com, silahkan di cek langsung. ☺



Download juga file **jquery.js** dan letakkan dalam folder “**assets/js**”. File jquery.js juga telah di sertakan di dalam folder ebook ini.



Membuat login dengan codeigniter, md5 & Pesan Notifikasi

Setelah segala persiapan selesai, kita akan mulai dengan membuat login untuk admin. Halaman login akan kita buat di halaman utama pada saat aplikasi rental mobil ini di akses.

Oh ya, untuk percobaan, kita harus menginput data username dan password admin terlebih dulu di tabel admin. Caranya teman-teman bisa klik *tambahkan* pada tabel admin. Kemudian isi admin_nama dengan “nama kamu”, admin_username dengan “admin”, dan admin_password dengan “admin123”. Jangan lupa pilih fungsi **md5** pada admin_password. Seperti contoh pada gambar berikut ini.

The screenshot shows the MySQL Workbench interface with the 'admin' table selected. The table has four columns: admin_id, admin_nama, admin_username, and admin_password. The data being inserted is:

Kolom	Jenis	Fungsi	Kosong	Nilai
admin_id	int(11)			1
admin_nama	varchar(255)			Diki Alfarabi Hadi
admin_username	varchar(255)			admin
admin_password	varchar(255)	MD5		admin123

Blue arrows point from the 'Nilai' column to the respective input fields in the table rows. A 'Kirim' button is at the bottom right.

Kemudian klik *kirim* untuk menyimpan data. Oke selesai, akun admin sudah kita buat, selanjutnya kita buat sistem loginnya.

Login akan kita buat pada controller Welcome.php. karena controller Welcome.php adalah controller yang default di gunakan pada codeigniter. Untuk mengubah default controller teman-teman bisa mengubahnya di **application/config/routes.php**. tapi di sini kita tidak akan mengubahnya, kita akan langsung menggunakan controller **Welcome.php** sebagai controller default. Halaman login pun akan kita buat di controller ini.

Buka Controller welcome.php dan tuliskan syntax berikut.

Application/controller/Welcome.php

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

class Welcome extends CI_Controller {

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

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

    function login(){
        $username = $this->input->post('username');
        $password = $this->input->post('password');
        $this->form_validation-
>set_rules('username','Username','trim|required');
```

```

        $this->form_validation-
>set_rules('password','Password','trim|required');
        if($this->form_validation->run() != false) {
            $where = array(
                'admin_username' => $username,
                'admin_password' => md5($password)

            );
            $data = $this->m_rental->edit_data($where,'admin');
            $d   =   $this->m_rental->edit_data($where,'admin')-
>row();
            $cek = $data->num_rows();
            if($cek > 0){
                $session = array(
                    'id'=> $d->admin_id,
                    'nama'=> $d->admin_nama,
                    'status' => 'login'
                );
                $this->session->set_userdata($session);
                redirect(base_url().'admin');
            }else{
                redirect(base_url().'welcome?pesan=gagal');

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

```

Setelah teman-teman selesai menulis syntax di atas pada controller Welcome.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.

Coba teman-teman perhatikan pada function construct() di atas, pada function construct() kita memanggil model ' m_rental '. kenapa kita perlu memanggil model ini ? karena kita akan berhubungan dengan model m_rental di dalam controller Welcome ini.

Selanjutnya perhatikan juga pada function index() di atas.

```

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

```

Pada function index() kita membuat perintah untuk membuka view login. Syntax \$this->load->view() adalah perintah untuk menampilkan view (tampilan). Di sini kita memerintahkan untuk menampilkan view login (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 welcome di jalankan, otomatis function index juga di jalankan, dan akan menampilkan halaman login (view login.php).

Karena pada function index kita ingin menampilkan view login.php, maka sekarang kita buat dulu view **login.php** nya.

Application/view/login.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Login - Aplikasi Rental Mobil Berbasis WEB - www.malasngoding.com</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>
    <div class="col-md-4 col-md-offset-4" style="margin-top: 50px">
        <center>
            <h2>APLIKASI RENTAL MOBIL</h2>
            <h3>LOGIN</h3>
        </center>
        <br/>
        <?php
            if(isset($_GET['pesan'])) {
                if($_GET['pesan'] == "gagal") {
                    echo "<div class='alert alert-danger'>Login gagal! Username dan password salah.</div>";
                } else if($_GET['pesan'] == "logout") {
                    echo "<div class='alert alert-danger'>Anda
```

```

telah logout.</div>";
        }else if($_GET['pesan'] == "belumlogin"){
            echo "<div class='alert alert-success'>Silahkan login dulu.</div>";
        }
    }
    ?>
<br/>
<div class="panel panel-default">
    <div class="panel-body">
        <br/>
        <br/>
        <form method="post" action="php echo base_url().'welcome/login' ?&gt;"&gt;
            &lt;div class="form-group"&gt;
                &lt;input type="text" name="username" placeholder="username" class="form-control"&gt;
                &lt;?php echo form_error('username'); ?&gt;
            &lt;/div&gt;
            &lt;div class="form-group"&gt;
                &lt;input type="password" name="password" placeholder="password" class="form-control"&gt;
                &lt;?php echo form_error('password'); ?&gt;
            &lt;/div&gt;
            &lt;div class="form-group"&gt;
                &lt;input type="submit" value="Login" class="btn btn-primary"&gt;
            &lt;/div&gt;
        &lt;/form&gt;
        &lt;br/&gt;
        &lt;br/&gt;
    &lt;/div&gt;
&lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre

```

Perhatikan pada tag `<head> ... </head>` di atas, Disinilah terlihat kegunaan dari `base_url()` yang sudah di jelaskan di atas tadi.

```

<link rel="stylesheet" type="text/css" href="php echo base_url().'assets/css/bootstrap.css' ?&gt;"&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;
</pre

```

Pada pengaturan config['base_url'] sebelumnya kita telah menetapkannya dengan http://localhost/rental_mobil/. Jadi saat kita gunakan seperti contoh menghubungkan template dengan bootstrap seperti di atas, sama seperti

```
<link rel="stylesheet" type="text/css"  
href="http://localhost/rental_mobil/assets/css/bootstrap.css' ?>">
```

Karena *base_url()* sudah sama dengan *http://localhost/rental_mobil/*.

Kemudian lagi perhatikan pada form yang sudah kita buat pada view login.php di atas.

```
<form method="post" action="<?php echo base_url().'welcome/login'  
?>">  
    <div class="form-group">  
        <input type="text" name="username" placeholder="username"  
class="form-control">  
        <?php echo form_error('username'); ?>  
    </div>  
    <div class="form-group">  
        <input type="password" name="password"  
placeholder="password" class="form-control">  
        <?php echo form_error('password'); ?>  
    </div>  
    <div class="form-group">  
        <input type="submit" value="Login" class="btn btn-  
primary">  
    </div>  
</form>
```

Action dari form ini telah kita atur ke function/method **login** di controller **welcome**.

```
<?php echo base_url().'welcome/login' ?>
```

Oke kembali lagi perhatikan method/function **login** pada controller welcome yang telah kita buat. Karena function ini yang akan di eksekusi sebagai *form handler* dari form login kita.

```
function login(){  
    $username = $this->input->post('username');  
    $password = $this->input->post('password');  
    $this->form_validation->set_rules('username','Username','trim|required');  
    $this->form_validation-
```

```

>set_rules('password','Password','trim|required');
    if($this->form_validation->run() != false){
        $where = array(
            'admin_username' => $username,
            'admin_password' => md5($password)
        );
        $data = $this->m_rental->edit_data($where,'admin');
        $d = $this->m_rental->edit_data($where,'admin')->row();
        $cek = $data->num_rows();
        if($cek > 0){
            $session = array(
                'id'=> $d->admin_id,
                'nama'=> $d->admin_nama,
                'status' => 'login'
            );
            $this->session->set_userdata($session);
            redirect(base_url().'admin');
        }else{
            redirect(base_url().'welcome?pesan=gagal');
        }
    }
}

```

Pertama kita menangkap data yang di kirim 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');
$this->form_validation-
>set_rules('username','Username','trim|required');
$this->form_validation-
>set_rules('password','Password','trim|required');

```

Kemudian kita cek validasinya.

```

if($this->form_validation->run() != false){
    $where = array(
        'admin_username' => $username,
        'admin_password' => md5($password)
    );
    $data = $this->m_rental->edit_data($where,'admin');
    $d = $this->m_rental->edit_data($where,'admin')->row();
    $cek = $data->num_rows();
    if($cek > 0){
        $session = array(
            'id'=> $d->admin_id,

```

```

        'nama'=> $d->admin_nama,
        'status' => 'login'
    );
    $this->session->set_userdata($session);
    redirect(base_url().'admin');
} else{
    redirect(base_url().'welcome?pesan=gagal');
}
else{
    $this->load->view('login');
}

```

Jika tidak ada masalah, kita cek username dan password yang di input sesuai atau tidak dengan username dan password pada tabel admin. Jika sesuai maka kita buat session id, nama dan status. Yang mana session id kita simpan id admin yang melakukan login, session nama kita mengimpaan nama admin yang login, dan status nya kita set dengan 'login'.

Setelah kita buat sessionnya, maka kita alihkan halamannya ke controller admin. Karena controller admin ini yang akan kita jadikan sebagai halaman admin.

```
redirect(base_url().'admin');
```

Jika login gagal maka kita alihkan ke controller welcome kembali. Yang mana halaman login ada di controller welcome. Sambil mengirimkan pesan melalui url untuk membuat pesan notifikasi bahwa login gagal.

```
redirect(base_url().'welcome?pesan=gagal');
```

Perhatikan pada view **login.php** yang sudah kita buat. Di sana kita telah membuat pesan notifikasi. Jika terdapat data pesan yang di kirim dari url, dan jika pesannya adalah "gagal", maka akan di tampilkan pesan notifikasi " login gagal! Username dan password salah ".

```

<?php
if(isset($_GET['pesan'])){
    if($_GET['pesan'] == "gagal"){
        echo "<div class='alert alert-danger'>Login gagal!
Username dan password salah.</div>";
    } else if($_GET['pesan'] == "logout"){
        echo "<div class='alert alert-danger'>Anda telah
logout.</div>";
    } else if($_GET['pesan'] == "belumlogin") {

```

```
        echo "<div class='alert alert-success'>Silahkan login  
dulu.</div>";  
    }  
}  
?>
```

Jika pesannya adalah logout, maka akan di tampilkan “ anda telah logout ”. Jika pesannya adalah “belumlogin”. Maka akan di tampilkan notifikasi “silahkan login dulu”.

Sampai di sini, sistem login kita belum selesai, tapi belum rampung 100%, karena kita belum menyiapkan halaman admin. Oleh sebab itu kita akan membuat dan menyiapkan halaman admin dulu. Baru setelah itu kita coba apakah login sudah berjalan sebagai mana yang kita perintahkan atau belum.

Menyiapkan Halaman Dashboard Admin

Jika login berhasil, halaman akan di alihkan ke controller admin. Maka mari kita siapkan dulu controller admin nya.

Buat sebuah controller baru lagi dengan nama Admin.php

Application/controller/Admin.php

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

class Admin extends CI_Controller {

    function __construct(){
        parent::__construct();
        // cek login
        if($this->session->userdata('status') != "login"){
            redirect(base_url().'welcome?pesan=belumlogin');
        }
    }

    function index(){
        $data['transaksi'] = $this->db->query("select * from transaksi order by transaksi_id desc limit 10")->result();
        $data['kostumer'] = $this->db->query("select * from kostumer order by kostumer_id desc limit 10")->result();
        $data['mobil'] = $this->db->query("select * from mobil order by mobil_id desc limit 10")->result();
        $this->load->view('admin/header');
        $this->load->view('admin/index', $data);
        $this->load->view('admin/footer');
    }

}
```

Pada function construct() kita membuat pengecekan jika si admin belum login, maka akan di alihkan ke halaman login. 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 akan membuat halaman dashboard.

```

function index(){
    $data['transaksi'] = $this->db->query("select * from transaksi
order by transaksi_id desc limit 10")->result();
    $data['kostumer'] = $this->db->query("select * from kostumer
order by kostumer_id desc limit 10")->result();
    $data['mobil'] = $this->db->query("select * from mobil order
by mobil_id desc limit 10")->result();
    $this->load->view('admin/header');
    $this->load->view('admin/index',$data);
    $this->load->view('admin/footer');
}

```

Pada function index() kita menampilkan view header, index dan footer. Yang mana pada view index kita memarsing data yang kita ambil dari database.

```

$data['transaksi'] = $this->db->query("select * from transaksi order
by transaksi_id desc limit 10")->result();
$data['kostumer'] = $this->db->query("select * from kostumer order
by kostumer_id desc limit 10")->result();
$data['mobil'] = $this->db->query("select * from mobil order by
mobil_id desc limit 10")->result();

```

\$this->db->query() adalah fungsi untuk menjalankan query database di codeigniter. Teman-teman bisa membaca panduannya pada user_guide codeigniter.

http://localhost/rental_mobil/user_guide/database/queries.html

Di sarankan untuk membaca tutorial membuat template web sederhana dengan codeigniter dan tutorial cara memarsing data ke view codeigniter terlebih dulu sebelum melanjutkan ke depan. Tutorial membuat tampilan atau template web sederhana dengan codeigniter dan cara memarsing data ke view sudah di jelaskan pada tutorial codeigniter dasar sebelumnya.

Pada variabel array transaksi di atas bertujuan untuk mengambil data transaksi dari tabel transaksi sebanyak 10 record. Untuk di tampilkan di halaman admin. Variabel array kostumer kita buat untuk mengambil 10 record data kostumer terakhir, dan variabel array mobil kita buat untuk mengambil 10 record terakhir data mobil dari tabel mobil. Semua data yang kita ambil ini akan kita tampilkan pada view index (halaman dashboard admin).

Buat sebuah folder dengan nama '**admin**' di dalam folder view.kemudian buat 3 buah view baru dengan nama index.php, header.php dan footer.php di dalam folder admin. pada view index.php ini kita akan membuat halaman dashboard admin.

Application/view/admin/header.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Dashboard - Aplikasi Rental Mobil</title>
    <link rel="stylesheet" type="text/css" href="<?php echo base_url().'assets/css/bootstrap.css' ?>">
        <link rel="stylesheet" type="text/css" href="<?php echo base_url().'assets/dataTables/datatables.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>
            <script type="text/javascript" src="<?php echo base_url().'assets/dataTables/jquery.dataTables.js'; ?>"></script>
            <script type="text/javascript" src="<?php echo base_url().'assets/dataTables/datatables.js'; ?>"></script>
</head>
<body>
    <nav class="navbar navbar-default">
        <div class="container">
            <!-- Brand and toggle get grouped for better mobile display -->
            <div class="navbar-header">
                <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="<?php echo base_url().'admin'; ?>">Rental Mobil</a>
            </div>

            <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
                <ul class="nav navbar-nav">
                    <li class="active"><a href="<?php echo base_url().'admin'; ?>"><span class="glyphicon glyphicon-home"></span> Dashboard <span class="sr-only">(current)</span></a></li>
                    <li><a href="<?php echo base_url().'admin/mobil'; ?>"><span class="glyphicon glyphicon-folder-open"></span> Data Mobil</a></li>
                    <li><a href="<?php echo base_url().'admin/kostumer'; ?>"><span class="glyphicon glyphicon-user"></span> Data Kostumer</a></li>
                    <li><a href="<?php echo base_url().'admin/transaksi'; ?>"><span class="glyphicon glyphicon-
```

```

sort"></span> Transaksi Rental</a></li>
            <li><a href="<?php echo
base_url().'admin/laporan'; ?>"><span class="glyphicon glyphicon-
list-alt"></span> Laporan</a></li>

        </ul>

        <ul class="nav navbar-nav navbar-right">
            <li><a href="<?php echo
base_url().'admin/logout'; ?>"><span class="glyphicon glyphicon-log-
out"></span> Logout</a></li>
            <li class="dropdown">
                <a href="#" class="dropdown-toggle" data-
toggle="dropdown" role="button" aria-haspopup="true" aria-
expanded="false"><?php echo "Halo, <b>".\$this->session-
userdata('nama'); ?></b> <span class="caret"></span></a>
                <ul class="dropdown-menu">
                    <li>
                        <a href="<?php echo
base_url().'admin/ganti_password' ?>"><i class="glyphicon glyphicon-
lock"></i> Ganti Password</a>
                    </li>
                </ul>
            </li>
        </ul>
    </div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>

<div class="container">

```

Application/view/admin/index.php

```

<div class="page-header">
    <h3>Dashboard</h3>
</div>

<div class="row">
    <div class="col-lg-3 col-md-6">
        <div class="panel panel-primary">
            <div class="panel-heading">
                <div class="row">
                    <div class="col-xs-3">
                        <i class="glyphicon glyphicon-
folder-open"></i>
                    </div>
                    <div class="col-xs-9 text-right">
                        <div class="huge">
                            <?php echo \$this->m_rental-
>get_data('mobil')->num_rows(); ?>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

                <div>Jumlah Mobil</div>
            </div>
        </div>
        <a href="<?php echo base_url().'admin/mobil' ?>">
            <div class="panel-footer">
                <span class="pull-left">View
Details</span>
                <span class="pull-right"><i
class="glyphicon glyphicon-arrow-right"></i></span>
                <div class="clearfix"></div>
            </div>
        </a>
    </div>
</div>
<div class="col-lg-3 col-md-6">
    <div class="panel panel-success">
        <div class="panel-heading">
            <div class="row">
                <div class="col-xs-3">
                    <i class="glyphicon glyphicon-
user"></i>
                </div>
                <div class="col-xs-9 text-right">
                    <div class="huge">
                        <?php echo $this->m_rental-
>get_data('kostumer')->num_rows(); ?>
                    </div>
                    <div>Jumlah Kostumer</div>
                </div>
            </div>
        </div>
        <a href="<?php echo base_url().'admin/kostumer' ?>">
            <div class="panel-footer">
                <span class="pull-left">View
Details</span>
                <span class="pull-right"><i
class="glyphicon glyphicon-arrow-right"></i></span>
                <div class="clearfix"></div>
            </div>
        </a>
    </div>
</div>
<div class="col-lg-3 col-md-6">
    <div class="panel panel-warning">
        <div class="panel-heading">
            <div class="row">
                <div class="col-xs-3">
                    <i class="glyphicon glyphicon-
sort"></i>
                </div>
                <div class="col-xs-9 text-right">
                    <div class="huge">

```

```

        <?php echo $this->m_rental-
>get_data('transaksi')->num_rows(); ?>
            </div>
            <div>Jumlah Transaksi</div>
        </div>
    </div>
    <a href="<?php echo base_url().'admin/transaksi'>
?>>
        <div class="panel-footer">
            <span class="pull-left">View
Details</span>
            <span class="pull-right"><i
class="glyphicon glyphicon-arrow-right"></i></span>
            <div class="clearfix"></div>
        </div>
        </a>
    </div>
<div class="col-lg-3 col-md-6">
    <div class="panel panel-danger">
        <div class="panel-heading">
            <div class="row">
                <div class="col-xs-3">
                    <i class="glyphicon glyphicon-
ok"></i>
                </div>
                <div class="col-xs-9 text-right">
                    <div class="huge">
                        <?php echo $this->m_rental-
>edit_data(array('transaksi_status'=>1), 'transaksi')->num_rows(); ?>
                    </div>
                    <div>Rental Selesai</div>
                </div>
            </div>
        </div>
        <a href="<?php echo base_url().'admin/transaksi'>
?>>
            <div class="panel-footer">
                <span class="pull-left">View
Details</span>
                <span class="pull-right"><i
class="glyphicon glyphicon-arrow-right"></i></span>
                <div class="clearfix"></div>
            </div>
            </a>
        </div>
    </div>
</div>
<hr>
<div class="row">
    <div class="col-lg-4">
        <div class="panel panel-default">

```

```

<div class="panel-heading">
    <h3 class="panel-title"><i class="glyphicon glyphicon-random arrow-right">/</i> Mobil</h3>
</div>
<div class="panel-body">
    <div class="list-group">
        <?php foreach($mobil as $m){ ?>
        <a href="#" class="list-group-item">
            <span class="badge"><?php if($m->mobil_status == 1){echo "Tersedia";}else{echo "D dirental";}>
        ?></span>
            <i class="glyphicon glyphicon-user">/</i> <?php echo $m->mobil_merk; ?>
        </a>
        <?php } ?>
    </div>
    <div class="text-right">
        <a href="<?php echo base_url().'admin/mobil' ?>">Lihat Semua Mobil <i class="glyphicon glyphicon-arrow-right">/</i></a>
    </div>
</div>
</div>
<div class="col-lg-3">
    <div class="panel panel-default">
        <div class="panel-heading">
            <h3 class="panel-title"><i class="glyphicon glyphicon-user o">/</i> Kostumer Terbaru</h3>
        </div>
        <div class="panel-body">
            <div class="list-group">
                <?php foreach($kostumer as $k){ ?>
                <a href="#" class="list-group-item">
                    <span class="badge"><?php echo $k->kostumer_jk ?></span>
                    <i class="glyphicon glyphicon-user">/</i> <?php echo $k->kostumer_nama; ?>
                </a>
                <?php } ?>
            </div>
            <div class="text-right">
                <a href="<?php echo base_url().'admin/kostumer' ?>">Lihat Semua Kostumer <i class="glyphicon glyphicon-arrow-right">/</i></a>
            </div>
        </div>
    </div>
<div class="col-lg-5">
    <div class="panel panel-default">
        <div class="panel-heading">
            <h3 class="panel-title"><i class="glyphicon

```

```

glyphicon-sort"></i> Peminjaman Terakhir</h3>
        </div>
        <div class="panel-body">
            <div class="table-responsive">
                <table class="table table-bordered table-
hover table-striped">
                    <thead>
                        <tr>
                            <th>Tgl. Transaksi</th>
                            <th>Tgl. Pinjam</th>
                            <th>Tgl. Kembali</th>
                            <th>Total</th>
                        </tr>
                    </thead>
                    <tbody>
<?php
foreach($transaksi as $t) {
?
<tr>
    <td><?php echo
date('d/m/Y', strtotime($t->transaksi_tgl)); ?></td>
    <td><?php echo
date('d/m/Y', strtotime($t->transaksi_tgl_pinjam)); ?></td>
    <td><?php echo
date('d/m/Y', strtotime($t->transaksi_tgl_kembali)); ?></td>
    <td><?php echo "Rp.
".number_format($t->transaksi_harga)."-"; ?></td>


```

Application/view/footer.php

```

</div>
<script type="text/javascript">

```

```

$(document).ready(function(){
    $('#table-datatable').DataTable();
});
</script>
</body>
</html>

```

C:\xampp\htdocs\rental_mobil\application\views\admin\header.php (rental_mobil) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- rental_mobil
- application
 - cache
 - config
 - controllers
 - Admin.php
 - index.html
 - Welcome.php
 - core
 - helpers
 - hooks
 - language
 - libraries
 - logs
 - models
 - third_party
 - views
 - admin
 - footer.php
 - header.php
 - index.php
 - errors
 - index.html
 - login.php
 - welcome_message.php
 - htaccess
 - index.html
 - assets
 - system

header.php index.php footer.php

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Dashboard - Aplikasi Rental Mobil</title>
5     <link rel="stylesheet" type="text/css" href="=base_url().'assets/css/bootstrap.css' ?">
6     <link rel="stylesheet" type="text/css" href="=base_url().'/assets/datatables/datatables.css' ?">
7     <script type="text/javascript" src="=base_url().'/assets/js/jquery.js'; ?&gt;"&gt;&lt;/script&gt;
8     &lt;script type="text/javascript" src="<?=base_url().'/assets/js/bootstrap.js'; ?&gt;"&gt;&lt;/script&gt;
9     &lt;script type="text/javascript" src="<?=base_url().'/assets/datatables/jquery.dataTables.js'; ?&gt;"&gt;&lt;/script&gt;
10    &lt;script type="text/javascript" src="<?=base_url().'/assets/datatables/datatables.js'; ?&gt;"&gt;&lt;/script&gt;
11 &lt;/head&gt;
12 &lt;body&gt;
13     &lt;nav class="navbar navbar-default"&gt;
14         &lt;div class="container"&gt;
15             &lt;!-- Brand and toggle get grouped for better mobile display --&gt;
16             &lt;div class="navbar-header"&gt;
17                 &lt;button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1"&gt;
18                     &lt;span class="sr-only"&gt;Toggle navigation&lt;/span&gt;
19                     &lt;span class="icon-bar"&gt;&lt;/span&gt;
20                     &lt;span class="icon-bar"&gt;Definition:&lt;/span&gt;
21                     &lt;span class="icon-bar"&gt;<a href="#"user_guide/_static/css/theme.css:1</span>
22                 </button>
23                 <a class="navbar-brand" href="#">Rental Mobil</a>
24             </div>
25         <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
26             <ul class="nav navbar-nav">
27                 <li class="active"><a href="#"><span class="glyphicon glyphicon-home"></span> Home</a></li>
28                 <li><a href="#"><span class="glyphicon glyphicon-user"></span> Admin</a></li>
29                 <li><a href="#"><span class="glyphicon glyphicon-th-list"></span> Kostumen</a></li>
30                 <li><a href="#"><span class="glyphicon glyphicon-th-list"></span> Transaksi</a></li>
31                 <li><a href="#"><span class="glyphicon glyphicon-th-list"></span> Laporan</a></li>
32             </ul>
33         </div>

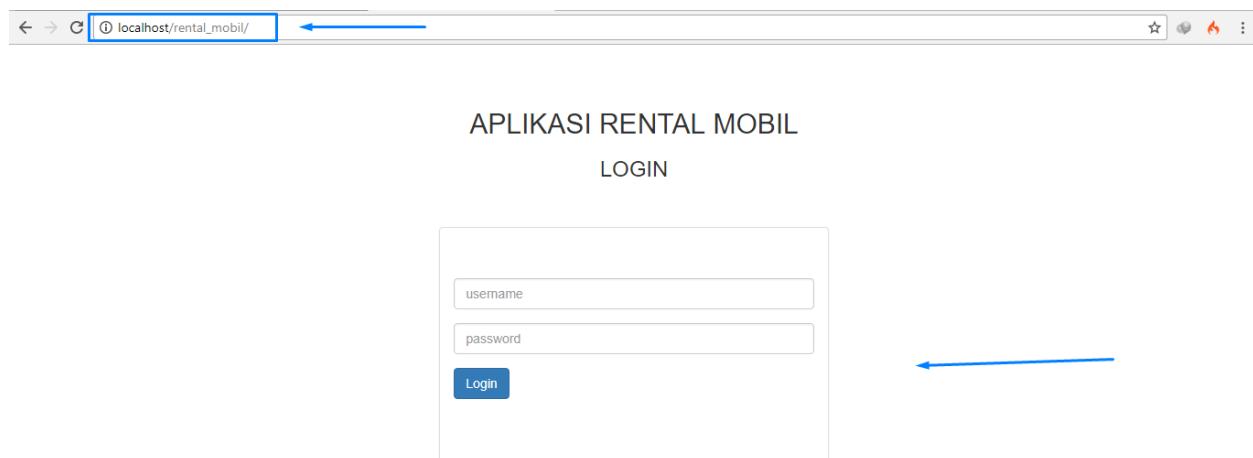
```

Tab Size: 5 PHP

Sampai di sini, login yang kita buat telah selesai. Silahkan buka controller welcome pada browser untuk melihat hasilnya.

http://localhost/rental_mobil/

perlu di perhatikan di sini, saat kita mengakses http://localhost/rental_mobil/ sama saja seperti kita mengakses http://localhost/rental_mobil/welcome. karena controller welcome telah kita set sebagai controller default.

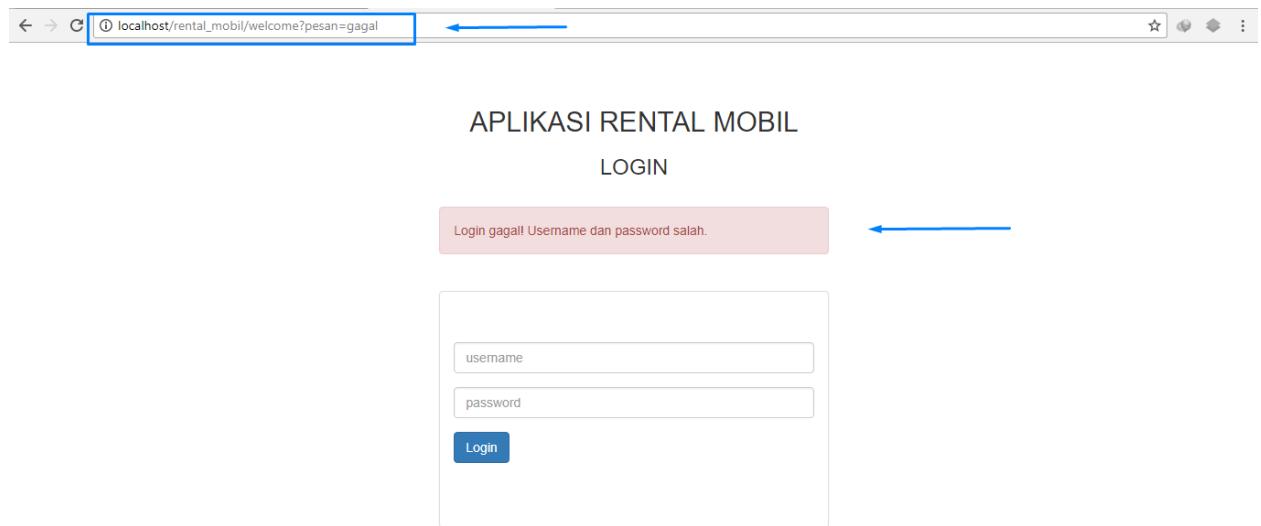


Jika teman-teman mengikuti panduan sebelumnya tentang membuat akun admin/menginsert data username dan password admin ke tabel admin di atas, maka akun adminnya adalah :

Username : admin

Password : admin123

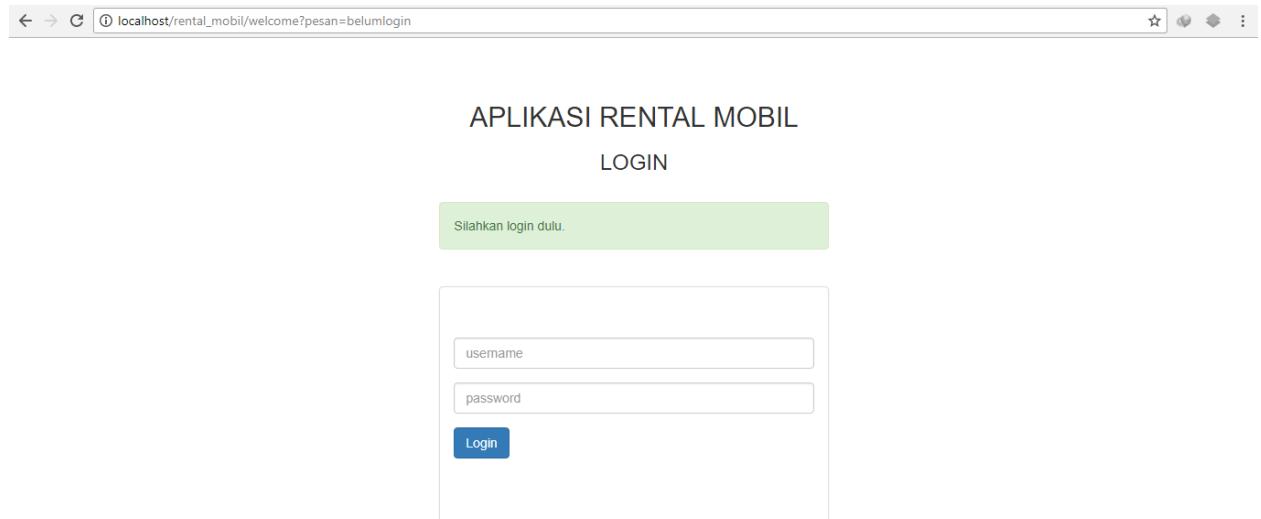
Pertama saya akan mencoba memasukkan username dan password yang salah. Untuk memastikan sistem login yang kita buat sudah berjalan sebagaimana mestinya. Di sini saya mencoba memasukkan username : samsul, dan passwordnya samsul123.



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/rental_mobil/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.

The screenshot shows the Admin Dashboard with the URL `localhost/rental_mobil/admin` in the browser's address bar. The dashboard has a top navigation bar with links: Rental Mobil, Dashboard (which is active), Data Mobil, Data Kostumer, Transaksi Rental, Laporan, Logout, and a user profile for Diki Alfarabi Hadi. Below the navigation is a main section titled "Dashboard" containing four cards:

- Jumlah Mobil:** 0. Includes a "View Details" button with a right arrow icon.
- Jumlah Kostumer:** 0. Includes a "View Details" button with a right arrow icon.
- Jumlah Transaksi:** 0. Includes a "View Details" button with a right arrow icon.
- Rental Selesai:** 0. Includes a "View Details" button with a right arrow icon.

Below these cards are three boxes:

- Mobil:** Includes a "Lihat Semua Mobil" button with a right arrow icon.
- Kostumer Terbaru:** Includes a "Lihat Semua Kostumer" button with a right arrow icon.
- Peminjaman Terakhir:** Includes a "Lihat Semua Transaksi" button with a right arrow icon. This box also contains a table header with columns: Tgl. Transaksi, Tgl. Pinjam, Tgl. Kembali, and Total.

Login selesai.

Membuat Logout

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 logout.

```
<li><a href="<?php echo base_url().'admin/logout'; ?>"><span class="glyphicon glyphicon-log-out"></span> Logout</a></li>
```

Link logout di atas, mengarahkan ke function/method **logout** di controller **admin**.

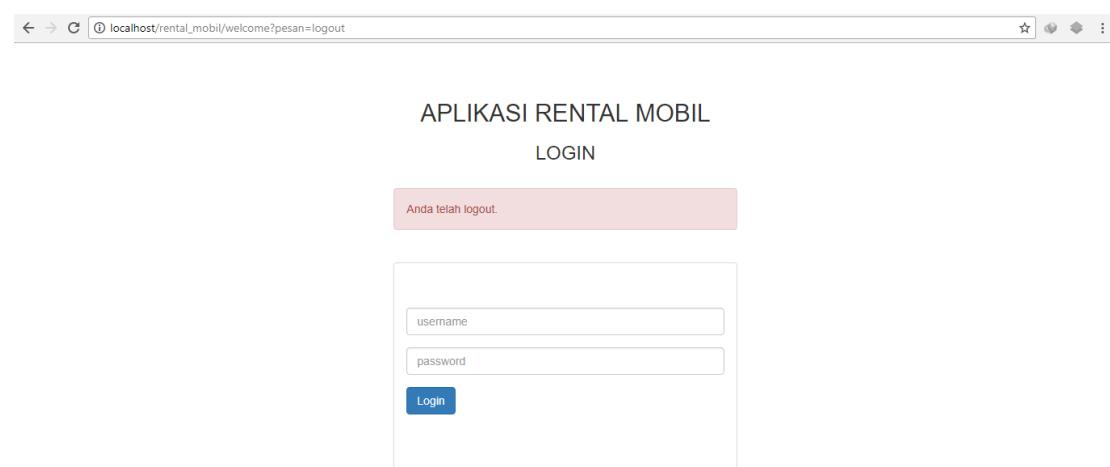
Oleh karena itu kita akan membuat function **logout** pada controller **admin**.

Tambahkan function logout berikut ini ke controller admin.

```
function logout() {  
    $this->session->sess_destroy();  
    redirect(base_url().'welcome?pesan=logout');  
}
```

Syntax `$this->session->sess_destroy();` merupakan fungsi codeigniter untuk menghapus semua **session**. Setelah menghapus session, kita alihkan halaman ke halaman login.

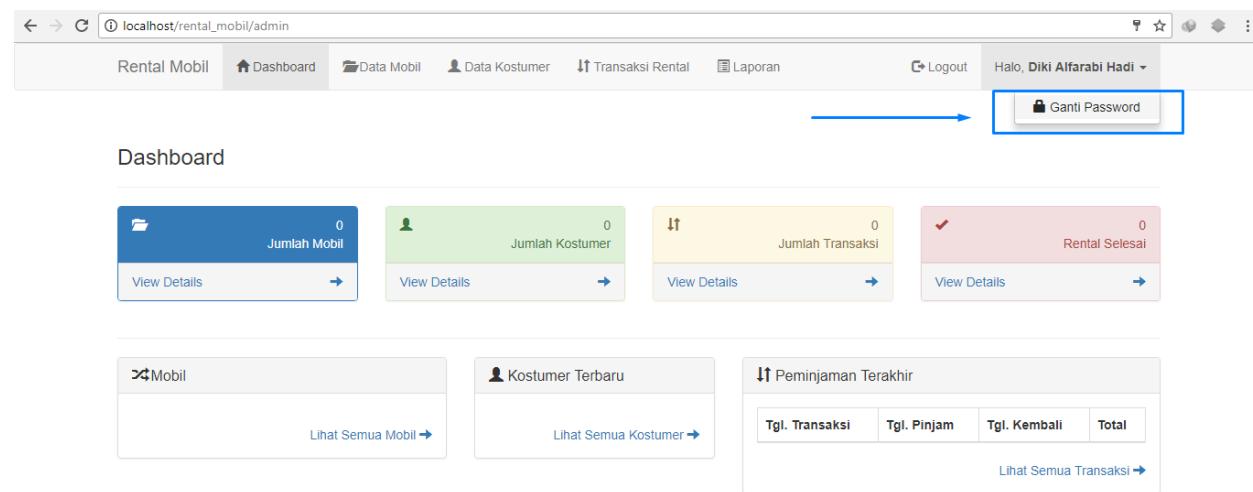
Sekarang coba klik pada menu logout. Maka halaman akan di alihkan kembali ke halaman login.



Membuat Fitur Ganti Password Admin

Sampai pada tahap ini, kita akan membuat sebuah fitur untuk admin, yaitu fitur untuk mengganti password. 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**.

```
<a href="=base_url('admin/ganti_password')?"&gt;&lt;i class="glyphicon glyphicon-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/controller/Admin.php

```
function ganti_password() {
    $this->load->view('admin/header');
    $this->load->view('admin/ganti_password');
    $this->load->view('admin/footer');
}
```

Seperti yang teman-teman lihat pada function di atas, kita menampilkan view **header.php**, **view ganti_password.php** dan **view footer.php** pada view

ganti_password.php akan kita buat form untuk mengganti password. Yuk kita buat sebuah file baru untuk view. Buat file baru dengan nama ganti_password.php. dalam folder admin dalam view.

Application/view/admin/ganti_password.php

```
<div class="page-header">
    <h3>Ganti Password</h3>
</div>

<div class="row">
    <div class="col-md-6 col-md-offset-3">
        <?php
            if(isset($_GET['pesan'])) {
                if($_GET['pesan'] ==
"berhasil") {
                    echo "<div class='alert alert-success'>Password berhasil di ganti.</div>";
                }
            }
        ?>
        <form action="<?php echo
base_url().'admin/ganti_password_act' ?>" method="post">
            <div class="form-group">
                <label>Password Baru</label>
                <input class="form-control" type="password"
name="pass_baru">
                    <?php echo form_error('pass_baru'); ?>
            </div>
            <div class="form-group">
                <label>Ulangi Password Baru</label>
                <input class="form-control" type="password"
name="ulang_pass">
                    <?php echo form_error('ulang_pass'); ?>
            </div>
            <div class="form-group">
                <input class="btn btn-primary btn-sm"
type="submit" value="Simpan">
            </div>
        </form>
    </div>
</div>
```

Aksi dari form ini akan di jalankan pada function **ganti_password_act** pada controller **admin**.

```
<form action="<?php echo base_url().'admin/ganti_password_act' ?>">
    method="post">
```

Jadi sekarang buat function **ganti_password_act()** pada controller **admin**.

Application/controllers/Admin.php

```
function ganti_password_act() {
    $pass_baru = $this->input->post('pass_baru');
    $ulang_pass = $this->input->post('ulang_pass');

    $this->form_validation->set_rules('pass_baru', 'Password Baru', 'required|matches[ulang_pass]');
    $this->form_validation->set_rules('ulang_pass', 'Ulangi Password Baru', 'required');

    if($this->form_validation->run() != false) {
        $data = array(
            'admin_password' => md5($pass_baru)
        );
        $w = array(
            'admin_id' => $this->session->userdata('id')
        );
        $this->m_rental->update_data($w, $data, 'admin');

        redirect(base_url().'admin/ganti_password?pesan=berhasil');

    } else{
        $this->load->view('admin/header');
        $this->load->view('admin/ganti_password');
        $this->load->view('admin/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 ulangi password.

```
$pass_baru = $this->input->post('pass_baru');
$ulang_pass = $this->input->post('ulang_pass');
$this->form_validation->set_rules('pass_baru', 'Password Baru', 'required|matches[ulang_pass]');
```

```
$this->form_validation->set_rules('ulang_pass','Ulangi Password Baru','required');
```

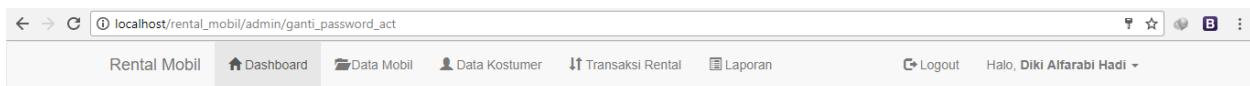
Kemudian kita cek validasinya, jika sesuai dan tidak ada masalah, maka kita update data password si admin.

```
if($this->form_validation->run() != false){  
    $data = array(  
        'admin_password' => md5($pass_baru)  
    );  
    $w = array(  
        'admin_id' => $this->session->userdata('id')  
    );  
    $this->m_rental->update_data($w,$data,'admin');  
    redirect(base_url().'admin/ganti_password?pesan=berhasil');  
  
} else{  
    $this->load->view('admin/header');  
    $this->load->view('admin/ganti_password');  
    $this->load->view('admin/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**.

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 di ketikkan pada form ulang password. Ini berguna untuk memvalidasi dan mencegah admin salah memasukkan password barunya (salah 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.



Ganti Password

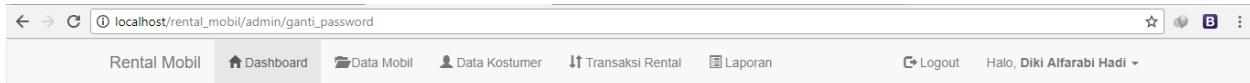
>Password Baru

The Password Baru field does not match the Ulangi Password Baru field.

Ulangi Password Baru

Simpan

Coba kita masukkan password baru yang sesuai, pada form password baru kita ketik “malasngoding123”. Dan pada form ulangi password baru kita ketik “malasngoding123”.



Ganti Password

Password Baru

.....

Ulangi Password Baru

.....

Simpan

Maka password admin pun berhasil di ganti. Dan muncul pesan notifikasi “password berhasil di ganti.”.

The screenshot shows a web application interface. At the top, there is a header with navigation links: 'Rental Mobil', 'Dashboard', 'Data Mobil', 'Data Kostumer', 'Transaksi Rental', 'Laporan', 'Logout', and a user profile 'Halo, Diki Alfarabi Hadi'. Below the header, the main content area has a title 'Ganti Password'. A green success message box contains the text 'Password berhasil di ganti.' Below the message are two input fields labeled 'Password Baru' and 'Ulangi Password Baru', both with placeholder text. A blue 'Simpan' button is located at the bottom of the form.

Sampai sini kita telah selesai membuat fitur ganti password untuk akun admin.

Membuat CRUD Data Mobil

Cara membuat CRUD sudah di jelaskan pada tutorial codeigniter bagian dasar sebelumnya. Pada tahapan ini kita akan membuat CRUD untuk data-data mobil. Seperti perancangan database khusus nya pada tabel mobil yang sudah kita buat, sesuai pada gambar perancangan database di atas, data-data mobil yang akan kita simpan adalah data id mobil (mobil_id), merk mobil (mobil_merk), nomor plat mobil(mobil_plat), warna mobil (mobil_warna), tahun mobil (mobil_tahun) dan status ketersediaan mobil (mobil_status).

Menampilkan Data Mobil dari database

Data mobil akan kita tampilkan pada function/method mobil(). Sesuai dengan link yang sudah kita tentukan di menu “Data Mobil” pada menu navigasi.

```
<li><a href="<?php echo base_url().'admin/mobil'; ?>"><span class="glyphicon glyphicon-folder-open"></span> Data Mobil</a></li>
```

Tambahkan sebuah function dengan mobil() pada controller admin.php.

Application/controllers/Admin.php

```
// CRUD MOBIL
function mobil()
{
    $data['mobil'] = $this->m_rental->get_data('mobil')->result();
    $this->load->view('admin/header');
    $this->load->view('admin/mobil',$data);
    $this->load->view('admin/footer');
}
```

Pertama, kita akan mengambil data mobil dari database dengan menggunakan function **get_data()** di model **m_rental**.

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

Dari data yang di dapatkan dari hasil query di function edit_data() di atas kita return kembali untuk kita kita masukkan ke variabel array mobil.

```

$data['mobil'] = $this->m_rental->get_data('mobil')->result();
$this->load->view('admin/header');
$this->load->view('admin/mobil',$data);
$this->load->view('admin/footer');

```

Lalu kita parsing datanya ke view mobil.php seperti pada syntax di atas. jadi di sini pada view mobil.php lah kita akan menampilkan data mobilnya.

Buat view baru lagi di dalam folder admin dengan nama mobil.php

Application/view/admin/mobil.php

```

<div class="page-header">
    <h3>Data Mobil</h3>
</div>

<a href="=base_url('admin/mobil_add');?" class="btn btn-primary btn-sm"><span class="glyphicon glyphicon-plus"></span> Mobil Baru</a>
<br/><br/>
<div class="table-responsive">
<table class="table table-bordered table-striped table-hover" id="table-datable">
    <thead>
        <tr>
            <th>No</th>
            <th>Merk Mobil</th>
            <th>Plat</th>
            <th>Warna</th>
            <th>Tahun Pembuatan</th>
            <th>Status</th>
            <th></th>
        </tr>
    </thead>
    <tbody>
        <?php
        $no = 1;
        foreach($mobil as $m) {
            ?>
            <tr>
                <td><?php echo $no++; ?></td>
                <td><?php echo $m->mobil_merk ?></td>
                <td><?php echo $m->mobil_plat ?></td>
                <td><?php echo $m->mobil_warna ?></td>
                <td><?php echo $m->mobil_tahun ?></td>
                <td>
                    <?php
                    if($m->mobil_status == "1") {
                        echo "Tersedia";
                    } else if($m->mobil_status == "2") {

```

```

        echo "Sedang Di Rental";
    }
?>
</td>
<td>
 Edit
 Hapus
</td>
</tr>
<?php
}
?>
</tbody>
</table>
</div>

```

Sekarang coba kita klik pada menu data mobil. Maka halaman akan di arahkan ke method/function mobil(). Dan data mobil pun berhasil di tampilkan.

Sampai di sini kita telah selesai menampilkan data mobil. Tapi karena kita masih belum memiliki data di tabel mobil, jadi masih kosong. Selanjutnya kita akan membuat form input data mobil.

Input Data Mobil ke Database

Form input data mobil akan kita buat pada method/function **mobil_add()** dalam controller **Admin.php**. seperti yang sudah kita arahkan pada tombol “Mobil Baru”.

```
<a href=<?php echo base_url().'admin/mobil_add'; ?>" class="btn  
btn-primary btn-sm"><span class="glyphicon glyphicon-plus"></span>  
Mobil Baru</a>
```

Buat function **mobil_add()** baru pada controller **Admin.php**.

Application/controller/Admin.php

```
function mobil_add() {  
    $this->load->view('admin/header');  
    $this->load->view('admin/mobil_add');  
    $this->load->view('admin/footer');  
}
```

Pada function **mobil_add()** ini kita akan membuat form penginputan data mobil baru. Yaitu akan kita buat pada view **mobil_add.php**. seperti yang sudah kita tetapkan pada function **mobil_add.php**.

Application/view/admin/mobil_add.php

```
<div class="page-header">  
    <h3>Mobil Baru</h3>  
</div>  
  
<form action=<?php echo base_url().'admin/mobil_add_act'; ?>"  
method="post">  
    <div class="form-group">  
        <label>Merk Mobil</label>  
        <input type="text" name="merk" class="form-control">  
        <?php echo form_error('merk'); ?>  
    </div>  
    <div class="form-group">  
        <label>No. Plat Kendaraan</label>  
        <input type="text" name="plat" class="form-control">  
    </div>  
    <div class="form-group">  
        <label>Warna</label>  
        <input type="text" name="warna" class="form-control">  
    </div>  
    <div class="form-group">
```

```

        <label>Tahun Kendaraan</label>
        <input type="text" name="tahun" class="form-control">
    </div>
    <div class="form-group">
        <label>Status Mobil</label>
        <select name="status" class="form-control">
            <option value="1">Tersedia</option>
            <option value="2">Sedang Di Rental</option>
        </select>
        <?php echo form_error('status'); ?>
    </div>
    <div class="form-group">
        <input type="submit" value="Simpan" class="btn btn-primary">
    </div>
</div>
</form>

```

Form tambah mobil ini akan di proses pada function/method **mobil_add_act()** di controller **admin.php**.

```
<form action="<?php echo base_url().'admin/mobil_add_act' ?>" method="post">
```

Jadi kita buat lagi function baru di controller **admin.php**. yaitu function **mobil_add_act()**.

Application/controller/Admin.php

```

function mobil_add_act(){
    $merk = $this->input->post('merk');
    $plat = $this->input->post('plat');
    $warna = $this->input->post('warna');
    $tahun = $this->input->post('tahun');
    $status = $this->input->post('status');
    $this->form_validation->set_rules('merk','Merk Mobil','required');
    $this->form_validation->set_rules('status','Status Mobil','required');

    if($this->form_validation->run() != false){
        $data = array(
            'mobil_merk' => $merk,
            'mobil_plat' => $plat,
            'mobil_warna' => $warna,
            'mobil_tahun' => $tahun,
            'mobil_status' => $status
        );
    }
}

```

```

        $this->m_rental->insert_data($data,'mobil');
        redirect(base_url().'admin/mobil');
    }else{
        $this->load->view('admin/header');
        $this->load->view('admin/mobil_add');
        $this->load->view('admin/footer');
    }
}

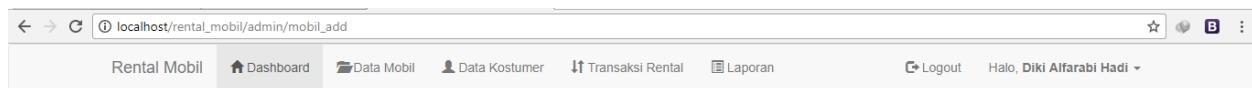
```

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 mobil baru dengan klik pada tombol “mobil baru”.

The screenshot shows a web browser window with the URL `localhost/rental_mobil/admin/mobil`. The page title is "Data Mobil". At the top, there is a navigation bar with links: "Rental Mobil", "Dashboard", "Data Mobil" (which is the active tab), "Data Kostumer", "Transaksi Rental", "Laporan", "Logout", and a user profile "Halo, Diki Alfarabi Hadi". Below the navigation, the main content area has a table titled "Data Mobil". The table header includes columns: "No", "Merk Mobil", "Plat", "Warna", "Tahun Pembuatan", and "Status". A blue box highlights the "Merk Mobil" column header. In the top-left corner of the table header, there is a blue button labeled "+ Mobil Baru". To the right of the table, there is a search bar labeled "Search: []". Below the table, it says "Showing 0 to 0 of 0 entries". At the bottom of the page, there are "Previous" and "Next" buttons.

Kemudian akan di alihkan ke halaman tambah mobil baru. Lalu isikan data mobil baru.



Dan klik simpan. Maka data mobil akan di simpan ke database.

Screenshot of the 'Data Mobil' (Car Data) page. The table shows one entry:

No	Merk Mobil	Plat	Warna	Tahun Pembuatan	Status	Actions
1	Toyota Avanza	B 7777 XX	Hitam Metalik	2015	Tersedia	Edit Hapus

Showing 1 to 1 of 1 entries

Oke, data mobil pun berhasil di tambahkan. Selanjutnya kita akan membuat fitur untuk mengedit data mobil.

Update Data Mobil di Database

Untuk megredit data mobil, akan kita buat sebuah form baru lagi pada function **mobil_edit()** di controller **admin.php**. seperti yang telah kita arahkan pada tombol edit di data mobil.

```
<a class="btn btn-warning btn-sm" href="<?php echo  
base_url().'admin/mobil_edit/'.\$m->mobil_id; ?>"><span class="glyphicon  
glyphicon-plus"></span> Edit</a>
```

Buat function baru lagi dengan nama **mobil_edit()** dalam controller **admin.php**.

Application/controller/Admin.php

```
function mobil_edit($id){  
    $where = array(  
        'mobil_id' => $id  
    );  
    $data['mobil'] = $this->m_rental-  
>edit_data($where,'mobil')->result();  
    $this->load->view('admin/header');  
    $this->load->view('admin/mobil_edit',$data);  
    $this->load->view('admin/footer');  
}
```

Di sini kita menangkap data id mobil yang ingin di edit di dalam parameter function **mobil_edit()**. Kemudian kita ambil data mobil yang memiliki id tersebut, kemudian kita parsing ke view **mobil_edit.php**. karena pada view **mobil_edit.php** inilah akan kita buat form untuk mengedit data mobil.

Application/view/admin/mobil_edit.php

```
<div class="page-header">  
    <h3>Edit Mobil</h3>  
</div>  
  
<?php foreach($mobil as $m){ ?>  
<form action="<?php echo base_url().'admin/mobil_update' ?>"  
method="post">  
    <div class="form-group">  
        <label>Merk Mobil</label>  
        <input type="hidden" name="id" value="<?php echo \$m-  
>mobil_id; ?>">  
        <input class="form-control" type="text" name="merk"  
value="<?php echo \$m->mobil_merk; ?>">  
        <?php echo form_error('merk'); ?>
```

```

        </div>

        <div class="form-group">
            <label>No. Plat Kendaraan</label>
            <input class="form-control" type="text" name="plat"
value="<?php echo $m->mobil_plat; ?>">
        </div>

        <div class="form-group">
            <label>Warna</label>
            <input class="form-control" type="text" name="warna"
value="<?php echo $m->mobil_warna; ?>">
        </div>

        <div class="form-group">
            <label>Tahun Kendaraan</label>
            <input class="form-control" type="text" name="tahun"
value="<?php echo $m->mobil_tahun; ?>">
        </div>

        <div class="form-group">
            <label>Status Mobil</label>
            <select name="status" class="form-control">
                <option <?php if($m->mobil_status == "1"){echo
"selected='selected'"; } echo $m->mobil_tahun; ?>
value="1">Tersedia</option>
                <option <?php if($m->mobil_status == "2"){echo
"selected='selected'"; } echo $m->mobil_tahun; ?> value="2">Sedang Di
Rental</option>
            </select>
            <?php echo form_error('status'); ?>
        </div>

        <div class="form-group">
            <input type="submit" value="Simpan" class="btn btn-
primary">
        </div>
    </form>
<?php } ?>

```

Form edit data mobil ini akan di proses pada function **mobil_update()** dalam controller **admin.php** pada saat di submit.

```
<form action="<?php echo base_url().'admin/mobil_update' ?>" method="post">
```

Jadi buat lagi sebuah function baru dengan nama **mobil_update()** pada controller **Admin.php**.

Application/controller/Admin.php

```
function mobil_update() {
    $id = $this->input->post('id');
    $merk = $this->input->post('merk');
    $plat = $this->input->post('plat');
    $warna = $this->input->post('warna');
    $tahun = $this->input->post('tahun');
    $status = $this->input->post('status');
    $this->form_validation->set_rules('merk', 'Merk
Mobil','required');
    $this->form_validation->set_rules('status', 'Status
Mobil','required');
    if($this->form_validation->run() != false) {
        $where = array(
            'mobil_id' => $id
        );
        $data = array(
            'mobil_merk' => $merk,
            'mobil_plat' => $plat,
            'mobil_warna' => $warna,
            'mobil_tahun' => $tahun,
            'mobil_status' => $status
        );
        $this->m_rental->update_data($where,$data,'mobil');
        redirect(base_url().'admin/mobil');
    }else{
        $where = array(
            'mobil_id' => $id
        );
        $data['mobil'] = $this->m_rental-
>edit_data($where,'mobil')->result();
        $this->load->view('admin/header');
        $this->load->view('admin/mobil_edit',$data);
        $this->load->view('admin/footer');
    }
}
```

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 mobil “toyota avanza” ke “honda jazz”.

Klik pada tombol edit.

Rental Mobil	Dashboard	Data Mobil	Data Kostumer	Transaksi Rental	Laporan	Logout	Halo, Diki Alfarabi Hadi
Data Mobil							

+ Mobil Baru	Search:
10 records per page	
Showing 1 to 1 of 1 entries	
← Previous	1
Next →	

Maka akan di alihkan ke halaman edit mobil.

Edit Mobil	
Merk Mobil	<input type="text" value="Toyota Avanza"/>
No. Plat Kendaraan	<input type="text" value="B 7777 XX"/>
Warna	<input type="text" value="Hitam Metalik"/>
Tahun Kendaraan	<input type="text" value="2015"/>
Status Mobil	<input type="text" value="Tersedia"/>
Simpan	

Lalu ubah merk mobil dengan “honda jazz”. Dan klik simpan.

No	Merk Mobil	Plat	Warna	Tahun Pembuatan	Status
1	Honda Jazz	B 7777 XX	Hitam Metalik	2015	Tersedia

Showing 1 to 1 of 1 entries

← Previous 1 Next →

Dan data pun berhasil di edit. Dan yang terakhir kita akan membuat hapus data dari database ketika tombol hapus di klik.

Menghapus Data Mobil dari Database

Sesuai dengan link yang sudah kita tentukan pada tombol hapus di table data mobil seperti di atas.

```
<a class="btn btn-danger btn-sm" href="php echo base_url().'admin/mobil_hapus/'.$m-&gt;mobil_id; ?&gt;"&gt;&lt;span class="glyphicon glyphicon-trash"&gt;&lt;/span&gt; Hapus&lt;/a&gt;</pre

```

Untuk aksi dari penghapusan data mobil kita akan membuatnya di function/method `mobil_hapus()` dalam controller admin.

Buat sebuah method baru dengan nama `mobil_hapus()` di controller `admin.php`.

Application/controllers/admin.php

```
function mobil_hapus($id) {
    $where = array(
        'mobil_id' => $id
    );
    $this->m_rental->delete_data($where, 'mobil');
```

```
    redirect(base_url().'admin/mobil');
}
// AKHIR CRUD MOBIL
```

Untuk lebih detail tentang menghapus data dari database menggunakan codeigniter, silahkan teman-teman baca kembali pada tutorial CRUD sebelumnya di buku ini. Yang sudah di jelaskan secara detail.

Sekarang coba klik tombol hapus pada mobil yang ingin di hapus.

The screenshot shows a web browser window with the URL `localhost/rental_mobil/admin/mobil`. The page title is "Data Mobil". At the top, there is a navigation bar with links: "Rental Mobil", "Dashboard", "Data Mobil" (which is active), "Data Kostumer", "Transaksi Rental", "Laporan", "Logout", and a user profile "Halo, Diki Alfarabi Hadi". Below the navigation bar is a table titled "Data Mobil" with the following columns: No, Merk Mobil, Plat, Warna, Tahun Pembuatan, Status, + Edit, and Hapus. The table contains one entry: No 1, Merk Mobil Honda Jazz, Plat B 7777 XX, Warna Hitam Metalik, Tahun Pembuatan 2015, Status Tersedia, + Edit button, and a red-bordered Hapus button. A blue arrow points from the text "Sekarang coba klik tombol hapus pada mobil yang ingin di hapus." to the Hapus button.

Dan mobil pun berhasil di hapus.

proses hapus data mobil 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.

Membuat CRUD Data Kostumer

Setelah selesai membuat CRUD data mobil, selanjutnya kita akan membuat crud data kostumer, tujuan kita membuat crud kostumer adalah untuk menyimpan data kostumer atau pelanggan. Sesuai dengan perancangan relasi database yang telah kita buat pada gambar xxxx , proses transaksi nantinya akan kita buat dengan memilih pelanggan yang ingin menyewa mobil, kemudian memilih mobil yang ingin di sewa, baru kemudian transaksi akan di simpan.

Menampilkan Data Kostumer dari database

Untuk menampilkan data kostumer dari database, kita akan membuat function atau method baru pada controller admin.php. function yang akan kita buat adalah **kostumer()**.

Application/controller/Admin.php

```
// CRUD KOSTUMER
function kostumer(){
    $data['kostumer'] = $this->m_rental->get_data('kostumer')-
>result();
    $this->load->view('admin/header');
    $this->load->view('admin/kostumer',$data);
    $this->load->view('admin/footer');
}
```

Sama saja seperti *cara menampilkan data mobil dari database* yang telah kita bahas sebelumnya.

Disini pertama kita mengambil data kostumer dari table 'kostumer'.

```
$data['kostumer'] = $this->m_rental->get_data('kostumer')->result();
```

Kemudian kita masukkan ke variabel array data. Dan kemudian kita parsing ke view kostumer.php.

Selanjutnya kita buat view baru lagi dengan nama kostumer.php pada folder view admin.

Application/view/admin/kostumer.php

```
<div class="page-header">
    <h3>Data Kostumer</h3>
</div>

<a href="php echo base_url().'admin/kostumer_add'; ?" class="btn
btn-sm btn-primary"><span class='glyphicon glyphicon-plus'></span>
Kostumer Baru</a>
<br/>
<br/>
<div class="table-responsive">
    <table class="table table-bordered table-hover table-striped"
id="table-datable">
        <thead>
            <tr>
                <th>No</th>
                <th>>Nama</th>
                <th>Jenis Kelamin</th>
                <th>Alamat</th>
                <th>HP</th>
                <th>No. KTP</th>
                <th></th>
            </tr>
        </thead>
        <tbody>
            <?php
                $no = 1;
                foreach($kostumer as $k) {
                    ?>
                    <tr>
                        <td><?php echo $no++; ?></td>
                        <td><?php echo $k->kostumer_nama ?></td>
                        <td><?php echo $k->kostumer_jk ?></td>
                        <td><?php echo $k->kostumer_alamat
?></td>
                        <td><?php echo $k->kostumer_hp ?></td>
                        <td><?php echo $k->kostumer_ktp ?></td>

                        <td>
                            <a class="btn btn-sm btn-warning"
href="php echo base_url().'admin/kostumer_edit/'.$k-&gt;kostumer_id;
?&gt;"&gt;&lt;span class="glyphicon glyphicon-wrench"&gt;&lt;/span&gt; Edit&lt;/a&gt;
                            &lt;a class="btn btn-sm btn-danger"
href="<?php echo base_url().'admin/kostumer_hapus/'.$k-&gt;kostumer_id;
?&gt;"&gt;&lt;span class="glyphicon glyphicon-trash"&gt;&lt;/span&gt; Hapus&lt;/a&gt;
                        &lt;/td&gt;
                    &lt;/tr&gt;
                &lt;?php
            }
            ?&gt;
        &lt;/tbody&gt;</pre
```

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

Data kostumer yang kita parsing dari controller tadi kita tangkap dan menggunakan function foreach() untuk di tampilkan dengan perulangan.

```
<table class="table table-bordered table-hover table-striped"
id="table-datatable">
    <thead>
        <tr>
            <th>No</th>
            <th>Nama</th>
            <th>Jenis Kelamin</th>
            <th>Alamat</th>
            <th>HP</th>
            <th>No. KTP</th>
            <th></th>
        </tr>
    </thead>
    <tbody>
        <?php
        $no = 1;
        foreach($kostumer as $k) {
            ?>
            <tr>
                <td><?php echo $no++; ?></td>
                <td><?php echo $k->kostumer_nama ?></td>
                <td><?php echo $k->kostumer_jk ?></td>
                <td><?php echo $k->kostumer_alamat ?></td>
                <td><?php echo $k->kostumer_hp ?></td>
                <td><?php echo $k->kostumer_ktp ?></td>

                <td>
                    <a class="btn btn-sm btn-warning"
                    href="<?php echo base_url().'admin/kostumer_edit/'.$k->kostumer_id;
                    ?>"><span class="glyphicon glyphicon-wrench"></span> Edit</a>
                    <a class="btn btn-sm btn-danger"
                    href="<?php echo base_url().'admin/kostumer_hapus/'.$k->kostumer_id;
                    ?>"><span class="glyphicon glyphicon-trash"></span> Hapus</a>
                </td>
            </tr>
        <?php
        }
        ?>
    </tbody>
</table>
```

Seperti yang terlihat pada gambar xx di atas, kita telah berhasil membuat tabel yang menampilkan data kostumer. Karena kita belum memiliki data kostumer di table kostumer, maka data masih kosong.

Input Data Kostumer Ke Database

Dan seperti yang telah teman-teman lihat pada syntax view kostumer.php di atas. kita telah membuat juga tombol untuk mengalihkan halaman ke method kostumer_add() di controller Admin.php, karena pada method nantinya akan kita buat halaman untuk input data kostumer ke database. Yaitu tombol "kostumer baru".

```
<a href="php echo base_url().'admin/kostumer add'; ?&gt;" class="btn btn-sm btn-primary"&gt;&lt;span class='glyphicon glyphicon-plus'&gt;&lt;/span&gt; Kostumer Baru&lt;/a&gt;</pre
```

Maka dari itu, akan kita buat method baru yaitu kostumer_add() pada controller Admin.php.

Application/controller/Admin.php

```
function kostumer_add() {
    $this->load->view('admin/header');
    $this->load->view('admin/kostumer_add');
```

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

Pada method atau function kostumer_add() ini kita hanya menampilkan view kostumer_add.php. pada view kostumer_add.php ini kita akan membuat form input data kostumer.

Buat view baru dalam folder admin. Dengan nama kostumer_add.php

Application/view/admin/kostumer_add.php

```
<div class="page-header">
    <h3>Kostumer Baru</h3>
</div>

<form action=<?php echo base_url().'admin/kostumer_add_act' ?>" method="post">
    <div class="form-group">
        <label>Nama Kostumer</label>
        <input type="text" name="nama" class="form-control">

        <?php echo form_error('nama'); ?>
    </div>
    <div class="form-group">
        <label>Alamat</label>
        <textarea name="alamat" class="form-control"></textarea>
    </div>
    <div class="form-group">
        <label>Jenis Kelamin</label>
        <select name="jk" class="form-control">
            <option value="L">Laki-laki</option>
            <option value="P">Perempuan</option>
        </select>
    </div>
    <div class="form-group">
        <label>HP</label>
        <input type="number" name="hp" class="form-control">
    </div>
    <div class="form-group">
        <label>No. KTP</label>
        <input type="text" name="ktp" class="form-control">
    </div>
    <div class="form-group">
        <input type="submit" value="Simpan" class="btn btn-primary">
    </div>
```

```
</form>
```

Kostumer Baru

Nama Kostumer

Alamat

Jenis Kelamin

Laki-laki

HP

No. KTP

Simpan

Form input data kostumer ini akan di proses pada method kostumer_add_act() sesuai dengan yang sudah kita setting pada form di atas.

```
<form action="php echo base_url().'admin/kostumer_add_act' ?&gt;"<br/method="post">
```

Maka sekarang buat sebuah method baru lagi dengan nama kostumer_add_act() pada controller Admin.php.

Application/controllers/Admin.php

```
function kostumer_add_act(){  
    $nama = $this->input->post('nama');  
    $alamat = $this->input->post('alamat');  
    $jk = $this->input->post('jk');  
    $hp = $this->input->post('hp');  
    $ktp = $this->input->post('ktp');  
    $this->form_validation->set_rules('nama','nama','required');  
  
    if($this->form_validation->run() != false){  
        $data = array(  
            'kostumer_nama' => $nama,  
            'kostumer_alamat' => $alamat,  
            'kostumer_jk' => $jk,  
            'kostumer_hp' => $hp,  
            'kostumer_ktp' => $ktp  
    }
```

```

    );
    $this->m_rental->insert_data($data, 'kostumer');
    redirect(base_url().'admin/kostumer');
} else{
    $this->load->view('admin/header');
    $this->load->view('admin/kostumer_add');
    $this->load->view('admin/footer');
}
}

```

Untuk proses aksi dari penginputan data kostumer ini sama saja intinya seperti proses aksi penginputan data mobil. Bedanya data mobil kita simpan ke table ‘mobil’. Dan data kostumer kita simpan ke table ‘kostumer’. Setelah proses input data selesai, halaman akan langsung kita alihkan ke function kostumer() di controller admin.php.

```
redirect(base_url().'admin/kostumer');
```

Sekarang akan kita coba menginput data kostumer. Silahkan klik pada tombol “kostumer baru”. Lalu isi data kostumer dan klik “simpan”.

Kostumer Baru

Nama Kostumer
Diki Alfarabi Hadi

Alamat
Jl. planet antariksa sejati no. 123, bumi, indonesia.

Jenis Kelamin
Laki-laki

HP
082277125900

No. KTP
3294729834792

Simpan

Setelah di klik pada tombol “simpan”. Maka data akan tersimpan. Di sini saya mencoba mengisi nama kostumer dengan nama saya sendiri. “Diki Alfarabi Hadi”. Alamat “jl. Planet antariksa sejati no.123, bumi, indonesia”. Jenis kelamin “laki-laki”, hp “082277125900”. No.KTP “3294729834792”.

No	Nama	Jenis Kelamin	Alamat	HP	No. KTP
1	Diki Alfarabi Hadi	L	Jl. planet antariksa sejati. no. 123, bumi, indonesia.	082277125900	3294729834792

Dan data kostumer berhasil di simpan.

Update Data Kostumer Di Database

Untuk membuat edit dan update data kostumer, kita akan membuat function `kostumer_edit()` di controller `Admin.php`. seperti yang sudah kita tetapkan pada tombol edit di tabel yang menampilkan data kostumer seperti di atas.

```
<a class="btn btn-sm btn-warning" href="<?php echo
base_url().'admin/kostumer_edit/'.$k->kostumer_id; ?>"><span
class="glyphicon glyphicon-wrench"></span> Edit</a>
```

Buat sebuah method baru dengan nama `kostumer_edit.php()`.

Application/controller/Admin.php

```
function kostumer_edit($id) {
    $where = array(
        'kostumer_id' => $id
    );
    $data['kostumer'] = $this->m_rental-
>edit_data($where, 'kostumer')->result();
    $this->load->view('admin/header');
    $this->load->view('admin/kostumer_edit', $data);
    $this->load->view('admin/footer');
}
```

Di sini kita menangkap data id yang dikirimkan dari url. Kita menangkapnya dalam function kostumer_id(). Kemudian kita ambil data kostumer yang memiliki id sesuai dengan id yang kita tangkap. Kemudian kita parsing data nya ke view kostumer_edit.php di folder admin.

Buat view baru dengan nama kostumer_edit.php di dalam folder admin.

Application/view/admin/kostumer_edit.php

```
<div class="page-header">
    <h3>Edit Kostumer</h3>
</div>

<?php foreach($kostumer as $k){ ?>
<form action="<?php echo base_url().'admin/kostumer_update' ?>" method="post">
    <div class="form-group">
        <label>Nama Kostumer</label>
        <input type="text" name="nama" value="<?php echo $k->kostumer_nama; ?>" class="form-control">
            <input type="hidden" name="id" value="<?php echo $k->kostumer_id; ?>" class="form-control">
                <?php echo form_error('nama'); ?>
    </div>
    <div class="form-group">
        <label>Alamat</label>
        <textarea name="alamat" class="form-control"><?php echo $k->kostumer_alamat; ?></textarea>
    </div>
    <div class="form-group">
        <label>Jenis Kelamin</label>
        <select name="jk" class="form-control">
            <option value="L">Laki-laki</option>
            <option value="P">Perempuan</option>
        </select>
    </div>
    <div class="form-group">
        <label>HP</label>
        <input type="number" name="hp" value="<?php echo $k->kostumer_hp; ?>" class="form-control">
    </div>
    <div class="form-group">
        <label>No. KTP</label>
        <input type="text" name="ktp" value="<?php echo $k->kostumer_ktp; ?>" class="form-control">
    </div>
    <div class="form-group">
        <input type="submit" value="Simpan" class="btn btn-
```

```
primary">
    </div>
</form>
<?php } ?>
```

Karena form edit kostumer ini akan di proses pada function kostumer_update() di controller admin.php.

```
<form action="<?php echo base_url().'admin/kostumer_update' ?>" method="post">
```

maka kita buat sebuah function/method baru lagi di controller admin.php dengan nama kostumer_update().

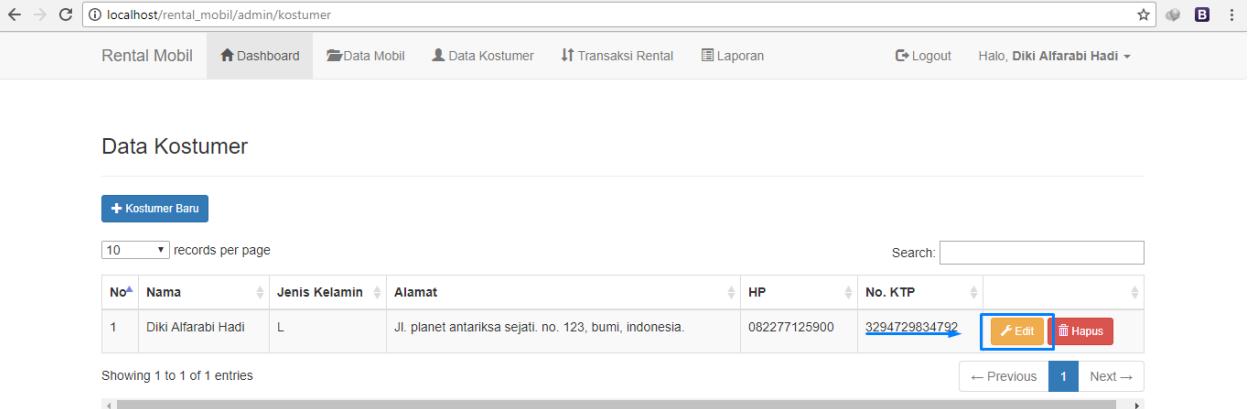
Application/controller/Admin.php

```
function kostumer_update(){
    $id = $this->input->post('id');
    $nama = $this->input->post('nama');
    $alamat = $this->input->post('alamat');
    $jk = $this->input->post('jk');
    $hp = $this->input->post('hp');
    $ktp = $this->input->post('ktp');
    $this->form_validation->set_rules('nama', 'nama', 'required');

    if($this->form_validation->run() != false) {
        $where = array(
            'kostumer_id' => $id
        );
        $data = array(
            'kostumer_nama' => $nama,
            'kostumer_alamat' => $alamat,
            'kostumer_jk' => $jk,
            'kostumer_hp' => $hp,
            'kostumer_ktp' => $ktp
        );
        $this->m_rental->update_data($where, $data, 'kostumer');
        redirect(base_url().'admin/kostumer');
    }else{
        $where = array(
            'kostumer_id' => $id
        );
        $data['kostumer'] = $this->m_rental-
>edit_data($where, 'kostumer')->result();
        $this->load->view('admin/header');
        $this->load->view('admin/kostumer_edit', $data);
        $this->load->view('admin/footer');
```

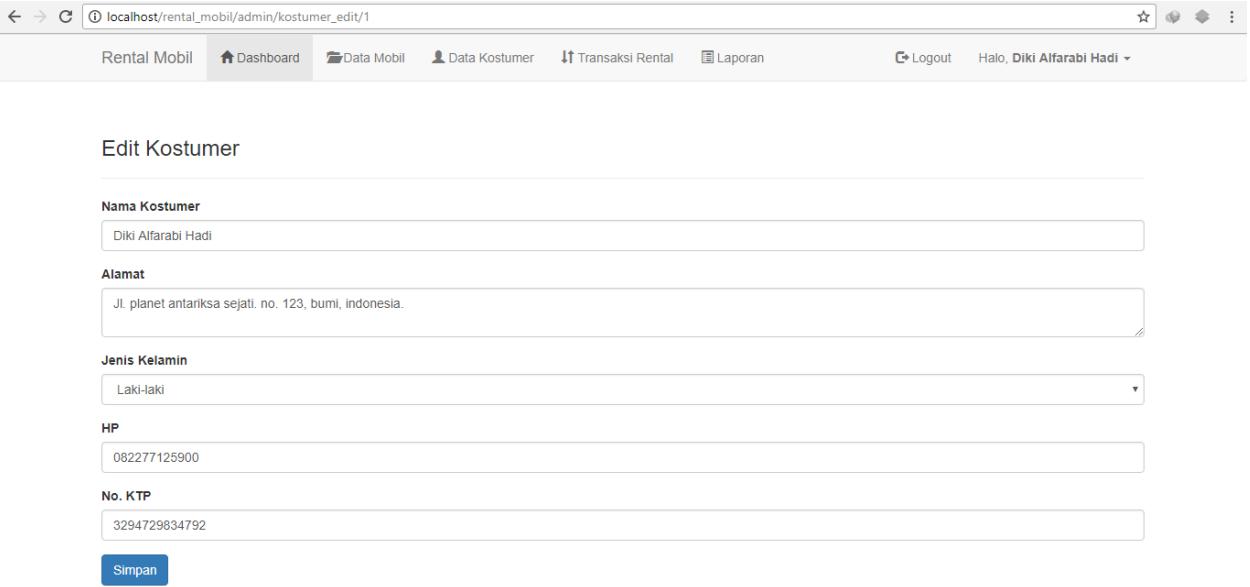
```
}
```

Dan coba sekarang kita klik pada tombol edit untuk mengubah data kostumer.



The screenshot shows a table of customer data with columns: No, Nama, Jenis Kelamin, Alamat, HP, and No. KTP. The first row contains the data: 1, Diki Alfarabi Hadi, L, Jl. planet antariksa sejati. no. 123, bumi, indonesia., 082277125900, 3294729834792. To the right of the table are two buttons: 'Edit' (highlighted with a blue box) and 'Hapus'. Below the table, there are navigation links for 'Previous', 'Next', and a page number '1'.

Maka kita akan di alihkan ke form edit kostumer.



The screenshot shows an 'Edit Kostumer' form with the following fields:
Nama Kostumer: Diki Alfarabi Hadi
Alamat: Jl. planet antariksa sejati. no. 123, bumi, indonesia.
Jenis Kelamin: Laki-laki
HP: 082277125900
No. KTP: 3294729834792
A blue box highlights the 'Simpan' button at the bottom of the form.

Di sini kita akan mencoba mengubah alamat dan nomor ktp dari kostumer.

Rental Mobil Dashboard Data Mobil Data Kostumer Transaksi Rental Laporan Logout Halo, Diki Alfarabi Hadi

Edit Kostumer

Nama Kostumer

Diki Alfarabi Hadi

Alamat

Jl. bumi alami. no. 123, indonesia.

Jenis Kelamin

Laki-laki

HP

082277125900

No. KTP

123456789

Simpan

Di sini saya mencoba merubah alamat kostumer menjadi "Jl. bumi alami. no. 123, indonesia.", dan mengubah nomor KTP kostumer menjadi "123456789". Lalu klik simpan.

Rental Mobil Dashboard Data Mobil Data Kostumer Transaksi Rental Laporan Logout Halo, Diki Alfarabi Hadi

Data Kostumer

+ Kostumer Baru

10 records per page

Search:

No	Nama	Jenis Kelamin	Alamat	HP	No. KTP	
1	Diki Alfarabi Hadi	L	Jl. bumi alami. no. 123, indonesia.	082277125900	123456789	Edit Hapus

Showing 1 to 1 of 1 entries

← Previous 1 Next →

Dan data kostumer pun berhasil di update.

Menghapus Data Kostumer dari Database

Dan terakhir untuk crud data kostumer, kita akan membuat fungsi untuk hapus data kostumer.

Coba teman-teeman perhatikan pada tombol hapus pada table data kostumer. Saat tombol hapus di klik maka akan di alihkan ke function kostumer_hapus() di controller Admin sambil membawa id kostumer yang akan di hapus.

```
<a class="btn btn-sm btn-danger" href="php echo<br/base_url().'admin/kostumer_hapus/'.$k->kostumer_id; ?>"><span  
class="glyphicon glyphicon-trash"></span> Hapus</a>
```

Jadi aksi penghapusan data kostumer ini akan kita buat di function/method kostumer_hapus() di controller admin.

Maka kita harus membuat method baru di controller admin. Yaitu method kostumer_hapus().

Application/controllers/admin.php

```
function kostumer_hapus($id) {  
    $where = array(  
        'kostumer_id' => $id  
    );  
    $this->m_rental->delete_data($where, 'kostumer');  
    redirect(base_url().'admin/kostumer');  
}  
// AKHIR CRUD MOBIL
```

Di sini kita tanggap id kostumer 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 di alihkan ke method kostumer().

Sebenarnya cara menghapus data kostumer masih sama saja dengan cara menghapus data mobil seperti yang sudah kita buat sebelumnya.

Coba klik tombol hapus.

The screenshot shows a table with one row of data. The columns are labeled: No, Nama, Jenis Kelamin, Alamat, HP, and No. KTP. The data in the first row is: 1, Diki Alfarabi Hadi, L, Jl. bumi alami. no. 123, indonesia., 082277125900, 123456789. To the right of the table are two buttons: 'Edit' and 'Hapus'. A blue arrow points from the text 'dan data kostumer pun berhasil dihapus.' to the 'Hapus' button.

No	Nama	Jenis Kelamin	Alamat	HP	No. KTP
1	Diki Alfarabi Hadi	L	Jl. bumi alami. no. 123, indonesia.	082277125900	123456789

dan data kostumer pun berhasil dihapus.

The screenshot shows an empty table with the same column headers as the previous screenshot. The message 'No data available in table' is displayed in the center of the table area. The entire table area is enclosed in a blue box.

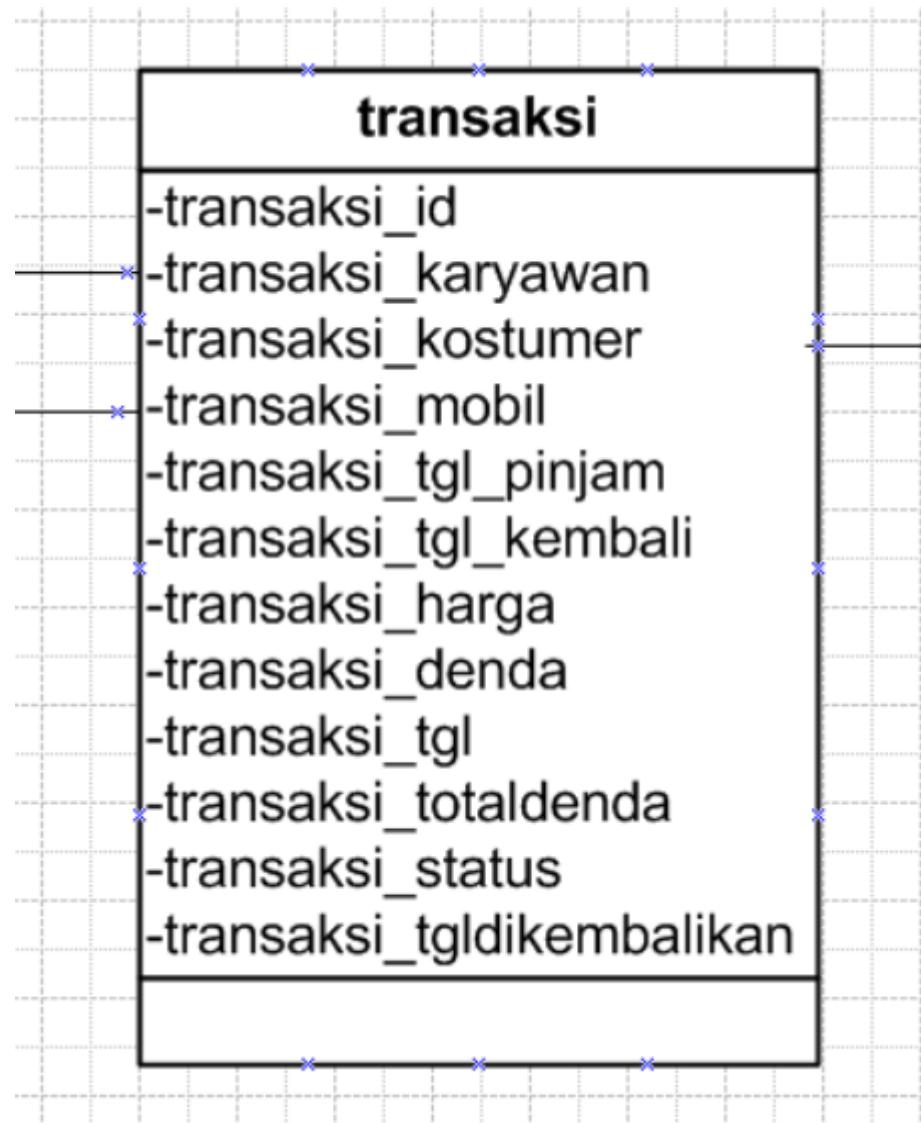
No	Nama	Jenis Kelamin	Alamat	HP	No. KTP
No data available in table					

Oke selesai sudah proses CRUD untuk kostumer. Selanjutnya kita akan lanjut ke modul transaksi rental mobil.

Membuat Proses Transaksi Rental Mobil

Setelah selesai membuat CRUD data mobil dan CRUD untuk data kostumer. Selanjutnya kita akan lanjut membuat proses transaksi rental mobil.

Silahkan perhatikan kembali perancangan table transaksi yang telah kita buat.



Seperti yang sudah di singgung sebelumnya. Pada table transaksi ini kita akan menyimpan data id karyawan yang menangani transaksi di kolumn transaksi_karyawan, menyimpan data id kostumer yang ingin menyewa mobil di kolumn transaksi_kostumer. Selengkapnya silahkan perhatikan pada table berikut.

transaksi_karyawan	Menyimpan data id karyawan yang menangani transaksi sewa/rental mobil.
---------------------------	--

transaksi_kostumer	Menyimpan id kostumer yang melakukan rental mobil.
transaksi_mobil	Menyimpan Id mobil yang di rental.
transaksi_tgl_pinjam	Menyimpan data tanggal mobil mulai di sewa.
transaksi_tgl_kembali	Menyimpan data tanggal mobil akan di kembalikan
transaksi_harga	Menyimpan harga rental mobil.
transaksi_denda	Menyimpan harga denda mobil perhari.
transaksi_tgl	Menyimpan data tanggal transaksi di lakukan.
transaksi_totaldenda	Menyimpan total denda untuk kostumer yang telat mengembalikan mobil.
transaksi_status	Menyimpan status transaksi. 0 = belum selesai (dalam masa rental). 1 = selesai.
transaksi_tgl_dikembalikan	Menyimpan tanggal di kembalikannya mobil oleh kostumer.

Menampilkan Data Transaksi Rental

Pertama kita akan menampilkan dulu data transaksi. Kita akan menampilkan data transaksi di method transaksi(). Dalam controller Admin.php. Buat sebuah method baru pada controller Admin.php. yaitu method transaksi().

Application/controllers/Admin.php

```
function transaksi(){
    $data['transaksi'] = $this->db->query("select * from
transaksi,mobil,kostumer where transaksi_mobil=mobil_id and
transaksi_kostumer=kostumer_id")->result();
    $this->load->view('admin/header');
    $this->load->view('admin/transaksi',$data);
    $this->load->view('admin/footer');
}
```

Query mysql yang kita jalankan di sini adalah “select * from transaksi,mobil,kostumer where transaksi_mobil=mobil_id and transaksi_kostumer=kostumer_id ”. yaitu untuk menampilkan atau mengambil data dari table transaksi, table mobil dan table kostumer yang mobil_id sama dengan transaksi_mobil. Dan kostumer_id sama dengan transaksi_kostumer. Setelah kita mendapatkan data kemudian kita parsing ke view transaksi.php untuk di tampilkan.

Buat view baru dalam folder admin dengan nama transksi.php

Application/view/admin/transaksi.php

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

```

<h3>Data Transaksi</h3>
</div>

<a href=<?php echo base_url() . 'admin/transaksi_add'; ?>" class="btn
btn-primary btn-sm"><span class="glyphicon glyphicon-plus"></span>
Transaksi Baru</a>
<br/><br/>
<div class="table-responsive">
    <table class="table table-bordered table-striped table-hover"
id="table-datatable">
        <thead>
            <tr>
                <th>No</th>
                <th>Tanggal</th>
                <th>Kostumer</th>
                <th>Mobil</th>
                <th>Tgl. Pinjam</th>
                <th>Tgl. Kembali</th>
                <th>Harga</th>
                <th>Denda / Hari</th>
                <th>Tgl. Dikembalikan</th>
                <th>Total Denda</th>
                <th>Status</th>
                <th></th>
            </tr>
        </thead>
        <tbody>
            <?php
            $no = 1;
            foreach($transaksi as $t) {
            ?>
            <tr>
                <td><?php echo $no++; ?></td>
                <td><?php echo date('d/m/Y', strtotime($t-
>transaksi_tgl)); ?></td>
                <td><?php echo $t->kostumer_nama; ?></td>
                <td><?php echo $t->mobil_merk; ?></td>
                <td><?php echo date('d/m/Y', strtotime($t-
>transaksi_tgl_pinjam)); ?></td>
                <td><?php echo date('d/m/Y', strtotime($t-
>transaksi_tgl_kembali)); ?></td>
                <td><?php echo "Rp. ".number_format($t-
>transaksi_harga); ?></td>
                <td><?php echo "Rp. ".number_format($t-
>transaksi_denda); ?></td>
                <td>
                    <?php
                    if($t->transaksi_tgldikembalikan == "0000-00-
00") {
                        echo "-";
                    }else{
                        echo date('d/m/Y', strtotime($t-
>transaksi_tgldikembalikan));
                    }
                </td>
            </tr>
        </tbody>
    </table>
</div>

```

```

        }
    ?>
</td>
<td><?php echo "Rp. ". number_format($t->transaksi_totaldenda).",-"; ?></td>
<td>
<?php
if($t->transaksi_status == "1") {
    echo "Selesai";
} else{
    echo "-";
}
?>
</td>
<td>
<?php
if($t->transaksi_status == "1") {
    echo "-";
} else{
    ?>
        <a class="btn btn-sm btn-success"
href=<?php echo base_url().'admin/transaksi_selesai/'.$t->transaksi_id; ?>"><span class="glyphicon glyphicon-ok"></span>
Transaksi Selesai</a>
        <br/>
        <a class="btn btn-sm btn-danger"
href=<?php echo base_url().'admin/transaksi_hapus/'.$t->transaksi_id; ?>"><span class="glyphicon glyphicon-remove"></span>
Batalkan Transaksi</a>
        <?php
    }
}
?>
</td>
</tr>
<?php
}
?>
</tbody>
</table>
</div>

```

Hasilnya seperti gambar berikut ini.

Pada table data transaksi di atas, kita telah berhasil menampilkan data transaksi. Namun masih kosong karena belum ada data transaksi yang kita input.

Oh ya, kita juga membuat kondisi, jika status transaksi nantinya sama dengan 1. Maka kita tampilkan "selesai" pada kolumn status di tabel transaksi di atas. dan pada kolumn terakhir kita membuat kondisi juga, jika status transaksi belum selesai, maka kita tampilkan tombol "batalkan transaksi" dan tombol "transaksi selesai".

```
<td>
    <?php
    if($t->transaksi_status == "1") {
        echo "Selesai";
    }else{
        echo "-";
    }
    ?>
</td>
<td>
    <?php
    if($t->transaksi_status == "1") {
        echo "-";
    }else{
    ?>
        <a class="btn btn-sm btn-success" href="php echo
base_url().'admin/transaksi_selesai/'.$t-&gt;transaksi_id; ?&gt;"&gt;&lt;span
class="glyphicon glyphicon-ok"&gt;&lt;/span&gt; Transaksi Selesai&lt;/a&gt;
        &lt;br/&gt;
        &lt;a class="btn btn-sm btn-danger" href="<?php echo
base_url().'admin/transaksi_hapus/'.$t-&gt;transaksi_id; ?&gt;"&gt;&lt;span
class="glyphicon glyphicon-remove"&gt;&lt;/span&gt; Batalkan Transaksi&lt;/a&gt;
    </pre

```

```
<?php  
}  
?>  
</td>
```

Menginput Data Transaksi Rental Mobil

Sekarang saya telah menginput data mobil dan beberapa data kostumer. Sebagai percobaan kita membuat proses transaksi. Silahkan teman-teman input beberapa data mobil dan data kostumer. Untuk data percobaan kita membuat proses transaksi rental mobil.

Mobil Baru						
<input type="button" value="10"/> records per page <input type="text" value="Search:"/>						
No	Merk Mobil	Plat	Warna	Tahun Pembuatan	Status	
1	Toyota Avanza	B 7777 xx	Hitam Metalik	2018	Tersedia	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
2	Honda Brio	B 1111 AA	Putih Mutiara	2016	Tersedia	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
3	Toyota Fortuner	BK 2222 XX	Hitam Metalik	2017	Tersedia	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
4	Suzuki Swift	B 2333 B	Putih	2016	Tersedia	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
5	Toyota Innova	B 1222 BB	Hitam	2015	Tersedia	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
6	Toyota Agya	B 6666 B	Putih	2017	Tersedia	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
7	Daihatsu Xenia	B 6666 GG	Hitam	2017	Tersedia	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
8	Honda HR-V	B 123 NN	Hitam Metalik	2017	Tersedia	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
9	Suzuki Ertiga	BG 7777 HG	Putih	2017	Tersedia	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
10	Daihatsu Sigra	B 7767 B	Biru Muda	2017	Tersedia	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>

Showing 1 to 10 of 10 entries

[← Previous](#) [1](#) [Next →](#)

Data Kostumer

No	Nama	Jenis Kelamin	Alamat	HP	No. KTP		
1	Zainal Abidin	L	Jl. gagak putih no 627, medan . Indonesia	082737373787	998798798798		
2	Marlon Suwanggai	L	Jl. supaka no.23 , Bekasi. Indonesia	081234554443	434534343434		
3	Muhammad Zainal	L	jl. merpati putih 2. no.66 , Indonesia	081212121212	232323232330		
4	Ratna Dewi	L	Jl. ampera selatan. no. 67 , jawa barat, indonesia	081112122219	232231131313		
5	Samsul Bahri	L	jl. peletok. jawa timur. indonesia	081212221221	876876876876		
6	Diki Alfarabi Hadi	L	jl. badak utara. no.123, aceh, indonesia	081221222212	876876876876		

Showing 1 to 6 of 6 entries

← Previous **1** Next →

Kita akan melanjutkan proses selanjutnya setelah memiliki beberapa data mobil dan data kostumer.

Selanjutnya kita akan membuat penginputan transaksi rental mobil. Penginputan transaksi akan kita buat pada method *transaksi_add()*. Langsung saja kita buat method *transaksi_add()* di controller **Admin.php**.

Application/controllers/Admin.php

```
function transaksi_add(){
    $w = array('mobil_status'=>'1');
    $data['mobil'] = $this->m_rental->edit_data($w,'mobil')-
>result();
    $data['kostumer'] = $this->m_rental->get_data('kostumer')-
>result();
    $this->load->view('admin/header');
    $this->load->view('admin/transaksi_add',$data);
    $this->load->view('admin/footer');
}
```

Seperti yang teman-teman lihat pada method *transaksi_add* di atas. di sini kita akan membuat form input transaksi nya di view *transaksi_add.php*. dan juga mengirimkan(parsing) 2 data ke view *transaksi_add.php*. yaitu data mobil dan data kostumer.

```
$data['mobil'] = $this->m_rental->edit_data($w,'mobil')->result();
$data['kostumer'] = $this->m_rental->get_data('kostumer')->result();
```

Kemudian buat view nya dengan nama transaksi_add.php. seperti yang telah kita tetapkan pada method transaksi_add.php.

Application/views/admin/transaksi_add.php

```
<div class="page-header">
    <h3>Transaksi Baru</h3>
</div>

<form action="php echo base_url().'admin/transaksi_add_act' ?" method="post">
    <div class="form-group">
        <label>Kostumer</label>
        <select name="kostumer" class="form-control">
            <option value="">-Pilih Kostumer-</option>
            <?php foreach($kostumer as $k){ ?>
            <option value="php echo $k-&gt;kostumer_id; ?&gt;"&gt;&lt;?php
echo $k-&gt;kostumer_nama; ?&gt;&lt;/option&gt;
            &lt;?php } ?&gt;
        &lt;/select&gt;
        &lt;?php echo form_error('kostumer'); ?&gt;
    &lt;/div&gt;

    &lt;div class="form-group"&gt;
        &lt;label&gt;Mobil&lt;/label&gt;
        &lt;select name="mobil" class="form-control"&gt;
            &lt;option value=""&gt;-Pilih Mobil-&lt;/option&gt;
            &lt;?php foreach($mobil as $m){ ?&gt;
            &lt;option value="<?php echo $m-&gt;mobil_id; ?&gt;"&gt;&lt;?php
echo $m-&gt;mobil_merk; ?&gt;&lt;/option&gt;
            &lt;?php } ?&gt;
        &lt;/select&gt;
        &lt;?php echo form_error('mobil'); ?&gt;
    &lt;/div&gt;

    &lt;div class="form-group"&gt;
        &lt;label&gt;Tanggal Pinjam&lt;/label&gt;
        &lt;input type="date" name="tgl_pinjam" class="form-control"&gt;
        &lt;?php echo form_error('tgl_pinjam'); ?&gt;
    &lt;/div&gt;

    &lt;div class="form-group"&gt;
        &lt;label&gt;Tanggal Kembali&lt;/label&gt;
        &lt;input type="date" name="tgl_kembali" class="form-control"&gt;
        &lt;?php echo form_error('tgl_kembali'); ?&gt;
    &lt;/div&gt;

    &lt;div class="form-group"&gt;
        &lt;label&gt;Harga&lt;/label&gt;</pre
```

```

<input type="number" name="harga" class="form-control">
    <?php echo form_error('harga'); ?>
</div>

<div class="form-group">
    <label>Harga Denda / Hari</label>
    <input type="text" name="denda" class="form-control">
        <?php echo form_error('denda'); ?>
</div>
<div class="form-group">
    <input type="submit" value="Simpan" class="btn btn-primary btn-sm">
</div>
</form>

```

Tadi pada method transaksi_add, kita mengirimkan data mobil dan data kostumer ke view transaksi_add.php.

Kemudian pada view transaksi_add.php kita tampilkan data mobil pada form mobil. Dan tampilkan data kostumer pada form kostumer.

```

<div class="form-group">
    <label>Kostumer</label>
    <select name="kostumer" class="form-control">
        <option value="">-Pilih Kostumer-</option>
        <?php foreach($kostumer as $k){ ?>
            <option value="<?php echo $k->kostumer_id; ?>"><?php echo
$ k->kostumer_nama; ?></option>
        <?php } ?>
    </select>
    <?php echo form_error('kostumer'); ?>
</div>

<div class="form-group">
    <label>Mobil</label>
    <select name="mobil" class="form-control">
        <option value="">-Pilih Mobil-</option>
        <?php foreach($mobil as $m){ ?>
            <option value="<?php echo $m->mobil_id; ?>"><?php echo
$m->mobil_merk; ?></option>
        <?php } ?>
    </select>
    <?php echo form_error('mobil'); ?>
</div>

```

Maka hasilnya akan seperti berikut.

localhost/rental_mobil/admin/transaksi_add

Rental Mobil Dashboard Data Mobil Data Kostumer Transaksi Rental Laporan Logout Halo, Diki Alfarabi Hadi

Transaksi Baru

Kostumer
-Pilih Kostumer-

Mobil
-Pilih Mobil-

Tanggal Pinjam
hh/bb/ttt

Tanggal Kembali
hh/bb/ttt

Harga
[empty input]

Harga Denda / Hari
[empty input]

Simpan

Data kostumer tampil pada form kostumer seperti berikut.

localhost/rental_mobil/admin/transaksi_add

Rental Mobil Dashboard Data Mobil Data Kostumer Transaksi Rental Laporan Logout Halo, Diki Alfarabi Hadi

Transaksi Baru

Kostumer
-Pilih Kostumer-
-Pilih Kostumer-
Zainal Abidin
Marlon Suwangan
Muhammad Zainal
Ratna Dewi
Samsul Bahri
Diki Alfarabi Hadi
hh/bb/ttt

Tanggal Kembali
hh/bb/ttt

Harga
[empty input]

Harga Denda / Hari
[empty input]

Simpan

Data mobil tampil pada form mobil.

The screenshot shows a web-based application interface for managing car rentals. At the top, there's a navigation bar with links like 'Rental Mobil', 'Dashboard', 'Data Mobil', 'Data Kostumer', 'Transaksi Rental', 'Laporan', 'Logout', and a user profile 'Halo, Diki Alfarabi Hadi'. Below the navigation, the main content area has a title 'Transaksi Baru' (New Transaction). It contains three dropdown menus: 'Kostumer' (Customer) with the placeholder '-Pilih Kostumer-', 'Mobil' (Car) with a list of vehicle models including Toyota Avanza, Honda Brio, Toyota Fortuner, Suzuki Swift, Toyota Innova, Toyota Agya, Daihatsu Xenia, Honda HR-V, Suzuki Ertiga, and Daihatsu Sigra, and 'Harga Denda / Hari' (Penalty Price / Day) which is currently empty. At the bottom of the form is a blue 'Simpan' (Save) button.

Selanjutnya kita akan membuat aksi dari form input transaksi ini. Aksi dari form ini akan kita buat di method `transaksi_add_act` di controller `Admin.php`. seperti yang sudah kita buat pada form input transaksi ini.

```
<form action="php echo base_url() . 'admin/transaksi_add_act' ?&gt;
method="post"&gt;</pre

```

Buat method baru dengan nama `transaksi_add_act` di controller `admin.php`.

Application/controllers/admin.php

```
function transaksi_add_act(){
    $kostumer = $this->input->post('kostumer');
    $mobil = $this->input->post('mobil');
    $tgl_pinjam = $this->input->post('tgl_pinjam');
    $tgl_kembali = $this->input->post('tgl_kembali');
    $harga = $this->input->post('harga');
    $denda = $this->input->post('denda');

    $this->form_validation-
>set_rules('kostumer','Kostumer','required');
    $this->form_validation->set_rules('mobil','Mobil','required');

    $this->form_validation->set_rules('tgl_pinjam','Tanggal
Pinjam','required');
    $this->form_validation->set_rules('tgl_kembali','Tanggal
Kembali','required');
```

```

$this->form_validation->set_rules('harga','Harga','required');

$this->form_validation->set_rules('denda','Denda','required');

if($this->form_validation->run() != false) {
    $data = array(
        'transaksi_karyawan' => $this->session-
>userdata('id'),
        'transaksi_kostumer' => $kostumer,
        'transaksi_mobil' => $mobil,
        'transaksi_tgl_pinjam' => $tgl_pinjam,
        'transaksi_tgl_kembali' => $tgl_kembali,
        'transaksi_harga' => $harga,
        'transaksi_denda' => $denda,
        'transaksi_tgl' => date('Y-m-d')
    );

    $this->m_rental->insert_data($data,'transaksi');

    // update status mobil yg di pinjam
    $d = array(
        'mobil_status' => '2'
    );

    $w = array(
        'mobil_id' => $mobil
    );

    $this->m_rental->update_data($w,$d,'mobil');

    redirect(base_url().'admin/transaksi');
} else{
    $w = array('mobil_status'=>'1');
    $data['mobil'] = $this->m_rental->edit_data($w,'mobil')-
>result();
    $data['kostumer'] = $this->m_rental-
>get_data('kostumer')->result();
    $this->load->view('admin/header');
    $this->load->view('admin/transaksi_add',$data);
    $this->load->view('admin/footer');
}
}
}

```

Perhatikan pada action/aksi dari form transaksi di atas. pertama kita menangkap data-data yang di kirim dari form ke dalam variabel masing-masing.

```

$kostumer = $this->input->post('kostumer');
$mobil = $this->input->post('mobil');

```

```

$tgl_pinjam = $this->input->post('tgl_pinjam');
$tgl_kembali = $this->input->post('tgl_kembali');
$harga = $this->input->post('harga');
$denda = $this->input->post('denda');

```

Kemudian kita setting form validation nya. Karena kita ingin form kostumer, mobil, tanggal pinjam, tanggal kembali, harga dan denda wajib di isi(*required*).

```

$this->form_validation->set_rules('kostumer','Kostumer','required');
$this->form_validation->set_rules('mobil','Mobil','required');
$this->form_validation->set_rules('tgl_pinjam','Tanggal Pinjam','required');
$this->form_validation->set_rules('tgl_kembali','Tanggal Kembali','required');
$this->form_validation->set_rules('harga','Harga','required');
$this->form_validation->set_rules('denda','Denda','required');

```

Dan jika validasi sesuai, maka kita input datanya ke table transaksi. Dengan menyesuaikan data yang di input sebagai berikut.

```

$data = array(
    'transaksi_karyawan' => $this->session->userdata('id'),
    'transaksi_kostumer' => $kostumer,
    'transaksi_mobil' => $mobil,
    'transaksi_tgl_pinjam' => $tgl_pinjam,
    'transaksi_tgl_kembali' => $tgl_kembali,
    'transaksi_harga' => $harga,
    'transaksi_denda' => $denda,
    'transaksi_tgl' => date('Y-m-d')
);

$this->m_rental->insert_data($data, 'transaksi');

```

Tabel berikut akan menjelaskan data apa saja yang kita input di table transaksi.

Transaksi_karyawan	Kita menginput data id dari session karyawan yang sedang login atau karyawan yang melayani transaksi rental.
Transaksi_kostumer	Kita menginput id kostumer yang melakukan penyewaan atau rental mobil.
Transaksi_mobil	Kita meginput id mobil yang ingin di sewa oleh kostumer.
Transaksi_tgl_pinjam	Kita menginput tanggal mobil mulai di rental.
Transaksi_tgl_kembali	Kita menginput tanggal mobil selesai di rental.atau tanggal mobil harus dikembalikan.
Transaksi_harga	Kita menginput harga sewa mobil tersebut.

Transaksi_denda	Kita menginput jumlah harga denda per hari jika mobil telat di kembalikan.
Transaksi_tgl	Kita menginput tanggal transaksi ini di lakukan atau tanggal hari ini.

Setelah menginput data transaksi, kita juga akan mengubah status ketersediaan mobil menjadi "2". Seperti yang kita rancang sebelumnya. Jika status nya 1, berarti mobil tersedia. dan mobil yang berstatus 2 berarti sedang di sewa atau sedang di rental.

Jadi di sini kita mengubah status mobil yang ingin di sewa menjadi '2'.

```
// update status mobil yg di pinjam
$d = array(
    'mobil_status' => '2'
);

$w = array(
    'mobil_id' => $mobil
);

$this->m_rental->update_data($w,$d,'mobil');
redirect(base_url().'admin/transaksi');
```

Sekarang yuk kita coba jalankan pada browser dan lihat bagaimana hasilnya.

Rental Mobil [Dashboard](#) [Data Mobil](#) [Data Kostumer](#) [Transaksi Rental](#) [Laporan](#) [Logout](#) Halo, Diki Alfarabi Hadi ▾

Transaksi Baru

Kostumer

Zainal Abidin

Mobil

Suzuki Swift

Tanggal Pinjam

29/10/2017

Tanggal Kembali

31/10/2017

Harga

500000

Harga Denda / Hari

100000

[Simpan](#)

Di sini saya menginput transaksi baru, dengan mengisi kostumer yang ingin melakukan rental mobil adalah zainal abidin, mobil yang ingin di sewa adalah suzuki swift, tanggal peminjamannya di mulai tanggal 29/10/2017 sampai dengan tanggal 31/10/2017 (selama 3 hari). Dengan biaya atau harga sebesar 500.000, dan denda perhari jika terlambat sebesar 100.000. dan klik simpan.

Rental Mobil [Dashboard](#) [Data Mobil](#) [Data Kostumer](#) [Transaksi Rental](#) [Laporan](#) [Logout](#) Halo, Diki Alfarabi Hadi ▾

Data Transaksi

[+ Transaksi Baru](#)

10 records per page

Search:

No	Tanggal	Kostumer	Mobil	Tgl. Pinjam	Tgl. Kembali	Harga	Denda / Hari	Tgl. Dikembalikan	Total Denda	Status	
1	29/10/2017	Zainal Abidin	Suzuki Swift	29/10/2017	31/10/2017	Rp. 500,000	Rp. 100,000	-	Rp. 0,-	-	✓ Transaksi Selesai ✗ Batalkan Transaksi

Showing 1 to 1 of 1 entries

[← Previous](#) [1](#) [Next →](#)

Sampai di sini kita telah berhasil menginput data transaksi ke table transaksi. Dan pada table di atas terdapat 2 buah tombol. Yaitu tombol batalkan transaksi, dan

tombol transaksi selesai untuk menyelesaikan status transaksi. Yang ingin kita buat di sini adalah jika kostumer telah selesai menyewa mobil, admin/karywan bisa klik pada transaksi selesai. Dan akan di alihkan ke sebuah halaman baru untuk mengkonfirmasi bahwa mobil telah di kembalikan.

Membuat Batalkan Transaksi

Pada table transaksi di atas, kita telah membuat tombol/link yang mengarah ke method *transaksi_hapus*.

```
<a class="btn btn-sm btn-danger" href="php echo<br/base_url().'admin/transaksi_hapus/'.$t->transaksi_id; ?>"><span  
class="glyphicon glyphicon-remove"></span> Batalkan Transaksi</a>
```

Pada method *transaksi_hapus* ini kita akan membuat aksi untuk menghapus transaksi. Buat method baru lagi dengan nama *transaksi_hapus()* pada controller Admin.php.

Application/controllers/Admin.php

```
function transaksi_hapus($id) {  
    $w = array(  
        'transaksi_id' => $id  
    );  
    $data = $this->m_rental->edit_data($w, 'transaksi')->row();  
  
    $ww = array(  
        'mobil_id' => $data->transaksi_mobil  
    );  
    $data2 = array(  
        'mobil_status' => '1'  
    );  
    $this->m_rental->update_data($ww, $data2, 'mobil');  
  
    $this->m_rental->delete_data($w, 'transaksi');  
    redirect(base_url().'admin/transaksi');
```

Pada method *transaksi_hapus()*, kita membuat aksi untuk menghapus mengubah kembali status mobil ke "1". Dan menghapus data transaksi dari table transaksi.

Membuat Transaksi Selesai (Konfirmasi)

Pada saat tombol “transaksi_selesai” di klik, maka halaman akan di alihkan ke method *transaksi_selesai()*.

Pada method *transaksi_selesai()* ini, kita akan menampilkan sebuah form baru, yaitu form untuk konfirmasi transaksi selesai.

Buat method baru dalam controllers Admin.php. yaitu method *transaksi_selesai()*.

```
function transaksi_selesai($id){  
    $data['mobil'] = $this->m_rental->get_data('mobil')->result();  
    $data['kostumer'] = $this->m_rental->get_data('kostumer')-  
>result();  
    $data['transaksi'] = $this->db->query("select * from  
transaksi,mobil,kostumer where transaksi_mobil=mobil_id and  
transaksi_kostumer=kostumer_id and transaksi_id='$id'")->result();  
  
    $this->load->view('admin/header');  
    $this->load->view('admin/transaksi_selesai',$data);  
    $this->load->view('admin/footer');  
}
```

Di sini kita memarsing data mobil, data kostumer ke view *transaksi_selesai.php*. kita juga memarsing data transaksi yang ingin di proses. Dan query untuk mendapatkan data transaksi yang ingin di proses/di konfirmasi adalah seperti berikut.

```
select * from transaksi,mobil,kostumer where  
transaksi_mobil=mobil_id and transaksi_kostumer=kostumer_id and  
transaksi_id='$id'
```

Ketiga data yang disebutkan di atas tadi kita parsing ke view *transaksi_selesai.php*.

Buat view baru dalam folder admin dengan nama *transaksi_selesai.php*.

Application/views/admin/transaksi_selesai.php

```
<div class="page-header">  
    <h3>Transaksi Selesai</h3>  
</div>  
  
<?php foreach($transaksi as $t){ ?>  
<form action="<?php echo base_url().'admin/transaksi_selesai_act'  
?>" method="post">  
    <input type="hidden" name="id" value="<?php echo $t-
```

```

>transaksi_id ?>">
    <input type="hidden" name="mobil" value="<?php echo $t-
>transaksi_mobil ?>">
    <input type="hidden" name="tgl_kembali" value="<?php echo $t-
>transaksi_tgl_kembali ?>">
    <input type="hidden" name="denda" value="<?php echo $t-
>transaksi_denda ?>">

    <div class="form-group">
        <label>Kostumer</label>
        <select class="form-control" name="kostumer" disabled>
            <option value="">-Pilih Kostumer-</option>
            <?php foreach($kostumer as $k){ ?>
                <option <?php if($t->transaksi_kostumer == $k-
>kostumer_id){echo "selected='selected'"; } ?> value="<?php echo $k-
>kostumer_id; ?>"><?php echo $k->kostumer_nama; ?></option>
            <?php } ?>
        </select>
    </div>

    <div class="form-group">
        <label>Mobil</label>
        <select class="form-control" name="mobil" disabled>
            <option value="">-Pilih Mobil-</option>
            <?php foreach($mobil as $m){ ?>
                <option <?php if($t->transaksi_mobil == $m-
>mobil_id){echo "selected='selected'"; } ?> value="<?php echo $m-
>mobil_id; ?>"><?php echo $m->mobil_merk; ?></option>
            <?php } ?>
        </select>
    </div>

    <div class="form-group">
        <label>Tanggal Pinjam</label>
        <input class="form-control" type="date" name="tgl_pinjam"
value="<?php echo $t->transaksi_tgl_pinjam ?>" disabled>
    </div>

    <div class="form-group">
        <label>Tanggal Kembali</label>
        <input class="form-control" type="date"
name="tgl_kembali" value="<?php echo $t->transaksi_tgl_kembali ?>" disabled>
    </div>

    <div class="form-group">
        <label>Harga</label>
        <input class="form-control" type="number" name="harga"
value="<?php echo $t->transaksi_harga ?>" disabled>
    </div>

    <div class="form-group">
        <label>Harga Denda / Hari</label>

```

```

        <input class="form-control" type="text" name="denda"
value=<?php echo $t->transaksi_denda ?>" disabled>
    </div>

    <div class="form-group">
        <label>Tanggal Dikembalikan Oleh Kostumer</label>
        <input class="form-control" type="date"
name="tgl_dikembalikan">
        <?php echo form_error('tgl_dikembalikan'); ?>
    </div>

    <div class="form-group">
        <input type="submit" value="Simpan" class="btn btn-
primary btn-sm">
    </div>
</form>
<?php } ?>
```

Maka hasilnya seperti berikut.

Transaksi Selesai

Kostumer
Zainal Abidin

Mobil
Suzuki Swift

Tanggal Pinjam
29/10/2017

Tanggal Kembali
31/10/2017

Harga
500000

Harga Denda / Hari
100000

Tanggal Dikembalikan Oleh Kostumer
hh/bb/ttt

The screenshot shows a web-based application for managing car rentals. The URL in the browser is `localhost/rental_mobil/admin/transaksi_selesai/1`. The page title is "Transaksi Selesai". There are several input fields and dropdown menus:

- Kostumer:** Zainal Abidin
- Mobil:** Suzuki Swift
- Tanggal Pinjam:** 29/10/2017
- Tanggal Kembali:** 31/10/2017
- PILIH TANGGAL MOBIL DI KEMBALIKAN OLEH PENYEWA/KOSTUMER:** Harga
- Harga:** 500000
- Harga Denda / Hari:** 100000
- Tanggal Dikembalikan Oleh Kostumer:** hh/bb/tttt (highlighted with a blue border)

A blue arrow points from the text "Jadi di sini data yang kita dapatkan dari query" to the "Tanggal Dikembalikan Oleh Kostumer" field.

Jadi di sini data yang kita dapatkan dari query

```
$data['transaksi'] = $this->db->query("select * from transaksi,mobil,kostumer where transaksi_mobil=mobil_id and transaksi_kostumer=kostumer_id and transaksi_id='$id'")->result();
```

Kita tampilkan pada form nya masing-masing. Saya yakin teman-teman bisa memahaminya karena pada dasarnya sama saja seperti kita membuat edit data pada proses CRUD. Bedanya di sini pada semua form kita berikan atribut “disabled='disabled'” kecuali pada form “tanggal di kembalikan oleh kostumer”.

Disinilah letak logika dari aplikasi rental mobil ini. Pada saat tanggal mobil di kembalikan oleh kostumer, “tanggal dikembalikan oleh kostumer” akan kita cek dengan data “tanggal kembali”. Apakah pengembalian mobil terlambat atau tidak. Jika terlambat, berapa hari keterlambatannya. Maka harga denda akan kita kalikan dengan berapa hari keterlambatan mobil di kembalikan. Jadi kita bisa mendapatkan berapa jumlah denda yang harus di bayar oleh kostumer pada saat pengembalian mobil.

Dan jika tanggal pengembalian mobil oleh kostumer sesuai dengan tanggal kembali yang sudah di tentukan pada saat transaksi, maka denda tidak ada. Tentu mudah bukan.

Mari kita simak penjelasan selanjutnya untuk membuat aksi dari form transaksi selesai (konfirmasi transaksi) ini.

Seperti yang sudah kita tetapkan pada form transaksi selesai di atas, kita menetapkan untuk aksi dari form ini akan di proses pada method *transaksi_selesai_act()* pada controller **Admin.php**.

Buat method baru dengan nama *transaksi_selesai_act()* dalam controller *Admin.php*.

Application/controllers/Admin.php

```
function transaksi_selesai_act(){
    $id = $this->input->post('id');
    $tgl_dikembalikan = $this->input->post('tgl_dikembalikan');
    $tgl_kembali = $this->input->post('tgl_kembali');
    $mobil = $this->input->post('mobil');
    $denda = $this->input->post('denda');

    $this->form_validation->set_rules('tgl_dikembalikan','Tanggal Di Kembalikan','required');

    if($this->form_validation->run() != false){
        // menghitung selisih hari
        $batas_kembali = strtotime($tgl_kembali);
        $dikembalikan = strtotime($tgl_dikembalikan);
        $selisih = abs((($batas_kembali -
$dikembalikan)/(60*60*24));
        $total_denda = $denda*$selisih;

        // update status transaksi
        $data = array(
            'transaksi_tgldikembalikan' => $tgl_dikembalikan,
            'transaksi_status' => '1',
            'transaksi_totaldenda' => $total_denda
        );
        $w = array(
            'transaksi_id' => $id
        );

        $this->m_rental->update_data($w,$data,'transaksi');

        // update status mobil
        $data2 = array(
            'mobil_status' => '1'
        );
        $w2 = array(
            'mobil_id' => $mobil
        );
    }
}
```

```

        $this->m_rental->update_data($w2,$data2,'mobil');

        redirect(base_url().'admin/transaksi');
    }else{
        $data['mobil'] = $this->m_rental->get_data('mobil')-
>result();
        $data['kostumer'] = $this->m_rental-
>get_data('kostumer')->result();
        $data['transaksi'] = $this->db->query("select * from
transaksi,mobil,kostumer where transaksi_mobil=mobil_id and
transaksi_kostumer=kostumer_id and transaksi_id='$id'")->result();

        $this->load->view('admin/header');
        $this->load->view('admin/transaksi_selesai',$data);
        $this->load->view('admin/footer');
    }
}
// AKHIR TRANSAKSI RENTAL

```

Perhatikan, pertama setelah kita menangkap data dan membuat validasinya.

```

$id = $this->input->post('id');
$tgl_dikembalikan = $this->input->post('tgl_dikembalikan');
$tgl_kembali = $this->input->post('tgl_kembali');
$mobil = $this->input->post('mobil');
$denda = $this->input->post('denda');

$this->form_validation->set_rules('tgl_dikembalikan','Tanggal Di
Kembalikan','required');

```

Selenanjutnya kita mengubah format tanggal kembali dan tanggal mobil di kembalikan. Tanggal kembali kita masukkan ke variabel \$batas_kembali. Dan tanggal mobil di kembalikan kita masukkan ke variabel \$dikembalikan. Selanjutnya kita menghitung selisih hari dari batas pengembalian mobil dan tanggal mobil di kembalikan oleh kostumer. Selanjutnya kita menggunakan fungsi abs() berguna untuk membulatkan bilangan. Baru terakhir kita kalikan jumlah denda perhari dengan selisih hari pengembalian mobil. Hasil dari perkalian ini adalah total denda yang harus di bayar oleh kostumer (total denda = denda perhari X berapa hari terlambat). Dan selanjutnya kita update datanya yang ada di table “transaksi”.

```

// menghitung selisih hari
$batas_kembali = strtotime($tgl_kembali);
$dikembalikan = strtotime($tgl_dikembalikan);
$selisih = abs(($batas_kembali - $dikembalikan) / (60*60*24));

$total_denda = $denda*$selisih;

```

```

// update status transaksi
$data = array(
    'transaksi_tgldikembalikan' => $tgl_dikembalikan,
    'transaksi_status' => '1',
    'transaksi_totaldenda' => $total_denda
);
$w = array(
    'transaksi_id' => $id
);

$this->m_rental->update_data($w,$data,'transaksi');

```

Terakhir kita juga mengupdate atau mengubah status mobil yang di kembalikan menjadi kembali tersedia (1). Tentu teman-teman masih ingat bukan, kalau kolumn mobil_status yang kita buat pada table mobil untuk membuat status mobil. Yang mana kita telah menentukan jika nilai nya 1 = tersedia dan 2 = sedang dalam rental.

```

// update status mobil
$data2 = array(
    'mobil_status' => '1'
);

$w2 = array(
    'mobil_id' => $mobil
);

$this->m_rental->update_data($w2,$data2,'mobil');
redirect(base_url().'admin/transaksi');

```

Selesai. Yuk kita coba.

localhost/rental_mobil/admin/transaksi_selesai/1

Transaksi Selesai

Kostumer
Zainal Abidin

Mobil
Suzuki Swift

Tanggal Pinjam Tanggal batas pengembalian mobil yang sudah di tetapkan sebelumnya pada saat proses transaksi
29/10/2017

Tanggal Kembali 31/10/2017

Harga 500000

Harga Denda / Hari Tanggal mobil di kembalikan oleh kostumer/penyewa
100000

Tanggal Dikembalikan Oleh Kostumer 02/11/2017

Simpan

Pada transaksi ini, sebelumnya batas pengembalian mobil adalah tanggal 31/10/2017. Dan di sini pada saat proses transaksi selesai kita masukkan tanggal pengembalian mobil oleh kostumer nya adalah tanggal 02/11/2017. Jadi otomatis kan si penyewa/kostumer sudah telat 2 hari pengembalian mobil nya dari batas yang sudah di tentukan. Karena denda perharinya adalah 100.000, maka total denda nya adalah 200.000, karena 100.000×2 .

Dan coba langsung kita klik simpan.

localhost/rental_mobil/admin/transaksi

No	Tanggal	Kostumer	Mobil	Tgl. Pinjam	Tgl. Kembali	Harga	Denda / Hari	Tgl. Dikembalikan	Total Denda	Status
1	29/10/2017	Zainal Abidin	Suzuki Swift	29/10/2017	31/10/2017	Rp. 500,000	Rp. 100.000	02/11/2017	Rp. 200.000,-	Selesai

Showing 1 to 1 of 1 entries

Hasilnya sesuai dengan perkiraan kita dan sesuai dengan yang seharusnya. Oh ya perlu teman-teman perhatikan di sini, pada column paling ujung sudah tidak ada lagi tombol “transaksi selesai” dan “batalkan transaksi”. Karena kita hanya menampilkannya pada transaksi yang berstatus 2 (belum selesai). Jadi pada transaksi yang sudah selesai, data tidak bisa lagi di ganggu gugat.

Coba kita input transaksi baru lagi. Dan input transaksi selesai nya sesuai jadwal pengembalian mobil. Apakah hasilnya sesuai atau tidak.

Transaksi Baru

Kostumer
→ Marlon Suwanggai

Mobil
→ Toyota Avanza

Tanggal Pinjam
→ 29/10/2017

Tanggal Kembali
→ 30/10/2017

Harga
→ 100000

Harga Denda / Hari
→ 200000

Simpan

Untuk percobaan kedua ini. Kostumer “marlon suwanggai” yang ingin menyewa mobil. Dia ingin menyewa atau merental mobil “toyota avanza”. Mulai di sewa pada tanggal “29/10/2017” dan harus di kembalikan pada tanggal “30/10/2017”. Dan harganya sebesar Rp.100.000,-. Dengan harga denda perhari sebesar Rp.200.000,-.

Dan klik simpan.

localhost/rental_mobil/admin/transaksi

Rental Mobil Dashboard Data Mobil Data Kostumer Transaksi Rental Laporan Logout Halo, Diki Alfarabi Hadi

Data Transaksi

+ Transaksi Baru

10 records per page Search:

No	Tanggal	Kostumer	Mobil	Tgl. Pinjam	Tgl. Kembali	Harga	Denda / Hari	Tgl. Dikembalikan	Total Denda	Status
1	29/10/2017	Zainal Abidin	Suzuki Swift	29/10/2017	31/10/2017	Rp. 500,000	Rp. 100,000	02/11/2017	Rp. 200,000	Selesai -
2	30/10/2017	Marlon Suwangan	Toyota Avanza	29/10/2017	30/10/2017	Rp. 100,000	Rp. 200,000	-	Rp. 0,-	✓ Transaksi Selesai ✗ Batalkan Transaksi

Showing 1 to 2 of 2 entries

← Previous 1 Next →

Data berhasil di simpan. Dan klik transaksi baru. Karena pura-pura hari ini sudah tanggal 30/10/2017. Dan si kostumer sudah datang untuk mengembalikan mobilnya.

localhost/rental_mobil/admin/transaksi_selesai/2

Transaksi Selesai

Kostumer: Marlon Suwangan

Mobil: Toyota Avanza

Tanggal Pinjam: 29/10/2017

Tanggal Kembali: 30/10/2017

Harga: 100000

Harga Denda / Hari: 200000

Tanggal Dikembalikan Oleh Kostumer: 30/10/2017

Simpan

Dan klik simpan.

Nd	Tanggal	Kostumer	Mobil	Tgl. Pinjam	Tgl. Kembali	Harga	Denda / Hari	Tgl. Dikembalikan	Total Denda	Status
1	29/10/2017	Zainal Abidin	Suzuki Swift	29/10/2017	31/10/2017	Rp. 500,000	Rp. 100,000	02/11/2017	Rp. 200,000,-	Selesai -
2	30/10/2017	Marlon Suwanganai	Toyota Avanza	29/10/2017	30/10/2017	Rp. 100,000	Rp. 200,000	30/10/2017	Rp. 0,-	Selesai -

Showing 1 to 2 of 2 entries

Karena si “marlon suwanganai” mengembalikan mobil nya tepat pada tanggal 30/10/2017 sesuai dengan batas pengembalian, maka tidak ada harga denda.

Selesai sudah aplikasi rental mobil yang kita buat. Tapi terakhir untuk menambah kelengkapannya, kita akan membuat sebuah fitur laporan. Untuk memfilter dan mencetak laporan rental mobil.

Membuat Filter dan Cetak Laporan Rental Mobil

Yang terakhir yang akan kita buat adalah fitur laporan. Dimana nanti kita bisa memfilter data berdasarkan tanggal. Membuat cetak laporan ke pdf dan membuat fitur untuk print laporan transaksi rental mobil.

Coba teman-teeman perhatikan pada menu laporan. Kita telah membuat link nya menuju ke method *laporan()* di controller *Admin.php*.

```
<li><a href=<?php echo base_url().'admin/laporan'; ?>"><span class="glyphicon glyphicon-list-alt"></span> Laporan</a></li>
```

Oleh karena itu kita akan membuat method baru lagi di controller Admin.php. yaitu method/function laporan().

Application/controllers/Admin.php

```
// LAPORAN
function laporan() {
    $dari = $this->input->post('dari');
    $sampai = $this->input->post('sampai');
    $this->form_validation->set_rules('dari', 'Dari Tanggal', 'required');
    $this->form_validation->set_rules('sampai', 'Sampai Tanggal', 'required');

    if($this->form_validation->run() != false) {

        $data['laporan'] = $this->db->query("select * from transaksi,mobil,kostumer where transaksi_mobil=mobil_id and transaksi_kostumer=kostumer_id and date(transaksi_tgl) >= '$dari'" )->result();
        $this->load->view('admin/header');
        $this->load->view('admin/laporan_filter',$data);
        $this->load->view('admin/footer');
    } else{
        $this->load->view('admin/header');
        $this->load->view('admin/laporan');
        $this->load->view('admin/footer');
    }
}
```

Pada method ini kita membuat kondisi. Jika terdapat data tanggal “dari” dan tanggal “sampai”, maka view yang kita tampilkan adalah view laporan_filter.php.

Tapi jika tidak ada data tanggal “dari” dan “sampai”, maka view yang kita tampilkan adalah view laporan.php.

Pada view laporan.php ini kita hanya akan membuat form input penginputan tanggal. Untuk di filter data laporan yang mau di tampilkan dari tanggal berapa dan sampai tanggal berapa.

Sedangkan pada view laporan_filter.php, kita menampilkan data yang kita ambil dari table transaksi dengan detail nya data transaksi dari tanggal yang di input, sampai dengan tanggal yang di input. Fitur ini sering di sebut dengan menampilkan data dengan filter tanggal.

Sekarang kita akan membuat 2 buah view baru dalam folder admin. Dengan nama *laporan.php* dan *laporan_filter.php*.

Application/views/admin/laporan.php

```
<div class="page-header">
    <h3>Laporan</h3>
</div>

<form method="post" action="php echo base_url() . 'admin/laporan' ?&gt;"&gt;
    &lt;div class="form-group"&gt;
        &lt;label&gt;Dari Tanggal&lt;/label&gt;
        &lt;input type="date" name="dari" class="form-control"&gt;
        &lt;?php echo form_error('dari'); ?&gt;
    &lt;/div&gt;
    &lt;div class="form-group"&gt;
        &lt;label&gt;Sampai Tanggal&lt;/label&gt;
        &lt;input type="date" name="sampai" class="form-control"&gt;
        &lt;?php echo form_error('sampai'); ?&gt;
    &lt;/div&gt;
    &lt;div class="form-group"&gt;
        &lt;input type="submit" value="CARI" name="cari" class="btn btn-sm btn-primary"&gt;
    &lt;/div&gt;
&lt;/form&gt;</pre
```

Application/views/admin/laporan_filter.php

```
<div class="page-header">
    <h3>Laporan</h3>
</div>

<form method="post" action="php echo base_url() . 'admin/laporan'"&gt;</pre
```

```

?>">
    <div class="form-group">
        <label>Dari Tanggal</label>
        <input type="date" name="dari" value="=set_value('dari');?" class="form-control">
            <?php echo form_error('dari'); ?>
    </div>
    <div class="form-group">
        <label>Sampai Tanggal</label>
        <input type="date" name="sampai" value="=set_value('sampai');?" class="form-control">
            <?php echo form_error('sampai'); ?>
    </div>
    <div class="form-group">
        <input type="submit" value="CARI" name="cari" class="btn btn-sm btn-primary">
    </div>
</form>

<div class="btn-group">
    <a class="btn btn-warning btn-sm" href="=base_url().'admin/laporan_pdf/?dari=' . set_value('dari') . '&amp;sampai=' . set_value('sampai') ?&gt;"&gt;&lt;span class="glyphicon glyphicon-print"&gt;&lt;/span&gt; Cetak PDF&lt;/a&gt;
    &lt;a class="btn btn-default btn-sm" href="<?=base_url().'admin/laporan_print/?dari=' . set_value('dari') . '&amp;sampai=' . set_value('sampai') ?&gt;"&gt;&lt;span class="glyphicon glyphicon-print"&gt;&lt;/span&gt; Print&lt;/a&gt;
&lt;/div&gt;
&lt;br/&gt;
&lt;br/&gt;
&lt;div class="table-responsive"&gt;
&lt;table border="1" class="table table-striped table-hover table-bordered" id="table-database"&gt;
    &lt;thead&gt;
        &lt;tr&gt;
            &lt;th&gt;No&lt;/th&gt;
            &lt;th&gt;Tanggal&lt;/th&gt;
            &lt;th&gt;Kostumer&lt;/th&gt;
            &lt;th&gt;Mobil&lt;/th&gt;
            &lt;th&gt;Tgl. Pinjam&lt;/th&gt;
            &lt;th&gt;Tgl. Kembali&lt;/th&gt;
            &lt;th&gt;Harga&lt;/th&gt;
            &lt;th&gt;Denda / Hari&lt;/th&gt;
            &lt;th&gt;Tgl. Dikembalikan&lt;/th&gt;
            &lt;th&gt;Total Denda&lt;/th&gt;
            &lt;th&gt;Status&lt;/th&gt;
        &lt;/tr&gt;
    &lt;/thead&gt;
    &lt;tbody&gt;
        &lt;?php
        $no = 1;
        foreach($laporan as $l) {
</pre

```

```

?>
<tr>
    <td><?php echo $no++; ?></td>
    <td><?php echo date('d/m/Y', strtotime($l-
>transaksi_tgl)); ?></td>
        <td><?php echo $l->kostumer_nama; ?></td>
        <td><?php echo $l->mobil_merk; ?></td>
        <td><?php echo date('d/m/Y', strtotime($l-
>transaksi_tgl_pinjam)); ?></td>
        <td><?php echo date('d/m/Y', strtotime($l-
>transaksi_tgl_kembalikan)); ?></td>
        <td><?php echo "Rp. ".number_format($l-
>transaksi_harga); ?></td>
        <td><?php echo "Rp. ".number_format($l-
>transaksi_denda); ?></td>
        <td>
            <?php
                if($l->transaksi_tgldikembalikan == "0000-
00-00") {
                    echo "-";
                } else{
                    echo date('d/m/Y', strtotime($l-
>transaksi_tgldikembalikan));
                }
            ?>
        </td>
        <td><?php echo "Rp. ". number_format($l-
>transaksi_totaldenda).", -"; ?></td>
        <td>
            <?php
                if($l->transaksi_status == "1") {
                    echo "Selesai";
                } else{
                    echo "-";
                }
            ?>
        </td>
    </tr>
    <?php
}
?>
</tbody>
</table>
</div>

```

Sekarang kita coba klik pada menu “laporan”.

The screenshot shows the application's dashboard at localhost/rental_mobil/admin/. The top navigation bar includes links for Rental Mobil, Dashboard, Data Mobil, Data Kostumer, Transaksi Rental, Laporan (which is highlighted with a blue border), Logout, and the user profile "Halo, Diki Alfarabi Hadi". Below the navigation is a section titled "Dashboard" containing four cards:

- Jumlah Mobil:** 10 (View Details)
- Jumlah Kostumer:** 6 (View Details)
- Jumlah Transaksi:** 2 (View Details)
- Rental Selesai:** 2 (View Details)

Below these cards are three main sections:

- Mobil:** A list of available cars (Daihatsu Sigra, Suzuki Ertiga, Honda HR-V, Daihatsu Xenia, Toyota Agya, Toyota Innova, Suzuki Swift) each labeled "Tersedia".
- Kostumer Terbaru:** A list of new customers (Diki Alfarabi Hadi, Samsul Bahri, Ratna Dewi, Muhammad Zainal, Marlon Suwangan, Zainal Abidin).
- Peminjaman Terakhir:** A table showing the last two rentals.

Tgl. Transaksi	Tgl. Pinjam	Tgl. Kembali	Total
30/10/2017	29/10/2017	30/10/2017	Rp. 100,000,-
29/10/2017	29/10/2017	31/10/2017	Rp. 500,000,-

[Lihat Semua Transaksi](#)

Maka akan diarahkan ke method laporan() yang menampilkan form untuk filter tanggal.

The screenshot shows the "Laporan" page at localhost/rental_mobil/admin/laporan. The top navigation bar is identical to the dashboard. The page title is "Laporan". It contains two input fields for date ranges and a search button:

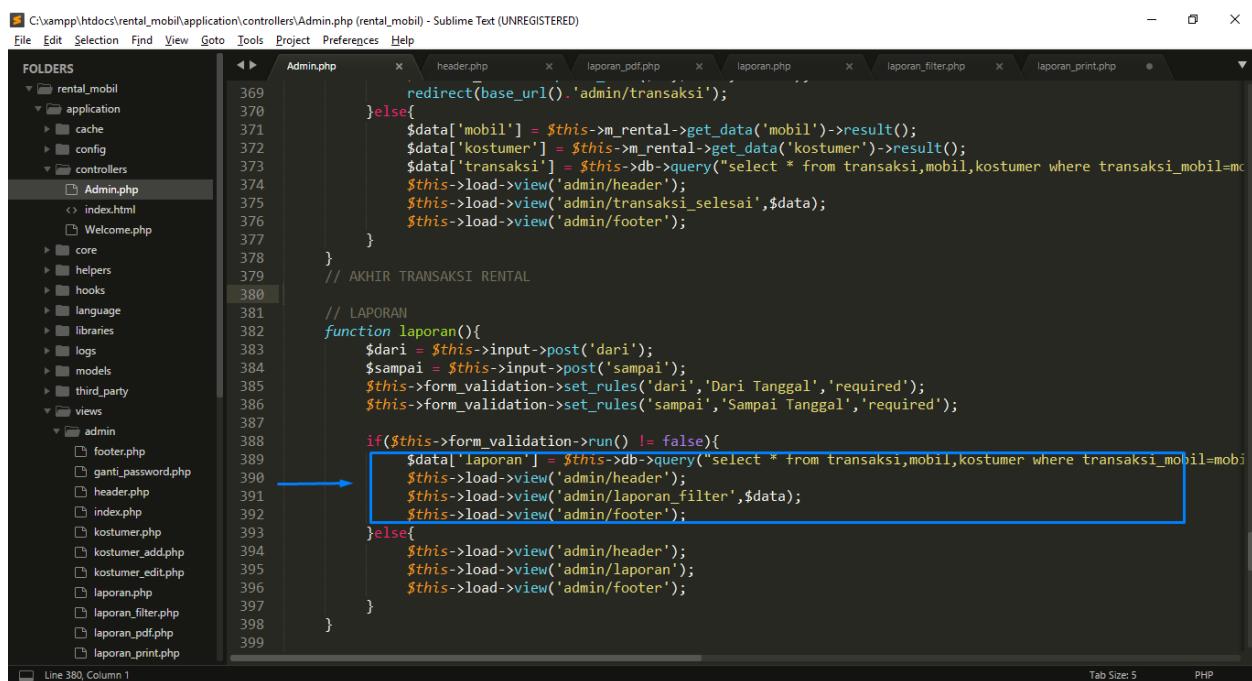
- Dari Tanggal:** hh/bb/tttt
- Sampai Tanggal:** hh/bb/tttt
- CARI**

Menampilkan Data Transaksi Berdasarkan Filter Tanggal

Pada form penginputan filter tanggal di atas, kita telah menetapkan aksi dari form tersebut ke method *laporan()* di controller *Admin.php*.

```
<form method="post" action="php echo base_url() . 'admin/laporan' ?&gt;"&gt;</pre
```

tapi karena ada data tanggal dari dan tanggal sampai yang dikirim, maka view yang ditampilkan adalah view *laporan_filter.php*. bukan lagi view *laporan.php*.



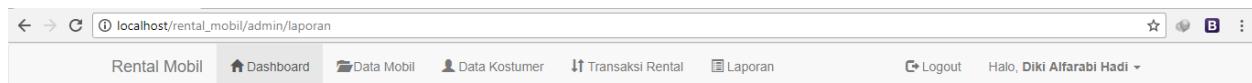
The screenshot shows the Admin.php file in Sublime Text. The code is as follows:

```
369     redirect(base_url() . 'admin/transaksi');
370 }else{
371     $data['mobil'] = $this->m_rental->get_data('mobil')->result();
372     $data['kostumer'] = $this->m_rental->get_data('kostumer')->result();
373     $data['transaksi'] = $this->db->query("select * from transaksi,mobil,kostumer where transaksi_mobil=mobil and kostumer=kostumer");
374     $this->load->view('admin/header');
375     $this->load->view('admin/transaksi_selesai',$data);
376     $this->load->view('admin/footer');
377 }
378 // AKHIR TRANSAKSI RENTAL
379
380 // LAPORAN
381 function laporan(){
382     $dari = $this->input->post('dari');
383     $sampai = $this->input->post('sampai');
384     $this->form_validation->set_rules('dari','Dari Tanggal','required');
385     $this->form_validation->set_rules('sampai','Sampai Tanggal','required');
386
387     if($this->form_validation->run() != false){
388         $data['laporan'] = $this->db->query("select * from transaksi,mobil,kostumer where transaksi_mobil=mobil and kostumer=kostumer");
389         $this->load->view('admin/header');
390         $this->load->view('admin/laporan_filter',$data);
391         $this->load->view('admin/footer');
392     }else{
393         $this->load->view('admin/header');
394         $this->load->view('admin/laporan');
395         $this->load->view('admin/footer');
396     }
397 }
398 }
```

A blue arrow points to the line `$this->load->view('admin/laporan_filter',$data);` in the *laporan()* function.

Dan view *laporan_filter.php* sudah kita buat pada penjelasan sebelumnya.

Jadi langsung saja kita coba jalankan pada browser dengan langsung menentukan data yang ingin ditampilkan dari tanggal berapa sampai tanggal berapa. Disini saya mencoba menampilkan data transaksi dari tanggal 1/10/2017 sampai dengan 31/10/2017.



Laporan

Dari Tanggal

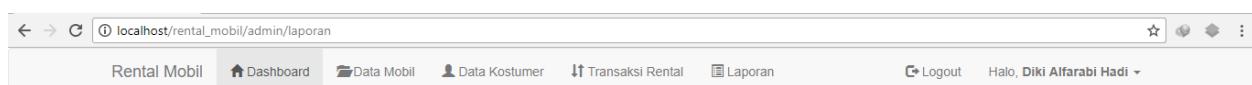
01/10/2017

Sampai Tanggal

31/10/2017

CARI

Dan klik tombol cari. Maka akan di tampilkan semua data transaksi sesuai tanggal yang kita tentukan.



Laporan

Dari Tanggal

01/10/2017

Sampai Tanggal

31/10/2017

CARI

Cetak PDF Print

10 records per page

Search:

No	Tanggal	Kostumer	Mobil	Tgl. Pinjam	Tgl. Kembali	Harga	Denda / Hari	Tgl. Dikembalikan	Total Denda	Status
1	29/10/2017	Zainal Abidin	Suzuki Swift	29/10/2017	31/10/2017	Rp. 500,000	Rp. 100,000	02/11/2017	Rp. 200,000,-	Selesai
2	30/10/2017	Marlon Suwanggai	Toyota Avanza	29/10/2017	30/10/2017	Rp. 100,000	Rp. 200,000	30/10/2017	Rp. 0,-	Selesai

Showing 1 to 2 of 2 entries

← Previous 1 Next →

Dan data transaksi pun berhasil tampil sesuai tanggal yang di filter.

Membuat Print Laporan

Perhatikan pada tombol print yang terdapat di atas tabel data transaksi.

```
<a class="btn btn-default btn-sm" href="#"><?php echo  
base_url().'admin/laporan/print/?dari='.$set_value('dari').'&sampai='
```

```
.set_value('sampai') ?>"><span class="glyphicon glyphicon-print"></span> Print</a>
```

Pada tombol tersebut kita telah membuat url s

```
href=<?php echo  
base_url().'admin/laporan_print/?dari='.$set_value('dari').'&sampai='.  
.set_value('sampai') ?>"
```

Pada tombol print tersebut, kita mengarahkan halaman ke method *laporan_print()* dengan mengirimkan data tanggal dari. Dan data tanggal sampai. *set_value('dari')* berisi data tanggal dari. Dan *set_value('sampai')* berisi data tanggal sampai. *set_value()* sudah di jelaskan pada penggunaan form_validation codeigniter pada "user_guide".

http://localhost/rental_mobil/user_guide/libraries/form_validation.html?highlight=form %20validation

sehingga url yang di hasilkan adalah seperti pada gambar berikut.

The screenshot shows a web application interface for reporting. At the top, there's a navigation bar with links for Rental Mobil, Dashboard, Data Mobil, Data Kostumer, Transaksi Rental, Laporan, Logout, and a user profile. Below the navigation is a search bar with placeholder 'Search:' and a dropdown for 'records per page' set to 10. The main area is titled 'Laporan' and contains two date input fields: 'Dari Tanggal' (01/10/2017) and 'Sampai Tanggal' (31/10/2017). Below these are two buttons: 'CARI' (Search) and 'Cetak PDF' (Print), with the 'Print' button highlighted by a blue box and an arrow. A table follows, showing two entries of rental transactions. The table has columns: No, Tanggal, Kostumer, Mobil, Tgl. Pinjam, Tgl. Kembali, Harga, Denda / Hari, Tgl. Dikembalikan, Total Denda, and Status. The first entry is for Zainal Abidin with a Suzuki Swift, and the second is for Marlton Suwanganai with a Toyota Avanza. At the bottom, a pagination bar shows 'Showing 1 to 2 of 2 entries' with 'Previous' and 'Next' buttons, and the current page is '1'. The status column for both entries is 'Selesai'. The bottom status bar of the browser window shows the URL: localhost/rental_mobil/admin/laporan_print/?dari=2017-10-01&sampai=2017-10-31.

Perhatikan pada gambar di atas. terlihat url tujuan dari link/tombol print saat cursor di letakkan pada tombol print. Seperti yang terlihat pada gambar. Halaman akan di arahkan ke method *laporan_print()*. Dengan mengirim data melalui url (GET).

Selanjutnya buat method baru dengan nama *transaksi_print()* pada controller *Admin.php*.

Application/controllers/Admin.php

```
function laporan_print(){
    $dari = $this->input->get('dari');
    $sampai = $this->input->get('sampai');

    if($dari != "" && $sampai != "") {
        $data['laporan'] = $this->db->query("select * from
transaksi,mobil,kostumer where transaksi_mobil=mobil_id and
transaksi_kostumer=kostumer_id and date(transaksi_tgl) >= '$dari'"-
>result();
        $this->load->view('admin/laporan_print', $data);

    }else{
        redirect("admin/laporan");
    }
}
```

Seperti yang terlihat pada method *laporan_print()* di atas, query berikut adalah query mysql untuk mendapatkan data transaksi sesuai dengan tanggal dari dan tanggal sampai. sama seperti cara menampilkan data transaksi yang sudah di bahas sebelumnya.

```
$data['laporan'] = $this->db->query("select * from
transaksi,mobil,kostumer where transaksi_mobil=mobil_id and
transaksi_kostumer=kostumer_id and date(transaksi_tgl) >= '$dari'"-
>result();
```

Hasil dari query di atas kita tampilkan pada view *laporan_print.php* dalam folder admin. Buat view baru dengan nama *laporan_print.php*.

Application/views/admin/laporan_print.php

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
<body>
    <style type="text/css">
```

```

.table-data{
    width: 100%;
    border-collapse: collapse;
}

.table-data tr th,
.table-data tr td{
    border:1px solid black;
    font-size: 10pt;
}
</style>

<h3>Laporan Transaksi Rental Mobil</h3>






```

```

        <td><?php echo $no++; ?></td>
        <td><?php echo date('d/m/Y', strtotime($l-
>transaksi_tgl)); ?></td>
        <td><?php echo $l->kostumer_nama; ?></td>
        <td><?php echo $l->mobil_merk; ?></td>
        <td><?php echo date('d/m/Y', strtotime($l-
>transaksi_tgl_pinjam)); ?></td>
        <td><?php echo date('d/m/Y', strtotime($l-
>transaksi_tgl_kembali)); ?></td>
        <td><?php echo "Rp. ".number_format($l-
>transaksi_harga); ?></td>
        <td><?php echo "Rp. ".number_format($l-
>transaksi_denda); ?></td>
        <td>
            <?php
            if($l->transaksi_tgldikembalikan == "0000-
00-00") {
                echo "-";
            } else{
                echo date('d/m/Y', strtotime($l-
>transaksi_tgldikembalikan));
            }
            ?>
        </td>
        <td><?php echo "Rp. ". number_format($l-
>transaksi_totaldenda). " , -"; ?></td>
        <td>
            <?php
            if($l->transaksi_status == "1") {
                echo "Selesai";
            } else{
                echo "-";
            }
            ?>
        </td>
    </tr>
    <?php
}
?>
</tbody>
</table>

<script type="text/javascript">
    window.print();
</script>

</body>
</html>

```

Cara menampilkan datanya sama saja seperti cara menampilkan data dari database yang sudah di jelaskan sebelumnya. Bedanya di sini kita menambahkan syntax **javascript** untuk membuat print data.

```
<script type="text/javascript">
    window.print();
</script>
```

Sekarang kita akan mencoba klik tombol print. Dan lihat hasilnya.

Dari Tanggal
01/10/2017

Sampai Tanggal
31/10/2017

CARI

Cetak PDF Print

10 records per page

Showing 1 to 2 of 2 entries

No	Tanggal	Kostumer	Mobil	Tgl. Pinjam	Tgl. Kembali	Harga	Denda / Hari	Tgl. Dikembalikan	Total Denda	Status
1	29/10/2017	Zainal Abidin	Suzuki Swift	29/10/2017	31/10/2017	Rp. 500,000	Rp. 100,000	02/11/2017	Rp. 200,000,-	Selesai
2	30/10/2017	Marlon Suwangan	Toyota Avanza	29/10/2017	30/10/2017	Rp. 100,000	Rp. 200,000	30/10/2017	Rp. 0,-	Selesai

localhost/rental_mobil/admin/laporan_print/?dari=2017-10-01&sampai=2017-10-31

Maka akan muncul halaman print seperti berikut.

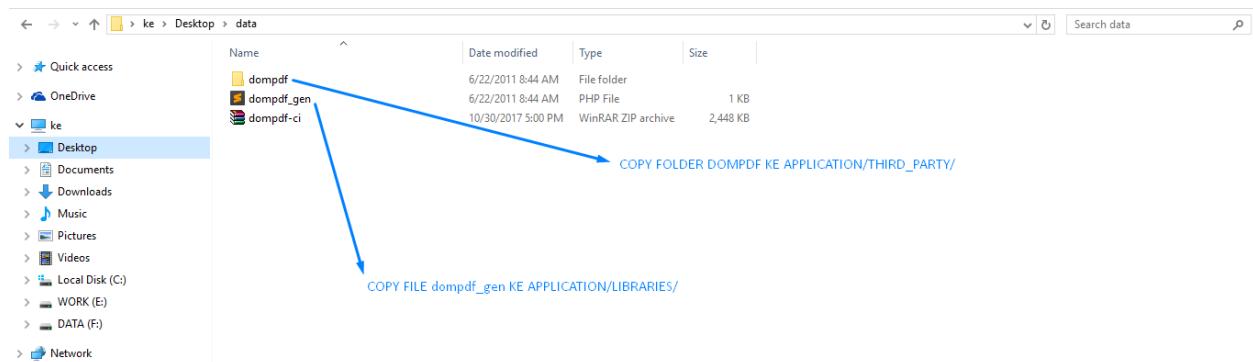
No	Tanggal	Kostumer	Mobil	Tgl. Rental	Tgl. Kembali	Harga	Denda / Hari	Tgl. Dikembalikan	Total Denda	Status
1	29/10/2017	Zainal Abidin	Suzuki Swift	29/10/2017	31/10/2017	Rp. 500,000	Rp. 100,000	02/11/2017	Rp. 200,000	Selesai
2	30/10/2017	Marlon Suwanggai	Toyota Avanza	29/10/2017	30/10/2017	Rp. 100,000	Rp. 200,000	30/10/2017	Rp. 0,-	Selesai

Membuat cetak laporan selesai. Dan selanjutnya kita akan membuat cetak laporan ke file PDF.

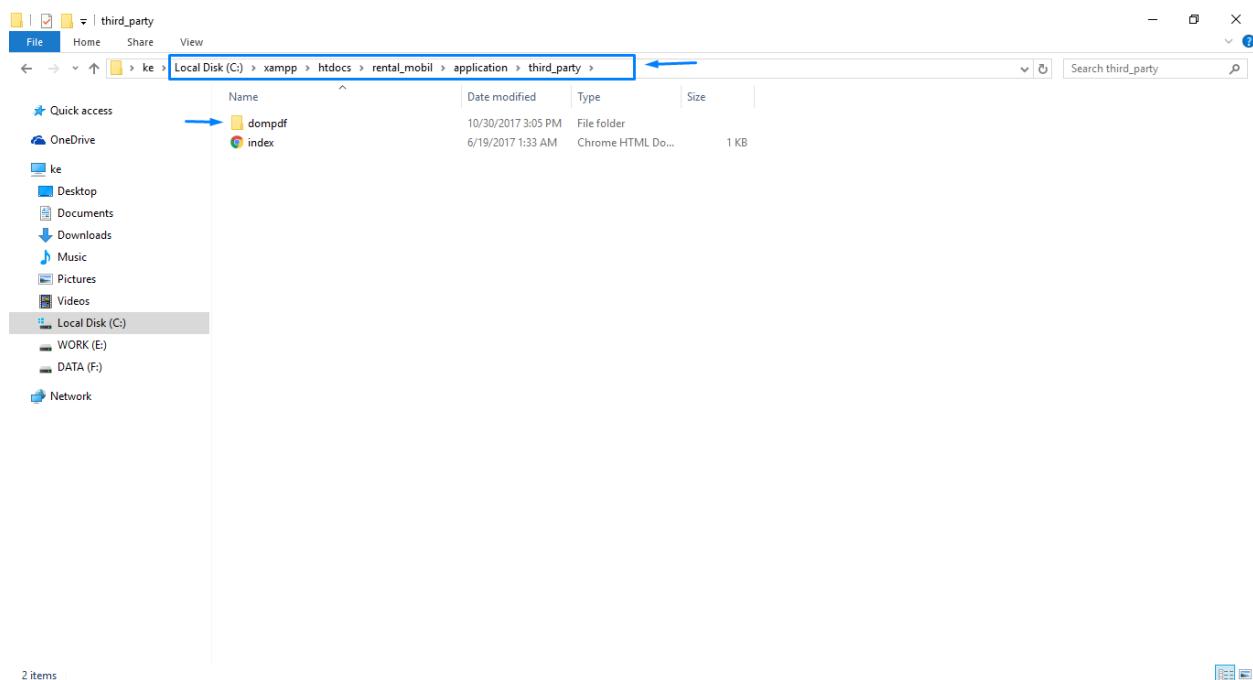
Membuat Cetak Laporan Ke PDF Menggunakan DOMPDF

Untuk membuat cetak laporan PDF pada PHP, kita bisa menggunakan library DOMPDF. DOMPDF salah satu library PHP yang bisa di gunakan untuk membuat cetak laporan PDF dengan sangat mudah.

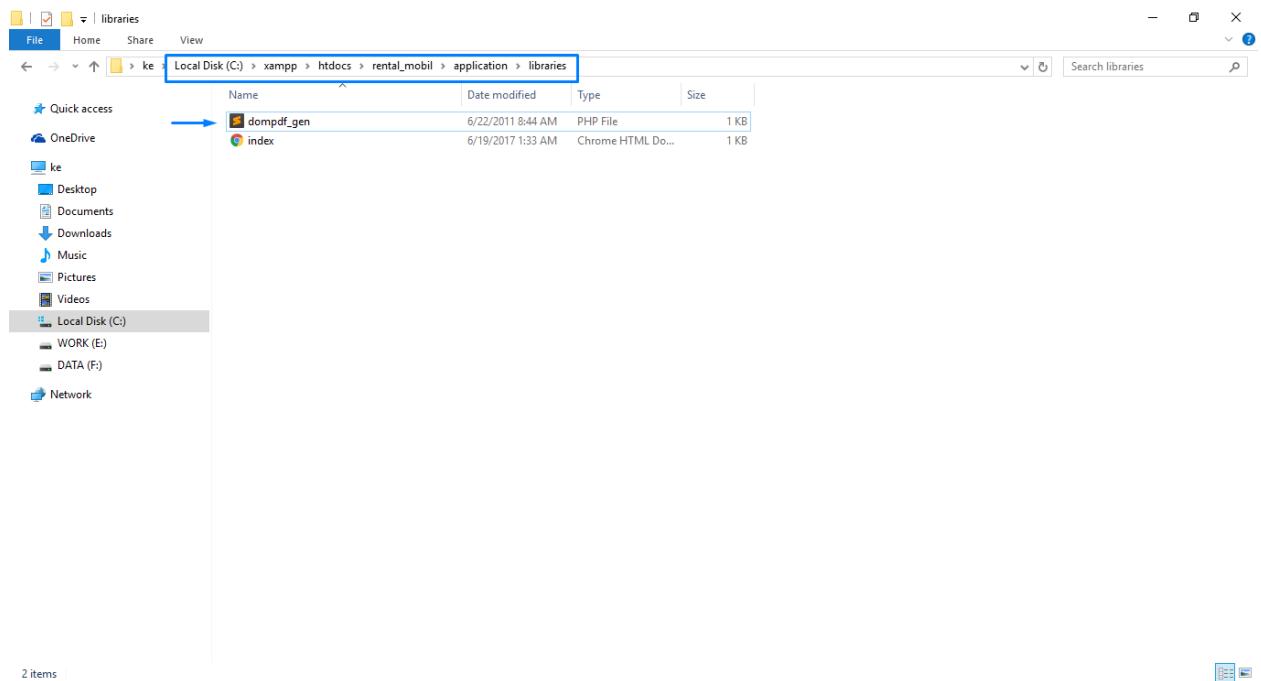
Untuk library DOMPDF nya sendiri sudah saya sertakan 1 paket dalam buku ini. Silahkan teman-teman ekstrak file dompdf-ci.zip. kemudian akan terdapat sebuah folder dompdf, dan satu buat file dompdf_gen.php.



Selanjutnya copy folder dompdf ke application/third_party/.



Dan copy file dompdf_gen.php ke application/libraries/.



Sampai di sini instalasi dompdf telah selesai.

Jika pada tombol print kita membuat link menuju method *laporan_print()*. Di tombol cetak kita membuat link menuju method *laporan_pdf()*.

```
<a class="btn btn-warning btn-sm" href="#"><?php echo  
base_url().'admin/laporan_pdf/?dari=' .set_value('dari') . '&sampai=' .  
set_value('sampai') ?>"><span class="glyphicon glyphicon-  
print"></span> Cetak PDF</a>
```

Jadi sekarang buat method baru lagi pada controller Admin.php dengan nama *laporan_pdf()*.

Application/controllers/Admin.php

```
function laporan_pdf() {
    $this->load->library('dompdf_gen');
    $dari = $this->input->get('dari');
    $sampai = $this->input->get('sampai');

    $data['laporan'] = $this->db->query("select * from
transaksi,mobil,kostumer where transaksi_mobil=mobil_id and
transaksi_kostumer=kostumer_id and date(transaksi_tgl) >= '$dari'"-
>result();
    $this->load->view('admin/laporan_pdf', $data);

    $paper_size = 'A4'; // ukuran kertas
    $orientation = 'landscape'; //tipe format kertas potrait atau
landscape
    $html = $this->output->get_output();

    $this->dompdf->set_paper($paper_size, $orientation);
    //Convert to PDF
    $this->dompdf->load_html($html);
    $this->dompdf->render();
    $this->dompdf->stream("laporan.pdf", array('Attachment'=>0));
    // nama file pdf yang di hasilkan
}
```

Seperti biasa, kita menangkap data tanggal dari, dan tanggal sampai. kemudian kita ambil data transaksi rental dari table **transaksi** dengan memfilter data yang di ambil yaitu dari tanggal yang sudah di tentukan, dan sampai tanggal yang sudah di tentukan.

Pertama, kita panggil dulu library “dompdf_gen” yang sudah kita copy tadi untuk mengaktifkannya.

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

Dan kemudian data yang kita dapat dari query tersebut kita tampilkan pada view *laporan_pdf.php*.

```
$dari = $this->input->get('dari');
$ Sampai = $this->input->get('sampai');

$data['laporan'] = $this->db->query("select * from
transaksi,mobil,kostumer where transaksi_mobil=mobil_id and
transaksi_kostumer=kostumer_id and date(transaksi_tgl) >= '$dari'"-
>result();
$this->load->view('admin/laporan_pdf', $data);
```

Dan untuk mengkonversinya ke file pdf,

```
$paper_size = 'A4'; // ukuran kertas
$orientation = 'landscape'; //tipe format kertas potrait atau
landscape
$html = $this->output->get_output();

$this->dompdf->set_paper($paper_size, $orientation);
//Convert to PDF
$this->dompdf->load_html($html);
$this->dompdf->render();
$this->dompdf->stream("laporan.pdf", array('Attachment'=>0)); // 
nama file pdf yang di hasilkan
```

Teman-teman bisa mengatur ukuran kertas, format kertas serta nama file pdf yang akan di hasilkan nantinya.

Selanjutnya ..

Buat view baru dengan nama laporan_pdf.php.

Application/views/admin/laporan_pdf.php

```
<!DOCTYPE html>
<html>
```

```

<head>
    <title></title>
</head>
<body>
    <style type="text/css">
        .table-data{
            width: 100%;
            border-collapse: collapse;
        }

        .table-data tr th,
        .table-data tr td{
            border:1px solid black;
            font-size: 10pt;
        }
    </style>

    <h3>Laporan Transaksi Rental Mobil</h3>

    <table>
        <tr>
            <td>Dari Tgl</td>
            <td>:</td>
            <td><?php echo
date('d/m/Y', strtotime($_GET['dari'])); ?></td>
            </tr>
        <tr>
            <td>Sampai Tgl</td>
            <td>:</td>
            <td><?php echo
date('d/m/Y', strtotime($_GET['sampai'])); ?></td>
            </tr>
        </table>

        <br/>
        <table class="table-data">
            <thead>
                <tr>
                    <th>No</th>
                    <th>Tanggal</th>
                    <th>Kostumer</th>
                    <th>Mobil</th>
                    <th>Tgl. <RP></RP>ental</th>
                    <th>Tgl. Kembali</th>
                    <th>Harga</th>
                    <th>Denda / Hari</th>
                    <th>Tgl. Dikembalikan</th>
                    <th>Total Denda</th>
                    <th>Status</th>
                </tr>
            </thead>
            <tbody>

```

```

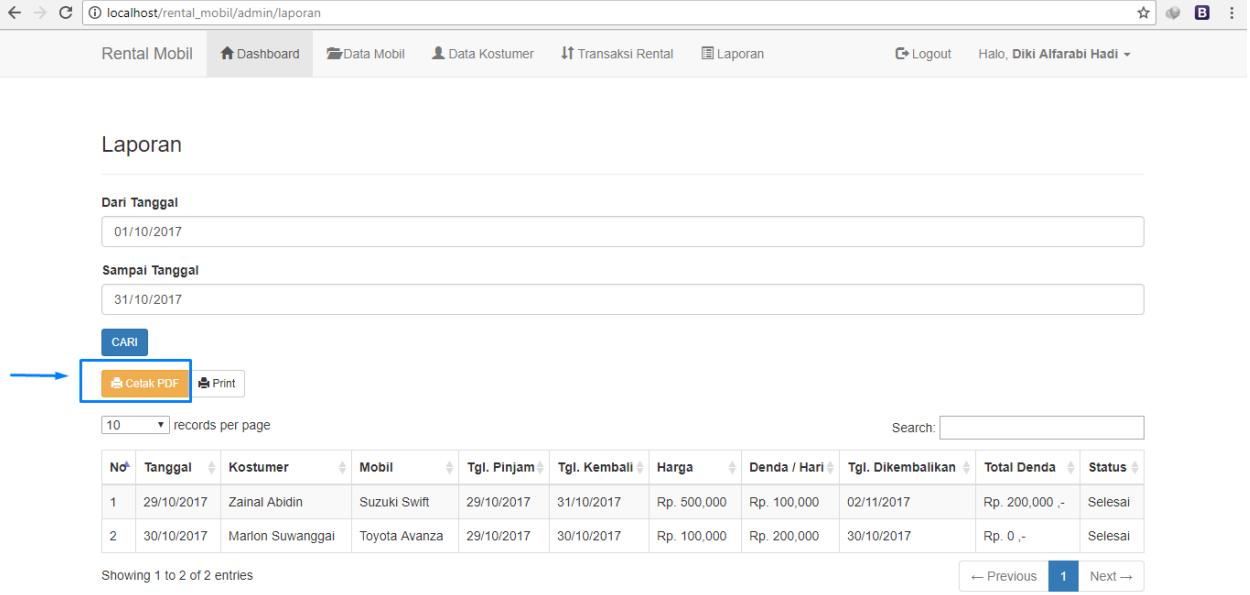
<?php
$no = 1;
foreach($laporan as $l) {
?>
<tr>
    <td><?php echo $no++; ?></td>
    <td><?php echo date('d/m/Y', strtotime($l->transaksi_tgl)); ?></td>
    <td><?php echo $l->kostumer_nama; ?></td>
    <td><?php echo $l->mobil_merk; ?></td>
    <td><?php echo date('d/m/Y', strtotime($l->transaksi_tgl_pinjam)); ?></td>
    <td><?php echo date('d/m/Y', strtotime($l->transaksi_tgl_kembali)); ?></td>
    <td><?php echo "Rp. ".number_format($l->transaksi_harga); ?></td>
    <td><?php echo "Rp. ".number_format($l->transaksi_denda); ?></td>
    <td>
        <?php
        if($l->transaksi_tgldikembalikan == "0000-00-00") {
            echo "-";
        } else{
            echo date('d/m/Y', strtotime($l->transaksi_tgldikembalikan));
        }
        ?>
    </td>
    <td><?php echo "Rp. ". number_format($l->transaksi_totaldenda).",-"; ?></td>
    <td>
        <?php
        if($l->transaksi_status == "1"){
            echo "Selesai";
        }else{
            echo "-";
        }
        ?>
    </td>
</tr>
<?php
}
?>
</tbody>
</table>

</body>
</html>

```

Yuk kita lihat hasil pembuatan cetak laporan PDF berdasarkan tanggalnya.

Klik pada tombol “Cetak PDF”.



Dari Tanggal
01/10/2017

Sampai Tanggal
31/10/2017

CARI

Cetak PDF Print

10 records per page

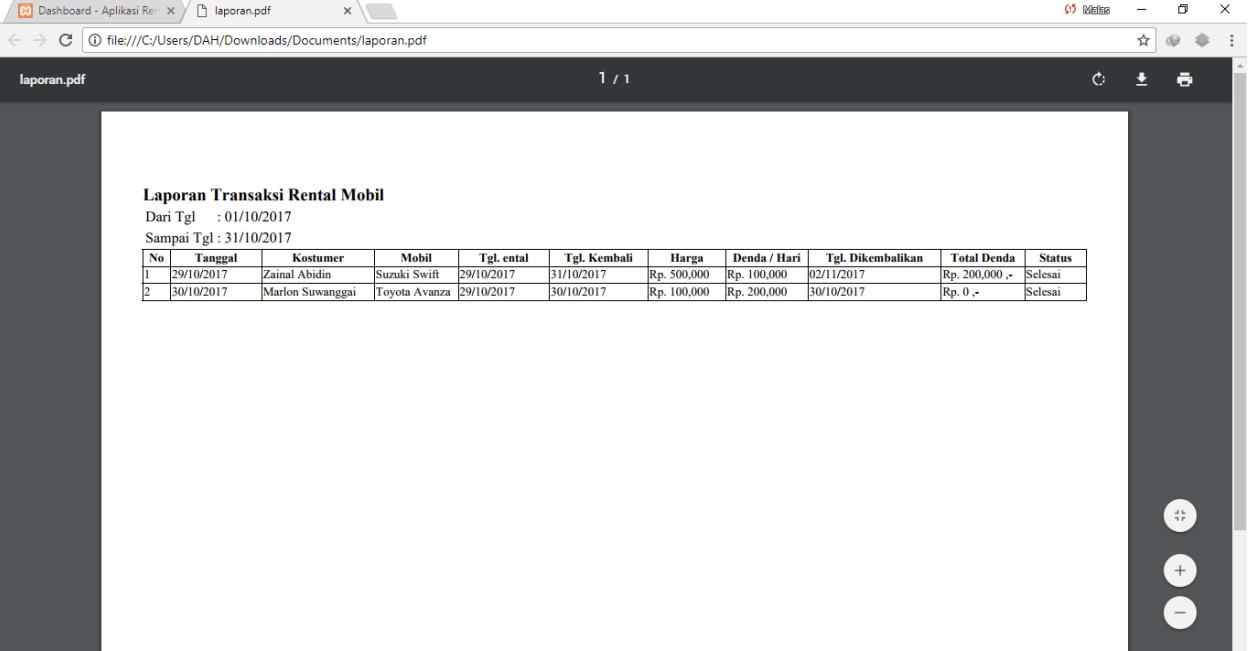
Search:

No	Tanggal	Kostumer	Mobil	Tgl. Pinjam	Tgl. Kembali	Harga	Denda / Hari	Tgl. Dikembalikan	Total Denda	Status
1	29/10/2017	Zainal Abidin	Suzuki Swift	29/10/2017	31/10/2017	Rp. 500,000	Rp. 100,000	02/11/2017	Rp. 200,000,-	Selesai
2	30/10/2017	Marlon Suwanganai	Toyota Avanza	29/10/2017	30/10/2017	Rp. 100,000	Rp. 200,000	30/10/2017	Rp. 0,-	Selesai

Showing 1 to 2 of 2 entries

← Previous 1 Next →

Dan hasilnya.



Laporan Transaksi Rental Mobil

Dari Tgl : 01/10/2017

Sampai Tgl : 31/10/2017

No	Tanggal	Kostumer	Mobil	Tgl. ental	Tgl. Kembali	Harga	Denda / Hari	Tgl. Dikembalikan	Total Denda	Status
1	29/10/2017	Zainal Abidin	Suzuki Swift	29/10/2017	31/10/2017	Rp. 500,000	Rp. 100,000	02/11/2017	Rp. 200,000,-	Selesai
2	30/10/2017	Marlon Suwanganai	Toyota Avanza	29/10/2017	30/10/2017	Rp. 100,000	Rp. 200,000	30/10/2017	Rp. 0,-	Selesai

Cetak laporan ke format PDF pun selesai.

Selesai