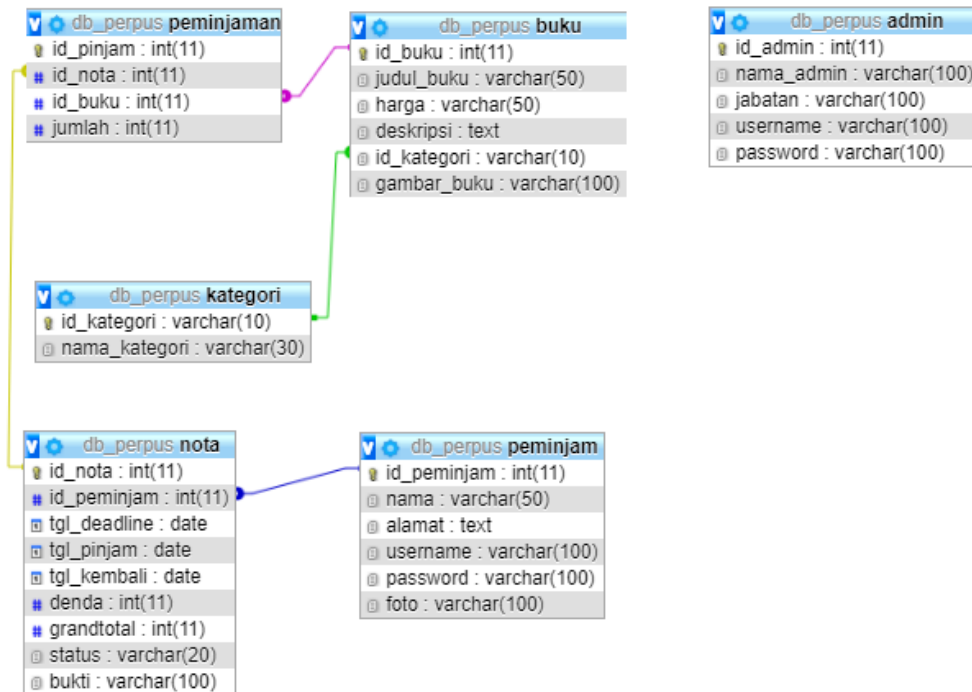


## Aplikasi ini menggunakan template Admin LTE 4.2.2

Bahan-bahan yang disiapkan :

- Xampp
- Framework Codeigniter
- Template admin\_lte 4.2.2
- Nama database db\_perpus, dan desain databasenya seperti dibawah ini:

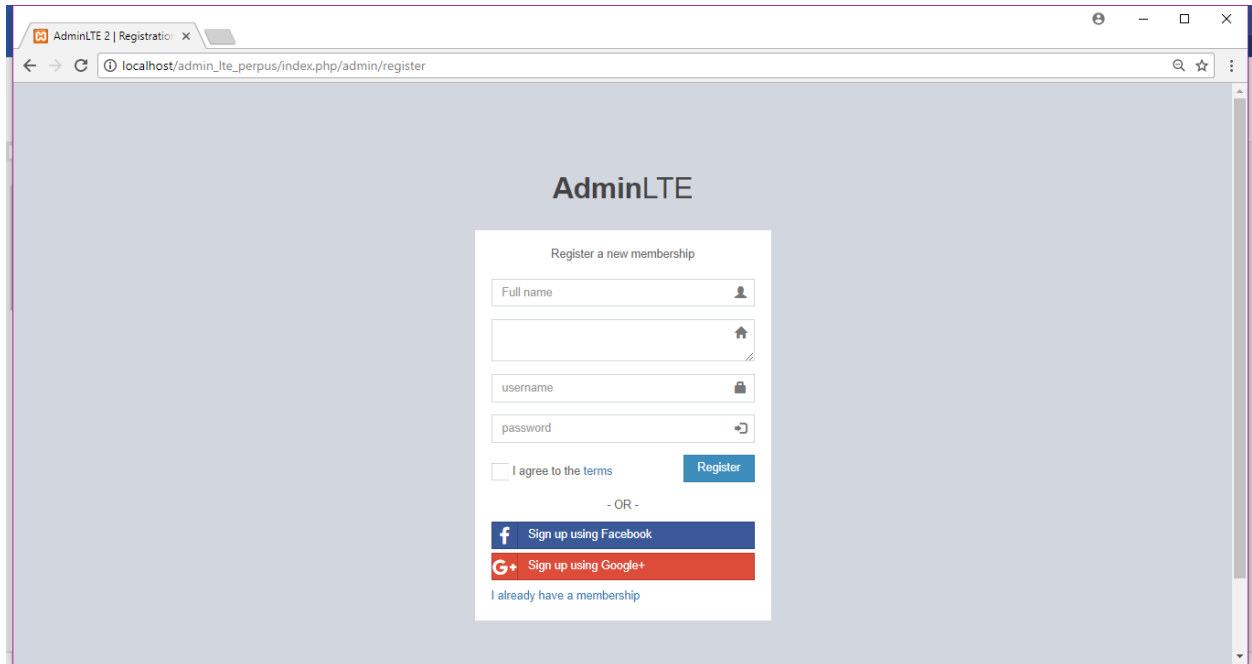


pertama yang harus diatur adalah :

- Pengaturan BASE\_URL di config/config.php  
Cari kata-kata base\_url tambahkan url dasar misal [http://localhost/admin\\_lte\\_perpus](http://localhost/admin_lte_perpus)
- Pengaktifan function BASE\_URL di config/autoload.php  
Cari helper tambahkan 'url'
- Pengaturan koneksi database di config/database.php  
Atur host, user, password dan nama databasenya
- Pengaktifan function database di config/autoload.php  
Cari libraries tambahkan 'database'
- Pengaktifan form\_validation di config/autoload.php  
Cari libraries tambahkan 'form\_validation'  
Gunanya untuk membuat validasi input
- Pengaktifan session di config/autoload.php  
Cari libraries tambahkan 'session'  
Gunanya untuk membuat halaman login dan menampilkan pesan flash\_data
- Pengaktifan cart di config/autoload.php

Cari libraries tambahkan 'cart'  
Gunanya untuk menyimpan sementara item-item yang dipesan

## Membuat Halaman Pendaftaran



### Controller Admin.php

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

### View register.php

```

register.php
34 </div>
35
36 <div class="register-box-body">
37 <p class="login-box-msg">Register a new membership</p>
38
39 <?php
40 if($this->session->flashdata('pesan')!=null){
41 echo "<div class='alert alert-success'>".$this->session->flashdata('pesan')."</div>";}}?>
42
43 <form action="<?=base_url('index.php/admin/simpan')?&gt;" method="post"&gt;
44 &lt;div class="form-group has-feedback"&gt;
45 &lt;input name="nama" type="text" class="form-control" placeholder="Full name"&gt;
46 &lt;span class="glyphicon glyphicon-user form-control-feedback"&gt;&lt;/span&gt;
47 &lt;/div&gt;
48 &lt;div class="form-group has-feedback"&gt;
49 &lt;textarea name="alamat" class="form-control"&gt;&lt;/textarea&gt;
50 &lt;span class="glyphicon glyphicon-home form-control-feedback"&gt;&lt;/span&gt;
51 &lt;/div&gt;
52 &lt;div class="form-group has-feedback"&gt;
53 &lt;input name="username" type="text" class="form-control" placeholder="username"&gt;
54 &lt;span class="glyphicon glyphicon-lock form-control-feedback"&gt;&lt;/span&gt;
55 &lt;/div&gt;
56 &lt;div class="form-group has-feedback"&gt;
57 &lt;input name="password" type="password" class="form-control" placeholder="password"&gt;
58 &lt;span class="glyphicon glyphicon-log-in form-control-feedback"&gt;&lt;/span&gt;
59 &lt;/div&gt;
60 &lt;div class="row"&gt;
61 &lt;div class="col-xs-8"&gt;
62 &lt;div class="checkbox icheck"&gt;
63 &lt;label&gt;
64 &lt;input type="checkbox"&gt; I agree to the &lt;a href="#"&gt;terms&lt;/a&gt;
65 &lt;/label&gt;
66 &lt;/div&gt;
67 &lt;/div&gt;
68 &lt;!-- /.col --&gt;
69 &lt;div class="col-xs-4"&gt;
70 &lt;input type="submit" class="btn btn-primary btn-block btn-flat" value="Register" name="submit"&gt;
71 &lt;/div&gt;
72 &lt;!-- /.col --&gt;
73 &lt;/div&gt;
74 &lt;/form&gt;
</pre

```

Ket.

**Script session->flashdata** berfungsi untuk menampilkan data yang tersimpan sekali proses. Biasanya digunakan untuk menampilkan pesan sukses atau gagal atau yang lainnya sesuai kebutuhan.

Pendeklarasiannya menggunakan function `$this->session->set_flashdata('name','isi pesannya')`

`<form action="<?=base_url('index.php/admin/simpan')?&gt;" method="post"&gt;</code untuk`

sekrup disamping ini yaitu form, jgn lupa tentukan isi action dengan url yang berfungsi untuk memproses data input yang sudah diisi di inputannya.

Setiap inputan harus diberi attribut name yang mana akan dipanggil di halaman action.

**Controller admin.php function simpan** untuk memproses insert data inputan ke db.

```

119     public function simpan()
120     {
121         if($this->input->post('submit')){
122             $this->form_validation->set_rules('nama', 'Nama Lengkap', 'trim|required');
123             $this->form_validation->set_rules('alamat', 'Alamat Lengkap', 'trim|required');
124             $this->form_validation->set_rules('username', 'Username', 'trim|required');
125             $this->form_validation->set_rules('password', 'Password', 'trim|required');
126             if ($this->form_validation->run() == TRUE) {
127                 $this->load->model('m_user');
128
129                 if($this->m_user->masuk()==TRUE){
130                     $this->session->set_flashdata('pesan', 'sukses simpan');
131                     redirect('admin/login','refresh');
132                 } else {
133                     $this->session->set_flashdata('pesan', 'gagal simpan');
134                     redirect('admin/register','refresh');
135                 }
136             } else {
137                 $this->session->set_flashdata('pesan', validation_errors());
138                 redirect('admin/register','refresh');
139             }
140         }
141     }
142 }
143
144 }
145
146 }

```

Ket:

```

$this->form_validation->set_rules('nama', 'Nama Lengkap', 'trim|required');
$this->form_validation->set_rules('alamat', 'Alamat Lengkap', 'trim|required');
$this->form_validation->set_rules('username', 'Username', 'trim|required');
$this->form_validation->set_rules('password', 'Password', 'trim|required');

```

Pada gambar diatas ada form\_validation untuk menfalidasi setiap inputan. Snippet untuk menampilkan validation adalah form\_validation kemudian **tekan control+spasi**.

```

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

if($this->m_user->masuk()==TRUE){

```

Setalah validation hasilnya true maka tindakan kita adalah proses menyimpan data inputan ke database.

**Load model m\_user** pada gambar diatas bertujuan untuk mengeload file model agar bisa memanggil function didalamnya.

**\$this->m\_user->masuk()** adalah proses pemanggilan function **masuk** yang ada di **file model m\_user**.

```

$this->session->set_flashdata('pesan', 'sukses simpan');

```

gambar disamping ini adalah proses pendeklarasian atau proses penyimpanan data ke flashdata sehingga bisa tampilkan data yang disimpan dengan memanggil function ci **\$this->session->flashdata('pesan')** diletakkan di file views.

```
redirect('admin/login','refresh');
```

script disamping ini berfungsi untuk menampilkan atau mengarahkan ke halaman lain. Di script ini diarahkan ke controller admin function login agar halaman login bisa tampil.

```
$this->session->set_flashdata('pesan', validation_errors());
```

script disamping adalah proses penyimpanan data ke flashdata agar bisa tampil di views. Disini yang disimpan adalah validation\_errors yang berfungsi menampilkan apa yang salah atau kurang pada validasinya.

### Model m\_user.php function masuk

```
public function masuk()
{
    $nama=$this->input->post('nama');
    $alamat=$this->input->post('alamat');
    $username=$this->input->post('username');
    $password=$this->input->post('password');
    $data_simpan=array(
        'nama'=>$nama,
        'alamat'=>$alamat,
        'username'=>$username,
        'password'=>$password
    );
    $this->db->insert('peminjam',$data_simpan);
    if($this->db->affected_rows()>0){
        return TRUE;
    } else {
        return FALSE;
    }
}
```

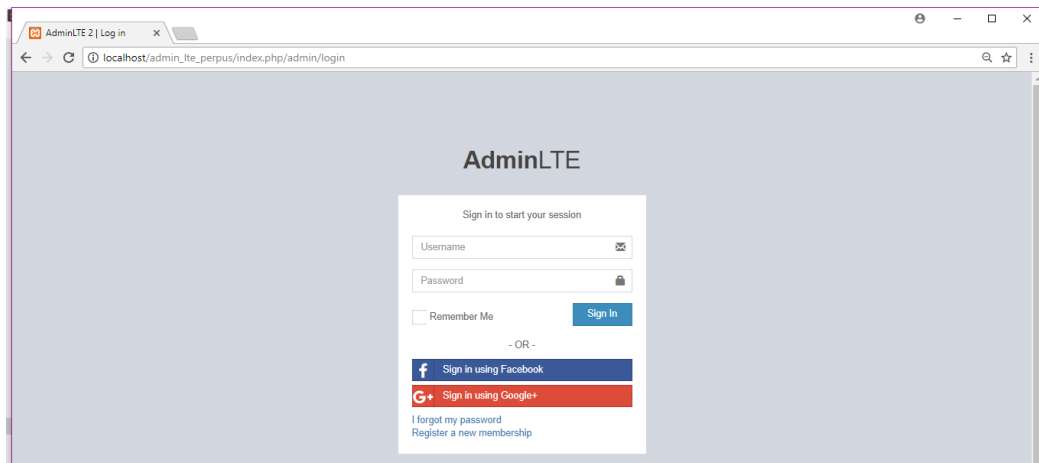
Ket: gambar diatas adalah proses insert data ke database.

```
if($this->db->affected_rows()>0){
```

gambar disamping ini adalah function ci yang berfungsi untuk mengecek apakah data yang dimasukkan menambah atau tidak (menampilkan jumlah baris yang bertambah).

Setiap di model harus diberi nilai return karena agar function tersebut ketika dipanggil bisa menampilkan nilai atau data yang dibawa oleh return.

## Membuat Halaman LOGIN



## Controller admin.php function login

```
public function login()
{
    if($this->session->userdata('login')==TRUE){
        redirect('admin','refresh');
    } else {
        $this->load->view('login');
    }
}
```

Ket:

```
if($this->session->userdata('login')==TRUE){
```

 script disamping ini

bermaksud untuk mengecek apakah session userdata menyimpan nilai true atau tidak. Jika true maka akan langsung diarahkan ke controller admin function index. Jika false akan diarahkan ke menampilkan file views login atau menampilkan halaman login.

## Views login.php

```

<?php
if($this->session->flashdata('pesan')!=null){
echo "<div class='alert alert-success'>".$this->session->flashdata('pesan')."</div>";}

<p class="login-box-msg">Sign in to start your session</p>

<form action="<?=base_url('index.php/admin/proses_login')?>" method="post">
<div class="form-group has-feedback">
<input type="text" name="username" class="form-control" placeholder="Username">
<span class="glyphicon glyphicon-envelope form-control-feedback"></span>
</div>
<div class="form-group has-feedback">
<input type="password" name="password" class="form-control" placeholder="Password">
<span class="glyphicon glyphicon-lock form-control-feedback"></span>
</div>
<div class="row">
<div class="col-xs-8">
<div class="checkbox icheck">
<label>
<input type="checkbox"> Remember Me
</label>
</div>
</div>
<!-- /.col -->
<div class="col-xs-4">
<input name="login" type="submit" class="btn btn-primary btn-block btn-flat" value="Sign In">
</div>
<!-- /.col -->
</div>
</form>

```

Ket: script diatas adalah untuk halaman login. Jangan lupa untuk mengatur atribut action dan method seperti pada script dibawah ini

```

<form action="<?=base_url('index.php/admin/proses_login')?>" method="post">

```

Jangan lupa juga setiap inputan selalu diberi atribut name yang mana akan dipanggil di function proses login.

**Controller admin.php function proses\_login untuk memproses data inputan**

```

public function proses_login()
{
    if($this->input->post('login')){
        $this->form_validation->set_rules('username', 'Username', 'trim|required');
        $this->form_validation->set_rules('password', 'Password', 'trim|required');
        if ($this->form_validation->run() == TRUE) {
            $this->load->model('m_user');
            if($this->m_user->get_login()->num_rows()>0){
                $data=$this->m_user->get_login()->row();
                $array = array(
                    'login' => TRUE,
                    'nama_lengkap'=>$data->nama,
                    'username'=>$data->username,
                    'password'=>$data->password,
                    'id_peminjam'=>$data->id_peminjam,
                    'foto'=>$data->foto
                );

                $this->session->set_userdata( $array );
                redirect('admin','refresh');
            } else {
                $this->session->set_flashdata('pesan', 'salah username dan password');
                redirect('admin/login','refresh');
            }
        } else {
            $this->session->set_flashdata('pesan', validation_errors());
            redirect('admin/login','refresh');
        }
    }
}

```

Ket:

```

$this->load->model('m_user');
if($this->m_user->get_login()->num_rows()>0){

```

script disamping ini adalah

berfungsi memanggil model m\_user function get\_login. **Num\_rows** berfungsi untuk menampilkan hasil yang berupa jumlah baris. Jika num\_rows menampilkan lebih dari 1 maka dinyatakan username dan password benar jika 0 maka username dan password salah atau menjalankan script else. Jika username dan password benar maka langkah berikutnya adalah menampilkan data detail data menggunakan ROW() kemudian disimpan ke session agar datanya akan tersimpan terus hingga di logout. Penulisan script penyimpanan data ke session seperti ini :



```

$data=$this->m_user->get_login()->row();

$array = array(
    'login' => TRUE,
    'nama_lengkap'=>$data->nama,
    'username'=>$data->username,
    'password'=>$data->password,
    'id_peminjam'=>$data->id_peminjam,
    'foto'=>$data->foto
);

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

```

Set\_userdata berfungsi menyimpan data ke session sedangkan cara menampilkannya dengan memanggil function CI \$this->session->userdata('name')

```

$data=$this->m_user->get_login()->row();

```

script disamping hampir sama dengan yang diatas. Cuma perbedaannya **num\_rows** dan **row**. **Row()** ini berfungsi untuk menampilkan hanya 1 data atau 1 record. Beda dengan **result()**, jika **result()** berfungsi untuk menampilkan data lebih dari 1 data. Sehingga untuk menampilkannya harus menggunakan looping atau perulangan.

Detail datanya akan tersimpan ke variable \$data. Cara menampilkan datanya adalah dengan memanggil \$data->nama\_kolom\_table\_db. Contoh \$data->id\_peminjam yang mana id\_peminjam adalah nama kolom dari table peminjam yang ada di database.

## Model M\_user.php function get\_login

```

public function get_login()
{
    return $this->db
        ->where('username',$this->input->post('username'))
        ->where('password',$this->input->post('password'))
        ->get('peminjam');
}

```

Ket:

Script diatas adalah query dengan menggunakan function CI yang mana sama dengan query SQL seperti ini :

```

select * from peminjam where username=$this->input->post('username') and password=$this->input->post('password')

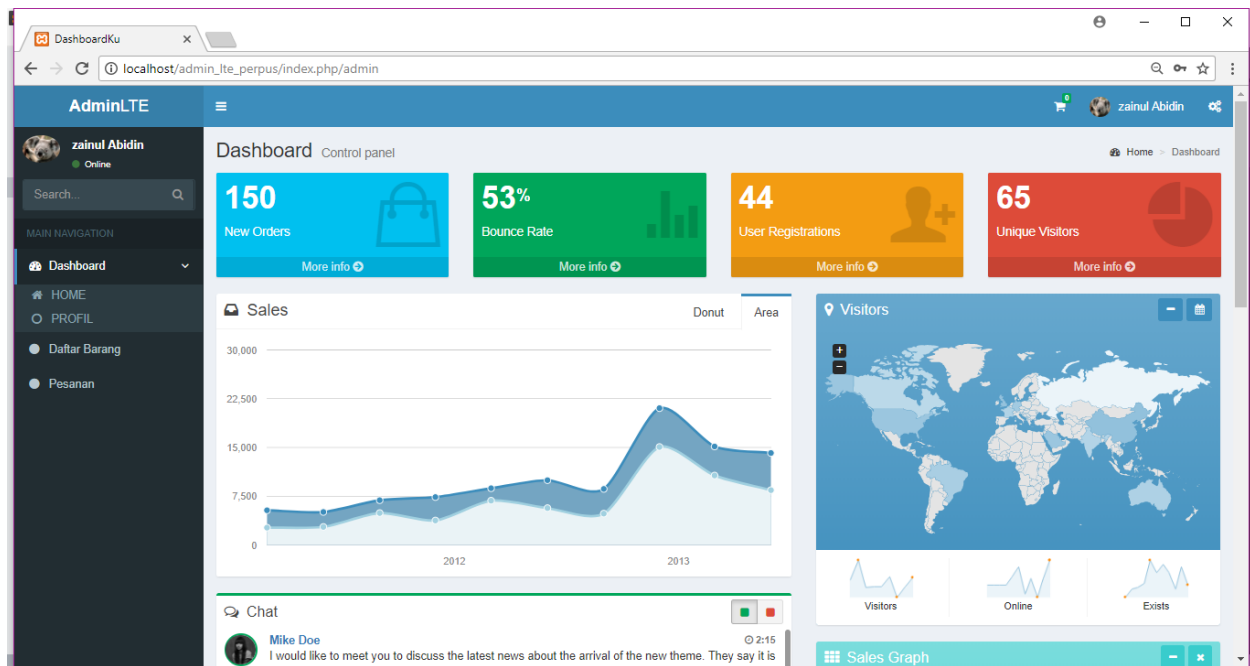
```

```
$this->input->post('username'))
```

ini adalah function CI yang berfungsi untuk menampilkan data yang tersimpan di dalam inputan berdasarkan name inputnya.

Diberi script return agar function ini ketika dipanggil bisa membawa nilai hasil query tersebut.

## Membuat halaman dashboard



### Controller admin.php function index

```
public function index()
{
    $data['konten']="home";
    $data['judul']="DashboardKu";
    $this->load->view('template', $data);
}
```

Ket:

Biasanya jika kita memanggil function index di url , maka nama index tidak perlu dipanggil, cukup nama controllernya. Misal localhost/index.php/admin maka yang akan terpenggil adalah function index.

Seperti contoh dashboard diatas.

A screenshot of a web browser's address bar. It shows the URL 'localhost/admin\_lte\_perpus/index.php/admin' with a search icon on the left and a star icon on the right.

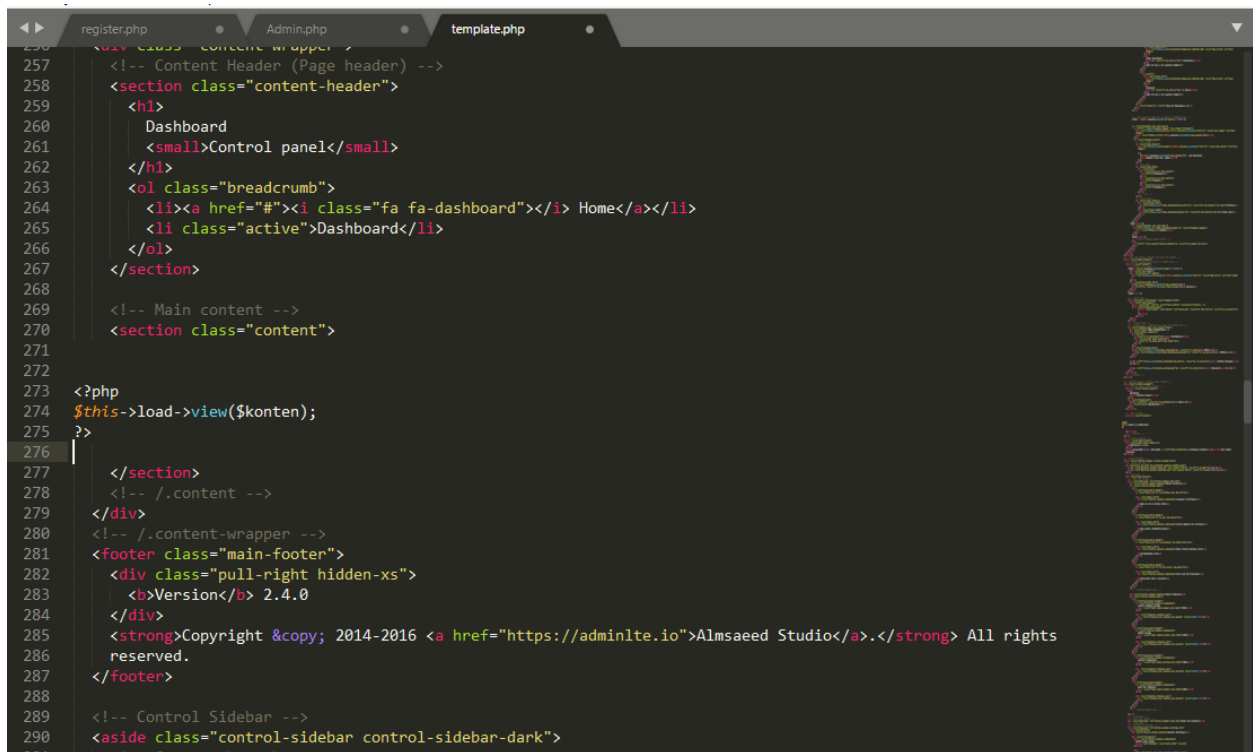
```
$data['konten']="home";
```

script disamping ini maksudnya adalah kita akan mensisipkan file home.php di views template.php agar bisa tampil di body template. Karena di view

template akan kita masukkan text home ke `$this->load->view($konten)` yang mana `$konten` berasal dari index `$data`.

`$data['judul']="DashboardKu";` script disamping ini adalah bertujuan untuk menampilkan text dashboardku di tag title. Agar bisa tampil text tersebut, maka di view kita panggil dengan `$judul` di view template contoh: `<title><?= $judul ?></title>`.

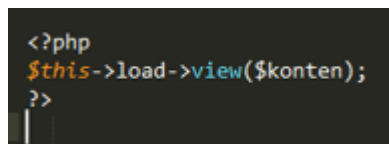
## Views template.php



```
257 <!-- Content Header (Page header) -->
258 <section class="content-header">
259   <h1>
260     Dashboard
261     <small>Control panel</small>
262   </h1>
263   <ol class="breadcrumb">
264     <li><a href="#"><i class="fa fa-dashboard"></i> Home</a></li>
265     <li class="active">Dashboard</li>
266   </ol>
267 </section>
268
269 <!-- Main content -->
270 <section class="content">
271
272
273 <?php
274 $this->load->view($konten);
275 ?>
276
277 </section>
278 <!-- /.content -->
279 </div>
280 <!-- /.content-wrapper -->
281 <footer class="main-footer">
282   <div class="pull-right hidden-xs">
283     <b>Version</b> 2.4.0
284   </div>
285   <strong>Copyright &copy; 2014-2016 <a href="https://adminlte.io">Almsaeed Studio</a>.</strong> All rights
286   reserved.
287 </footer>
288
289 <!-- Control Sidebar -->
290 <aside class="control-sidebar control-sidebar-dark">
```

Ket:

Di template ini kita harus menentukan desain yang tetap, desain yang tidak tetap. Desain yang tetap akan ada di template. Desain yang tidak tetap akan kita letakkan di file lain alias dipindah ke file lain. Dengan ini agar halaman web ini bisa lebih dinamis dan mudah untuk diatur karena desain tetap tidak perlu kita ketik lagi di setiap halaman dan desain yang tidak tetap akan kita sisipkan ke halaman desain tetap sehingga desain tetap dan tidak tetap akan tergabung. Cara menggabungkan atau mensisipkan file tidak tetap dengan cara :



```
<?php
$this->load->view($konten);
?>
```

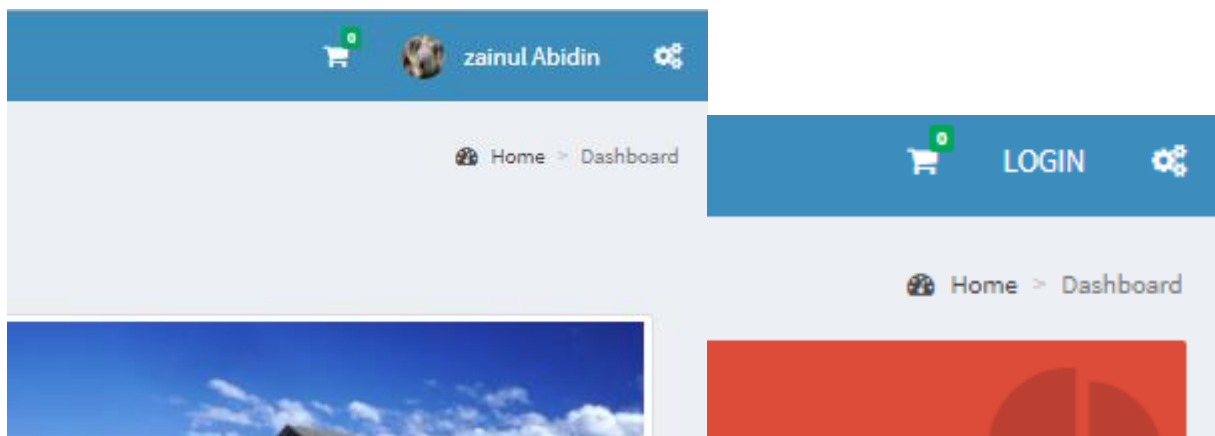
kita ketikkan script disamping ini ke bagian body desain yang tidak tetap atau yang selalu berubah-ubah. `$konten` berasal dari nama index `$data` yang ada di controller.

## Isi dari home.php

```
1 <!-- Small boxes (Stat box) -->
2 <div class="row">
3   <div class="col-lg-3 col-xs-6">
4     <!-- small box -->
5     <div class="small-box bg-aqua">
6       <div class="inner">
7         <h3>150</h3>
8
9         <p>New Orders</p>
10      </div>
11      <div class="icon">
12        <i class="ion ion-bag"></i>
13      </div>
14      <a href="#" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>
15    </div>
16  </div>
17  <!-- ./col -->
18  <div class="col-lg-3 col-xs-6">
19    <!-- small box -->
20    <div class="small-box bg-green">
21      <div class="inner">
22        <h3>53<sup style="font-size: 20px">%</sup></h3>
23
24        <p>Bounce Rate</p>
25      </div>
26      <div class="icon">
27        <i class="ion ion-stats-bars"></i>
28      </div>
29      <a href="#" class="small-box-footer">More info <i class="fa fa-arrow-circle-right"></i></a>
30    </div>
31  </div>
32  <!-- ./col -->
33  <div class="col-lg-3 col-xs-6">
34    <!-- small box -->
35    <div class="small-box bg-yellow">
```

Ket: isi dari home.php ini bebas. Terserah saja. Sesuai kebutuhan.

## Menampilkan tombol login jika belum login dan menampilkan nama jika sudah login di template



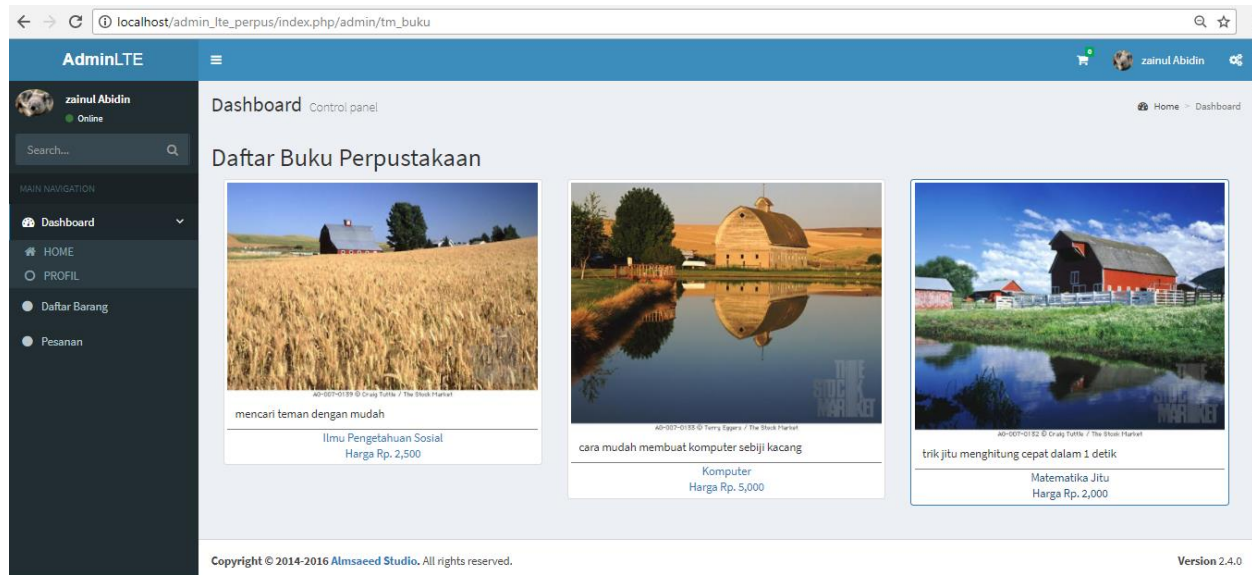
Views template.php

```

ter.php Admin.php template.php home.php x
<?php if ($this->session->userdata('login')==TRUE): ?>
<li class="dropdown user user-menu">
  <a href="#" class="dropdown-toggle" data-toggle="dropdown">
    
    <span class="hidden-xs"><?= $this->session->userdata('nama_lengkap')></span>
  </a>
  <ul class="dropdown-menu">
    <!-- User image -->
    <li class="user-header">
      
      <p>
        <?= $this->session->userdata('nama_lengkap')> - Web Developer
        <small>Member since Nov. 2012</small>
      </p>
    </li>
    <!-- Menu Footer-->
    <li class="user-footer">
      <div class="pull-left">
        <a href="<?=base_url('index.php/admin/data_diri')>" class="btn btn-default btn-flat">Profile</a>
      </div>
      <div class="pull-right">
        <a href="<?=base_url('index.php/admin/logout')>" class="btn btn-default btn-flat">Sign out</a>
      </div>
    </li>
  </ul>
</li>
</li>
<?php else: ?>
<li class="dropdown user user-menu">
  <a href="<?=base_url('index.php/admin/login')>" class="dropdown-toggle">
    <span class="hidden-xs">LOGIN</span>
  </a>
</li><?php endif ?>

```

## MEMBUAT HALAMAN TAMPIL DATA BUKU



### CONTROLLER ADMIN.PHP FUNCTION TM\_BUKU

```

public function tm_buku()
{
    $this->load->model('m_buku');
    $data['tampil_buku']=$this->m_buku->tampil_buku();
    $data['konten']="buku";
    $data['judul']="Daftar Buku";
    $this->load->view('template', $data);
}

```

## MODEL M\_BUKU

```

public function tampil_buku()
{
    $tm_buku=$this->db
        ->join('kategori','kategori.id_kategori=buku.id_kategori')
        ->get('buku')->result();
    return $tm_buku;
}

```

Ket: script pada gambar diatas merupakan query join table dengan table kategori dan buku.

## VIEWS BUKU.PHP

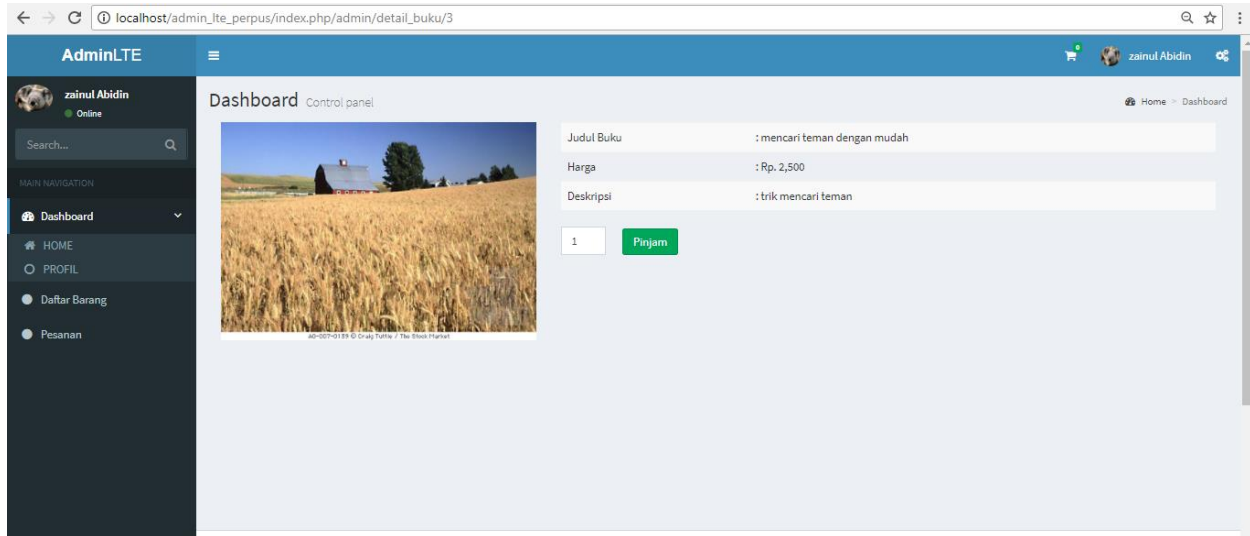
```

1 <h2>Daftar Buku Perpustakaan</h2>
2 <?php
3     foreach ($tampil_buku as $buku) {
4     ?>
5         <div class="col-md-4">
6             <a href="<?=base_url('index.php/admin/detail_buku/'.$buku->id_buku)>?" class="thumbnail">
7                 
8                 <div class="caption" style="border-bottom:1px solid grey">
9                     <?= $buku->judul_buku;>
10                </div>
11                <center><?= $buku->nama_kategori ?><br>
12                Harga Rp. <?= number_format($buku->harga)></center>
13            </a>
14        </div>
15    }
16    <?php
17    }
18    ?>
19    <?php
20    }
21    ?>
22

```

Ket: disini menggunakan perulangan foreach untuk menampilkan hasil query dari model. Karena di modelnya m\_buku.php function tampil\_buku menggunakan result(), menyimpan lebih dari 1 data.

# MEMBUAT DETAIL BUKU



Membuat link dulu. Perhatikan pada script `<a href=...` Seperti dibawah ini:

```
1 <h2>Daftar Buku Perpustakaan</h2>
2 <?php
3     foreach ($tampil_buku as $buku) {
4     ?>
5         <div class="col-md-4">
6             <a href="<?=base_url('index.php/admin/detail_buku/'.$buku->id_buku)?" class="thumbnail">
7                 
8                 <div class="caption" style="border-bottom:1px solid grey">
9                     <?= $buku->judul_buku;?>
10                </div>
11                <center><?= $buku->nama_kategori ?><br>
12                Harga Rp. <?= number_format($buku->harga)?></center>
13            </a>
14        </div>
15    }
16    <?php
17    }
18    ?>
```

Ket: pada tag `<a href="<?=base_url('index.php/admin/detail_buku/'.$buku->id_buku)?"` memanggil controller admin function detail\_buku dan parameter yaitu id\_buku. Agar dihalaman detail\_buku bisa memproses nilai id\_bukunya. Karena di query detail\_buku kita men-select berdasarkan id\_buku maka membutuhkan nilai id\_buku.

## CONTROLLER ADMIN.PHP FUNCTION DETAIL\_BUKU

```

public function detail_buku($id_buku='')
{
    $this->load->model('m_buku','buk');
    $data['detail']=$this->buk->detail($id_buku);
    $data['konten']="detail_buku";
    $data['judul']="Detail Buku";
    if($this->session->userdata('login')==TRUE){
        $data['url']="

```

**KET:** parameter yang dipasang di link dibawah ini di views buku.php

`<a href="<?=-base_url('index.php/admin/detail_buku/' . $buku->id_buku)?"` yaitu \$buku->id\_buku akan ditangkap atau disimpan pada parameter yang ada di detail\_buku

`public function detail_buku($id_buku='')` yaitu variabel \$id\_buku.

```

if($this->session->userdata('login')==TRUE){
    $data['url']="

```

Pada logika if diatas melogikakan jika sudah login maka \$data['url'] menyimpan script tombol pinjam, jika belum login menyimpan tombol login dulu.

```

$this->load->model('m_buku','buk');
$data['detail']=$this->buk->detail($id_buku);

```

Karena kita butuh men-select berdasarkan id\_buku maka kita panggil file model m\_buku lalu panggil function detail dengan membawa parameter id\_buku. Nilai id\_buku akan ditangkap oleh parameter yang ada di function detail di model m\_buku.php

Hasil query nya akan disimpan ke variable \$data['detail'] yang siap dipanggil di views detail\_buku.php

## MODEL M\_BUKU.PHP FUNCTION DETAIL

```

public function detail($a)
{
    $tm_buku=$this->db
        ->join('kategori','kategori.id_kategori=buku.id_kategori')
        ->where('id_buku',$a)
        ->get('buku')
        ->row();
    return $tm_buku;
}

```

**Ket:**

`public function detail($a)` \$a adalah variable yang akan menyimpan atau menangkap nilai parameter yang dikirim dari controller admin function detail\_buku.



Pada Script diatas, diakhir script kita beri row() itu berarti kita akan menampilkan 1 data atau 1 record.

## VIEWS DETAIL\_BUKU.PHP

```
1 <div class="col-md-4">
2     
3 </div>
4 <div class="col-md-8">
5     <table class="table table-hover table-striped">
6         <tr>
7             <td>Judul Buku </td><td>:
8             <?= $detail->judul_buku;?></td>
9         </tr>
10        <tr>
11            <td>Harga </td><td>:
12            Rp. <?= number_format($detail->harga);?>
13            </td>
14        </tr>
15        <tr>
16            <td>Deskripsi </td><td>:
17            <?= $detail->deskripsi;?>
18            </td>
19        </tr>
20    </table><form action="<?=base_url('index.php/cart/add_cart')?>" method="post">
21        <input type="hidden" name="id_buku" value="<?= $detail->id_buku?>">
22        <input type="number" class="form-control" style="width:55px;display: inline-block;margin-right: 20px"
23            " value="1" name="qty"><?=$url?>
24    </form>
25</div>
```

Ket:

Untuk menampilkan data hasil query tidak perlu perulangan karena hanya menampilkan 1 data atau 1 record. `<?= $detail->judul_buku;?>` script disamping berarti \$detail berasal dari file

controller admin function detail\_buku yang memanggil halaman views ini. Yaitu dari script dibawah ini

```
$data['detail']=$this->buk->detail($id_buku);
```

```
</table><form action="<?=base_url('index.php/cart/add_cart')?>" method="post">
<input type="hidden" name="id_buku" value="<?= $detail->id_buku?>">
<input type="number" class="form-control" style="width:55px;display: inline-block;margin-right: 20px"
" value="1" name="qty"><?=$url?>
</form>
```

Perhatikan pada script diatas ini. Disini kita buat input form agar nilai jumlah barang yang dipesan bisa dimasukkan ke cart(keranjang). Yang disimpan adalah id\_buku dan qty.

## Membuat tombol PINJAM proses simpan ke cart

```
</table><form action="<?=base_url('index.php/cart/add_cart')?>" method="post">
<input type="hidden" name="id_buku" value="<?= $detail->id_buku?>">
<input type="number" class="form-control" style="width:55px;display: inline-block;margin-right: 20px"
" value="1" name="qty"><?=$url?>
</form>
```

Ket: perhatikan attribut action di tag form. Proses penyimpanan item ke cart akan diproses di url **index.php/cart/add\_cart** artinya controller cart dan function add\_cart.

## CONTROLLER CART.PHP FUNCTION ADD\_CART

```
public function add_cart()
{
    if($this->input->post('tambah')){
        $this->load->model('m_buku');
        $detail=$this->m_buku->detail($this->input->post('id_buku'));

        $data = array(
            'id'      => $detail->id_buku,
            'qty'     => $this->input->post('qty'),
            'price'   => $detail->harga,
            'name'    => $detail->judul_buku,
            'options' => array('genre'=>$detail->nama_kategori)
        );

        $this->cart->insert($data);
        redirect('admin/detail_buku/' . $this->input->post('id_buku'), 'refresh');
    }
}
```

Ket:

Di cart ini komponen yang akan kita simpan adalah id\_buku, qty, harga, judul\_buku dan nama\_kategori sehingga kita perlu men-select berdasarkan id\_buku agar bisa menampilkan detail bukunya. Sehingga kita perlu file model untuk memanggil hasil querynya.

```
$this->load->model('m_buku');
$detail=$this->m_buku->detail($this->input->post('id_buku'));
```

Hasil querynya dari model akan disimpan ke variabel \$detail kemudian untuk menampilkan datanya langsung panggil \$detail->nama\_kolom\_table\_db.

```
$this->cart->insert($data);
```

Script diatas ini adalah untuk menyimpan data items sudah dimasukkan di array \$data ke function cart.

```
redirect('admin/detail_buku/' . $this->input->post('id_buku'), 'refresh');
```

Scrit diatas berfungsi untuk mengarahkan kembali ke halaman admin/detail\_buku/nilai id\_buku .

## MODEL M\_BUKU FUNCTION DETAIL

```

public function detail($a)
{
    $tm_buku=$this->db
        ->join('kategori','kategori.id_kategori=buku.id_kategori')
        ->where('id_buku',$a)
        ->get('buku')
        ->row();
    return $tm_buku;
}

```

## Membuat halaman tampil detail items yang dipesan

Dashboard Control panel

### Detail Peminjaman Buku

ID Buku	Nama Buku	Jumlah	Harga	Genre	Subtotal	Aksi
3	mencari teman dengan mudah	8	2500	Ilmu Pengetahuan Sosial	20000	Hps
1	trik jitu menghitung cepat dalam 1 detik	4	2000	Matematika Jitu	8000	Hps
GrandTotal					28000	

[Check-Out](#)

Membuat link untuk menampilkan halaman tampil detail items yang dipesan

Script untuk membuat simbol keranjang dan sebagai tombol untuk menampilkan detail items pesanan

```

<div class="navbar-custom-menu">
  <ul class="nav navbar-nav">
    <!-- Messages: style can be found in dropdown.less -->
    <li class="dropdown messages-menu">
      <a href="{base_url('index.php/cart')}" class="dropdown-toggle">
        <i class="fa fa-shopping-cart"></i>
        <span class="label label-success">
          <?= $this->cart->total_items();?>
        </span>
      </a>
    </li>
  </ul>
</div>

```

`<?= $this->cart->total_items()?>` script ini adalah untuk menampilkan total items yang telah tersimpan ke cart.

#### Controller cart function index

```
public function index()
{
    $data['konten']="showcart";
    $this->load->view('template', $data, FALSE);
}
```

#### Views showcart.php

```
1 <h2>Detail Peminjaman Buku</h2>
2 <form action="<?= base_url('index.php/cart/simpan')?>" method="post">
3 <table class="table table-hover table-striped">
4     <tr>
5         <td>ID Buku</td><td>Nama Buku</td><td>Jumlah</td><td>Harga</td><td>Genre</td><td>Subtotal</td><td>Aksi</td>
6     </tr>
7     <?php
8     foreach($this->cart->contents() as $items){
9     ?>
10    <tr>
11        <td>
12            <input type="hidden" name="id_buku[]" value="<?=$items['id']?>">
13            <input type="hidden" name="qty[]" value="<?=$items['qty']?>">
14
15            <?=$items['id']?></td><td><?=$items['name']?></td><td><?=$items['qty']?></td><td><?=$items['price']?></td><td>
16
17            <?=$items['options']['genre']?></td><td><?=$items['subtotal']?></td>
18            <td><a href="<?=base_url('index.php/cart/hapus_item/'.$items['rowid'])?>" onclick="return confirm('apakah yakin?')>Hps</a></td>
19        </tr>
20
21    <?php ?>
22    <tr>
23        <input type="hidden" name="grandtotal" value="<?=$this->cart->total() ?>">
24        <td colspan="5">GrandTotal</td><td><?=$this->cart->total() ?></td>
25    </tr>
26 </table>
27 <input type="submit" name="simpan" value="Check-Out" class="btn btn-success">
28 </form>
```

## MEMBUAT PROSES SIMPAN DARI CART KE DATABASE

Perhatikan halaman views showcart.php

```

1 <h2>Detail Peminjaman Buku</h2>
2 <form action="<?= base_url('index.php/cart/simpan')?>" method="post">
3 <table class="table table-hover table-striped">
4   <tr>
5     <td>ID Buku</td><td>Nama Buku</td><td>Jumlah</td><td>Harga</td><td>Genre</td><td>Subtotal</td><td>Aksi</td>
6   </tr>
7   <?php
8   foreach($this->cart->contents() as $items){
9     ?>
10    <tr>
11      <td>
12        <input type="hidden" name="id_buku[]" value="<?=$items['id']?>">
13        <input type="hidden" name="qty[]" value="<?=$items['qty']?>">
14
15        <?=$items['id']?></td><td><?=$items['name']?></td><td><?=$items['qty']?></td><td><?=$items['
16        price']?></td><td>
17
18        <?=$items['options']['genre']?></td><td><?=$items['subtotal']?></td>
19        <td><a href="<?=base_url('index.php/cart/hapus_item/'.$items['rowid'])?>" onclick="return
20        confirm('apakah yakin?')">Hps</a></td>
21      </tr>
22    <?php }?>
23    <tr>
24      <input type="hidden" name="grandtotal" value="<?=$this->cart->total() ?>">
25      <td colspan="5">GrandTotal</td><td><?=$this->cart->total() ?></td>
26    </tr>
27  </table>
28  <input type="submit" name="simpan" value="Check-Out" class="btn btn-success">
29 </form>

```

Ket:

Pada gambar diatas, setiap nilai data yang akan kita simpan ke database, kita masukkan ke input form type hidden. Data yang akan kita masukkan ke database adalah id\_buku yang tersimpan di items id, qty yang tersimpan di items qty dan grandtotal yang tersimpan di cart total.

Jangan lupa `<form action="<?= base_url('index.php/cart/simpan')?>" method="post">` pada action disamping ini harus ditentukan untuk membuat proses simpan ke database. Yaitu controller cart pada function simpan.

```

<input type="hidden" name="id_buku[]" value="<?=$items['id']?>">
<input type="hidden" name="qty[]" value="<?=$items['qty']?>">

```

Perhatikan name pada script diatas, isi dari name diakhiri menggunakan kurung siku menandakan data array agar bisa menyimpan lebih dari satu data.

## CONTROLLER CART.PHP FUNCTION SIMPAN

```

public function simpan()
{
    if($this->input->post('simpan')){
        $this->load->model('m_cart');
        $id_nota=$this->m_cart->simpan_cart();
        if($id_nota>0){
            $this->cart->destroy();
            redirect('cart/pembayaran/'.$id_nota,'refresh');
        } else {
            redirect('cart');
        }
    }
}
}

```

Ket: `if($this->input->post('simpan')){` ini adalah logika jika tombol input dengan nama simpan di klik atau mengandung nilai maka akan menjalankan script berikutnya.

```

$this->load->model('m_cart');
$id_nota=$this->m_cart->simpan_cart();

```

Script diatas menyatakan bahwa kita akan meng-insert-kan data ke database sehingga kita butuh file model. Oleh karena itu kita load model m\_cart dulu lalu kita panggil function simpan\_cart nya. Nilai hasil querynya kita simpan ke variable \$id\_nota. Kenapa kok variable id\_nota, karena setelah berhasil insert data nota maka kita ambil data id\_nota nya dengan men-select berdasarkan data yang terakhir dan berdasarkan data id\_peminjam nya.

```

if($id_nota>0){
    $this->cart->destroy();
    redirect('cart/pembayaran/'.$id_nota,'refresh');
} else {
    redirect('cart');
}

```

Logika diatas ini menyatakan bahwa jika \$id\_nota nilainya 1 keatas, maka

`$this->cart->destroy();` script disamping ini akan mendestroy atau menghapus semua data yang tersimpan di cart dan akan mengarahkan ke controller cart function pembayaran dengan membawa nilai parameter id\_nota. Jika nilainya 0 maka akan langsung diarahkan ke controller cart function index.

**MODEL M\_CART.PHP FUNCTION SIMPAN\_CART.**

```

public function simpan_cart()
{
    $tgl_deadline=date('Y-m-d',mktime(date('H'),date('i'),date('s'),date('m'),date('d')+7,date('Y')));
    $object=array(
        'id_peminjam'=>$this->session->userdata('id_peminjam'),
        'tgl_deadline'=>$tgl_deadline,
        'tgl_pinjam'=>date('Y-m-d'),
        'tgl_kembali'=>'',
        'denda'=>0,
        'grandtotal'=>$this->input->post('grandtotal'),
        'status'=>'',
        'bukti'=>''
    );
    $this->db->insert('nota', $object);
    $tm_nota=$this->db->order_by('id_nota','desc')
        ->limit(1)
        ->get('nota')
        ->row();
    $hasil=array();
    for($i=0;$i<count($this->input->post('id_buku'));$i++){
        $hasil[]=array(
            'id_nota'=>$tm_nota->id_nota,
            'id_buku'=>$this->input->post('id_buku')[$i],
            'jumlah'=>$this->input->post('qty')[$i]
        );
    }
    $proses=$this->db->insert_batch('peminjaman', $hasil);
    if($proses){
        return $tm_nota->id_nota;
    } else {
        return 0;
    }
}

```

Ket:

```

$tgl_deadline=date('Y-m-d',mktime(date('H'),date('i'),date('s'),date('m'),date('d')+7,date('Y')));

```

Script diatas bermaksud mencari tanggal setelah 7 hari dari hari sekarang dengan cara menggunakan function php mktime(jam,menit,detik,bulan,tanggal,tahun). Hasilnya akan disimpan ke variabel \$tgl\_deadline.

```

$object=array(
    'id_peminjam'=>$this->session->userdata('id_peminjam'),
    'tgl_deadline'=>$tgl_deadline,
    'tgl_pinjam'=>date('Y-m-d'),
    'tgl_kembali'=>'',
    'denda'=>0,
    'grandtotal'=>$this->input->post('grandtotal'),
    'status'=>'',
    'bukti'=>''
);
$this->db->insert('nota', $object);

```

Ini adalah script insert data ke database.

```
$tm_nota=$this->db->order_by('id_nota','desc')
->where('id_peminjam',$this->session->userdata('id_peminjam'))
->limit(1)
->get('nota')
->row();
```

Ini adalah script untuk menampilkan data nota yang baru saja insert kan. Kita ambil berdasarkan data yang terakhir menggunakan order by id\_nota desc dan where berdasarkan id\_peminjam. Dan kita tampilkan hanya satu data menggunakan limit 1. Serta diberi row() juga berfungsi hanya menyimpan 1 data.

```
$hasil=array();
for($i=0;$i<count($this->input->post('id_buku'));$i++){
    $hasil[]=array(
        'id_nota'=>$tm_nota->id_nota,
        'id_buku'=>$this->input->post('id_buku')[$i],
        'jumlah'=>$this->input->post('qty')[$i]
    );
}
$proses=$this->db->insert_batch('peminjaman', $hasil);
```

Script ini berfungsi untuk proses insert data item yang dipesan ke database secara masal atau lebih dari 1 data. Dengan cara menggunakan perulangan for karena inputannya berupa data array. Didalam perulangan, datanya akan kita masukkan ke dalam \$hasil. ini adalah data array 2 dimensi yaitu di dalam array ada array agar bisa menyimpan lebih dari satu record atau data. Kemudian pada script ini

```
$proses=$this->db->insert_batch('peminjaman', $hasil);
```

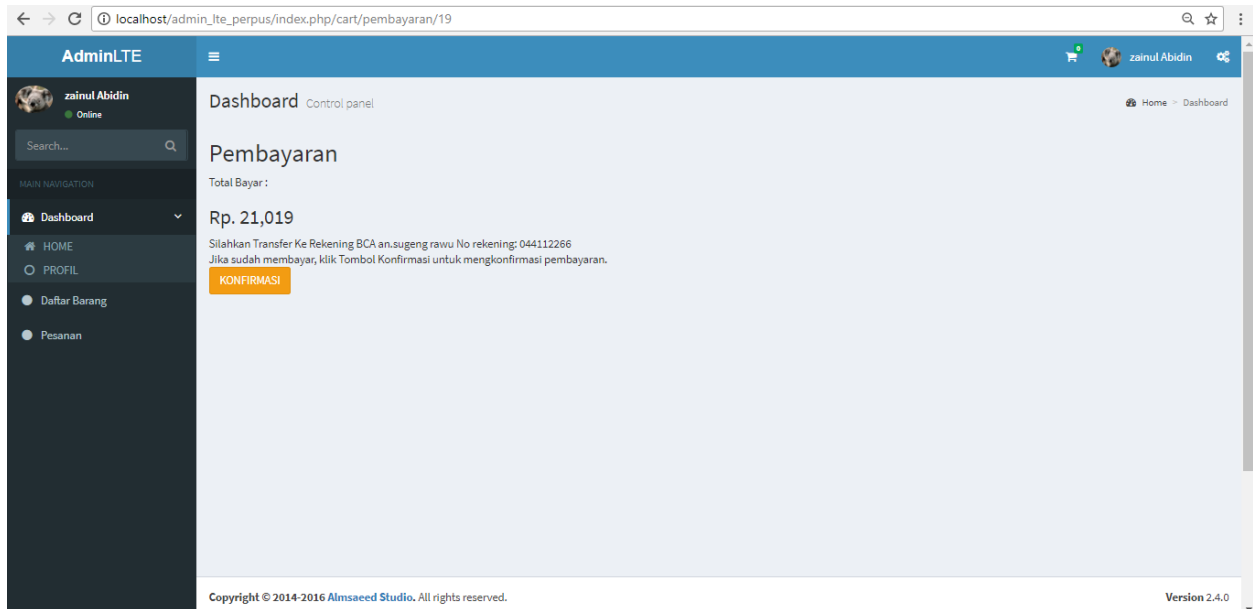
\$hasil yang menyimpan lebih dari 1 data ini disimpan ke function insert\_batch yaitu function yang berguna untuk meng-insert data lebih dari 1 data sekaligus.

```
if($proses){
    return $tm_nota->id_nota;
} else {
    return 0;
}
```

Logika diatas ini menyatakan bahwa jika \$proses menyimpan nilai 1 maka return membawa nilai id\_nota sedangkan jika \$proses bernilai 0 maka return membawa nilai 0.

## MEMBUAT HALAMAN KONFIRMASI PEMBAYARAN





## CONTROLLER CART FUNCTION PEMBAYARAN

```
public function pembayaran($id)
{
    $this->load->model('m_cart');
    $nota=$this->m_cart->get_total($id);
    $data['total']=$nota->grandtotal+$id;
    $data['konten']="v_pembayaran";
    $this->load->view('template', $data, FALSE);
}
```

`public function pembayaran($id)` pada script disamping \$id menangkap atau menyimpan nilai parameter id nota dari url yaitu 19 ->

`localhost/admin_lte_perpus/index.php/cart/pembayaran/19`

```
$nota=$this->m_cart->get_total($id);
$data['total']=$nota->grandtotal+$id;
```

\$data['total'] akan menyimpan penjumlahan nilai grandtotal dari db dengan \$id yang mana menyimpan nilai parameter id nota dari url. Sehingga \$data['total'] bisa dipanggil di views.

```
$data['konten']="v_pembayaran";
```

\$data['konten'] menyimpan file views

v\_pembayaran.php akan disisipkan di views template.

## MODEL M\_CART FUNCTION GET\_TOTAL DENGAN MEMBAWA PARAMETER \$ID

```
public function get_total($id)
{
    return $this->db->where('id_nota',$id)
        ->get('nota')
        ->row();
}
```

`public function get_total($id)` function ini dengan membawa parameter \$id yang akan menyimpan nilai ketika dipanggil di controller.

Jangan lupa selalu diberi **return** didalam function model agar function tersebut menampilkan atau membawa nilai jika dianggil.

Row() bertujuan untuk menampilkan hanya satu data.

## VIEWS V\_PEMBAYARAN.PHP

```
<h2>Pembayaran</h2>
Total Bayar : <h3><?= "Rp. ".number_format($total)?></h3>
Silahkan Transfer Ke Rekening BCA an.sugeng rawu No rekening: 044112266
<br>
Jika sudah membayar, klik Tombol Konfirmasi untuk mengkonfirmasi pembayaran.
<br><a href="<?=base_url('index.php/pesanan/konfirmasi/'. $this->uri->segment(3))?>" class="btn btn-warning">KONFIRMASI</a>
```

`<h3><?= "Rp. ".number_format($total)?></h3>` script ini memanggil variable dari controllernya dengan nama \$total yang asli nama variable di cotrollernya adalah \$data['total'].

Function Number\_format ini berasal dari librari PHP yang berfungsi untuk memberi pemisah antar ribuan.

`<a href="<?=base_url('index.php/pesanan/konfirmasi/'. $this->uri->segment(3))?>"` script ini adalah link untuk membuka halaman konfirmasi yang didalamnya ada fitur upload bukti pembayaran.

Coba dilihat isi base\_urlnya, disana memanggil controller pesanan function konfirmasi dan membawa nilai parameter yang berasal dari URL langsung yaitu 19 urlnya seperti ini ->

[localhost/admin\\_lte\\_perpus/index.php/cart/pembayaran/19](localhost/admin_lte_perpus/index.php/cart/pembayaran/19)

\$this->uri->segment(1) mengambil nilai nama controllernya misal **cart**

[localhost/admin\\_lte\\_perpus/index.php/cart/pembayaran/19](localhost/admin_lte_perpus/index.php/cart/pembayaran/19)

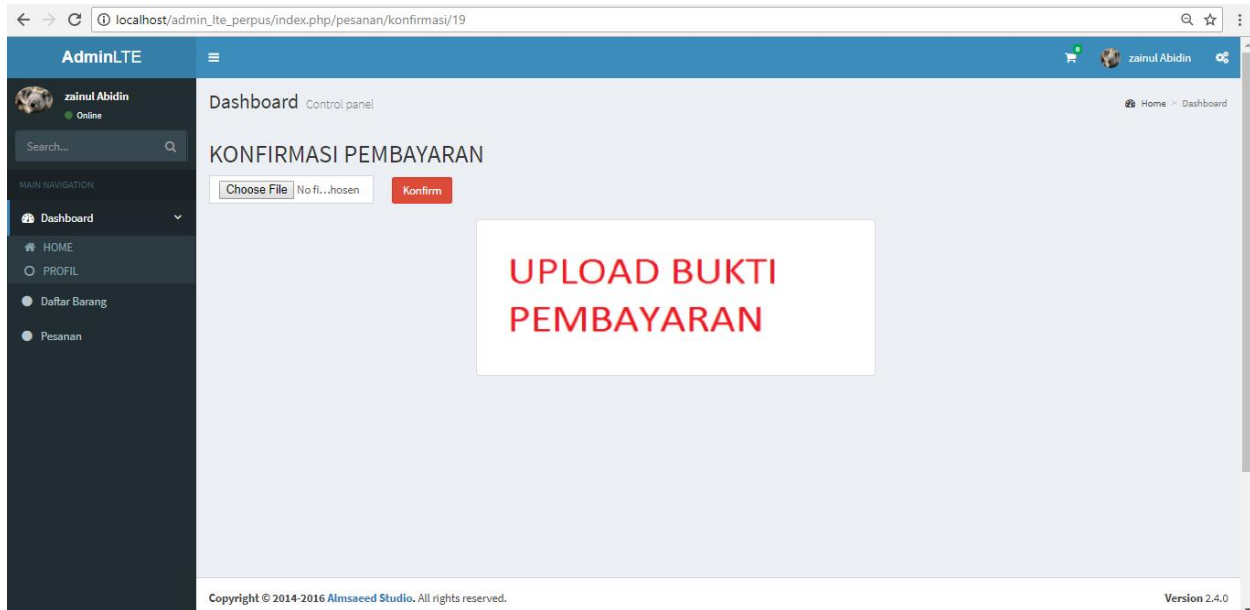
\$this->uri->segment(2) mengambil nilai nama functionnya misal **pembayaran**

[localhost/admin\\_lte\\_perpus/index.php/cart/pembayaran/19](localhost/admin_lte_perpus/index.php/cart/pembayaran/19)

\$this->uri->segment(3) mengambil nilai parameternya misal 19

localhost/admin\_lte\_perpus/index.php/cart/pembayaran/19

## MEMBUAT HALAMAN UPLOAD BUKTI PEMBAYARAN



### CONTROLLER PESANAN.PHP FUNCTION KONFIRMASI DENGAN MEMBAWA PARAMETER \$ID

```
class Pesanan extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        $this->load->model('m_pesanan', 'pesan');
    }
}
```

Load\_model kita letakkan di construct bertujuan untuk memberi script default yang artinya satu script untuk semua function. agar kita tidak menyebutkan berulang-ulang script yang sama di setiap function. Construct ini diletakkan di bawahnya **CLASS**.

`$this->load->model('m_pesanan', 'pesan');` penulisan load\_model bisa kita beri 2 parameter, parameter 1 nama file modelnya, parameter 2 alias atau penamaan dari file model. Jadi ketika kita akan memanggil function model maka kita beri nama aliasnya dulu misal "pesan". Jika tanpa alias maka pemanggilannya bisa menggunakan nama file modelnya langsung misal "m\_pesanan".

```
public function konfirmasi($id)
{
    $data['id_nota']=$id;
    $data['det_pesanan']=$this->pesan->detail_pesanan($id);
    $data['judul']="Konfirmasi Pembayaran";
    $data['konten']="v_konfirmasi";
    $this->load->view('template', $data, FALSE);
}
```

**public function konfirmasi(\$id)** \$id akan menyimpan nilai id\_nota dari url yaitu 19

localhost/admin\_lte\_perpus/index.php/pesanan/konfirmasi/19

**\$data['id\_nota']=\$id;** nilai id\_nota yang disimpan di variable \$id disimpan ke \$data['id\_nota'] agar bisa ditampilkan di views v\_konfirmasi.php

**\$data['det\_pesanan']=\$this->pesan->detail\_pesanan(\$id);** \$data['det\_pesanan'] akan menyimpan hasil query dari file model pesan function detail\_pesanan dengan membawa nilai id\_nota di \$id. Sehingga det\_pesanan bisa dipanggil di file views v\_konfirmasi.php.

**\$data['konten']="v\_konfirmasi";** \$data['konten'] menyimpan file views v\_konfirmasi.php agar bisa disisipkan ke views template.php

#### MODEL M\_PESANAN.PHP FUNCTION DETAIL\_PESANAN

```
public function detail_pesanan($id)
{
    return $this->db->where('id_nota',$id)->get('nota')->row();
}
```

Jangan lupa selalu diberi return di setiap function model agar bisa membawa atau menampilkan nilai ketika dipanggil. Jika yang kita return hasil dari query select atau get() maka function akan membawa hasil query select.

#### VIEWS V\_KONFIRMASI.PHP

```
1 <h2>KONFIRMASI PEMBAYARAN</h2>
2 <?php if ($this->session->flashdata('pesan')): ??>
3 <div class="alert alert-success">
4 <?= $this->session->flashdata('pesan');??>
5 </div>
6 <?php endif ??>
7
8 <form action="<?=base_url('index.php/pesanan/simpan_konfirm')?>" method="post" enctype="multipart/
form-data">
9 <input type="hidden" name="id_nota" value="<?= $id_nota;??>">
10 <input type="file" name="bukti" class="form-control" style="width:200px;display: inline-block;
margin-right: 20px">
11 <input type="submit" name="konfirmasi" value="Konfirmasi" class="btn btn-danger">
12 </form>
13 <br>
14 <div class="col-md-5 col-md-offset-3">
15 <div class="thumbnail">
16 <?php if ($det_pesan->bukti!=null): ??>
17 
18 <?php else: ??>
19 
20 <?php endif ??>
21 </div>
22 </div>
```

## MEMBUAT PROSES SIMPAN KONFIRM DAN UPDATE TABLE NOTA

### CONTROLLER PESANAN.PHP FUNCTION SIMPAN\_KONFIRM

```
25 public function simpan_konfirm()
26 {
27     if($this->input->post('konfirmasi')){
28
29         $config['upload_path'] = './asset/bukti/';
30         $config['allowed_types'] = 'gif|jpg|png';
31         $config['max_size'] = '1000';
32         $config['max_width'] = '1024';
33         $config['max_height'] = '768';
34
35         $this->load->library('upload', $config);
36
37         if ( ! $this->upload->do_upload('bukti')){
38             $error = $this->upload->display_errors();
39             $this->session->set_flashdata('pesan', $error);
40             redirect('pesanan/konfirmasi/' . $this->input->post('id_nota'),'refresh');
41         }
42         else{
43             $this->load->model('M_pesanan','pesan');
44             $proses=$this->pesan->simp_db($this->input->post('id_nota'),$this->upload->data('
file_name'));
45             if($proses){
46                 $this->session->set_flashdata('pesan', 'sukses mengirim bukti. Mohon Tunggu
Konfirmasi dari Admin');
47                 redirect('pesanan/konfirmasi/' . $this->input->post('id_nota'),'refresh');
48             } else {
49                 $this->session->set_flashdata('pesan', 'gagal mengirim bukti');
50                 redirect('pesanan/konfirmasi/' . $this->input->post('id_nota'),'refresh');
51             }
52         }
53     }
54 }
55 }
```

```
$config['upload_path'] = './asset/bukti/';
$config['allowed_types'] = 'gif|jpg|png';
$config['max_size'] = '1000';
$config['max_width'] = '1024';
$config['max_height'] = '768';
```

Ket: \$config ini adalah validation untuk upload. Upload\_path harus sudah dibuat folder tempat penyimpanan file gambarnya, file gambar harus berextensi gif,jpg, atau png dan max\_size nya harus dibawah 1000kb. Dan lain lain.

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

script ini berfungsi memanggil librari upload dari CI.

```
if ( ! $this->upload->do_upload('bukti')){
```

script ini melogikakan jika upload tidak bernilai true maka akan menjalankan script berikutnya. Yaitu menyimpan pesan ke flash data dan mengarahkan halaman ke sebelumnya. Sperti ini scriptnya:

```
if ( ! $this->upload->do_upload('bukti')){
    $error = $this->upload->display_errors();
    $this->session->set_flashdata('pesan', $error);
    redirect('pesanan/konfirmasi/' . $this->input->post('id_nota'),'refresh');
}
```

Jika bernilai true akan menjalankan script else berikut scriptnya:

```
else{
    $this->load->model('M_pesanan','pesan');
    $proses=$this->pesan->simp_db($this->input->post('id_nota'),$this->upload->data('file_name'));
    if($proses){
        $this->session->set_flashdata('pesan', 'sukses mengirim bukti. Mohon Tunggu Konfirmasi dari Admin');
        redirect('pesanan/konfirmasi/' . $this->input->post('id_nota'),'refresh');
    } else {
        $this->session->set_flashdata('pesan', 'gagal mengirim bukti');
        redirect('pesanan/konfirmasi/' . $this->input->post('id_nota'),'refresh');
    }
}
```

Script ini adalah proses pengupdate-an table nota dengan memanggil file model m\_pesan function simp\_db. Hasil dari query di model pesan function simp\_db akan disimpan ke \$proses. Kemudian akan dilogikakan jika \$proses bernilai 1 atau true maka akan menjalankan script berikutnya yaitu menyimpan pesan ke flashdata “sukses mengirim bukti” dan mengarahkan halaman ke halaman sebelumnya. Jika bernilai 0 maka akan menyimpan pesan flashdata “gagal mengirim bukti” dan mengarahkan ke halaman sebelumnya.

**MODEL M\_PESANAN FUNCTION SIMP\_DB (PARAMETER 1 ID\_NOTA, PARAMETER 2 FILE NAME)**

```
public function simp_db($id,$nama_file)
{
    $object=array(
        'bukti'=>$nama_file,
    );
    return $this->db->where('id_nota',$id)
        ->update('nota', $object);
}
```

Script diatas gunakan untuk mengupdate isi kolom bukti dengan nama file gambar di table nota. Scriptnya hampir sama dengan insert data Cuma disini ada function where nya karena mengupdate berdasarkan id\_nota.

## VIEWS V\_KONFIRM.PHP

```
1 <h2>KONFIRMASI PEMBAYARAN</h2>
2 <?php if ($this->session->flashdata('pesan')): ?>
3 <div class="alert alert-success">
4     <?= $this->session->flashdata('pesan');?>
5 </div>
6 <?php endif ?>
7
8 <form action="<?=base_url('index.php/pesanan/simpan_konfirm')?>" method="post" enctype="multipart/
  form-data">
9     <input type="hidden" name="id_nota" value="<?= $id_nota;?>">
10    <input type="file" name="bukti" class="form-control" style="width:200px;display: inline-block;
      margin-right: 20px">
11    <input type="submit" name="konfirmasi" value="Konfirmasi" class="btn btn-danger">
12 </form>
13 <br>
14 <div class="col-md-5 col-md-offset-3">
15     <div class="thumbnail">
16         <?php if ($det_pesan->bukti!=null): ?>
17             
18         <?php else: ?>
19             
20         <?php endif ?>
21     </div>
22 </div>
```

Ket:

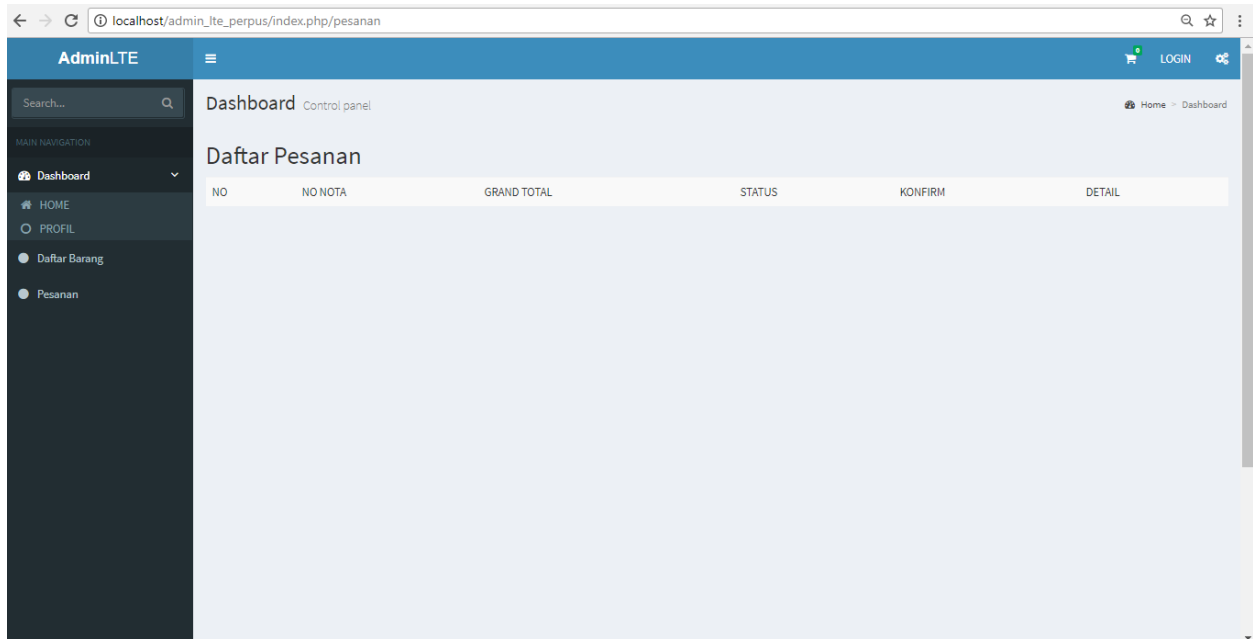
```
<?php if ($this->session->flashdata('pesan')): ?>
<div class="alert alert-success">
    <?= $this->session->flashdata('pesan');?>
</div>
<?php endif ?>
```

Script diatas menampilkan pesan yang sudah disimpan di flashdata.

```
<div class="col-md-5 col-md-offset-3">
    <div class="thumbnail">
        <?php if ($det_pesan->bukti!=null): ?>
            
        <?php else: ?>
            
        <?php endif ?>
    </div>
</div>
```

Script diatas menampilkan gambar bukti dari kolom bukti di table nota.

## MEMBUAT HALAMAN DAFTAR PESANAN



## CONTROLLER PESANAN.PHP FUNCTION INDEX

Jangan lupa harus kamu isi juga script construct ini: agar load\_model m\_pesanan bisa di load oleh semua function yang ada di controller ini. Tidak perlu me-load model lagi di setiap function.

```
class Pesanan extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        $this->load->model('m_pesanan','pesan');
    }
}
```

Inilah script controller pesanan.php function index

```
public function index()
{
    $data['judul']="daftar pesanan";
    $data['daftar_pesanan']=$this->pesan->tm_pesanan();
    $data['konten']="v_pesanan";
    $this->load->view('template', $data, FALSE);
}
```

## MODEL M\_PESANAN FUNCTION TM\_PESANAN()

```
public function tm_pesanan()
{
    return $this->db->where('id_peminjam',$this->session->userdata('id_peminjam'))->get('nota')->result();
}
```

## VIEWS V\_PESANAN.PHP



```

v_pesanan.php x
1 <h2>Daftar Pesanan</h2>
2 <?php if ($this->session->flashdata('pesan')): ?>
3     <div class="alert alert-success">
4         <?= $this->session->flashdata('pesan');?>
5     </div>
6 <?php endif ?>
7
8 <table class="table table-hover table-striped">
9     <tr>
10         <td>NO</td><td>NO NOTA</td><td>GRAND TOTAL</td><td>STATUS</td><td>KONFIRM</td><td>DETAIL</td>
11     </tr>
12     <?php
13         $no=0;
14         foreach ($daftar_pesanan as $pesan): |
15             $no++;?>
16         <tr>
17             <td><?= $no?></td><td><?= $pesan->id_nota?></td><td><?= $pesan->grandtotal?></td><td><?= $pesan->status?></td><td>
18 <?php if ($pesan->status!=null): ?>
19     Lunas
20 <?php else: ?>
21     <a href="<?=base_url('index.php/pesanan/konfirmasi/'.$pesan->id_nota)?>">Konfirm Pembayaran</a> | <a href="<?=base_url('
index.php/pesanan/hapus_pesanan/'.$pesan->id_nota)?>" onclick="return confirm('Apakah anda yakin menghapus Pesanan?')
">Hapus</a>
22 <?php endif ?>
23
24
25         </td><td><a href="#detail<?= $pesan->id_nota?>" data-toggle="modal">Lihat Barang</a>
26
27         <!-- modal menampilkan detail barang yang dipesan-->
28         <div class="modal fade" id="detail<?= $pesan->id_nota?>">
29             <div class="modal-dialog">
30                 <div class="modal-content">
31                     <div class="modal-header">
32                         <button type="button" class="close" data-dismiss="modal"><span aria-hidden="true">&times;</span><span class="sr-only">Close</span></button>
33                     <h4 class="modal-title">Modal title</h4>
34                     </div>
35                     <div class="modal-body">
36                         <?php
37                         foreach ($this->pesan->trans($pesan->id_nota as $value): ?>
38 <div class="row">
39 <div class="col-md-4">
40 <div class="thumbnail">
41 
42 </div>
43 </div>
44 <div class="col-md-8">
45 <h3><?= $value->judul_buku?></h3>
46 </div>
47 </div>
48 <?php endforeach ?>
49
50 </div>
51 <div class="modal-footer">
52 <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
53 <button type="button" class="btn btn-primary">Save changes</button>
54 </div>
55 </div><!-- /.modal-content -->
56 </div><!-- /.modal-dialog -->
57 </div><!-- /.modal -->
58
59         </td>
60     </tr>
61     <?php endforeach ?>
62 </table>

```