



Membuat CRUD Sederhana pada Framework Laravel

Abdul Rohman

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

Abstrak

CRUD (*Create, Read, Update dan Delete*) merupakan program dasar yang harus dikuasai oleh seorang programmer. Pembuatan CRUD pada masing-masing bahasa pemrograman sangat beragam dalam cara pembuatannya, tapi konsepnya sama. Saat ini ada juga yang menggunakan CRUD Generator dan banyak jenisnya. Dalam framework Laravel juga demikian, tapi disini hanya akan menjelaskan pembuatan CRUD sederhana tanpa CRUD Generator.

Kata Kunci: public, URL, Framework PHP, Laravel.

Pendahuluan

Pada sesi ini kita akan menggunakan Laravel untuk membuat proses *create, read, update, dan delete* (CRUD) sederhana. CRUD merupakan proses standar yang biasa ditemukan pada sebuah sistem aplikasi web. Berikut adalah *user stories* dari fitur CRUD yang akan kita buat:

1. Pengguna dapat melihat semua data user.
2. Pengguna dapat menambahkan data *user* dengan memasukkan data email, nama serta password.
3. Pengguna dapat melihat informasi salah satu *user* secara mendetil.
4. Pengguna dapat memperbaharui data user.
5. Pengguna dapat menghapus user tertentu.

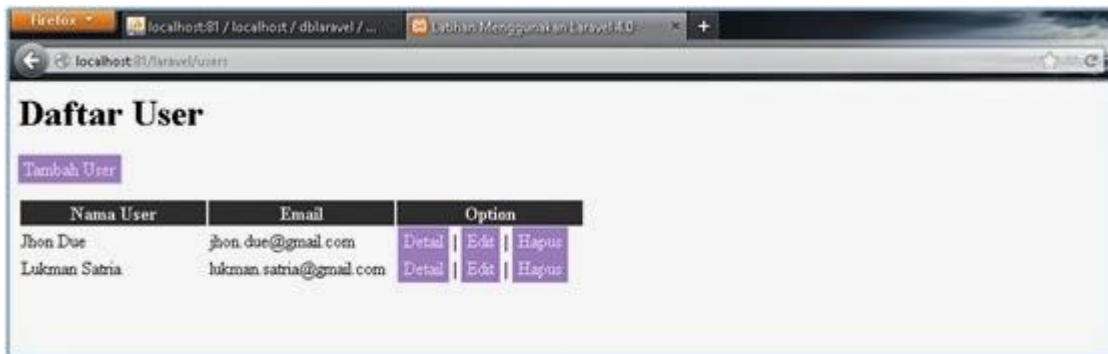
Lisensi Dokumen:

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



Lebih kurang kita akan membuat sebuah aplikasi web sederhana seperti gambar berikut:



Pembahasan

Langkah 1 : Pembuatan Database

Berdasarkan user stories di atas, langkah awal yang kita lakukan adalah membuat basisdata yang untuk menyimpan data user. Kita akan buat sebuah basisdata dengan nama dblaravel yang memiliki sebuah tabel *Users*. Informasi yang perlu kita simpan adalah nama, email, nomor hp, alamat dan password, serta informasi kapan dibuat dan kapan diperbaharui sebagai atribut standar yang wajib ada pada model Laravel. Langsung saja ke *phpMyAdmin* untuk membuat tabel tersebut.

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	id	int(10)			Tidak	Tidak ada	AUTO_INCREMENT	Ubah Hapus Lainnya
2	email	varchar(100)	latin1_swedish_ci		Tidak	Tidak ada		Ubah Hapus Lainnya
3	name	varchar(100)	latin1_swedish_ci		Tidak	Tidak ada		Ubah Hapus Lainnya
4	password	varchar(100)	latin1_swedish_ci		Tidak	Tidak ada		Ubah Hapus Lainnya
5	address	varchar(100)	latin1_swedish_ci		Tidak	Tidak ada		Ubah Hapus Lainnya
6	phone	varchar(50)	latin1_swedish_ci		Tidak	Tidak ada		Ubah Hapus Lainnya
7	created_at	timestamp		on update CURRENT_TIMESTAMP	Tidak	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP	Ubah Hapus Lainnya
8	updated_at	timestamp			Tidak	0000-00-00 00:00:00		Ubah Hapus Lainnya

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



Langkah 2: Setting Konfigurasi Database

Langkah selanjutnya adalah mengatur database pada file `laravel/app/config/database.php` agar bisa diakses. Secara default basisdata yang digunakan Laravel adalah mysql. Karena kita menggunakan mysql sebagai basisdata, maka atur koneksi basisdata seperti berikut :

```
55 'mysql' => array(  
56     'driver' => 'mysql',  
57     'host' => 'localhost',  
58     'database' => 'dblaravel',  
59     'username' => 'root',  
60     'password' => '',  
61     'charset' => 'utf8',  
62     'collation' => 'utf8_unicode_ci',  
63     'prefix' => '',  
64 ),  
65
```

Langkah 3 : Pembuatan Model

Kita akan mulai dengan model. Sederhananya sebuah model pada Laravel merupakan perawakilan dari sebuah tabel yang ada pada basisdata. Sebuah model merupakan sebuah objek yang sudah memiliki berbagai fungsi turunan dari *Eloquent* laravel. Untuk membuat sebuah model itu sangat mudah, kita cukup menulis beberapa baris kode saja seperti berikut:

```
Proyek.php  
1  <?php  
2  
3  class Proyek extends Eloquent{  
4  
5      /**  
6       * The database table used by the model.  
7       *  
8       * @var string  
9       */  
10     protected $table = 'proyek';  
11  
12 }
```

Pada gambar di atas, kita membuat sebuah model untuk tabel proyek. Cukup dengan menuliskan nama tabel yang sesuai dengan model yang ingin kita buat. Lho?

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



Bukannya kita membuat tabel user? Benar. Gambar di atas hanya sebuah demo untuk membuat sebuah model. Pada kasus sesi kali ini, karena tabel yang kita buat adalah tabel user, Laravel sudah menyediakan model *User* secara default. Kita tinggal mengganti nama tabel yang ada pada model *User* sesuai dengan tabel yang sudah kita buat.

```
Proyek.php
1 <?php
2
3 use Illuminate\Auth\UserInterface;
4 use Illuminate\Auth\Reminders\RemindableInterface;
5
6 class User extends Eloquent implements UserInterface, RemindableInterface {
7
8     /**
9      * The database table used by the model.
10      *
11      * @var string
12      */
13     protected $table = 'users';
14
15     /**
16      * The attributes excluded from the model's JSON form.
17      *
18      * @var array
19      */
20     protected $hidden = array('password');
21
22     /**
23      * Get the unique identifier for the user.
24      *
25      * @return mixed
26      */
27     public function getAuthIdentifier()
28     {
29         return $this->getKey();
30     }
31
32     /**
33      * Get the password for the user.
34      *
35      * @return string
36      */
37     public function getAuthPassword()
38     {
39         return $this->password;
40     }
41 }
```

Catatan:

- Jangan pusing dengan kode pada baris 3 dan 4 dari gambar di atas.
- Model user menggunakan *User Interface* dan *Remindable Interface* untuk proses Authentication sehingga pada baris 3 dan 4 harus dideklarasikan seperti hal tersebut.

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



- Jangan pusing dengan istilah *User Interface* dan *Remindable Interface*.

Langkah 4 : Pembuatan Controller

Bagian ini merupakan inti nantinya dari apa yang akan kita buat. Kita akan menggunakan *command prompt* untuk membuat controller. Tinggal buka CMD anda dari sistem windows dan masuk ke folder tempat laravel berada lalu jalankan perintah seperti gambar berikut :

```
C:\xampp\htdocs\laravel>php artisan controller:nake UserController
Controller created successfully!
C:\xampp\htdocs\laravel>
```

Sekarang pada folder *app/controller* sudah ada file *UserController.php* sebagai tanda pembuatan *controller* berhasil. Laravel akan memberikan kerangka fungsi CRUD pada setiap controller yang kita generate. Silahkan buka file tersebut, lebih kurang seperti berikut :

```
UserController.php
1 <?php
2
3 class UserController extends \BaseController {
4
5     /**
6      * Display a listing of the resource.
7      *
8      * @return Response
9      */
10    public function index()
11    {
12        //
13    }
14
15    /**
16     * Show the form for creating a new resource.
17     *
18     * @return Response
19     */
20    public function create()
21    {
22        //
23    }
24
25    /**
26     * Store a newly created resource in storage.
27     *
28     * @return Response
29     */
30    public function store()
31    {
32        //
33    }
34
35    /**
36     * Display the specified resource.
37     *
38     * @param int $id
39     * @return Response
40     */
41    public function show($id)
42    {
43    }
```

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



Langkah 5 : Pembuatan Route

Untuk dapat mengakses fungsi yang ada pada controller, kita harus mendeklarasikan controller tersebut pada routes. Silahkan buka file *App/routes.php* lalu tambahkan kode berikut :

```
Route::resource('users', 'UserController');
```

Sekarang coba Anda menambahkan satu baris kode seperti gambar di bawah pada controller user.

```
10 public function index()  
11 {  
12     return "Hello CRUD!";  
13 }
```

Lalu buka **http://localhost/laravel/public/users** dari browser Anda dan Anda akan melihat tulisan "Hello CRUD!". Jika Anda berhasil sampai pada tahap tersebut, berarti Anda sudah siap untuk membuat sistem CRUD user.

Anda bisa melihat fungsi dan format url yang dapat diakses dari UserController dengan menggunakan perintah *php artisan routes* pada CMD anda. Informasi yang ada dalam garis berwarna hijau pada gambar di bawah merupakan url yang dapat Anda akses dari browser. Informasi yang ada pada kotak garis berwarna biru merupakan nama Route yang dapat Anda gunakan. Sementara itu, garis putus-putus berwarna orange menandakan controller yang diakses dan garis putus-putus berwarna kuning menandakan fungsi yang digunakan pada controller. Jadi, jika Anda membuka alamat */users* maka Route akan memanggil fungsi *index* pada controller *users*. Sederhananya begitu. Nanti kita akan bahas lebih jauh penggunaan Route ini.

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



```
C:\xampp\htdocs\laravel>php artisan routes
```

Domain	URI	Name	Action	Before Filters	After Filters
GET	/users	users.index	UserController@index		
GET	/users/create	users.create	UserController@create		
POST	/users	users.store	UserController@store		
GET	/users/{users}	users.show	UserController@show		
GET	/users/{users}/edit	users.edit	UserController@edit		
PUT	/users/{users}	users.update	UserController@update		
PATCH	/users/{users}	users.update	UserController@update		
DELETE	/users/{users}	users.destroy	UserController@destroy		
GET	/		Closure		

```
C:\xampp\htdocs\laravel>
```

Langkah 6 : Menampilkan Daftar User

Sekarang kita kembali pada `UserController`. Fungsi `index` merupakan fungsi default yang akan dipanggil jika pengguna mengakses `/users`. Pada `index` kita akan menampilkan semua data user yang ada pada model. Sangat mudah sekali untuk melakukan hal tersebut.

1. Ambil data semua user menggunakan model `User` yang sudah kita buat. Semua data dari model kita simpan pada sebuah array yang nantinya dapat kita akses dari view.

```
10 public function index()  
11 {  
12     // Mengambil semua data user  
13     $data['daftar_user'] = User::all();  
14  
15     return view::make('user/list', $data);  
16 }
```

Lisensi Dokumen:

Copyright © 2008-2014 ilmuti.org

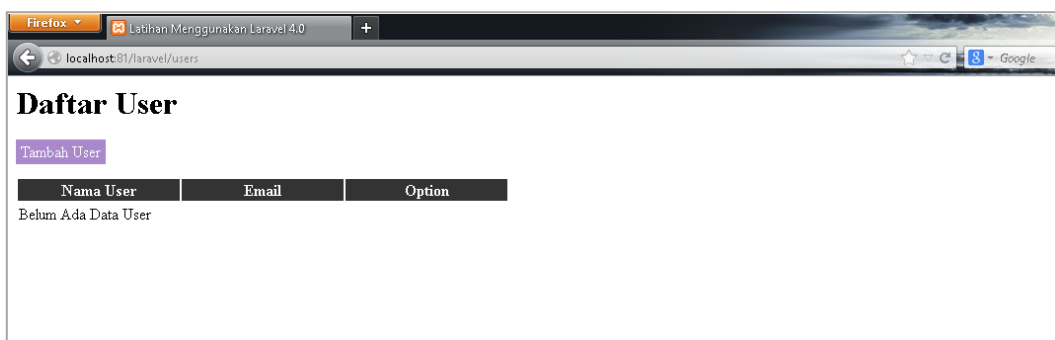
Seluruh dokumen di ilmuti.org dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali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



2. Buat file **list.php**, simpan pada folder view dengan folder baru **user**. Pada **view list.php** kita dapat memanggil data user melalui object **daftar_user**. Jika belum ada data user, kita tampilkan pesan **“Belum Ada Data User”**. Untuk semua data user yang ada pada **daftar_user**, kita akan menampilkan data nama dan email. Berikut kode sederhananya.

```
list.php
1 <html>
2 <head>
3 <title>Latihan Menggunakan Laravel 4.0</title>
4 <style type="text/css">
5   th{ width: 33%; text-align: center; background-color: #333; color: #fff;}
6   a{ padding: 3px 5px; background-color: #aa88cc; text-decoration: none; color: #fff;}
7   a:hover{ background-color: #aa55cc;}
8 </style>
9 </head>
10 <body>
11 <h1>Daftar User</h1>
12 <p><a href="">Tambah User</a></p>
13 <table style="width: 500px;">
14 <tr><th>Nama User</th><th>Email</th><th>Option</th></tr>
15 <?php
16 // Memastikan bahwa ada data user
17 if(count($daftar_user) == 0){
18   echo '<tr><td colspan="3">Belum Ada Data User</td></tr>';
19 }else{
20 // Menampilkan data untuk semua user yang ada
21 foreach ($daftar_user as $usr) {
22   echo '<tr><td>'. $usr->name. '</td><td>'. $usr->email. '</td>';
23
24   echo '<a href="">Edit</a> <a href="">Edit</a></tr>';
25 }
26 }
27 </table>
28 </body>
29 </html>
```

3. Buka **http://localhost/laravel/users** pada browser Anda, maka akan muncul tampilan seperti gambar di bawah. Belum ada data user yang akan ditampilkan.



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



Kita selesai bermain pada bagian ini. Selanjutnya kita akan masuk ke fungsi create. Tapi sebelum itu tambahkan link pada kode view *list.php* seperti gambar di bawah agar ketika pengguna mengklik tombol **Tambah User**, kita akan dilempar ke halaman pendaftaran user.

```
10 </body>
11 <h1>Daftar User</h1>
12 <p><a href="<?php url('/users/create'); ?>">Tambah User</a></p>
13 <table style="width: 500px;">
14 <tr><th>Nama User</th><th>Email</th><th>Option</th></tr>
```

Langkah 7 : Menambahkan Data User

Pada bagian ini kita hanya akan menampilkan form pendaftaran user. Sekarang kita buat view untuk hal tersebut. Buat file baru seperti berikut lalu simpan dengan nama **add.php** pada folder *app/view/user*. Perlu diperhatikan bahwa, kode pada gambar dibawah menggunakan tag `<form>` dengan **method POST**. Sehingga kita akan merujuk pada Route dengan method POST yang dimiliki oleh UserController yaitu route */users*. Sehingga pada tag `<form>` kita akan set action nya adalah menuju ke link */users*.

```
1 <html>
2 <head>
3 <title>Latihan Menggunakan Laravel 4.0</title>
4 <style type="text/css">
5 th{ width: 33%; text-align: center; background-color: #333; color: #fff;}
6 a{ padding: 3px 5px; background-color: #aa88cc; text-decoration: none;color:#fff;}
7 a:hover{background-color: #aa55cc;}
8 </style>
9 </head>
10 <body>
11 <h1>Tambah User</h1>
12 <form method="post" action="<?php echo url('/users'); ?>">
13 <table style="width: 500px;">
14 <tr><td>Nama</td><td><input type="text" name="name" /></td></tr>
15 <tr><td>Email</td><td><input type="text" name="email" /></td></tr>
16 <tr><td>Password</td><td><input type="password" name="password" /></td></tr>
17 <tr><td>Re-Password</td><td><input type="password" name="re-password" /></td></tr>
18 <tr><td>Phone</td><td><input type="text" name="phone" /></td></tr>
19 <tr><td>Address</td><td><textarea name="address"></textarea></td></tr>
20 </table>
21 <input type="submit" name="submit" value="Simpan" />
22 </form>
23 <p><a href="<?php echo url('/users'); ?>">Kembali</a></p>
24 </body>
25 </html>
```

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



Sehingga ketika pengguna mengklik tombol **Tambah User**, pengguna akan dilempar ke halaman berikut :

Tambah User

Nama: Jhon Due
Email: jhon.due@gmail.com
Password:
Re-Password:
Phone: 085678345655
Address: Jl. Pinang 2 No.8, Depok

Kita kembali ke UserController untuk memproses data yang diterima UserController ketika pengguna menekan tombol **Simpan**. Berikut kode untuk menyimpan entry user yang baru saja dimasukkan oleh pengguna.

```
public function store()
{
    $usr = new User();
    $usr->email = Input::get('email');
    $usr->name = Input::get('name');
    $usr->phone = Input::get('phone');
    $usr->address = Input::get('address');
    $usr->password = md5(Input::get('password'));
    $usr->save();

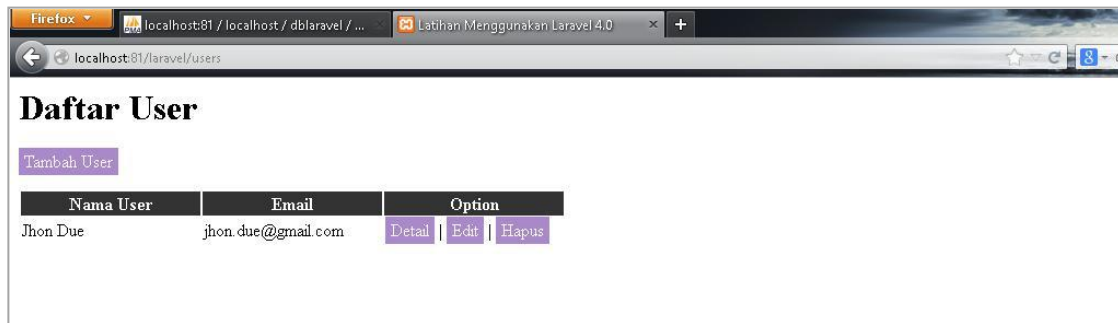
    return Redirect::to('users');
}
```

Kita membuat sebuah objek turunan dari model User dan dengan objek tersebut kita dapat mengakses atribut yang dimilikinya, sesuai dengan atribut yang dimiliki tabel users. Selanjutnya panggil setiap input yang diberikan pengguna untuk di-assing ke atribut yang bersesuaian. Setelah kita save, bawa pengguna kembali ke halaman yang berisi daftar user dengan perintah **Redirect::to('users')**. Hasilnya akan kelihatan seperti berikut:

Lisensi Dokumen:

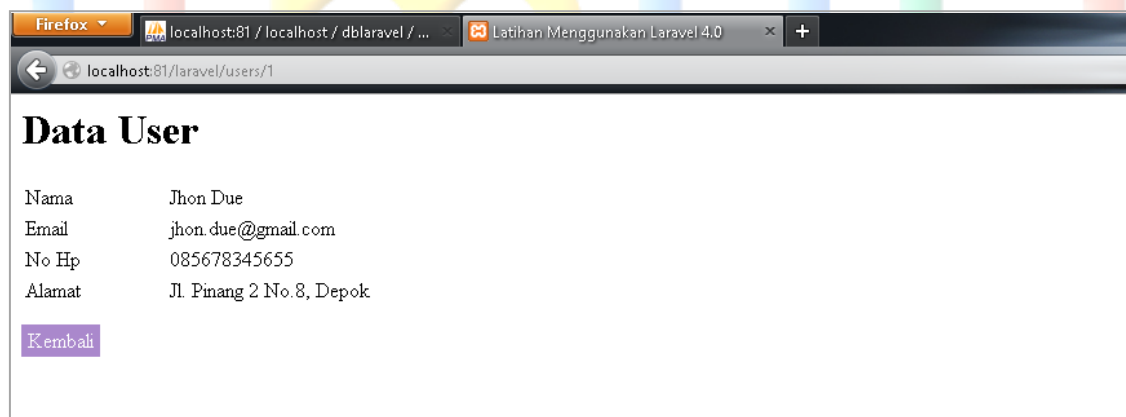
Copyright © 2008-2014 ilmuti.org

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



Sekarang kita memiliki tiga tombol baru, yaitu **Detail**, **Edit**, dan **Hapus**. Detail digunakan untuk melihat informasi secara detil dari suatu nama user. Edit untuk melakukan perubahan data dan Hapus untuk menghapus data user.

Langkah 8 : Menampilkan Detail User



Gambar diatas merupakan contoh halaman yang akan muncul ketika pengguna menekan tombol Detail pada halaman daftar user. Untuk melakukannya sangat mudah, kita cukup memanggil user dengan id bersangkutan melalui model dan melempar datanya ke view untuk menampilkan data.

1. Pengaturan pada UserController. Kita cukup memerintahkan model User untuk mencari record dengan id yang sudah ditentukan.

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



```
53 public function show($id)
54 {
55     $data['data_user'] = User::find($id);
56     return View::make('user/detail', $data);
57 }
58
```

2. Membuat view baru bernama *detail.php* lalu simpan pada folder *app/view/user*. Untuk menampilkan data yang ada pada *\$data_user*, kita dapat langsung memanggilnya sesuai dengan nama atributnya.

detail.php (htdocs) - Sublime Text 2

```
1 <html>
2 <head>
3     <title>Latihan Menggunakan Laravel 4.0</title>
4     <style type="text/css">
5         th{ width: 33%; text-align: center; background-color: #333; color: #fff;}
6         a{ padding: 3px 5px; background-color: #aa88cc; text-decoration: none;color:#fff;}
7         a:hover{background-color: #aa55cc;}
8     </style>
9 </head>
10 <body>
11     <h1>Data User</h1>
12     <table style="width: 500px;">
13         <tr><td>Nama</td><td><?php echo $data_user->name; ?></td></tr>
14         <tr><td>Email</td><td><?php echo $data_user->email; ?></td></tr>
15         <tr><td>No Hp</td><td><?php echo $data_user->phone; ?></td></tr>
16         <tr><td>Alamat</td><td><?php echo $data_user->address; ?></td></tr>
17     </table>
18     <p><a href="<?php echo url('/users'); ?>">Kembali</a></p>
19 </body>
20 </html>
```

Langkah 9 : Melakukan Update Data User

Firefox localhost:81 / localhost / dlaravel / ... Latihan Menggunakan Laravel 4.0

localhost:81 / laravel / users / 1 / edit

Edit Data User

Nama	<input type="text" value="Jhon Due"/>
Email	<input type="text" value="jhon.due@gmail.com"/>
Phone	<input type="text" value="085678345655"/>
Address	<input type="text" value="Jl. Pinang 2 No.8, Depok"/>

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



Hal yang sama juga terjadi ketika kita akan melakukan perubahan pada data user. Gambar di atas adalah contoh tampilan halaman untuk melakukan perubahan data user. Langkahnya lebih kurang sama seperti menambah user.

1. Fungsi edit pada UserController. Pada fungsi ini **kita ambil data user berdasarkan id nya**, lalu tampilkan halaman edit yang berisi data user.

```
65 public function edit($id)
66 {
67     $data['data_user'] = User::find($id);
68     return View::make('user/edit', $data);
69 }
```

2. Buat view baru dengan nama **edit.php** lalu simpan pada folder *app/view/user*. Tampilkan data yang ada pada `$data_user` dengan memanggilnya nama atributnya. Perlu diperhatikan untuk code yang berada pada lingkaran berwarna merah, bahwa saya menggunakan code Form laravel untuk mengakses fungsi update pada UserController. Ini kenapa? Karena jika anda lihat pada Route melalui perintah `php artisan route` melalui cmd Anda akan melihat bahwa url-nya sama dengan url untuk menampilkan detail user tapi memiliki method yang berbeda. Meskipun Anda menulis method PUT pada tab `<form>`, hal tersebut tidak akan berhasil. Oleh karena itu saya menggunakan code `form::open()` untuk melakukan hal tersebut.

```
UserController.php x list.php x detail.php x edit.php x add.php x
1 <html>
2 <head>
3 <title>Latihan Menggunakan Laravel 4.0</title>
4 <style type="text/css">
5     th{ width: 33%; text-align: center; background-color: #333; color: #fff;}
6     a{ padding: 3px 5px; background-color: #aa88cc; text-decoration: none; color: #fff;}
7     a:hover{ background-color: #aa55cc;}
8 </style>
9 </head>
10 <body>
11 <h1>Edit Data User</h1>
12 <?php
13 echo Form::open(array('action' => array('UserController@update', $data_user->id), 'method' => 'put',
14 'class' => 'form-horizontal', 'enctype' => "multipart/form-data"));
15 ?>
16 <table style="width: 500px;">
17 <tr><td>Nama</td><td><input type="text" name="name" value="<?php echo $data_user->name; ?>" /></td></tr>
18 <tr><td>Email</td><td><input type="text" name="email" value="<?php echo $data_user->email; ?>" /></td></tr>
19 <tr><td>Phone</td><td><input type="text" name="phone" value="<?php echo $data_user->phone; ?>" /></td></tr>
20 <tr><td>Address</td><td><textarea name="address"><?php echo $data_user->address; ?></textarea></td></tr>
21 </table>
22 <input type="submit" name="submit" value="Simpan" />
23 </form>
24 <p><a href="<?php echo url('/users'); ?>">Kembali</a></p>
25 </body>
26 </html>
```

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



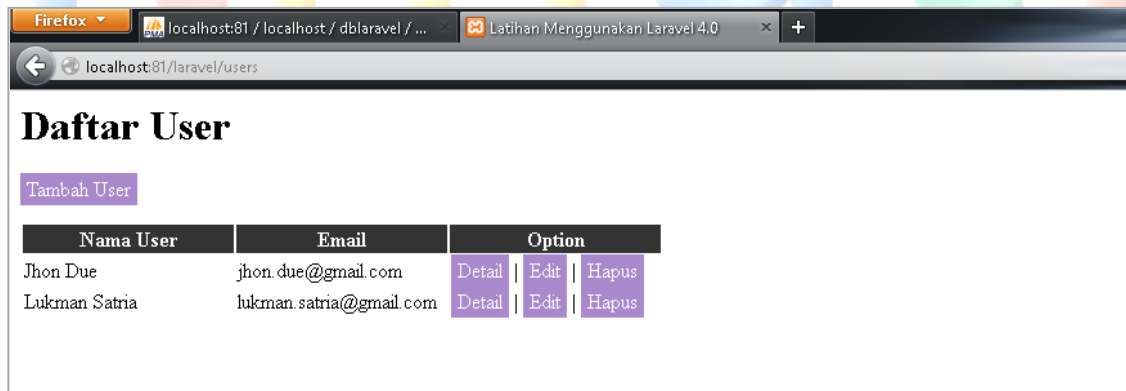
3. Pada fungsi update UserController kita lakukan perintah lebih kurang sama dengan yang ada pada fungsi store. Kita dapat hanya menulis code untuk atribut yang kita ubah saja.

```
public function update($id)
{
    $usr = User::find($id);
    $usr->email = Input::get('email');
    $usr->name = Input::get('name');
    $usr->phone = Input::get('phone');
    $usr->address = Input::get('address');
    $usr->save();

    return Redirect::to('users');
}
```

Langkah 10 : Menghapus Data User

Proses terakhir dari proses CRUD adalah Delete yaitu menghapus sebuah entri. Misalkan kita punya dua entri seperti gambar di bawah. Kita akan menghapus salah satu user.



Kita kembali ke daftar Route. Silahkan buka lagi cmd anda lalu jalankan perintah `php artisan route`. Dapat Anda lihat pada gambar di bawah bahwa untuk mengakses fungsi destroy dari UserController kita menggunakan route dengan **method DELETE** yang mana memiliki url sama dengan route `users.show` dan `users.update`. Hal

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



tersebut mengakibatkan kita tidak dapat menghapus data dengan mengakses url yang dimiliki oleh route `users.destroy` secara langsung.

```
C:\xampp\htdocs\laravel>php artisan routes
```

Domain	URI	Name	Action
GET	/users	users.index	UserController@index
GET	/users/create	users.create	UserController@create
POST	/users	users.store	UserController@store
GET	/users/{users}	users.show	UserController@show
GET	/users/{users}/edit	users.edit	UserController@edit
PUT	/users/{users}	users.update	UserController@update
PATCH	/users/{users}		UserController@update
DELETE	/users/{users}	users.destroy	UserController@destroy
GET	/		Closure

```
C:\xampp\htdocs\laravel>
```

Untuk itu, langkah yang harus kita lakukan adalah seperti berikut:

1. Buat route baru dengan nama `hapus_user` yang merujuk pada fungsi `destroy` pada `UserController`. Buka file `route.php` yang ada pada folder `app`. Tambahkan code berikut :

```
15  
16 Route::get('hapus_user/{id}', array('as' => 'delete_user', 'uses' => 'UserController@destroy'));
```

2. Buat perintah menghapus pada fungsi `destroy` `UserController`.

```
95 public function destroy($id)  
96 {  
97     $usr = User::find($id);  
98     $usr->delete();  
99  
100     return Redirect::to('users');  
101 }  
102
```

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



3. Tambahkan link *hapus_user* pada halaman daftar user seperti berikut.

```
20 // Menampilkan data untuk semua user yang ada
21 foreach ($daftar_user as $usr) {
22     echo '<tr><td>'. $usr->name. '</td><td>'. $usr->email. '</td>';
23
24     echo '<td><a href="'. url('/users/'. $usr->id). '">Detail</a> | <a href="'. url('/users/'. $usr->id. '/edit') .
25     echo '<a href="'. url('/hapus_user/'. $usr->id). '">Hapus</a></td></tr>';
26 }
```

Penutup

Berdasarkan tutorial di atas, saya berharap Anda mengerti dengan proses **create**, **read**, **update**, dan **delete** menggunakan Laravel. Sekarang silahkan Anda coba semua proses CRUD mulai dari membuat *user*, kemudian melihat informasi detail user tersebut, lakukan update, dan kemudian hapus. Jika Anda melakukannya dengan benar, semua proses CRUD tersebut akan berjalan lancar.

Tutorial di atas hanya sekedar mengetahui proses CRUD dengan sangat sederhana. Anda dapat melakukan percobaan dan eksplorasi sendiri sesuai kreativitas Anda. Pada dasarnya proses yang akan anda lewati sama dengan pembahasan kita di atas.

Anda dapat belajar dari dokumentasi yang disediakan oleh Laravel secara online pada websitenya. Setelah Anda paham dengan cara penggunaan MVC pada Laravel, mulailah masuk mendalami bagian lainnya pada Laravel seperti *Session*, *Login via Facebook* dan sebagainya. Komunitas yang membahas Laravel termasuk banyak seperti di **Stackoverflow**. Jika Anda mengalami kesulitan, Anda dapat merujuk ke sana.

Semoga artikel ini sangat membantu bagi Anda yang ingin belajar lebih dalam pada 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

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



Referensi

Ebook “Paduan Sederhana Menguasai Framework PHP Laravel” oleh Indra El Firasy, 2014.

<http://elfirasy.blogspot.com/2014/04/simple-crud-laravel-laravel-series.html#more>

<http://digitalmapia.com/membuat-crud-sederhana-pada-laravel-part-1/>

<http://inote.finzaiko.com/laravel/practice-simple-crud-laravel/>

<http://www.babastudio.com/blog/tutorial-laravel-contoh-insert-data/>

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