



Membuat Autentikasi Pengguna pada Laravel

Abdul Rohman

admin@abdulrohman.web.id :: <http://www.abdulrohman.web.id>

Abstrak

Autentikasi merupakan hal yang sangat penting dalam sebuah website. Dengan menggunakan autentikasi kita bisa mencegah orang yang tidak berkepentingan untuk mengakses situs yang kita buat. Dalam framework Laravel kita bisa membuat autentikasi pengguna pada aplikasi web kita. Tujuan membuat autentikasi ini untuk keamanan pada aplikasi web kita.

Kata Kunci: Autentikasi, User, Framework PHP, Laravel.

Pendahuluan

Autentikasi adalah suatu mekanisme untuk mengatur hak akses suatu halaman web, biasanya diawali dengan adanya proses login. Dimana user diminta menginput user id dan password. Jika berhasil login, Anda berhak masuk ke suatu halaman, jika tidak..anda tidak akan bisa masuk ke halaman tersebut. Saya rasa anda sudah paham tentang apa itu autentikasi, jadi saya pikir cukup sudah untuk penjelasannya. Mari kita lanjutkan ke cara membuat autentikasi menggunakan Laravel. Sebelumnya, saya asumsikan anda telah melakukan instalasi Laravel pada localhost dan melakukan setting database.

Lisensi Dokumen:

Copyright © 2008-2014 ilmuti.org

Seluruh dokumen di ilmuti.org dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari ilmuti.org



Pembahasan

Pertama-tama yang harus kita buat adalah tabel user. Untuk membuat tabel user, buat migrasi dengan menjalankan perintah sebagai berikut:

```
php artisan migrate:make create_user_table
```

Buka folder `app/migration`, anda akan melihat sebuah file berisi class yang bernama `'CreateUserTable'`. Pada class tersebut terdapat dua method bernama `'up'` dan `'down'`. Method `'up'` digunakan untuk membuat tabel, sedangkan method `'down'` digunakan untuk menghapus tabel. Tambahkan definisi tabel sebagai berikut pada method `'up'`:

```
Schema::create('users', function($table)
{
    $table->increments('id');
    $table->string('firstname', 20);
    $table->string('lastname', 20);
    $table->string('email', 100)->unique();
    $table->string('password', 64);
    $table->timestamps();
});
```

Kemudian pada method `'down'` tambahkan perintah sebagai berikut:

```
Schema::drop('users');
```

Selanjutnya, pada *command line* jalankan perintah:

```
php artisan migrate
```

Dengan begitu, sebuah tabel bernama `'users'` akan dibuat pada database, lengkap beserta seluruh field-field yang dibutuhkan. Untuk membuat sistem autentikasi, diperlukan sebuah model, controller dan view. Model yang digunakan adalah model bernama `'User'` yang biasanya sudah ada secara default pada instalasi Laravel.



Kemudian untuk controller kita akan membuat sebuah controller baru bernama **'UserController'**, untuk membuat **'UserController'**, jalankan perintah sebagai berikut menggunakan *command line*:

```
php artisan controller:make UserController
```

Maka, sebuah file baru bernama **'UserController.php'** akan tercipta pada folder **app/controller**.

Membuat Halaman Registrasi

Setelah controller kita buat, sekarang kita akan menampilkan sebuah halaman registrasi. Pertama-tama, kita akan mengubah fungsi **'create'** pada **'UserController'** menjadi fungsi **'register'**:

```
public function register() { return View::make('register'); }
```

Pada fungsi register diatas, kita akan diarahkan pada sebuah view bernama **'register'**. Namun, view tersebut masih belum kita buat. Untuk itu pada folder **app/views** tambahkan sebuah file baru bernama **'register.blade.php'**:

```
@extends('layout')
@section('content')
    <div class="container"> <h2>Register</h2>

    @if(Session::has('pesan'))
        <div class="alert alert-success">{{ Session::get('pesan')}}</div>
    @endif

    {{Form::open(array('action' => 'UserController@store')) }}
    {{Form::label('firstname', 'First Name') }}
    {{Form::text('firstname', '', array('class' => 'form-control'))}}
    {{Form::label('lastname', 'Last Name') }}
    {{Form::text('lastname', '', array('class' => 'form-control'))}}
```

Lisensi Dokumen:

Copyright © 2008-2014 ilmuti.org

Seluruh dokumen di ilmuti.org dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari ilmuti.org



```
{{Form::label('email', 'Email')}}
{{Form::text('email', '', array('class' => 'form-control'))}}
{{Form::label('password', 'Password')}}
{{Form::password('password', array('class' => 'form-control'))}}
<br>
{{Form::submit('Registerin Dong!', array('class' => 'btn btn-primary')) }}
{{ Form::close() }} </div> @stop
```

Setelah itu, buatlah route baru pada file **routes.php** seperti berikut ini:

```
Route::get('register', 'UserController@register');
Route::post('store', 'UserController@store');
```

Setelah route dibuat, sekarang anda bisa membuka halaman register yang barusan anda buat melalui browser dengan cara mengakses route **'register'**. Misal:

<http://localhost/laravel/register>

CRUD App - Google Chrome

localhost/register

Register

First Name

Last Name

Email

Password

Registerin Dong!

Lisensi Dokumen:

Copyright © 2008-2014 ilmuti.org

Seluruh dokumen di ilmuti.org dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari ilmuti.org



Registrasikan Pengguna

Setelah halaman registrasi kita buat, kita akan memasukan data dari *form* yang barusan kita buat ke dalam database. Pada form tersebut terdapat kode sebagai berikut:

```
{{Form::open(array('action' => 'UserController@store')) }}
```

Itu berarti, ketika disubmit, data dari form tersebut akan di-*handle* Oleh method **'store'** pada **'UserController'**. Untuk itu silahkan buka **'UserController'** kemudian tambahkan perintah sebagai berikut ke dalam method **'store'**:

```
$user = new User();

$user->firstname = Input::get('firstname');
$user->lastname  = Input::get('lastname');
$user->email     = Input::get('email');
$user->password  = Hash::make(Input::get('password'));

$user->save();

return Redirect::to('register')->with('pesan', 'Registrasi
berhasil!');
```

Sekarang, apabila anda isi form tersebut kemudian tekan tombol **'Registerin Dong!'** (seperti submit), maka data yang anda masukan akan tersimpan di database.

Membuat Fitur Login

Untuk membuat halaman Login, pertama-tama buatlah dua buah fungsi baru yaitu **'login'** dan **'authenticate'**:

```
public function login(){
}

public function authenticate(){
}
```

Lisensi Dokumen:

Copyright © 2008-2014 ilmuti.org

Seluruh dokumen di ilmuti.org dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari ilmuti.org



Selain itu, buatlah dua buah route baru pada file **routes.php** sebagai berikut:

```
Route::get('login', 'UserController@login');  
Route::post('authenticate', 'UserController@authenticate');
```

Halaman Login

Sekarang, kita akan membuat tampilan halaman login. Pada method **'login'** tambahkan baris perintah sebagai berikut:

```
return View::make('login');
```

Kemudian pada direktori **app/views** tambahkan sebuah view baru dengan nama file **login.blade.php**:

```
@extends('layout')  
  
@section('content')  
<div class="container">  
<div class="col-md-5 col-md-offset-3">  
<h2>Login</h2>  
<hr />  
  
@if(Session::has('pesan_error'))  
    <div class="alert alert-danger">{{ Session::get('pesan_error')}}</div>  
@endif  
  
{{Form::open(array('action' => 'UserController@authenticate')) }}  
{{Form::label('email', 'Email') }}  
{{Form::text('email', '', array('class' => 'form-control'))}}  
{{Form::label('password', 'Password') }}  
{{Form::password('password', array('class' => 'form-control'))}}  
{{Form::submit('Login', array('class' => 'btn btn-primary')) }}  
{{Form::close() }}  
  
</div>  
</div>  
@stop
```

Lisensi Dokumen:

Copyright © 2008-2014 ilmuti.org

Seluruh dokumen di ilmuti.org dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari ilmuti.org



Autentikasi

Ketika tombol login di-klik, *form* akan diarahkan ke method **'authenticate'** pada *UserController*. Pada method inilah autentikasi dilakukan, untuk itu tambahkan baris perintah sebagai berikut pada method *authenticate*:

```
if (Auth::attempt(array('email' => Input::get('email'),  
'password' => Input::get('password'))))  
{  
    echo "login berhasil";  
}  
else{  
    return Redirect::to('login')->with('pesan_error', 'Login gagal,  
email atau password salah!');  
}
```

Sekarang, jika anda tekan tombol **'login'** dan autentikasi berhasil maka akan tampil tulisan **"login berhasil"**. Namun, jika autentikasi gagal, maka sebuah pesan kesalahan akan ditampilkan pada halaman login.

Login

Login gagal, email atau password salah!

Email

Password

Login

Lisensi Dokumen:

Copyright © 2008-2014 ilmuti.org

Seluruh dokumen di ilmuti.org dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari ilmuti.org



Membuat Halaman Dashboard

Sebelumnya, apabila kita sudah berhasil login kita hanya akan melihat tulisan **'login berhasil'**. Nah, kali ini kita akan membuat supaya pengguna diarahkan ke halaman dashboard apabila login berhasil. Buat sebuah controller baru bernama **'DashboardController'**:

```
<?php
class DashboardController extends BaseController
{
    public function index(){
        return View::make('dashboard');
    }
}
?>
```

Setelah itu, buatlah sebuah route baru pada **routes.php** sebagai berikut:

```
Route::get('dashboard', 'DashboardController@index');
```

Kemudian, jangan lupa untuk membuat sebuah view baru bernama **'dashboard.blade.php'**. Setelah itu, ubah autentikasi yang ada pada method **authenticate** menjadi sebagai berikut:

```
function authenticate(){
    if (Auth::attempt(array('email' => Input::get('email'),
    'password' => Input::get('password'))))
    {
        return Redirect::to('dashboard');
    }

    return Redirect::to('login')->with('pesan_error', 'Login gagal,
    email atau password salah!');
}
```

Sampai disini, apabila anda mencoba login kembali maka akan diarahkan ke halaman dashboard. Tentu halaman dashboard yang dimaksud adalah halaman kosong, mengingat kita baru membuat viewnya saja, tanpa ada isinya.

Lisensi Dokumen:

Copyright © 2008-2014 ilmuti.org

Seluruh dokumen di ilmuti.org dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari ilmuti.org



Kami adalah sekelompok orang yang sadar akan pentingnya ilmu :: Orang lain bisa saja berfikir sebaliknya :: Mungkin juga sejalan dengan pemahaman kami :: Ini sangat tergantung sudut pandang sebuah ilmu :: Tetapi bagi kami tidak ada kata lain :: Mengusai ilmu adalah segalanya :: Entah sampai kapan kami memegang prinsip ini :: Namun kami yakin bahwa segalanya butuh ilmu :: Indonesia harus menjunjung ini setinggi tingginya :: Lembaga terkait harus mampu mendukung sepenuh hati :: Majukan anak bangsa dari sisi kecerdasan :: Untuk bekal kemandirian dan kemajuan serta kebesaran bangsa :: Tekad kami kedepan adalah pasti :: Ilmu harus bisa diciptakan dan digunakan bersama :: Jika kami sadar ilmu datang dan hadir bukan karena kami ciptakan :: Untuk sebaliknya kami pun sadar ilmu datang dan hadir diciptakan untuk kami :: Namun ilmu harus terus dilakukan pengkajian, penelitian, pengembangan, pengabdian :: Akhir nya ilmu pun harus mendapat pengakuan dan manfaat :: Ilmu kami tidak hanya berfikir sebatas apa yang kami ketahui dan dapat kami lakukan :: Dan lebih jauh dari itu kami berfikir tentang apa yang tidak kami ketahui dan tidak dapat kami lakukan :: Ilmu kami tidak hanya mencapai batas maksimal kemampuan, lebih jauh dari itu kami harus berfikir menembus batas, hingga diluar batas maksimal kemampuan

Membuat Fungsi Logout

Sekarang, saatnya kita membuat fungsi logout. Untuk itu, kita akan membuat view 'dashboard' yang telah kita buat dari halaman kosong menjadi sebagai berikut:

```
@extends('layout')

@section('content')
<nav class="navbar navbar-inverse navbar-static-top"
role="navigation">
  <div class="container-fluid">
    <!-- Brand and toggle get grouped for better mobile display -
  ->
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1">
        <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="#">AuthApp</a>
    </div>

    <!-- Collect the nav links, forms, and other content for
    toggling -->
    <div class="collapse navbar-collapse" id="bs-example-navbar-
collapse-1">
      <ul class="nav navbar-nav navbar-right">
        <li><a href="logout"><i class="glyphicon glyphicon-
off"></i> logout</a></li>
      </ul>
    </div><!-- /.navbar-collapse -->
  </div><!-- /.container-fluid -->
</nav>

<div class="container">
  <h1>Dashboard</h1>
</div>
@stop
```

Lisensi Dokumen:

Copyright © 2008-2014 ilmuti.org

Seluruh dokumen di ilmuti.org dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari ilmuti.org



Selanjutnya, perhatikan pada halaman dashboard diatas terdapat kode sebagai berikut:

```
<li><a href="logout"><i class="glyphicon glyphicon-off"></i>
```

Pada atribut href, kita bisa melihat bahwa kita akan diarahkan pada route **'logout'** apabila link tersebut kita klik. Namun, route **'logout'** masih belum kita buat, oleh itu tambahkan route **'logout'** sebagai berikut pada file **routes.php**:

```
Route::get('logout', 'UserController@logout');
```

Kemudian, tambahkan sebuah method pada **'UserController'** sebagai berikut:

```
public function logout(){  
    Auth::logout();  
    return Redirect::to('login')->with('pesan', 'berhasil  
logout');  
}
```

Selanjutnya, untuk menampilkan pesan berhasil logout, tambahkan perintah sebagai berikut pada bagian atas *form login*:

```
@if(Session::has('pesan'))  
{{ Session::get('pesan') }}  
  
@endif
```

Lisensi Dokumen:

Copyright © 2008-2014 ilmuti.org

Seluruh dokumen di ilmuti.org dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari ilmuti.org



Penutup

Sejauh ini Anda telah berhasil membuat halaman *registrasi*, *login* dan *logout*. Tentunya, masih ada beberapa kelemahan dalam aplikasi yang barusan kita buat. Misalnya, akan terjadi error apabila kita memasukan email yang sama saat registrasi. Kemudian, jika kita belum login dan mengakses **http://localhost/dashboard** seharusnya pengguna diarahkan ke halaman login. Dan masih ada beberapa hal lagi yang harus diperbaiki namun tidak saya bahas dalam tulisan. Saya berharap Anda bisa **mencobanya** sendiri.

Semoga artikel ini sangat membantu bagi Anda yang ingin belajar membuat aplikasi web menggunakan framework Laravel. Jika Anda menyukai artikel ini atau memiliki pertanyaan, silakan posting komentar dibawah sehingga mendorong saya untuk menulis artikel yang lebih berguna. Selamat berjumpa di artikel saya selanjutnya. Terima kasih telah membaca artikel ini. :D

Referensi

<http://digitalmapia.com/membuat-autentikasi-pengguna-dengan-menggunakan-laravel/>
<http://blog.novay.web.id/blog/2013/04/03/membuat-autentikasi-login-laravel-bagian-1/>
<http://blog.novay.web.id/blog/2014/03/05/membuat-autentikasi-login-laravel-bagian-2/>
<http://code.tutsplus.com/tutorials/authentication-with-laravel-4--net-35593>
<http://laravel.com/docs/security#authenticating-users>

Lisensi Dokumen:

Copyright © 2008-2014 ilmuti.org

Seluruh dokumen di ilmuti.org dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari ilmuti.org



Biografi



Abdul Rohman

Lahir di Tangerang pada tanggal 10 Mei 1990. Merupakan seorang yang berpenampilan sederhana tapi berkompeten dalam hal-hal yang berhubungan dengan komputer. Dia mulai memperdalam dunia komputer pada jurusan Multimedia di SMKN 1 Tangerang. Saat ini dia sedang menempuh pendidikan perkuliahan di STMIK Raharja jurusan Sistem Informasi konsentrasi *Computer Accountancy* dan selanjutnya

akan menempuh program masternya di Universitas Negeri atau Luar Negeri (*Insyallah...amin*). Disamping kuliah dia juga bekerja di perusahaan Digital Printing yang mana menurutnya disitu ada keseimbangan antara ide kreatif, teknologi dan konsep. Cita-citanya yang sekarang adalah ingin menjadi seorang *Technopreneur*.

Info lebih lanjut (jangan hubungi dokter)→ www.abdulrohman.web.id

Lisensi Dokumen:

Copyright © 2008-2014 ilmuti.org

Seluruh dokumen di ilmuti.org dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari ilmuti.org