



**LARAVEL TUTORIAL
FOR
SMK NEGERI 4 BANDUNG**

Integrate AdminLTE & Make Simple CRUD
with Laravel

Table of Contents

Table of Contents.....	2
Installing Laravel.....	3
Install melalui Composer.....	4
Setup Composer.....	5
Install Laravel menggunakan cara Copy Fresh Laravel.....	6
Jalankan Laravel pada Web Browser.....	6
Struktur Folder Laravel	7
Integrasi Laravel dengan Template	9
CRUD – Read Data.....	16
Membuat Tabel dengan Migration	16
Membuat Dummy Data menggunakan Database Seed	19
Menampilkan data di views.....	21
CRUD – Create Data	25
Menambahkan Feedback Sukses atau Gagal	28
Validasi Form	31
CRUD – Updating Data	33
CRUD – Delete Data	37

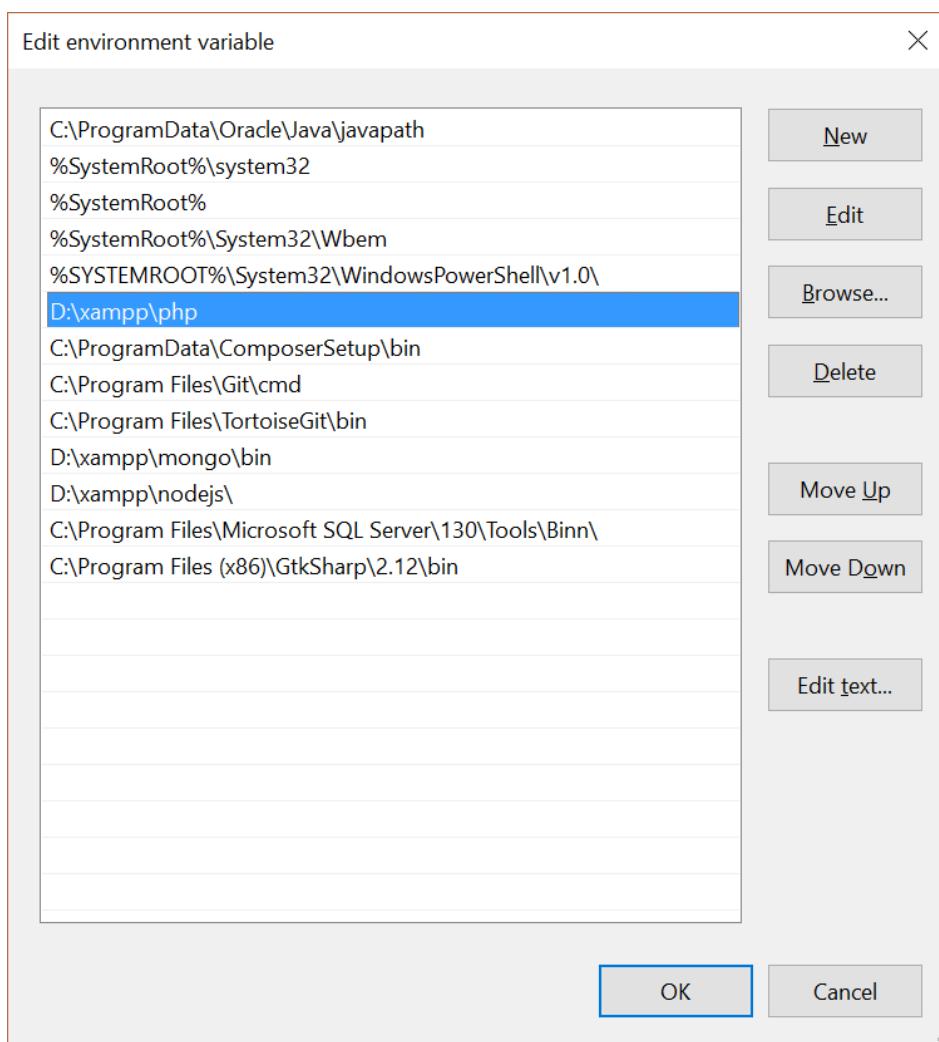
Installing Laravel

Pada pembahasan kali ini, saya akan menjelaskan bagaimana cara menginstall Laravel pada perangkat kalian. Laravel dapat diinstall melalui dua cara, yaitu install dari Composer atau copy fresh laravel.

Kebutuhan Sistem untuk mengikuti tutorial ini:

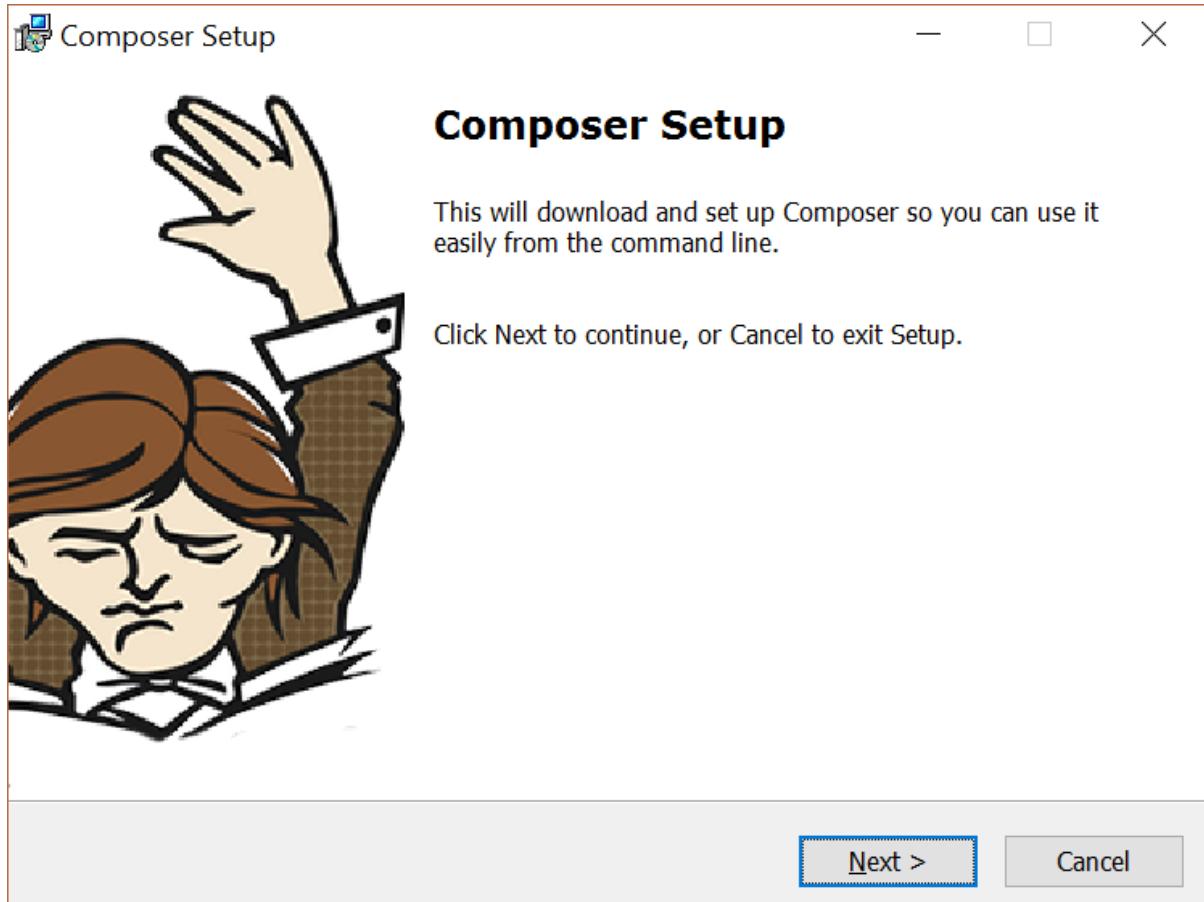
- Web Browser
- PHP 5.6 atau lebih tinggi
- MySQL Server

Pastikan juga kalian sudah memasukan PHP kedalam Path pada System Environment Variables.



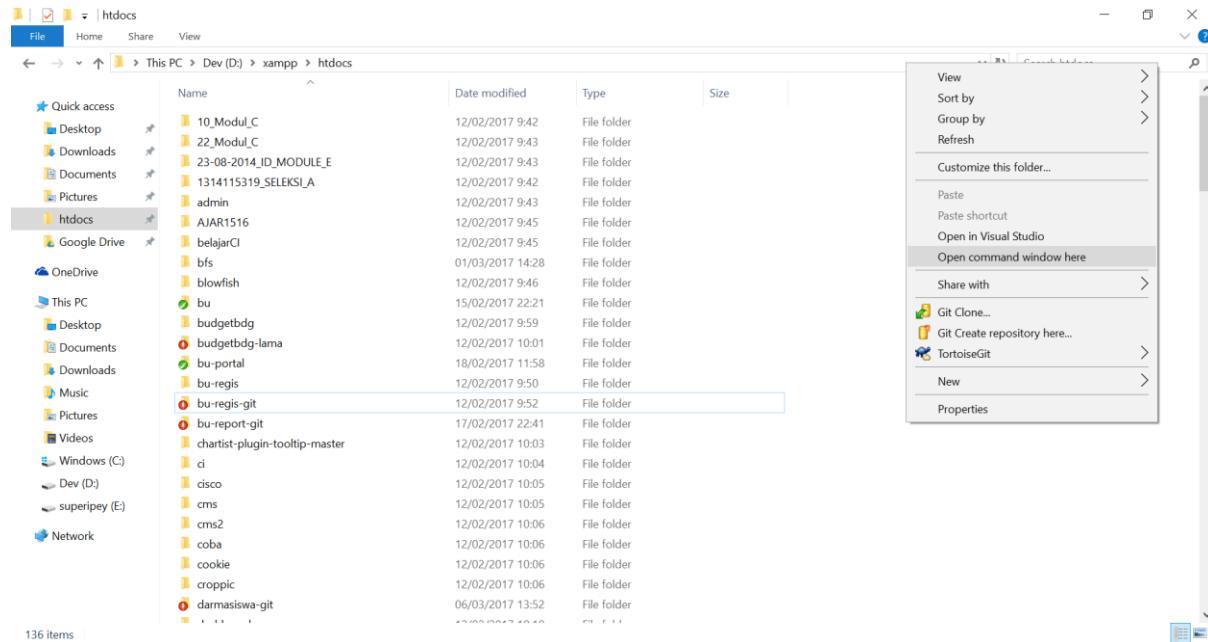
Install melalui Composer

Untuk melakukan instalasi Laravel melalui Composer, install terlebih dahulu Composer, bisa didapatkan dari <https://getcomposer.org/Composer-Setup.exe>



Setup Composer

Masuk ke folder htdocs kalian, tekan Shift + Klik Kanan pada sembarang tempat, pilih *Open command windows here* untuk membuk command prompt pada folder htdocs.



Ketik *command* berikut pada *command prompt* untuk membuat project baru Laravel

5.3:

```
composer create-project --prefer-dist laravel/laravel belajarLaravel "5.3.*"
```

Proses instalasi Laravel akan berlangsung. Pastikan perangkat kamu terkoneksi dengan internet karena composer akan mengambil package-package laravel nya secara online.

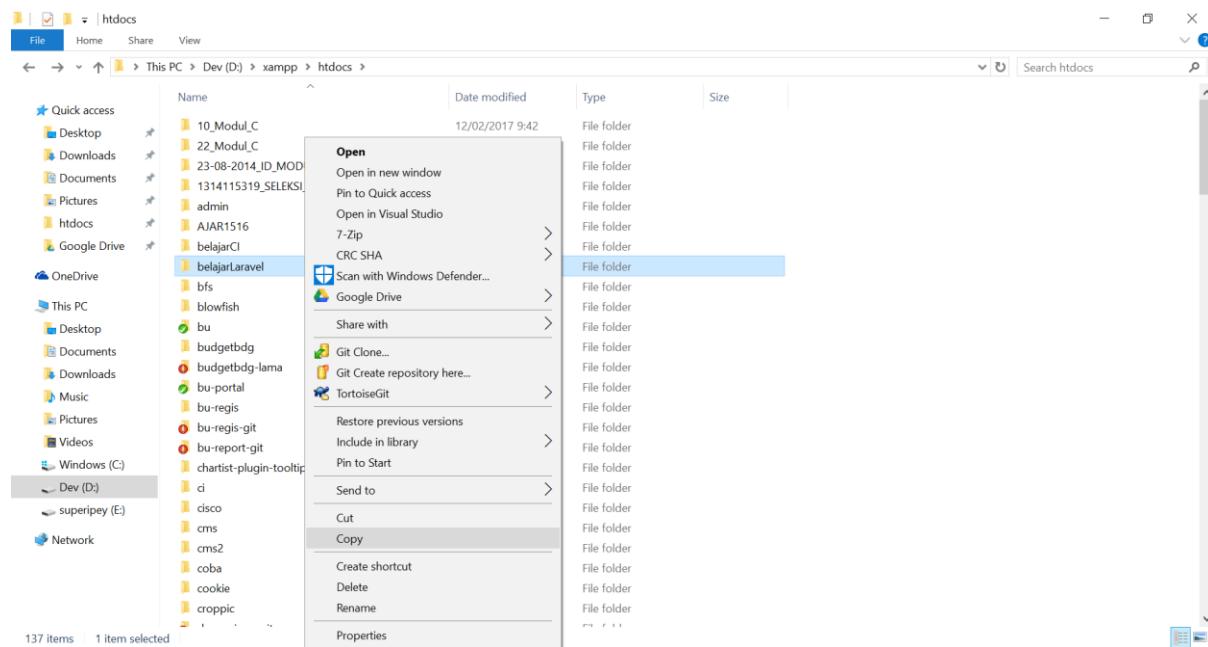
Laravel akan di install pada folder **belajarLaravel**.

A screenshot of a Windows Command Prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The command 'composer create-project --prefer-dist laravel/laravel belajarLaravel "5.3.*"' is being typed. The output shows the process of installing Laravel 5.3.30 from cache, creating a new project named 'belajarLaravel', and updating dependencies. The command prompt window has a black background with white text.

Proses ini akan memakan waktu yang agak lama tergantung dari koneksi internet kalian.

Install Laravel menggunakan cara Copy Fresh Laravel.

Cara ini bisa digunakan untuk mendapatkan fresh laravel hasil install dari composer orang lain (Sudah ada folder vendor nya). Caranya copy saja satu folder fresh install laravel dari orang lain ke perangkat kamu.



Jalankan Laravel pada Web Browser

Akses <http://localhost/belajarLaravel/public>, jika instalasi laravel berhasil maka akan muncul tampilan seperti dibawah ini:



Laravel

DOCUMENTATION LARACASTS NEWS FORGE GITHUB

Struktur Folder Laravel

Pada bagian ini akan dibahas mengenai struktur foldering dari Laravel 5.3.

View			
PC > Dev (D:) > xampp > htdocs > belajarLaravel >			
Name	Date modified	Type	Size
app	12/03/2017 11:10	File folder	
bootstrap	12/03/2017 11:10	File folder	
config	12/03/2017 11:10	File folder	
database	12/03/2017 11:10	File folder	
public	12/03/2017 11:10	File folder	
resources	12/03/2017 11:10	File folder	
routes	12/03/2017 11:10	File folder	
storage	12/03/2017 11:10	File folder	
tests	12/03/2017 11:10	File folder	
vendor	12/03/2017 11:19	File folder	
.env	12/03/2017 11:19	ENV File	1 KB
.env.example	12/03/2017 11:10	EXAMPLE File	1 KB
.gitattributes	12/03/2017 11:10	Text Document	1 KB
.gitignore	12/03/2017 11:10	Text Document	1 KB
artisan	12/03/2017 11:10	File	2 KB
composer.json	12/03/2017 11:10	JSON File	2 KB
composer.lock	12/03/2017 11:18	LOCK File	123 KB
gulpfile.js	12/03/2017 11:10	JavaScript File	1 KB
package.json	12/03/2017 11:10	JSON File	1 KB
phpunit.xml	12/03/2017 11:10	XML Document	1 KB
readme.md	12/03/2017 11:10	MD File	3 KB
server.php	12/03/2017 11:10	PHP File	1 KB

- **Folder app**

Berisikan aplikasi kalian, tempat menyimpan Controller, Model, Middleware

- **Folder config**

Berisikan file-file config dari framework laravel ini. Mulai dari koneksi database, authentikasi, dan lainnya

- **Folder public**

Berisikan file index, dan tempat menyimpan file-file asset web kalian.

- **Folder resources**

Tempat menyimpan views kalian

- Folder routes

Tempat menyimpan file route web kalian.

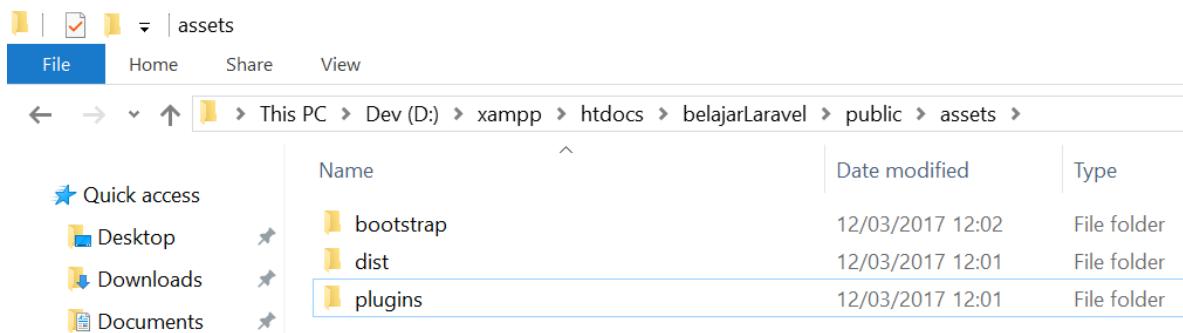
- File artisan

Fitur dari Laravel untuk memudahkan kita membuat file-file seperti controller, model, dan lainnya.

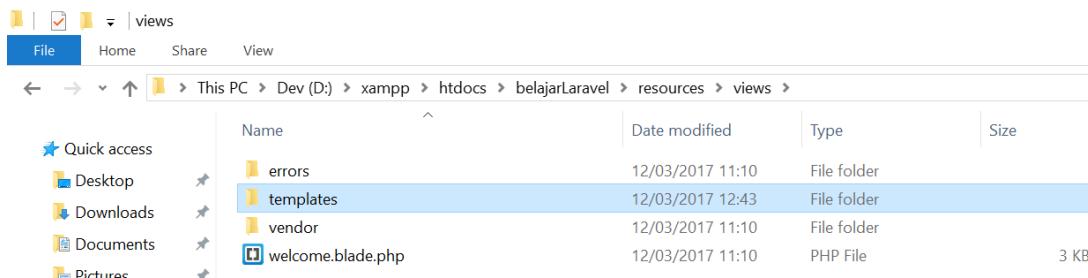
Integrasi Laravel dengan Template

Pembahasan mengenai fungsi-fungsi dari framework laravel ini akan dijelaskan bersamaan dengan integrase template. Ikuti langkah-langkah dibawah ini dengan baik. Persiapkan template **AdminLTE 2.3.11** yang bisa didapatkan dari alamat <https://codeload.github.com/alamasaeed2010/AdminLTE/zip/v2.3.11>

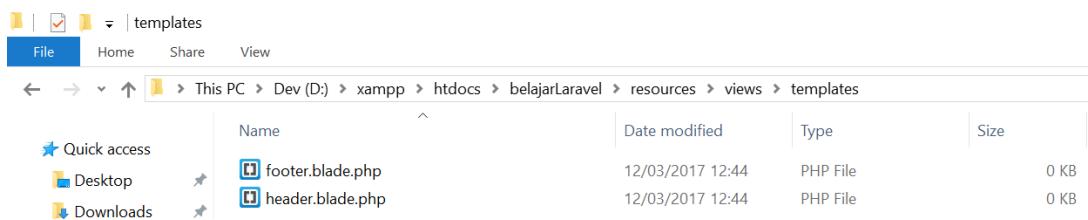
1. Buat folder **assets** di folder **public** laravel. Copy folder **dist**, **bootstrap** dan **plugins** dari template **AdminLTE** ke folder tersebut.



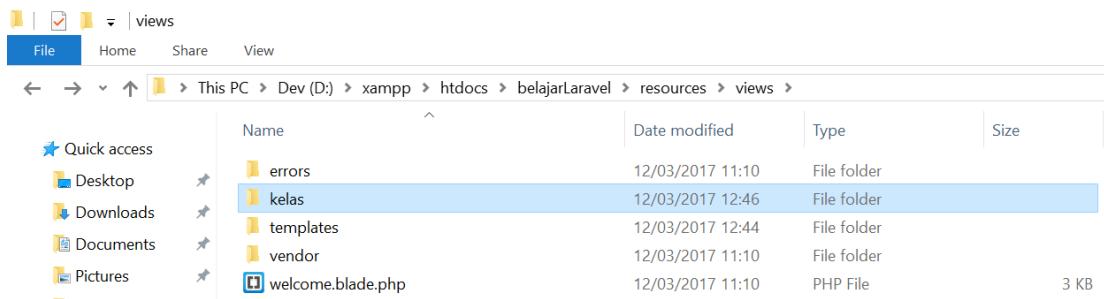
2. Buat folder baru dengan nama **templates** pada folder **resources/views** di laravel kamu.



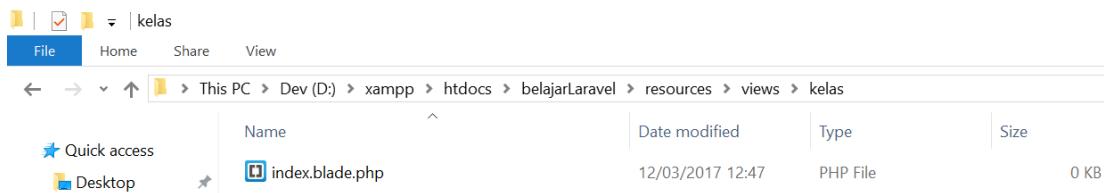
3. Didalam folder templates, buat 2 file dengan nama **header.blade.php** dan **footer.blade.php**



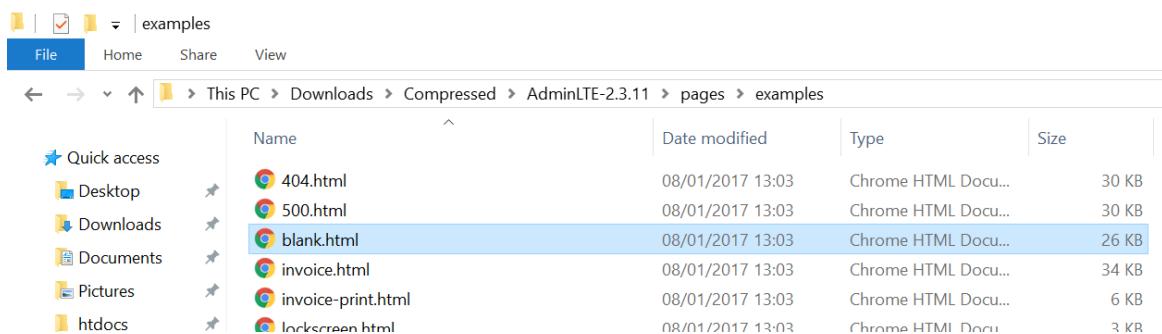
4. Buat folder baru dengan nama **kelas** pada folder **resources/views** di laravel kamu.



5. Didalam folder **kelas**, buat file dengan nama **index.blade.php**



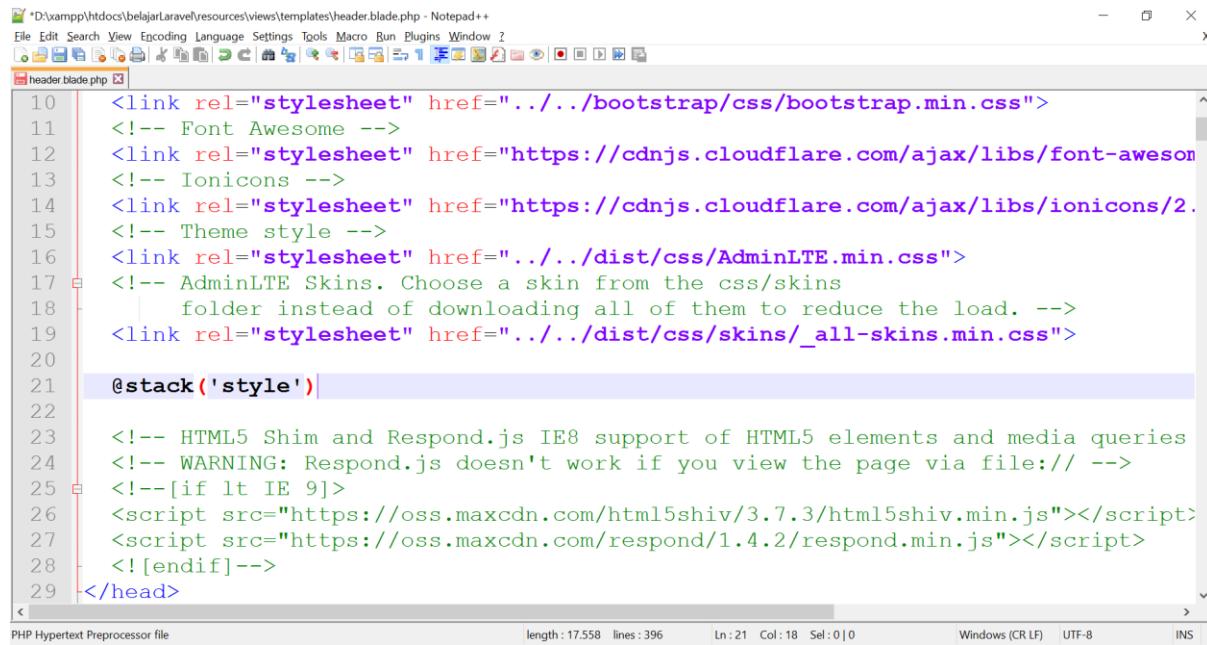
6. Buka folder **AdminLTE-2.3.11** kamu, kemudian cari file **blank.html** di folder **pages/examples**



7. Buka file tersebut dengan editor kesayanganmu. Lakukan langkah-langkah berikut:

- Copy Line 1 s.d 391, paste di file **templates/header.blade.php**
- Copy line 392 s.d 432, paste di file **index.blade.php**
- Copy line 433 s.d 654, paste di file **templates/footer.blade.php**

8. Pada file **templates/header.blade.php** baris 21, tambahkan:



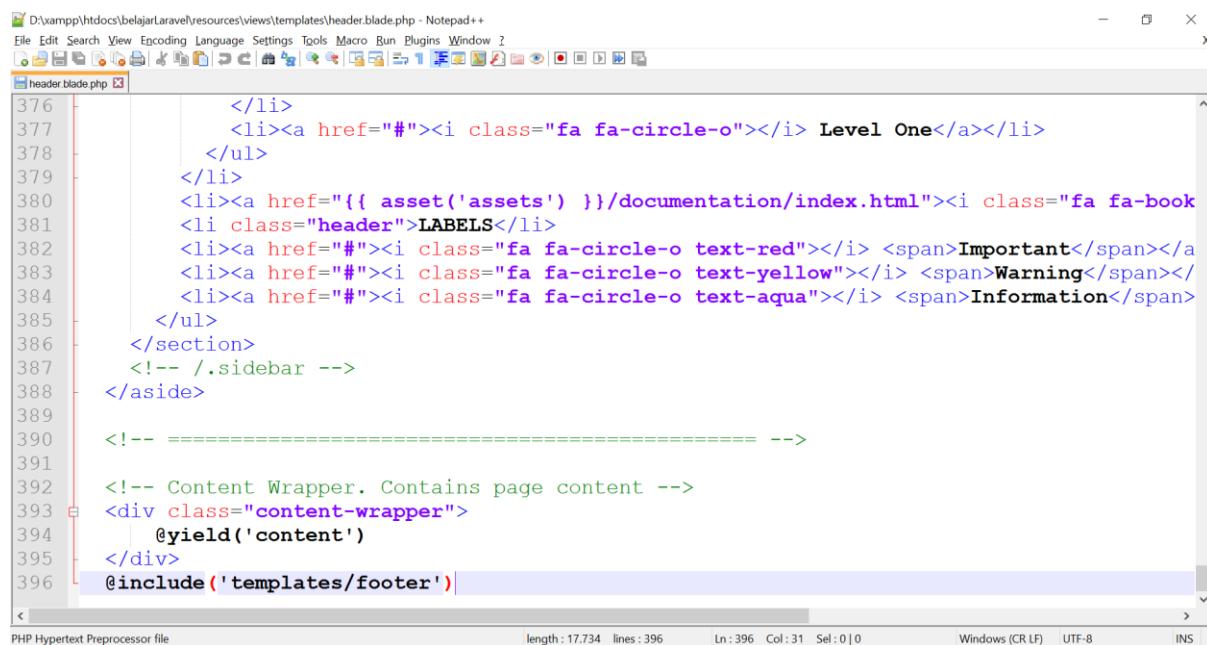
```
* D:\xampp\htdocs\belajarLaravel\resources\views\templates\header.blade.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 2
header blade.php
10 | <link rel="stylesheet" href="../../bootstrap/css/bootstrap.min.css">
11 | <!-- Font Awesome -->
12 | <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
13 | <!-- Ionicons -->
14 | <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/ionicons/2.0.1/css/ionicons.min.css">
15 | <!-- Theme style -->
16 | <link rel="stylesheet" href="../../dist/css/AdminLTE.min.css">
17 | <!-- AdminLTE Skins. Choose a skin from the css/skins
18 |   folder instead of downloading all of them to reduce the load. -->
19 | <link rel="stylesheet" href="../../dist/css/skins/_all-skins.min.css">
20 |
21 | @stack('style')
22 |
23 | <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries
24 | <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
25 | <!--[if lt IE 9]>
26 | <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
27 | <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
28 | <![endif]-->
29 | </head>
```

PHP Hypertext Preprocessor file length : 17.558 lines : 396 Ln : 21 Col : 18 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

Notes:

- @stack digunakan untuk menyimpan potongan script dari hasil @push

Masih pada file **templates/header.blade.php** baris terakhir, tambahkan 2 baris sintaks berikut:



```
* D:\xampp\htdocs\belajarLaravel\resources\views\templates\header.blade.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 2
header blade.php
376 |     </li>
377 |     <li><a href="#"><i class="fa fa-circle-o"></i> Level One</a></li>
378 |   </ul>
379 | </li>
380 | <li><a href="{{ asset('assets') }}/documentation/index.html"><i class="fa fa-book"></i> LABELS</a></li>
381 | <li class="header">LABELS</li>
382 | <li><a href="#"><i class="fa fa-circle-o text-red"></i> <span>Important</span></a>
383 | <li><a href="#"><i class="fa fa-circle-o text-yellow"></i> <span>Warning</span></a>
384 | <li><a href="#"><i class="fa fa-circle-o text-aqua"></i> <span>Information</span></a>
385 |   </ul>
386 | </section>
387 | <!-- /.sidebar -->
388 | </aside>
389 |
390 | <!-- -----
391 | <!-- Content Wrapper. Contains page content -->
392 | <div class="content-wrapper">
393 |   @yield('content')
394 | </div>
395 | @include('templates/footer')
```

PHP Hypertext Preprocessor file length : 17.734 lines : 396 Ln : 396 Col : 31 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

Notes:

- @yield digunakan untuk menyimpan potongan script dari hasil @section
- @include digunakan untuk menyisipkan file lain

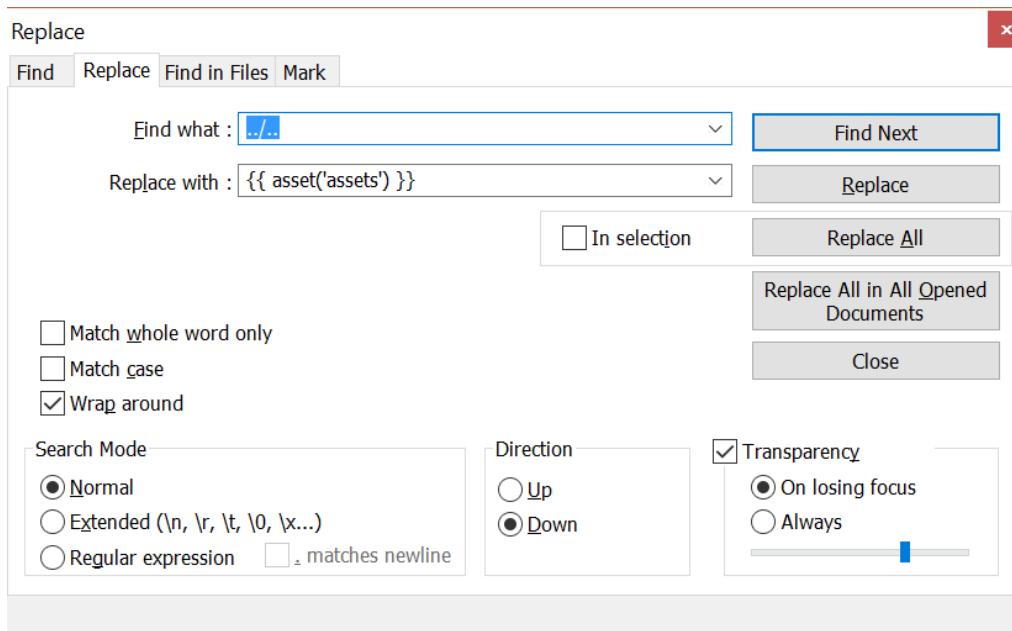
9. Buka file **kelas/index.blade.php**, tambahkan pada baris pertama dan terakhir menjadi seperti pada gambar ini:

```
D:\xampp\htdocs\belajarLaravel\resources\views\kelas\index.blade.php - Notepad+
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
index blade.php
1 @extends('templates/header')
2
3 @section('content')
4 <!-- Content Header (Page header) -->
5 <section class="content-header">
6 <h1>
7     Blank page
8     <small>it all starts here</small>
9 </h1>
10
11     <div class="box-body">
12         Start creating your amazing application!
13     </div>
14     <!-- /.box-body -->
15     <div class="box-footer">
16         Footer
17     </div>
18     <!-- /.box-footer-->
19     </div>
20     <!-- /.box -->
21
22     </section>
23     <!-- /.content -->
24
25 @endsection
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
99 PHP Hypertext Preprocessor file length : 1.385 lines : 45 Ln : 45 Col : 12 Sel : 0 | 0 Windows (CR LF) UTF-8 INS
```

Notes:

- @extends digunakan untuk melakukan extends ke file yang dituju
- @section digunakan untuk menandakan block section yang nanti akan disimpan pada bagian @yield

10. Pada file **templates/header.blade.php** lakukan **Find and Replace** untuk mengganti semua link asset ke folder asset yang sudah kita copy kan tadi.



```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <title>AdminLTE 2 | Blank Page</title>
7   <!-- Tell the browser to be responsive to screen width -->
8   <meta content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no" 
9   <!-- Bootstrap 3.3.6 -->
10  <link rel="stylesheet" href="{{ asset('assets') }}/bootstrap/css/bootstrap.min.css">
11  <!-- Font Awesome -->
12  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.5.0/
13  <!-- Ionicons -->
14  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/ionicons/2.0.1/css/
15  <!-- Theme style -->
16  <link rel="stylesheet" href="{{ asset('assets') }}/dist/css/AdminLTE.min.css">
17  <!-- AdminLTE Skins. Choose a skin from the css/skins
18  |   folder instead of downloading all of them to reduce the load. -->
19  <link rel="stylesheet" href="{{ asset('assets') }}/dist/css/skins/_all-skins.min.css">
20
21 @stack('style')
22

```

PHP Hypertext Preprocessor file length : 17.734 lines : 396 Ln : 10 Col : 53 Sel : 21 | 1 Windows (CR LF) UTF-8 INS

Lakukan langkah yang sama untuk file **templates/footer.blade.php**

Notes:

- {{ }} adalah sintaks blade pengganti echo
- Fungsi **asset** digunakan untuk menciptakan url **http://localhost/folderLaravel/public** dan parameter nya digunakan untuk masuk kedalam folder yang ada di folder public

11. Buka Command Prompt (dengan cara tekan Shift + Klik Kanan di sembarang tempat, pilih Open windows command windows here) di folder belajarLaravel, buat Controller baru menggunakan **artisan** dengan syntax berikut:

```
> php artisan make:controller KelasController
```

```

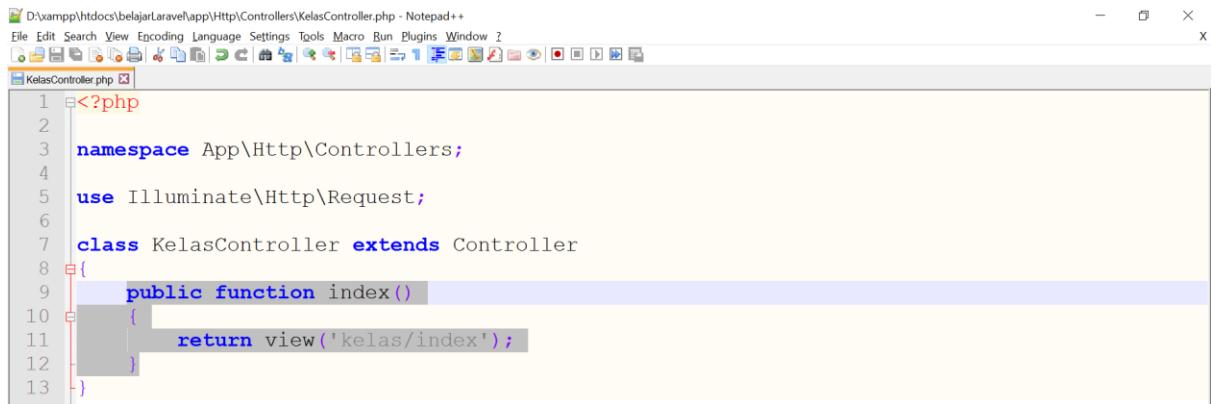
C:\WINDOWS\system32\cmd.exe
D:\xampp\htdocs\belajarLaravel>php artisan make:controller KelasController
Controller created successfully.

D:\xampp\htdocs\belajarLaravel>

```

Jika berhasil maka akan muncul file **KelasController.php** di folder **app/Http/Controllers**

12. Buka file **app/Http/Controllers/KelasController.php**, tambahkan sintaks berikut:



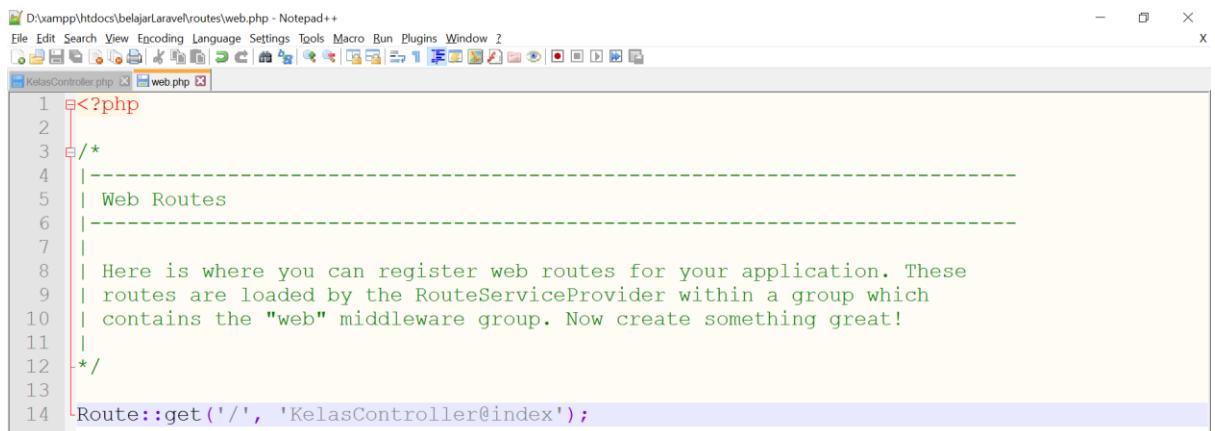
```
D:\xampp\htdocs\belajarLaravel\app\Http\Controllers\KelasController.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ? 
KelasController.php

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class KelasController extends Controller
8 {
9     public function index()
10    {
11        return view('kelas/index');
12    }
13 }
```

Notes:

- return view akan menampilkan view sesuai dengan yang ada pada parameternya. Pada sintaks diatas akan menampilkan view kelas/index.php

13. Buka file **routes/web.php**, tambahkan ubah routes yang sudah ada menjadi seperti gambar berikut:



```
D:\xampp\htdocs\belajarLaravel\routes\web.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ? 
KelasController.php web.php

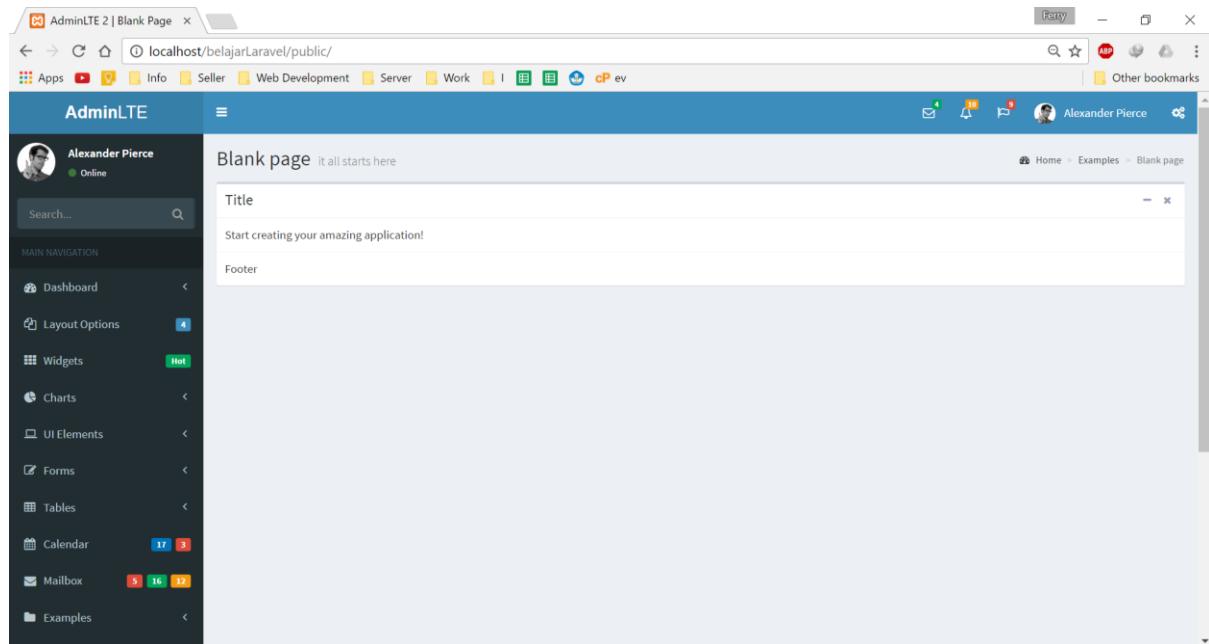
1 <?php
2
3 /*
4 | -----
5 | Web Routes
6 | -----
7 |
8 | Here is where you can register web routes for your application. These
9 | routes are loaded by the RouteServiceProvider within a group which
10 | contains the "web" middleware group. Now create something great!
11 |
12 */
13
14 Route::get('/', 'KelasController@index');
```

Notes:

- Route::get digunakan untuk menghandle request GET. Parameter pertama merupakan URL yang diakses. Parameter kedua adalah aksi dari request tersebut. Pada contoh diatas URL yang diakses adalah root atau index dari web kalian (<http://localhost/belajarLaravel/public>). Kalau sudah dihosting berarti mengakses <http://www.webkalian.com>

14. Akses web kalian dengan alamat url <http://localhost/belajarLaravel/public>

Jika berhasil maka tampilannya akan seperti ini:



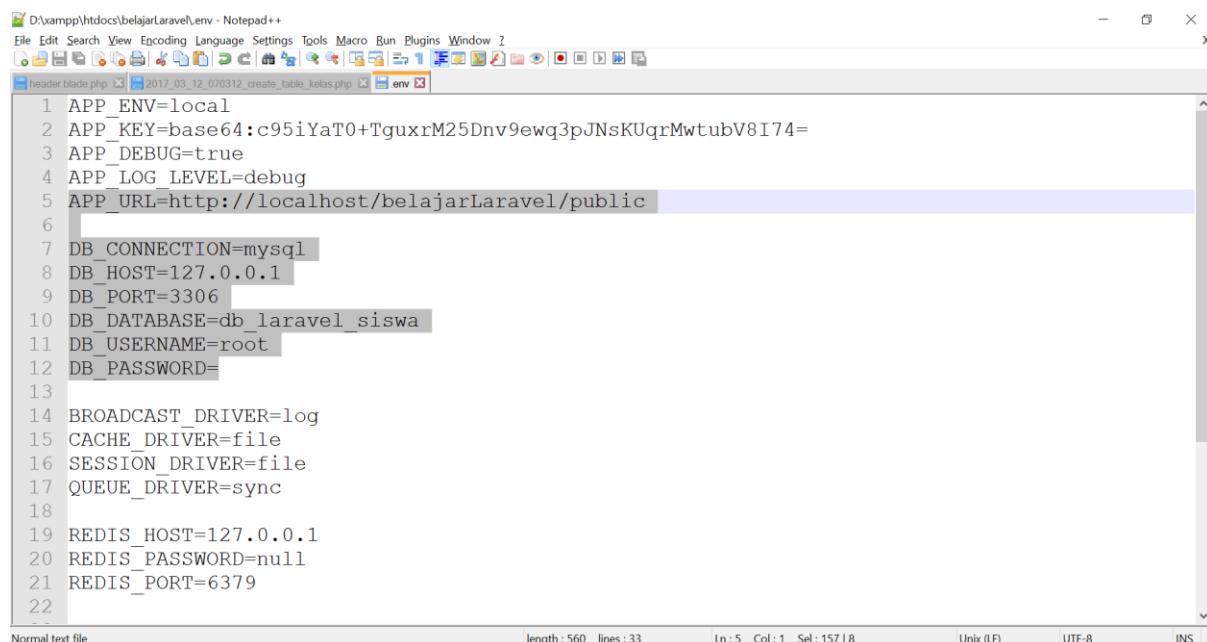
CRUD – Read Data

Pada pembahasan kali ini akan dijelaskan bagaimana membaca data dari database menggunakan Laravel dan Eloquent. Eloquent sendiri merupakan fitur dari Laravel yang memungkinkan kita memanggil data dari database dalam bentuk Entity Object, tanpa syntax MySQL sama sekali. Untuk membuat database MySQL nya pun akan digunakan fitur migration dari Laravel. Fitur ini bermanfaat sekali apabila kalian sudah membangun project laravel dalam suatu tim. Migration ini membantu web artisan dalam mendefinisikan table (DDL) dalam bentuk sintaks pemrograman PHP.

Membuat Tabel dengan Migration

Ikuti langkah-langkah berikut ini:

1. Setting database kamu dan setting juga app_url kamu di pada .env (Di root folder project laravel)



```
D:\xampp\htdocs\belajarLaravel\.env - Notepad++  
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window Z  
header blade.php 2017_03_12_070312_create_table_kelas.php env  
1 APP_ENV=local  
2 APP_KEY=base64:c95iYaT0+TguxrM25Dnv9ewq3pJNsKUqrMwtubV8I74=  
3 APP_DEBUG=true  
4 APP_LOG_LEVEL=debug  
5 APP_URL=http://localhost/belajarLaravel/public  
6  
7 DB_CONNECTION=mysql  
8 DB_HOST=127.0.0.1  
9 DB_PORT=3306  
10 DB_DATABASE=db_laravel_siswa  
11 DB_USERNAME=root  
12 DB_PASSWORD=  
13  
14 BROADCAST_DRIVER=log  
15 CACHE_DRIVER=file  
16 SESSION_DRIVER=file  
17 QUEUE_DRIVER=sync  
18  
19 REDIS_HOST=127.0.0.1  
20 REDIS_PASSWORD=null  
21 REDIS_PORT=6379  
22
```

Normal text file length : 560 lines : 33 Ln : 5 Col : 1 Sel : 157 | 8 Unix (LF) UTF-8 INS

2. Buat database baru dengan nama **db_laravel_siswa** di phpmyadmin / aplikasi RDBMS yang biasa kamu gunakan.

The screenshot shows the phpMyAdmin interface. In the top navigation bar, there are tabs for Databases, SQL, Status, Users, Export, Import, Settings, Replication, Variables,Charsets, and Engines. The 'Databases' tab is selected. On the left sidebar, there's a tree view of databases, including 'New', '10_modul_c', 'ccdol', 'coba', 'db', 'gresearch', 'guruku', 'iciscom_lcis', 'information_schema', 'inkanans', 'madmin', 'mysql', 'performance_schema', 'phpmyadmin', 'portal', 'remedial', 'rems_cmi_db', 'rpjmd_silakip', 'simak', 'simita', 'siti4', 'test', and 'tokosaya'. A 'Create database' button is visible at the top of the main content area, with the input field containing 'db_laravel_siswa'. Below it, a dropdown menu for 'Collation' is set to 'latin1_swedish_ci'. A 'Create' button is also present. A note at the bottom of the database list says: 'Note: Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server.'

3. Membuat migration baru untuk membuat tabel **t_kelas**. Pertama kita buat dulu migration nya dengan menggunakan artisan.

```
> php artisan make:migration create_table_kelas
```

The terminal window shows the command being run: 'C:\WINDOWS\system32\cmd.exe - php artisan make:migration create_table_kelas'. The output shows the migration was created successfully: 'D:\xampp\htdocs\belajarLaravel>php artisan make:migration create_table_kelas' and 'Created Migration: 2017_03_12_070312_create_table_kelas'.

Jika berhasil maka akan muncul file baru pada folder **database/migrations**

The file explorer shows the 'migrations' folder. It contains three files: '2014_10_12_000000_create_users_table.php', '2014_10_12_100000_create_password_resets_table.php', and '2017_03_12_070312_create_table_kelas.php'. The 'File' tab is selected in the top left.

Name	Date modified	Type	Size
2014_10_12_000000_create_users_table.php	12/03/2017 11:10	PHP File	1 KB
2014_10_12_100000_create_password_resets_table.php	12/03/2017 11:10	PHP File	1 KB
2017_03_12_070312_create_table_kelas.php	12/03/2017 14:03	PHP File	1 KB

File migration `create_user_table` dan `create_password_reset_table` boleh dihapus, karena kita belum memerlukan tabel tersebut.

4. Buka file **create_table_kelas.php**, masukan sintaks berikut ini pada blok fungsi **up** dan **down**:

```
D:\xampp\htdocs\belajarLaravel\database\migrations\2017_03_12_070312_create_table_kelas.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
index.blade.php Kelas.php 2017_03_12_070312_create_table_kelas.php [3]
7 class CreateTableKelas extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13     */
14    public function up()
15    {
16        Schema::create('t_kelas', function (Blueprint $table) {
17            $table->increments('id_kelas');
18            $table->string('nama_kelas', 100);
19            $table->string('jurusan', 100);
20            $table->timestamps();
21        });
22    }
23
24    /**
25     * Reverse the migrations.
26     *
27     * @return void
28     */
29    public function down()
30    {
31        Schema::dropIfExists('t_kelas');
32    }
33 }
PHP Hypertext Preprocessor file length : 686 lines : 33 Ln : 22 Col : 6 Sel : 0 | 0 Unix (LF) UTF-8 INS
```

Notes:

- Fungsi **up** akan dijalankan ketika kita melakukan `php artisan migrate`
- Fungsi **down** akan dijalankan ketika kita melakukan `php artisan migrate:rollback`
- Fungsi `$table->string` untuk membuat field dengan tipe data varchar. Parameter kedua adalah length dari field tersebut. Tipe data lainnya dapat dilihat pada <https://laravel.com/docs/5.3/migrations#creating-columns>

5. Buka kembali artisan, ketik perintah dibawah ini:

```
> php artisan migrate
```

```
C:\WINDOWS\system32\cmd.exe
D:\xampp\htdocs\belajarLaravel>php artisan migrate
Migration table created successfully.
Migrated: 2017_03_12_070312_create_table_kelas
```

Jika berhasil maka tabel akan otomatis dibuat oleh artisan.

The screenshot shows the phpMyAdmin interface. On the left, there's a tree view of databases: db_ion, db_laravel_siswa (which is expanded to show New, migrations, and t_kelas), db_lks, db_jppm, db_lvb, and db_lvb4. The main area shows the 'Table structure' for the t_kelas table. The table has five columns: id_kelas (int(10), UNSIGNED, AUTO_INCREMENT), nama_kelas (varchar(100), utf8_unicode_ci), jurusan (varchar(100), utf8_unicode_ci), created_at (timestamp), and updated_at (timestamp). Each column has 'Change', 'Drop', 'Primary', and 'More' actions.

Membuat Dummy Data menggunakan Database Seed

Langkah berikutnya adalah memasukan data kedalam tabel t_siswa dengan database seeder bawaan dari laravel. Ikuti langkah berikut ini:

1. Buka command prompt pada folder laravel kamu, ketik perintah ini untuk membuat seeder.

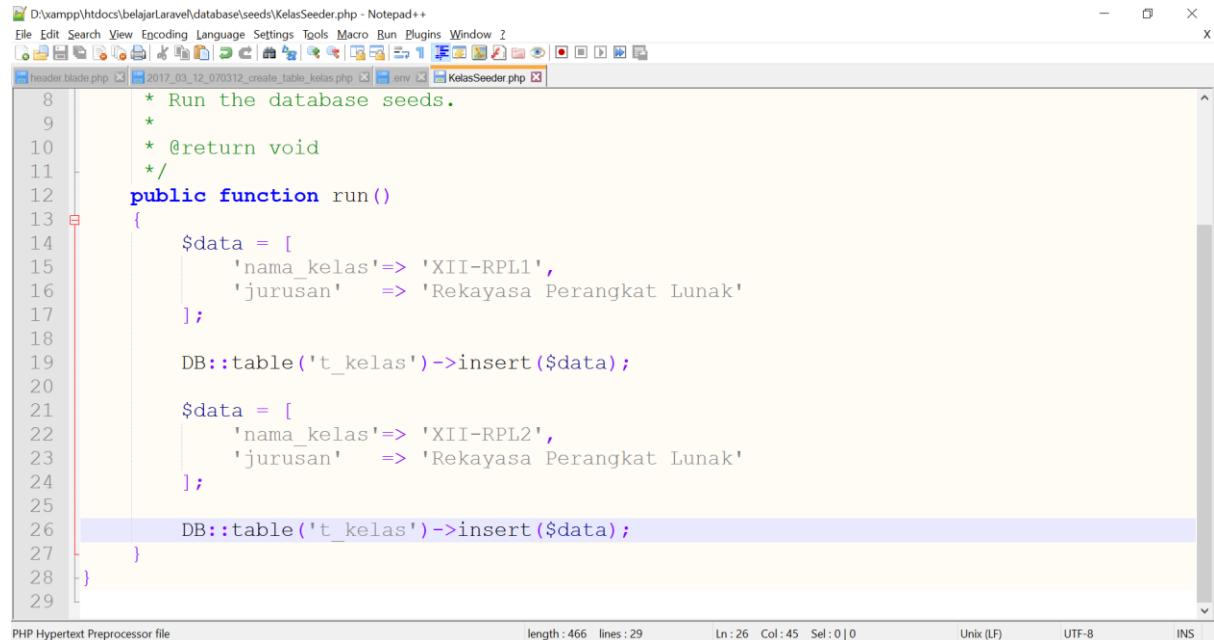
```
> php artisan make:seeder KelasSeeder
```

```
C:\WINDOWS\system32\cmd.exe
D:\xampp\htdocs\belajarLaravel>php artisan make:seed KelasSeeder
Seeder created successfully.

D:\xampp\htdocs\belajarLaravel>
```

Jika berhasil akan muncul file **KelasSeeder.php** pada folder **database/seeds**

2. Buat perintah untuk memasukan data kedalam table t_kelas pada block function run.



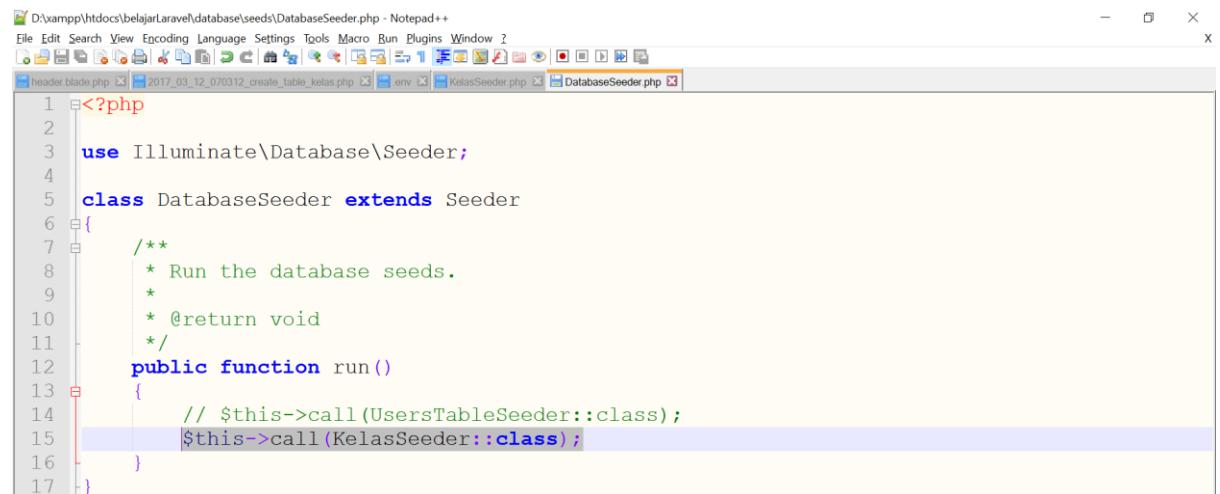
```

D:\xampp\htdocs\belajarLaravel\database\seeds\KelasSeeder.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
header.blade.php 2017_03_12_070312_create_table_kelas.php env KelasSeeder.php
8     * Run the database seeds.
9
10    *
11    * @return void
12    */
13    public function run()
14    {
15        $data = [
16            'nama_kelas'=> 'XII-RPL1',
17            'jurusan' => 'Rekayasa Perangkat Lunak'
18        ];
19
20        DB::table('t_kelas')->insert($data);
21
22        $data = [
23            'nama_kelas'=> 'XII-RPL2',
24            'jurusan' => 'Rekayasa Perangkat Lunak'
25        ];
26
27        DB::table('t_kelas')->insert($data);
28    }
29

```

PHP Hypertext Preprocessor file length : 466 lines : 29 Ln : 26 Col : 45 Sel : 0 | 0 Unix (LF) UTF-8 INS

3. Buka file **database/seeds/DatabaseSeeder.php**, kemudian tambahkan sintaks berikut:



```

D:\xampp\htdocs\belajarLaravel\database\seeds\DatabaseSeeder.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
header.blade.php 2017_03_12_070312_create_table_kelas.php env KelasSeeder.php DatabaseSeeder.php
1 <?php
2
3 use Illuminate\Database\Seeder;
4
5 class DatabaseSeeder extends Seeder
6 {
7     /**
8      * Run the database seeds.
9      *
10     * @return void
11     */
12     public function run()
13     {
14         // $this->call(UsersTableSeeder::class);
15         $this->call(KelasSeeder::class);
16     }
17 }

```

- Buka kembali command prompt, ketik perintah berikut ini untuk memasukan data dummy tersebut kedalam table t_kelas.

```
> php artisan db:seed
```

```
C:\WINDOWS\system32\cmd.exe
D:\xampp\htdocs\belajarLaravel>php artisan db:seed
Seeding: KelasSeeder
```

Apabila langkah diatas berhasil maka akan muncul datanya pada tabel t_kelas

	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	<input type="checkbox"/> 1	XII-RPL1	Rekayasa Perangkat Lunak	<input type="checkbox"/> created_at	<input type="checkbox"/> updated_at
	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	<input type="checkbox"/> 2	XII-RPL2	Rekayasa Perangkat Lunak	<input type="checkbox"/> NULL	<input type="checkbox"/> NULL

Menampilkan data di views

Setelah database, tabel dan data sudah dibuat, sekarang buat tampilan dalam bentuk tabel untuk menampilkan data-data tersebut di web yang kita buat.

Ikuti langkah berikut:

- Buka command prompt, ketik sintaks berikut untuk membuat Model

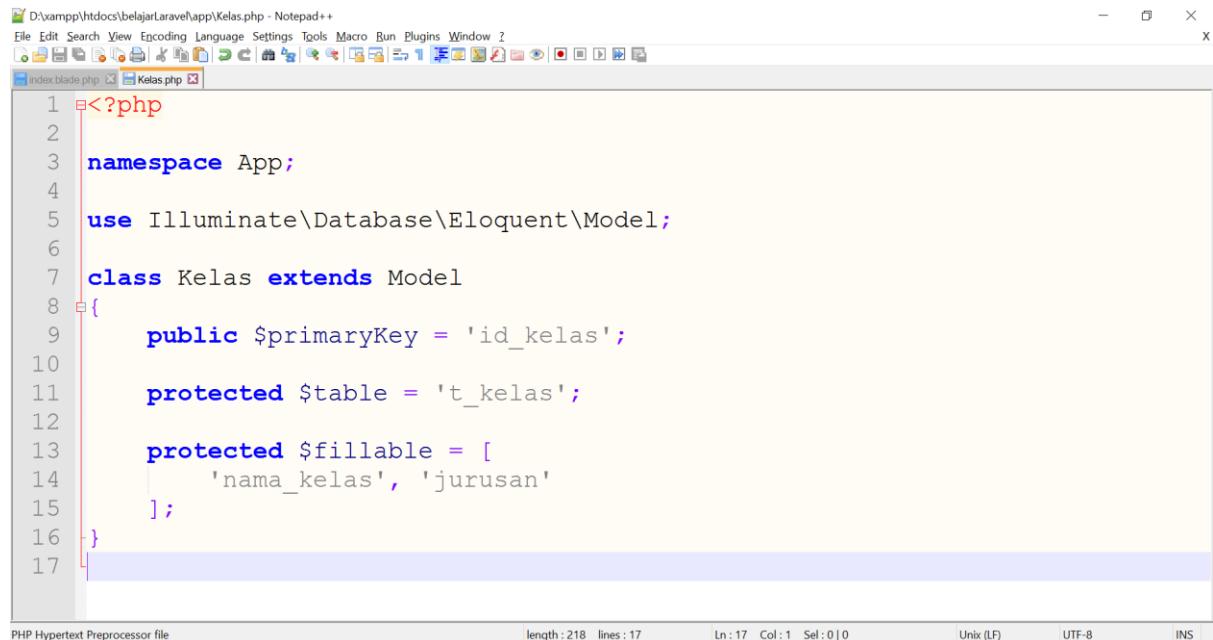
```
> php artisan make:model Kelas
```

```
C:\WINDOWS\system32\cmd.exe
D:\xampp\htdocs\belajarLaravel>php artisan make:model Kelas
Model created successfully.

D:\xampp\htdocs\belajarLaravel>
```

Jika berhasil maka model akan muncul di folder **app**

2. Buka file **app/Kelas.php**, kemudian edit file tersebut menjadi seperti pada gambar:

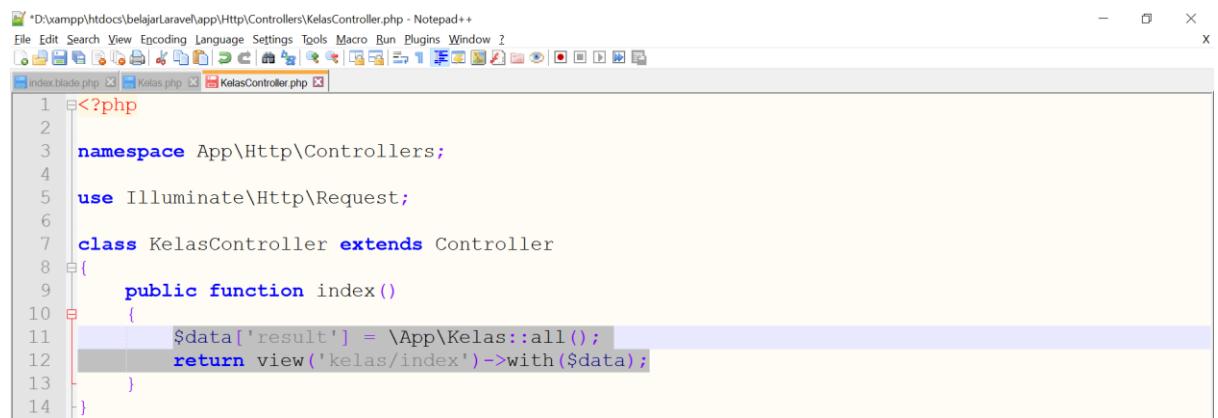


```
1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Kelas extends Model
8 {
9     public $primaryKey = 'id_kelas';
10
11     protected $table = 't_kelas';
12
13     protected $fillable = [
14         'nama_kelas', 'jurusan'
15     ];
16 }
17
```

Notes:

- \$primaryKey untuk menentukan primary key pada tabel
- \$table untuk menentukan nama tabel
- \$fillable digunakan untuk menentukan field yang mana saja yang akan di insert kedalam database

3. Buka file **app/Http/Controllers/KelasController.php**, kemudian modifikasi sintaks berikut ini:



```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class KelasController extends Controller
8 {
9     public function index()
10    {
11        $data['result'] = \App\Kelas::all();
12        return view('kelas/index')->with($data);
13    }
14 }
```

4. Buka file **resources/views/kelas/index.blade.php**, modifikasi file tersebut sehingga mendapatkan tampilan seperti dibawah ini:

D:\xampp\htdocs\belajarLaravel\resources\views\kelas\index.blade.php - Notepad++

```

1 @extends('templates/header')
2
3 @section('content')
4 <!-- Content Header (Page header) -->
5   <section class="content-header">
6     <h1>
7       Data Kelas
8       <small>SMK Negeri 4 Bandung</small>
9     </h1>
10    <ol class="breadcrumb">
11      <li><a href="{{ url('/') }}"><i class="fa fa-dashboard"></i> Home</a></li>
12      <li class="active">Data Kelas</li>
13    </ol>
14  </section>
15
16  <!-- Main content -->
17  <section class="content">
18
19    <!-- Default box -->
20    <div class="box">
21      <div class="box-header with-border">
22        <a href="{{ url('kelas/add') }}" class="btn btn-success"><i class="fa fa-plus-circle"></i> Tambah</a>

```

PHP Hypertext Preprocessor file length : 1.818 lines : 60 Ln : 2 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

```

22 <a href="{{ url('kelas/add') }}" class="btn btn-success"><i class="fa fa-plus-circle"></i> Tambah</a>
23 </div>
24 <div class="box-body">
25 <table class="table table-striped">
26 <thead>
27 <tr>
28 <th>No</th>
29 <th>Nama Kelas</th>
30 <th>Jurusan</th>
31 <th>Action</th>
32 </tr>
33 </thead>
34
35 <tbody>
36 @foreach ($result as $row)
37 <tr>
38 <td>{{ !empty($i) ? ++$i : $i = 1 }}</td>
39 <td>{{ $row->nama_kelas }}</td>
40 <td>{{ $row->jurusan }}</td>
41 <td>
42 <a href="{{ url("kelas/$row->id_kelas/edit") }}" class="btn btn-sm btn-warning"><i class="fa fa-pencil"></i></a>

```

PHP Hypertext Preprocessor file length : 1.818 lines : 60 Ln : 2 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

```

42 <a href="{{ url("kelas/$row->id_kelas/edit") }}" class="btn btn-sm btn-warning"><i class="fa fa-pencil"></i></a>
43 <form action="{{ url("kelas/$row->id_kelas/delete") }}" method="POST" style="display:inline;">
44 {{ csrf_field() }}
45 {{ method_field('DELETE') }}
46 <button class="btn btn-sm btn-danger"><i class="fa fa-trash"></i></a>
47 </form>
48 </td>
49 </tr>
50 @endforeach
51 </tbody>
52 </table>
53 </div>
54 <!-- /.box-body -->
55 </div>
56 <!-- /.box -->
57
58 </section>
59 <!-- /.content -->
60 @endsection

```

PHP Hypertext Preprocessor file length : 1.818 lines : 60 Ln : 60 Col : 12 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

Notes:

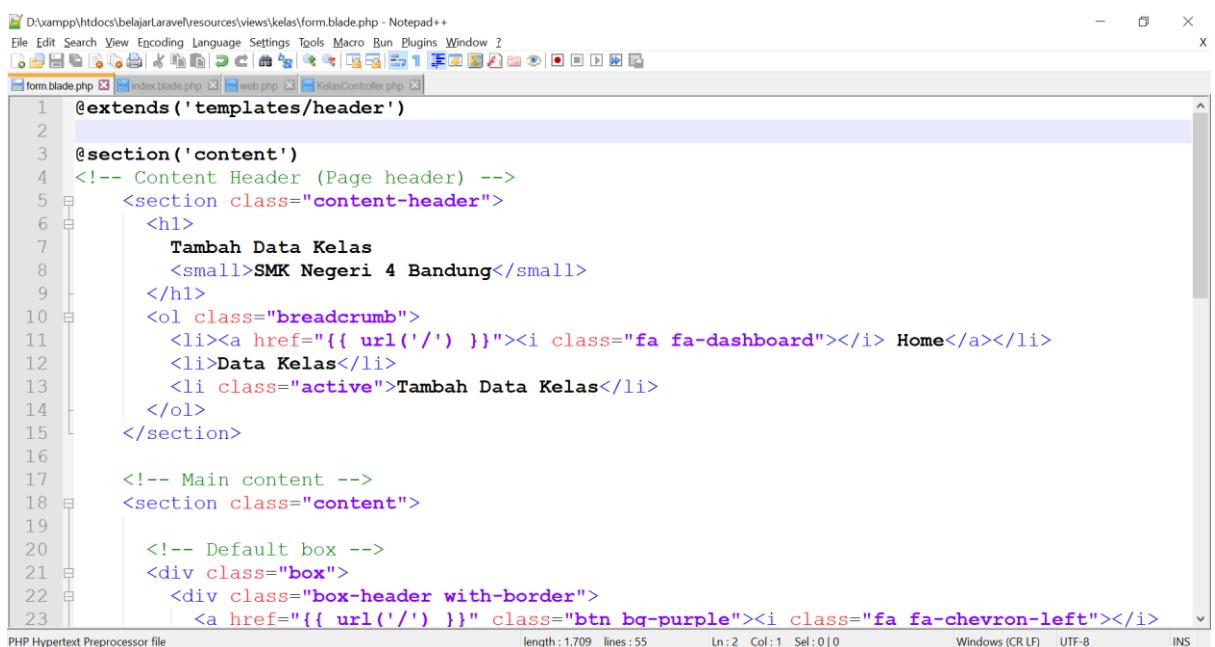
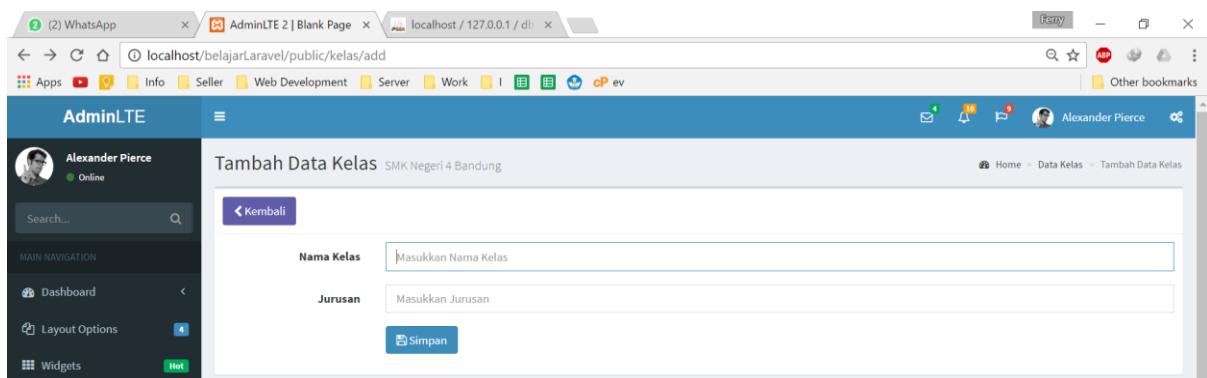
- @foreach digunakan untuk melakukan looping data, sama seperti php foreach, tapi ini menggunakan blade, lebih simple tanpa tag php
- Fungsi **url('kelas/add')** digunakan untuk menghasilkan url <http://localhost/belajarLaravel/kelas/add>
- Untuk menghapus data diperlukan form, token, dan method field delete seperti pada gambar diatas.

CRUD – Create Data

Pada pembahasan kali ini akan dijelaskan tentang bagaimana carana memasukan data kedalam database dengan Laravel Eloquent.

Ikuti langkah-langkah berikut ini:

1. Buat file baru di **resources/views/kelas/form.blade.php** (Copy saja dari **index.blade.php**). Kedepannya file ini juga akan digunakan untuk edit data.
2. Modifikasi file **resources/views/kelas/form.blade.php** untuk membuat form seperti pada gambar dibawah ini:



```
D:\xampp\htdocs\belajarLaravel\resources\views\kelas\form.blade.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ? 
form blade.php index.blade.php web.php KelasController.php
1 @extends('templates/header')
2
3 @section('content')
4 <!-- Content Header (Page header) -->
5 <section class="content-header">
6   <h1>
7     Tambah Data Kelas
8     <small>SMK Negeri 4 Bandung</small>
9   </h1>
10  <ol class="breadcrumb">
11    <li><a href="{{ url('/') }}><i class="fa fa-dashboard"></i> Home</a></li>
12    <li>Data Kelas</li>
13    <li class="active">Tambah Data Kelas</li>
14  </ol>
15 </section>
16
17 <!-- Main content -->
18 <section class="content">
19
20   <!-- Default box -->
21   <div class="box">
22     <div class="box-header with-border">
23       <a href="{{ url('/') }}" class="btn bg-purple"><i class="fa fa-chevron-left"></i>
```

```

22     <div class="box-header with-border">
23         <a href="{{ url('/') }}" class="btn bg-purple"><i class="fa fa-chevron-left"></i>
24             Kembali</a>
25     </div>
26     <div class="box-body">
27         <form action="{{ url('kelas/add') }}" class="form-horizontal" method="POST">
28             {{ csrf_field() }}
29             <div class="form-group">
30                 <label class="control-label col-sm-2">Nama Kelas</label>
31                 <div class="col-sm-10">
32                     <input type="text" name="nama_kelas" class="form-control" placeholder="Masukkan Nama Kelas" />
33                 </div>
34             </div>
35             <div class="form-group">
36                 <label class="control-label col-sm-2">Jurusan</label>
37                 <div class="col-sm-10">
38                     <input type="text" name="jurusan" class="form-control" placeholder="Masukkan Jurusan" />
39                 </div>
40             </div>
41         </form>

```

PHP Hypertext Preprocessor file length : 1.709 lines : 55 Ln : 2 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

Notes:

- Fungsi **csrf_field()** digunakan untuk menciptakan field _token yang diperlukan sebagai validasi keamanan dari cross site attack pada saat request post. Sintaks tersebut diperlukan setiap kali akan melakukan request POST (Form method POST).

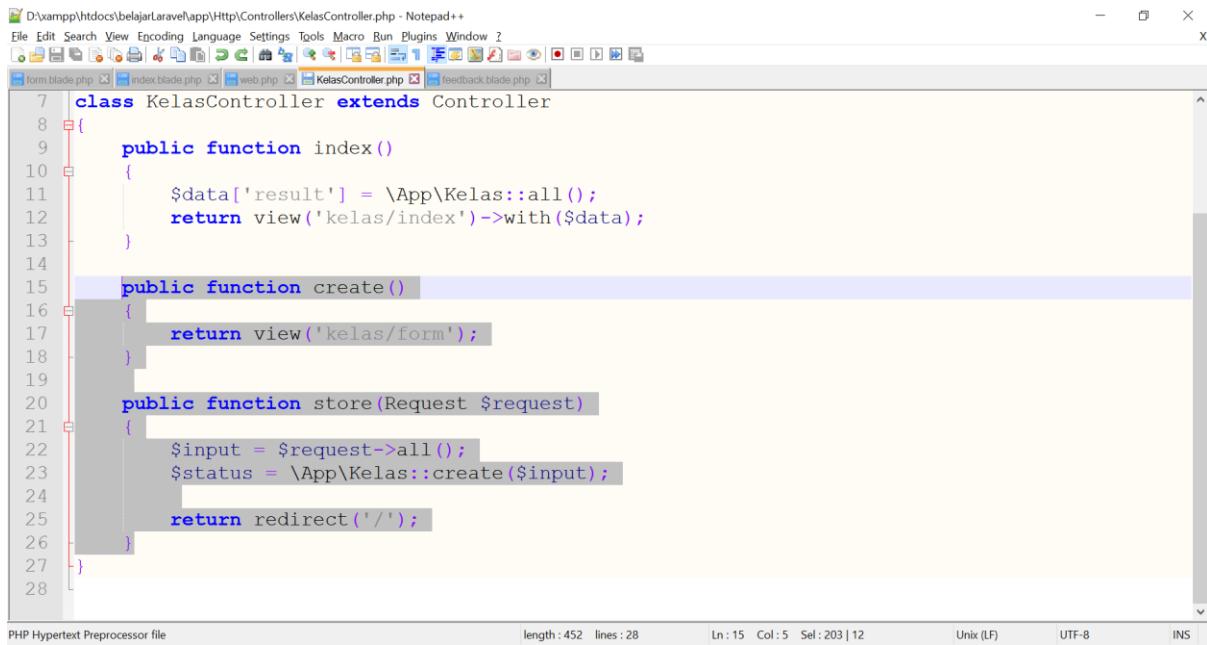
```

36             <label class="control-label col-sm-2">Jurusan</label>
37             <div class="col-sm-10">
38                 <input type="text" name="jurusan" class="form-control" placeholder="Masukkan Jurusan" />
39             </div>
40         </div>
41
42         <div class="form-group">
43             <div class="col-sm-10 col-sm-offset-2">
44                 <button type="submit" class="btn btn-primary"><i class="fa fa-save"></i> Simpan</button>
45             </div>
46         </div>
47     </div>
48     <!-- /.box-body -->
49 </div>
50 <!-- /.box -->
51
52
53 </section>
54 <!-- /.content -->
55 @endsection

```

PHP Hypertext Preprocessor file length : 1.709 lines : 55 Ln : 55 Col : 12 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

3. Buka file **KelasController.php**, tambahkan sintaks berikut ini:



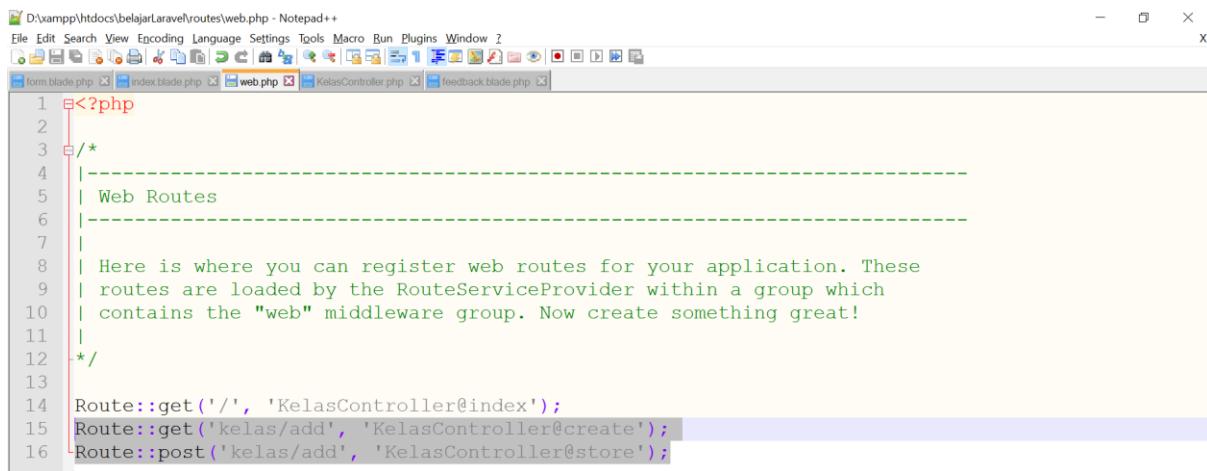
```
D:\xampp\htdocs\belajarLaravel\app\Http\Controllers\KelasController.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 2
form.blade.php index.blade.php web.php KelasController.php feedback.blade.php
7 class KelasController extends Controller
8 {
9     public function index()
10    {
11        $data['result'] = \App\Kelas::all();
12        return view('kelas/index')->with($data);
13    }
14
15    public function create()
16    {
17        return view('kelas/form');
18    }
19
20    public function store(Request $request)
21    {
22        $input = $request->all();
23        $status = \App\Kelas::create($input);
24
25        return redirect('/');
26    }
27
28 }
```

PHP Hypertext Preprocessor file length : 452 lines : 28 Ln : 15 Col : 5 Sel : 203 | 12 Unix (LF) UTF-8 INS

Notes:

- Fungsi create digunakan untuk menampilkan halaman form kelas.
- Fungsi store digunakan untuk menyimpan data ke database
 - \App\Kelas::create adalah fungsi untuk memasukan data ke model Kelas (Eloquent). Parameter \$input ini berasal dari request->all(), yaitu mengambil seluruh field form. Namun jangan khawatir, yang di input ke database hanya name-name yang kamu set di app\Kelas.php (Model) pada bagian \$fillable.

4. Buka file **routes/web.php**, tambahkan sintaks berikut ini:



```
D:\xampp\htdocs\belajarLaravel\routes\web.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 2
form.blade.php index.blade.php web.php KelasController.php feedback.blade.php
1 <?php
2 /*
3 | -----
4 | Web Routes
5 | -----
6 |
7 |
8 | Here is where you can register web routes for your application. These
9 | routes are loaded by the RouteServiceProvider within a group which
10 | contains the "web" middleware group. Now create something great!
11 |
12 */
13
14 Route::get('/', 'KelasController@index');
15 Route::get('kelas/add', 'KelasController@create');
16 Route::post('kelas/add', 'KelasController@store');
```

Notes:

- Route::get digunakan untuk menampilkan halaman form input data kelas
 - Route::post digunakan ketika tombol simpan ditekan (Form Action)
- Kamu bisa membuat nama URL yang sama seperti pada contoh diatas, tapi berbeda-beda action nya tergantung request nya. (GET | POST | PATCH / PUT | DELETE)

5. Coba buka <http://localhost/belajarLaravel/public>, klik tombol Tambah, isi form dan kemudian klik tombol **Simpan**.

Jika script diatas berhasil, maka ketika klik tombol **Simpan** akan kembali ke halaman tabel kelas dan data yang baru akan muncul.

No	Nama Kelas	Jurusan	Action
1	XII-RPL1	Rekayasa Perangkat Lunak	
2	XII-RPL2	Rekayasa Perangkat Lunak	
3	XII-RPL3	Rekayasa Perangkat Lunak	

Menambahkan Feedback Sukses atau Gagal

Pada tahap sebelumnya, belum ada informasi data sukses atau gagal disimpan, hanya langsung saja pindah ke halaman data kelas. Untuk memberikan *feedback* kepada user jika data berhasil disimpan atau memberikan informasi jika data gagal disimpan, ikuti langkah-langkah berikut ini:

1. Buat file baru di **resources/templates/feedback.blade.php**

```

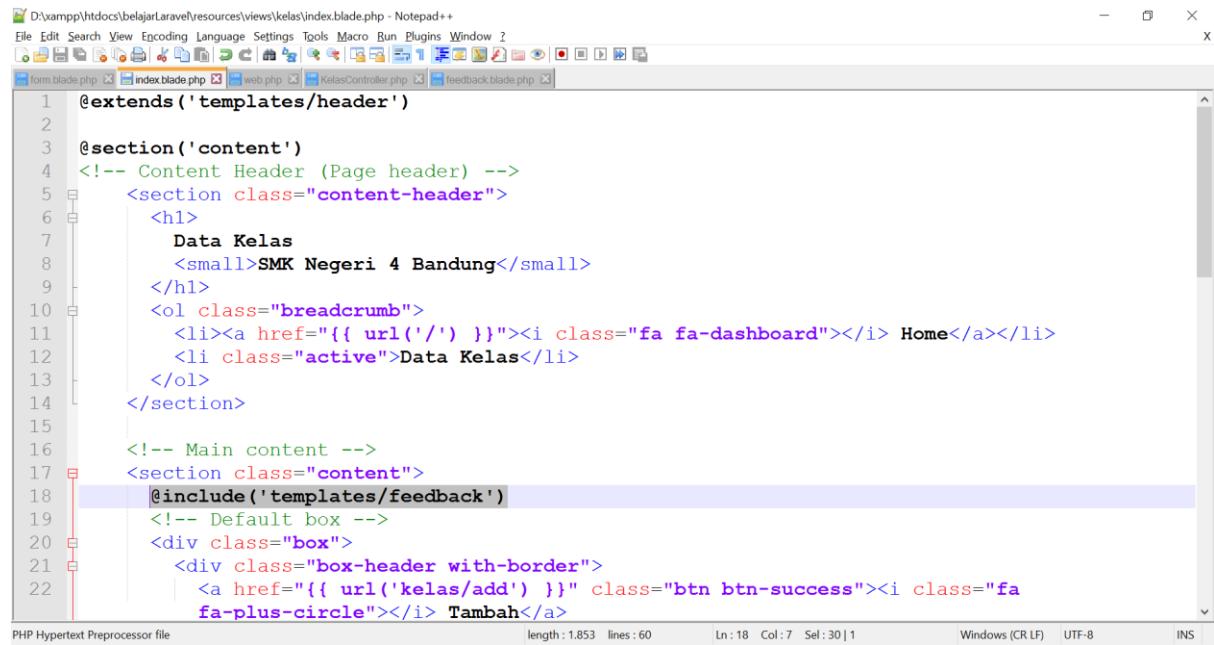
D:\xampp\htdocs\belajarLaravel\resources\views\templates\feedback.blade.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 2
form.blade.php index.blade.php web.php KelasController.php feedback.blade.php
1 @if(session('success'))
2     <div class="alert alert-success fade in">
3         <button type="button" class="close pull-right" data-dismiss="alert" aria-label="Close">
4             <span aria-hidden="true">&times;&times;

```

Notes:

- Fungsi **session** digunakan untuk mengambil session dengan nama yang ada pada parameter tersebut. **session('success')** artinya mengambil **session** dengan nama **success**.

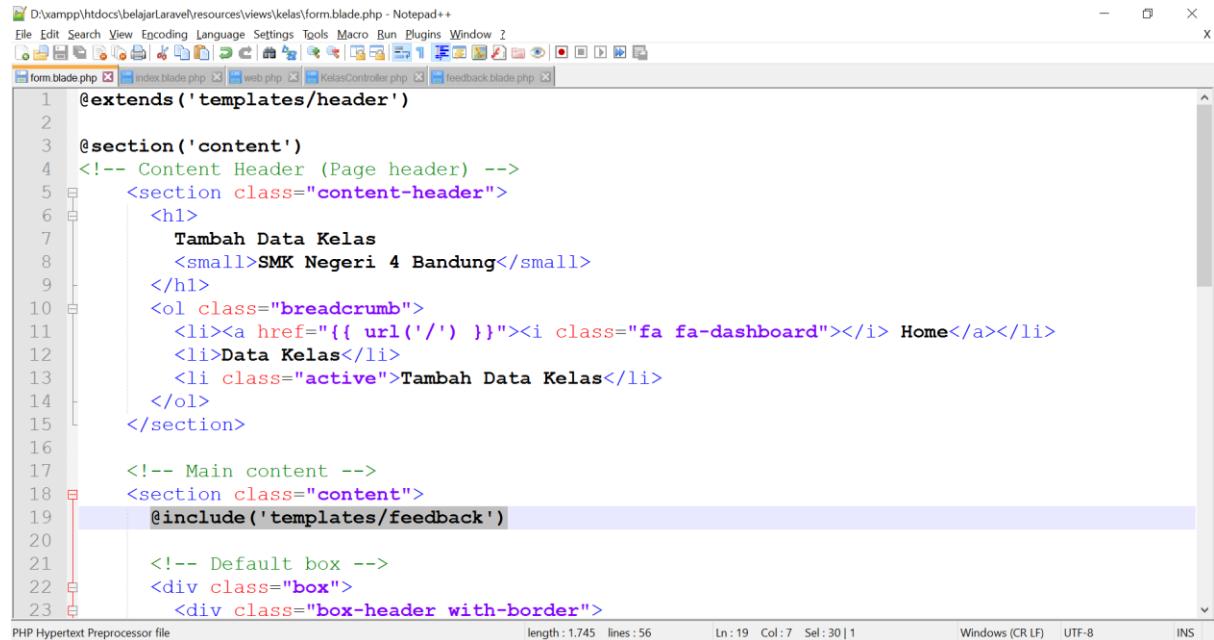
2. Tambahkan include pada file **resources/kelas/index.blade.php**



```
D:\xampp\htdocs\belajarLaravel\views\kelas\index.blade.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 
form.blade.php index.blade.php web.php KelasController.php feedback.blade.php

1 @extends('templates/header')
2
3 @section('content')
4 <!-- Content Header (Page header) -->
5     <section class="content-header">
6         <h1>
7             Data Kelas
8             <small>SMK Negeri 4 Bandung</small>
9         </h1>
10        <ol class="breadcrumb">
11            <li><a href="{{ url('/') }}"><i class="fa fa-dashboard"></i> Home</a></li>
12            <li class="active">Data Kelas</li>
13        </ol>
14    </section>
15
16    <!-- Main content -->
17    <section class="content">
18        @include('templates/feedback')
19        <!-- Default box -->
20        <div class="box">
21            <div class="box-header with-border">
22                <a href="{{ url('kelas/add') }}" class="btn btn-success"><i class="fa fa-plus-circle"></i> Tambah</a>
```

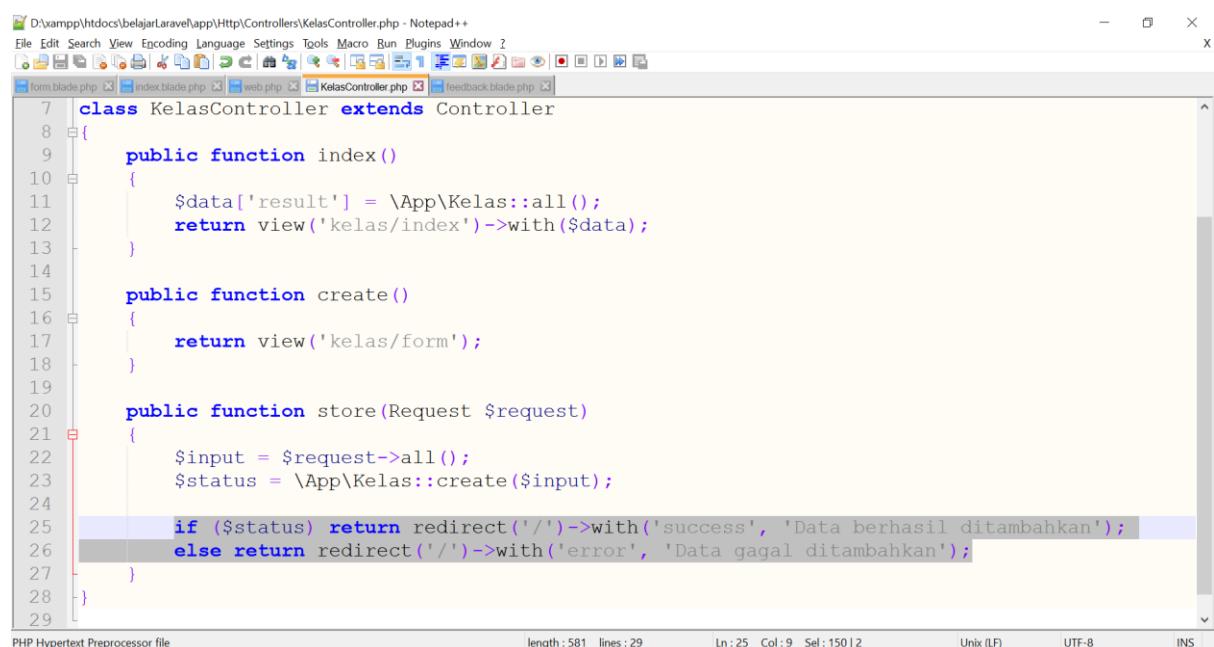
3. Tambahkan juga pada file **resources/kelas/form.blade.php**



```
D:\xampp\htdocs\belajarLaravel\views\kelas\form.blade.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 
form.blade.php index.blade.php web.php KelasController.php feedback.blade.php

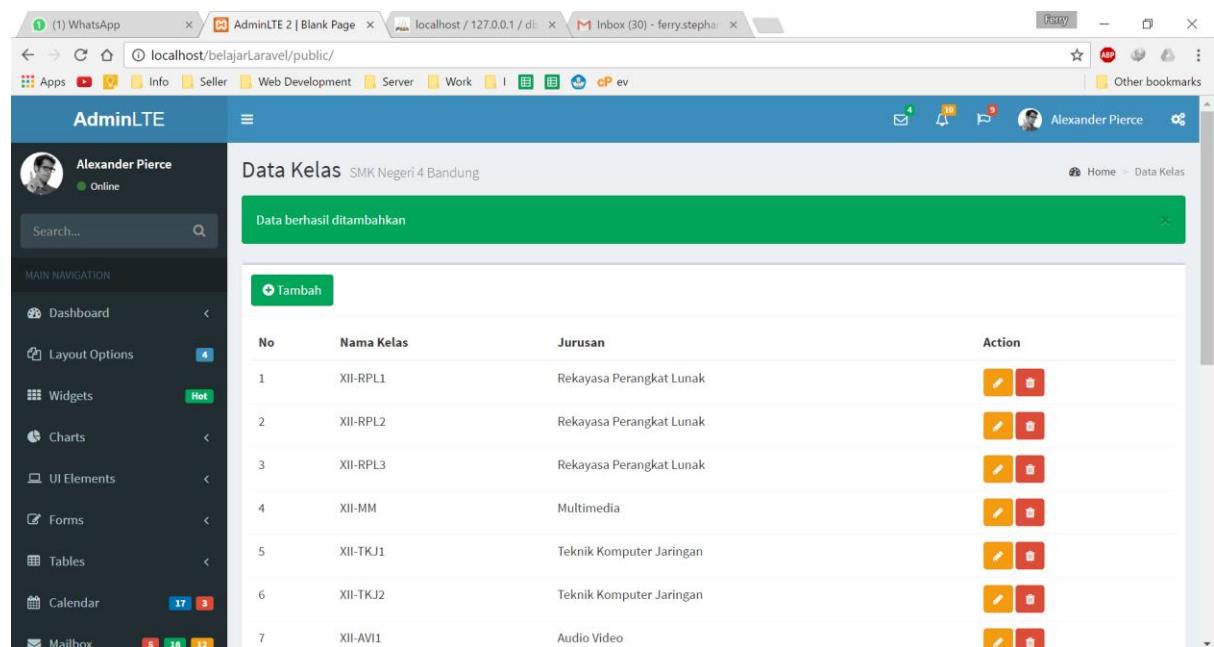
1 @extends('templates/header')
2
3 @section('content')
4 <!-- Content Header (Page header) -->
5     <section class="content-header">
6         <h1>
7             Tambah Data Kelas
8             <small>SMK Negeri 4 Bandung</small>
9         </h1>
10        <ol class="breadcrumb">
11            <li><a href="{{ url('/') }}"><i class="fa fa-dashboard"></i> Home</a></li>
12            <li>Data Kelas</li>
13            <li class="active">Tambah Data Kelas</li>
14        </ol>
15    </section>
16
17    <!-- Main content -->
18    <section class="content">
19        @include('templates/feedback')
20
21        <!-- Default box -->
22        <div class="box">
23            <div class="box-header with-border">
```

4. Buka file **app/Http/Controllers/KelasController.php**, tambahkan sintaks berikut ini:



```
D:\xampp\htdocs\belajarLaravel\app\Http\Controllers\KelasController.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
form.blade.php index.blade.php web.php KelasController.php feedback.blade.php
1 class KelasController extends Controller
2 {
3     public function index()
4     {
5         $data['result'] = \App\Kelas::all();
6         return view('kelas/index')->with($data);
7     }
8
9     public function create()
10    {
11        return view('kelas/form');
12    }
13
14    public function store(Request $request)
15    {
16        $input = $request->all();
17        $status = \App\Kelas::create($input);
18
19        if ($status) return redirect('/')->with('success', 'Data berhasil ditambahkan');
20        else return redirect('/')->with('error', 'Data gagal ditambahkan');
21    }
22
23 }
24
25 }
26
27 }
28
29 }
```

5. Lakukan kembali input data, dan hasilnya akan seperti pada gambar ini:



The screenshot shows a browser window with the AdminLTE theme. The URL is `localhost/belajarLaravel/public/`. The main content area displays a table titled "Data Kelas" with the following data:

No	Nama Kelas	Jurusan	Action
1	XII-RPL1	Rekayasa Perangkat Lunak	
2	XII-RPL2	Rekayasa Perangkat Lunak	
3	XII-RPL3	Rekayasa Perangkat Lunak	
4	XII-MM	Multimedia	
5	XII-TKJ1	Teknik Komputer Jaringan	
6	XII-TKJ2	Teknik Komputer Jaringan	
7	XII-AVI1	Audio Video	

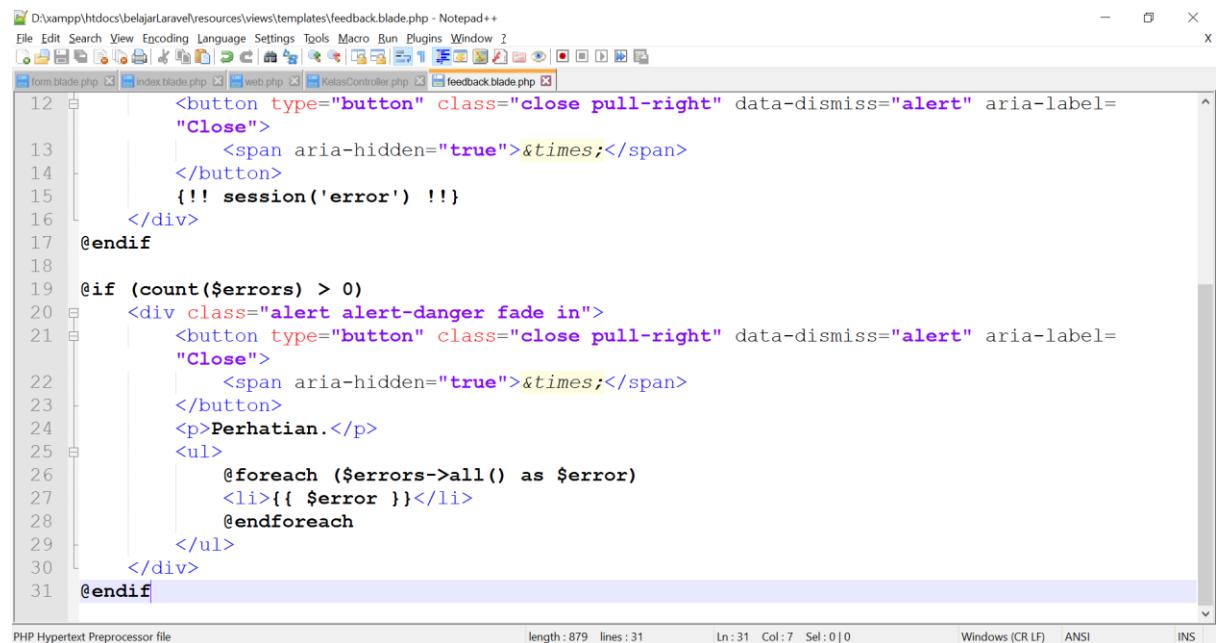
A green success message box at the top of the table area contains the text "Data berhasil ditambahkan".

Validasi Form

Validasi form dilakukan untuk memastikan form yang diisi oleh user sesuai dengan ketentuan yang kita buat pada web kita. Misalnya, Nama Kelas dan Jurusan tidak boleh kosong, atau contoh lainnya input harus berupa angka, dan sebagainya.

Mari kita coba validasi form pada input data kelas. Ikuti langkah berikut ini:

1. Buka file **resources/views/templates/feedback.blade.php**, tambahkan sintaks berikut ini (Line 19-31):

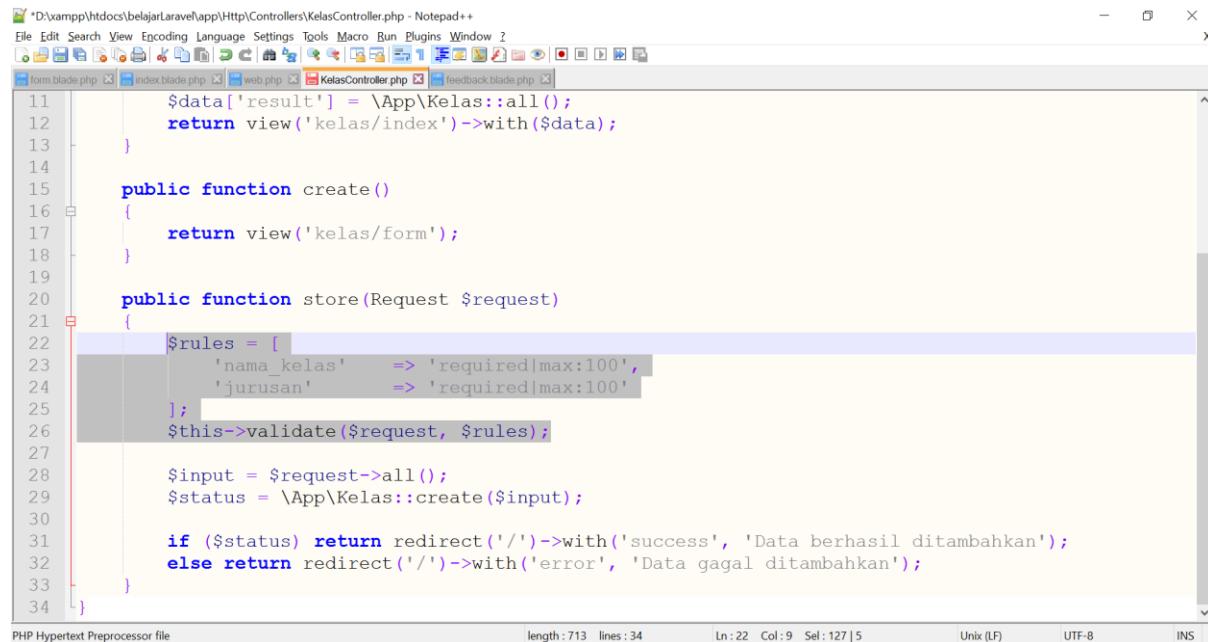


```
D:\xampp\htdocs\belajarLaravel\resources\views\templates\feedback.blade.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
form.blade.php index.blade.php web.php KelasController.php feedback.blade.php
12     <button type="button" class="close pull-right" data-dismiss="alert" aria-label="Close">
13         <span aria-hidden="true">&times;</span>
14     </button>
15     {!! session('error') !!}
16 </div>
17 @endif
18
19 @if (count($errors) > 0)
20     <div class="alert alert-danger fade in">
21         <button type="button" class="close pull-right" data-dismiss="alert" aria-label="Close">
22             <span aria-hidden="true">&times;</span>
23         </button>
24         <p>Perhatian.</p>
25         <ul>
26             @foreach ($errors->all() as $error)
27                 <li>{{ $error }}</li>
28             @endforeach
29         </ul>
30     </div>
31 @endif
```

Notes:

- `$errors->all()` mengambil semua error dari hasil validasi, kemudian di-looping untuk menampilkan error apa saja dari hasil validation.

2. Buka file **app/Http/Controllers/KelasController.php**, pada fungsi store tambahkan sintaks berikut ini:

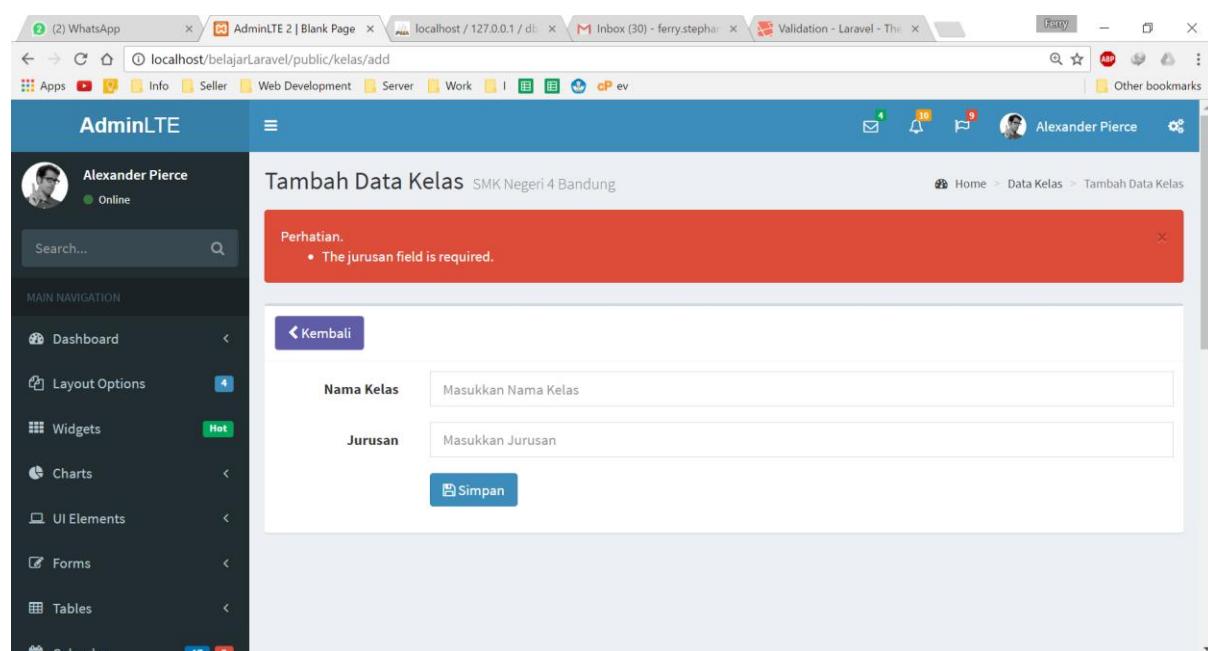


```
*D:\xampp\htdocs\belajarLaravel\app\Http\Controllers\KelasController.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
form.blade.php index.blade.php web.php KelasController.php feedback.blade.php
11 $data['result'] = \App\Kelas::all();
12     return view('kelas/index')->with($data);
13 }
14
15 public function create()
16 {
17     return view('kelas/form');
18 }
19
20 public function store(Request $request)
21 {
22     $rules = [
23         'nama_kelas'    => 'required|max:100',
24         'jurusan'       => 'required|max:100'
25     ];
26     $this->validate($request, $rules);
27
28     $input = $request->all();
29     $status = \App\Kelas::create($input);
30
31     if ($status) return redirect('/')->with('success', 'Data berhasil ditambahkan');
32     else return redirect('/')->with('error', 'Data gagal ditambahkan');
33 }
34 }
```

Notes:

- Rules yang lainnya dapat kamu lihat pada dokumentasi Laravel di alamat <https://laravel.com/docs/5.3/validation#available-validation-rules>

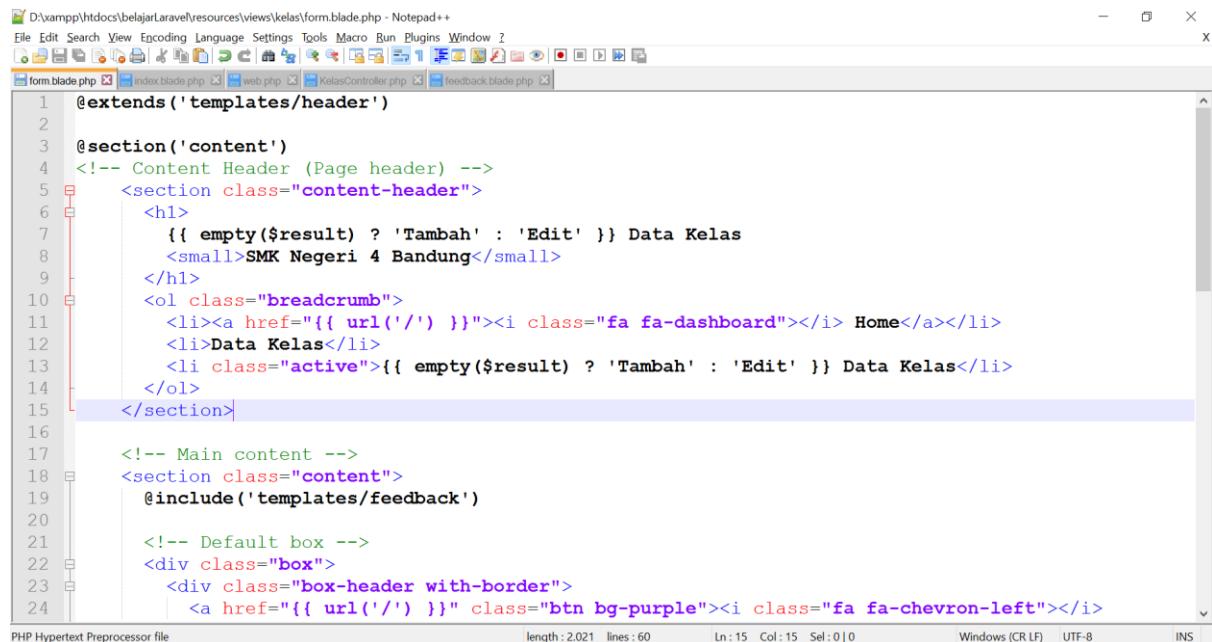
3. Lakukan kembali input data tetapi coba form nya dikosongkan, dan hasilnya akan seperti pada gambar ini:



CRUD – Updating Data

Setelah berhasil melakukan langkah-langkah diatas, sekarang kita coba melakukan editing data. Form yang digunakan masih sama dengan form yang dipakai untuk melakukan input data, hanya saja menambahkan logika untuk melakukan cek apabila terdapat data \$result, maka artinya form tersebut dalam mode Edit, jika tidak terdapat data, maka form tersebut dalam mode Input. Mari kita mulai dengan mengikuti langkah-langkah yang akan dibahas pada bagian ini.

1. Buka file **resources/views/kelas/form.blade.php**, lakukan modifikasi seperti gambar dibawah ini:



```
1 @extends('templates/header')
2
3 @section('content')
4 <!-- Content Header (Page header) -->
5 <section class="content-header">
6   <h1>
7     {{ empty($result) ? 'Tambah' : 'Edit' }} Data Kelas
8     <small>SMK Negeri 4 Bandung</small>
9   </h1>
10  <ol class="breadcrumb">
11    <li><a href="{{ url('/') }}"><i class="fa fa-dashboard"></i> Home</a></li>
12    <li>Data Kelas</li>
13    <li class="active">{{ empty($result) ? 'Tambah' : 'Edit' }} Data Kelas</li>
14  </ol>
15 </section>
16
17 <!-- Main content -->
18 <section class="content">
19   @include('templates/_feedback')
20
21   <!-- Default box -->
22   <div class="box">
23     <div class="box-header with-border">
24       <a href="{{ url('/')}}" class="btn bg-purple"><i class="fa fa-chevron-left"></i>
```

```

25 </div>
26 <div class="box-body">
27   <form action="{{ $result ? url('kelas/add') : url("kelas/$result->id_kelas/edit") }}" class="form-horizontal" method="POST">
28     {{ csrf_field() }}
29
30     @if (!$result)
31       {{ method_field('patch') }}
32     @endif
33     <div class="form-group">
34       <label class="control-label col-sm-2">Nama Kelas</label>
35       <div class="col-sm-10">
36         <input type="text" name="nama_kelas" class="form-control" placeholder="Masukkan Nama Kelas" value="{{ @$result->nama_kelas }}" />
37       </div>
38     </div>
39
40     <div class="form-group">
41       <label class="control-label col-sm-2">Jurusan</label>
42       <div class="col-sm-10">
43         <input type="text" name="jurusan" class="form-control" placeholder="Masukkan Jurusan" value="{{ @$result->jurusan }}" />
44       </div>
45     </div>

```

PHP Hypertext Preprocessor file length : 2.021 lines : 60 Ln : 15 Col : 15 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

Notes:

- Pada sintaks diatas, dilakukan cek apabila ada data \$result, jika ada maka mode form adalah mode edit data. Jika tidak mode form adalah mode tambah data.
- Method field yang digunakan adalah patch untuk melakukan update data.
- Pada kolom value, terdapat sintaks {{ @\$result->nama_kelas }}, fungsi @ didepan adalah untuk men-hide warning jika data \$result tidak ada.

2. Buka file **app/Http/Controllers/KelasController.php**, tambahkan sintaks untuk fungsi **edit** dan **update**:

```

33 }
34
35 public function edit($id)
36 {
37   $data['result'] = \App\Kelas::where('id_kelas', $id)->first();
38   return view('kelas/form')->with($data);
39 }
40
41 public function update(Request $request, $id)
42 {
43   $rules = [
44     'nama_kelas'    => 'required|max:100',
45     'jurusan'       => 'required|max:100'
46   ];
47   $this->validate($request, $rules);
48
49   $input = $request->all();
50   $result = \App\Kelas::where('id_kelas', $id)->first();
51   $status = $result->update($input);
52
53   if ($status) return redirect('/')->with('success', 'Data berhasil diubah');
54   else return redirect('/')->with('error', 'Data gagal diubah');
55 }
56

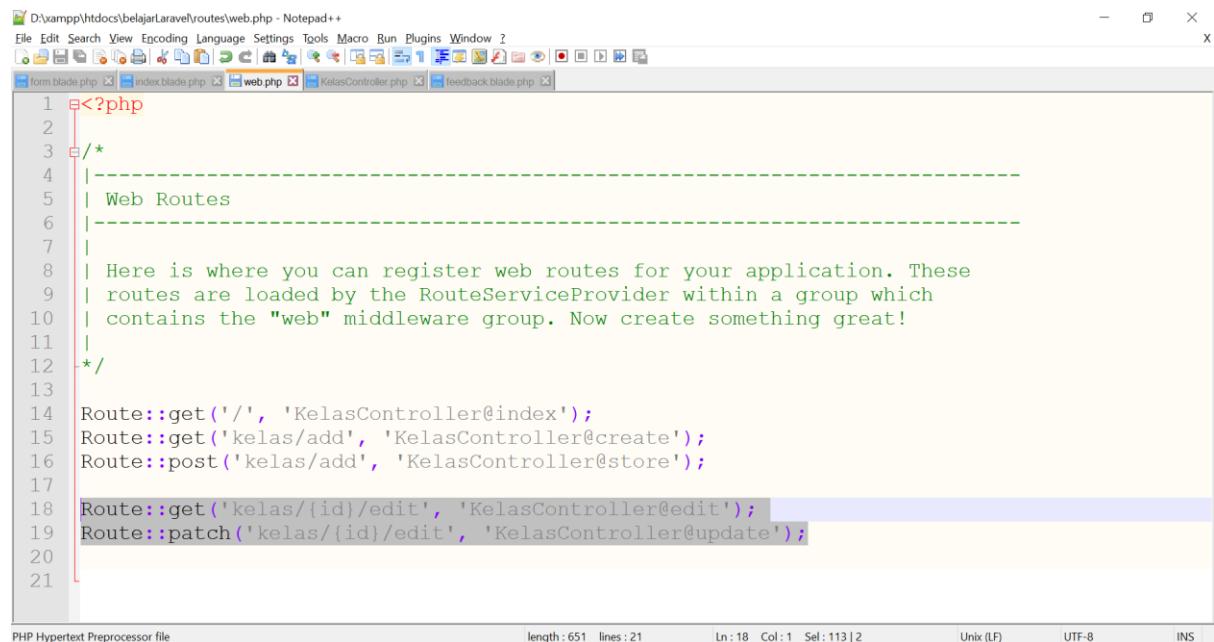
```

PHP Hypertext Preprocessor file length : 1.311 lines : 56 Ln : 56 Col : 2 Sel : 0 | 0 Unix (LF) UTF-8 INS

Notes:

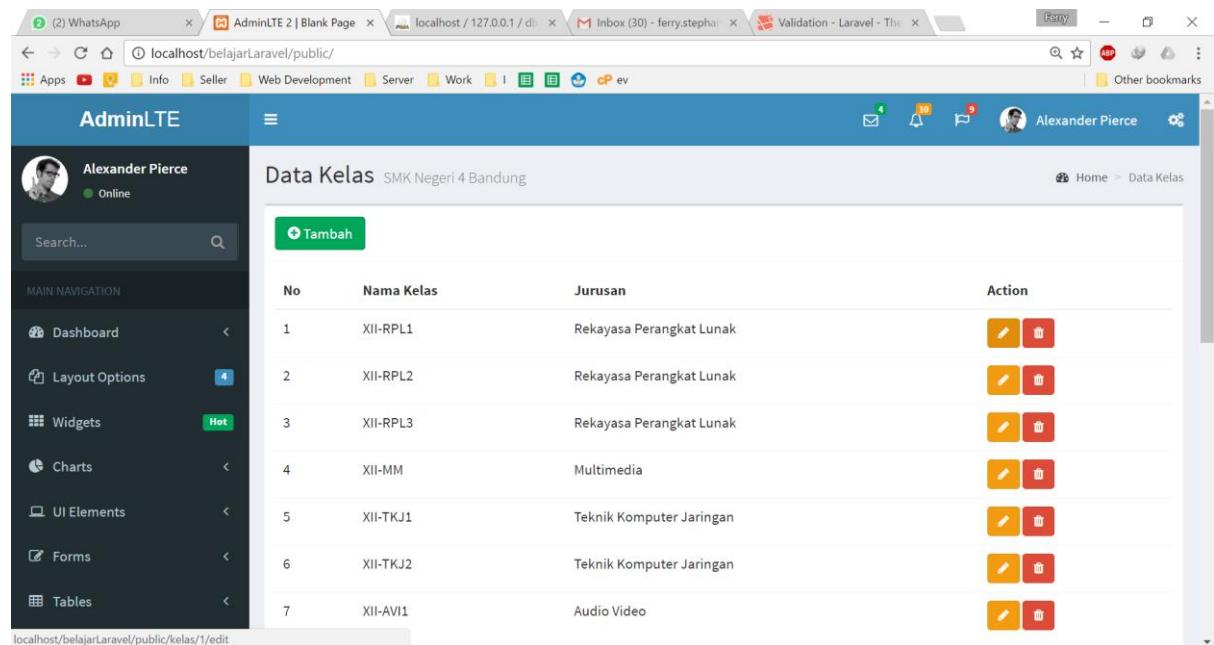
- \App\Kelas::where digunakan untuk mencari data dengan kondisi yang ada pada parameter nya.
- Fungsi first() digunakan untuk mengambil row pertama dari data Kelas.

3. Buka file **routes/web.php**, tambahkan sintaks berikut ini:



```
D:\xampp\htdocs\belajarLaravel\routes\web.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 2
form blade.php index blade.php web.php KelasController.php feedback blade.php
1 <?php
2 /*
3  * Web Routes
4  */
5
6 | Here is where you can register web routes for your application. These
7 | routes are loaded by the RouteServiceProvider within a group which
8 | contains the "web" middleware group. Now create something great!
9 |
10 */
11
12 Route::get('/', 'KelasController@index');
13 Route::get('kelas/add', 'KelasController@create');
14 Route::post('kelas/add', 'KelasController@store');
15
16 Route::get('kelas/{id}/edit', 'KelasController@edit');
17 Route::patch('kelas/{id}/edit', 'KelasController@update');
```

4. Buka kembali web anda, coba klik tombol edit, hasilnya adalah seperti tampilan pada gambar ini:



The screenshot shows a browser window with multiple tabs open. The active tab is 'localhost/belajarLaravel/public/'. The page displays a table titled 'Data Kelas' for SMK Negeri 4 Bandung. A single row is selected for editing, indicated by a blue border around the second column. The table has columns: No, Nama Kelas, Jurusan, and Action. The 'Action' column contains edit and delete icons.

No	Nama Kelas	Jurusan	Action
1	XII-RPL1	Rekayasa Perangkat Lunak	
2	XII-RPL2	Rekayasa Perangkat Lunak	
3	XII-RPL3	Rekayasa Perangkat Lunak	
4	XII-MM	Multimedia	
5	XII-TKJ1	Teknik Komputer Jaringan	
6	XII-TKJ2	Teknik Komputer Jaringan	
7	XII-AVI1	Audio Video	

AdminLTE

Edit Data Kelas SMK Negeri 4 Bandung

Kembali

Nama Kelas: XII-RPL1

Jurusan: RPL

Simpan

AdminLTE

Data Kelas SMK Negeri 4 Bandung

Data berhasil diubah

Tambah

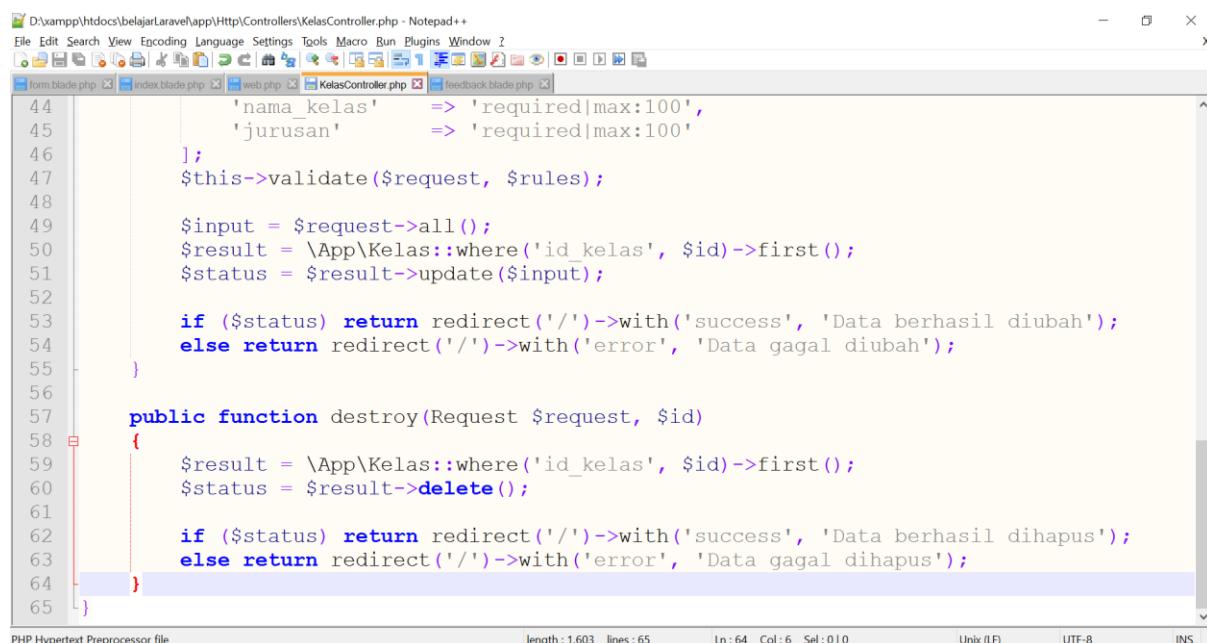
No	Nama Kelas	Jurusan	Action
1	XII-RPL1	RPL	

CRUD – Delete Data

Pada bagian ini akan dijelaskan langkah-langkah untuk menghapus data. Caranya cukup mudah, pada tahap sebelumnya kita sudah membuat tombol untuk hapus dengan menggunakan form dan method field **DELETE**.

Ikuti langkah-langkah berikut ini:

1. Buka file **app/Http/Controllers/KelasController.php**, tambahkan sintaks untuk fungsi **destroy**.



The screenshot shows a Notepad++ window with the file 'KelasController.php' open. The code is a PHP class definition for 'KelasController'. It includes validation rules for 'nama_kelas' and 'jurusan', a validation method call, and two methods: 'update' and 'destroy'. The 'destroy' method uses the delete() method on the \$result object to delete a record from the database.

```
D:\xampp\htdocs\belajarLaravel\app\Http\Controllers\KelasController.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 2
form.blade.php index.blade.php web.php KelasController.php feedback.blade.php
44     'nama_kelas' => 'required|max:100',
45     'jurusan'      => 'required|max:100'
46 ];
47 $this->validate($request, $rules);
48
49 $input = $request->all();
50 $result = \App\Kelas::where('id_kelas', $id)->first();
51 $status = $result->update($input);
52
53 if ($status) return redirect('/')->with('success', 'Data berhasil diubah');
54 else return redirect('/')->with('error', 'Data gagal diubah');
55 }
56
57 public function destroy(Request $request, $id)
58 {
59     $result = \App\Kelas::where('id_kelas', $id)->first();
60     $status = $result->delete();
61
62     if ($status) return redirect('/')->with('success', 'Data berhasil dihapus');
63     else return redirect('/')->with('error', 'Data gagal dihapus');
64 }
65 }
```

Notes:

- `$result->delete()` digunakan untuk menghapus data.

2. Buka kembali web anda, coba klik tombol **delete**, hasilnya adalah seperti tampilan pada gambar ini:

The image contains two screenshots of a Laravel application interface. Both screenshots show the same basic layout: a dark sidebar on the left with a user profile picture and 'Alexander Pierce' status, and a main content area titled 'Data Kelas' for 'SMK Negeri 4 Bandung'. The main content area includes a 'Tambah' button and a table with four rows of data. In the first screenshot, the table data is:

No	Nama Kelas	Jurusan	Action
1	XII-RPL1	RPL	[Edit] [Delete]
2	XII-RPL2	Rekayasa Perangkat Lunak	[Edit] [Delete]
3	XII-RPL3	Rekayasa Perangkat Lunak	[Edit] [Delete]
4	XII-MM	Multimedia	[Edit] [Delete]

In the second screenshot, a green notification bar at the top of the main content area says 'Data berhasil dihapus'. The table data is identical to the first screenshot.