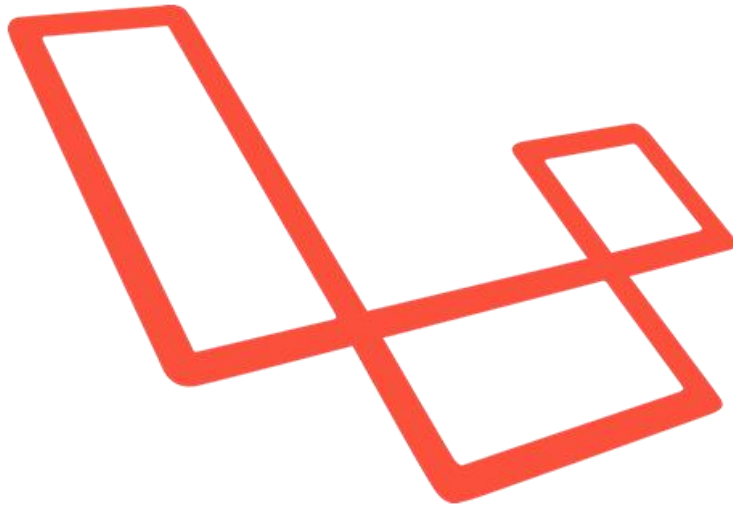


BELAJAR LARAVEL UNTUK PEMULA



Perkenalkan nama saya Dadan Hamdani , untuk kali ini kita akan belajar framework Laravel , pasti di antara kalian sudah pada tau apa itu laravel bagaimana cara kerja laravel , untuk yang belum tau apa itu laravel kali ini akan saya jelaskan dan akan saya praktekan cara kerja laravel , mohon maaf apabila kata kata kurang jelas , selamat belajar LARAVEL!

DAFTAR ISI

| | |
|---|-------------|
| 1. Pengertian Laravel | Hal : 3 |
| 2. Cara Instal Laravel | |
| 2.1. Instal Laravel Versi Windows | Hal : 3-4 |
| 2.2. Instal Laravel Versi Ubuntu/Linux | Hal : 5-6 |
| 3. Persiapan Pembuatan CRUD Di Laravel | |
| 3.1. Persiapkan File | Hal : 7 |
| 3.2. Pembuatan Database | Hal : 7-8 |
| 3.3. Setting Database di Laravel | Hal : 8 |
| 3.4. Menghapus Public Di Laravel | Hal : 8-10 |
| 3.5 Integrasi Bootstrap Di Laravel | Hal : 10-14 |
| 4. Pembuatan CRUD Di Laravel | |
| 4.1. Create / Penambahan Data Di Laravel | Hal : 15-18 |
| 4.2. Read / Menampilkan Data Di Laravel | Hal : 19-20 |
| 4.3. Delete / Menghapus Data Di Laravel | Hal : 21 |
| 4.4. Update / Mengubah Data Di Laravel | Hal : 22-24 |
| 5. Login Di Laravel | |
| 5.1. Pembuatan Halaman Login Di Laravel | Hal : 25-26 |
| 5.2. Penambahan Table Di Database | Hal : 26 |
| 5.3. Pembuatan Halaman Register Di Laravel | Hal : 26-28 |
| 5.4. Proses Login Di Laravel | Hal : 28-30 |
| 5.5. Memberikan Hak Akses Tertentu Di Laravel | Hal : 30-34 |
| 6. Validasi Di Laravel | |
| 6.1 Validasi Login Laravel | Hal : 35-37 |
| 6.2 Validasi Register Laravel | Hal : 37-39 |
| 6.3 Validasi Tambah Data Laravel | Hal : 39-40 |
| 7. Kesimpulan | Hal : 41 |
| 8. Tentang Penulis | Hal : 42 |

1. APA ITU LARAVEL ?

Laravel adalah sebuah framework yang digunakan untuk web development di PHP. Umur framework ini, terbilang masih baru (walaupun sudah mulai beranjak ke versi 5), dan mungkin seumuran dengan PhalconPHP.

2. INSTAL LARAVEL

2.1. INSTAL VERSI WINDOWS

LANGKAH PERTAMA

Instalasi Laravel di Windows terbilang cukup mudah dibandingkan dengan distro Linux. Sebelum melakukan instalasi Laravel ada baiknya sahabat 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.

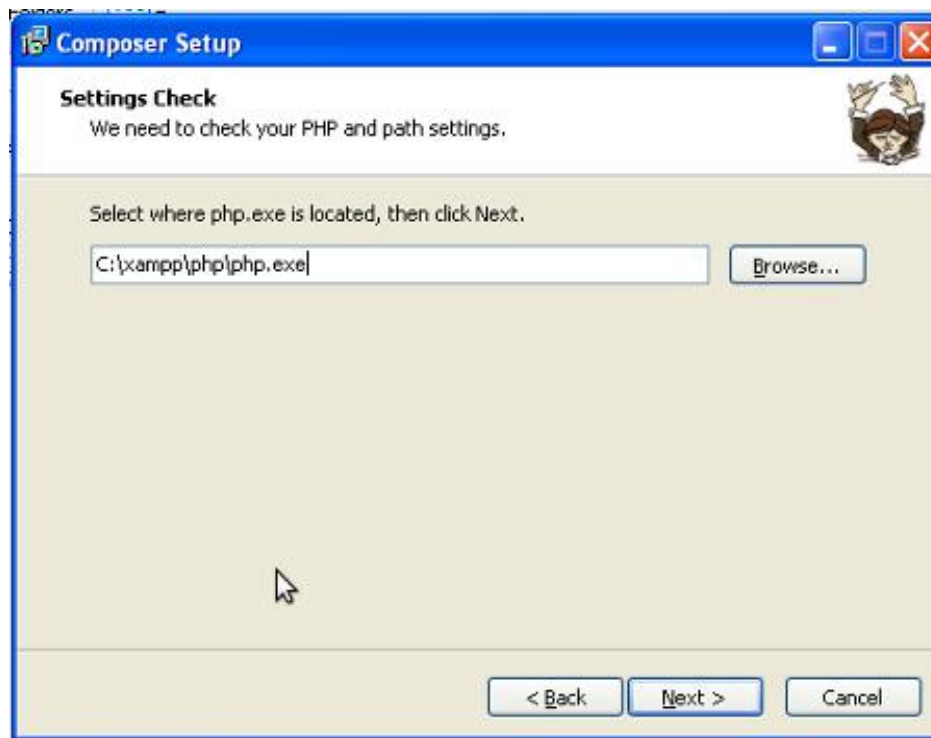
Make sure that you fix the issues listed below and run this script again:

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

LANGKAH KEDUA

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



Pada gambar di atas lokasi PHP saya adalah di C:\xampp\php\php.exe, nah silahkan sesuaikan dengan lokasi PHP sahabat.

LANGKAH KETIGA

Buka command prompt, kemudian pindah ke direktori htdocs dengan ketik `cd C:\xampp\htdocs` (silahkan sesuaikan dengan lokasi htdocs sahabat). Ketikkan perintah di command prompt sebagai berikut untuk instalasi Laravel versi terbaru.

```
composer create-project laravel/laravel nama-proyek --prefer-dist
```

Silahkan ganti nama-proyek sesuai dengan keinginan sahabat, sekali lagi perlu diingat bahwa sahabat perlu terkoneksi ke internet untuk melakukan instalasi. Setelah instalasi selesai silahkan buka Laravel di lokasi C:\xampp\htdocs\nama-proyek

Catatan: Jika sahabat sudah melakukan langkah 1-3, untuk berikutnya bila mau instalasi Laravel cukup lakukan langkah 3 saja.

2. INSTAL VERSI LINUX / UBUNTU

Untuk distro Linux agak terasa rumit memang, tetapi asik. Perlu sahabat ketahui bahwa saya menggunakan XAMPP for Linux dan sudah terinstal di direktori /opt/lampp, saya tidak tahu bagaimana dengan sahabat? Oleh karena itu saya akan menjelaskan konfigurasi sesuai dengan bundelan paket web server dari XAMPP tersebut pada langkah 1.

Langkah 1

Saya akan menggunakan PHP dan Curl yang terdapat pada bundelan paket web server dari XAMPP tadi, nah PHP dan Curl tersebut belum terdeteksi di sistem Linux Ubuntu saya, namun apabila sahabat sudah punya instalasi web server dengan PHP dan Curl terdeteksi di sistem silahkan lewati langkah ini menuju langkah 2. Oke, supaya PHP dan Curl terdeteksi di sistem Linux maka ketikan perintah berikut di terminal.

```
$ sudo update-alternatives --install "/usr/bin/php" "php" "/opt/lampp/bin/php" 1
$ sudo update-alternatives --install "/usr/bin/curl" "curl" "/opt/lampp/bin/curl" 1
```

Langkah 2

Sekarang waktunya instal Composer, ketikan perintah berikut di terminal.

```
$ curl -ksS https://getcomposer.org/installer | php
$ sudo mv composer.phar /usr/local/bin/composer
```

Langkah 3

Sekarang waktunya instal Laravel, terlebih dahulu pindah ke direktori htdocs, pada terminal dengan ketik cd htdocs (silahkan sesuaikan dengan lokasi htdocs atau lokasi virtual host sahabat). Ketikan perintah berikut untuk memulai instalasi Laravel versi terbaru.

```
composer create-project laravel/laravel nama-proyek --prefer-dist
```

Silahkan ganti nama-proyek sesuai dengan keinginan sahabat, sekali lagi saya ingatkan bahwa sahabat perlu terkoneksi ke internet untuk melakukan instalasi. Setelah instalasi maka Laravel sudah terinstal di htdocs dengan nama folder nama-proyek.

Langkah 4

Langkah terakhir adalah mengubah permisi direktori storage, ketik di terminal perintah berikut (silahkan disesuaikan dengan direktori sahabat):

```
chmod 777 -R nama-proyek/app/storage
```

Catatan: Jika sahabat sudah melakukan langkah 1-4, untuk berikutnya bila mau instalasi Laravel cukup lakukan langkah 3 dan 4 saja.

Catatan : Laravel 5.1 harus sudah instal PHP Terbaru

3. Persiapan Pembuatan CRUD Di Laravel

3.1. PERSIAPKAN FILE

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

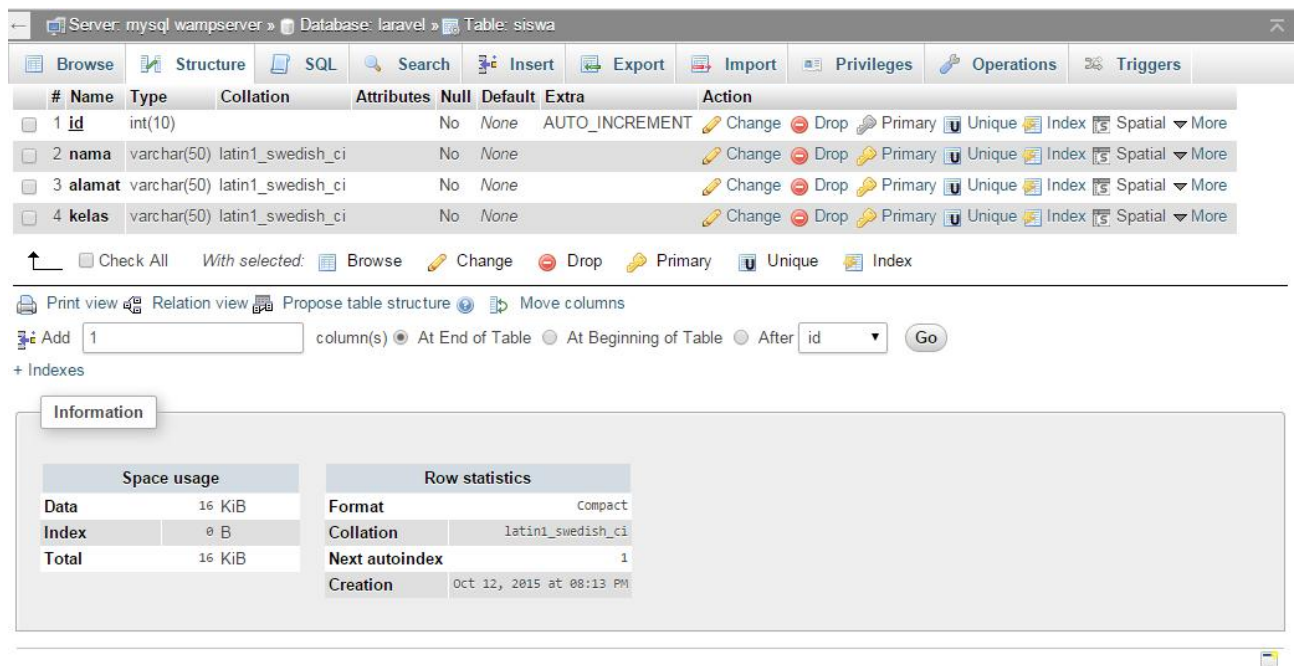
apabila nama folder sudah di ubah dan sudah dibuka , siapkan database untuk pembuatan crud di laravel anda .

3.2. PEMBUATAN 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

3.3. SETTING DATABASE DI LARAVEL

Untuk penyitangan 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=
```

3.4. 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 , untuk lebih jelasnya silahkan lihat gambar berikut :

| Name | Date modified | Type | Size |
|-------------|------------------|---------------|------|
| privasi | 12/10/2015 20:18 | File folder | |
| .htaccess | 12/10/2015 20:03 | HTACCESS File | 1 KB |
| favicon.ico | 12/10/2015 20:03 | Icon | 0 KB |
| index.php | 12/10/2015 20:19 | PHP File | 2 KB |
| robots.txt | 12/10/2015 20:03 | Text Document | 1 KB |

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.
 */

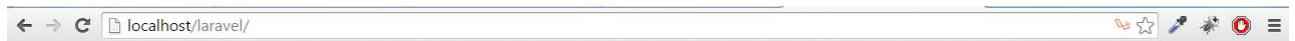
$kernel = $app->make(Illuminate\Contracts\Http\Kernel::class);

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

$response->send();

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

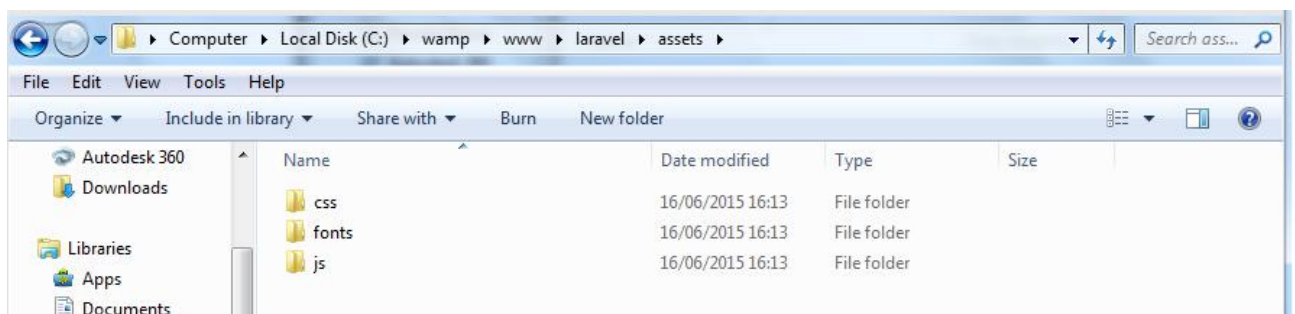
Ubahlah konfigurasi index.php anda seperti gambar diatas , apabila sudah buka browser anda dan coba untuk membuka url : localhost/laravel dan tara url anda sudah berganti menjadi localhost/laravel tidak lagi menjadi localhost/laravel/public



Laravel 5

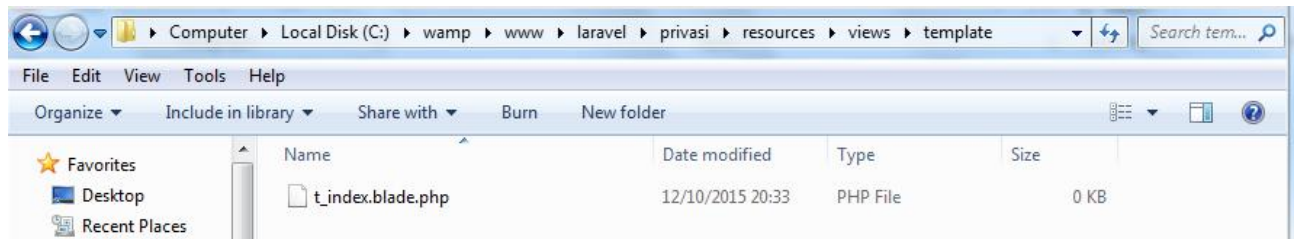
3.5. INTEGRASI BOOTSTRAP DI LARAVEL

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 :

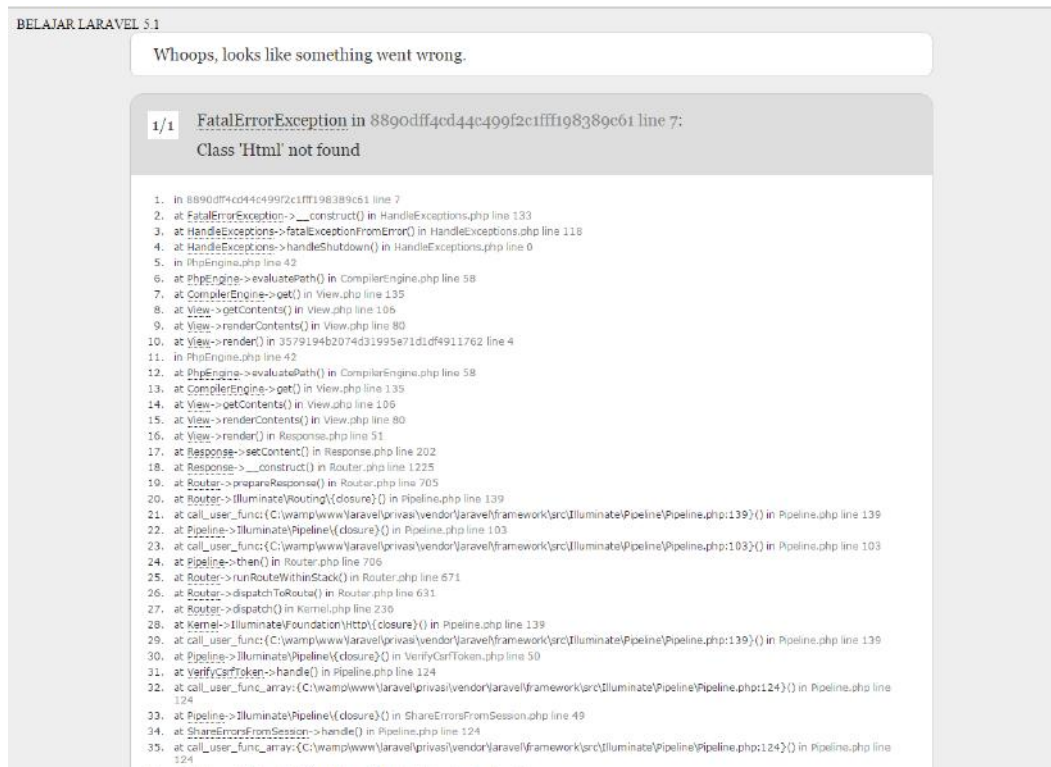
```
@extends('template/t_index')

@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 error 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,

```
'providers' => [
    /*
     * Laravel Framework Service Providers...
     */
    Illuminate\Foundation\Providers\ArtisanServiceProvider::class,
    Illuminate\Auth\AuthServiceProvider::class,
    Illuminate\Broadcasting\BroadcastServiceProvider::class,
    Illuminate\Bus\BusServiceProvider::class,
    Illuminate\Cache\CacheServiceProvider::class,
    Illuminate\Foundation\Providers\ConsoleSupportServiceProvider::class,
    Illuminate\Routing\ControllerServiceProvider::class,
    Illuminate\Cookie\CookieServiceProvider::class,
    Illuminate\Database\DatabaseServiceProvider::class,
    Illuminate\Encryption\EncryptionServiceProvider::class,
    Illuminate\Filesystem\FilesystemServiceProvider::class,
    Illuminate\Foundation\Providers\FoundationServiceProvider::class,
    Illuminate\Hashing\HashServiceProvider::class,
    Illuminate\Mail\MailServiceProvider::class,
    Illuminate\Pagination\PaginationServiceProvider::class,
    Illuminate\Pipeline\PipelineServiceProvider::class,
    Illuminate\Queue\QueueServiceProvider::class,
    Illuminate\Redis\RedisServiceProvider::class,
    Illuminate\Auth\Passwords>PasswordResetServiceProvider::class,
    Illuminate\Session\SessionServiceProvider::class,
    Illuminate\Translation\TranslationServiceProvider::class,
    Illuminate\Validation\ValidationServiceProvider::class,
    Illuminate\View\ViewServiceProvider::class,
    Illuminate\Html\HtmlServiceProvider::class,

    /*
     * Application Service Providers...
     */
    App\Providers\AppServiceProvider::class,
    App\Providers\AuthServiceProvider::class,
    App\Providers\EventServiceProvider::class,
    App\Providers\RouteServiceProvider::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 .

4. PEMBUATAN CRUD DI LARAVEL

4.1 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() !!}
  </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 :



sudah sama bukan ? apabila form sudah di buat selanjutkan proses pembuatan create data nya .

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');
```

sebelumnya saya jelaskan terlebih dahulu perbedaan get dan post

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');  
}
```

sedikit saya jelaskan input::get itu mengambil name yang ada di form yang sudah kita buat tadi , sedangkan db::table itu akan kemana data itu dimasukan yaitu ke table siswa , sedangkan with message itu notifikasi session yang sudah disediakan laravel bahwa apabila kita berhasil menambah data akan muncul message tersebut , lalu bagaimana memunculkan notifikasi diatas ?

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

1/1 FatalErrorException in Crudcontroller.php line 37:
Class 'App\Http\Controllers\DB' not found

1/1 FatalErrorException in Crudcontroller.php line 38:
Class 'App\Http\Controllers\Redirect' not found

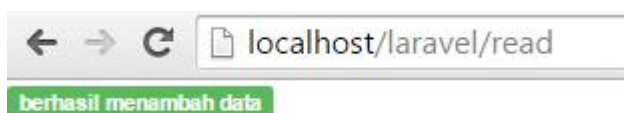
lalu bagaimana cara mengatasi eror berikut ? untuk mengatasi nya cukup mudah caranya buka file Crudcontroller lalu masukan code berikut

```
<?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 lakukan 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

4.2 Read / Menampilkan Data Di Laravel

step selanjutnya adalah menampilkan data yang sudah di input tadi bagaimana caranya ?

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 tambah kan 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);  
}
```

sedikit saya perjelas `DB::table('siswa')->get();` sama hal nya dengan `select * from siswa` , sedangkan `view::make` itu menampilkan untuk route get read dan disana ada siswa yang berarti nilai variable yang akan kita ambil untuk menampilkan data , apabila sudah menambah function lihatdata coba jalankan `localhost/laravel/read` dan anda akan menemukan eror View notfound cara untuk mengatasinya bagaimana ? sama seperti halnya yang di atas tambahkan use view;
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
```

sedikit saya jelaskan juga \$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 |

sama bukan ? yeee akhirnya read atau menampilkan data dari database sudah kita buat sekarang ke step selanjutnya yaitu menghapus data yang sudah kita buat :D cmiwiw :D

4.3 Delete / Menghapus Data Di Laravel

tahap selanjutnya menghapus data dari database menggunakan laravel , bagaimana caranya ? untuk memulai silahkan buka file routes.php di file laravel anda lalu tambahkan code seperti gambar berikut :

```
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');  
}
```

dan jangan lupa untuk memodifikasi a href yang ada di read tambahkan code berikut :

```
<a href="hapus/{ { $data->id} }">Hapus</a>
```

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

| berhasil menghapus data | | | | |
|-------------------------|-------|--------|-------|---------------|
| No | Nama | Alamat | Kelas | Action |
| 1 | dadan | dadan | dadan | Hapus Edit |
| 2 | Dadan | Dadan | Dadan | Hapus Edit |

tahap selanjutnya kita membuat Edit data di laravel (UPDATE)

4.4 Update / Mengubah Data Di Laravel

tahap selanjutnya adalah mengubah data yang ada di database menggunakan laravel caranya hampir sama dengan menghapus data tapi disini kita menggunakan dua method ikuti dan simak caranya ,

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

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

ini berfungsi untuk mengambil formedit dan menampilkan data berdasarkan id , dan jangan lupa memodifikasi a href juga di read.blade.php seperti berikut :

```
<a href="formedit/{{ $data->id }}">Edit</a>
```

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() !!}
@stop
</div>
</div>
</div>
```

sedikit saya jelaskan code gmabar diatas form hidden itu artinya sama dengan input type hidden , \$siswa itu mengambil dari function editdata with siswa , sedang \$siswa->id , id ini mengambil field yang ada di database cukup paham bukan ?

apabila sudah menambah code seperti gambar di atas selanjutnya menambah action route 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



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

yee alhamdulillah ubah data sudah berjalan dengan baik , tahap selanjutnya membuat LOGIN di laravel :D

5. LOGIN DI LARAVEL

5.1. PEMBUATAN HALAMAN LOGIN DI LARAVEL

Kali ini kita akan membuat login di laravel untuk caranya silahkan buat terlebih dahulu file yang bernama login.blade.php di laravel anda , apabila sudah membuat file tersebut selanjutnya buka file routes anda lalu tambahkan code seperti gambar dibawah ini :

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

ini bertujuan untuk menampilkan form login yang sudah kita buat tadi filenya , selanjutnya design form login anda , buka file login.blade.php dan tambahkan code seperti gambar dibawah ini :

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

@section('content')
<div class="container">
<p></p>
<div class="panel panel-default">
    <div class="panel-heading"><span class="glyphicon glyphicon-user"></span> Login
    @if(Session::has('message'))
    <span class="label label-danger">{{ Session::get('message') }}</span>
    @endif </div>
    <div class="panel-body">

        {!! Form::open(['url' => '/login']) !!}
        Username: @if($errors->has())
        <br/>
        <span class="label label-danger">{!! $errors->first('username') !!}</span>
        <p></p>
        @endif
        {!! Form::text('username', '', ['placeholder' => 'Username','class' => 'form-control']) !!}
        Password: @if($errors->has())
        <br/>
        <span class="label label-danger">{!! $errors->first('password') !!}</span>
        <p></p>
        @endif
        {!! Form::password('password', array('class' => 'form-control','placeholder' => 'Password')) !!}
        <p></p>
        {!! Form::submit('Login', ['class' => 'btn btn-danger']) !!}
        {!! Form::close() !!}

    </div>
</div>
</div>

@stop
```

Tambahkan code login.blade.php seperti diatas dan tampilannya akan seperti ini :

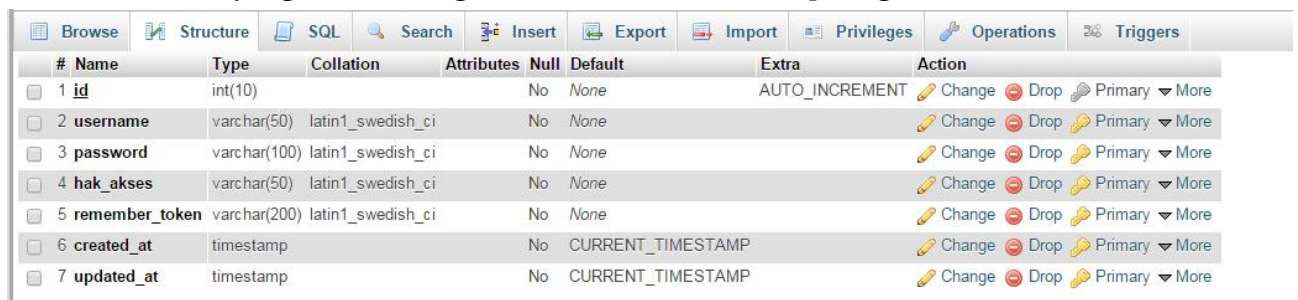


The screenshot shows a web browser window with the address bar displaying 'localhost/laravel/login'. The page content is a simple login form with a title 'Login'. It contains two input fields: 'Username:' and 'Password:'. Below these fields is a red button labeled 'Login'.

sama bukan ? sebelum kita masuk membuat proses login kita buat terlebih dahulu register agar data register yang kitaa buat bisa di pakai untuk login

5.2. PENAMBAHAN TABLE DI DATABASE

Tahap pertama yang harus kita lakukan adalah membuat struktur database untuk penyimpanan data register buka localhost/phpmyadmin masuk ke database laravel dan tambahkan table yang bernama login dan tambahkan field seperti gambar dibawah ini



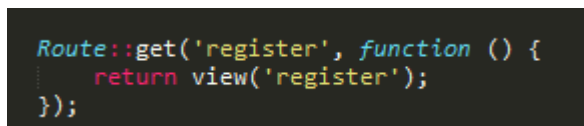
The screenshot shows the 'Structure' tab in phpMyAdmin for a table named 'login'. The table has 7 columns: #, Name, Type, Collation, Attributes, Null, Default, Extra, and Action. The columns are: 1. id (int(10), No, None, AUTO_INCREMENT, Primary), 2. username (varchar(50), latin1_swedish_ci, No, None, Primary), 3. password (varchar(100), latin1_swedish_ci, No, None, Primary), 4. hak_akses (varchar(50), latin1_swedish_ci, No, None, Primary), 5. remember_token (varchar(200), latin1_swedish_ci, No, None, Primary), 6. created_at (timestamp, No, CURRENT_TIMESTAMP, Primary), and 7. updated_at (timestamp, No, CURRENT_TIMESTAMP, Primary).

| # | Name | Type | Collation | Attributes | Null | Default | Extra | Action |
|---|----------------|--------------|-------------------|------------|------|-------------------|----------------|--------------------------|
| 1 | id | int(10) | | | No | None | AUTO_INCREMENT | Change Drop Primary More |
| 2 | username | varchar(50) | latin1_swedish_ci | | No | None | | Change Drop Primary More |
| 3 | password | varchar(100) | latin1_swedish_ci | | No | None | | Change Drop Primary More |
| 4 | hak_akses | varchar(50) | latin1_swedish_ci | | No | None | | Change Drop Primary More |
| 5 | remember_token | varchar(200) | latin1_swedish_ci | | No | None | | Change Drop Primary More |
| 6 | created_at | timestamp | | | No | CURRENT_TIMESTAMP | | Change Drop Primary More |
| 7 | updated_at | timestamp | | | No | CURRENT_TIMESTAMP | | Change Drop Primary More |

apabila sudah membuat table dan menambah field field di table login selanjutnya membuat form registrasi di laravel

5.3. PEMBUATAN HALAMAN REGISTER DI LARAVEL

caranya buat lah file yang bernama register.blade.php di resources>view>register.blade.php apabila sudah selanjutnya buka kembali routes dan tambahkan sama seperti login yaitu menambahkan code seperti gambar berikut :



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

apabila sudah menambah code seperti gambar diatas selanjut nya buka file yang bernama register.blade.php lalu tambahkan code seperti gambar berikut :

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

@section('content')
<div class="container">
<p></p>
<div class="panel panel-default">
  <div class="panel-heading"><span class="glyphicon glyphicon-user"></span> Register
  @if(Session::has('message'))
    <span class="label label-danger">{{ Session::get('message') }}</span>
  @endif </div>
  <div class="panel-body">

    {!! Form::open(['url' => '/tambahlogin']) !!}
    Username: @if($errors->has())
    <br/>
    <span class="label label-danger">{!! $errors->first('username') !!}</span>
    <p></p>
    @endif
    {!! Form::text('username', '', ['placeholder' => 'Username','class' => 'form-control']) !!}
    Password: @if($errors->has())
    <br/>
    <span class="label label-danger">{!! $errors->first('password') !!}</span>
    <p></p>
    @endif
    {!! Form::password('password', array('class' => 'form-control','placeholder' => 'Password')) !!}
    <p></p>
    {!! Form::submit('Register', ['class' => 'btn btn-danger']) !!}
    {!! Form::close() !!}
  </div>
</div>
</div>

@stop
```

apabila anda sudah menambahkan code seperti gambar di atas selanjutnya buka file routes.php dan tambahkan code seperti gambar di bawah ini :

```
Route::post('tambahlogin','Crudcontroller@tambahlogin');
```

code di atas bertujuan untuk mengirim data untuk di proses , tambahlogin itu adalah url yang kita buat

apabila sudah menambah code seperti di atas selanjutnya buka Crudcontroller dan tambahkan code seperti gambar di bawah ini :

```
public function tambahlogin() {

    $data = array(
        'username' => Input::get('username'),
        'password' => bcrypt(Input::get('password')),
        'hak_akses' => 'user'
    );

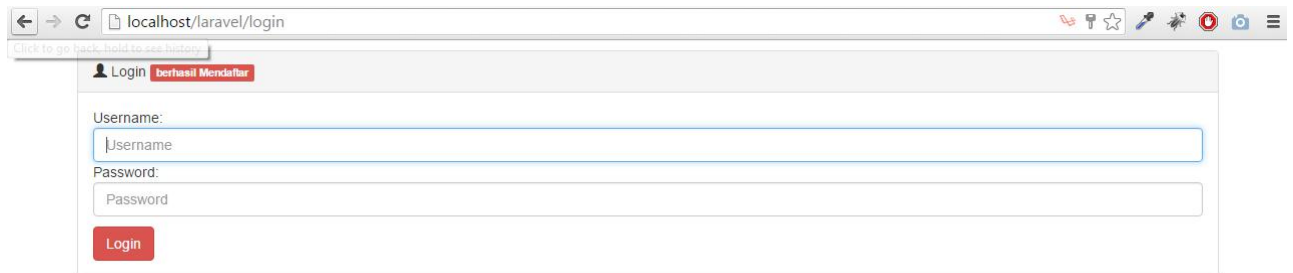
    DB::table('login')->insert($data);
    return Redirect::to('/login')->with('message','berhasil Mendaftar');

}
```

code di atas bertujuan untuk menambah data ke tabel login

sedikit sya jelasin bahkan bcrypt di atas sama seperti md5 pada umumnya akan tetapi bcrypt di atas khusus di buat di laravel

apabila anda sudah menambahkan function diatas ke crudcontroller anda selanjutnya mecoba untuk menambah kan data / register , buka localhost/laravel/register lalu isi username dan password dan hasilnya akan seperti ini :



sudah cukup jelas bukan untuk register data di laravel ? selanjut kita membuat proses login buka terlebih dahulu login.blade.php tambah kan a href untuk register di atas
Daftar <p></p> coba buka localhost/laravel/login dan hasilnya akan seperti ini



5.4. PROSES LOGIN DI LARAVEL

apabila sudah sama selanjutnya buka file routes dan tambahkan code di seperti gambar dibawah ini:

```
Route::post('login', 'Crudcontroller@login');
```

tahap selanjutnya buka kembali crudcontroller anda dan tambahkan code seperti gambar dibawah ini juga yang bertujuan untuk memproses login :

```
public function login()
{
    if (Auth::attempt(['username' => Input::get('username'), 'password' => Input::get('password')]))
    {
        if(Auth::user()->hak_akses=="admin"){
            echo "admin";
        }
        else{
            echo "user";
        }
    }
    else{
        echo "gagal login";
    }
}
```

kode di atas adalah untuk proses login sebelum kita mencoba login buka terlebih dahulu file yang bernama user.php di privasi>app>user.php dan ubah code protected table , protected fillable dan protected hidden samakan seperti gambar di bawah ini

```
protected $table = 'login';

/**
 * The attributes that are mass assignable.
 *
 * @var array
 */
protected $fillable = ['username', 'password'];

/**
 * The attributes excluded from the model's JSON form.
 *
 * @var array
 */
protected $hidden = ['password', 'remember_token'];
```

akan saya jelaskan terlebih dahulu , untuk protected \$table . itu adalah nama tablenya apa , dan \$fillable adalah input yang dipakai untuk login , sedangkan \$hidden itu adalah pemberian remember_token ke table kita apabila code nya sudah sama seperti gambar di atas selanjutnya kita coba mendaftar dan login di laravel kita buka terlebih dahulu localhost/laravel/register , coba tambahkan data lalu buka kembali localhost/laravel/login masukan data yang tadi di buat



dan tara hasilnya akan seperti berikut bahwa login sukses :D oh ya flashback ke artikel tadi, disini saya membuat login hak akses ya agar anda lebih mengerti dan lebih memahami saya membuat hak akses user dan admin bisa anda cek di atas lagi .

```
public function login()
{
    if (Auth::attempt(['username' => Input::get('username'), 'password' => Input::get('password')]))
    {
        if(Auth::user()->hak_akses=="admin"){
            echo "admin";
        }
        else{
            echo "user";
        }
    }
    else{
        echo "gagal login";
    }
}
```

sedikit saya perjelas untuk script di atas

auth::attempt itu untuk mengecek username dan password pada saat di masukan
auth::user() ini untuk mengecek pada saat dia login di lihat hak_akses nya admin atau bukan , apabila admin di tampilkan admin apabila bukan admin ditampilkan user
cukup mengerti bukan ?

Apabila kita sudah mengetahui bahwa yang login itu admin atau bukan selanjutnya kita berikan hak akses setiap file yang dibuka

contoh kasus : seperti penambahan data , pengeditan data , delete data , detail data itu hanya bisa dilakukan oleh hak akses admin , sedangkan user hanya bisa melihat namanya saja

5.1. MEMBERIKAN HAK AKSES TERTENTU DI LARAVEL

bagaimana caranya ? pertama buka terlebih dahulu crudcontroller anda lalu ubah code berikut :

```
if(Auth::user()->hak_akses=="admin"){
    echo "admin";
}
else{
    echo "user";
}
```

menjadi :

```
if(Auth::user()->hak_akses=="admin"){
    return Redirect::to("");
}
else{
    return Redirect::to('user');
}
```

sebelum mencoba menjalankannya terlebih dahulu kita membuat file yang bernama user.blade.php lalu buka routes.php dan tambahkan code berikut :

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

apabila anda sudah membuat file dan menambahkan code di atas ke routes.php anda selanjutnya buka file user.blade.php dan beri text haii user apabila sudah coba jalankan di browser anda localhost/laravel/login dan login dengan akun yang sudah anda buat dan hasilnya akan seperti gambar di bawah ini :

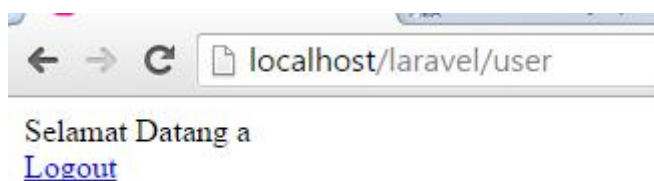


nah tahap selanjut adalah menampilkan username dan membuat logout :D caranya bagaimana ?

buka file yang bernama user.blade.php lalu hapus semua text yang ada di file tersebut lalu tambahkan code seperti gambar di bawah ini untuk menampilkan username login :

```
1 Selamat Datang {{ Auth::user()->username}} <br>  
2 <a href="{{ URL('logout') }}">Logout</a>
```

apabila sudah selanjutnya mencoba untuk login dan hasilnya akan seperti ini :



sama bukan ?

nah tahap selanjutnya adalah membuat logout , buka routes.php anda lalu tambah kan code seperti gambar di bawah ini :

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

lalu buka crudcontroller anda dan tambahkan code seperti gambar di bawah ini :

```
public function logout()
{
    Auth::logout();
    return Redirect::to('login')->with('message','Berhasil Logout !');
}
```

sudah ? login berhasil di buat sempurna , selanjutnya adalah mengecek session yang sudah login ,

sebelumnya saya jelaskan lagi terlebih dahulu perbedaan hak akses untuk admin dan user

Admin : - menambah data , menghapus data , melihat data , mengedit data

User : - hanya bisa melihat nama nya saja

Paham bukan ?

lalu bagaimana mengunci sessionnya ?

buka terlebih dahulu routes.php

Ubah code ini :

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

Menjadi :

```
Route::get('/', function () {
    if(Auth::user()){
        if(Auth::user()->hak_akses=="admin"){
            return view('home');
        }else{
            return view('user');
        }
    }
    else{
        return view('login');
    }
});
```


Dan Ubah Code ini juga :

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

Menjadi :

```
Route::get('login', function () {  
    if(Auth::user()){  
        if(Auth::user()->hak_akses=="admin"){  
            return view('home');  
        }else{  
            return view('user');  
        }  
    }  
    else{  
        return view('login');  
    }  
});
```

Dan Ubah Code ini juga :

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

Menjadi :

```
Route::get('register', function () {  
    if(Auth::user()){  
        if(Auth::user()->hak_akses=="admin"){  
            return view('home');  
        }else{  
            return view('user');  
        }  
    }  
    else{  
        return view('register');  
    }  
});
```

```
}  
  
});
```

Dan Ubah Code ini juga :

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

Menjadi :

```
Route::get('user', function () {  
    if(Auth::user()){  
        if(Auth::user()->hak_akses=="admin"){  
            return view('home');  
        }else{  
            return view('user');  
        }  
    }  
    else{  
        return view('login');  
    }  
});
```

Ubah code menjadi seperti di atas , sebelumnya saya jelaskan terlebih dahulu

```
if(Auth::user()){  
}  
else{  
}
```

Auth::user() : ini bertujuan untuk mengecek session apakah ada atau tidak ada apabila session ada cek lagi apakah si yang login ini mempunyai hak akses admin atau user , apabila admin tampilkan home sedangkan user tampilkan user ,begitu penjelasannya apakah cukup di mengerti ?

itu hanya beberapa file yang kita cek anda bisa mencoba untuk menambahkan beberapa file lagi :D selamat mencoba LOGIN SUDAH BERHASIL DI BUAT ! :D,

oh iya lupa admin belum membuat logout ya ? silahkan buat logout ya url nya sama kaya user ko :D

6. VALIDASI DI LARAVEL

5.1. VALIDASI LOGIN LARAVEL

Kali ini kita akan membuat validasi di laravel kita akan membuat 3 validasi laravel pertama yang akan kita buat validasi pada saat login , untuk langkah nya buka terlebih dahulu file privasi>app>http>request lalu buat file yang bernama validasilogin.php apabila anda sudah buat file tersebut lalu masukan koding gambar dibawah kepada file yang sudah anda buat ,

```
validasilogin.php
1 <?php
2
3 namespace App\Http\Requests;
4
5 use App\Http\Requests\Request;
6
7 class validasilogin extends Request
8 {
9     /**
10      * Determine if the user is authorized to make this request.
11      *
12      * @return bool
13      */
14     public function authorize()
15     {
16         return true;
17     }
18
19     /**
20      * Get the validation rules that apply to the request.
21      *
22      * @return array
23      */
24     public function rules()
25     {
26         return [
27             'username'=>'email|required',
28             'password'=>'required'
29         ];
30     }
31
32     public function messages()
33     {
34         return [
35             'username.required'=>'harus mengisi username',
36             'username.email'=>'format bukan email',
37             'password.required'=>'harus mengisi password',
38         ];
39     }
40 }
41
42
```

apabila anda sudah memasukan code seperti gambar di atas ke file validasilogin.php anda selanjutnya buka Crudcontroller anda lalu tambahkan
use App\Http\Requests\validasilogin;
ke file Crudcontroller anda , apabila sudah cari public function login() yang sudah dibuat

dan modifikasi code seperti gambar di bawah ini :

```
public function login(validasilogin $validasi)
{
    if (Auth::attempt(['username' => Input::get('username'), 'password' => Input::get('password')]))
    {
        if(Auth::user()->hak_akses=="admin"){
            return Redirect::to('');
        }
        else{
            return Redirect::to('user');
        }
    }
    else{
        echo "gagal login";
    }
}
```

sebetulnya tinggal menambah (validasilogin \$validasi)

apabila sudah anda modifikasi function login anda selanjutnya buka file login.blade.php anda lalu modifikasi seperti gambar dibawah ini :

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

@section('content')
<div class="container">
<p></p>
<div class="panel panel-default">
    <div class="panel-heading"><span class="glyphicon glyphicon-user"></span> Login
    @if(Session::has('message'))
    <span class="label label-danger">{{ Session::get('message') }}</span>
    @endif </div>
    <div class="panel-body">

        {!! Form::open(['url' => '/login']) !!}
        Username: @if($errors->has())
        <br/>
        <span class="label label-danger">{!! $errors->first('username') !!}</span>
        <p></p>
        @endif
        {!! Form::text('username', '', ['placeholder' => 'Username','class' => 'form-control']) !!}
        Password: @if($errors->has())
        <br/>
        <span class="label label-danger">{!! $errors->first('password') !!}</span>
        <p></p>
        @endif
        {!! Form::password('password', array('class' => 'form-control','placeholder' => 'Password')) !!}
        <a href="{{ URL('register') }}">Daftar</a>
        <p></p>
        {!! Form::submit('Login', ['class' => 'btn btn-danger']) !!}
        {!! Form::close() !!}

    </div>
</div>
</div>

@stop
```

sebetulnya hanya menambahkan code :

```
@if($errors->has())
```

```
<br/>
```

```
<span class="label label-danger">{!! $errors->first('username') !!}</span>
```

```
<p></p>
```

@endif

code tersebut mengecek error , apabila ada error tampilkan span class yang berisi username itu , seperti itu ,username itu mengambil dari file validasilogin.php yang sudah kita buat tadi , validasi login sudah selesai selanjutnya validasi register

5.2. VALIDASI REGISTER LARAVEL

Selanjutnya kita akan membuat validasi untuk form pembuatan register cara nya sama seperti validasi login di atas buka folder request dan buatlah file yang bernama validasiregister.php dan modifikasi file validasiregister.php seperti gambar dibawah ini :

```
<?php

namespace App\Http\Requests;

use App\Http\Requests\Request;

class validasiregister extends Request
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'username'=>'email|required',
            'password'=>'required'
        ];
    }

    public function messages()
    {
        return [
            'username.required'=>'harus mengisi Email',
            'username.email'=>'format bukan email',
            'password.required'=>'harus mengisi password',
        ];
    }
}
```

apabila sudah di modifikasi atau di tambah code seperti gambar di atas selanjutnya buka Crudcontroller anda dan tambahkan code

```
use App\Http\Requests\validasiregister;
```

ini bertujuan untuk memanggil file request validasinya

Apabila sudah menambahkan code di atas selanjutnya cari public function tambahlogin()

dan modifikasi function tambahlogin seperti gambar di bawah ini

```
public function tambahlogin(validasiregister $data) {
    $username = $data->username;
    $password = bcrypt($data->password);
    $data = array(
        'username' => $username,
        'password' => $password,
        'hak_akses' => 'user'
    );

    DB::table('login')->insert($data);
    return Redirect::to('/login')->with('message','berhasil Mendaftar');
}
```

sebelumnya saya jelaskan terlebih dahulu

validasiregister , ini untuk mengambil function dari request , \$data untuk mengambil data nya , cukup di mengertikan ?

sedangkan username , password itu sudah kita siapkan sebelumnya di file request coba lihat terlebih dahulu ,

tahap selanjutnya tinggal menambahkan kondisi di file register nya , buka file register.blade.php dan modifikasi seperti gambar di bawah ini .

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

@section('content')
<div class="container">
<p></p>
<div class="panel panel-default">
    <div class="panel-heading"><span class="glyphicon glyphicon-user"></span> Register
    @if(Session::has('message'))
    <span class="label label-danger">{{ Session::get('message') }}</span>
    @endif </div>
    <div class="panel-body">

        {!! Form::open(['url' => '/tambahlogin']) !!}
        Username: @if($errors->has())
        <br/>
        <span class="label label-danger">{!! $errors->first('username') !!}</span>
        <p></p>
        @endif
        {!! Form::text('username', '', ['placeholder' => 'Username','class' => 'form-control']) !!}
        Password: @if($errors->has())
        <br/>
        <span class="label label-danger">{!! $errors->first('password') !!}</span>
        <p></p>
        @endif
        {!! Form::password('password', array('class' => 'form-control','placeholder' => 'Password')) !!}
        <p></p>
        {!! Form::submit('Register', ['class' => 'btn btn-danger']) !!}
        {!! Form::close() !!}
    </div>
</div>
</div>

@stop
```

sama seperti login kita hanya menambahkan if error has saja untuk menampilkan erornya

, alhamdulillah register dan login sudah kita buat selanjutnya kita buat validasi tambah data untuk admin

5.3. VALIDASI TAMBAH DATA LARAVEL

Tahap selanjutnya kita membuat validasi tambah data untuk admin sama seperti validasi login, register kita harus membuat terlebih dahulu file di folder request, sebenarnya validasi di laravel bermacam macam tapi saya lebih suka ini, 1 file = 1 validasi lebih enak untuk saya jadi saya lebih mengerti, anda bebas mau memakai validasi apa terserah saja itu tergantung pemilihan anda

pertama buat file yang bernama validasitambah.php di folder request laravel anda apabila sudah buka file tersebut dan tambahkan code seperti gambar di bawah ini :

```
<?php

namespace App\Http\Requests;

use App\Http\Requests\Request;

class validasitambah extends Request
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'nama'=>'required',
            'alamat'=>'required',
            'kelas'=>'required'
        ];
    }

    public function messages()
    {
        return [
            'nama.required'=>'harus mengisi Nama',
            'alamat.required'=>'harus mengisi alamat',
            'kelas.required'=>'harus mengisi kelas',
        ];
    }
}
```

apabila anda sudah menambahkan code seperti gambar di atas selanjutnya tambah code :
use App\Http\Requests\validasitambah; di Crudcontroller anda apabila sudah menambahkan code diatas selanjutnya cari public function tambahdata() dan modifikasi

function tersebut seperti gambar di bawah ini :

```
public function tambahdata(validasitambah $data) {
    $nama=$data->nama;
    $alamat=$data->alamat;
    $kelas=$data->kelas;

    $data = array(
        'nama' =>$nama,
        'alamat' =>$alamat,
        'kelas' =>$kelas
    );

    DB::table('siswa')->insert($data);
    return Redirect::to('/read')->with('message','berhasil menambah data');
}
```

selanjutnya buka file yang bernama [home.blade.php](#) dan modifikasi code tersebut seperti gambar dibawah ini :

```
@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: @if($errors->has())
        <br/>
        <span class="label label-danger">{!! $errors->first('nama')!!}</span>
        <p></p>
        @endif
        {!! Form::text('nama', '', ['placeholder' => 'Nama','class' => 'form-control']) !!}
        Alamat:
        @if($errors->has())
        <br/>
        <span class="label label-danger">{!! $errors->first('alamat')!!}</span>
        <p></p>
        @endif
        {!! Form::text('alamat', '', ['placeholder' => 'Alamat','class' => 'form-control']) !!}
        Kelas:
        @if($errors->has())
        <br/>
        <span class="label label-danger">{!! $errors->first('kelas')!!}</span>
        <p></p>
        @endif
        {!! Form::text('kelas', '', ['placeholder' => 'Kelas','class' => 'form-control']) !!}
        <p></p>
        {!! Form::submit('Tambah Data', ['class' => 'btn btn-danger']) !!}
        {!! Form::close() !!}
    @stop
    </div>
</div>
</div>
```

validasi sudah selesai di buat dan silahkan untuk mencoba menjalankan validasi tersebut

7. KESIMPULAN

Alhamdulillah Tutorial Belajar Laravel Untuk Pemula sudah selesai di buat dari instal laravel , Crud , Login , Register dsb , semoga tutorial ini bermanfaat untuk anda pembaca , dan tujuan saya membuat tutorial ini untuk membantu anda semua agar bisa menggunakan framework laravel , apabila saya ada waktu insha allah saya akan membuat turoial selanjutnya dari mulai tutorial Codeigniter , Pembuatan E-commerce Dengan Laravel ataupun Codeigniter , Tutorial Bootstrap , Tutorial PHP Native , Pembuatan Website Dengan Ajax+Php Dll ,

Apabila anda mempunyai waktu saya mohon untuk mengkritik kekurangan saya ataupun Saran Untuk Saya agar bisa saya ubah dan menjadi lebih baik kedepannya

Untuk Kritik dan Sarannya : Dhamdani666@gmail.com / Facebook.com/dhamdani666

untuk anda yang sudah mempunyai pengalaman yang lainnya, saya anjurkan untuk membuat tutorial seperti saya agar orang orang yang awam bisa belajar dari tutorial anda , dan maaf untuk semuanya apabila kata kata saya masih banyak kesalahan , masih banyak kata kata yang melantur kemana mana , karena asya masih belajar dalam pembuatan tutorial , terimakasih.

8. TENTANG PENULIS



Nama : Dadan Hamdani
Tanggal Lahir : 02, January 1997
Usia : 18 Tahun
Agama : Islam
Alamat : Rancaekek Wetan , Kabupaten Bandung
Email : Dhamdani666@gmail.com
Facebook : Facebook.com/Dhamdani666

“ Sekali lagi untuk yang mempunyai waktu kirimkan kritik dan saran kepada email atau facebook saya agar next saya bisa lebih baik lagi , selamat mencoba dan selamat belajar. “