

Laravel

Laravel adalah sebuah framework yang digunakan untuk web development di PHP.

A. INSTAL LARAVEL

Instalasi Laravel di Windows terbilang cukup mudah dibandingkan dengan distro Linux. Sebelum melakukan instalasi Laravel ada baiknya mengecek apakah ekstensi openssl di php.ini sudah diaktifkan atau belum karena Laravel memerlukan ekstensi ini dalam mode CLI, jika belum silahkan buka php.ini untuk mengaktifkan ekstensi openssl dan cari tulisan `;extension=php_openssl.dll` kemudian hilangkan tanda titik koma (;) didepannya dan simpan perubahan pada php.ini. Jika langkah satu ini dilewati maka akan keluar pesan berikut ini: `Some settings on your machine make Composer unable to work properly. Make sure that you fix the issues listed below and run this script again: The openssl extension is missing, which means that secure HTTPS transfers are impossible. If possible you should enable it or recompile php with --with-openssl`

- B. Langkah selanjutnya adalah kita memerlukan Composer untuk instalasi dan depensi manajemen framework Laravel, silahkan download Composer-Setup.exe Terlebih dahulu dan install.

Apabila anda sudah menginstal laravel rename nama folder laravel anda dengan nama 'laravel' dan buka file laravel tersebut dengan memanggil : `localhost/laravel/public` Dan hasilnya sebagai berikut :



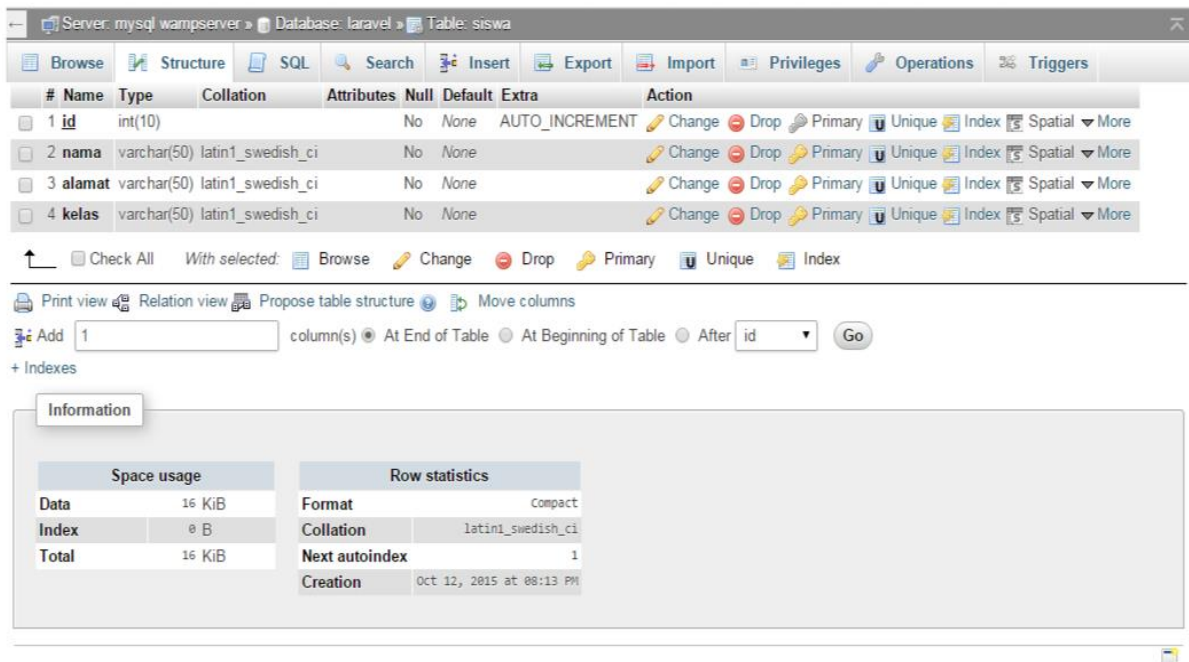
Laravel 5

apabila nama folder sudah di ubah dan sudah dibuka , siapkan database untuk pembuatan crud di laravel anda .

C. MEMBUAT DATABASE

Buat sebuah database dengan nama laravel apabila database sudah dibuat selanjutnya untuk belajar laravel kita buat tabel dengan nama 'siswa' Field diantaranya :

id(int)10 autoincrement,nama(varchar)50,alamat(varchar)50,kelas(varchar)50



apabila field sudah dibuat dengan benar tahap selanjut nya buka file laravel dan setting database di laravel

D. SETTING DATABASE DI LARAVEL

Untuk database di laravel cukup mudah buka folder laravel anda lalu buka file yang berekstensi .env setting database seperti berikut .

```
DB_HOST=localhost
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=
```

E. MENGHAPUS /PUBLIC DI LARAVEL

Untuk tahap berikutnya kita akan mencoba menghapus /public di laravel , karena pada saat kita instal laravel kita harus membuka file laravel dengan cara localhost/laravel/public , yang menjadi pertanyaan bagaimana cara public nya dihilangkan dari url tersebut, cukup mudah

untuk menghilangkan public tersebut . Pertama yang harus anda lakukan buat folder terlebih dahulu dengan nama 'privasi' folder privasi ini bertujuan untuk menyimpan data data laravel anda kecuali data public , apabila anda sudah membuat folder public langkah berikutnya cut file laravel anda kecuali folder 'public' apabila sudah di cut masukan file tersebut ke folder privasi , apabila sudah yang tersisa hanya ada 2 folder , folder public dan folder privasi langkah selanjutnya buka folder public dan cut semua file di folder public lalu paste kan di luar , Apabila folder laravel anda sudah dibuat percis seperti gambar di atas langkah selanjutnya adalah mengubah index.php di file tersebut menjadi seperti gambar berikut :

```
<?php

/**
 * Laravel - A PHP Framework For Web Artisans
 *
 * @package  Laravel
 * @author   Taylor Otwell <taylorotwell@gmail.com>
 */

/**
 * Register The Auto Loader
 *
 * Composer provides a convenient, automatically generated class loader for
 * our application. We just need to utilize it! We'll simply require it
 * into the script here so that we don't have to worry about manual
 * loading any of our classes later on. It feels nice to relax.
 */

require __DIR__.'/privasi/bootstrap/autoload.php';

/**
 * Turn On The Lights
 *
 * We need to illuminate PHP development, so let us turn on the lights.
 * This bootstraps the framework and gets it ready for use, then it
 * will load up this application so that we can run it and send
 * the responses back to the browser and delight our users.
 */

$app = require_once __DIR__.'/privasi/bootstrap/app.php';

/**
 * Run The Application
 *
 * Once we have the application, we can handle the incoming request
 * through the kernel, and send the associated response back to
 * the client's browser allowing them to enjoy the creative
 * and wonderful application we have prepared for them.
 */

$kernel = $app->make(Illuminate\Contracts\Http\Kernel::class);

$response = $kernel->handle(
    $request = Illuminate\Http\Request::capture()
);

$response->send();

$kernel->terminate($request, $response);
```

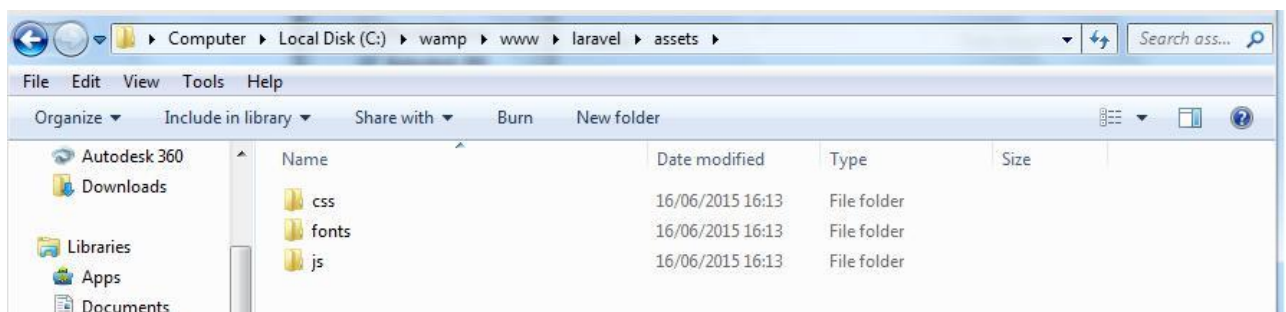
Ubahlah konfigurasi index.php anda seperti gambar diatas , apabila sudah buka browser anda dan coba untuk membuka url : localhost/laravel dan tara url anda sudah berganti menjadi localhost/laravel tidak lagi menjadi localhost/laravel/public



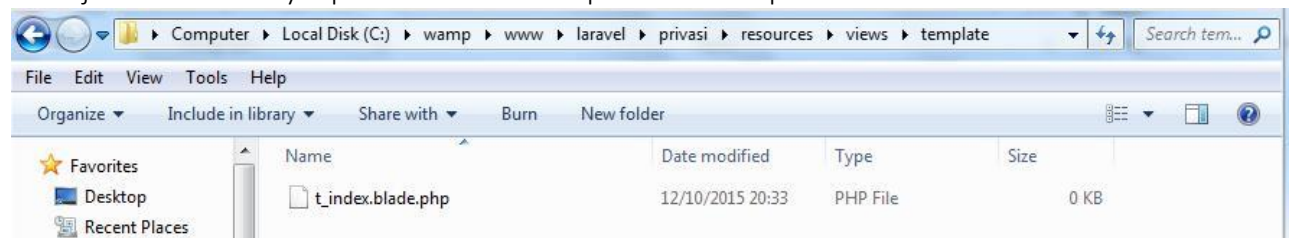
Laravel 5

F. INTEGRASI BOOTSTRAP DI LARAVEL

Untuk mengintegrasikan bootstrap di laravel anda harus mendownload terlebih dahulu file bootstrap , buka link berikut : <http://getbootstrap.com>, apabila bootstrap sudah didownload extract file bootstrap tersebut dan ambil folder css,js,dan fonts di bootstrap , agar terlihat rapih buat folder assets di laravel anda lalu masukan file bootstrap kedalam folder assets anda



Folder bootstrap sudah dimasukan ke laravel anda selanjut nya buka folder laravel anda lalu masuk ke folder privasi lalu masuk lagi ke folder resource masuk lagi ke folder views dan buat lah folder template yang bertujuan untuk membuat template bootstrap dan jangan lupa download terlebih dahulu jquery.min.js dan masukan ke folder js , apabila anda sudah membuat folder template buat file yang bernama t_index.blade.php file ini bertujuan untuk menyimpan link rel dan template bootstrap



File sudah dibuat selanjutnya menambah code di file tersebut , tulis kode di file t_index.blade.php seperti gambar berikut :

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Laravel 5.1</title>
  {!! Html::style('assets/css/bootstrap.min.css') !!}
</head>
<body>

  @yield('content')

  {!! HTML::script('assets/js/jquery.min.js') !!}
  {!! HTML::script('assets/js/bootstrap.min.js') !!}
</body>
</html>
```

Selanjutnya buka file routes.php di privasi>app>http>routes.php ubah return view(welcome); menjadi return view('home');

ini bertujuan agar file yang di buka pertama itu adalah [home.index.blade.php](#) selanjutnya buat file [home.index.blade.php](#) di privasi>resources>views>[home.index.blade.php](#) dan masukan code seperti gambar dibawah ini :

```
@extends('template/t_index')

@section('content')

BELAJAR LARAVEL 5.1
```

extends : untuk mengambil template bootstrap ,

@section untuk memasukan data [home.blade.php](#) ke template bootstrap tersebut apabila sudah selesai coba buka di browser anda : localhost/laravel dan akan menemukan eror tersebut , eror tersebut adalah kita belum memasukan illuminate di laravel , cara untuk memasukkannya adalah buka file composer.json di privasi>composer.json lalu tambahkan "illuminate/html": "5.*"

```
type: project,  
"require": {  
    "php": ">=5.5.9",  
    "laravel/framework": "5.1.*",  
    "illuminate/html": "5.*"  
},
```

apabila composer.json sudah seperti gambar di atas lalu buka cmd kita masuk ke folder laravel kita lalu masuk ke privasi ketik : composer update tunggu beberapa saat composer telah terupdate , selanjutnya buka file app.php di privasi>config>app.php lalu buka file tersebut tambah kan Illuminate\Html\HtmlServiceProvider::class,

Dan tambahkan juga

```
'aliases' => [  
    'App' => Illuminate\Support\Facades\App::class,  
    'Artisan' => Illuminate\Support\Facades\Artisan::class,  
    'Auth' => Illuminate\Support\Facades\Auth::class,  
    'Blade' => Illuminate\Support\Facades\Blade::class,  
    'Bus' => Illuminate\Support\Facades\Bus::class,  
    'Cache' => Illuminate\Support\Facades\Cache::class,  
    'Config' => Illuminate\Support\Facades\Config::class,  
    'Cookie' => Illuminate\Support\Facades\Cookie::class,  
    'Crypt' => Illuminate\Support\Facades\Crypt::class,  
    'DB' => Illuminate\Support\Facades\DB::class,  
    'Eloquent' => Illuminate\Database\Eloquent\Model::class,  
    'Event' => Illuminate\Support\Facades\Event::class,  
    'File' => Illuminate\Support\Facades\File::class,  
    'Gate' => Illuminate\Support\Facades\Gate::class,  
    'Hash' => Illuminate\Support\Facades\Hash::class,  
    'Input' => Illuminate\Support\Facades\Input::class,  
    'Inspiring' => Illuminate\Foundation\Inspiring::class,  
    'Lang' => Illuminate\Support\Facades\Lang::class,  
    'Log' => Illuminate\Support\Facades\Log::class,  
    'Mail' => Illuminate\Support\Facades\Mail::class,  
    'Password' => Illuminate\Support\Facades>Password::class,  
    'Queue' => Illuminate\Support\Facades\Queue::class,  
    'Redirect' => Illuminate\Support\Facades\Redirect::class,  
    'Redis' => Illuminate\Support\Facades\Redis::class,  
    'Request' => Illuminate\Support\Facades\Request::class,  
    'Response' => Illuminate\Support\Facades\Response::class,  
    'Route' => Illuminate\Support\Facades\Route::class,  
    'Schema' => Illuminate\Support\Facades\Schema::class,  
    'Session' => Illuminate\Support\Facades\Session::class,  
    'Storage' => Illuminate\Support\Facades\Storage::class,  
    'URL' => Illuminate\Support\Facades\URL::class,  
    'Validator' => Illuminate\Support\Facades\Validator::class,  
    'View' => Illuminate\Support\Facades\View::class,  
    'Form' => Illuminate\Html\FormFacade::class,  
    'Html' => Illuminate\Html\HtmlFacade::class,  
],
```



```
'Form' => Illuminate\Html\FormFacade::class, 'Html'
=> Illuminate\Html\HtmlFacade::class,
```

Apabila sudah selesai menambahkan coba buka localhost/laravel dan tara sudah berjalan dengan baik.



Selanjutnya kita membuat form untuk membuat data .

PEMBUATAN CRUD DI LARAVEL

1. Create / Penambahan Data Di Laravel

Membuat Create di laravel atau bisa di sebut dengan menambah data ke database (CREATE) tahap yang harus di lakukan pertama buka file [home.blade.php](#) ini bertujuan untuk menampilkan form yang akan di input dan dimasukan ke database untuk datanya

,
isikan [home.blade.php](#) seperti gambar dibawah ini :

```
home.blade.php
@extends('template/t_index')
@section('content')

<div class="container">
<p></p>
<div class="panel panel-default">
  <div class="panel-heading">
    Tambah Data </div>
  <div class="panel-body">
    {!! Form::open(['url' => '/prosestambah']) !!}
    Nama:
    {!! Form::text('nama', '', ['placeholder' => 'Nama','class' => 'form-control']) !!}
    Alamat:
    {!! Form::text('alamat', '', ['placeholder' => 'Alamat','class' => 'form-control']) !!}
    Kelas:
    {!! Form::text('kelas', '', ['placeholder' => 'Kelas','class' => 'form-control']) !!}
  <p></p>
  {!! Form::submit('Tambah Data', ['class' => 'btn btn-danger']) !!}
  {!! Form::close() !!}
  @stop
</div>
</div>
</div>
```

@extends('template/t_index') : ini bertujuan mendapatkan design dari template bootstrap

@section('content') : ini bertujuan untuk menampilkan content

@form::open : ini sama dengan form method post

@form:text : sama dengan dengan input type text

@stop : untuk menghentikan code tersebut

sebetulnya memakai form method juga bisa tetapi di laravel kita sudah di sediakan untuk pembuatan yang lebih simple dan sedikit , itu tergantung lagi kepada anda , anda ingin memakai yang mana ?

apabila sudah sama seperti gambar diatas selanjutnya kita coba tampilkan dan hasilnya akan sama seperti gambar dibawah ini :



The screenshot shows a web browser window with the address bar displaying 'localhost/laravel/'. The main content area contains a form titled 'Tambah Data'. The form has three input fields: 'Nama', 'Alamat', and 'Kelas'. Below the 'Kelas' field is a red button labeled 'Tambah Data'.

pada tahap ini kita akan membuat controller terlebih dahulu berfungsi untuk menyambungkan antara routes dan controller tersebut

untuk caranya silahkan buka cmd terlebih dahulu lalu masuk ke folder laravel anda dan masuk ke folder privasi dengan cara

cd C:\wamp\www\laravel\privasi (tergantung folder anda disimpan) apabila sudah masukan code ini untuk membuat controller anda di laravel : php artisan make:controller Crudcontroller dan hasilnya akan seperti berikut :

```
C:\wamp\www\laravel\privasi>php artisan make:controller Crudcontroller
Controller created successfully.
```


ini artinya controller sudah di buat silahkan cek ke folder privasi>app>http>controllers dan file yang bernama Crudcontroller sudah berhasil di buat , tahap selanjutnya buka file routes.php di privasi>app>http>routes.php apabila file sudah di buka tambah kan kode seperti gambar berikut :

```
Route::post('prosestambah', 'Crudcontroller@tambahdata');
```

post itu untuk memproses data sedangkan get untuk menampilkan data itu perbedaannya sedangkan prosestambah yang ada di gambar itu menunjukan class/function nya apa , dan crudcontroller@tambahdata itu proses nya akan kemana cukup mengerti ? apabila tidak mengerti akan saya lanjutkan terlebih dahulu sekarang coba anda buka file Crudcontroller yang sudah anda buat dan masukan data seperti gambar berikut :

```
public function tambahdata() {  
  
    $data = array(  
        'nama' => Input::get('nama'),  
        'alamat' => Input::get('nama'),  
        'kelas' => Input::get('nama'),  
    );  
  
    DB::table('siswa')->insert($data);  
    return Redirect::to('/read')->with('message', 'berhasil menambah data');  
}
```

kita buka lagi file routes.php lalu tambah kan code seperti gambar berikut :

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

sedikit saya jelaskan kita harus membuat route::get tersebut yang bertujuan untuk menampilkan file read.blade.php , sekarang kita buat juga file read.blade.php di privasi>resources>views>read.blade.php apabila file sudah di buat tambahkan code berikut di file tersebut :

```
@extends('template/t_index')  
@section('content')  
  
@if(Session::has('message'))  
    <span class="label label-success">{{ Session::get('message') }}</span>  
@endif  
  
@stop
```

apabila sudah sesuai dengan code tersebut kita langsung membuat data di home tadi caranya buka browser anda : localhost/laravel dan coba masukan data apakah data nya akan masuk ke database atau tidak ,

pada saat anda mencoba memasukan data pasti akan muncul error seperti gambar gambar dibawah ini : , input not found , db notfound , redirect not found

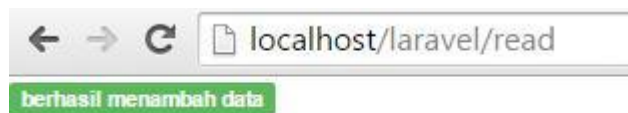
1/1 FatalErrorException in Crudcontroller.php line 31:
Class 'App\Http\Controllers\Input' not found

lalu bagaimana cara mengatasi eror berikut ? untuk mengatasi nya cukup mudah caranya buka file Crudcontroller lalu masukan code berikut

```
<?php  
  
namespace App\Http\Controllers;  
  
use Illuminate\Http\Request;  
use App\Http\Requests;  
use App\Http\Controllers\Controller;  
use Input;  
use DB;  
use Redirect;
```

tambahkan use input; use db; use redirect;

ini bertujuan agar eror yang tadi di alami kembali normal , nah apabila anda menemukan eror seperti di atas lalukan hal yang sama memakai use, sekarang coba anda praktekan localhost/laravel dan masukan data sesuai yang anda input dan hasilnya akan seperti berikut :



yee akhirnya menambah data berhasil :D sekarang langkah selanjutnya kita membuat Read data / menampilkan data yang sudah kita inputkan

Read / Menampilkan Data Di Laravel

tahap yang pertama ubah code read yang ada di routes , buka routes.php dan ubah code read seperti gambar di bawah ini :

```
Route::get('read','Crudcontroller@lihatdata');
```

apabila sudah memodifikasi read tersebut sekarang buka file Crudcontroller lalu tambahkan function lihatdata seperti gambar berikut bertujuan untuk menampilkan data data yang ada di database:

```
public function lihatdata(){  
    $data = DB::table('siswa')->get();  
    return View::make('read')->with('siswa',$data);  
}
```

apabila script sudah jalan tinggal kita modifikasi file read.blade.php , buka file tersebut dan modifikasi code tersebut seperti gambar di bawah ini :

```
@extends('template/t_index')  
@section('content')  
  
<div class="container">  
    @if(Session::has('message'))  
        <span class="label label-success">{{ Session::get('message') }}</span>  
    @endif  
    <p></p>  
    <div class="table-responsive">  
        <table class="table table-bordered">  
            <tr>  
                <th>No</th>  
                <th>Nama</th>  
                <th>Alamat</th>  
                <th>Kelas</th>  
                <th>Action</th>  
            </tr>  
            <?php $no=1;?>  
            @foreach ($siswa as $data)  
                <tr>  
                    <td>{{ $no++ }}</td>  
                    <td>{{ $data->nama }}</td>  
                    <td>{{ $data->alamat }}</td>  
                    <td>{{ $data->kelas }}</td>  
                    <td><a href="">Hapus</a>  
                    || <a href="">Edit</a></td>  
                </tr>  
            @endforeach  
        </table>  
    </div>  
</div>  
  
@stop
```

\$siswa tersebut mengambil nilai variable yang ada di Crudcontroller tadi , nah {{ }} sama dengan <?php echo ?> perbedaannya {{ }} agar lebih simple dan sedikit cmiwiw :D apabila sudah silahkan lihat localhost/laravel/read dan hasilnya kurang lebih akan seperti gambar dibawah ini :



No	Nama	Alamat	Kelas	Action
1	dadan	dadan	dadan	Hapus Edit
2	ad	ad	ad	Hapus Edit
3	Dadan	Dadan	Dadan	Hapus Edit

Delete / Menghapus Data Di Laravel

```
Route::get('hapus/{id}', 'Crudcontroller@hapusdata');
```

ini bertujuan untuk menghapus data selanjutnya buka juga file Crudcontroller dan tambahkan function berikut di file Crudcontroller

```
public function hapusdata($id) {
    DB::table('siswa')->where('id', '=', $id)->delete();
    return Redirect::to('/read')->with('message', 'berhasil menghapus data');
}
```

id }}">Hapus

cukup di mengerti dan mudah bukan untuk membuat action hapus di laravel ? dan hasilnya akan terlihat seperti gambar berikut :



No	Nama	Alamat	Kelas	Action
1	dadan	dadan	dadan	Hapus Edit
2	Dadan	Dadan	Dadan	Hapus Edit

Update / Mengubah Data Di Laravel

pertama buka file routes.php dan tambah route seperti gambar dibawah ini :

```
Route::get('formedit/{id}', 'Crudcontroller@editdata');
```

apabila sudah selesai sekarang buka file Crudcontroller.php dan tambah kan function seperti gambar berikut :

```
public function editdata($id) {  
  
    $data = DB::table('siswa')->where('id','=',$id)->first();  
  
    return View::make('form_edit')->with('siswa',$data);  
  
}
```

first di atas itu untuk mengambil nilai pertama berdasarkan id yang kita panggil apabila sudah menyimpan function tersebut , sekarang buat file yang bernama form_edit.blade.php di privasi>resources>views>form_edit.blade.php apabila

file sudah di buat sekarang tambah kan code seperti gambar dibawah ini ke dalam file form_edit.blade.php

```
@extends('template/t_index')  
@section('content')  
  
    <div class="container">  
        @if(Session::has('message'))  
            <span class="label label-success">{{ Session::get('message') }}</span>  
        @endif  
        <p></p>  
        <div class="panel panel-default">  
            <div class="panel-heading">  
Ubah Data </div>  
            <div class="panel-body">  
                {!! Form::open(['url' => '/prosesedit']) !!}   
                {!! Form::hidden('id', $siswa->id, ['class' => 'form-control']) !!}   
                Nama:  
                {!! Form::text('nama', $siswa->nama, ['class' => 'form-control']) !!}   
                Alamat:  
                {!! Form::text('alamat', $siswa->alamat, ['class' => 'form-control']) !!}   
                Kelas:  
                {!! Form::text('kelas', $siswa->kelas, ['class' => 'form-control']) !!}   
                <p></p>  
                {!! Form::submit('Ubah Data', ['class' => 'btn btn-danger']) !!}   
                {!! Form::close() !!}   
            </div>  
        </div>  
    </div>
```

untuk memproses mengubah data , buka kembali file routes.php dan masukan code seperti gambar di bawah ini :

```
Route::post('prosesedit', 'Crudcontroller@proseseditdata');
```

apabila sudah menambah route::post sekarang buka juga Crudcontroller untuk menambah function proseseditdata , dan tambahkan function seperti gambar berikut :

```
public function proseseditdata() {  
    $data = array(  
        'nama' => Input::get('nama'),  
        'alamat' => Input::get('alamat'),  
        'kelas' => Input::get('kelas')  
    );  
  
    DB::table('siswa')->where('id','=',Input::get('id'))->update($data);  
  
    return Redirect::to('read')->with('message','berhasil mengedit data');  
}
```

apabila function sudah di tambahkan waktunya mencoba untuk mengedit data buka localhost/laravel/read dan coba klik a href edit yang sudah tertera disana dan ubah data data tersebut dan hasilnya akan seperti berikut

← → ↻ localhost/laravel/read

berhasil mengedit data

No	Nama	Alamat	Kelas	Action
1	a	a	a	Hapus Edit
2	Dadan	Dadan	Dadan	Hapus Edit