



MEMBUAT BLOG

LARAVEL 5.3 + MYSQL

DATABASE

KHUSUS PEMULA

Halo coder, ketemu lagi dengan saya orang paling ganteng sekecamatan. Kali ini saya akan berbagi ilmu tentang bagaimana cara membuat blog sederhana di Laravel 5.3.

Dalam tutorial ini akan di ajarkan step by step, yang dimulai dari penginstalan Laravel 5.3 di localhost sampai dengan Proses CRUD nya serta pembuatan Bootstrap templatanya. Jadi silahkan siapkan kopi dan fokus ke laptop masing-masing ya guys.

#Step 1 Intro

pada step ini tidak terlalu penting, hanya saja untuk mengetahui seperti apa nantinya aplikasi yang akan kita buat, langsung saja lihat video tutorial berikut <https://youtu.be/ctsr4A-5XEQ> dan <http://www.hc-kr.com/2016/09/laravel-5-tutorial-crud-operations-example.html>

#Step 2 Database & Migration

Pertama, silahkan buat database yang akan kita gunakan untuk penyimpanan data nantinya, buat database MySQL dengan nama **laravel_blog**, biar serasi.

kemudian buka cmd kamu dan untuk membuat project baru di Laravel ketikkan command berikut :

```
cd c:\server\htdocs  
  
composer create-project --prefer-dist  
laravel/laravel web  
  
cd web
```

proses penginstalan Laravel biasanya memakan waktu beberapa menit, jadi silahkan ngopi dulu sambil nunggu sampai selesai.

Jika sudah selesai, tinggal buka project (web) dengan text editor kesayangan kamu guys, kalau saya menggunakan Atom Text Editor.

Ada beberapa hal yang penting kita atur sebelum melanjutkan tutorial ini.

membuat koneksi Ke database

untuk membuat koneksi ke database, cukup dengan mengatur pada file .ENV seperti dibawah ini.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel_blog
DB_USERNAME=root
DB_PASSWORD=password_kamu
```

Migrasi Database

Untuk membuat migrasi di Laravel 5.3, langsung saja tuliskan command berikut :

```
php artisan make:migration create_blog_table
```

Kalau sudah berhasil, file migrasi kamu akan berada di **database\migrations\2016_09_13_113838_crate_blog_table.php**

Kita akan menambahkan function untuk membuat table dan column secara otomatis, tuliskan code pada file migrasi kamu.

```
<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateBlogTable extends Migration
{
    /**
     * Run the migrations.
     */
```

```
* @return void
*/
public function up()
{
    Schema::create('blog_post', function
(Blueprint $table) {
        $table->increments('id');
        $table->string('title');
        $table->string('description');
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::drop('blog_post');
}
}
```

Terakhir silahkan simpan, dan mari eksekusi dengan command

```
php artisan migrate
```

Video tutorialnya dimari <https://youtu.be/afmcqHTCCQk> dan full source codenya dimari <http://www.hc-kr.com/2016/09/laravel-5-blog-tutorial-database-migration.html>

#Step 3 Restful Controllers & Routing System

Selanjutnya, untuk membuat controller di laravel 5.3, tuliskan command berikut

```
php artisan make:controller BlogController --resource
```

File controller kamu sudah berhasil dibuat dan tersimpan di folder **app\Http\Controllers\BlogController.php**

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Requests;

class BlogController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return Illuminate\Http\Response
     */
    public function index()
    {
        //
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return Illuminate\Http\Response
     */
}
```

```

public function create()
{
    //

}

/**
 * Store a newly created resource in storage.
 *
 * @param IlluminateHttpRequest $request
 * @return IlluminateHttpResponse
 */
public function store(Request $request)
{
    //

}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return IlluminateHttpResponse
 */
public function show($id)
{
    //

}

/**
 * Show the form for editing the specified
resource.
 *
 * @param int $id
 * @return IlluminateHttpResponse
 */
public function edit($id)
{
    //

```



```
| Web Routes
|-----
|
| This file is where you may define all of the
| routes that are handled
| by your application. Just tell Laravel the URIs
| it should respond
| to using a Closure or controller method. Build
| something great!
|
| */
```

```
Route::get('/', function () {
    return view('welcome');
});
```

// we will add new route into middleware group

```
Route::group(['middleware' => ['web']], function()
{
    Route::resource('blog', 'BlogController');
});
```

Video tutorial dimari https://youtu.be/eHNRpYB_wJk dan source code dimari <http://www.hc-kr.com/2016/09/laravel-5-blog-tutorial-database-migration.html>

#Step 4 Model Eloquent & Displaying Data from Database

Pertama, kita akan membuat Model baru dengan mengetikkan command berikut

```
php artisan make:model Blog
```


Model yang baru kamu buat akan di simpan di App\Blog.php

```
<?php
```

```
namespace App;
```

```
use Illuminate\Database\Eloquent\Model ;
```

```
class Blog extends Model
```

```
{
```

```
    protected $table = 'blog_post' ;
```

```
}
```

Selanjutnya, untuk mengambil semua data dari database, kita kembali ke controller (BlogController.php) dan tuliskan code berikut

Tambahkan **namespaces** di header controller kita

```
use App\Blog;
```

pada controller tambahkan function ini

```
public function index()
```

```
{
```

```
    // we will create index function
```

```
    // we need to show all data from "blog"
```

```
table
```

```
    $blogs = Blog::all();
```

```
    // show data to our view
```

```
    return view('blog.index', ['blogs' =>
```

```
    $blogs]);
```

```
}
```

Next, untuk menampilkannya di browser, kita akan membuat view baru yang disimpan di resources\views\, silahkan buat folder baru dengan nama blog dalam folder resources\views\ tersebut. Jadi kita membuat view baru dengan nama index.blade.php dan akan tersimpan di resources\views\blog\index.blade.php

```

<h1>My First Blog</h1>
@foreach ($blogs as $blog)
    <h2>{{ $blog->title }}</h2>
    <p>{{ $blog->description }}</p>
    <hr>
@endforeach

```

Sekarang kamu bisa membuka blog pertama kamu dengan laravel di <http://localhost:8080/blog>

Video tutorialnya dimari <https://youtu.be/GvLvGbkUbVs> dan full source code di mari <http://www.hc-kr.com/2016/09/part4-laravel-5-blog-tutorial-model.html>

#Step 5 Insert Data, Validation, Redirect Page and Flash Message

Selanjutnya kita akan membuat function baru di controller (**BlogController.php**)

```

public function create()
{
    // we will return to our new views
    return view('blog.create');
}

/**
 * Store a newly created resource in storage.
 *
 * @param IlluminateHttpRequest $request
 * @return IlluminateHttpResponse
 */
public function store(Request $request)
{
    // we will create validation function here

```

```

$this->validate($request, [
    'title' => 'required',
    'description' => 'required',
]);

$blog = new Blog;
$blog->title = $request->title;
$blog->description = $request->description;
// save all data
$blog->save();
//redirect page after save data
return redirect('blog')-
>with('message', 'data hasbeen updated!');
}

```

Selanjutnya buat View baru dengan nama create.blade.php yang disimpan di resources\views\blog\create.blade.php

```

<!-- @if (count($errors) > 0)
    <ul>
        @foreach ($errors->all() as $error )
            <li>{{ $error }}</li>
        @endforeach
    </ul>
@endif -->

<h2>Add new post</h2>
<form class="" action="/blog" method="post">
    <input type="text" name="title" value=""
placeholder="this is title"><br>
    {{ ($errors->has('title')) ? $errors-
>first('title') : '' }} <br>

    <textarea name="description" rows="8" cols="40"
placeholder="this is description"></textarea><br>
    {{ ($errors->has('description')) ? $errors-
>first('description') : '' }} <br>

```

```

<input type="hidden" name="_token" value="{{
csrf_token() }}">
<input type="submit" name="name" value="post">
</form>

```

Tambahkan Session di index.blade.php

```
{{ Session::get('message') }}
```

```

<h1>My First Blog</h1>
@foreach ($blogs as $blog)
    <h2>{{ $blog->title }}</h2>
    <p>{{ $blog->description }}</p>
    <hr>
@endforeach

```

Video tutorialnya dimari <https://youtu.be/0X6hkeR91ko> dan source codenya dimari <http://www.hc-kr.com/2016/09/laravel-5-create-blog-insert-validation-redirect-page.html>

#Step 6 membuat Single Page and 404 Page

pada controller (**BlogController.php**) kita, tambahkan function berikut

```

public function show($id)
{
    $blog = Blog::find($id);

    // return to 404 page
    if(!$blog){
        abort(404);
    }

    // display the article to single page

```

```

        return view('blog.detail') -
>with('blog', $blog);
    }

```

Dan kemudian buatlah view baru dengan nama detail.blade.php

```

<h1>Detail Page</h1>
<h2>{{ $blog->title }}</h2>
<p>
    {{ $blog->description }}
</p>
<br>
<a href="/blog">Back to Home</a>

```

Kemudian edit kembali index.blade.php kita sebelumnya

```

{{ Session::get('message') }}

<h1>My First Blog</h1>
@foreach ($blogs as $blog)
    <h2><a href="/blog/{{ $blog->id }}">{{ $blog->
>title }}</a></h2>
    <p>{{ $blog->description }}</p>
    <hr>
@endforeach

```

Video tutorialnya dimari <https://youtu.be/Or4e1okgFuc> dan full source codenya dimari <http://www.hc-kr.com/2016/09/laravel-5-create-blog-insert-validation-redirect-page.html>

#Step 7 Update Database Recorded

Tambah edit dan update function di Controller kita
(**BlogController.php**)

edit function

```
public function edit($id)
{
    // edit function here
    $blog = Blog::find($id);

    // return to 404 page
    if(!$blog){
        abort(404);
    }

    // display the article to single page
    return view('blog.edit')-
>with('blog', $blog);
}
```

update function

```
public function update(Request $request, $id)
{
    // we will create validation function here
    $this->validate($request, [
        'title' => 'required',
        'description' => 'required',
    ]);

    $blog = Blog::find($id);
```

```

        $blog->title = $request->title;
        $blog->description = $request->description;
        // save all data
        $blog->save();
        //redirect page after save data
        return redirect('blog')->with('message', 'data
hasbeen edited!');
    }

```

Selanjutnya buat view baru dengan nama edit.blade.php

```

<h2>Edit Article Post</h2>
<form class="" action="/blog/{{ $blog->id }}"
method="post">
    <input type="text" name="title" value="{{ $blog-
>title }}" placeholder="this is title"><br>
    {{ ($errors->has('title')) ? $errors-
>first('title') : '' }} <br>

    <textarea name="description" rows="8" cols="40"
placeholder="this is description">{{ $blog-
>description }}</textarea><br>
    {{ ($errors->has('description')) ? $errors-
>first('description') : '' }} <br>

    <input type="hidden" name="_method" value="put">
    <input type="hidden" name="_token" value="{{
csrf_token() }}">
    <input type="submit" name="name" value="edit">
</form>

```

Edit kembali file index.blade.php kita

```

{{ Session::get('message') }}

<h1>My First Blog</h1>

```

```

@foreach ($blogs as $blog)
    <h2><a href="/blog/{{ $blog->id }}">{{ $blog->title }}</a></h2>
    <p>{{ $blog->description }}</p>
    <a href="/blog/{{ $blog->id }}/edit">Edit this Post</a>
    <hr>
@endforeach

```

Video tutorialnya dimari <https://youtu.be/t2z2k2EJiec> dan source codenya dimari <http://www.hc-kr.com/2016/09/laravel-5-blog-tutorial-update-database.html>

#Step 8 Delete data with Eloquent Syntax

Tambahkan function destroy di controller (**BlogController.php**)

```

public function destroy($id)
{
    $blog = Blog::find($id);
    $blog->delete();
    return redirect('blog')-
>with('message', 'data hasbeen deleted!');
}

```

Selanjutnya edit file index.blade.php yang ada di folder views

```

{{ Session::get('message') }}

<h1>My First Blog</h1>
@foreach ($blogs as $blog)
    <h2><a href="/blog/{{ $blog->id }}">{{ $blog->title }}</a></h2>
    {{ date('F d, Y', strtotime($blog->created_at)) }}

    <p>{{ $blog->description }}</p>

```



```

    <a href="/blog/{{ $blog->id }}/edit">Edit this
Post</a><br>
    <form class="" action="/blog/{{ $blog->id }}"
method="post">
        <input type="hidden" name="_method"
value="delete">
        <input type="hidden" name="_token" value="{{
csrf_token() }}">
        <input type="submit" name="name"
value="delete">
    </form>
    <hr>
@endforeach

```

Video tutorialnya dimari <https://youtu.be/vgHuJFU0CyE> dan Full source codenya dimari <http://www.hc-kr.com/2016/09/laravel-5-blog-tutorial-deleting-data-eloquent.html>

#Step 9 Pagination with Eloquent & Query Builder

Pada step ini, ada dua macam method yang bisa kita gunakan untuk membuat pagination di laravel 5.3.

Pagination dengan Query Builder

pada controller (BlogController.php)

```

public function index()
{
    // we will create index function
    // we need to show all data from "blog"
    table
        // $blogs = Blog::all();
        // first, pagination using query builder

```

```

        $blogs = DB::table('blog_post')-
>paginate(2);

        // pagination using Eloquent
        // $blogs = Blog::paginate(2);

        // show data to our view
        return view('blog.index', ['blogs' =>
$blogs]);
    }

```

Pagination dengan Eloquent

```

public function index()
{
    // we will create index function
    // we need to show all data from "blog"
    table
        // $blogs = Blog::all();
        // first, pagination using query builder
        // $blogs = DB::table('blog_post')-
>paginate(2);

        // pagination using Eloquent
        $blogs = Blog::paginate(2);

        // show data to our view
        return view('blog.index', ['blogs' =>
$blogs]);
    }

```

Kamu bisa memilih salah satu method diatas.

dan pada **index.blade.php** yang ada di folder views,

```
{{ Session::get('message') }}
```

```
<h1>My First Blog</h1>
```

```

@foreach ($blogs as $blog)
    <h2><a href="/blog/{{ $blog->id }}">{{ $blog->title }}</a></h2>
    {{ date('F d, Y', strtotime($blog->created_at)) }}

    <p>{{ $blog->description }}</p>
    <a href="/blog/{{ $blog->id }}/edit">Edit this Post</a><br>
    <form class="" action="/blog/{{ $blog->id }}" method="post">
        <input type="hidden" name="_method" value="delete">
        <input type="hidden" name="_token" value="{{ csrf_token() }}">
        <input type="submit" name="name" value="delete">
    </form>
    <hr>
@endforeach

{!! $blogs->links() !!}

```

Video tutorialnya dimari <https://youtu.be/mwb9V4E0wm4> dan full source codenya dimari <http://www.hc-kr.com/2016/09/laravel-5-blog-tutorial-pagination-eloquent.html>

#Step 10 Menambahkan Bootstrap Theme

untuk step ini, langsung saja kamu lihat video tutorialnya dibawah ini

Video tutorialnya dimari <https://youtu.be/2n3q2oof084> dan full source codenya dimari <http://www.hc-kr.com/2016/09/laravel-5-blog-tutorial-adding-bootstrap-theme.html>

Sekian tutorial sederhananya.

Informasi lebih lengkapnya tentang belajar Laravel 5.3 silahkan langsung saja ke blog kami <http://www.hc-kr.com/p/create-blog-using-laravel-5-for.html>