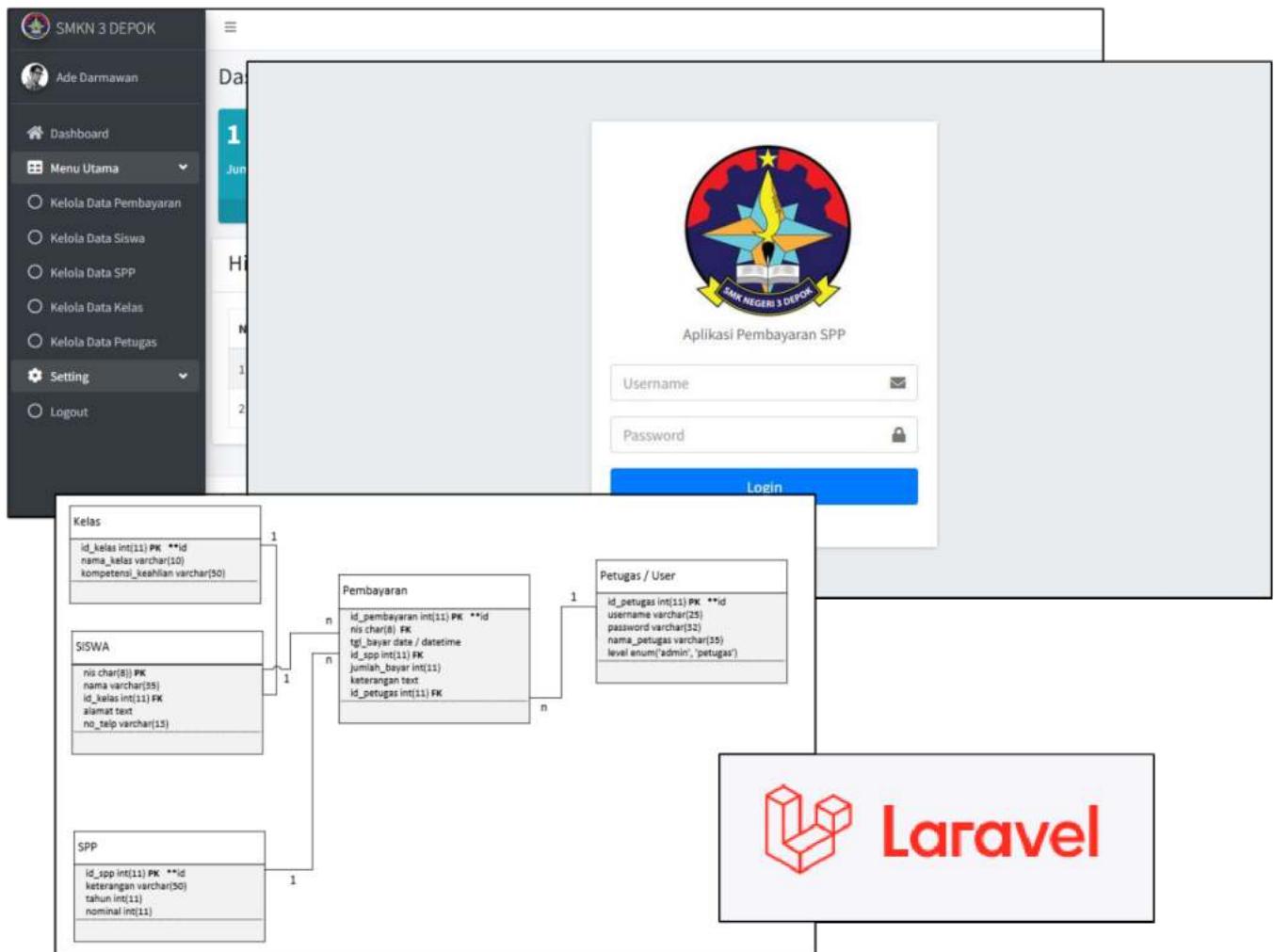


MODUL PEMBELAJARAN CONTOH - PEMBUATAN PROJECT PEMBAYARAN SPP DI SMK

(LARAVEL DASAR)



Modul ini dibuat sebagai bahan bagi para siswa untuk belajar dan mempersiapkan kemampuan untuk pembelajaran pemrograman web di kelas XII, dan Harap di ingat modul ini hanya membahas sebagian kecil hal yang berkaitan dengan pembuatan project aplikasi pembayaran spp dan mohon untuk digunakan sebagai mana mestinya

Disusun Oleh :

Guru Produktif Rekayasa Perangkat Lunak
Tahun : 2023

Untuk Bahan Projek Terkait Dapat di Akses Pada link Berikut :

1. <https://bit.ly/UJIKOM-RPL-2023> atau bit.ly/UJIKOM-RPL-2023

Berikut isi dari bahan yang dibagikan :

	AdminLTE-3.2.0.zip	1/29/2023 7:18 PM	WinRAR ZIP archive	28,333 KB
	datatable.zip	1/21/2023 9:33 PM	WinRAR ZIP archive	32 KB
	dist (tambahan assets untuk login).zip	1/2/2023 10:46 PM	WinRAR ZIP archive	5,855 KB
	laravel8.zip	1/29/2023 8:02 AM	WinRAR ZIP archive	11,824 KB
	laravel9.zip	1/29/2023 8:02 AM	WinRAR ZIP archive	22,976 KB
	LogoSekolah.zip	1/29/2023 5:45 PM	WinRAR ZIP archive	531 KB
	TEMPLATE.rar	12/30/2022 6:16 AM	WinRAR archive	63,859 KB

- **AdminLTE-3.2.0.zip** : Berisi template Admin LTE yang masih pure
- **datatable.zip** : Berisi assets atau source yang digunakan untuk membuat datatable (pencarian otomatis dan template table)
- **laravel8.zip** : Berisi source file Laravel 8 yang masih pure alias belum digunakan
- **laravel9.zip** : Berisi source file Laravel 9 yang masih pure alias belum digunakan

File Source pure Laravel ini sengaja dibuat agar siswa bisa langsung digunakan **namun apabila tetap ingin mengcreate kembali** folder Laravel versi masing-masing dipersilahkan Menggunakan :

```
composer create-project laravel/laravel="8.4.*" nama_project
```

atau

```
composer create-project laravel/laravel nama_project
```

- **LogoSekolah.zip** : Berisi sedikit logo sekolah yang bisa digunakan
- **Template.zip** : Berisi beberapa **template master/index, sidebar, dan dashboard** yang dapat digunakan untuk membuat view blade.
- **dist.zip** : Berisi assets yang digunakan untuk halaman **login.blade.php** apabila tampilan atau css tidak ter-load (namun silahkan dicek Kembali assets yang dicopykan dari template sesuai atau tidak)

The screenshot shows a dark-themed dashboard for SMKN 3 DEPOK. On the left, a sidebar menu includes 'Dashboard', 'Menu Utama' (selected), 'Kelola Data Pembayaran', 'Kelola Data Siswa', 'Kelola Data SPP', 'Kelola Data Kelas', 'Kelola Data Petugas', 'Setting' (selected), and 'Logout'. The main content area has a header 'Dashboard' and 'Home / Dashboard'. It features three cards: '1 Jumlah Siswa' (More info), '2 Jumlah Kelas' (More info), and 'Rp. 500.000 Total Transaksi (Bulan ini)' (More info). Below this is a section titled 'History Transaksi Terakhir' with a table:

No	NIS	SPP	Tanggal Bayar	Jumlah Bayar	Petugas
1	123 - Diah Puspita	SPP JANUARI 2023	2023-01-29	Rp. 250.000,00	1 - Ade Darmawan
2	123 - Diah Puspita	SPP FEBRUARI 2023	2023-01-29	Rp. 250.000,00	1 - Ade Darmawan

At the bottom, it says 'Copyright © 2014-2021 AdminLTE.io. All rights reserved.' and 'Version 3.2.0'.

CHEATSEAT YANG HARUS DIINGAT

Membuat Project :

```
composer create-project laravel/laravel nama_project
```

Menjalankan Project :

```
php artisan serve
```

Membuat Model Project :

```
php artisan make:model NamaTabel
```

Membuat Model + Buat Database migration di laravel :

```
php artisan make:model NamaTabel -m
```

Membuat controller :

```
php artisan make:controller NamaController
```

Membuat Migrasi / Migration hasil model / data create yg dibuat :

```
php artisan migrate
```

Jika Diperlukan untuk merollback / merefresh ulang migrasi :

```
php artisan migrate:rollback
```

```
php artisan migrate:refresh
```

YANG HARUS DIINGAT :

.ENV = untuk mensetting environment termasuk koneksi ke database yang dibuat

Folder Controller : di App/Http/Controller

File controller digunakan untuk membuat fungsi yang nantinya bisa dijalankan
Seperti membuat : index, create, show, update, destroy, dan cetak dan lainnya

Folder Model : di App/Http/Model

File Model digunakan untuk setting model yang terkoneksi ke table di database
Dimana yang perlu di ingat adalah :

`fillable < alias melakukan set atribut / field yang dapat di isi oleh aplikasi`
`primaryKey < set primaryKey`

`$table < set nama table`

`belongsTo < Many di database`

`hasMany < One di database`

File Web.php di folder Routes :

untuk membuat link / portal alamat yang dapat digunakan sistem

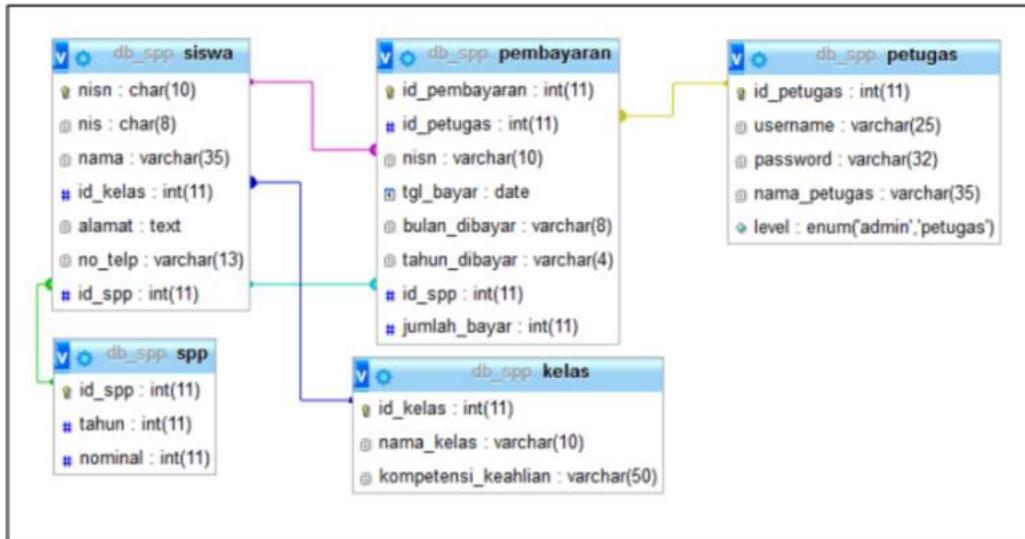
Folder Public : digunakan untuk menyimpan asset seperti gambar, css, js dan lainnya

Folder Resource-Views : untuk menyimpan view/blade atau tampilan berupa html dan php
lainnya yang berfungsi sebagai tempat coding template html dan lainnya.

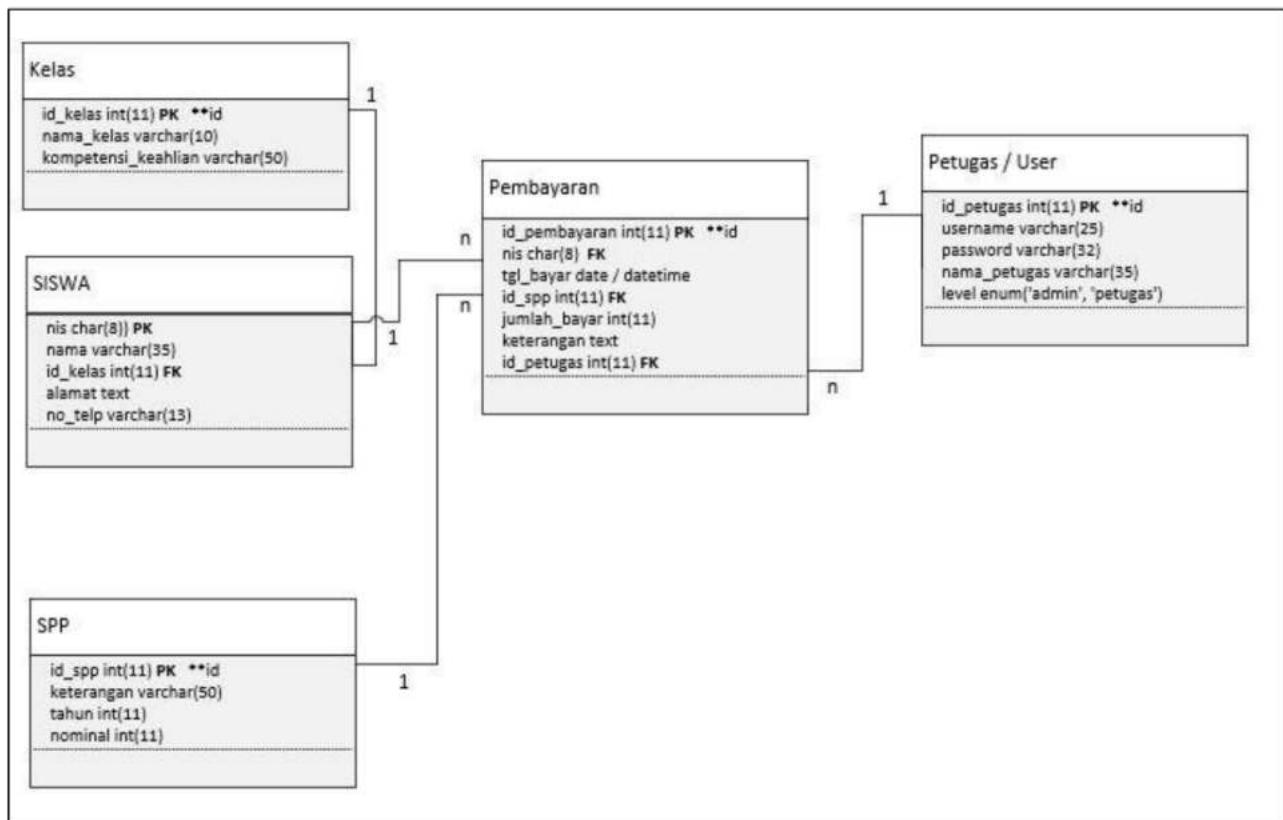
BAB 1 : PERSIAPAN DATABASE DAN TEMPLATE YANG DIGUNAKAN

Persiapan Database :

Database yang digunakan dalam projek kali ini merupakan database yang *bisa kalian sesuaikan kembali sesuai dengan kebutuhan aplikasi* yang akan kalian buat, Berikut Database sesuai dengan soal :



Dalam modul ini pun *database yang digunakan sudah di sesuaikan kembali sesuai kebutuhan contoh aplikasi* yang dibuat dengan gambaran database seperti berikut :



Database diatas dibuat dengan nama : “dbspp”

Noted : Kalian *boleh merubah nama database* sesuai dengan kenginan kalian (**asalkan tetap berhubungan dengan judul aplikasi**)

Untuk Pembuatan Database :

Kalian dapat menggunakan phpMyAdmin seperti biasa atau bisa membuat dengan **model -m** dengan contoh coding berikut :

Schema untuk skema table siswa :

```
Schema::create('siswas', function (Blueprint $table) {
    $table->char('nis', 8)->primary();
    $table->string('nama', 35);
    $table->integer('id_kelas');
    $table->string('alamat');
    $table->char('no_telp', 15);
    $table->timestamps();
});
```

File dibawah akan didapat apabila kalian sudah membuat model dengan menggunakan

Php artisan make:model NamaTabel -m

```
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14    public function up()
15    {
16        Schema::create('siswas', function (Blueprint $table) {
17            $table->char('nis', 8)->primary();
18            $table->string('nama', 35);
19            $table->integer('id_kelas');
20            $table->string('alamat');
21            $table->char('no_telp', 15);
22            $table->timestamps();
23        });
24    }
25}
```

Note :

Untuk sisa source skema model akan disimpan di lampiran.

*Tidak digunakan untuk orang yang belum mencoba menggunakan file migration

Persiapan Template / Admin Template :

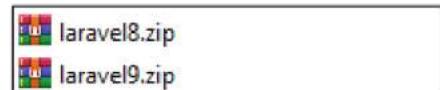
Anda diperbolehkan menggunakan template yang telah diberikan (<https://bit.ly/UJIKOM-RPL-2023>), dimana yang tersedia adalah template dasar yang sering digunakan yaitu **adminLTE** namun bagi kalian yang ingin menggunakan template lain harap hubungi guru produktif agar template tersebut bisa disatukan dengan bahan projek yang ada.

Name	Date modified	Type
adminlte	12/30/2022 5:28 AM	File folder
adminlte-new	12/30/2022 6:08 AM	File folder
concept-min	12/30/2022 5:29 AM	File folder
klorofil	12/30/2022 5:29 AM	File folder
monster-fix	12/30/2022 5:29 AM	File folder
sadmin2	12/30/2022 5:31 AM	File folder

CARA PENGGUNAAN TEMPLATE / AWALAN UNTUK SET PROJEK + CRUD USER / PENGGUNA

1. Siapkan folder *project laravel*

Anda dapat **mengekstrak** file dari bahan project dengan memilih salah satu versi :



2. Setelah file laravel di **extract** silahkan extract file TEMPLATE.rar

Name	Date modified	Type
adminlte	12/30/2022 5:28 AM	File folder
adminlte-new	12/30/2022 6:08 AM	File folder
concept-min	12/30/2022 5:29 AM	File folder
klorofil	12/30/2022 5:29 AM	File folder
monster-fix	12/30/2022 5:29 AM	File folder
sadmin2	12/30/2022 5:31 AM	File folder

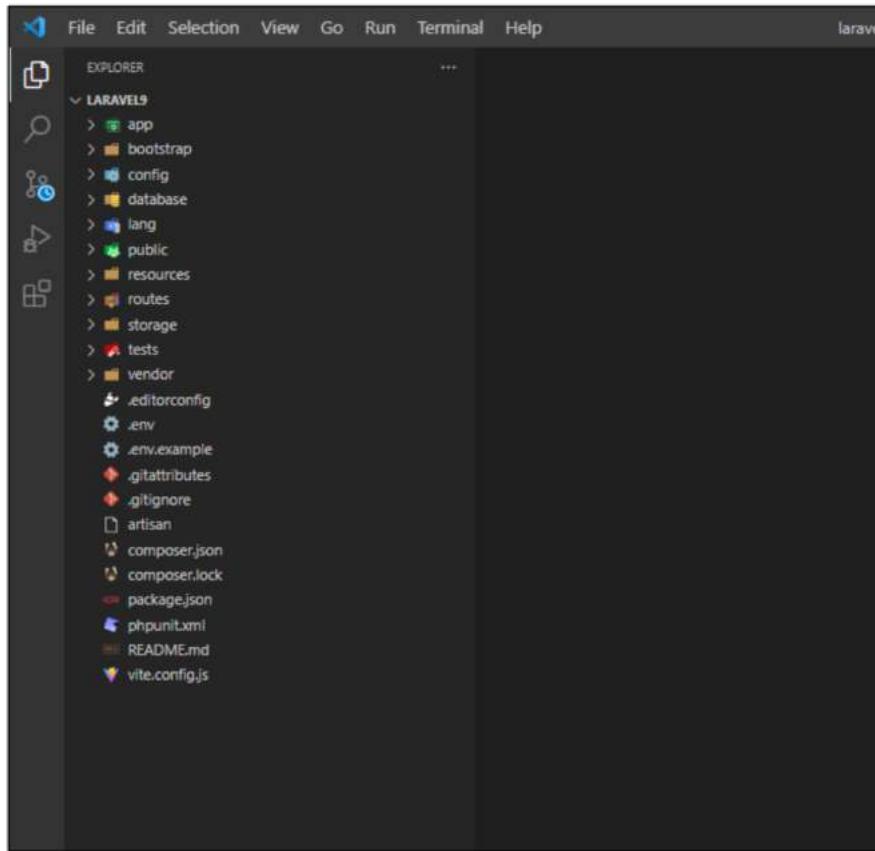
Silahkan pilih salah satu template yang bisa anda gunakan misal : disini kita menggunakan **adminlte-new** dengan isian folder berikut :

Name	Date modified	Type
admin	12/30/2022 6:07 AM	File folder
assets	1/29/2023 12:01 PM	File folder
config	12/30/2022 6:08 AM	File folder
database	2/7/2020 1:48 PM	File folder

3. Silahkan **copykan** folder **assets** dari folder tersebut ke folder **public** di **project laravel** anda

This PC > Documents > {BIMBINGAN PKL} > BAHAN UJIKOM > laravel9 > public >			
	Name	Date modified	Type
ss	assets	1/29/2023 8:44 PM	File folder
s	.htaccess	1/11/2023 7:50 AM	HTACCESS File
ts	favicon.ico	1/11/2023 7:50 AM	Icon
ts	index.php	1/11/2023 7:50 AM	PHP Source File
ts	robots.txt	1/11/2023 7:50 AM	Text Document

4. Setelah assets dicopykan maka Langkah berikutnya adalah membuka projek :
- Untuk membuka projek silahkan anda dapat membukanya dengan **Visual Studio Code dengan cara biasa.**
 - Atau silahkan yang memiliki caranya masing-masing pastikan saja visual studi code tepat pada projek anda



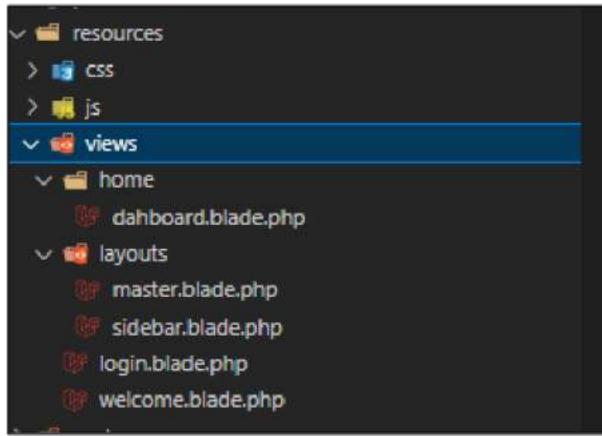
5. Setelah dipastikan terbuka maka silahkan rubah **.env** terlebih dahulu (rubah DB_DATABASE = localhost menjadi DB_DATABASE = dbspp) (atau sesuaikan dengan nama database yang sudah dibuat)

```

.env
.env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:qIS/1mdS/iJ/Rix9XSk1Likl0dzEFtD82mn77b5t488=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=dbspp
15 DB_USERNAME=root
16 DB_PASSWORD=

```

6. Setelah **.env** dirubah kemudian buatlah **2 folder** dan **4 file** di folder **Resource/views**
 Buatlah 2 folder : **home** dan **layouts** serta buat 4 file dengan nama :
Master.blade.php (dilayout), sidebar.blade.php (dilayout), dashboard.blade.php(dihome) dan login.blade.php di folder views (diluar home dan layout sejajar welcome)



7. Kemudian silahkan ambil *source template* di folder **adminlte-new** yang berada di folder admin

AHAN UJIKOM > TEMPLATE UJIKOM SEMENTARA > adminlte-new > admin >		
Name	Date modified	Type
errors	12/30/2022 6:07 AM	File folder
dashboard.php	12/30/2022 6:06 AM	PHP Source File
index.php	12/30/2022 6:12 AM	PHP Source File
sidebar.php	12/30/2022 6:14 AM	PHP Source File

File index -> untuk master

File dashboard -> untuk dashboard

Dan file sidebar -> untuk sidebar

Sedangkan untuk login silahkan nanti dibuat saja / dikosongkan terlebih dahulu

8. Kemudian sekarang buka tugas anda adalah merubah semua pemanggilan css dan js dengan menggunakan pemanggilan `{{asset(' ')}}` di laravel. *Silahkan rubah semua menjadi menggunakan `{{asset()}}`*
Misal : *Termasuk yang Javascript / js / script dan juga bagian gambar

```
<link rel="stylesheet" href="../assets/plugins/fontawesome-free/css/all.min.css">

<link rel="stylesheet" href="{{asset('assets/plugins/fontawesome-free/css/all.min.css')}}">
```

```
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Aplikasi Pembayaran SPP SMKN 3 Depok</title>

<!-- Google Font: Source Sans Pro -->
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
<!-- Font Awesome -->
{{-- jika mau silahkan semua asset diarahkan dengan menggunakan source seperti ini --}}
<link rel="stylesheet" href="{{asset('assets/plugins/fontawesome-free/css/all.min.css')}}">
<!-- Ionicons -->
<link rel="stylesheet" href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css">
<!-- Tempusdominus Bootstrap 4 -->
<link rel="stylesheet" href="{{asset('assets/plugins/tempusdominus-bootstrap-4/css/tempusdominus-bootstrap-4.min.css')}}">
<!-- iCheck -->
<link rel="stylesheet" href="{{asset('assets/plugins/iCheck-bootstrap/icheck-bootstrap.min.css')}}">
<!-- JQVMap -->
<link rel="stylesheet" href="{{asset('assets/plugins/jqvmap/jqvmap.min.css')}}">
<!-- Theme style -->
<link rel="stylesheet" href="{{asset('assets/css/adminlte.min.css')}}">
<!-- overlayScrollbars -->
<link rel="stylesheet" href="{{asset('assets/plugins/overlayScrollbars/css/OverlayScrollbars.min.css')}}">
<!-- Daterange picker -->
<link rel="stylesheet" href="{{asset('assets/plugins/daterangepicker/daterangepicker.css')}}">
<!-- summernote -->
```

9. Setelah Asset disesuaikan silahkan rubah beberapa hal juga masih di file master.blade.php

Rubah :

```
<?php  
include 'sidebar.php';  
?>
```

Menjadi :

```
@include('layouts.sidebar')
```

```
<?php  
include 'dashboard.php';  
?>
```

Menjadi :

```
@yield('content')
```

10. Setelah master.blade.php selesai baru ke dashboard.blade.php

Tambahkan :

```
@extends('layouts.master')  
@section('title', 'Dashboard')  
@section('content')  
<div class="content-wrapper">  
    ... (isi sourcecode dashboard)  
</div>  
@endsection  
@endsection <<< posisi ada dibagian paling bawah dashboard.blade.php
```

11. Setelah dashboard selesai maka silahkan rubah file sidebar.blade.php untuk menambahkan menu dengan cara menambahkan atau merubah yang ada pada menu silahkan cek sourcecode berikut :

```
Sidebar.blade.php  
<!-- Sidebar Menu -->  
<nav class="mt-2">  
    <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview"  
        role="menu" data-accordion="false">  
        <li class="nav-item">  
            <a href="/dashboard" class="nav-link">  
                <i class="nav-icon fas fa-home"></i>  
                <p>  
                    Dashboard  
                </p>  
            </a>  
        </li>
```

```
<li class="nav-item">
    <a href="#" class="nav-link">
        <i class="nav-icon fas fa-table"></i>
        <p>
            Menu Utama
            <i class="fas fa-angle-left right"></i>
        </p>
    </a>
<ul class="nav nav-treeview">
    <li class="nav-item">
        <a href="/pembayaran" class="nav-link">
            <i class="far fa-circle nav-icon"></i>
            <p>Kelola Data Pembayaran</p>
        </a>
    </li>
    <li class="nav-item">
        <a href="/siswa" class="nav-link">
            <i class="far fa-circle nav-icon"></i>
            <p>Kelola Data Siswa</p>
        </a>
    </li>
    <li class="nav-item">
        <a href="/spp" class="nav-link">
            <i class="far fa-circle nav-icon"></i>
            <p>Kelola Data SPP</p>
        </a>
    </li>
    <li class="nav-item">
        <a href="/kelas" class="nav-link">
            <i class="far fa-circle nav-icon"></i>
            <p>Kelola Data Kelas</p>
        </a>
    </li>
    <li class="nav-item">
        <a href="/user" class="nav-link">
            <i class="far fa-circle nav-icon"></i>
            <p>Kelola Data Petugas</p>
        </a>
    </li>
</ul>
```

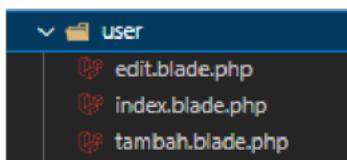
```

</li>
<li class="nav-item">
    <a href="#" class="nav-link">
        <i class="nav-icon fas fa-cog"></i>
        <p>
            Setting
            <i class="fas fa-angle-left right"></i>
        </p>
    </a>
    <ul class="nav nav-treeview">
        <li class="nav-item">
            <a href="/logout" class="nav-link">
                <i class="far fa-circle nav-icon"></i>
                <p>Logout</p>
            </a>
        </li>
    </ul>
</li>
</ul>
</nav>
<!-- /.sidebar-menu -->

```

12. Setelah **menu ditambahkan** buatlah sebuah folder pada views/**home user/petugas** dengan **isi 3 file** yaitu :

- index.blade.php
- tambah.blade.php
- edit.blade.php



13. Untuk coding program nya bisa menggunakan code program berikut :

Index.blade.php
<pre> @extends('layouts.master') @section('title', 'Data User') @section('content') <div class="content-wrapper">
 <section class="content"> <div class="container-fluid"> <div class="row"> <div class="col-lg-12"> <div class="card"> </div></pre>

```

        <div class="card-header">
<h4>Kelola Data User / Petugas</h4>
<a href="/user/tambah" class="btn btn-primary">Tambah Data</a>
        </div>
        <div class="card-body">
            <table id="example" class="table table-striped table-hover table-bordered">
                <thead>
                    <tr>
                        <th>No / ID</th>
                        <th>Nama Petugas</th>
                        <th>Username</th>
                        <th>Tanggal Daftar</th>
                        <th>Aksi</th>
                    </tr>
                </thead>
                <tbody>
                    @foreach ($user as $u)
                    <tr>
                        <td>{{ $u->id }}</td>
                        <td>{{ $u->name }}</td>
                        <td>{{ $u->username }}</td>
                        <td>{{ $u->created_at }}</td>
                        <td>
<a href="/user/{{ $u->id }}/edit" class="btn btn-warning">Edit</a>
<a href="/user/{{ $u->id }}/hapus" class="btn btn-danger" onClick="return confirm('Yakin Data Akan di Hapus?')">Hapus</a>
                        </td>
                    </tr>
                @endforeach
                </tbody>
            </table>
        </div>
    </div>
</div>
</div>
</section>
</div>
@endsection

```

```
tambah.blade.php

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


<br>
<section class="content">
<div class="container-fluid">
<div class="row">
<div class="col-lg-12">
<div class="card">
<div class="card-header">
<h4>Form Tambah Data User / Petugas</h4>
</div>
<div class="card-body">
<form action="/user/simpan" method="POST">
@csrf

<div class="form-group">
<label for="">Nama Petugas</label>
<input type="text" class="form-control"
name="name" id="" placeholder="Masukan Nama Petugas" required>
</div>

<div class="form-group">
<label for="">Username</label>
<input type="text" class="form-control"
name="username" id="" placeholder="Masukan Username untuk Akun" required>
</div>

<div class="form-group">
<label for="">Password</label>
<input type="password" class="form-control"
name="password" id="" placeholder="Masukan Password" required>
</div>

<button type="submit" class="btn btn-primary">Simpan</button>
<button type="reset" class="btn btn-secondary">Cancel</button>
</form>


```

```
        </div>
    </div>
</div>
</div>
</div>
</section>
</div>
@endsection
```

```
edit.blade.php (untuk edit bisa diambil dari tambah.blade.php)
hanya saya tinggal menambahkan value="{{$table->field}}"

@extends('layouts.master')
@section('title', 'Edit Data User')
@section('content')


<br>
<section class="content">


#### Form Edit Data User / Petugas



<form action="/user/{{ $user->id }}/update" method="POST">
@csrf


<label for="">Nama Petugas</label>
<input type="text" class="form-control" name="name"
value="{{ $user->name }}" placeholder="Masukan Nama Petugas" required>
</div>



<label for="">Username</label>
<input type="text" class="form-control"
name="username" value="{{ $user->username }}" placeholder="Masukan Username
untuk Akun" required>
</div>


```

14. Setelah folder user dibuat dan filenya sudah dibuat code seperti diatas maka tinggal menambahkan controller di folder App/Http/Controller

Dengan Cara buka new terminal / diterminal

Ketikan :

```
php artisan make:controller UserController
```

Kemudian buka file UserController

15. Buatlah fungsi-fungsi berikut pada UserController :

```
<?php

namespace App\Http\Controllers;
use Illuminate\Http\Request;

class UserController extends Controller
{
    public function index()
    }

    public function create()
    }
}
```

```
public function store(Request $request){  
}  
  
public function destroy($id) {  
}  
  
public function show($id){  
}  
  
public function update($id, Request $request){  
}  
}
```

Index : untuk menampilkan data & view user

Create : untuk menampilkan form tambah data alias tambah.blade.php

Store : digunakan untuk menyimpan data ke database

Destroy : digunakan untuk menghancurkan atau menghapus data

Show : digunakan untuk menampilkan data yang akan di edit atau edit.blade.php

Update : digunakan untuk mengupdate data atau merubah data di database

16. Lengkapi Source diatas dengan code berikut :

```
UserController.php  
<?php  
namespace App\Http\Controllers;  
use App\Models\User;  
use Illuminate\Http\Request;  
  
class UserController extends Controller  
{  
    public function index()  
    {  
        $user = User::all();  
        return view('home.user.index', compact(['user']));  
    }  
  
    public function create()  
    {  
        return view('home.user.tambah');  
    }  
}
```

```

public function store(Request $request)
{
    User::create([
        'name' => $request->name,
        'username' => $request->username,
        'password' => bcrypt($request->password),
        $request->except(['_token']),
    ]);
    return redirect('/user');
}

public function destroy($id)
{
    $user = User::find($id);
    $user->delete();
    return redirect('/user');
}

public function show($id)
{
    $user = User::find($id);
    return view('home.user.edit', compact(['user']));
}

public function update($id, Request $request)
{
    $user = User::find($id);
    $user->update([
        'name' => $request->name,
        'username' => $request->username,
        'password' => bcrypt($request->password),
        $request->except(['_token']),
    ]);
    return redirect('/user');
}
}

```

use App\Models\User; || digunakan untuk menghubungkan controller ke model agar penyimpanan atau perubahan data dapat singkron.

17. Setelah view di set dan Controller dibuat maka barulah buat codingan tambahan di **Routes / Web.php**

Web.php
<?php use App\Http\Controllers\UserController; use Illuminate\Support\Facades\Route; Route::get('/dashboard', function () { return view('home.dashboard'); }); Route::get('/user', [UserController::class, 'index']); Route::get('/user/tambah',[UserController::class, 'create']); Route::post('/user/simpan',[UserController::class, 'store']); Route::get('/user/{id}/hapus',[UserController::class, 'destroy']); Route::get('/user/{id}/edit',[UserController::class, 'show']); Route::post('/user/{id}/update',[UserController::class, 'update']); ?>

use App\Http\Controllers\UserController; || ditambahkan karena route membutuhkan instruksi atau event fungsi Ketika suatu alamat diakses dengan tujuan agar tiap alamat yang sudah dibuat dapat digunakan dengan baik sesuai fungsi yang diharapkan. Note: Route::get dan Route::post << memiliki kegunaan yang berbeda dimana get digunakan biasanya untuk menampilkan data saja sedangkan post digunakan untuk mengirimkan data biasanya ada erat hubungannya dengan insert atau input data dan edit data.

18. Setelah di Set Controller, Views, dan Routes tidak lupa model pun di sesuaikan: kebetulan karena input data kali ini tidak menggunakan email maka pada file **User** di lokasi :App/Http/Models

Lama	Baru
protected \$fillable = ['name', 'email', 'password',];	protected \$fillable = ['name', 'username', 'password',];

19. Setelah semua file dipastikan sudah selesai maka cobalah menjalankan program dengan menggunakan

Php artisan serve

Dan buka alamat 127.0.0.1:8000/user

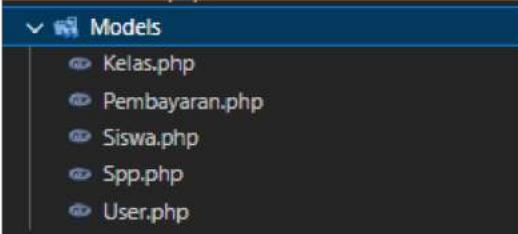
Kemudian Silahkan UjiCoba (untuk crud Petugas atau user)

BAB 2 : MEMBUAT CRUD UNTUK TABEL LAIN

Pada bab ini langsung ringkas saja:

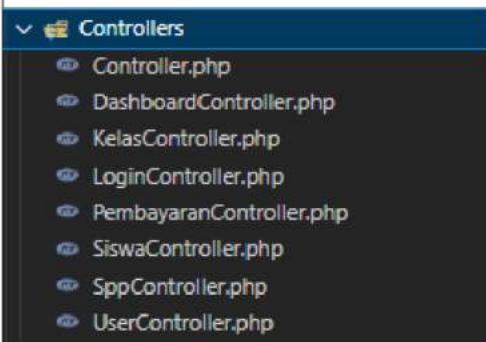
1. Silahkan buat Model dengan menggunakan

```
Php artisan make:model  
Buat :  
    php artisan make:model Kelas -m  
    php artisan make:model Spp -m  
    php artisan make:model Siswa -m  
    php artisan make:model Pembayaran -m
```



2. Kemudian Buat Controller

```
Php artisan make:controller  
Buat :  
    php artisan make:controller KelasController  
    php artisan make:controller SppController  
    php artisan make:controller SiswaController  
    php artisan make:controller PembayaranController  
    php artisan make:controller LoginController  
    php artisan make:controller DashboardController
```



3. Setelah Controller dibuat kemudian silahkan **set terlebih dahulu models** di folder App/Http/Models

```
Kelas.php  
<?php  
namespace App\Models;  
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;
```

```
class Kelas extends Model
{
    protected $fillable = ['nama_kelas', 'kompetensi_keahlian'];
    protected $table = 'kelas';
    public function Kelas()
    {
        return $this->hasMany(Kelas::class, 'id_kelas', 'id');
    }
}
```

Spp.php

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Spp extends Model
{
    protected $fillable = ['keterangan', 'tahun', 'nominal'];
    protected $table = 'spps';

    public function Spp()
    {
        return $this->hasMany(Spp::class, 'id_spp', 'id');
    }
}
```

Siswa.php

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Siswa extends Model
{
    protected $fillable = ['nis', 'nama', 'id_kelas', 'alamat', 'no_telp'];
    protected $table = 'siswas';
    protected $primaryKey = 'nis';
```

```

public function Kelas()
{
    return $this->belongsTo(Kelas::class, 'id_kelas', 'id');
}

public function Siswa()
{
    return $this->hasMany(Siswa::class, 'nis', 'nis');
}
}

```

Pembayaran.php

```

<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Pembayaran extends Model
{
    protected $fillable = ['nis', 'tgl_bayar', 'id_spp', 'jumlah_bayar',
'keterangan', 'id_petugas'];
    protected $table = 'pembayarans';

    public function Spp()
    {
        return $this->belongsTo(Spp::class, 'id_spp', 'id');
    }

    public function Siswa()
    {
        return $this->belongsTo(Siswa::class, 'nis', 'nis');
    }

    public function User()
    {
        return $this->belongsTo(User::class, 'id_petugas', 'id');
    }
}

```

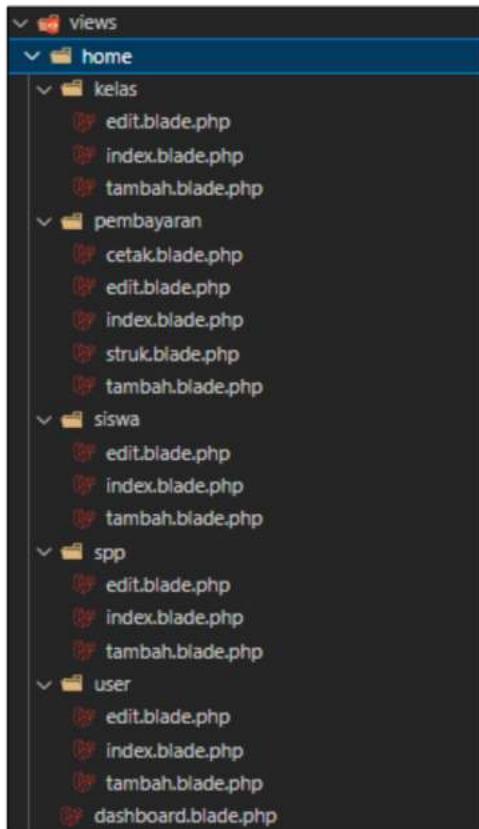
- Setelah model dibuat Maka silahkan berlanjut ke Controller :

Note: Untuk Controller kalian bisa mengambil contoh dari UserController

UserController.php (bagian store)	Yang harus dilakukan
<pre>public function store(Request \$request) { User::create(['name' => \$request->name, 'username' => \$request->username, 'password' => bcrypt(\$request->password), \$request->except(['_token']),]); return redirect('/user'); }</pre>	<ol style="list-style-type: none"> 1. Cukup rubah models yang digunakan menjadi models yang diinginkan misal <code>User::create</code> dirubah menjadi <code>Kelas::create</code> 2. Rubahlah <code>redirect</code> sesuai folder yang disiapkan nanti misal <code>redirect('/kelas')</code> 3. Rubah isian <code>create</code> dan <code>update</code> disesuaikan dengan isian atribut database / table atau models yang digunakan

5. Setelah Controller disesuaikan dan dirubah sesuai keinginan maka selanjutnya adalah **buatlah views-nya** Kalian cukup buat folder baru di **view/home** disesuaikan dengan **nama table yang akan dibuat CRUDnya** Misal : **kelas**

Maka buatlah folder kelas dengan isi : index.blade.php, tambah.blade.php, dan edit.blade.php



6. Silahkan untuk index, tambah, edit tugas kalian adalah menyesuaikan tampilan masing-masing dengan field atau isian table yang disesuaikan dengan database : Note (contoh codingan index dan lainnya bisa cek contoh aplikasi/lampiran yang diberikan guru)
7. Contoh hal yang perlu kalian rubah atau perhatikan Ketika membuat index atau tambah dan edit untuk table baru :

Kelas/index.blade.php

```

@extends('layouts.master')

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


<br>
<section class="content">
    <div class="container-fluid">
        <div class="row">
            <div class="col-lg-12">
                <div class="card">
                    <div class="card-header">
                        <h4>Kelola Data Kelas</h4>
                        <a href="/kelas/tambah" class="btn btn-primary">Tambah Data</a>
                    </div>
                    <div class="card-body">
                        <table id="example" class="table table-striped table-hover table-bordered">
                            <thead>
                                <tr>
                                    <th>No / ID</th>
                                    <th>Nama Kelas</th>
                                    <th>Kompetensi Keahlian</th>
                                    <th>Aksi</th>
                                </tr>
                            </thead>
                            <tbody>
                                @foreach ($kelas as $u)
                                <tr>
                                    <td>{{ $u->id }}</td>
                                    <td>{{ $u->nama_kelas }}</td>
                                    <td>{{ $u->kompetensi_keahlian }}</td>
                                    <td>
                                        <a href="/kelas/{{ $u->id }}/edit" class="btn btn-warning">Edit</a>
                                        <a href="/kelas/{{ $u->id }}/hapus" class="btn btn-danger" onClick="return confirm('Yakin Data Akan di Hapus?')">Hapus</a>
                                    </td>
                                </tr>
                            @endforeach
                        </tbody>
                    </table>
                </div>
            </div>
        </div>
    </div>
</section>


```

Kelas/tambah.blade.php

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

@section('title', 'Tambah Data Kelas')
@section('content')
<div class="content-wrapper">
    <br>
    <section class="content">
        <div class="container-fluid">
            <div class="row">
                <div class="col-lg-12">
                    <div class="card">
                        <div class="card-header">
                            <h4>Form Tambah Data Kelas</h4>
                        </div>
                        <div class="card-body">
                            <form action="/kelas/simpan" method="POST">
                                @csrf

                                <div class="form-group">
                                    <label for="">Nama Kelas</label>
                                    <input type="text" class="form-control" name="nama_kelas" id="" placeholder="Masukan Nama Kelas" required>
                                </div>
                            
```

Disini pastikan kalian membuat form-group atau inputan sesuai kebutuhan table

Kelas/edit.blade.php

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

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


<br>
<section class="content">
<div class="container-fluid">
<div class="row">
<div class="col-lg-12">
<div class="card">
<div class="card-header">
<h4>Form Edit Data Kelas</h4>
</div>
<div class="card-body">
<form action="/kelas/{{ $kelas->id }}/update" method="POST">
@csrf

<div class="form-group">
<label for="">Nama Kelas</label>
<input type="text" class="form-control" name="nama_kelas"
value="{{ $kelas->nama_kelas }}" placeholder="Masukan Nama Kelas" required>


```

Sebenarnya untuk edit dibandingkan menyalin ke file table lain alangkah lebih enaknya kalian menyalin dari tambah.blade.php karena cukup mengganti dan menambahkan beberapa baris saja yaitu value="{{{\$table->field}}}"

BAB 3 : SOURCE TAMBAHAN / FUNGSI TAMBAHAN

Untuk bab ini hanya akan dibahas beberapa saja dan sisanya anda dapat melihat di contoh aplikasi yang diberikan oleh guru atau di lampiran yang ada.

MEMBUAT LOGIN

1. Untuk membuat Halaman Login Silahkan kalian buat terlebih dahulu source code login.blade.php seperti berikut :

```
Login.blade.php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>SMKN 3 DEPOK</title>

    <!-- Google Font: Source Sans Pro -->
    <link rel="stylesheet"
        href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
    <!-- Font Awesome -->
    <link rel="stylesheet" href="{{ asset('assets/plugins/fontawesome-free/css/all.min.css') }}>
    <!-- iCheck bootstrap -->
    <link rel="stylesheet" href="{{ asset('assets/plugins/iCheck-bootstrap/icheck-bootstrap.min.css') }}>
    <!-- Theme style -->
    <link rel="stylesheet" href="{{ asset('assets/dist/css/adminlte.min.css') }}>
</head>
<body class="hold-transition login-page">
<div class="login-box">
    <!-- /.login-logo -->
    <div class="card">
        <div class="card-body login-card-body">
            <center>
                
            </center>
            <p class="login-box-msg">Aplikasi Pembayaran SPP</p>
```

```
<form action="/PostLogin" method="post">
    {{ csrf_field() }}
    <div class="input-group mb-3">
        <input type="username" class="form-control" placeholder="Username" name="username">
            <div class="input-group-append">
                <div class="input-group-text">
                    <span class="fas fa-envelope"></span>
                </div>
            </div>
        </div>
        <div class="input-group mb-3">
            <input type="password" class="form-control" placeholder="Password" name="password">
                <div class="input-group-append">
                    <div class="input-group-text">
                        <span class="fas fa-lock"></span>
                    </div>
                </div>
            </div>
            <div class="row">
                <div class="col-12">
                    <button type="submit" class="btn btn-primary btn-block">Login</button>
                </div>
                <!-- /.col -->
            </div>
        </div>
    </form>
    <br>
    @if (session('error'))
        <div class="alert alert-danger">
            {{ session('error') }}
        </div>
    @endif

    </div>
</div>
</div>
<!-- /.login-box -->

<!-- jQuery -->
<script src="{{ asset('assets/plugins/jquery/jquery.min.js')}}"></script>
<!-- Bootstrap 4 -->
<script src="{{
asset('assets/plugins/bootstrap/js/bootstrap.bundle.min.js')}}"></script>
<!-- AdminLTE App -->
```

```
<script src="{{ asset('assets/dist/js/adminlte.min.js')}}"></script>
</body>
</html>
```

Untuk halaman login :

Disini login tidak menggunakan email alias menggunakan username
Dan password yang ada pada table user/pengguna

Note:

1. Untuk gambar / img pastikan kalian menyimpan logo / gambar logo sekolah di public terlebih dahulu agar dapat menampilkan logo sekolah di halaman login
2. Untuk form action silahkan disesuaikan dengan route yang dibuat misal disana menggunakan /PostLogin << maka pastikan huruf besar dan kecil sama
3. Untuk input name="username" dan name="password" harus ada di form + tidak lupa **csrf** setelah pembuka <form>
4. Bagian if session dibawah merupakan alert yang digunakan untuk menampilkan pesan eror Ketika password atau username salah

2. Setelah source diatas dibuat selanjutnya ada buatlah source code **LoginController**

LoginController

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use Alert;
use Illuminate\Support\Facades\Auth;

class LoginController extends Controller
{
    public function index()
    {
        return view('login');
    }

    public function login(Request $request)
    {
        if (Auth::attempt($request->only('username', 'password'))){
            return redirect('/dashboard');
        }else{
            return redirect('/login')->with('error', 'Maaf Username dan Password yang Anda Masukan Salah!');
        }
    }
}
```

```

public function logout()
{
    Auth::logout();
    return view('login');
}

```

- Setelah Controller dibuat maka buatlah Routes di web.php seperti berikut :

Web.php (untuk Login)
<pre> Route::get('/', [LoginController::class, 'index']); Route::get('/login', [LoginController::class, 'index'])->name('login'); Route::post('/PostLogin', [LoginController::class, 'login']); Route::get('/logout', [LoginController::class, 'logout']); // Gunakan ini jika ingin membatasi hak akses ke backend sistem // Route::group(['middleware'=> ['auth']], function(){ // // }); </pre>

MEMBUAT FIELD ID KELAS MENJADI MENAMPAILKAN NAMA KELAS DI HALAMAN INDEX

- Hal ini digunakan di folder Siswa/index.blade.php
- Dimana di index siswa ada foreign key yaitu id_kelas sedangkan kita tidak ingin menampilkan id_kelas melainkan ingin menampilkan nama kelasnya seperti berikut : (*padahal di database menggunakan id_kelas*)

Kelola Data Siswa						
Tambah Data						
Show <input type="button" value="10"/> entries <input type="text" value="Search:"/>						
No	NIS	Nama	Kelas	No Telepon/HP	Aksi	
1	123	Diah Puspita	X PPLG 1	08946466488	Edit Hapus	

Showing 1 to 1 of 1 entries

Previous 1 Next

- Agar bisa muncul seperti gambar diatas cukup pastikan model yang sudah dibuat telah memiliki **hasMany** dan **belongsTo**.

Models/Siswa.php
<pre> <?php namespace App\Models; use Illuminate\Database\Eloquent\Factories\HasFactory; use Illuminate\Database\Eloquent\Model; </pre>

```

class Siswa extends Model
{
    protected $fillable = ['nis', 'nama', 'id_kelas', 'alamat', 'no_telp'];
    protected $table = 'siswas';
    protected $primaryKey = 'nis';

    public function Kelas()
    {
        return $this->belongsTo(Kelas::class, 'id_kelas', 'id');
    }

    public function Siswa()
    {
        return $this->hasMany(Siswa::class, 'nis', 'nis');
    }
}

```

Arti dari 2 hal tersebut adalah :

belongsTo << artinya id_kelas pada model siswa itu bergantung / berelasi kepada id di table atau model Kelas

hasMany << artinya nis yang ada disiswa berelasi kepada nis di table lain (pembayaran) dimanahasMany ini sebagai sumber atau one-nya sedangkan belongsTo = Many-nya pada relasi

4. Kemudian cukup tambahkan coding berikut pada file index.blade.php (siswa)

Home/Siswa/Index.blade.php
<pre> ... @foreach (\$siswa as \$u) <tr> <td>{{ \$loop->iteration }}</td> <td>{{ \$u->nis }}</td> <td>{{ \$u->nama }}</td> <td>{{ \$u->Kelas->nama_kelas }}</td> <td>{{ \$u->no_telp }}</td> <td> nis }}/edit" class="btn btn-warning">Edit nis }}/hapus" class="btn btn-danger" onClick="return confirm('Yakin Data Akan di Hapus?')">Hapus </td> </tr> @endforeach ... </pre> <p>Cukup Menambahkan : \$u->Kelas->nama_kelas <<< Kelas disana merupakan fungsi di models</p>

MEMBUAT DASHBOARD

1. Untuk membuat dashboard seperti berikut :

The screenshot shows a dark-themed dashboard for SMKN 3 DEPOK. On the left is a sidebar with user info (Ade Darmawan) and navigation links (Dashboard, Menu Utama, Setting). The main area has a title 'Dashboard'. It features three cards: 'Jumlah Siswa' (1), 'Jumlah Kelas' (2), and 'Total Transaksi (Bulan ini)' (Rp. 500.000). Below these is a section titled 'History Transaksi Terakhir' with a table:

No	NIS	SPP	Tanggal Bayar	Jumlah Bayar	Petugas
1	123 - Diah Puspita	SPP JANUARI 2023	2023-01-29	Rp. 250.000,00	1 - Ade Darmawan
2	123 - Diah Puspita	SPP FEBRUARI 2023	2023-01-29	Rp. 250.000,00	1 - Ade Darmawan

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

2. Untuk tersebut maka buatlah DashboardController dengan source berikut :

```
DashboardController

<?php

namespace App\Http\Controllers;

use App\Models\Kelas;
use App\Models\Pembayaran;
use App\Models\Siswa;
use Carbon\Carbon;
use Illuminate\Http\Request;

class DashboardController extends Controller
{
    public function index()
    {
        $jumlah_siswa = Siswa::count();
        $jumlah_kelas = Kelas::count();
        $pembayaran = Pembayaran::Select()->orderBy('tgl_bayar', 'desc')
            ->limit(5)
            ->get();

        $today = Carbon::today();
        $endDate = Carbon::today()->addDays(7);
        $total_minggu = Pembayaran::Select(Pembayaran::raw('SUM(jumlah_bayar) as total_price'))
            ->whereBetween('tgl_bayar', [$today, $endDate])->first();
    }
}
```

```

        return view('home.dashboard', compact('pembayaran', 'jumlah_siswa',
'jumlah_kelas'), ['total_minggu' => $total_minggu]);

    }

}

```

Pada controller ini kalian menggunakan compact dan array untuk menyimpan data yang dipanggil menggunakan query builder

- Setelah dibuat controllernya maka dilanjutkan untuk membuat codingan *di Web.php* atau controller yang disesuaikan lagi untuk dashboard

Web.php

```

<?php

use App\Http\Controllers\DashboardController;
use Illuminate\Support\Facades\Route;

Route::get('/dashboard', [DashboardController::class, 'index']);

Pastikan Dashboard controller di use di Web.php

```

MENAMBAHKAN PENCARIAN (DATATABLE)

Kelola Data Siswa						
Tambah Data <div style="display: flex; justify-content: space-between;"> Show <select>10</select> entries Search: <input type="text"/> </div>						
No	NIS	Nama	Kelas	No Telepon/HP	Aksi	
1	123	Diah Puspita	X PPLG 1	08946466488	Edit	Hapus
Showing 1 to 1 of 1 entries						
Previous 1 Next						

- Untuk menambahkan pencarian dengan menggunakan datatable pastikan anda menyiapkan asset **datatable** di folder Public, Silahkan extract datatable.zip (yang ada pada Bahan Projek) lalu hasil extract simpan ke folder public.
- Setelah datatable sudah masuk ke public maka selanjutnya adalah buka file **master.blade.php**

Master.blade.php

```

<html>
...
<link rel="stylesheet" href="{{asset('assets/plugins/summernote/summernote-bs4.min.css')}}>
<link rel="stylesheet" href="{{ asset('datatables/dataTables.bootstrap4.min.css') }}>
...

```

... Kemudian ke bagian paling bawah Tambahkan

```
<script>
    $(document).ready( function () {
        $('#example').DataTable();
    } );
</script>
</body>
</html>
```

Setelah ditambahkan maka setiap `<table>` di `index.blade` yang memiliki `<table id="example">` maka akan memiliki fungsi search dan style datatable

MEMBUAT CETAK / PRINT / STRUK

- Untuk membuat cetak pastikan anda membuat fungsi di controller (misal : di controller Pembayaran)

PembayaranController

```
public function create()
{
    $siswa = Siswa::all();
    $spp = Spp::all();
    return view('home.pembayaran.tambah', compact('siswa','spp'));
}

public function cetak()
{
    $pembayaran = Pembayaran::all();
    return view('home.pembayaran.cetak', compact('pembayaran'));
}

public function struk($id)
{
    $pembayaran = Pembayaran::find($id);
    return view('home.pembayaran.struk',compact('pembayaran'));
}
```

Tambahkan fungsi `cetak()` dan `struk($id)`

- Kemudian pastikan membuat Route di `web.php` untuk cetak data

Web.php

```
Route::get('/pembayaran', [PembayaranController::class,'index']);
..
Route::post('/pembayaran/{id}/update',[PembayaranController::class,'update']);
Route::get('/pembayaran/{id}/struk',[PembayaranController::class,'struk']);
Route::get('/pembayaran/cetak',[PembayaranController::class,'cetak']);
```

Tambahkan 2 Route Baru di Pembayaran yaitu struk dan cetak

3. Siapkan coding berikut untuk jenis cetak / struk yang disesuaikan

```
Cetak.blade.php

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Laporan Transaksi Pembayaran</title>
</head>
<body onload="window.print()">
    <center>
        <h2>Laporan Transaksi Pembayaran</h2>
//bagian ditandai warna merah dapat diambil di index.blade.php
        <table border="1">
            <thead>
                <tr>
                    <th>No</th>
                    <th>NIS</th>
                    <th>SPP</th>
                    <th>Tanggal Bayar</th>
                    <th>Jumlah Bayar</th>
                    <th>Petugas</th>
                </tr>
            </thead>
            <tbody>
                @foreach ($pembayaran as $u)
                <tr>
                    <td>{{ $loop->iteration }}</td>
                    <td>{{ $u->nis }} - {{ $u->Siswa->nama }} </td>
                    <td>{{ $u->Spp->keterangan }} </td>
                    <td>{{ $u->tgl_bayar }}</td>
                    <td>Rp. {{ number_format($u->jumlah_bayar,2,',','.')}}</td>
                    <td>{{ $u->id_petugas }} - {{ $u->User->name }}</td>
                </tr>
                @endforeach
            </tbody>
        </table>
    </center>

    <div class="kotak" style="margin-left: 800px;">
        <p>Bandung, .....</p>
    </div>

```

```

<br>
<br>
<p>Administrasi Sekolah</p>
</div>
</body>
</html>

```

Codingan diatas ini untuk bagian <table> bisa mengambil dari index.blade.php milik pembayaran, sisanya tinggal membuat manual misal dengan mengetikan **html:5 (enter)** lalu membuat **center h2** dan seterusnya

Note: Onload = "window.print()" -> digunakan untuk mengeksekusi **ctrl+p otomatis**

Ketika halaman dibuka

Struk.blade.php

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Struk</title>
</head>
<body onload="window.print()">
    <table width="400px">
        <tr align="center">
            <td colspan="3"><b>SMK 3 Depok</b></td>
        </tr>
        <tr align="center">
            <td colspan="3">Struk Pembayaran SPP</td>
        </tr>
        <tr>
            <td colspan="3"><hr></td>
        </tr>
        <tr>
            <td colspan="2"> Petugas : {{ Auth()->User()->name }}</td>
            <td align="right">Tanggal : <?php echo date("Y/m/d");?></td>
        </tr>
        <tr>
            <td colspan="3"><hr></td>
        </tr>
        <tr>
            <td> <b>NIS</b> </td>
            <td align="right">:</td>

```

```

        <td>{{ $pembayaran->nis }}</td>
    </tr>
    <tr>
        <td> <b>Nama</b></td>
        <td align="right">:</td>
        <td>{{ $pembayaran->Siswa->nama }}</td>
    </tr>
    <tr>
        <td> <b>SPP</b></td>
        <td align="right">:</td>
        <td>{{ $pembayaran->id_spp }} - {{ $pembayaran->Spp->keterangan }}</td>
    </tr>
    <tr>
        <td> <b>Jumlah Bayar</b></td>
        <td align="right">:</td>
        <td>Rp. {{ number_format($pembayaran->jumlah_bayar,2,',','.')}}</td>
    </tr>
    <tr>
        <td colspan="3"><hr></td>
    </tr>
    <tr>
        <td colspan="3">
            Note:
            <br>
            <b>- Struk ini sebagai bukti pembayaran</b>
            <br>
            <b>- Harap simpan dengan baik</b>
        </td>
    </tr>
</table>
</body>
</html>

```

Untuk bagian cetak struk ini belajarlah bagaimana cara menggunakan table di html
Misal penggunaan <tr> <td> align center left dan lainnya + colspan dkk

- Setelah halaman cetak dan struk dibuat maka tambahkan button untuk cetak data di index.blade.php (milik pembayaran)

Struk.blade.php
<pre> @extends('layouts.master') @section('title', 'Data Pembayaran') @section('content') <div class="content-wrapper">
 </pre>

```

<section class="content">
    <div class="container-fluid">
        <div class="row">
            <div class="col-lg-12">
                <div class="card">
                    <div class="card-header">
                        <h4>Kelola Data Pembayaran (History Transaksi)</h4>
                        <a href="/pembayaran/tambah" class="btn btn-primary">Tambah Data</a>
                        <a href="/pembayaran/cetak" target="_blank" class="btn btn-success"><span class="fa fa-print"></span> Keseluruhan</a>
                    </div>
                    <div class="card-body">
                        <table id="example" class="table table-striped table-hover table-bordered">
                            <thead>
                                <tr>
                                    <th>No</th>
                                    <th>NIS</th>
                                    <th>SPP</th>
                                    <th>Tanggal Bayar</th>
                                    <th>Jumlah Bayar</th>
                                    <th>Petugas</th>
                                    <th>Aksi</th>
                                </tr>
                            </thead>
                            <tbody>
                                @foreach ($pembayaran as $u)
                                <tr>
                                    <td>{{ $loop->iteration }}</td>
                                    <td>{{ $u->nis }} - {{ $u->Siswa->nama }} </td>
                                    <td>{{ $u->Spp->keterangan }} </td>
                                    <td>{{ $u->tgl_bayar }}</td>
                                    <td>Rp. {{ number_format($u->jumlah_bayar,2,',','.') }}</td>
                                    <td>{{ $u->id_petugas }} - {{ $u->User->name }}</td>
                                    <td>
                                        <a href="/pembayaran/{{ $u->id }}/edit" class="btn btn-sm btn-warning">Edit</a>
                                        <a href="/pembayaran/{{ $u->id }}/struk" target="_blank" class="btn btn-sm btn-success"><span class="fa fa-print"></span> Cetak Struk</a>
                                        <a href="/pembayaran/{{ $u->id }}/hapus" class="btn

```

Cukup membuat yang di tandai merah apabila index.blade.php milik pembayaran sudah ada.

LAMPIRAN

Web.php (total)

```
<?php

use App\Http\Controllers\DashboardController;
use App\Http\Controllers\KelasController;
use App\Http\Controllers\LoginController;
use App\Http\Controllers\PembayaranController;
use App\Http\Controllers\SiswaController;
use App\Http\Controllers\SppController;
use App\Http\Controllers\UserController;
use Illuminate\Support\Facades\Route;

Route::get('/', [LoginController::class, 'index']);
Route::get('/login', [LoginController::class, 'index'])->name('login');
Route::post('/PostLogin', [LoginController::class, 'login']);
Route::get('/logout', [LoginController::class, 'logout']);

// Gunakan ini jika ingin membatasi hak akses ke backend sistem
// Route::group(['middleware'=> ['auth']], function(){
// 
// });

Route::get('/dashboard', [DashboardController::class, 'index']);

Route::get('/user', [UserController::class, 'index']);
Route::get('/user/tambah',[UserController::class, 'create']);
Route::post('/user/simpan',[UserController::class, 'store']);
Route::get('/user/{id}/hapus',[UserController::class, 'destroy']);
Route::get('/user/{id}/edit',[UserController::class, 'show']);
Route::post('/user/{id}/update',[UserController::class, 'update']);

Route::get('/kelas', [KelasController::class, 'index']);
Route::get('/kelas/tambah',[KelasController::class, 'create']);
Route::post('/kelas/simpan',[KelasController::class, 'store']);
Route::get('/kelas/{id}/hapus',[KelasController::class, 'destroy']);
Route::get('/kelas/{id}/edit',[KelasController::class, 'show']);
Route::post('/kelas/{id}/update',[KelasController::class, 'update']);

Route::get('/spp', [SppController::class, 'index']);
Route::get('/spp/tambah',[SppController::class, 'create']);
Route::post('/spp/simpan',[SppController::class, 'store']);
Route::get('/spp/{id}/hapus',[SppController::class, 'destroy']);
Route::get('/spp/{id}/edit',[SppController::class, 'show']);
Route::post('/spp/{id}/update',[SppController::class, 'update']);

Route::get('/siswa', [SiswaController::class, 'index']);
Route::get('/siswa/tambah',[SiswaController::class, 'create']);
Route::post('/siswa/simpan',[SiswaController::class, 'store']);
Route::get('/siswa/{nis}/hapus',[SiswaController::class, 'destroy']);
Route::get('/siswa/{nis}/edit',[SiswaController::class, 'show']);
Route::post('/siswa/{nis}/update',[SiswaController::class, 'update']);
```

```
Route::get('/pembayaran', [PembayaranController::class, 'index']);
Route::get('/pembayaran/tambah',[PembayaranController::class, 'create']);
Route::post('/pembayaran/simpan',[PembayaranController::class, 'store']);
Route::get('/pembayaran/{id}/hapus',[PembayaranController::class, 'destroy']);
Route::get('/pembayaran/{id}/edit',[PembayaranController::class, 'show']);
Route::post('/pembayaran/{id}/update',[PembayaranController::class, 'update']);
Route::get('/pembayaran/{id}/struk',[PembayaranController::class, 'struk']);
Route::get('/pembayaran/cetak',[PembayaranController::class, 'cetak']);
```

KelasController

```
<?php

namespace App\Http\Controllers;

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

class KelasController extends Controller
{
    public function index()
    {
        $kelas = Kelas::all();
        return view('home.kelas.index', compact(['kelas']));
    }

    public function create()
    {
        return view('home.kelas.tambah');
    }
    public function store(Request $request)
    {
        Kelas::create([
            'nama_kelas' => $request->nama_kelas,
            'kompetensi_keahlian' => $request->kompetensi_keahlian,
            $request->except(['_token']),
        ]);
        return redirect('/kelas');
    }

    public function destroy($id)
    {
        $kelas = Kelas::find($id);
        $kelas->delete();
        return redirect('/kelas');
    }

    public function show($id)
    {
        $kelas = Kelas::find($id);
        return view('home.kelas.edit',compact(['kelas']));
    }
}
```

```

    }

    public function update($id, Request $request)
    {
        $kelas = Kelas::find($id);
        $kelas->update([
            'nama_kelas' => $request->nama_kelas,
            'kompetensi_keahlian' => $request->kompetensi_keahlian,
            $request->except(['_token']),
        ]);
        return redirect('/kelas');
    }
}

```

SppController

```

<?php

namespace App\Http\Controllers;

use App\Models\Spp;
use Illuminate\Http\Request;

class SppController extends Controller
{
    public function index()
    {
        $spp = Spp::all();
        return view('home.spp.index', compact(['spp']));
    }

    public function create()
    {
        return view('home.spp.tambah');
    }
    public function store(Request $request)
    {
        Spp::create([
            'tahun' => $request->tahun,
            'nominal' => $request->nominal,
            'keterangan' => $request->keterangan,
            $request->except(['_token']),
        ]);
        return redirect('/spp');
    }

    public function destroy($id)
    {
        $spp = Spp::find($id);
        $spp->delete();
        return redirect('/spp');
    }
}

```

```

    }

    public function show($id)
    {
        $spp = Spp::find($id);
        return view('home.spp.edit',compact(['spp']));
    }

    public function update($id, Request $request)
    {
        $spp = Spp::find($id);
        $spp->update([
            'tahun' => $request->tahun,
            'nominal' => $request->nominal,
            'keterangan' => $request->keterangan,
            $request->except(['_token']),
        ]);
        return redirect('/spp');
    }
}

```

PembayaranController

```

<?php

namespace App\Http\Controllers;

use App\Models\Pembayaran;
use App\Models\Siswa;
use App\Models\Spp;
use Illuminate\Http\Request;

class PembayaranController extends Controller
{
    public function index()
    {
        $pembayaran = Pembayaran::all();
        return view('home.pembayaran.index', compact('pembayaran'));
    }

    public function create()
    {
        $siswa = Siswa::all();
        $spp = Spp::all();
        return view('home.pembayaran.tambah', compact('siswa','spp'));
    }

    public function cetak()
    {
        $pembayaran = Pembayaran::all();
        return view('home.pembayaran.cetak', compact('pembayaran'));
    }
}

```

```
}

public function struk($id)
{
    $pembayaran = Pembayaran::find($id);
    return view('home.pembayaran.struk',compact('pembayaran'));
}

public function store(Request $request)
{
    Pembayaran::create([
        'nis' => $request->nis,
        'tgl_bayar' => $request->tgl_bayar,
        'id_spp' => $request->id_spp,
        'jumlah_bayar' => $request->jumlah_bayar,
        'keterangan' => $request->keterangan,
        'id_petugas' => Auth()->User()->id,
        $request->except(['_token']),
    ]);
    return redirect('/pembayaran');
}

public function destroy($id)
{
    $pembayaran = Pembayaran::find($id);
    $pembayaran->delete();
    return redirect('/pembayaran');
}

public function show($id)
{
    $pembayaran = Pembayaran::find($id);
    $siswa = Siswa::all();
    $spp = Spp::all();
    return view('home.pembayaran.edit',compact('pembayaran', 'spp', 'siswa'));
}

public function update($id, Request $request)
{
    $pembayaran = Pembayaran::find($id);
    $pembayaran->update([
        // 'nis' => $request->nis,
        'tgl_bayar' => $request->tgl_bayar,
        'id_spp' => $request->id_spp,
        'jumlah_bayar' => $request->jumlah_bayar,
        'keterangan' => $request->keterangan,
        'id_petugas' => Auth()->User()->id,
        $request->except(['_token']),
    ]);
    return redirect('/pembayaran');
}
}
```

SiswaController

```
<?php

namespace App\Http\Controllers;

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

class SiswaController extends Controller
{
    public function index()
    {
        $siswa = Siswa::all();
        return view('home.siswa.index', compact(['siswa']));
    }

    public function create()
    {
        $kelas = Kelas::all();
        return view('home.siswa.tambah', compact('kelas'));
    }

    public function store(Request $request)
    {
        Siswa::create([
            'nis' => $request->nis,
            'nama' => $request->nama,
            'id_kelas' => $request->id_kelas,
            'alamat' => $request->alamat,
            'no_telp' => $request->no_telp,
            $request->except(['_token']),
        ]);
        return redirect('/siswa');
    }

    public function destroy($nis)
    {
        $siswa = Siswa::find($nis);
        $siswa->delete();
        return redirect('/siswa');
    }

    public function show($nis)
    {
        $siswa = Siswa::find($nis);
        $kelas = Kelas::all();
        // $siswa = Siswa::where('nis', '=', $nis);
        return view('home.siswa.edit',compact('siswa', 'kelas'));
    }
}
```

```
public function update($nis, Request $request)
{
    $siswa = Siswa::find($nis);
    $siswa->update([
        'nama' => $request->nama,
        'id_kelas' => $request->id_kelas,
        'alamat' => $request->alamat,
        'no_telp' => $request->no_telp,
        $request->except(['_token']),
    ]);
    return redirect('/siswa');
}
```

UserController

```
<?php

namespace App\Http\Controllers;

use App\Models\User;
use Illuminate\Http\Request;

class UserController extends Controller
{
    public function index()
    {
        $user = User::all();
        return view('home.user.index', compact(['user']));
    }

    public function create()
    {
        return view('home.user.tambah');
    }
    public function store(Request $request)
    {
        User::create([
            'name' => $request->name,
            'username' => $request->username,
            'password' => bcrypt($request->password),
            $request->except(['_token']),
        ]);
        return redirect('/user');
    }

    public function destroy($id)
    {
        $user = User::find($id);
        $user->delete();
    }
}
```

```

        return redirect('/user');
    }

    public function show($id)
    {
        $user = User::find($id);
        return view('home.user.edit', compact(['user']));
    }

    public function update($id, Request $request)
    {
        $user = User::find($id);
        $user->update([
            'name' => $request->name,
            'username' => $request->username,
            'password' => bcrypt($request->password),
            $request->except(['_token']),
        ]);
        return redirect('/user');
    }
}

```

DashboardController

```

<?php

namespace App\Http\Controllers;

use App\Models\Kelas;
use App\Models\Pembayaran;
use App\Models\Siswa;
use Carbon\Carbon;
use Illuminate\Http\Request;

class DashboardController extends Controller
{
    public function index()
    {
        $jumlah_siswa = Siswa::count();
        $jumlah_kelas = Kelas::count();
        $pembayaran = Pembayaran::Select()->orderBy('tgl_bayar', 'desc')
            ->limit(5)
            ->get();

        $today = Carbon::today();
        $endDate = Carbon::today()->addDays(7);
        $total_minggu = Pembayaran::Select(Pembayaran::raw('SUM(jumlah_bayar) as total_price'))
            ->whereBetween('tgl_bayar', [$today, $endDate])->first();
    }
}

```

```
        return view('home.dashboard', compact('pembayaran', 'jumlah_siswa',
'jumlah_kelas'), ['total_minggu' => $total_minggu]);

    }

}
```

LoginController

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Alert;
use Illuminate\Support\Facades\Auth;

class LoginController extends Controller
{
    public function index()
    {
        return view('login');
    }

    public function login(Request $request)
    {
        if (Auth::attempt($request->only('username', 'password'))){
            return redirect('/dashboard');
        }else{
            return redirect('/login')->with('error', 'Maaf Username dan Password
yang Anda Masukan Salah!');
        }
    }

    public function logout()
    {
        Auth::logout();
        return view('login');
    }
}
```