

TUGAS BESAR
PEMROGRAMAN WEB LANJUT
“WEB SEKOLAH”



Disusun sebagai
MATA KULIAH : PRAKTIKUM JARINGAN KOMPUTER

oleh :

NELI ALVIONITA
1931710087
MI-2D

PROGRAM STUDI D-III MANAJEMEN INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2020

A. PENDAHULUAN KONSEP WEB

Web yang saya buat ini adalah Web dengan tema sekolah yang berisi info mengenai data siswa dan data kelas di sekolah tersebut. Disini saya hanya akan membuat Web versi adminnya, jadi alur dari Web yang akan saya buat adalah

1. Halaman depan

Ketika admin masuk ke Web, Admin akan melihat halaman depan sebelum melakukan register dan login. Selain itu halaman depan juga berisi Visi Misi dari Sekolah tersebut, dan juga alasan kenapa harus memakai Web ini.

2. Halaman Login dan Register

Untuk register dan login, fungsinya seperti pada umumnya yakni, register digunakan bagi admin yang belum mempunyai akun masuk. Sedangkan admin digunakan untuk admin yang sudah mempunyai akun.

3. Halaman Admin

Saat admin sudah Login, maka admin akan diarahkan ke halaman admin, dimana disini admin akan mendapatkan akses untuk memanipulasi data siswa dan data kelas. Untuk penjelasan fungsi kedua data tersebut, sebagai berikut :

a. Data siswa

Disini admin dapat melakukan manipulasi atau *CRUD (Create, Read, Update, Delete)* pada data siswa. Adapun isi dari data siswa adalah nis(nomor induk siswa), nama, tanggal lahir, tahun ajar, dan juga kelas dari siswa. Dimana untuk kolom kelas ini akan berelasi dengan tabel kelas.

b. Data kelas

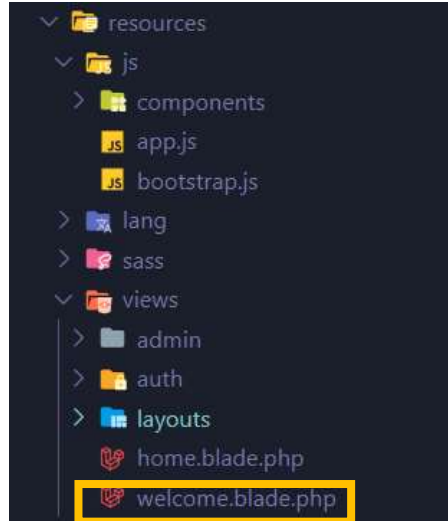
Sama halnya seperti data siswa, disini Admin memiliki fungsi manipulasi data. Adapun data kelas berisi nama kelas, jurusan, dan jumlah siswa di dalam kelas tersebut

Selain itu, di halaman admin juga terdapat tombol *logout* jika admin sudah menyelesaikan tugasnya di halaman tersebut. Setelah *logout* admin akan masuk ke halaman *welcome/depan*

B. PENJELASAN SOURCE CODE DAN TAMPILAN WEB

HALAMAN DEPAN

1. Pertama, masuk ke resource/views lalu buat welcome.blade.php, ini berfungsi untuk membuat code tampilan dari halaman depan



2. Setelah itu, isi file tersebut dengan code sebagai berikut

```
resources > views > welcome.blade.php > html > body > nav.navbar.navbar-dark
<html>
<head>
<title>SMA Negeri 1 Pacet - Welcome Page</title>
<style>
#header {
background-image: url("assets/head4-bg.jpg");
background-attachment: fixed;
background-size: cover;
padding-top: 200px;
padding-bottom: 100px;
}

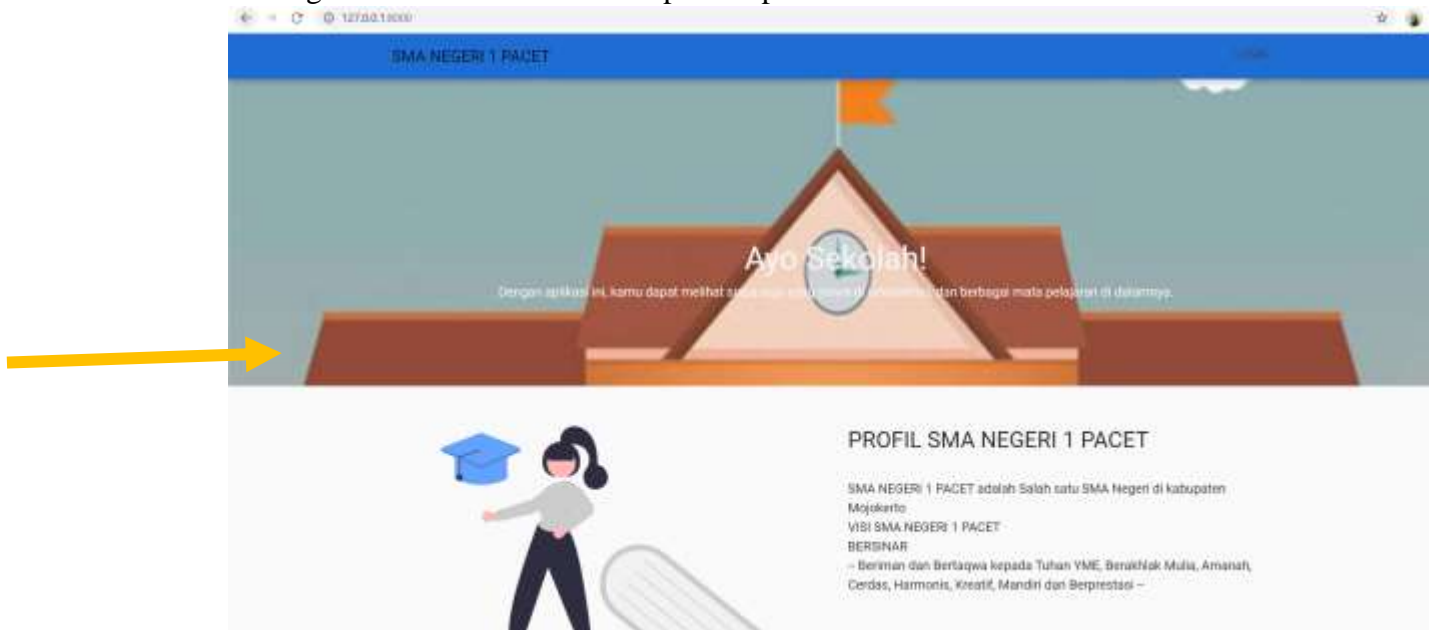
.benefit {
padding-top: 50px;
padding-bottom: 50px;
}

.img-benefit {
width: 100%;
height: auto;
}

.circle {
margin: auto;
margin-top: 25px;
width: 200px;
height: 200px;
border: 10px solid #ffff;
border-radius: 50%;
}
```

[illegible]

3. Codingan diatas akan muncul tampilan seperti dibawah ini



4. Untuk runningnya harus membuka terminal di vscode terlebih dahulu

```
Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\xampp\htdocs\sekolahapp> php artisan serve
Laravel development server started: http://127.0.0.1:8000
[Sun Dec 13 19:58:42 2020] PHP 7.4.13 Development Server (http://127.0.0.1:8000) started
[Sun Dec 13 20:00:10 2020] 127.0.0.1:55157 Accepted
```

Lalu salin <http://127.0.0.1:8000> tersebut pada browser. Maka akan muncul tampilan seperti diatas

5. Tambahkan code dibawah ini di dalam direktori welcome.blade.php,

```
<div class="container benefit">
  <div class="row">
    <div class="col-lg">
      <h1 style="font-weight: 500;">apa hari ini jadwal mata pelajaran apa?</h1>
      <br />
    </div>
    <div class="col-lg">
      <p>Jangan khawatir. Di aplikasi ini, kamu dapat mengetahui mata pelajaran apa sih yang harus kamu pelajari hari ini, bukan hanya hari ini, tapi kamu juga bisa menyiapkan untuk mata pelajaran besok, sehingga sekolahmu bisa lancar! Ayo semangat sekolah!</p>
    </div>
  </div>
  <div class="col-lg">
    
  </div>
</div>
<div class="container-fluid benefit" id="filter">
```

- 

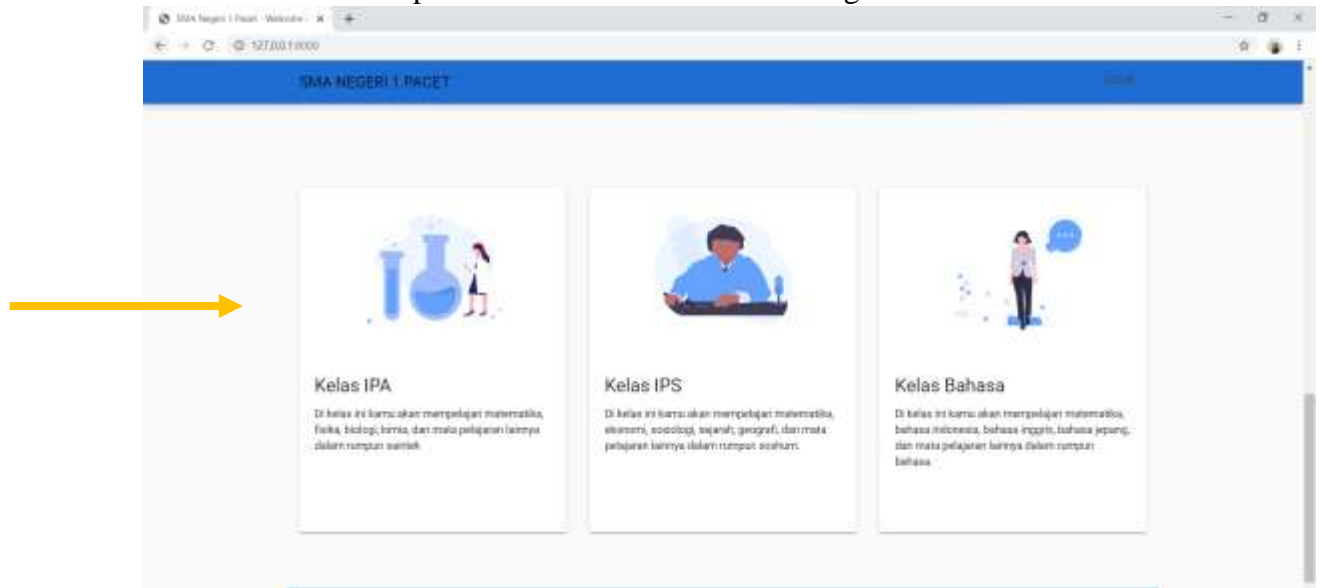


- ```

 </div>
 <div class="card-body">
 <div class="card-title">Kelas Bahasa</div>
 <div class="card-text">
 Di kelas ini kamu akan mempelajari matematika, bahasa Indonesia, bahasa Inggris, bahasa Jepang, dan mata pelajaran lainnya
 dalam rumpus bahasa.
 </div>
 </div>
 </div>
</div>
</div>
</div>
</div>

```

8. Kemudian untuk tampilan dari kode diatas adalah sebagai berikut



9. Dibawah tampilan kelas itu, juga terdapat kalimat ajakan untuk daftar akun di web ini, untuk kodenya seperti dibawah ini

```
</div>
</div>
</div>
</div>
<div class="container benefit text-center" style="background-color: #a1f6d3;">
 <h2>Bagaimana? sudah menemukan kelas yang cocok untuk kamu?</h2>
 <p>Yuk semangat sekolah!</p>
 Daftar
</div>
</body>
</html>
```

10. Tampilan paling bawah



11. Agar halaman tersebut bisa dipanggil di browser seperti tampilan diatas, maka perlu dilakukan pemanggilan melalui routing.

Untuk membuatnya maka, masuklah ke routes/web.php dan tuliskan kode berikut

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

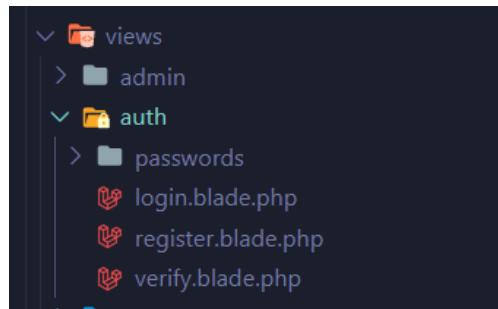
Jadi seperti route diatas, untuk memanggil halaman depan hanya perlu menuliskan link http yang muncul setelah menuliskan perintah “php artisan serve”

```
PS C:\xampp\htdocs\sekolahapp> php artisan serve
Laravel development server started http://127.0.0.1:8000
[Sun Dec 12 10:58:43 2020] PHP 7.4.17 Development Server (4075)
```

## HALAMAN LOGIN DAN REGISTER

1. Untuk membuat halaman login dan register ini saya menggunakan fitur yang disediakan oleh Laravel,
  - a. Pertama, tuliskan perintah berikut di terminal “composer require Laravel/ui:^2.4 di Laravel
  - b. Selanjutkan tuliskan perintah “php artisan ui bootstrap”
  - c. Lalu “php artisan ui bootstrap –auth”
  - d. Terakhir lakukan perintah “npm install” dan “npm run dev”

Maka akan muncul otomatis file file yang dibutuhkan di halaman login, yang terdapat di resource/views/auth



Jangan lupa lakukan “php artisan migrate” untuk menyimpan database





## 2. Isi dari tabel users

```
database > migrations > 2014_10_12_000000_create_users_table.php > ...
<?php

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

class CreateUsersTable extends Migration
{
 /**
 * Run the migrations.
 *
 * @return void
 */
 public function up()
 {
 Schema::create('users', function (Blueprint $table) {
 $table->id();
 $table->string('name');
 $table->string('email')->unique();
 $table->timestamp('email_verified_at')->nullable();
 $table->string('password');
 $table->rememberToken();
 $table->timestamps();
 });
 }

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

## 3. Isi dari tabel password

```
class CreatePasswordResetsTable extends Migration
{
 /**
 * Run the migrations.
 *
 * @return void
 */
 public function up()
 {
 Schema::create('password_resets', function (Blueprint $table) {
 $table->string('email')->index();
 $table->string('token');
 $table->timestamp('created_at')->nullable();
 });
 }

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

#### 4. Login.blade.php untuk tampilan login

```
#extends('layouts/app')

#section('content')
<div class="container">
 <div class="row justify-content-center">
 <div class="col-md-6">
 <div class="card">
 <div class="card-header">{{ __('login') }}</div>

 <div class="card-body">
 <form method="POST" action="{{ route('login') }}">
 @csrf

 <div class="form-group row">
 <label for="email" class="col-md-4 col-form-label text-md-right">{{ __('E-Mail Address') }}</label>

 <div class="col-md-6">
 <input id="email" type="email" class="form-control @error('email') is-invalid @enderror" name="email"

 @error('email')
 @error class="invalid-feedback" role="alert"
 [[$message]]
 </div>
 </div>

 <div class="form-group row">
 <label for="password" class="col-md-4 col-form-label text-md-right">{{ __('Password') }}</label>

 <div class="col-md-6">
 <input id="password" type="password" class="form-control @error('password') is-invalid @enderror" name="password"

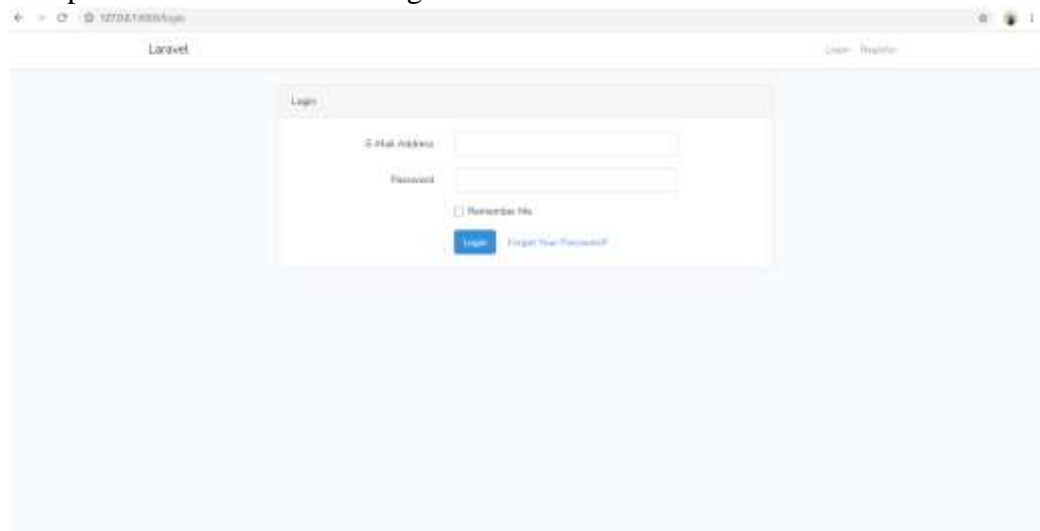
 @error('password')
 @error class="invalid-feedback" role="alert"
 [[$message]]
 </div>
 </div>
 </div>
 </div>
 </form>
 </div>
 </div>
 </div>
 </div>
</div>
```

```

1 resources/views/auth/login.blade.php < ...
2
3 @enderror
4 </div>
5
6 </div>
7
8 <div class="form-group row">
9 <div class="col-md-6 offset-md-4">
10 <div class="form-check">
11 <input class="form-check-input" type="checkbox" name="remember" id="remember" {{ old('remember') ? 'checked' : '' }}>
12
13 <label class="form-check-label" for="remember">
14 {{ __('Remember Me') }}
15 </label>
16 </div>
17 </div>
18 </div>
19
20 <div class="form-group row mb-0">
21 <div class="col-md-8 offset-md-4">
22 <button type="submit" class="btn btn-primary">
23 {{ __('Login') }}
24 </button>
25
26 @if (Route::has('password.request'))
27
28 {{ __('Forgot Your Password?') }}
29
30 @endif
31 </div>
32 </div>
33 </form>
34 </div>
35 </div>
36 </div>
37 </div>
38 </div>
39 </div>
40 @endsection

```

5. Tampilan dari kode diatas sebagai berikut



6. Lalu untuk isi dari kode register.blade.php

```
@extends('layouts.nav')

@section('content')

Register

Message

Message

Message

Message


```

7. Tampilan dari kode diatas adalah

8. Lalu untuk verifikasi data tersebut berada di halaman `verify.blade.php`

```

@portands('layouts.app')

@section('content')

{ {{ __('Verify Your Email Address') }} }

@if(session('resent'))

{{ __('A fresh verification link has been sent to your email address.') }}

@endif

[[__('Before proceeding, please check your email for a verification link.')]]
[[__('If you did not receive the email')]],
<div class="d-inline-block" method="POST" action="{ route('verification.resend') }">
 <button type="submit" class="btn btn-link p-0 m-0 align-baseline">{{ __('click here to request another') }}</button>
</div>
</div>
</div>
</div>
</div>
</div>
</div>


```

9. Ketika akan login, maka program akan melakukan verifikasi password yang sudah dimasukkan admin apakah sudah benar atau tidak, kodenya sebagai berikut

[illegible]

10. Untuk mereset password, maka akan diarahkan ke email untuk mengirimkan link reset password, dan ini akan dimasukkan ke file email.blade.php

```
@section('content')
<div class="container">
 <div class="row justify-content-center">
 <div class="col-md-8">
 <div class="card">
 <div class="card-header">{{ __('Reset Password') }}</div>
 <div class="card-body">
 @if (session('status'))
 <div class="alert alert-success" role="alert">
 {{ session('status') }}
 </div>
 @endif
 <form method="POST" action="{{ route('password.email') }}">
 @csrf
 <div class="form-group row">
 <label for="email" class="col-md-4 col-form-label text-md-right">{{ __('E-Mail Address') }}</label>
 <div class="col-md-6">
 <input id="email" type="email" class="form-control @error('email') is-invalid @enderror" name="email"
 @error('email')

 {{ $message }}

 @enderror
 </div>
 </div>
 <div class="form-group row mb-0">
 <div class="col-md-6 offset-md-4">
 <button type="submit" class="btn btn-primary">
 {{ __('Send Password Reset Link') }}
 </button>
 </div>
 </div>
 </form>
 </div>
 </div>
 </div>
 </div>
</section>
```

11. Lalu setelah dikirimkan link reset melalui email, maka akan masuk ke halaman reset password, ini terdapat dalam file reset.blade.php

```
@extends('layouts.app')

@section('content')
<div class="container">
 <div class="row justify-content-center">
 <div class="col-md-8">
 <div class="card">
 <div class="card-header">{{ __('Reset Password') }}</div>
 <div class="card-body">
 <form method="POST" action="{{ route('password.update') }}">
 @csrf
 <input type="hidden" name="token" value="{{ $token }}">
 <div class="form-group row">
 <label for="email" class="col-md-4 col-form-label text-md-right">{{ __('E-Mail Address') }}</label>
 <div class="col-md-6">
 <input id="email" type="email" class="form-control @error('email') is-invalid @enderror" name="email"
 @error('email')

 {{ $message }}

 @enderror
 </div>
 </div>
 <div class="form-group row">
 <label for="password" class="col-md-4 col-form-label text-md-right">{{ __('Password') }}</label>
 <div class="col-md-6">
 <input id="password" type="password" class="form-control @error('password') is-invalid @enderror" name="password"
 @error('password')

 {{ $message }}

 @enderror
 </div>
 </div>
 </form>
 </div>
 </div>
 </div>
 </div>
</section>
```

```

resources > views > auth > register.blade.php > ...
1 <input id="password" type="password" class="form-control @error('password') is-invalid @enderror" name="password" required autocomplete="new-password">
2 @error('password')
3
4 {{ $message }}
5
6 @enderror
7 </div>
8
9 <div class="form-group row">
10 <label for="password-confirm" class="col-md-4 col-form-label text-md-right">{{ __('Confirm Password') }}</label>
11
12 <div class="col-md-6">
13 <input id="password-confirm" type="password" class="form-control" name="password_confirmation" required autocomplete="new-password">
14 </div>
15
16 <div class="form-group row mb-0">
17 <div class="col-md-8 offset-md-4">
18 <button type="submit" class="btn btn-primary">{{ __('Reset Password') }}</button>
19 </div>
20 </div>
21 </div>
22 </div>
23 </div>
24 </div>
25 </div>
26 @endsection

```

12. Untuk halaman login dan register, tampilannya headernya karena file login.blade.php dan register.blade.php melakukan extends file app.blade.php yang terdapat di folder layouts.

```

resources > views > auth > register.blade.php > ...
1 @extends('layouts.app')
2
resources > views > auth > login.blade.php > ...
1 @extends('layouts.app')
2

```

Sama

13. Isi dari file app.blade.php sebagai layout dari login dan register

```

resources > views > auth > login.blade.php > ...
1 @extends('layouts.app')
2
3 @section('content')
4 <div class="container">
5 <div class="row justify-content-center">
6 <div class="col-md-8">
7 <div class="card">
8 <div class="card-header">{{ __('login') }}</div>
9
10 <div class="card-body">
11 <form method="POST" action="{{ route('login') }}">
12 @csrf
13
14 <div class="form-group row">
15 <label for="email" class="col-md-4 col-form-label text-md-right">{{ __('E-Mail Address') }}</label>
16
17 <div class="col-md-6">
18 <input id="email" type="email" class="form-control @error('email') is-invalid @enderror" name="email" value="{{ old('email') }}" required="">
19
20 @error('email')
21
22 {{ $message }}
23
24 @enderror
25 </div>
26 </div>
27
28 <div class="form-group row">
29 <label for="password" class="col-md-4 col-form-label text-md-right">{{ __('Password') }}</label>
30
31 <div class="col-md-6">
32 <input id="password" type="password" class="form-control @error('password') is-invalid @enderror" name="password" required="">
33
34 @error('password')
35
36 {{ $message }}
37
38 @enderror
39 </div>
40 </div>
41
42 <div class="form-group row">
43 <div class="col-md-8 offset-md-4">
44 <button type="submit" class="btn btn-primary">{{ __('Login') }}</button>
45 </div>
46 </div>
47 </form>
48 </div>
49 </div>
50 </div>
51 </div>
52 @endsection

```



```

<div class="form-group row">
 <label for="password" class="col-md-4 col-form-label text-md-right">{{ __('Password') }}</label>

 <div class="col-md-8">
 <input id="password" type="password" class="form-control @error('password') is-invalid @enderror" name="password"

 @error('password')

 {{ $__message__ }}
 @enderror
 </div>
 </div>

 <div class="form-group row">
 <div class="col-md-6 offset-md-4">
 <div class="form-check">
 <input class="form-check-input" type="checkbox" name="remember" id="remember" {{ old('remember') ? 'checked' : '' }}>

 <label class="form-check-label" for="remember">
 {{ __('Remember me') }}
 </label>
 </div>
 </div>
 </div>

 <div class="form-group row mb-0">
 <div class="col-md-8 offset-md-4">
 <button type="submit" class="btn btn-primary">
 {{ __('Login') }}
 </button>

 @if (Route::has('password.request'))

 {{ __('Forgot Your Password?') }}

 @endif
 </div>
 </div>

```

14. Ketika berhasil login nanti akan tampil pesan bahwa sudah login, ini disimpan pada [home.blade.php](https://laravel.com/docs/5.4/blade#session-status-messages)

```

resources > views > home.blade.php > ...
@extends('layouts.app')

@section('content')
<div class="container">
 <div class="row justify-content-center">
 <div class="col-md-8">
 <div class="card">
 <div class="card-header">{{ __('Dashboard') }}</div>

 <div class="card-body">
 @if (session('status'))
 <div class="alert alert-success" role="alert">
 {{ session('status') }}
 </div>
 @endif

 {{ __('You are logged in!') }}
 </div>
 </div>
 </div>
 </div>
</div>
@endsection

```

15. Lalu akan dipanggil melalui homecontroller

```
class HomeController extends Controller
{
 /**
 * Create a new controller instance.
 *
 * @return void
 */
 public function __construct()
 {
 $this->middleware('auth');
 }

 /**
 * Show the application dashboard.
 *
 * @return \Illuminate\Contracts\Support\Renderable
 */
 public function index()
 {
 return redirect('/admin');
 }
}
```

16. Routing pada Login

```
/*Route Login */
Auth::routes();

Route::get('/home', 'HomeController@index')->name('home');

Auth::routes();

Route::get('/home', 'HomeController@index')->name('home');

Auth::routes();

Route::get('/home', 'HomeController@index')->name('home');
```

## HALAMAN ADMIN

Halaman Admin ini berisi halaman setelah login, dimana admin bisa memanipulasi data,

1. Buat database dengan nama sekolah terlebih dahulu, ini berfungsi agar ketika kita melakukan migrasi bisa langsung masuk ke dalam database yang terdapat di localhost/phpMyAdmin

Basis data	Penyortiran	Tindakan
<input type="checkbox"/> information_schema	utf8_general_ci	<a href="#">Check privileges</a>
<input checked="" type="checkbox"/> mysql	utf8mb4_general_ci	<a href="#">Check privileges</a>
<input type="checkbox"/> performance_schema	utf8_general_ci	<a href="#">Check privileges</a>
<input type="checkbox"/> phpmyadmin	utf8_tda	<a href="#">Check privileges</a>
<input type="checkbox"/> sekolah	utf8mb4_general_ci	<a href="#">Check privileges</a>

Untuk membuat database baru



2. Buka folder .env di Laravel dan pastikan nama databasenya sama dengan yang dibuat di localhost/phpMyAdmin

```
.env
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:/G+sd9D0lqfpA2Rs20jDUL27ygrnyw94C2/PL27iNA=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack

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

3. Disini nanti saya akan membuat tabel siswa dan juga tabel kelas.
4. Pertama, buat migration dari tabel siswa terlebih dahulu dengan melakukan perintah di terminal  
php artisan make:migration create\_siswa\_table;
5. Setelah itu lakukan create kolom di dalam migration tersebut

```
class CreateSiswaTable extends Migration
{
 /**
 * Run the migrations.
 *
 * @return void
 */
 public function up()
 {
 Schema::create('siswa', function (Blueprint $table) {
 $table->bigIncrements('nis');
 $table->string('nama');
 $table->date('tgl_lahir');
 $table->integer('tahunajar');
 $table->unsignedBigInteger('kelas_id');
 });
 }

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

6. Selanjutnya lakukan perintah `php artisan make:migration create_kelas_table` untuk membuat tabel kelas, dan isi kolomnya

```
class CreateKelasTable extends Migration
{
 /**
 * Run the migrations.
 *
 * @return void
 */
 public function up()
 {
 Schema::create('kelas', function (Blueprint $table) {
 $table->id();
 $table->string("nama_kelas");
 $table->char("jurusan", 3);
 $table->integer("jumlah_siswa");
 });
 }

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

7. Lalu, lakukan perintah `php artisan migrate` untuk menyimpan tabel di database sekolah
8. Setelah itu, buatlah model siswa untuk mengatur data dari tabel siswa dengan perintah `php artisan make:model siswa`
9. Isi model tersebut dengan nama tabel dan kolom-kolomnya

```
class Siswa extends Model
{
 protected $table = "siswa";
 protected $primaryKey = "nis";
 protected $fillable = ["nama", "tgl_lahir", "tahun_ajar", "kelas_id"];

 public $timestamps = false;

 public function kelas(){
 return $this->belongsTo('App\Kelas', 'kelas_id', 'id');
 }
}
```

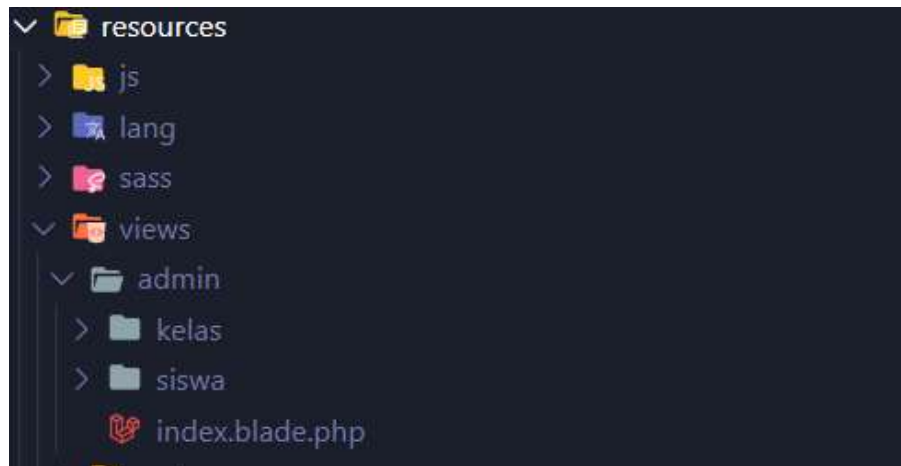
10. Buat model juga untuk tabel kelas dengan perintah `php artisan make:model kelas`, lalu isi

```
class Kelas extends Model
{
 protected $table = "kelas";
 protected $primaryKey = "id";
 protected $fillable = ["nama_kelas", "jurusan", "jumlah_siswa"];

 public $timestamps = false;

 public function siswa()
 {
 return $this->hasMany('App\Siswa');
 }
}
```

11. Setelah itu, buat folder admin di dalam direktori resources/views seperti berikut, di dalam folder admin juga buat file `index.blade.php` untuk meng extends layout dari halaman admin.



```
resources > views > admin > index.blade.php
1 @extends('layouts.master')
2
```

## 12. Isi dari layout halaman admin

```

<div class="main-header navbar navbar-expand navbar-white navbar-light">
 <!-- Left navbar links -->
 <ul class="navbar-nav">
 <li class="nav-item">
 <i class="fas fa-bars"></i>

 <!-- Right navbar links -->
 <li class="nav-item nav-right">
 <!-- Authentication links -->
 <div>
 <li class="nav-item">
 <a class="nav-link" href="{< route('login') >}">{< 'login' >}

 <li class="nav-item">
 <a href="{< route('register') >}">{< 'register' >}

 </div>
 <div>
 <li class="nav-item">
 <a href="{< route('logout') >}">{< 'logout' >}

 </div>

</div>

<div class="content-wrapper">
 <div class="main-sidebar sidebar-dark primary elevation-4">
 <!-- Brand logo -->

 AdminLTE

 <!-- Sidebar -->
 <div class="sidebar">
 <!-- Sidebar user panel (optional) -->
 <div class="user-panel mt-3 pb-3 px-3 d-flex">
 <div class="info">
 Siswa
 </div>
 </div>

 <!-- Sidebar Menu -->
 <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu" data-accordion="false">
 <!-- Add icons to the links using the .nav-icon class
 with font-awesome or any other icon font library -->
 <li class="nav-item has-treeview menu-open">

 <i class="nav-icon fas fa-tachometer-alt"></i>
 <div>
 <div>
 <i class="right fas fa-angle-left"></i>
 </div>
 </div>

 <ul class="nav nav-treeview">
 <li class="nav-item">

 <i class="fas fa-circle nav-icon"></i>
 <p>Data Siswa</p>

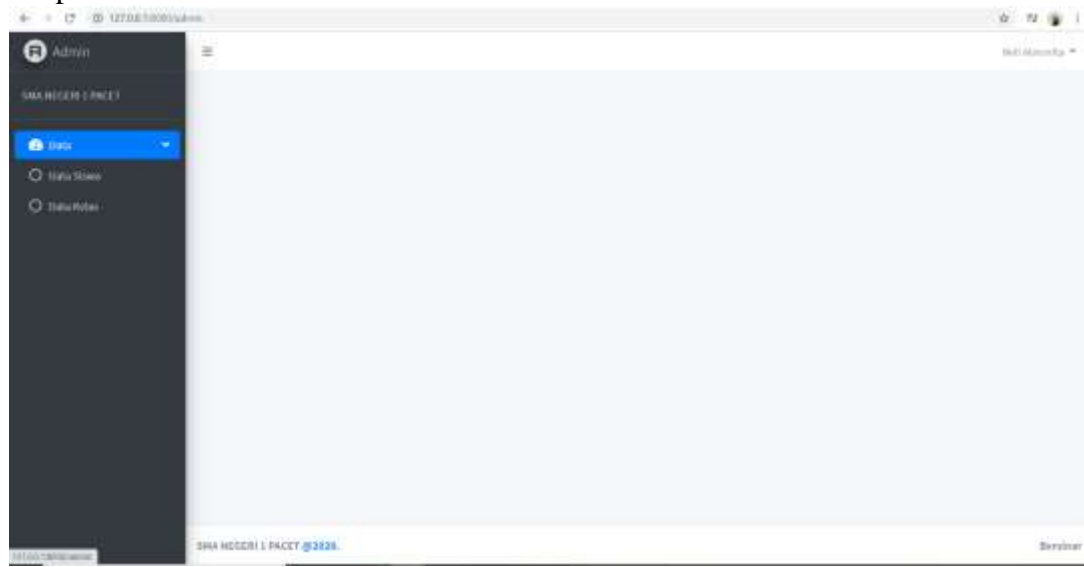
 <li class="nav-item">

 <i class="fas fa-circle nav-icon"></i>
 <p>Data Kelas</p>

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

```

### 13. Tampilan dari halaman admin



## DATA SISWA

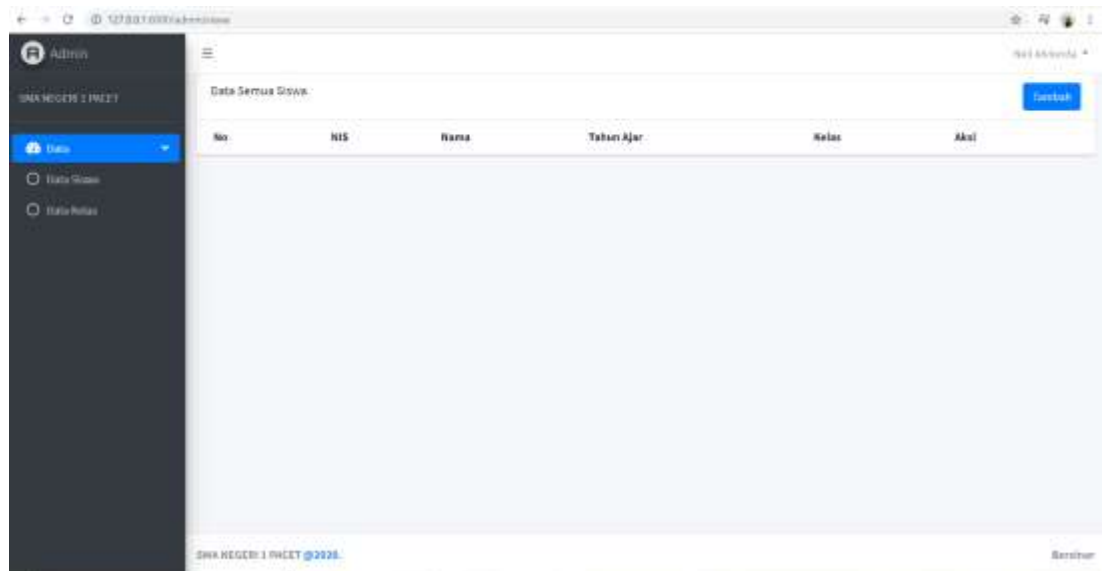
### 14. Buat file index.blade.php untuk menampilkan isi dari tabel siswa

```
resources > siswa > admin > siswa > index.blade.php > ...
@extends('layouts.master')

@section('title', 'Data Siswa')

@section('content')

<div class="card">
 <div class="card-header">
 <h3 class="card-title">Data Siswa</h3>
 Tambah
 </div>
 <div class="card-body p-0">
 <table class="table table-hover">
 <thead>
 <tr>
 <th>No</th>
 <th>NIS</th>
 <th>Nama</th>
 <th>Tahun Ajar</th>
 <th>Kelas</th>
 <th>Aksi</th>
 </tr>
 </thead>
 <tbody>
 @foreach ($siswa as $index => $item)
 <tr>
 <td>{{ $index + 1 }}</td>
 <td>{{ $item->nis }}</td>
 <td>{{ $item->nama }}</td>
 <td>{{ $item->tahun_ajar }}</td>
 <td>{{ $item->kelas->nama_kelas }}</td>
 <td>
 nis }}/detail" class="btn btn-info"><i class="fa fa-eye" aria-hidden="true"
 </td>
 </tr>
 @endforeach
 </tbody>
 </table>
 </div>
</div>
```



15. Buat file tambah.blade.php untuk melakukan kode supaya admin bisa menambahkan data siswa dengan menekan tombol tambah di kanan atas tabel

```

@extends('layouts.master')
@section('title', 'Tambah Data Siswa')

@section('content')
@if ($errors->any())
<div class="alert alert-danger">

 @foreach ($errors->all() as $error)
 {{ $error }}
 @endforeach

</div>
@endif
<div class="card">
 <div class="card-header">
 <h3 class="card-title">Tambah Data Siswa</h3>
 </div>
 <div class="card-body">
 <form action="/admin/siswa" method="POST">
 <div>
 <div class="form-group">
 <label for="nis">NIS</label>
 <input type="text" name="nis" id="nis" class="form-control" placeholder="NIS" aria-describedby="helpid" readonly>
 <small id="helpid" class="text-muted">isi dengan NIS siswa</small>
 </div>
 <div class="form-group">
 <label for="nama">Nama</label>
 <input type="text" name="nama" id="nama" class="form-control" placeholder="Nama Siswa" aria-describedby="helpid">
 <small id="helpid" class="text-muted">Nama sesuai dengan Aktas</small>
 </div>
 <div class="form-group">
 <label for="tgl_lahir">Tanggal lahir</label>
 <input type="date" name="tgl_lahir" id="tgl_lahir" class="form-control" placeholder="tanggal lahir" aria-describedby="helpid">
 <small id="helpid" class="text-muted">tanggal lahir sesuai akta</small>
 </div>
 <div class="form-group">
 <label for="tahun_ajar">tahun ajar</label>
 <input type="number" name="tahun_ajar" id="tahun_ajar" class="form-control" placeholder="tahun ajar" aria-describedby="helpid">
 <small id="helpid" class="text-muted">tahun ajar</small>
 </div>
 <div class="form-group">
 <label for="kelas">kelas</label>
 <select name="id_kelas" id="id_kelas" class="form-control">
 @foreach ($list_kelas as $kelas)
 <option value="{{ $kelas->id }}" {{ $kelas->nama_kelas }}</option>
 @endforeach
 </select>
 <small id="helpid" class="text-muted">kelas yang diminia</small>
 </div>
 <button type="submit" class="btn btn-success">simpan</button>
 </div>
 </form>
 </div>
</div>
</section>

```

**Tambah Data Siswa**

**NIS**  
NIS

**Nama**  
Nama Siswa

**Tanggal Lahir**  
DD-MM-YYYY

**Tahun Ajar**  
Tahun Ajar

**Kelas**  
Kelas

**Simpan**

➤ Misal saya menambahkan data dibawah ini

**Tambah Data Siswa**

**NIS**  
100

**Nama**  
Ruli Ananda

**Tanggal Lahir**  
01-01-2000

**Tahun Ajar**  
2019

**Kelas**  
10 IPA 1

**Simpan**

➤ Klik simpan maka akan muncul data dibawah ini

**Data Semua Siswa**

No	NIS	Nama	Tahun Ajar	Kelas	Aksi
1	100	Ruli Ananda	2019	10 IPA 1	

**Tambah**

16. Buat file detail.blade.php untuk melihat detail dari tabel siswa

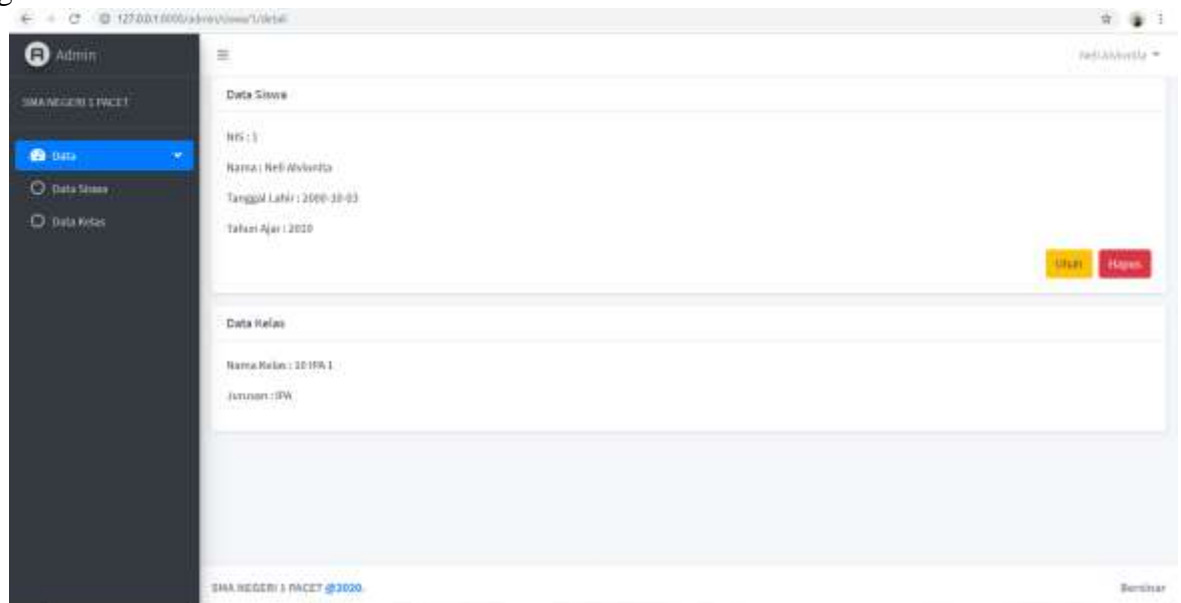
```
resources > views > admin > siswa > detail.blade.php > ...
@extends('layouts.master')
@section('title', 'Detail Siswa')

@section('content')

<div class="card">
 <div class="card-header">
 <h3 class="card-title">Data Siswa</h3>
 </div>
 <div class="card-body">
 <p>NIS : {{ $siswa->nis }}</p>
 <p>Nama : {{ $siswa->nama }}</p>
 <p>Tanggal Lahir : {{ $siswa->tgl_lahir }}</p>
 <p>Tahun Ajar : {{ $siswa->tahun_ajar }}</p>
 <form action="/admin/siswa/{{ $siswa->nis }}" method="post">
 @csrf
 @method('DELETE')
 <button class="btn btn-danger float-right ml-2" type="submit">Hapus</button>
 </form>
 nis }}/ubah" class="btn btn-warning float-right">Ubah
 </div>
</div>

<div class="card">
 <div class="card-header">
 <h3 class="card-title">Data Kelas</h3>
 </div>
 <div class="card-body">
 <p>Nama Kelas : {{ $siswa->kelas->nama_kelas }}</p>
 <p>Jurusan : {{ $siswa->kelas->jurusan }}</p>
 </div>
</div>
@endsection
```

17. Untuk tampilan detailnya, klik ikon mata di tampilan seperti no 16, maka akan muncul gambar dibawah





18. Buat ubah.blade.php supaya admin bisa mengedit/mengubah data siswa

```

<extends('layouts.master')
@section('title', 'Tambah Data Siswa')

@section('content')

<div class="card">
<div class="card-header">
<h3 class="card-title">Tambah Data Siswa</h3>
</div>
<div class="card-body">
<form action="/admin/siswa/{{ $siswa->nis }}" method="POST">
@csrf
@method('PUT')
<div class="form-group">
<label for="nis">NIS</label>
<input type="text" name="nis" id="nis" class="form-control" placeholder="NIS" aria-describedby="helpid"
value="{{ $siswa->nis }}" readonly>
<small id="helpid" class="text-muted">Iri dengan NIS Siswa</small>
</div>
<div class="form-group">
<label for="nama">Nama</label>
<input type="text" name="nama" id="nama" class="form-control" placeholder="Nama Siswa" aria-describedby="hel"
value="{{ $siswa->nama }}">
<small id="helpid" class="text-muted">Nama sesuai dengan Aktas</small>
</div>
<div class="form-group">
<label for="tgl_lahir">Tanggal lahir</label>
<input type="date" name="tgl_lahir" id="tgl_lahir" class="form-control" placeholder="Tanggal lahir"
aria-describedby="helpid" value="{{ $siswa->tgl_lahir }}">
<small id="helpid" class="text-muted">Tanggal lahir sesuai aktas</small>
</div>
<div class="form-group">
<label for="tahun_ajar">Tahun Ajar</label>
<input type="number" name="tahun_ajar" id="tahun_ajar" class="form-control" placeholder="Tahun Ajar"
aria-describedby="helpid" value="{{ $siswa->tahun_ajar }}">
<small id="helpid" class="text-muted">Tahun Ajar</small>
</div>
<div class="form-group">
<label for="kelas">Kelas</label>
<select name="id_kelas" id="id_kelas" class="form-control">
@foreach ($listkelas as $kelas)
<option value="{{ $kelas->id }}">{{ $kelas->nama_kelas }}</option>
</foreach>
</select>
<small id="helpid" class="text-muted">Kelas yang diwakil</small>
</div>
<button type="submit" class="btn btn-success">Simpan</button>
</form>
</div>
</div>
<script>
$(document).ready(function () {
 $('#id_kelas').val('{{ $siswa->kelas_id }}');
});
</script>
@endsection

```

- Misal dari data yang kita isi di poin 6 kita ubah tahun ajar dari siswa. Untuk alurnya kita perlu melihat detail poin yang akan kita edit, setelah masuk ke halaman detail, maka kita akan melihat ikon ubah dan hapus di sebelah kanan



- Klik ikon tersebut, maka akan muncul gambar dibawah

The screenshot shows the 'Tambah Data Siswa' (Add Student Data) form in the Admin dashboard. The form includes fields for NIS, Nama, Tanggal Lahir, Tahun Ajar, and Kelas. The 'Tahun Ajar' field is currently set to 2019. A green 'Simpan' (Save) button is at the bottom.

- Ubah tahun ajarnya menjadi 2018

This screenshot shows the same 'Tambah Data Siswa' form, but the 'Tahun Ajar' field has been updated to 2018. The 'Simpan' button remains at the bottom.

- Hasil setelah diubah

The screenshot shows the 'Data Semua Siswa' (All Student Data) table. The table has columns for No, NIS, Nama, Tahun Ajar, Kelas, and Aksi. The first row shows a student with NIS 1, Nama Neli Akhronita, Tahun Ajar 2018, and Kelas 20 IPA 1. A green 'Edit' button is visible in the 'Aksi' column.

No	NIS	Nama	Tahun Ajar	Kelas	Aksi
1	1	Neli Akhronita	2018	20 IPA 1	Edit

19. Untuk perintah menghapus data tidak perlu dibuat file .blade.php, tapi cukup diberikat fungsi pada controllernya, berikut controller dari tabel siswa secara keseluruhan dari melihat index, tambah, ubah, lihat detail, dan hapus. Sebelum itu lakukan perintah `php artisan make:controller SiswaController` untuk membuat controllernya.

## 20. Detail SiswaController

### a. Fungsi index dan tambah

```
<?php

namespace App\Http\Controllers;

use App\Kelas;
use App\Siswa;
use Illuminate\Http\Request;

class SiswaController extends Controller
{
 public function __construct()
 {
 $this->middleware('auth');
 }

 public function index(){
 $siswa = Siswa::all();

 return view('admin/siswa/index', ['siswa' => $siswa]);
 }

 public function create()
 {
 $listkelas = Kelas::all();
 return view('/admin/siswa/tambah', ['listkelas' => $listkelas]);
 }
}
```

### b. Fungsi untuk menyimpan data

```
public function store(Request $request)
{
 $siswa = new Siswa();

 $this->validate($request, [
 'nama' => 'required',
 'tgl_lahir' => 'required|date',
 'tahun_ajar' => 'required|numeric',
 'id_kelas' => 'required|numeric'
]);

 $siswa->nama = $request->nama;
 $siswa->tgl_lahir = $request->tgl_lahir;
 $siswa->tahun_ajar = $request->tahun_ajar;
 $siswa->kelas_id = $request->id_kelas;

 if ($siswa->save()) {
 return redirect('/admin/siswa');
 }
}
```

### c. Fungsi detail dan update data

```

10 public function detail(Request $request)
11 {
12 $siswa = siswa::find($request->id);
13 return view('/admin/siswa/detail', ['siswa' => $siswa]);
14 }
15
16 public function ubah(Request $request)
17 {
18 $siswa = siswa::find($request->id);
19 $listKelas = Kelas::all();
20 return view('/admin/siswa/ubah', ['siswa' => $siswa, 'listKelas' => $listKelas]);
21 }

```

d. Fungsi ubah dan hapus data

```

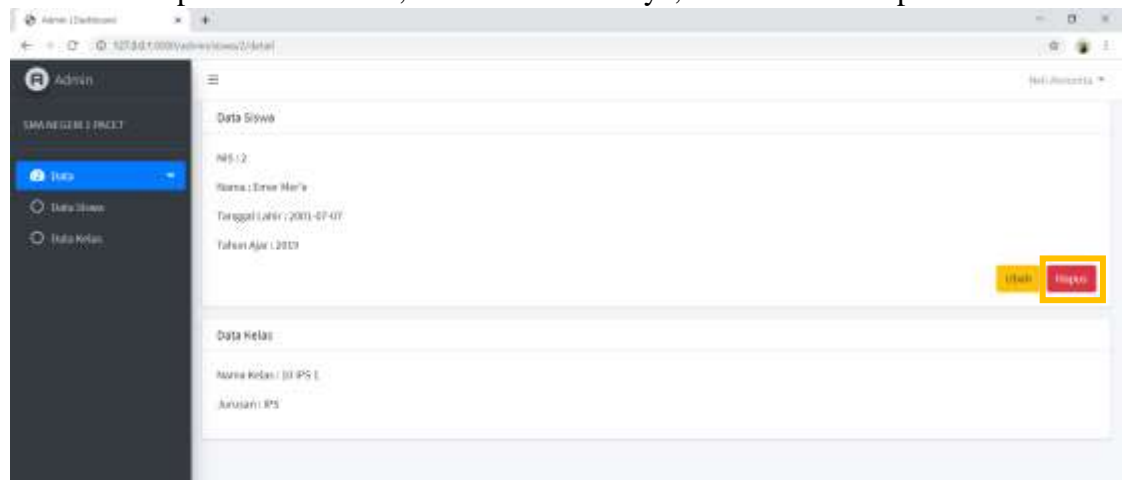
1 public function ubah(Request $request)
2 {
3 $siswa = siswa::find($request->id);
4 $listKelas = Kelas::all();
5 return view('/admin/siswa/ubah', ['siswa' => $siswa, 'listKelas' => $listKelas]);
6 }
7
8 public function update(Request $request){
9 $siswa = Siswa::find($request->id);
10
11 $this->validate($request, [
12 'nama' => 'required',
13 'tgl_lahir' => 'required|date',
14 'tahun_ajar' => 'required|numeric',
15 'id_kelas' => 'required|numeric'
16]);
17
18 $siswa->nama = $request->nama;
19 $siswa->tgl_lahir = $request->tgl_lahir;
20 $siswa->tahun_ajar = $request->tahun_ajar;
21 $siswa->kelas_id = $request->id_kelas;
22
23 if ($siswa->save()) {
24 return redirect('/admin/siswa');
25 }
26 }
27
28 public function delete(Request $request)
29 {
30 $siswa = Siswa::find($request->id);
31 if ($siswa->delete()) {
32 return redirect('/admin/siswa');
33 }
34 }
35 }

```

- Tampilan data untuk menghapus, maka alurnya kita harus melihat detail data yang akan kita hapus lalu klik ikon hapus



- Misal kita hapus data nomor 2, maka lihat detailnya, dan klik ikon hapus



- Tampilan setelah di klik hapus



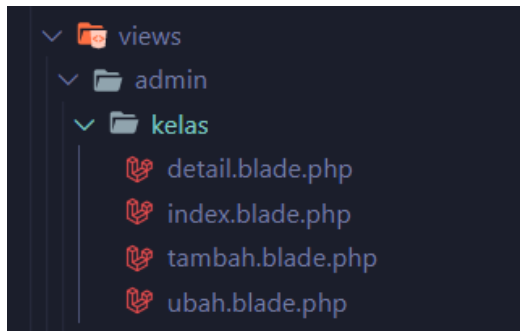
Data nomor 2 sudah terhapus

## 21. Routing tabel siswa

```
21 /* Route Tabel siswa */
22 Route::get('/admin/siswa', 'SiswaController@index');
23 Route::get('/admin/siswa/tambah', 'SiswaController@create');
24 Route::post('/admin/siswa', 'SiswaController@store');
25 Route::get('/admin/siswa/{id}/detail', 'SiswaController@detail');
26 Route::get('/admin/siswa/{id}/ubah', 'SiswaController@ubah');
27 Route::put('/admin/siswa/{id}', 'SiswaController@update');
28 Route::delete('/admin/siswa/{id}', 'SiswaController@delete');
```

## DATA KELAS

22. Buat file `index.blade.php` di folder `resources/views/admin/kelas` sebagai tampilan dari isi tabel kelas



23. Isi file tersebut (`index.blade.php`) untuk memanggil kolom-kolom ditabel kelas

```
resources > views > admin > kelas > index.blade.php > diuotow > diuotow-md-4
@extends('layouts.master')

@section('title', 'Data Kelas')

@section('content')
 @if (session('message'))
 <div class="alert alert-success alert-dismissible">
 {{ session('message') }}
 </div>
 @endif
 <div class="row">
 @foreach ($kelas as $item)
 <div class="col-md-4">
 id }}/detail" class="card">
 <div class="card-body">
 <h4 class="card-title">{{ $item->nama_kelas }}</h4>
 <p class="card-text">{{ $item->jurusan }}</p>
 </div>

 </div>
 @endforeach
 </div>
 Tambah
@endsection
```

24. Controller dari index, untuk membuat controller buat melalui terminal dengan mengetikkan perintah “`php artisan make:controller KelasController`”

```
class KelasController extends Controller
{
 public function __construct()
 {
 $this->middleware('auth');
 }

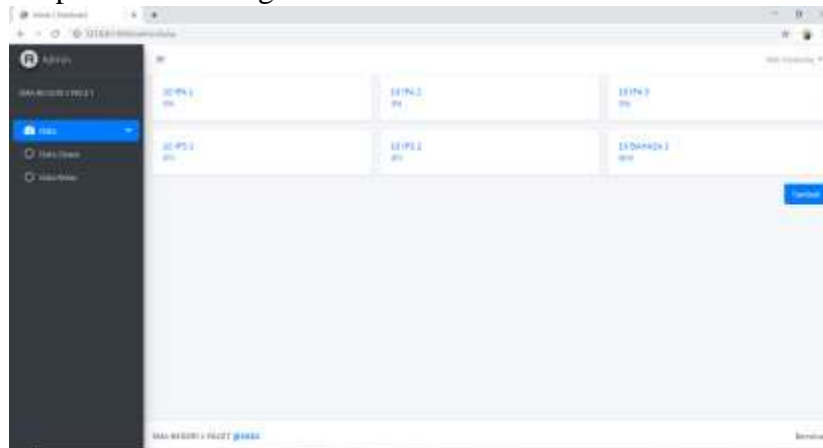
 public function index()
 {
 $kelas = Kelas::all();

 return view('admin/kelas/index', ['kelas' => $kelas]);
 }
}
```

25. Routing nya untuk pemanggilan di browser

```
Route::get('/admin/siswa', 'SiswaController@index');
```

26. Tampilan dari kodingan diatas adalah



27. Buat file tambah.blade.php untuk memasukkan query dalam menambahkan data siswa

```
<code>@section('content')
<div class="card">
 <div class="card-header">
 <div class="card-title">Tambah Data</div>
 </div>
 <div class="card-body">
 <form action="/admin/kelas" method="post">
 <div class="form-group">
 <label for="nama">Nama Kelas</label>
 <input type="text" name="nama" id="nama" class="form-control" placeholder="Nama Kelas"
 aria-describedby="helpid">
 <small id="helpid" class="text-muted">Contoh : IB IPA 1</small>
 </div>
 <div class="form-group">
 <label for="jurusan">Jurusan</label>
 <input type="text" name="jurusan" id="jurusan" class="form-control" placeholder="Jurusan (3 Huruf Saja)"
 aria-describedby="helpid">
 <small id="helpid" class="text-muted">Contoh : IPA</small>
 </div>
 <div class="form-group">
 <label for="jumlah_siswa">Jumlah Siswa</label>
 <input type="number" name="jumlah_siswa" id="jumlah_siswa" class="form-control" placeholder="Jumlah Siswa"
 aria-describedby="helpid">
 <small id="helpid" class="text-muted">Contoh : 30</small>
 </div>
 <button type="submit" class="btn btn-success">Simpan</button>
 </form>
 </div>
</div>
</section></code>
```

28. Tambahkan fungsi create di KelasController

```
<code>public function create()
{
 return view('/admin/kelas/tambah');
}

public function store(Request $request)
{
 $this->validate($request, [
 'nama' => 'required',
 'jurusan' => 'required',
 'jumlah_siswa' => 'required|numeric',
]);

 $kelas = new Kelas();
 $kelas->nama_kelas = $request->nama;
 $kelas->jurusan = $request->jurusan;
 $kelas->jumlah_siswa = $request->jumlah_siswa;
 if ($kelas->save()) {
 return redirect('/admin/kelas');
 }
}
```

Routingnya → `Route::get('/admin/siswa/tambah', 'SiswaController@create');`

29. Klik ikon tambah di kanan bawah lalu akan muncul gambar dibawah ini

- Terdapat data-data yang harus diisi yaitu nama kelas, jurusan dan jumlah siswa
- Coba kita isikan kelas baru yaitu nama kelas 10 BAHASA 3, untuk jurusan harus diisi dengan maksimal 3 huruf, jadi kita tuliskan BHS, dan jumlah siswa 25

- Klik kirim untuk menyimpan data

No	Nama Kelas	Jurusan	Jumlah Siswa
1	10 IPA 1	IPA	30
2	10 IPA 2	IPA	30
3	10 IPA 3	IPA	30
4	10 BAHASA 1	BHS	25

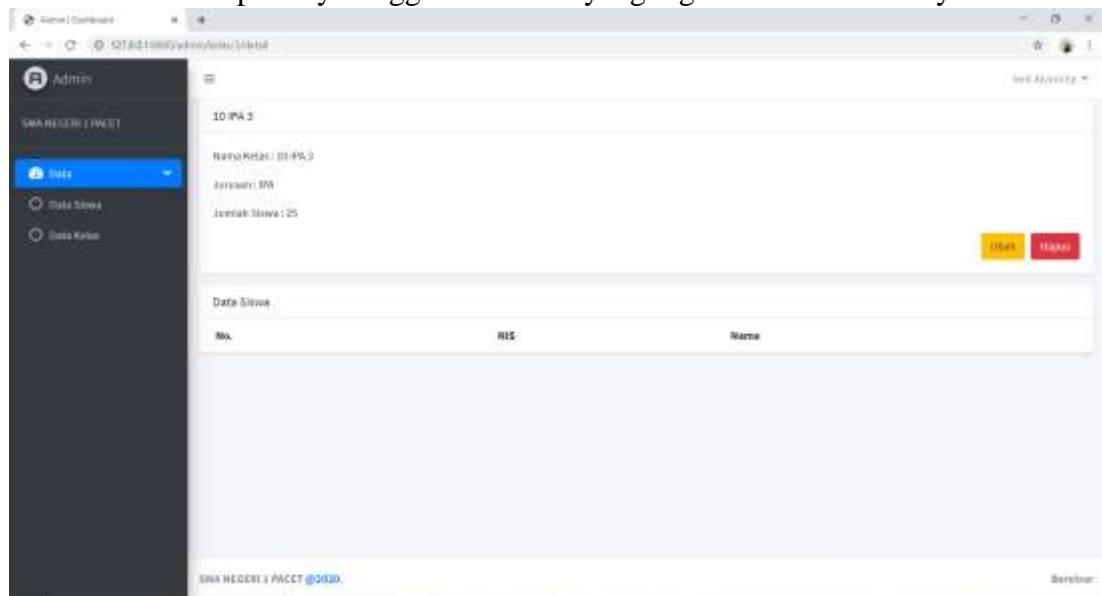
Data bertambah

30. Buat file detail.blade.php untuk menampilkan detail dari kelas tersebut

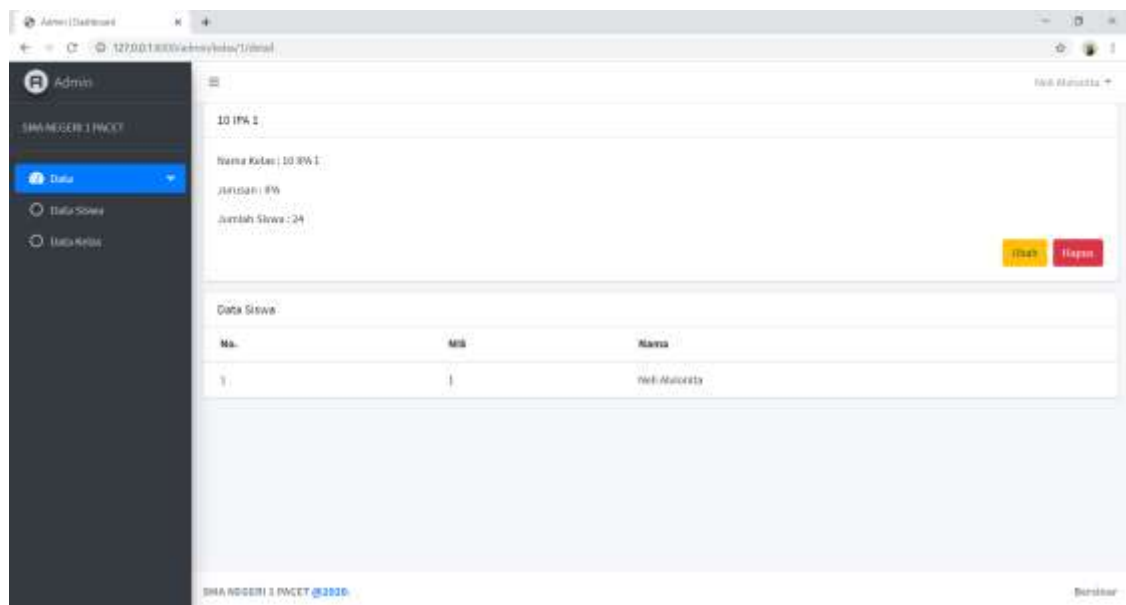




34. Untuk melihat tampilannya tinggal klik kelas yang ingin kita lihat detailnya



Disini selain kita bisa melihat detail dari kelas tersebut kita juga bisa mengubah dan menghapus, dimana kodenya akan dijelaskan di langkah berikutnya. Ditambah juga kita sekaligus bisa melihat nama-nama siswa yang ada di kelas tersebut. Untuk kelas 10 IPA 3 ini saya belum menambahkan siswa. Coba kita lihat di kelas lain yaitu 10 IPA 1



Sudah terlihat di kelas 10 IPA 1 terdapat 1 siswa yaitu Neli Alvionita

35. Nah setelah melihat detail siswa, kita langsung membuat file `ubah.blade.php` untuk mengubah data

```
resources > views > admin > kelas > ubah.blade.php > ...
@extends('layouts.master')
@section('title', 'Ubah Data Kelas')

@section('content')

Ubah Data</h3>

<form action="/admin/kelas/{{ $kelas->id }}" method="post">
@method('put')
@csrf

<label for="nama">Nama Kelas</label>
<input type="text" name="nama" id="nama" class="form-control" placeholder="Nama Kelas"
value="{{ $kelas->nama_kelas }}" aria-describedby="helpId">
<small id="helpId" class="text-muted">Contoh : 10 IPA 1</small>

<label for="jurusan">Jurusan</label>
<input type="text" name="jurusan" id="jurusan" class="form-control" placeholder="Jurusan (3 Huruf Saja)"
aria-describedby="helpId" value="{{ $kelas->jurusan }}">
<small id="helpId" class="text-muted">Contoh : IPA</small>

<label for="jumlah_siswa">Jumlah Siswa</label>
<input type="number" name="jumlah_siswa" id="jumlah_siswa" class="form-control" placeholder="Jumlah Siswa"
aria-describedby="helpId" value="{{ $kelas->jumlah_siswa }}">
<small id="helpId" class="text-muted">Contoh : 30</small>

<button type="submit" class="btn btn-success">Kirim</button>
</form>


```

36. Tambahkan fungsi `ubah` untuk mengubah data di `KelasController` dan fungsi `update` untuk mengirimkan data yang sudah diubah

```
public function ubah(Request $request)
{
 $kelas = kelas::find($request->id);
 return view('/admin/kelas/ubah', ['kelas' => $kelas]);
}

public function update(Request $request)
{
 $this->validate($request, [
 'nama' => 'required',
 'jurusan' => 'required',
 'jumlah_siswa' => 'required|numeric',
]);

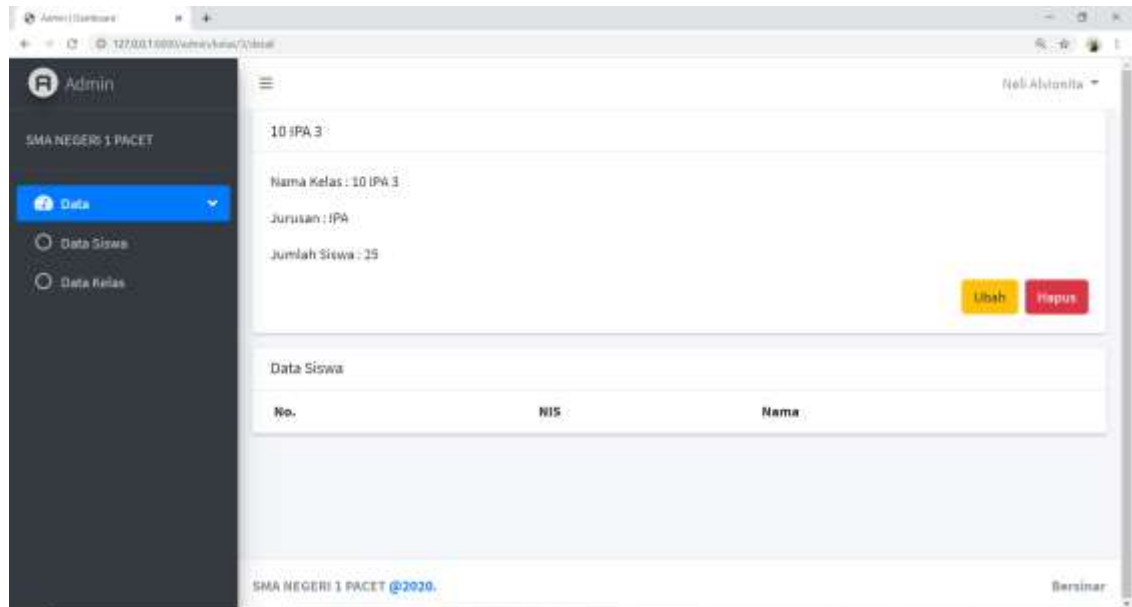
 $kelas = Kelas::findOrFail($request->id);
 $kelas->nama_kelas = $request->nama;
 $kelas->jurusan = $request->jurusan;
 $kelas->jumlah_siswa = $request->jumlah_siswa;

 if ($kelas->save()) {
 return redirect('/admin/kelas');
 }
}
}
```

37. Routing

```
34 Route::get('/admin/kelas/{id}/ubah', 'KelasController@ubah');
35 Route::put('/admin/kelas/{id}', 'KelasController@update');
```

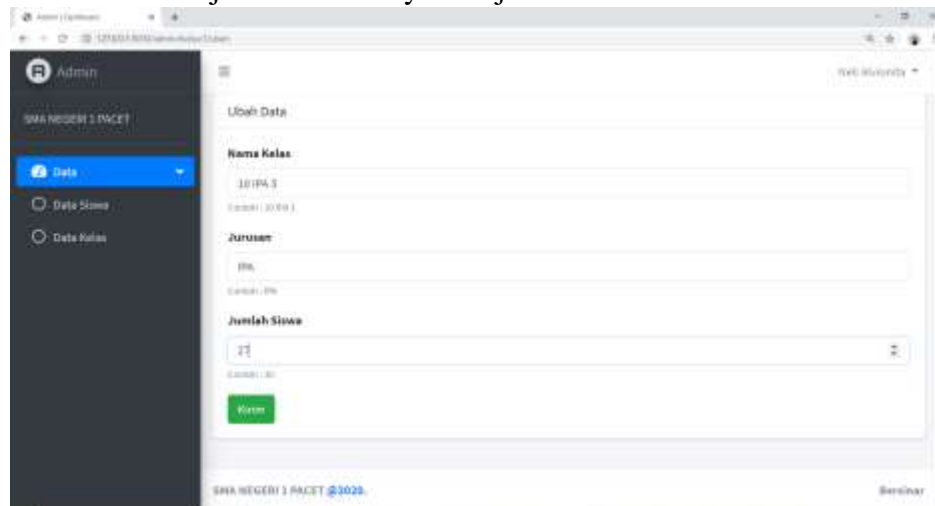
38. Tampilan misal kita ingin mengubah kelas 10 IPA 3, kita lihat detailnya terlebih dahulu kemudian klik ubah



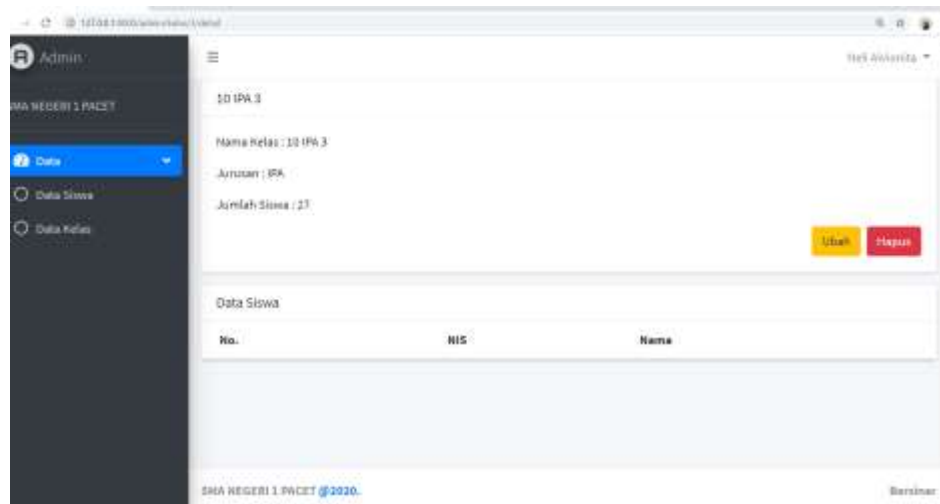
- Klik ubah



- Coba kita ubah jumlah siswa nya menjadi 27



➤ Kirimkan data



Data sudah terupdate

39. Untuk hapus data, tambahkan fungsi delete di KelasController

```
public function delete(Request $request)
{
 $kelas = Kelas::findOrFail($request->id);

 if (count($kelas->siswa) == 0) {
 $kelas->delete();
 return redirect('/admin/kelas')->with('message', 'Berhasil Menghapus Data');
 } else {
 return redirect('/admin/kelas')->with('message', 'Gagal Menghapus Data. Pindahkan siswa ke kelas lain te');
 }
}
```

Terdapat message, data kelas bisa terhapus apabila kelas tersebut tidak memiliki siswa, apabila kita ingin menghapus kelas yang masih ada siswanya tidak bisa, karena di dalam data siswa berelasi dengan data kelas. Jadi akan muncul pesan :

1. Jika kelas kosong maka pesannya adalah “Berhasil menghapus data”
2. Jika kelas tidak kosong maka pesannya “Gagal menghapus data. Pindahkan siswa ke kelas lain terlebih dahulu”

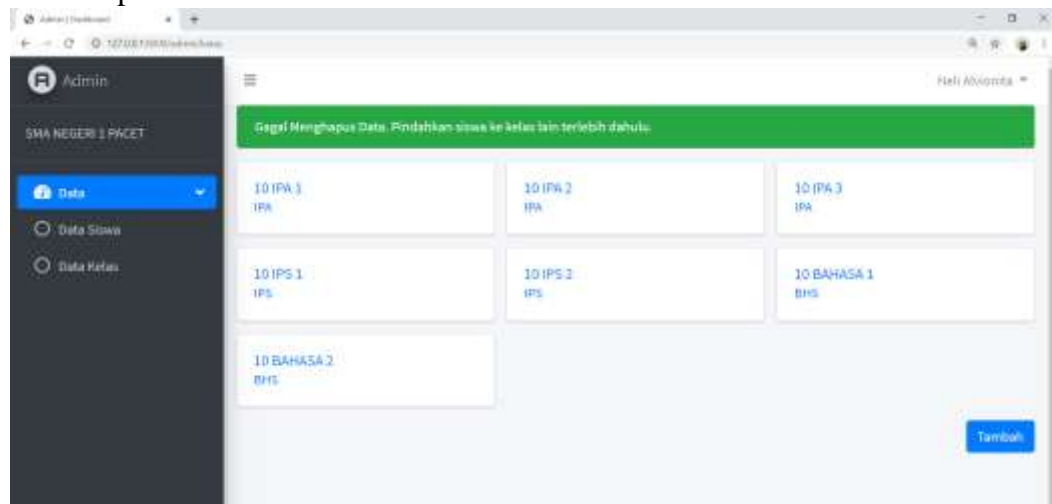
40. Routing

```
36 Route::delete('/admin/kelas/{id}', 'KelasController@delete');
37 /*Route Login */
```

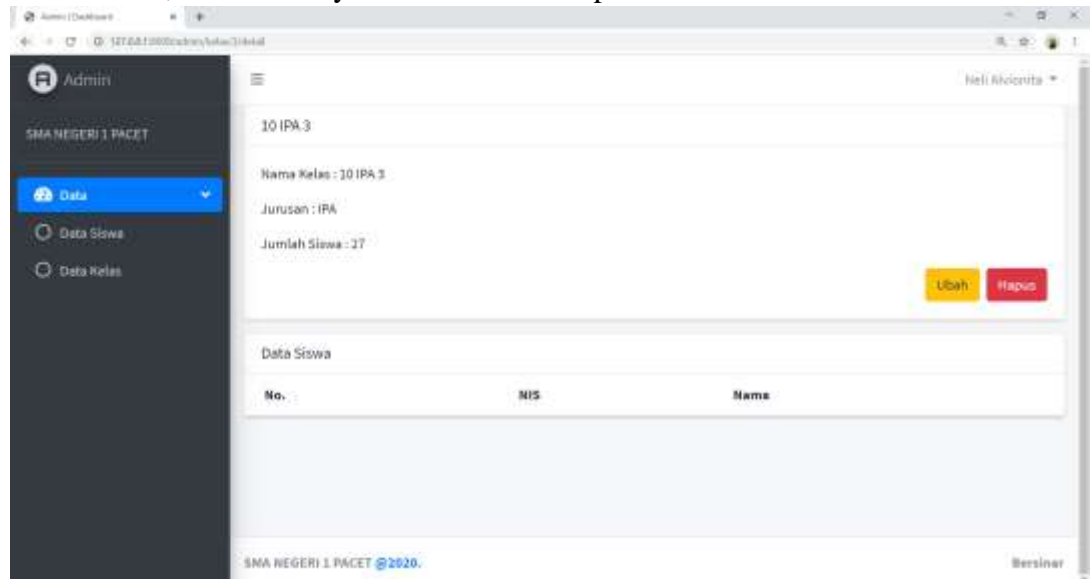
41. Misal kita menghapus data kelas 10 IPA 1, dimana kelas tersebut masih ada kelasnya, lihat detailnya kemudian klik hapus



- Muncul pesan



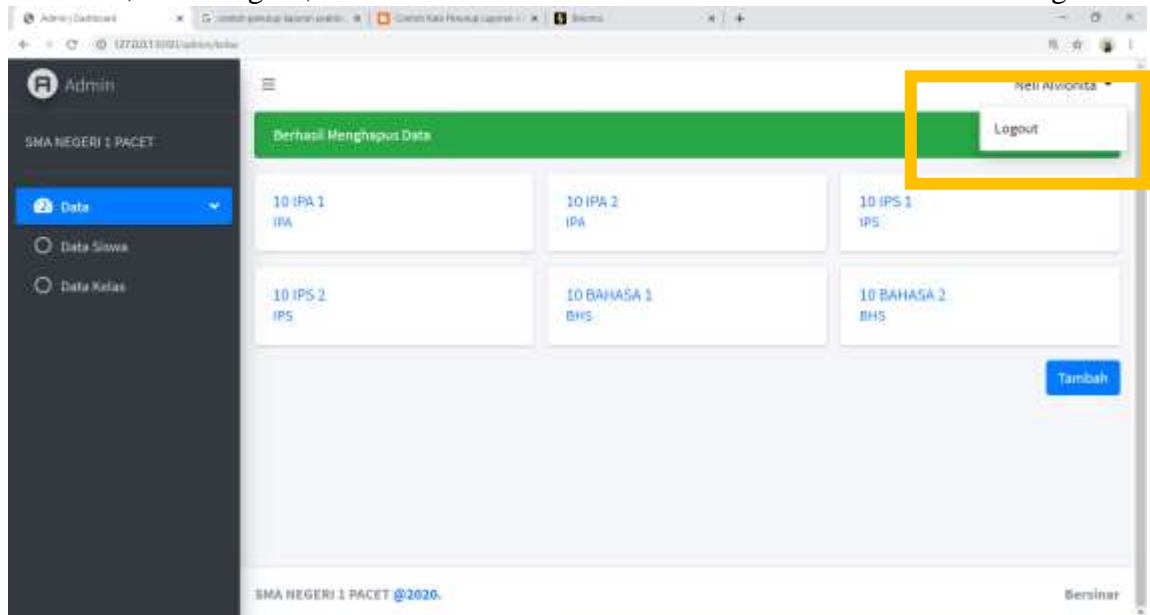
- Oke, kita lanjut menghapus data yang kelasnya kosong, misal kita hapus kelas 10 BAHASA 2, lihat detailnya kemudian klik hapus



- Data berhasil terhapus



42. Terakhir, untuk logout, klik nama dari admin di kanan atas maka akan muncul logout



43. Tampilan setelah logout akan mengarah ke halaman depan

