

LAPORAN PROJEK AKHIR
PEMROGRAMAN WEB LANJUT
BLOG SINOPSIS KDRAMA
“TALKKDRAMA”



Oleh :
Wahyu Indah Puspitasari
1931710067
MI - 2D

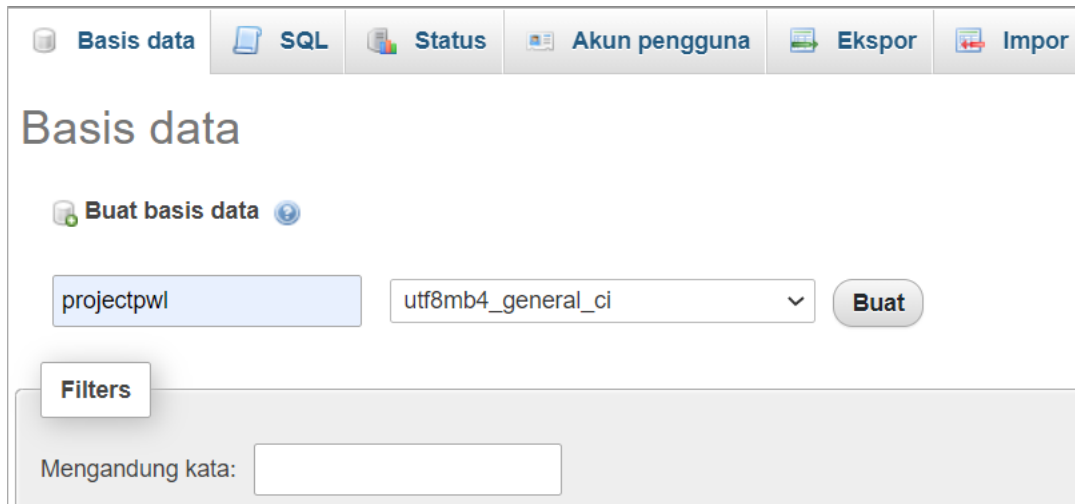
PRODI DIII-MANAJEMEN INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
2020

A. MEMBUAT PROJEK LARAVEL

Pastikan Composer telah terinstall, kemudian masuk ke CMD. Running composer pada Command Prompt, arahkan dulu direktori ke folder htdocs dengan memasukkan sintaks `cd c:\xampp\htdocs/`. Kemudian buat projek baru laravel dengan memasukkan sintaks `composer create-project --prefer-dist laravel/laravel <projectname> <versi>`.

B. MEMBUAT DATABASE

Nyalakan xampp, kemudian buka <http://localhost/phpmyadmin/> di browser. Buat database baru pada phpmyadmin sesuai keinginan. Disini saya membuat database dengan nama “projectpwl”.



The screenshot shows the 'Basis data' (Database) section of phpMyAdmin. At the top, there are tabs for 'Basis data', 'SQL', 'Status', 'Akun pengguna', 'Ekspor', and 'Impor'. Below the tabs, the 'Basis data' title is displayed. Underneath, there is a 'Buat basis data' (Create database) button. Below this, there is a form with two input fields: the first contains 'projectpwl' and the second is a dropdown menu showing 'utf8mb4_general_ci'. To the right of these fields is a 'Buat' (Create) button. Below the form, there is a 'Filters' section with a label 'Mengandung kata:' followed by an empty search input field.

C. MENGUBAH DATABASE DI .env

Buka Visual Studio Code kemudian buka file `.env` yang ada pada direktori paling luar dalam projek laravelnya. Koneksikan laravel dengan database yang telah dibuat.

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

D. MEMBUAT PAGE LOGIN

Buka folder migration terdapat file `create-users-table`. Pada folder ini laravel sudah menyediakan file migration untuk table user. Kita bisa menambahkan field level/role jika ingin membuat multiple user.

Source untuk menambah field level

```
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->increments('id');
        $table->string('name');
        $table->string('level');
```

```

        $table->string('email')->unique();
        $table->timestamp('email_verified_at')->nullable();
        $table->string('password');
        $table->rememberToken();
        $table->timestamps();
    });
}

```

Setelah itu kita bisa migrate table user di terminal agar terdaftar di database. Berikut Perintahnya :

php artisan migrate

Kemudian cek di database. Jika di database sudah ada table user maka migration table berhasil dilakukan.

users Jelajahi Struktur Cari Tambahkan Kosongkan Hapus 3 InnoDB utf8mb4_unicode_ci 32.0 KB

Setelah itu, kita membuat seeder untuk User agar bisa membuat akun user. Berikut perintah membuat seeder :

php artisan make:seed UserSeeder

Kemudian kita buka file seeder yang barusan dibuat. Untuk mengetikkan perintah yang berfungsi membuat data usernya. Dan karena kita memanggil model User pada class UserSeeder dan memanggil fungsi string dari laravel maka kita tidak boleh lupa mengimport *“use App\User;”* dan *“use Illuminate\Support\Str;”*.

Source Code UserSeeder

```

<?php

use Illuminate\Database\Seeder;
use Illuminate\Support\Str;
use App\User;
class UserSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        User::truncate();
        User::create([
            'name' => 'Admin talk',
            'level' => 'admin',

```

```

        'email' => 'admintalk@admin.com',
        'password' => bcrypt('talkactive'),
        'remember_token' => Str::random(60),
    ]);
}
}

```

Jika sudah kita buka file DatabaseSeeder di folder seeders. Kita aktifkan perintah berikut :

```

public function run()
{
    $this->call(UserSeeder::class);
}

```

Kalau sudah, kita bisa run di terminal dengan perintah :

php artisan db:seed;

Selanjutnya kita refresh phpmyadmin untul mengecek data yang baru dibuat.

	id	name	level	email	email_verified_at	password	remember_token
<input type="checkbox"/> Ubah <input checked="" type="checkbox"/> Salin <input checked="" type="checkbox"/> Hapus	1	Admin talk	admin	admintalk@admin.com	NULL	\$2y\$10\$djrk00CyyYCGYZi9Ls1.NwSGnMW4gti6yJmGAJa0...	hfuj2eahv6M0isZouGrJSqug6x0wWfd9rhgG8WICV5tyuoPey...

Selanjutnya jika kita ingin menambah data user yang baru kita bisa menggunakan perintah tinker. Kita bisa ketikkan perintah ***“php artisan tinker”*** di terminal. Untuk pengisian tinker kita sesuaikan dengan data pada model User.php.

```

protected $fillable = [
    'name', 'level','email', 'password',
];

```

Berikut adalah contoh pengisian dengan perintah tinker :

```

PS C:\xampp\htdocs\laravelwahyu> php artisan tinker
Psy Shell v0.10.4 (PHP 7.4.4 - cli) by Justin Hileman
>>> $user = new \App\User;
=> App\User {#3228}
>>> $user->name = "wahyu Indah"
=> "wahyu Indah"
>>> $user->level = "admin"
=> "admin"
>>> $user->email = "wahyu@gmail.com"
=> "wahyu@gmail.com"
>>> $user->password = bcrypt("12345")
=> "$2y$10$IvYyAASPer4EdBAQAABgIOqinlprGVanQZorvFdZwbuzU0reAZMge"
>>> $user->remember_token=Str::random(60)
=> "J15BQSSwMinPwvSKoDQnLU0Ha62hlw9FT7GGwigexHLKEu2r3vre8qyicfp4"
>>> $user->save()
=> true

```

Data akan otomatis tercatat di database.

	id	name	level	email	email_verified_at	password	remember_token
<input type="checkbox"/> Ubah <input checked="" type="checkbox"/> Salin <input checked="" type="checkbox"/> Hapus	1	Admin talk	admin	admintalk@admin.com	NULL	\$2y\$10\$djrk00CyyYCGYZi9Ls1.NwSGnMW4gti6yJmGAJa0...	hfuj2eahv6M0isZouGrJSqug6x0wWfd9rhgG8WICV5tyuoPey...
<input type="checkbox"/> Ubah <input checked="" type="checkbox"/> Salin <input checked="" type="checkbox"/> Hapus	2	Wahyu Indah	admin	wahyu@gmail.com	NULL	\$2y\$10\$IvYyAASPer4EdBAQAABgIOqinlprGVanQZorvFdZwbuzU0reAZMge	J15BQSSwMinPwvSKoDQnLU0Ha62hlw9FT7GGwigexHLKEu2r3...
<input type="checkbox"/> Ubah <input checked="" type="checkbox"/> Salin <input checked="" type="checkbox"/> Hapus	3	Rosie Rose	user	rose5@gmail.com	NULL	\$2y\$10\$4mZ25diedZnPnmlE4JLGBTQyH8M8MavpXEm8Pa...	EyTYFUtm8kugsbJxTYgysfekiRmXeLbbovZ0mKJ3uPB5ZvqZ...

Selanjutnya kita bisa membuat template untuk halaman login.

Source Code Login.blade.php :

```
<!doctype html>
<html lang="en" dir="ltr">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" href="{{asset('/css/lgn.css')}}">
    <title>Login</title>
  </head>
  <body>
    <nav>
      <div class="logo">
        <a href="">tal\k drama</a>
      </div>
    </nav>
    <div class="contain">
      <div class="form-box">
        <div class="button-box">
          <div id="btn"></div>
          <button type="button" class="toggle-btn">Login</button>
        </div>
        <form class="input-group" action="{{route('postlogin')}}" method="post">
          @csrf
          @method('POST')
          <input type="email" class="input-
field" name="email" placeholder="Email" required>
          <input type="password" class="input-
field" name="email" placeholder="Password" required>
          <button type="submit" class="submit-btn">LOGIN</button>
        </form>
      </div>
    </div>
  </body>
</html>
```

Css untuk halaman Login :

```
*{
  margin: 5px;
  padding: 0;
  color: rgb(155, 0, 0);
  box-sizing: border-box;
```

```
font-family: 'Arial Narrow Bold', sans-serif;
text-decoration: none;
}
body{
    background: grey;
}
nav{
    background: black;
}
nav::after{
    content: '';
    clear: both;
    display: table;
}
nav .logo{
    float: left;
    font-style: italic;
    font-weight: bold;
    font-size: 40px;
    font-weight: 600px;
    line-height: 70px;
    padding-left: 40px;
}
.contain{
    position: absolute;
}
.form-box{
    width: 380px;
    height: 270px;
    position: relative;
    margin-top: 100px;
    margin-left: 110%;
    background: rgb(255, 206, 235);
    padding: 5px;
}
.button-box{
    width: 220px;
    margin-top: 30px;
    margin-left: 80px;
    position: relative;
    box-shadow: 0 0 30px 10px rgb(255, 124, 124);
}
.toggle-btn{
    padding: 10px 85px;
    cursor: pointer;
}
```

```
background: transparent;
border: 0;
outline: none;
position: relative;
}
#btn{
  top: 0;
  left: 0;
  position: absolute;
  width: 110px;
  height: 100%;
  border-radius: 30px;
  transition: .5s;
}
.input-group{
  top: 100px;
  position: absolute;
  width: 280px;
  margin-left: 50px;
}
.input-field{
  width: 100%;
  padding: 10px 30px;
  margin: 5px 0;
  border-left: 0;
  border-top: 0;
  border-right: 0;
  border-bottom: 1px solid darkmagenta;
  outline: none;
  background: transparent;
}
.submit-btn{
  width: 55%;
  padding: 10px 30px;
  cursor: pointer;
  display: block;
  margin-top: 20px;
  margin-left: 60px;
  background: black;
  border: 0;
  outline: none;
  border-radius: 30px;
}
.check-box{
  margin: 30px 10px 30px 0px;
```

```

}
span{
    color: gray;
    bottom: 64px;
    position: absolute;
}
#login{
    left: 50px;
}

```

Selanjutnya kita membuat Controller untuk login. Kita bisa ketikkan perintah berikut di terminal untuk membuat controller.

Php artisan make:controller LogintatController

Selanjutnya kita buka file controller di folder Http->Controller. Pertama kita buat function untuk mengakses halaman login. Jangan lupa untuk menambahkan ***“use Auth;”***

```

<?php

namespace App\Http\Controllers;

use Auth;
use Illuminate\Http\Request;

class LogintatController extends Controller
{
    public function login()
    {
        return view('ProjectDrama.Login');
    }

    public function postlogin(Request $request)
    {
        if(Auth::attempt($request->only('email','password'))){
            return redirect('/kdrama');
        }
        return redirect('/loginta');
    }
}

```

Selanjutnya adalah membuat route agar halaman login dapat diakses di browser

```

Route::get('/loginta', 'LogintatController@login')->name('login');

```

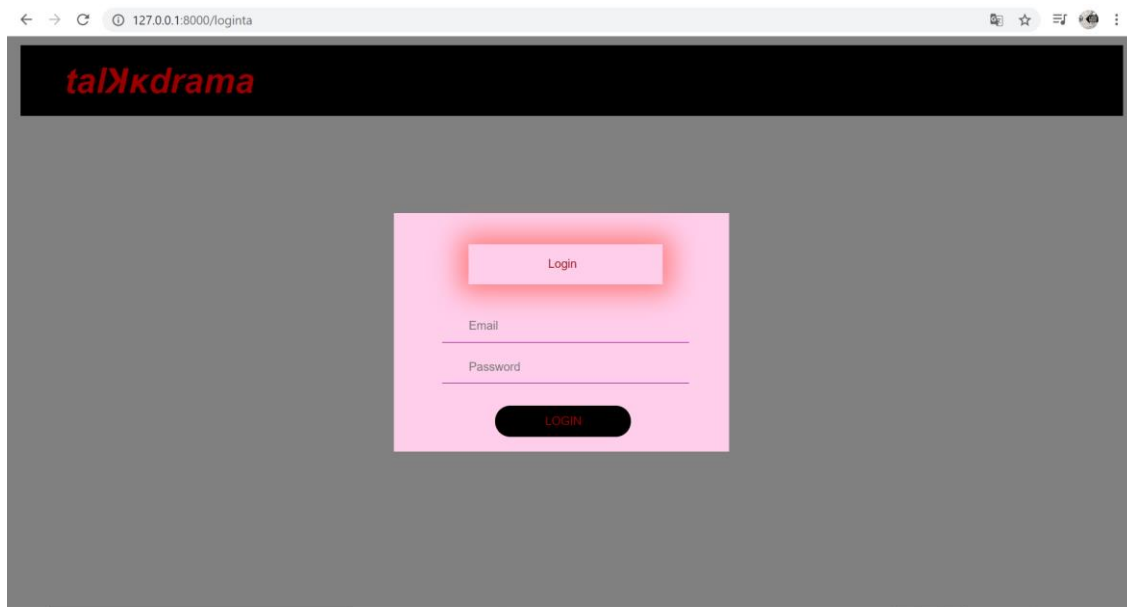


```
Route::POST('/postlogin', 'LogintatController@postlogin')->name('postlogin');
```

Jika ingin mengakses halaman yang telah dibuat ke browser jangan lupa mengaktifkan perintah :

php artisan serve

Berikut adalah tampilan login yang telah dibuat :



E. MEMBUAT MASTER CLASS

Selanjutnya kita perlu membuat master class. Master class dibuat agar kita tidak menuliskan perintah atau source code yang sama pada class yang berbeda. Kita bisa membuat folder untuk menampung class master kita. Disini saya membuat folder layouts dengan class master Masterkd.blade.php.

Source Code Masterkd.blade.php

```
<!doctype html>
<html lang="en" dir="ltr">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>@yield('title')</title>
  </head>
  <body>
    <nav>
      <div class="logo">
```

```

        <a href="/kdrama">tal\kdrama</a>
    </div>

    <ul>
        <li>
            <a href="/romance">ROMANCE</a>
        </li>
        <li>
            <a href="/action">ACTION</a>
        </li>
        <li>
            <a href="/melo">MELODRAMA</a>
        </li>
        <li>
            <a href="/Comedy">COMEDY</a>
        </li>
        <li>
            <a href="/history">HISTORY</a>
        </li>
        <li>
            <a href="/crudDrama">EDIT DRAMA</a>
        </li>
        <li>
            <a href="/kategori">KATEGORI</a>
        </li>
        <li>
            <a href="/aboutkd">ABOUT ME</a>
        </li>
        <li>
            <a href="/loginta">LOGOUT</a>
        </li>
    </ul>
</nav>

<section class="greeting">
    @yield('greeting')
</section>

<section>
    @yield('content')
</section>
</body>
</html>

```

F. MEMBUAT HALAMAN HOME

Untuk membuat tampilan halaman home. Disini saya menampilkan slideshow beberapa poster drama korea untuk menunjukkan bahwa web ini membahas tentang drama Korea. Dalam class home ini kita perlu mextends class Master. Dalam section greeting adalah upacan selamat datang di web talkkdrama.

Source Code kdrama.blade.php

```
@extends('layouts.Masterkd')
@section('title', 'Talk-Kdrama')
<link rel="stylesheet" type="text/css" href="{{asset('/css/kd.css')}}">

@section('greeting')
<div>
    <center>
        <h1 class="display-4">Welcome To Talk-Kdrama</h1>
    </center>
</div>
@endsection

@section('content')
<div class="slidershow middle">
    <div class="slides">
        <input type="radio" name="r" id="r1" checked>
        <input type="radio" name="r" id="r2">
        <input type="radio" name="r" id="r3">
        <input type="radio" name="r" id="r4">
        <input type="radio" name="r" id="r5">
        <input type="radio" name="r" id="r6">
        <div class="slide s1">
            
        </div>
        <div class="slide">
            
        </div>
        <div class="slide">
            
        </div>
        <div class="slide">
            
        </div>
        <div class="slide">
            
        </div>
        <div class="slide">
            
        </div>
    </div>
</div>
```

```

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

<div class="navigation">
    <label for="r1" class="bar"></label>
    <label for="r2" class="bar"></label>
    <label for="r3" class="bar"></label>
    <label for="r4" class="bar"></label>
    <label for="r5" class="bar"></label>
    <label for="r6" class="bar"></label>
</div>
@endsection

```

Dalam halaman home ini menggunakan file css kd.css. berikut **source codenya** :

```

*{
    margin: 5px;
    padding: 0;
    color: rgb(155, 0, 0);
    box-sizing: border-box;
    font-family: 'Arial Narrow Bold', sans-serif;
    text-decoration: none;
}
body{
    background: grey;
}
nav{
    background: black;
}
nav::after{
    content: '';
    clear: both;
    display: table;
}
nav .logo{
    float: left;
    font-style: italic;
    font-weight: bold;
    font-size: 40px;
    font-weight: 600px;
    line-height: 70px;
    padding-left: 40px;
}
nav ul{

```

```
float: right;
list-style: none;
margin-right: 50px;
position: relative;
}
nav ul li{
float: left;
display: block;
margin: 10px 0px;
}
nav ul li a{
color: rgb(190, 162, 162);
text-decoration: none;
line-height: 50px;
font-weight: bolder;
font-size: 12px;
padding: 8px 10px;
}
nav ul li a:hover{
color: whitesmoke;
background: rgba(238, 4, 4, 0.658);
border-radius: 5px;
box-shadow: 0 0 15px rgb(228, 228, 245),
            0 0 15px rgb(219, 230, 255);
}
nav ul ul{
position: absolute;
background: rgba(255, 4, 4, 0.658);
border-top: 3px solid rgb(163, 159, 159);
opacity: 0;
visibility: hidden;
transition: bottom .3s;
}
nav ul li:hover > ul{
opacity: 1;
visibility: visible;
}
nav ul ul li{
position: relative;
margin: 0;
width: 130px;
float: none;
display: list-item;
border-bottom: 1px solid rgba(0,0,0,0.3);
}
```

```
nav ul ul li a{
    line-height: 50px;
}
.greeting h1{
    margin-top: 15px;
    font-size: 50px;
    font-family: 'Times New Roman';
    font-weight: bolder;
    border-top: 4px solid black;
    border-bottom: 4px solid black;
    color: whitesmoke;
}
.slideshow{
    position: relative;
    width: 1300px;
    height: 450px;
    overflow: hidden;
    margin-top: 35px;
}
.middle{
    position: fixed;
    top: 57%;
    left: 45%;
    transform: translate(-50%,-50%);
}
.navigation{
    position: absolute;
    bottom: 10px;
    left: 50%;
    transform: translate(-50%);
    display: flex;
}
.bar{
    width: 50px;
    height: 10px;
    border: 2px solid rgb(132, 136, 136);
    margin: 6px;
    cursor: pointer;
}
.bar:hover{
    background: rgba(19, 255, 224, 0.644);
}
input[name="r"]{
    visibility: hidden;
}
```

```

.slides{
    width: 610%;
    height: 100%;
    display: flex;
}
.slide{
    width: 20%;
    height: 100%;
    justify-content: center;
}
.slide img{
    width: 1100px;
    height: 100%;
    justify-content: center;
}
#r1:checked ~ .s1{
    margin: -0%;
}
#r2:checked ~ .s1{
    margin: -10%;
}
#r3:checked ~ .s1{
    margin: -20%;
}
#r4:checked ~ .s1{
    margin: -30%;
}
#r5:checked ~ .s1{
    margin: -40%;
}
#r6:checked ~ .s1{
    margin: -50%;
}

```

Selanjutnya kita perlu membuat controller untuk class kdrama.blade.php. kita ketikkan perintah “*php artisan make:controller KdHomeController*” pada terminal. Setelah itu kita membuat function untuk menghubungkan route dengan class kdrama. Berikut **source codenya** :

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class KdHomeController extends Controller

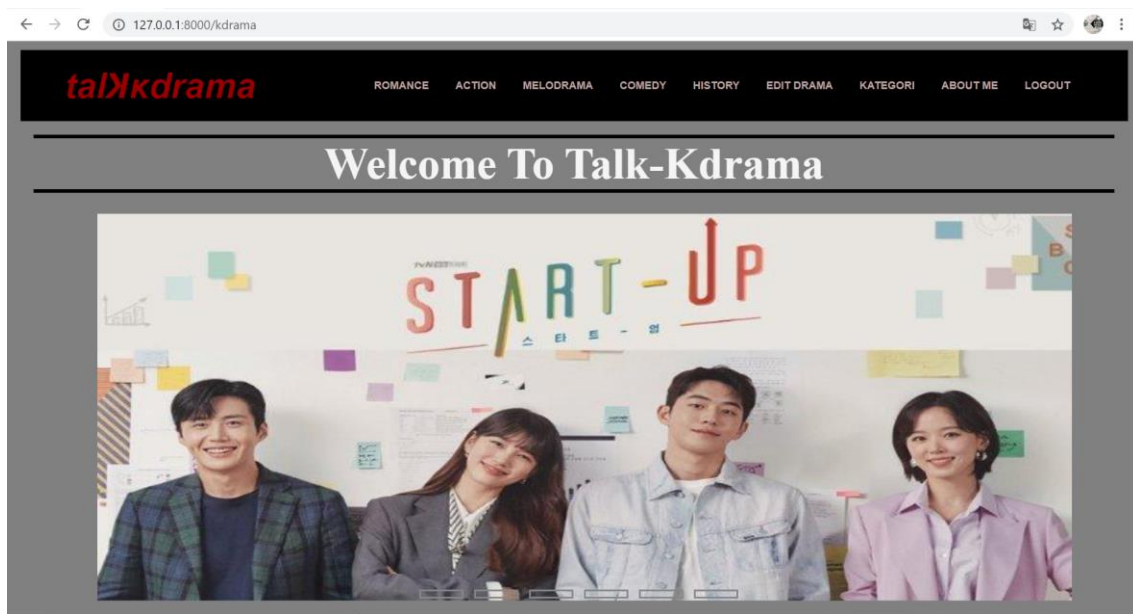
```

```
{
    public function Homedrama()
    {
        return view('ProjectDrama.kdrama');
    }
}
```

Setelah itu kita membuat routenya.

```
Route::get('/kdrama', 'KdHomeController@Homedrama');
```

Berikut adalah tampilan Home dari web talkkdrama



G. MEMBUAT CRUD

Dalam web ini saya membuat dua crud yaitu crud untuk data drama dan untuk kategori. Karena table drama dan kategori akan berelasi, sebab data drama ditampilkan sesuai kategorinya. Sebenarnya untuk crud editdrama dan kategori memiliki langkah yang sama.

Pertama kita harus membuat table editdata dan kategori menggunakan migrations. Kita bisa ketikkan perintah pada terminal => ***php artisan make:migrations create_nama_table***. Disini saya membuat table dengan nama editdata dan kategori. Setelah table berhasil dibuat kita bisa menyetting kolomnya.

Source code table editdata

```
<?php

use Illuminate\Database\Migrations\Migration;
```



```

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

class CreateEditdramaTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('editdrama', function (Blueprint $table) {
            $table->bigIncrements('id_drama');
            $table->string('judul');
            $table->string('episode');
            $table->string('durasi');
            $table->string('cast');
            $table->longText('sinopsis');
            $table->timestamps();
        });
    }

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

```

Source code table kategori

```

<?php

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

class CreateKategoriTable extends Migration
{
    /**

```

```

* Run the migrations.
*
* @return void
*/
public function up()
{
    Schema::create('kategori', function (Blueprint $table) {
        $table->bigIncrements('id_kategori');
        $table->string('namakategori');
        $table->timestamps();
    });
}

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

```

Setelah itu kita bisa melakukan migrate. Pada terminal kita ketikkan perintah “*php artisan migrate*”. Maka kolom akan secara otomatis bertambah di database.

Server: 127.0.0.1 » Basis data: projectpwl » Tabel: editdrama									
Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor Hak Akses Operasi Pelacakan									
Struktur tabel Tampilan hubungan									
#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1 id_drama	bigint(20)		UNSIGNED	Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
<input type="checkbox"/>	2 judul	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	3 id_kategori	bigint(20)			Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	4 episode	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	5 durasi	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	6 cast	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	7 sinopsis	longtext	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	8 created_at	timestamp			Ya	NULL			Ubah Hapus Lainnya
<input type="checkbox"/>	9 updated_at	timestamp			Ya	NULL			Ubah Hapus Lainnya

Server: 127.0.0.1 » Basis data: projectpwl » Tabel: kategori									
Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor Hak Akses Operasi Pelacakan									
Struktur tabel Tampilan hubungan									
#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1 id_kategori	bigint(20)		UNSIGNED	Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
<input type="checkbox"/>	2 namakategori	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
<input type="checkbox"/>	3 created_at	timestamp			Ya	NULL			Ubah Hapus Lainnya
<input type="checkbox"/>	4 updated_at	timestamp			Ya	NULL			Ubah Hapus Lainnya

Sekanjutnya kita harus membuat model supaya data yang telah ditambahkan di database dapat ditampilkan ke laravel. Untuk membuat table ketikkan perintah di terminal *php artisan make:Model namaModel*. Setelah itu kita panggil table yang di database dengan model yang sudah dibuat.

Setelah itu kita buat route untuk editdrama dan kategori. Dibawah ini adalah route lengkap dari tampil data sampai delete. Jadi route dibuat dulu agar tampilan atau source code dapat langsung dijalankan.

Route editdrama :

```
Route::get('/crudDrama', 'editdramaController@editdrama')->name('editdrama');

Route::get('/crudDrama/create', 'editdramaController@create')->name('editdrama.create');

Route::PUT('/crudDrama/store', 'editdramaController@store')->name('editdrama.store');

Route::get('/crudDrama/edit/{id}', 'editdramaController@edit')->name('editdrama.edit');

Route::PUT('/crudDrama/update/{id}', 'editdramaController@update')->name('editdrama.update');

Route::get('/crudDrama/delete/{id}', 'editdramaController@destroy')->name('editdrama.delete');
```

Route Kategori :

```
Route::get('/kategori', 'KategoriController@index')->name('kategori');

Route::get('/kategori/create', 'KategoriController@create')->name('kategori.create');

Route::PUT('/kategori/store', 'KategoriController@store')->name('kategori.store');

Route::get('/kategori/edit/{id}', 'KategoriController@edit')->name('kategori.edit');

Route::PUT('/kategori/update/{id}', 'KategoriController@update')->name('kategori.update');

Route::get('/kategori/delete/{id}', 'KategoriController@destroy')->name('kategori.delete');
```

Model Editdrama:

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Editdrama extends Model
{
    protected $table='editdrama';
}
```

Model Kategori :

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Kategori extends Model
{
    protected $table='kategori';
}
```

Karena table Editdrama dan kategori harus saling berelasi, kita buat dulu relasinya. Jadi kita membuat relasinya itu di Model yang sudah dibuat. Pertama kita harus menambahkan kolom id_kategori pada table edit drama. Penambahan kolom dapat dilakukan di database langsung. Jadi relasinya adalah 1 kategori dapat memiliki banyak drama. Berikut relasinya :

Model Editdrama :

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Editdrama extends Model
{
    protected $table='editdrama';
    protected $primaryKey = "id_drama";
    protected $guarded=[];
    protected $fillable = [
```

```

        'id_drama', 'judul', 'id_kategori', 'episode', 'durasi', 'cast', 'sinopsis'
    ];

    public function kategori()
    {
        return $this->belongsTo(Kategori::class, 'id_kategori');
    }
}

```

Model Kategori :

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Kategori extends Model
{
    protected $table='kategori';
    protected $primaryKey = "id_kategori";
    protected $guarded=[];
    protected $fillable=[
        'id_kategori','namakategori'];

    public function editdrama()
    {
        return $this->hasMany(Editdrama::class, 'id_kategori');
    }
}

```

Kemudian kita panggil model ke Controller. Untuk memanggilnya jangan lupa gunakan perintah *use*.

Controller Editdrama dengan function untuk menampilkan data dari database

```

use App>Editdrama;

class editdramaController extends Controller
{
    public function editDrama()
    {
        $edit_drama=Editdrama::with('kategori')->latest()->paginate();
        return view('ProjectDrama.editdrama', compact('edit_drama'));
    }
}

```

Controller kategori dengan function untuk menampilkan data dari database

```
use App\Kategori;
class KategoriController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $list_kategori=Kategori::all();
        return view('ProjectDrama.kategori',compact('list_kategori'));
    }
}
```

Dalam controller menunjuk class editdrama dan kategori. Maka kita harus membuat class editdrama dan kategori. Setelah menampilkan data, selanjutnya adalah insert data atau menambah data. Disini kita harus menambahkan button tambah data. Caranya adalah dengan menuliskan perintah untuk tambah data drama `<p>Tambah Data</p>` dan untuk tambah kategori adalah `<p>Tambah Kategori</p>`. Jadi ketika nanti kita klik tombol tambah data atau tambah kategori, maka akan mengarah ke class create data atau kategori. Maka kita harus buat views baru untuk create data dan kategori.

Source Code class editdrama

```
@extends('layouts.Masterkd')
@section('title', 'Edit Data Kdrama')
<link rel="stylesheet" type="text/css" href="{{asset('/css/edit.css')}}">

@section('greeting')
<div>
    <center>
        <h1 class="display-4">Welcome To Edit Data Kdrama</h1>
    </center>
</div>
@endsection

@section('content')
    <div class="edit">
        <div class="card-header">Data Kdrama</div>
        <p><a href="{{route('editdrama.create')}}" class="btn btn-tambah">Tambah Data</a></p>
        <div class="data">
```

```

        <table>
            <tr>
                <th>No</th>
                <th>Judul</th>
                <th>Kategori</th>
                <th>Episode</th>
                <th>Durasi</th>
                <th>Cast</th>
                <th>Sinopsis</th>
                <th>Action</th>
            </tr>
            @php $no=0; @endphp
            @foreach($edit_drama as $item)
                @php $no++; @endphp
                <tr>
                    <td>{{$no}}</td>
                    <td>{{$item['judul']}}</td>
                    <td>{{$item->kategori-> namakategori}}</td>
                    <td>{{$item['episode']}}</td>
                    <td>{{$item['durasi']}}</td>
                    <td>{{$item['cast']}}</td>
                    <td>{{$item['sinopsis']}}</td>
                    <td><a href="{{route('editdrama.edit',$item['id_drama'
                    ])}}" class="btn btn_edit">Edit</a><a href="{{route('editdrama.delete',$item['id_d
                    rama'])}}" onclick="return confirm('Yakin Menghapus Data Drama {{$item['judul']}}
                    ?')" class="btn btn_del">Delete</a></td>
                </tr>
            @endforeach
        </table>
    </div>
</div>
@endsection

```

Source Code class kategori

```

@extends('layouts.Masterkd')
@section('title', 'Edit Data Kdrama')
<link rel="stylesheet" type="text/css" href="{{asset('/css/ktg.css')}}">

@section('greeting')
<div>
    <center>
        <h1 class="display-4">Welcome To Kategori Kdrama</h1>
    </center>
</div>

```

```

@endsection

@section('content')
    <div class="edit">
        <div class="card-header">Data Kategori Kdrama</div>
        <p><a href="{{route('kategori.create')}}" class="btn btn-
tambah">Tambah Kategori</a></p>
        <div class="data">
            <table>
                <tr>
                    <th>No</th>
                    <th>Nama Kategori</th>
                    <th>Action</th>
                </tr>
                @php $no=0; @endphp
                @foreach($list_kategori as $item)
                @php $no++; @endphp
                <tr>
                    <td>{{$no}}</td>
                    <td>{{$item['namakategori']}}</td>
                    <td><a href="{{route('kategori.edit',$item['id_kategor
i'])}}" class="btn btn_edit">Edit</a><a href="{{route('kategori.delete',$item['id_
kategori'])}}" onclick="return confirm('Yakin Menghapus Kategori {{$item['namakate
gori']}} ?)" class="btn btn_del">Delete</a></td>
                </tr>
                @endforeach
            </table>
        </div>
    </div>
@endsection

```

Pada form create data saat tombol save diklik maka data harus tersimpan untuk itu kita menggunakan `"action="{{route('editdrama.store')}}"` pada create datadrama dan untuk kategori adalah `"action="{{route('kategori.store')}}"`. Jadi ketika nanti kita klik tombol save, data yang baru ditambahkan dapat secara otomatis tampil baik di database ataupun laravel.

Source Code Create data Drama :

```

@extends('layouts.Masterkd')
@section('title', 'Edit Talk-Kdrama')
<link rel="stylesheet" type="text/css" href="{{asset('/css/edit.css')}}">

@section('content')
    <div class="edit">
        <div class="card-header">Tambah Data Kdrama</div>

```



```

<div class="card-body">
  <form method="POST" action="{route('editdrama.store')}}">
    @csrf
    @method('PUT')
    <div class="form-group row">
      <label class="col-md-4 text-md-right">Judul :</label>
      <div class="col-md-6">
        <input type="text" name="judul" class="form-control">
      </div>
    </div>

    <div class="form-group row">
      <label class="col-md-4 text-md-right">Kategori :</label>
      <div class="col-md-6">
        <select class="kategori" name="id_kategori" id="id_kategor
i">
          @foreach($ktg as $item)
            <option value="{{ $item->id_kategori }}">{{ $item-
>namakategori}}</option>
          @endforeach
        </select>
      </div>
    </div>

    <div class="form-group row">
      <label class="col-md-4 text-md-right">Episode :</label>
      <div class="col-md-6">
        <input type="text" name="episode" class="form-control">
      </div>
    </div>

    <div class="form-group row">
      <label class="col-md-4 text-md-right">Durasi :</label>
      <div class="col-md-6">
        <input type="text" name="durasi" class="form-control">
      </div>
    </div>

    <div class="form-group row">
      <label class="col-md-4 text-md-right">Cast :</label>
      <div class="col-md-6">
        <input type="text" name="cast" class="form-cast">
      </div>
    </div>
  </div>

```

```

        <div class="form-group row">
            <label class="col-md-4 text-md-right">Sinopsis :</label>
            <div class="col-md-6">
                <input type="text" name="sinopsis" class="form-sinopsis">
            </div>
        </div>

        <div class="form-group row">
            <label class="col-md-4 text-md-right"></label>
            <div class="col-md-6">
                <button class="btn btn-save" type="submit">Save</button>
            </div>
        </div>
    </form>
</div>
@endsection

```

Source Code Create kategori :

```

@extends('layouts.Masterkd')
@section('title', 'Edit Data Kdrama')
<link rel="stylesheet" type="text/css" href="{{asset('/css/ktg.css')}}">

@section('greeting')
<div>
    <center>
        <h1 class="display-4">Welcome To Kategori Kdrama</h1>
    </center>
</div>
@endsection

@section('content')
    <div class="edit">
        <div class="card-header">Tambah Data Kategori Kdrama</div>
        <div class="card-body">
            <form method="POST" action="{{route('kategori.store')}}">
                @csrf
                @method('PUT')
                <div class="form-group row">
                    <label class="col-md-4 text-md-right">Nama Kategori :</label>
                    <div class="col-md-6">
                        <input type="text" name="namakategori" class="form-
control">
                    </div>
                </div>
            </form>
        </div>
    </div>

```

```

        </div>

        <div class="form-group row">
            <label class="col-md-4 text-md-right"></label>
            <div class="col-md-6">
                <button class="btn btn-save" type="submit">Save</button>
            </div>
        </div>
    </form>
</div>
</div>
@endsection

```

Selanjutnya adalah menambah function create dan store di Controller editdrama dan kategori.

Editdrama Controller :

```

public function create()
{
    $ktg= Kategori::all();
    return view('ProjectDrama.kd_create', compact('ktg'));
}

public function store(Request $request)
{
    Editdrama::create($request->all());
    return redirect()->route('editdrama');
}

```

Kategori Controller :

```

public function create()
{
    return view('ProjectDrama.ktg_create');
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    Kategori::create($request->all());
    return redirect()->route('kategori');
}

```

```
}
```

Fitur berikutnya adalah edit. Fitur edit sebenarnya sama seperti create yaitu kita harus membuat form edit nya terlebih dahulu. Perbedaan nya adalah pada edit terdapat penambahan value. Jadi ketika kita klik edit form tidak kosong melainkan sudah terdapat data sewaktu kita create jadi kita hanya perlu memilih data mana yang akan diedit tanpa harus mengisi ulang data dari awal.

Pada form edit data saat tombol save diklik maka data yang harus tersimpan adalah data setelah diupdate atau diedit untuk itu kita menggunakan `"action="{{route('editdrama.update')}}"` pada create datadrama dan untuk kategori adalah `"action="{{route('kategori.update')}}"`. Jadi ketika nanti kita klik tombol save, data yang baru diupdate dapat secara otomatis tampil baik di database ataupun laravel.

Source Code edit editdrama:

```
@extends('layouts.Masterkd')
@section('title', 'Edit Talk-Kdrama')
<link rel="stylesheet" type="text/css" href="{{asset('/css/edit.css')}}">

@section('content')
    <div class="edit">
        <div class="card-header">Edit Data Kdrama</div>

        <div class="card-body">
            <form method="POST" action="{{route('editdrama.update',$data['id_drama
'])}}">
                @csrf
                @method('PUT')
                <div class="form-group row">
                    <label class="col-md-4 text-md-right">Judul :</label>
                    <div class="col-md-6">
                        <input type="text" name="judul" value="{{ $data['judul'] }}"
class="form-control">
                    </div>
                </div>

                <div class="form-group row">
                    <label class="col-md-4 text-md-right">Kategori :</label>
                    <div class="col-md-6">
                        <select class="kategori" name="id_kategori" id="id_kategor
i">
                            <option disabled value>Pilihan Kategori</option>
                            <option class="pilihan" value="{{ $data-
>id_kategori}}">{{ $data->kategori->namakategori}}</option>
                            @foreach($ktg as $item)
```

```

                                <option value="{{ $item->id_kategori }}">{{ $item-
>namakategori}}</option>
                                @endforeach
                                </select>
                                </div>
                            </div>

                            <div class="form-group row">
                                <label class="col-md-4 text-md-right">Episode :</label>
                                <div class="col-md-6">
                                    <input type="text" name="episode" value="{{ $data['episode']
}}}" class="form-control">
                                </div>
                            </div>

                            <div class="form-group row">
                                <label class="col-md-4 text-md-right">Durasi :</label>
                                <div class="col-md-6">
                                    <input type="text" name="durasi" value="{{ $data['durasi']
}}}" class="form-control">
                                </div>
                            </div>

                            <div class="form-group row">
                                <label class="col-md-4 text-md-right">Cast :</label>
                                <div class="col-md-6">
                                    <input type="text" name="cast" value="{{ $data['cast']
}}}" class="form-cast">
                                </div>
                            </div>

                            <div class="form-group row">
                                <label class="col-md-4 text-md-right">Sinopsis :</label>
                                <div class="col-md-6">
                                    <input type="text" name="sinopsis" value="{{ $data['sinopsi
s'] }}" class="form-sinopsis">
                                </div>
                            </div>

                            <div class="form-group row">
                                <label class="col-md-4 text-md-right"></label>
                                <div class="col-md-6">
                                    <button class="btn btn-save" type="submit">Save</button>
                                </div>
                            </div>

```

```

        </form>
    </div>
@endsection

```

Source Code edit kategori :

```

@extends('layouts.Masterkd')
@section('title', 'Edit Data Kdrama')
<link rel="stylesheet" type="text/css" href="{{asset('/css/ktg.css')}}">

@section('greeting')
<div>
    <center>
        <h1 class="display-4">Welcome To Kategori Kdrama</h1>
    </center>
</div>
@endsection

@section('content')
    <div class="edit">
        <div class="card-header">Edit Data Kategori Kdrama</div>
        <div class="card-body">
            <form method="POST" action="{{route('kategori.update',$data['id_kategori'])}}">
                @csrf
                @method('PUT')
                <div class="form-group row">
                    <label class="col-md-4 text-md-right">Nama Kategori :</label>
                    <div class="col-md-6">
                        <input type="text" name="namakategori" value="{{ $data['namakategori'] }}" class="form-control">
                    </div>
                </div>

                <div class="form-group row">
                    <label class="col-md-4 text-md-right"></label>
                    <div class="col-md-6">
                        <button class="btn btn-save" type="submit">Save</button>
                    </div>
                </div>
            </form>
        </div>
    </div>
@endsection

```

CSS CRUD Editdrama :

```
*{
  margin: 5px;
  padding: 0;
  color: rgb(155, 0, 0);
  box-sizing: border-box;
  font-family: 'Arial Narrow Bold', sans-serif;
  text-decoration: none;
}
body{
  background: grey;
}
nav{
  background: black;
}
nav::after{
  content: '';
  clear: both;
  display: table;
}
nav .logo{
  float: left;
  font-style: italic;
  font-weight: bold;
  font-size: 40px;
  font-weight: 600px;
  line-height: 70px;
  padding-left: 40px;
}
nav ul{
  float: right;
  list-style: none;
  margin-right: 50px;
  position: relative;
}
nav ul li{
  float: left;
  display: block;
  margin: 10px 0px;
}
nav ul li a{
  color: rgb(190, 162, 162);
  text-decoration: none;
  line-height: 50px;
  font-weight: bolder;
  font-size: 12px;
}
```

```

padding: 8px 10px;
}
nav ul li a:hover{
color: whitesmoke;
background: rgba(155, 0, 0, 0.658);
border-radius: 5px;
box-shadow: 0 0 15px rgb(228, 228, 245),
            0 0 15px rgb(219, 230, 255);
}
nav ul ul{
position: absolute;
background: rgba(155, 0, 0, 0.658);
border-top: 3px solid rgb(163, 159, 159);
opacity: 0;
visibility: hidden;
transition: bottom .3s;
}
nav ul li:hover > ul{
opacity: 1;
visibility: visible;
}
nav ul ul li{
position: relative;
margin: 0;
width: 130px;
float: none;
display: list-item;
border-bottom: 1px solid rgba(0,0,0,0.3);
}
nav ul ul li a{
line-height: 50px;
}
.greeting h1{
margin-top: 15px;
font-size: 50px;
font-family: 'Times New Roman';
font-weight: bolder;
border-top: 4px solid black;
border-bottom: 4px solid black;
color: whitesmoke;
}
.card-header{
font-weight: bold;
font-size: 40px;
color: rgb(0, 0, 0);

```



```

    text-align: center;
}
.data{
    border: 3px solid #000000;
    margin-top: 20px;
    background-color: rgba(255, 255, 255, 0.637);
}
.data tr th{
    color: black;
    border: 3px solid black;
    padding: 10px 8px;
}
.data tr td{
    padding: 25px 8px;
    color: black;
    text-align: justify;
    border: 3px solid black;
}
.btn-tambah {
    position: relative;
    background-color: black;
    color: white;
    font-size: 20px;
    font-weight: 30px;
    padding: 10px 20px;
    text-decoration: none;
}
form{
    width: 800px;
    height: 710px;
    position: relative;
    margin-top: 10px;
    margin-left: 210px;
    background: rgba(29, 25, 25, 0.219);
    padding: 5px;
    border-radius: 30px;
}
.text-md-right{
    position: relative;
    font-size: 15px;
    font-weight: bold;
    font-family: cursive;
    color: rgb(0, 0, 0);
    margin-left: 45px;
}

```

```
.form-control{
    height: 25px;
    width: 700px;
    margin-left: 40px;
    border: 1.5px solid rgb(0, 0, 0);
    color: black;
    font-family: monospace;
    font-size: 18px;
}
.kategori{
    height: 30px;
    width: 200px;
    margin-left: 40px;
    border: 1.5px solid rgb(0, 0, 0);
    color: black;
    font-family: monospace;
    font-size: 18px;
}
.pilihan{
    color: black;
    font-style: italic;
    font-weight: bold;
}
.form-cast{
    width: 700px;
    height: 80px;
    margin-left: 40px;
    border: 1.5px solid rgb(0, 0, 0);
    color: black;
    font-family: monospace;
    font-size: 18px;
}
.form-sinopsis{
    width: 700px;
    height: 200px;
    margin-left: 40px;
    border: 1.5px solid rgb(0, 0, 0);
    color: black;
    font-family: monospace;
    font-size: 18px;
}
.btn-save{
    margin-left: 350px;
    position: relative;
    background-color: rgba(148, 26, 26, 0.884);
```

```

    color: rgb(0, 0, 0);
    font-size: 20px;
    font-weight: bold;
    padding: 10px 30px;
    border-radius: 5px;
    text-decoration: none;
}
.btn_edit{
    position: relative;
    background-color: rgb(255, 0, 0);
    color: rgb(0, 0, 0);
    font-size: 20px;
    font-weight: bold;
    padding: 10px 30px;
    border-radius: 5px;
    text-decoration: none;
    margin-bottom: 40px;
}
.btn_del{
    position: relative;
    background-color: rgb(0, 17, 255);
    color: rgb(0, 0, 0);
    font-size: 20px;
    font-weight: bold;
    padding: 10px 30px;
    border-radius: 5px;
    text-decoration: none;
    margin-top: 40px;
}

```

CSS CRUD Kategori :

```

*{
    margin: 5px;
    padding: 0;
    color: rgb(155, 0, 0);
    box-sizing: border-box;
    font-family: 'Arial Narrow Bold', sans-serif;
    text-decoration: none;
}
body{
    background: grey;
}
nav{
    background: black;

```

```
}
nav::after{
  content: '';
  clear: both;
  display: table;
}
nav .logo{
  float: left;
  font-style: italic;
  font-weight: bold;
  font-size: 40px;
  font-weight: 600px;
  line-height: 70px;
  padding-left: 40px;
}
nav ul{
  float: right;
  list-style: none;
  margin-right: 50px;
  position: relative;
}
nav ul li{
  float: left;
  display: block;
  margin: 10px 0px;
}
nav ul li a{
  color: rgb(190, 162, 162);
  text-decoration: none;
  line-height: 50px;
  font-weight: bolder;
  font-size: 12px;
  padding: 8px 10px;
}
nav ul li a:hover{
  color: whitesmoke;
  background: rgba(155, 0, 0, 0.658);
  border-radius: 5px;
  box-shadow: 0 0 15px rgb(228, 228, 245),
             0 0 15px rgb(219, 230, 255);
}
nav ul ul{
  position: absolute;
  background: rgba(155, 0, 0, 0.658);
  border-top: 3px solid rgb(163, 159, 159);
```

```

    opacity: 0;
    visibility: hidden;
    transition: bottom .3s;
}
nav ul li:hover > ul{
    opacity: 1;
    visibility: visible;
}
nav ul ul li{
    position: relative;
    margin: 0;
    width: 130px;
    float: none;
    display: list-item;
    border-bottom: 1px solid rgba(0,0,0,0.3);
}
nav ul ul li a{
    line-height: 50px;
}
.greeting h1{
    margin-top: 15px;
    font-size: 50px;
    font-family: 'Times New Roman';
    font-weight: bolder;
    border-top: 4px solid black;
    border-bottom: 4px solid black;
    color: whitesmoke;
}
.card-header{
    font-weight: bold;
    font-size: 40px;
    color: rgb(0, 0, 0);
    text-align: center;
}
.data{
    border: 3px solid #000000;
    margin-top: 20px;
    width: 900px;
    margin-left: 14%;
    background-color: rgba(255, 255, 255, 0.637);
}
.data tr th{
    color: black;
    border: 3px solid black;
    padding: 5px 80px;
}

```

```
    font-family: cursive;
    font-weight: bolder;
    font-size: 20px;
}
.data tr td{
    padding: 15px 80px;
    color: black;
    text-align: justify;
    border: 3px solid black;
    font-family: 'times new roman';
    font-size: 20px;
}
.btn-tambah {
    position: relative;
    background-color: black;
    color: white;
    font-size: 20px;
    font-weight: 30px;
    padding: 10px 20px;
    text-decoration: none;
    margin-left: 13.7%;
}
form{
    width: 400px;
    position: relative;
    margin-top: 70px;
    margin-left: 35%;
    background: rgba(255, 255, 255, 0.904);
    padding: 5px;
    border-radius: 30px;
}
.text-md-right{
    position: relative;
    font-size: 15px;
    font-weight: bold;
    font-family: cursive;
    color: rgb(0, 0, 0);
    margin-left: 45px;
}
.form-control{
    height: 25px;
    width: 300px;
    margin-left: 40px;
    border: 1.5px solid rgb(0, 0, 0);
}
```

```

.btn-save{
  margin-left: 35%;
  position: relative;
  background-color: rgba(148, 26, 26, 0.884);
  color: rgb(0, 0, 0);
  font-size: 20px;
  font-weight: bold;
  padding: 10px 30px;
  border-radius: 5px;
  text-decoration: none;
}

.btn_edit{
  position: relative;
  background-color: rgb(255, 0, 0);
  color: rgb(0, 0, 0);
  font-size: 20px;
  font-weight: bold;
  padding: 10px 30px;
  border-radius: 5px;
  text-decoration: none;
  margin-bottom: 40px;
}

.btn_del{
  position: relative;
  background-color: rgb(0, 17, 255);
  color: rgb(0, 0, 0);
  font-size: 20px;
  font-weight: bold;
  padding: 10px 30px;
  border-radius: 5px;
  text-decoration: none;
  margin-top: 40px;
}

```

Selanjutnya adalah menambah function edit dan store di Controller editdrama dan kategori.

Editdrama Controller :

```

public function edit($id)
{
    $ktg= Kategori::all();
    $data=EditDrama::with('kategori')->findorfail($id);
    return view ('ProjectDrama.kd_edit', compact('data','ktg'));
}

public function update(Request $request, $id)

```

```

{
    $data=EditDrama::findorfail($id);
    $data->update($request->all());
    return redirect(route('editdrama'))->with('toasts_success','Data Berhasil Diupdate');
}

```

Kategori Controller :

```

public function edit($id)
{
    $data=Kategori::where('id_kategori', $id)->firstorfail();
    return view ('ProjectDrama.ktg_edit', compact('data'));
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $id)
{
    $data=Kategori::where('id_kategori', $id)
        ->update([
            'namakategori' => $request['namakategori']
        ]);
    return redirect()->route('kategori');
}

```

Fitur terakhir dari CRUD adalah delete. Pada tombol delete kita menggunakan `"action="{{route('editdrama.hapus')}}"` seperti terlihat pada class editdrama . Jadi ketika nanti kita klik tombol delete, data secara otomatis hilang dari database ataupun laravel. Untuk itu pada Controller edit drama dan kategori kita perlu menambahkan function destroy. Fungsi onclick pada tombol delete adalah untuk menyakinkan apakah data benar banar ingin dihapus.

Editdrama Controller :

```

public function destroy($id)
{
    $data=EditDrama::where('id_drama', $id)
        ->delete();
    return redirect()->route('editdrama');
}

```


Kategori Controller :

```
public function destroy($id)
{
    $data=Kategori::where('id_kategori', $id)
        ->delete();
    return redirect()->route('kategori');
}
```

Tampilan CRUD Editdrama

← → ↻ 127.0.0.1:8000/crudDrama ☆ ⋮

talKdrama ROMANCE ACTION MELODRAMA COMEDY HISTORY EDIT DRAMA KATEGORI ABOUT ME LOGOUT

Welcome To Edit Data Kdrama

Data Kdrama

Tambah Data

No	Judul	Kategori	Episode	Durasi	Cast	Sinopsis	Action
1	Start Up	Romance	16	± 1 jam 10	Bae Suzy as Seo Dal Mi, Nam Joo Hyuk as Nam Do San, Kim Seon Ho	Mengambil latar belakang di Sandbox yang sering disebut sebagai Silicon Valley-nya Korea dan bercerita tentang orang-orang yang bergelut dalam dunia startup. Seo Dal Mi (Bae Suzy) bermimpi untuk menjadi Steve Jobs. Ketika ia membutuhkan sejumlah uang untuk memulai perusahaan startup, Dal Mi memutuskan berhenti kuliah dan bekerja paruh waktu. Di tengah keterbatasan yang ia miliki, Dal Mi punya impian dan semangat yang besar. Di sisi lain ada Nam Do San, pendiri Samsan Tech yang dikenal jenius dalam bidang matematika. Sayangnya selama 2 tahun terakhir ia mengalami banyak kegagalan saat melakukan investasi. Do San ternyata adalah cinta pertama Dal Mi yang menurutnya sangat keren. Do San kemudian memutuskan untuk bekerja sama mendirikan startup baru bersama dengan Dal Mi. Sementara itu, Han Ji Pyung (Kim Seon Ho) mendirikan startup untuk	Edit Delete

← → ↻ 127.0.0.1:8000/crudDrama/create ⋮ ☆

talKdrama ROMANCE ACTION MELODRAMA COMEDY HISTORY EDIT DRAMA KATEGORI ABOUT ME LOGOUT

Tambah Data Kdrama

Judul :

Kategori :

Episode :

Durasi :

Cast :

Sinopsis :

← → ↻ 127.0.0.1:8000/crudDrama/edit/9

Edit Data Kdrama

Judul :
Start Up

Kategori :
Romance

Episode :
16

Durasi :
± 1 jam 10 menit

Cast :
Bae Suzy as Seo Dal Mi, Nam Joo Hyuk as Nam Do San, Kim Seon Ho as Han

Sinopsis :
Mengambil latar belakang di Sandbox yang sering disebut sebagai Silicon

← → ↻ 127.0.0.1:8000/crudDrama

Tambah Data

127.0.0.1:8000 says
Yakin Menghapus Data Drama Start Up ?

No	Judul	Kategori	Episode	Durasi	Action
1	Start Up	Romance	16	± 1 jam 10 menit	<div>Mengambil latar belakang di Sandbox yang sering disebut sebagai Silicon Valley-nya Korea dan bercerita tentang orang-orang yang bergelut dalam dunia startup. Seo Dal Mi (Bae Suzy) bermimpi untuk menjadi Steve Jobs. Ketika ia membutuhkan sejumlah uang untuk memulai perusahaan startup, Dal Mi memutuskan berhenti kuliah dan bekerja paruh waktu. Di tengah keterbatasan yang ia miliki, Dal Mi punya impian dan semangat yang besar. Di sisi lain ada Nam Do San, pendiri Samsan Tech yang dikenal jenius dalam bidang matematika. Sayangnya selama 2 tahun terakhir ia mengalami banyak kegagalan saat melakukan investasi. Do San ternyata adalah cinta pertama Dal Mi yang menurutnya sangat keren. Do San kemudian memutuskan untuk bekerja sama mendirikan startup baru bersama dengan Dal Mi. Sementara itu, Han Ji Pyung (Kim Seon Ho) mendirikan startup untuk melunasi hutang-hutangnya. Sebelumnya ia bekerja sebagai ketua tim di SH Venture Capital. Kecakapan dan keahliannya membuat Ji Pyung dijuluki sebagai "Gordon Ramsay dalam investasi". Meski ia terkenal dingin dan cuek, ternyata Ji Pyung punya sisi hangat untuk seseorang yang berarti di masa lalunya. Terakhir ada CEO Won In Jae (Kang Han Na) yang merupakan tipe impian semua orang: latar belakang pendidikan baik, cantik, dan kaya. Namun ia justru menganggap semua yang dimilikinya sebagai kelemahan dan ingin meraih kesuksesan dengan caranya sendiri, bukan karena keluarganya. Karena tidak ingin diuntungkan sebagai pion</div> <div>Edit Delete</div>

Tampilan CRUD Kategori

← → ↻ 127.0.0.1:8000/kategori

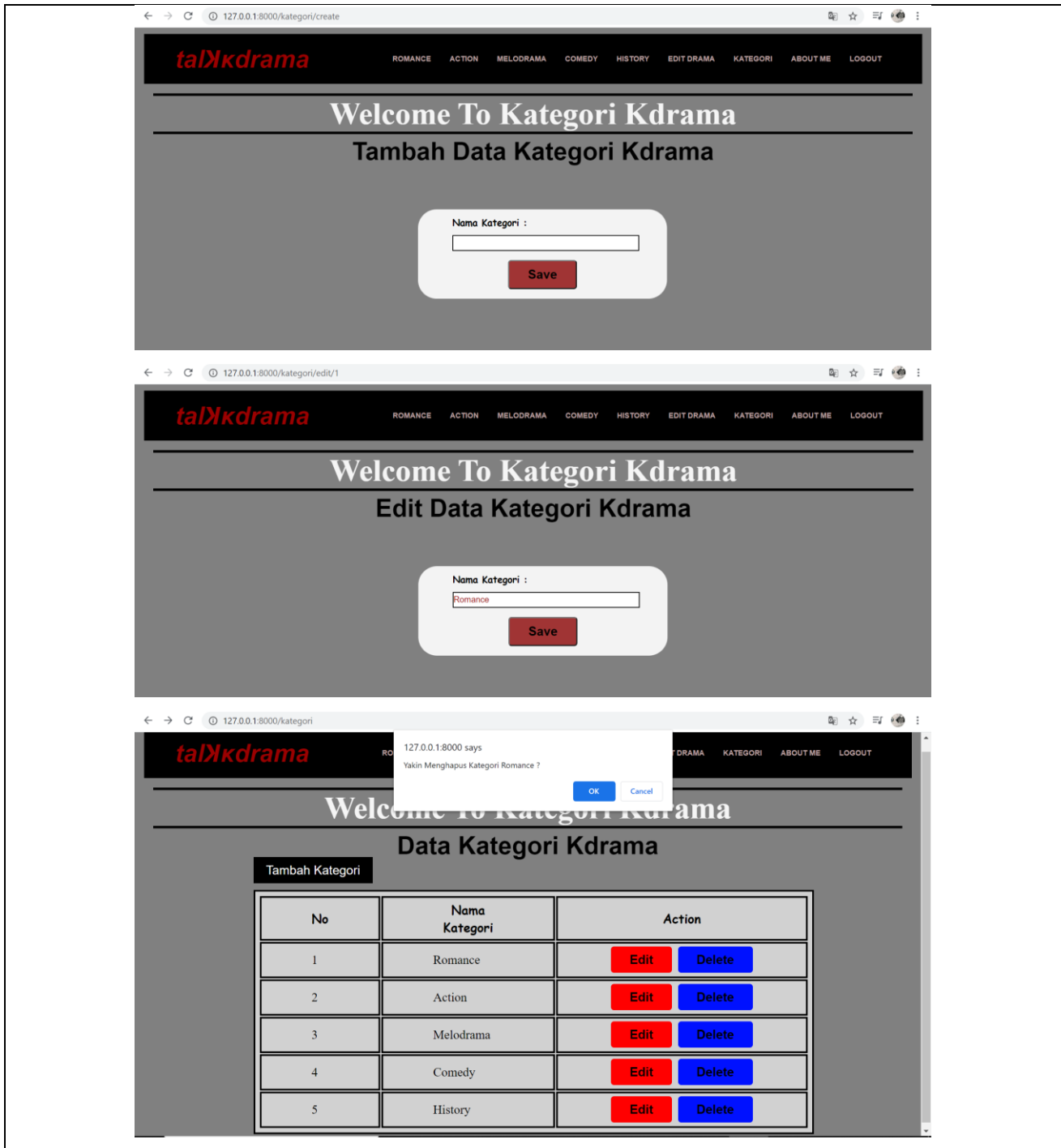
talDkdrama ROMANCE ACTION MELODRAMA COMEDY HISTORY EDIT DRAMA KATEGORI ABOUT ME LOGOUT

Welcome To Kategori Kdrama

Data Kategori Kdrama

Tambah Kategori

No	Nama Kategori	Action
1	Romance	Edit Delete
2	Action	Edit Delete
3	Melodrama	Edit Delete
4	Comedy	Edit Delete
5	History	Edit Delete



H. MENAMPILKAN DATA SESUAI KATEGORI

Pertama kita buat Routenya dulu agar lebih mudah dalam pemanggilan route di class tiap genre.

```
Route::get('/romance', 'GenreController@romance')->name('romance');
Route::get('/romance/{data}', 'GenreController@show')->name('romance.artikel');
```

```

Route::get('/action', 'GenreController@action')->name('action');

Route::get('/action/{data}', 'GenreController@showAc')->name('action.artikel');

Route::get('/melo', 'GenreController@melo')->name('melo');

Route::get('/melo/{data}', 'GenreController@showmelo')->name('melo.artikel');

Route::get('/Comedy', 'GenreController@comedy')->name('comedy');

Route::get('/Comedy/{data}', 'GenreController@showcom')->name('comedy.artikel');

Route::get('/history', 'GenreController@history')->name('history');

Route::get('/history/{data}', 'GenreController@showhis')->name('history.artikel');

```

Selanjutnya adalah menuliskan function tambahan pada model Editdrama agar data dapat dibedakan berdasar kategori.

```

public function getRouteKeyName()
{
    return 'id_kategori';
}

```

Selanjutnya adalah membuat GenreController untuk menampung function show dari setiap genre. Untuk membuat genre controller adalah dengan perintah ***php artisan make:controller GenreController***. Dalam genre controller kita juga mengimport model Editdrama dan Kategori seperti terlihat dibawah ini.

Source Code Genre Controller :

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App/Editdrama;
use App\Kategori;

class GenreController extends Controller
{
    public function romance(Kategori $id_kategori)
    {

```

```

    $editdrama = Editdrama::latest()->get();
    $kategori = $id_kategori;
    $kategori = 1;
    return view('ProjectDrama.Romance', compact('editdrama','kategori'));
}

public function show(Editdrama $editdrama)
{
    $editdrama = Editdrama::latest()->get();
    $ktg = Kategori::all();
    return view('ProjectDrama.Romance', compact('editdrama','ktg'));
}

public function action(Kategori $id_kategori)
{
    $editdrama = Editdrama::latest()->get();
    $kategori = $id_kategori;
    $kategori = 2;
    return view('ProjectDrama.Action', compact('editdrama','kategori'));
}

public function showAc(Editdrama $editdrama)
{
    $editdrama = Editdrama::latest()->get();
    $ktg = Kategori::all();
    return view('ProjectDrama.Action', compact('editdrama','ktg'));
}

public function melo(Kategori $id_kategori)
{
    $editdrama = Editdrama::latest()->get();
    $kategori = $id_kategori;
    $kategori = 3;
    return view('ProjectDrama.Melodrama', compact('editdrama','kategori'));
}

public function showmelo(Editdrama $editdrama)
{
    $editdrama = Editdrama::latest()->get();
    $ktg = Kategori::all();
    return view('ProjectDrama.Melodrama', compact('editdrama','ktg'));
}

public function comedy(Kategori $id_kategori)
{
    $editdrama = Editdrama::latest()->get();
    $kategori = $id_kategori;

```

```

    $kategori = 4;
    return view('ProjectDrama.Comedy', compact('editdrama','kategori'));
}
public function showcom(Editdrama $editdrama)
{
    $editdrama = Editdrama::latest()->get();
    $ktg = Kategori::all();
    return view('ProjectDrama.Comedy', compact('editdrama','ktg'));
}

public function history(Kategori $id_kategori)
{
    $editdrama = Editdrama::latest()->get();
    $kategori = $id_kategori;
    $kategori = 3;
    return view('ProjectDrama.History', compact('editdrama','kategori'));
}
public function showhis(Editdrama $editdrama)
{
    $editdrama = Editdrama::latest()->get();
    $ktg = Kategori::all();
    return view('ProjectDrama.History', compact('editdrama','ktg'));
}
}

```

Setelah controller berhasil dibuat maka kita buat class untuk tiap genrenya. Untuk memisahkan data perkategori saya menggunakan percabangan if. Jadi ketika data drama tidak memiliki id_kategori yang memenuhi syarat data tidak ditampilkan. Dibagian bawah synopsis drama juga saya beri kategori agar

Source Code Romance :

```

@extends('layouts.MasterKd')
@section('title', 'Romance')
<link rel="stylesheet" type="text/css" href="{{asset('/css/c.css')}}">

@section('greeting')
<div>
    <center>
        <h1 class="display-4">Welcome To Genre Romance</h1>
    </center>
</div>
@endsection
@section('content')
<div class="container">
    <div class="row">

```

```

@foreach($editdrama as $all)
<div class="col-md-8">
    <a href="{{route('romance.artikel',$all->id_kategori)}}">
        </a>
        @if($all->id_kategori=='1')
            <div class="Judul">
                <h2>{{ $all->judul }}</h2>
                
                <h3>Info Drama :
                    <p>Episode : {{ $all->episode }}</p>
                    <p>Durasi : {{ $all->durasi }}</p></h3>
                <h3>Cast:
                    <p>{{ $all->cast }}</p></h3>
                <h3>Sinopsis :
                    <p>{{ $all->sinopsis }}</p></h3>
            </div>
        </div>
        @endif
    @endforeach

    <!-- catagories -->
    <div class="ctg">
        <div class="section-title">
            <h2>Pilih Genre</h2>
        </div>
        <div class="category-widget">
            <ul>
                <li><a href="/romance" class="cat-1">Romance<span>->
                </span></a></li>
                <li><a href="/action" class="cat-2">Action<span>->
                </span></a></li>
                <li><a href="/melo" class="cat-4">MeloDrama<span>->
                </span></a></li>
                <li><a href="/Comedy" class="cat-3">Comedy<span>->
                </span></a></li>
                <li><a href="/history" class="cat-5">History<span>->
                </span></a></li>
            </ul>
        </div>
    </div>
</div>
</div>
</div>

```

```
@endsection
```

Source Code Action :

```
@extends('layouts.MasterKd')
@section('title', 'Action')
<link rel="stylesheet" type="text/css" href="{{asset('/css/c.css')}}">

@section('greeting')
<div>
    <center>
        <h1 class="display-4">Welcome To Genre Action</h1>
    </center>
</div>
@endsection
@section('content')
<div class="container">
    <div class="row">
        @foreach($editdrama as $all)
            <div class="col-md-8">
                <a href="{{route('action.artikel',$all->id_kategori)}}">
                    </a>
                @if($all->id_kategori=='2')
                    <div class="Judul">
                        <h2>{{ $all->judul }}</h2>
                        
                        <h3>Info Drama :
                        <p>Episode : {{ $all->episode }}</p>
                        <p>Durasi : {{ $all->durasi }}</p></h3>
                        <h3>Cast:
                        <p>{{ $all->cast }}</p></h3>
                        <h3>Sinopsis :
                        <p>{{ $all->sinopsis }}</p></h3>
                    </div>
                </div>
                @endif
            @endforeach

            <!-- catagories -->
            <div class="ctg">
                <div class="section-title">
                    <h2>Pilih Genre</h2>
                </div>
                <div class="category-widget">
```



```

        <ul>
            <li><a href="/romance" class="cat-1">Romance<span>-
></span></a></li>
            <li><a href="/action" class="cat-2">Action<span>-
></span></a></li>
            <li><a href="/melo" class="cat-4">MeloDrama<span>-
></span></a></li>
            <li><a href="/Comedy" class="cat-3">Comedy<span>-
></span></a></li>
            <li><a href="/history" class="cat-5">History<span>-
></span></a></li>
        </ul>
    </div>
</div>
</div>
</div>
</div>
@endsection

```

Source Code Melodrama :

```

@extends('layouts.MasterKd')
@section('title', 'Melodrama')
<link rel="stylesheet" type="text/css" href="{{asset('/css/c.css')}}">

@section('greeting')
<div>
    <center>
        <h1 class="display-4">Welcome To Genre Melodrama</h1>
    </center>
</div>
@endsection
@section('content')
    <div class="container">
        <div class="row">
            @foreach($editdrama as $all)
                <div class="col-md-8">
                    <a href="{{route('melo.artikel',$all->id_kategori)}}">
                        </a>
                    @if($all->id_kategori=='3')
                        <div class="Judul">
                            <h2>{{ $all->judul }}</h2>
                            

```

```
<h3>Info Drama :  
    <p>Episode : {{$all->episode}}</p>  
<p>Durasi : {{$all->durasi}}</p></h3>  
<h3>Cast:  
    <p>{{$all->cast}}</p></h3>  
    <h3>Sinopsis :  
        <p>{{$all->sinopsis}}</p></h3>  
    </div>  
</div>  
@endif  
@endforeach  
  
<!-- catagories -->  
<div class="ctg">  
    <div class="section-title">  
        <h2>Pilih Genre</h2>  
    </div>  
    <div class="category-widget">  
        <ul>  
            <li><a href="/romance" class="cat-1">Romance<span>-  
</span></a></li>  
            <li><a href="/action" class="cat-2">Action<span>-  
</span></a></li>  
            <li><a href="/melo" class="cat-4">MeloDrama<span>-  
</span></a></li>  
            <li><a href="/Comedy" class="cat-3">Comedy<span>-  
</span></a></li>  
            <li><a href="/history" class="cat-5">History<span>-  
</span></a></li>  
        </ul>  
    </div>  
</div>  
</div>  
</div>  
</div>  
@endsection
```

```
@extends('layouts.MasterKd')
@section('title', 'Comedy')
<link rel="stylesheet" type="text/css" href="{{asset('/css/c.css')}}">
@section('greeting')
```

```

<div>
  <center>
    <h1 class="display-4">Welcome To Genre Comedy</h1>
  </center>
</div>
@endsection
@section('content')
  <div class="container">
    <div class="row">
      @foreach($editdrama as $all)
        <div class="col-md-8">
          <a href="{{route('comedy.artikel',$all->id_kategori)}}">
            </a>
            @if($all->id_kategori=='4')
              <div class="Judul">
                <h2>{{ $all->judul }}</h2>
                
                <h3>Info Drama :
                <p>Episode : {{ $all->episode }}</p>
                <p>Durasi : {{ $all->durasi }}</p></h3>
                <h3>Cast:
                <p>{{ $all->cast }}</p></h3>
                <h3>Sinopsis :
                <p>{{ $all->sinopsis }}</p></h3>
              </div>
            </div>
            @endif
          @endforeach

          <!-- catagories -->
          <div class="ctg">
            <div class="section-title">
              <h2>Pilih Genre</h2>
            </div>
            <div class="category-widget">
              <ul>
                <li><a href="/romance" class="cat-1">Romance<span>->
                </span></a></li>
                <li><a href="/action" class="cat-2">Action<span>->
                </span></a></li>
                <li><a href="/melo" class="cat-4">MeloDrama<span>->
                </span></a></li>
                <li><a href="/Comedy" class="cat-3">Comedy<span>->
                </span></a></li>

```

```

<li><a href="/history" class="cat-5">History<span>-
></span></a></li>
</ul>
</div>
</div>
</div>
</div>
</div>
@endsection

```

Source Code History:

```
@extends('layouts.MasterKd')
@section('title', 'History')
<link rel="stylesheet" type="text/css" href="{{asset('/css/c.css')}}">

@section('greeting')
<div>
    <center>
        <h1 class="display-4">Welcome To Genre History</h1>
    </center>
</div>
@endsection

@section('content')
    <div class="container">
        <div class="row">
            @foreach($editdrama as $all)
                <div class="col-md-8">
                    <a href="{{route('history.artikel',$all->id_kategori)}}">
                        </a>
                        @if($all->id_kategori=='5')
                            <div class="Judul">
                                <h2>{{ $all->judul }}</h2>
                                
                                <h3>Info Drama :
                                    <p>Episode : {{ $all->episode }}</p>
                                    <p>Durasi : {{ $all->durasi }}</p></h3>
                                <h3>Cast:
                                    <p>{{ $all->cast }}</p></h3>
                                <h3>Sinopsis :
                                    <p>{{ $all->sinopsis }}</p></h3>
                                </div>
                            </div>
                        @endif
                    </div>
                @endforeach
            </div>
        </div>
    </div>
endsection
```

```

                @endif
            @endforeach

            <!-- catagories -->
            <div class="ctg">
                <div class="section-title">
                    <h2>Pilih Genre</h2>
                </div>
                <div class="category-widget">
                    <ul>
                        <li><a href="/romance" class="cat-1">Romance<span>-
></span></a></li>
                        <li><a href="/action" class="cat-2">Action<span>-
></span></a></li>
                        <li><a href="/melo" class="cat-4">MeloDrama<span>-
></span></a></li>
                        <li><a href="/Comedy" class="cat-3">Comedy<span>-
></span></a></li>
                        <li><a href="/history" class="cat-5">History<span>-
></span></a></li>
                    </ul>
                </div>
            </div>
        </div>
    </div>
</div>
</div>
@endsection

```

CSS Page Setiap Genre :

```

*{
    margin: 5px;
    padding: 0;
    color: rgb(155, 0, 0);
    box-sizing: border-box;
    font-family: 'Arial Narrow Bold', sans-serif;
    text-decoration: none;
}
body{
    background: grey;
    color: rgb(155, 0, 0);
}
nav{
    background: black;

```

```
}
nav::after{
  content: '';
  clear: both;
  display: table;
}
nav .logo{
  float: left;
  font-style: italic;
  font-weight: bold;
  font-size: 40px;
  font-weight: 600px;
  line-height: 70px;
  padding-left: 40px;
}
nav ul{
  float: right;
  list-style: none;
  margin-right: 50px;
  position: relative;
}
nav ul li{
  float: left;
  display: block;
  margin: 10px 0px;
}
nav ul li a{
  color: rgb(190, 162, 162);
  text-decoration: none;
  line-height: 50px;
  font-weight: bolder;
  font-size: 12px;
  padding: 8px 10px;
}
nav ul li a:hover{
  color: whitesmoke;
  background: rgba(155, 0, 0, 0.658);
  border-radius: 5px;
  box-shadow: 0 0 15px rgb(228, 228, 245),
             0 0 15px rgb(219, 230, 255);
}
nav ul ul{
  position: absolute;
  background: rgba(155, 0, 0, 0.658);
  border-top: 3px solid rgb(163, 159, 159);
}
```

```
    opacity: 0;
    visibility: hidden;
    transition: bottom .3s;
}
nav ul li:hover > ul{
    opacity: 1;
    visibility: visible;
}
nav ul ul li{
    position: relative;
    margin: 0;
    width: 130px;
    float: none;
    display: list-item;
    border-bottom: 1px solid rgba(0,0,0,0.3);
}
nav ul ul li a{
    line-height: 50px;
}
.greeting h1{
    margin-top: 15px;
    font-size: 50px;
    font-family: 'Times New Roman';
    font-weight: bolder;
    border-top: 4px solid black;
    border-bottom: 4px solid black;
    color: whitesmoke;
}
.judul{
    border: 2px solid white;
    margin: 20px;
    padding: 10px;
}
.judul h2{
    font-size: 40px;
    color: rgb(255, 0, 0);
}
.judul p{
    text-align: justify;
    font-size: 18px;
    color: black;
}
.judul h3{
    font-size: 20px;
    color: rgb(126, 0, 38);
}
```

```

    font-style: oblique;
    border: 2px solid rgb(128, 0, 60);
    padding: 3px;
}
.img-responsive{
    position: relative;
    width: 70%;
    height: 50%;
    margin-left: 15%;
    border: 2px solid blue;
}
/*-----*\
    category
\*-----*/
.ctg{
    border: 2px solid rgb(0, 0, 0);
    padding: 15px;
    width: 1150px;
    height: 345px;
    margin-top: 50px;
}
.section-title{
    color: rgb(252, 252, 252);
    font-style: italic;
    font-size: 25px;
    margin-bottom: 15px;
    border: 2px solid rgb(255, 1, 1);
    font-family: cursive;
}
.category-widget a{
    font-size: 20px;
    text-decoration: none;
}
.category-widget ul li {
    display: block;
    padding-bottom: 10px;
    border-bottom: 1px solid #ecef2;
}
.category-widget ul li + li {
    margin-top: 10px;
}
.category-widget ul li > a {
    display: block;
    -webkit-transition: 0.2s color;
    transition: 0.2s color;
}

```

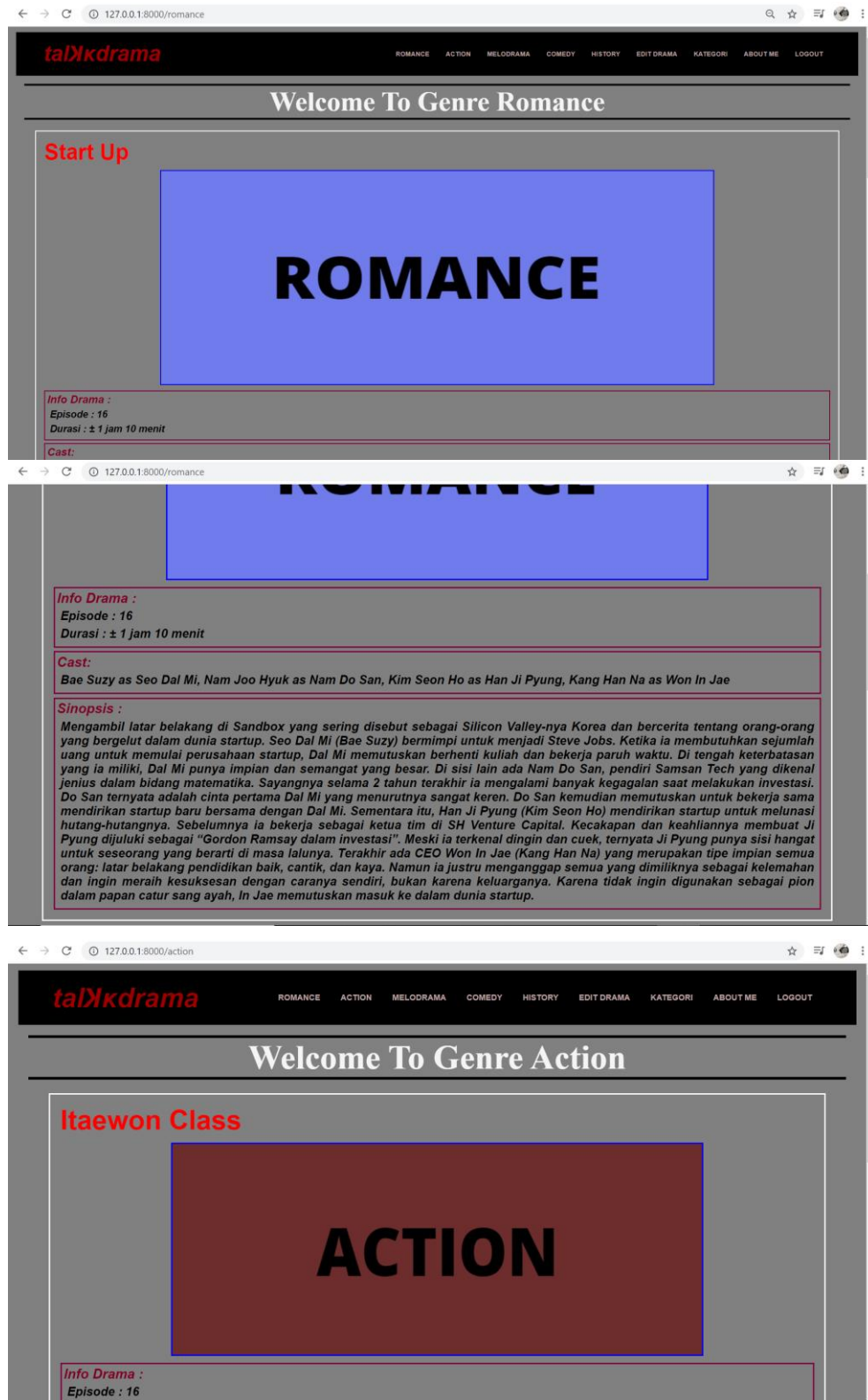


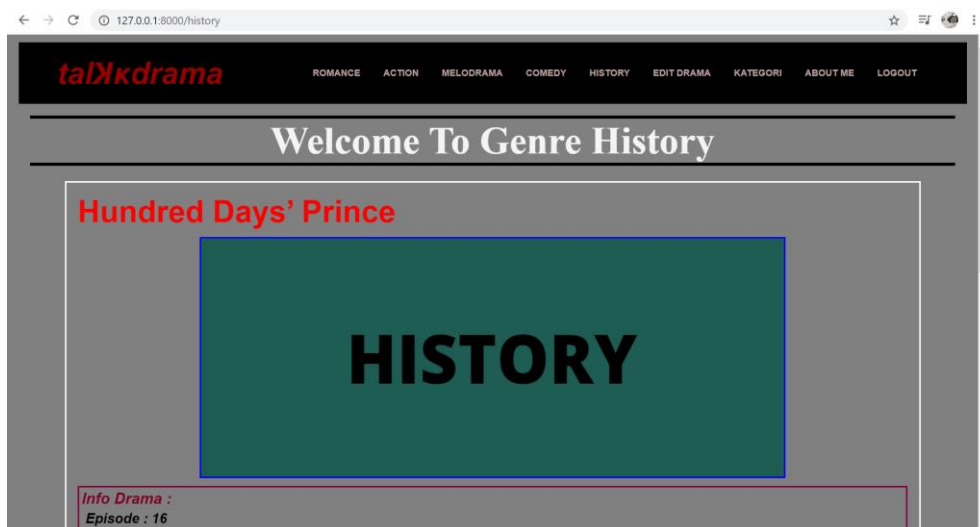
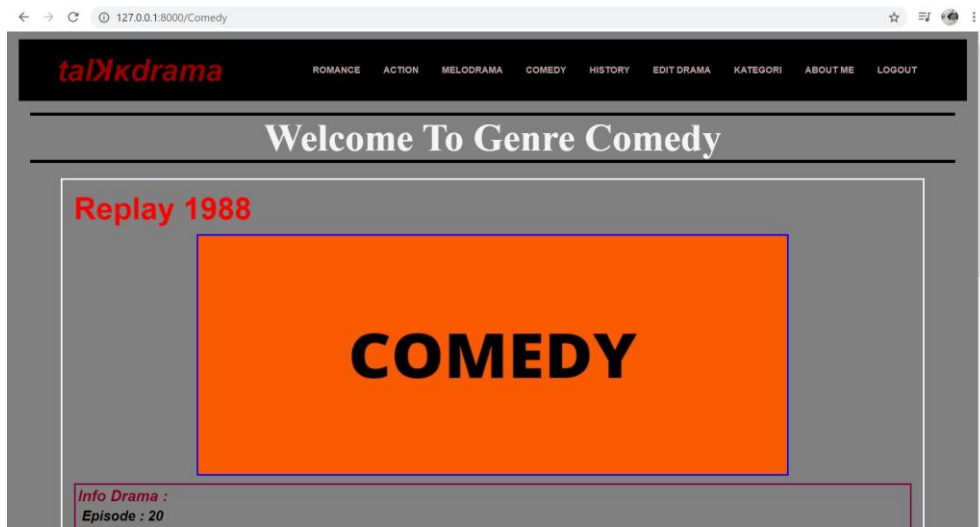
```
}
.category-widget ul li > a > span {
    float:right;
    color:#fff;
    padding:0px 5px;
    font-weight:600;
    border-radius:2px;
}
.category-widget ul li > a.cat-1 > span {
    background-color: #f8b600;
}
.category-widget ul li > a.cat-2 > span {
    background-color: #ff9100;
}
.category-widget ul li > a.cat-3 > span {
    background-color: #8d00ff;
}
.category-widget ul li > a.cat-4 > span {
    background-color: #0078ff;
}

.category-widget ul li > a.cat-5 > span {
    background-color: #ff0040;
}
.category-widget ul li > a:hover, .category-widget ul li > a:focus {
    text-decoration:none;
}
.category-widget ul li > a.cat-1:hover, .category-widget ul li > a.cat-1:focus {
    color: #4BB92F;
}
.category-widget ul li > a.cat-2:hover, .category-widget ul li > a.cat-2:focus {
    color: #ff8700;
}
.category-widget ul li > a.cat-3:hover, .category-widget ul li > a.cat-3:focus {
    color: #0078ff;
}
.category-widget ul li > a.cat-4:hover, .category-widget ul li > a.cat-4:focus {
    color: #8d00ff;
}

.category-widget ul li > a.cat-5:hover, .category-widget ul li > a.cat-4:focus {
    color: #00ff9d;
}
```

Berikut adalah tampilan tiap genre :





I. MEMBUAT TAMPILAN ABOUT ME

Pertama kita buat dulu untuk routenya

```
Route::get('/aboutkd', 'AboutKdController@kdAbout');
```

Lalu kita buat class aboutnya. Sebenarnya untuk pembuatan class about hamper sama dengan home. Yang membedakan adalah isi kontennya

```
@extends('layouts.Masterkd')
@section('title', 'About Talk-Kdrama')
<link rel="stylesheet" type="text/css" href="{{asset('/css/abkd.css')}}">

@section('greeting')
<div>
    <center>
        <h1 class="display-4">Welcome To About Me</h1>
    </center>
</div>
@endsection

@section('content')
<div class="about-area">
    <h2>Stay Safe, Stay Healthy, Stay Happy</h2>
    
    <p>
        Hai Kenalin aku Wahyu Indah Puspitasari, kalian bisa panggil
        aku Wahyu atau Indah. Aku lahir di Jombang, 30 Juli 2000. Aku adalah
        Mahasiswa Semester tiga Jurusan Teknologi Informasi di Politeknik Negeri Malang.
        Aku suka banget baca novel dan nonton film atau drama korea kalau
        punya waktu yang lumayan senggang. Sebelum nonton sebuah film atau kdrama aku
        biasanya baca sinopsis dan reviewnya dulu. Karena itulah aku terinspirasi membuat web
        ini.</p>
    <center>
        <a href="https://www.instagram.com/wahyuindah_p?igshid=1xg2ethytw813">Follow Me</a>
    </center>
</div>
@endsection
```

CSS Page About :

```
*{
    margin: 5px;
    padding: 0;
    color: rgb(155, 0, 0);
    box-sizing: border-box;
```

```
font-family: 'Arial Narrow Bold', sans-serif;
text-decoration: none;
}
```

```
body{
  background: grey;
}
```

```
nav{
  background: black;
}
```

```
nav::after{
  content: '';
  clear: both;
  display: table;
}
```

```
nav .logo{
  float: left;
  font-style: italic;
  font-weight: bold;
  font-size: 40px;
  font-weight: 600px;
  line-height: 70px;
  padding-left: 40px;
}
```

```
nav ul{
  float: right;
  list-style: none;
  margin-right: 50px;
  position: relative;
}
```

```
nav ul li{
  float: left;
  display: block;
  margin: 10px 0px;
}
```

```
nav ul li a{
  color: rgb(190, 162, 162);
  text-decoration: none;
  line-height: 50px;
  font-weight: bolder;
  font-size: 12px;
  padding: 8px 10px;
}
```

```
nav ul li a:hover{
  color: whitesmoke;
  background: rgba(238, 4, 4, 0.658);
}
```

```
border-radius: 5px;
box-shadow: 0 0 15px rgb(228, 228, 245),
           0 0 15px rgb(219, 230, 255);
}
nav ul ul{
  position: absolute;
  background: rgba(255, 4, 4, 0.658);
  border-top: 3px solid rgb(163, 159, 159);
  opacity: 0;
  visibility: hidden;
  transition: bottom .3s;
}
nav ul li:hover > ul{
  opacity: 1;
  visibility: visible;
}
nav ul ul li{
  position: relative;
  margin: 0;
  width: 130px;
  float: none;
  display: list-item;
  border-bottom: 1px solid rgba(0,0,0,0.3);
}
nav ul ul li a{
  line-height: 50px;
}
.greeting h1{
  margin-top: 15px;
  font-size: 50px;
  font-family: 'Times New Roman';
  font-weight: bolder;
  border-top: 4px solid black;
  border-bottom: 4px solid black;
  color: whitesmoke;
}
.about-area h2{
  color: rgb(214, 212, 212);
  font-family: lobster;
  font-size: 50px;
  background-color:rgb(155, 0, 0);
  border-radius: 10px;
  text-align: justify;
  padding: 15px 10px;
  padding-left: 22%;
}
```

```

}
.about-area img{
    position: relative;
    width: 1210px;
    height: 150px;
    padding: 10px;
    border-radius: 50px;
}
.about-area p{
    font-size: 30px;
    color: rgb(255, 218, 218);
    background-color:rgb(155, 0, 0);
    border-radius: 10px;
    text-align: justify;
    padding: 10px 10px;
}
.about-area a{
    background-color: black;
    font-family: monospace;
    font-size: 20px;
    color: red;
    padding: 7px 15px;
    display: inline-block;
    margin-top: 20px;
}

```

Kemudian kita buat controller untuk about. Seperti controller sebelumnya kita bisa ketikkan perintah *php artisan make:controller namaController*. Berikut adalah source code controller about.

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class AboutKdController extends Controller
{
    public function kdAbout()
    {
        return view ('ProjectDrama.aboutkd');
    }
}

```

Tampilan Halaman About :

Welcome To About Me

Stay Safe, Stay Healthy, Stay Happy



Hai Kenalin aku Wahyu Indah Puspitasari, kalian bisa panggil aku Wahyu atau Indah. Aku lahir di Jombang, 30 Juli 2000. Aku adalah Mahasiswa Semester tiga Jurusan Teknologi Informasi di Politeknik Negeri Malang. Aku suka banget baca novel dan nonton film atau drama korea kalau punya waktu yang lumayan senggang. Sebelum nonton sebuah film atau kdrama aku biasanya baca sinopsis dan reviewnya dulu. Karena itulah aku terinspirasi membuat web ini.

[Follow Me](#)