# LAPORAN APLIKASI

# 

# APLIKASI PEMINJAMAN PERPUSTAKAAN

# DI SMKN 1 KARANG BARU

Disusun untuk memenuhi syarat mengikuti ujian kompetensi kejuruan pada

Kompetensi keahlian Rekayasa Perangkat Lunak

**Disusun oleh:**

**NAMA : AHMADI MUSLIM**

**NISN : - DUMMY-**

**REKAYASA PERANGKAT LUNAK**

**SMK NEGERI 1 KARANG BARU**

**TAHUN 2024**

# LEMBAR PERSETUJUAN

# APLIKASI PEMINJAMAN PERPUSTAKAAN

# DI SMKN 1 KARANG BARU

**DISUSUN OLEH :**

**NAMA : AHMADI MUSLIM**

**NISN : - DUMMY -**

Telah disetujui oleh Guru Pembimbing pada tanggal 29 Agustus 2024

Untuk dipertanggungjawabkan di depan Penguji

Kompetensi Keahlian Rekayasa Perangkat Lunak

SMK Negeri 1 Karang baru

Mengetahui,

|  |  |
| --- | --- |
| Kepala Kompetensi Keahlian Rekayasa Perangkat Lunak | Guru Pembimbing |
| **Safrizal, S.ST**  NIP. 19870530 201003 1 001 | **Ahmadi Muslim, MP**  NIP. 19860505 201403 1 003 |

# BAB I ANALISIS KEBUTUHAN APLIKASI KASIR TOKO KOMPUTER

## Kuesioner Analisis Kebutuhan Aplikasi

### Bagian I: Data Responden

Nama :

Jabatan :

Pendidikan Terakhir :

Divisi/Bidang :

Lama bekerja di lokasi :

* < 1 tahun
* 1-3 tahun
* 3-8 tahun
* > 8 tahun

### Bagian 2: Penggunaan Teknologi

* Gawai/gadget apakah yang Anda miliki (bisa memilih lebih dari satu)?
* Laptop/notebook
* PC
* Tablet
* Smartphone
* Seberapa sering Anda menggunakan laptop/PC?
* Setiap hari
* Beberapa kali dalam seminggu
* Kurang dari sekali dalam seminggu
* Kurang dari sekali dalam sebulan
* Seberapa sering Anda menggunakan smartphone?
* Setiap hari
* Beberapa kali dalam seminggu
* Kurang dari sekali dalam seminggu
* Kurang dari sekali dalam sebulan
* Hal apakah yang anda lakukan dengan gawai anda (bisa memilih lebih dari satu)?
* Main game
* Komunikasi (calling atau chating)
* Membuat laporan dan kerjaan kantor lainnya
* Nonton film
* Editing Desain grafis
* Konten kreator
* Pemrograman
* Aplikasi/software apa saja yang pernah anda gunakan sehari-hari dengan gawai anda (bisa memilih lebih dari satu)?
* Game
* Ms. Word
* Ms. Excel
* Powerpoint
* Web browser
* Aplikasi untuk tugas khusus (seperti digilib, Olshop, POS, Absensi, dll)

### Bagian 3: Proses Pendataan kasir Saat Ini

* Hal apa saja yang Anda lakukan di lokasi kerja anda sehari hari (bisa memilih lebih dari satu)?
* Mendata barang masuk dan keluar
* Mendata absensi pegawai
* Mendata gaji pegawai
* Melakukan transaksi penjualan
* Melakukan pencatatan penjualan
* Merekap hasil penjualan
* Berapa volume/trafik data proses transaksi penjualan di toko anda per hari?
* 0-5 transaksi
* 6-50 transaksi
* 51-500 transaksi
* 501-1000 transaksi
* Diatas 1000 transaksi
* Siapa saja yang terlibat dalam proses penjualan/kasir toko Anda?
* Staf khusus
* Atasan langsung
* Penjualan terkait
* Bagaimana cara pelayanan transaksi kasir di toko anda yang dilakukan saat ini?
* Manual (menggunakan kertas/buku penjualan)
* Spreadsheet (Excel, Google Sheets)
* Sistem/aplikasi lain
* Seberapa nyaman Anda dalam menggunakan model layanan tersebut?
* Sangat nyaman
* Nyaman
* Cukup nyaman
* Tidak nyaman
* Sangat tidak nyaman
* Apa saja kesulitan yang Anda hadapi dalam proses kasir/penjualan toko anda saat ini (bisa memilih lebih dari satu)?
* Data tidak akurat
* Proses memakan waktu lama
* Proses perhitungan rentan kekeliruan
* Tidak adanya laporan secara menyeluruh secara priodik
* Tidak ada bukti pembayaran bagi customer
* Sulit mencari riwayat penjualan

### Bagian 4: Kebutuhan Fitur Aplikasi

* Menurut Anda, fitur apa saja yang seharusnya ada dalam aplikasi kasir toko komputer? (bisa memilih lebih dari satu)
* Input data
* Update data
* Delete data
* Edit data
* Scan barcode
* Total transaksi
* Cetak transaksi
* Laporan transaksi
* Akses ke Olshop

### Bagian 5: Saran dan Masukan

* Apakah Anda memiliki saran atau masukan lainnya terkait pengembangan aplikasi kasir toko komputer ini?

## Daftar Pertanyaan Wawancara

### Bagian 1: Pengalaman dan Kebutuhan Umum

* Apakah Anda sudah pernah menggunakan aplikasi kasir toko komputer sebelumnya?
* Bagaimana proses absensi di tempat Anda bekerja saat ini?
* Kegiatan apa saja yang anda lakukan di toko komputer pada saat ini?
* Menurut Anda, seberapa penting toko anda untuk memiliki aplikasi kasir/POS?

### Bagian 2: Fitur dan Fungsi yang Diinginkan

* Apa saja fitur yang Anda anggap penting dalam aplikasi kasir toko komputer?
* Bagaiman fitur laporan di dalam aplikasi yang anda harapkan?
* Bagaimana fitur transaksi/layanan penjualan yang anda harapkan?
* Hal apa saja yang anda harapkan pada fitur barang?
* Bagaimana proses pencetakan bukti pembayaran yang diberikan ke customer yang anda harapkan?
* Bagiamana proses pemberitahuan (notifikasi) transaksi penjualan yang anda inginkan?

### Bagian 3: Antarmuka Pengguna dan Pengalaman

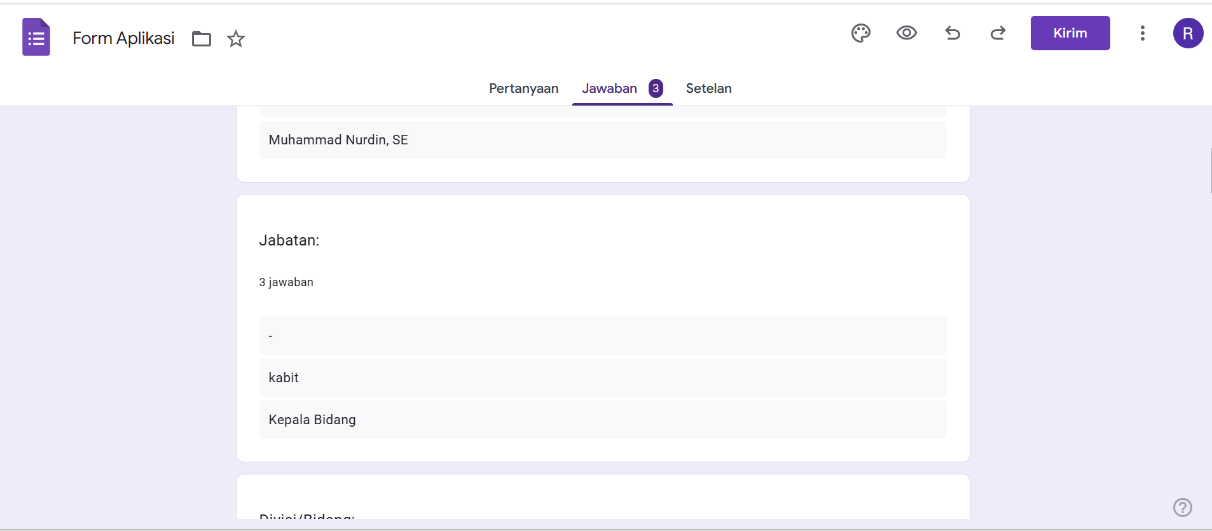
* Bagaimana tampilan login yang anda inginkan (boleh digambarkan)?
* Bagaimana tampilan dashboard/halaman depan yang anda inginkan (boleh digambarkan)?
* Bagaimana tampilan tampil barang yang anda inginkan (boleh digambarkan)?
* Bagaimana tampilan tambah barang depan yang anda inginkan (boleh digambarkan)?
* Bagaimana tampilan transaksi yang anda inginkan (boleh digambarkan)?
* Bagaimana tampilan cetak/struk yang anda inginkan (boleh digambarkan)?

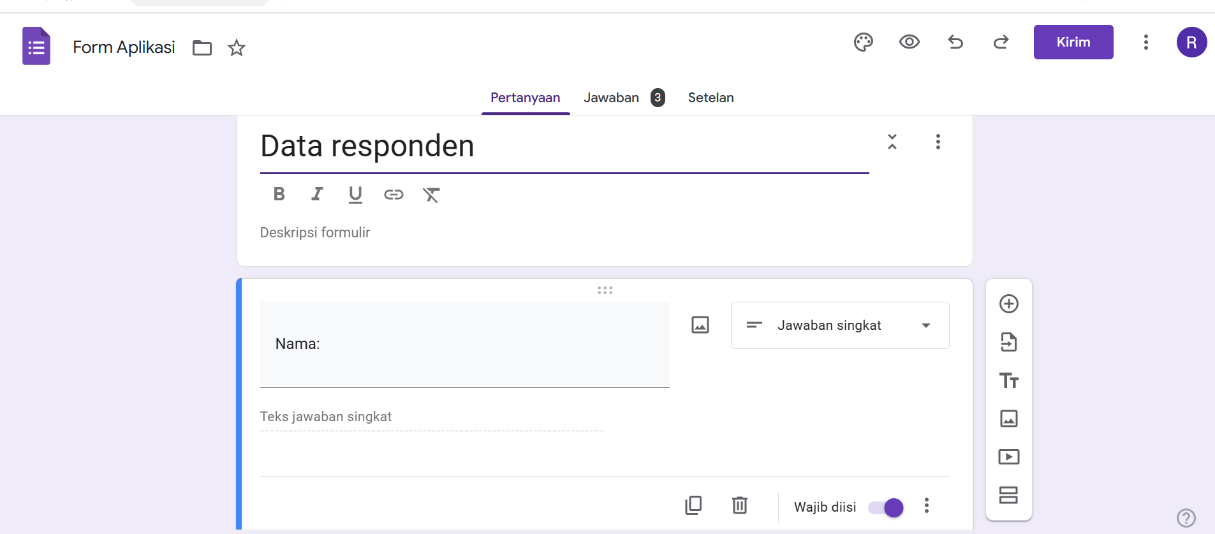
### Bagian 4: Masukan dan Saran

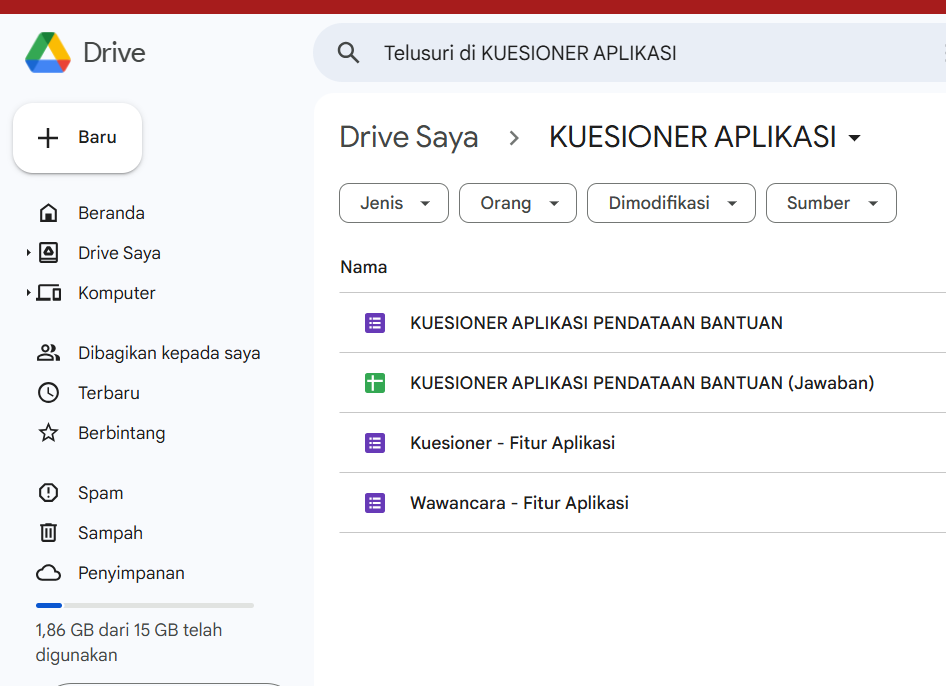
* Apa saran atau masukan tambahan yang Anda ingin berikan untuk pengembangan aplikasi ini?

## Dokumen Observasi

1. **Formulir dengan google form**





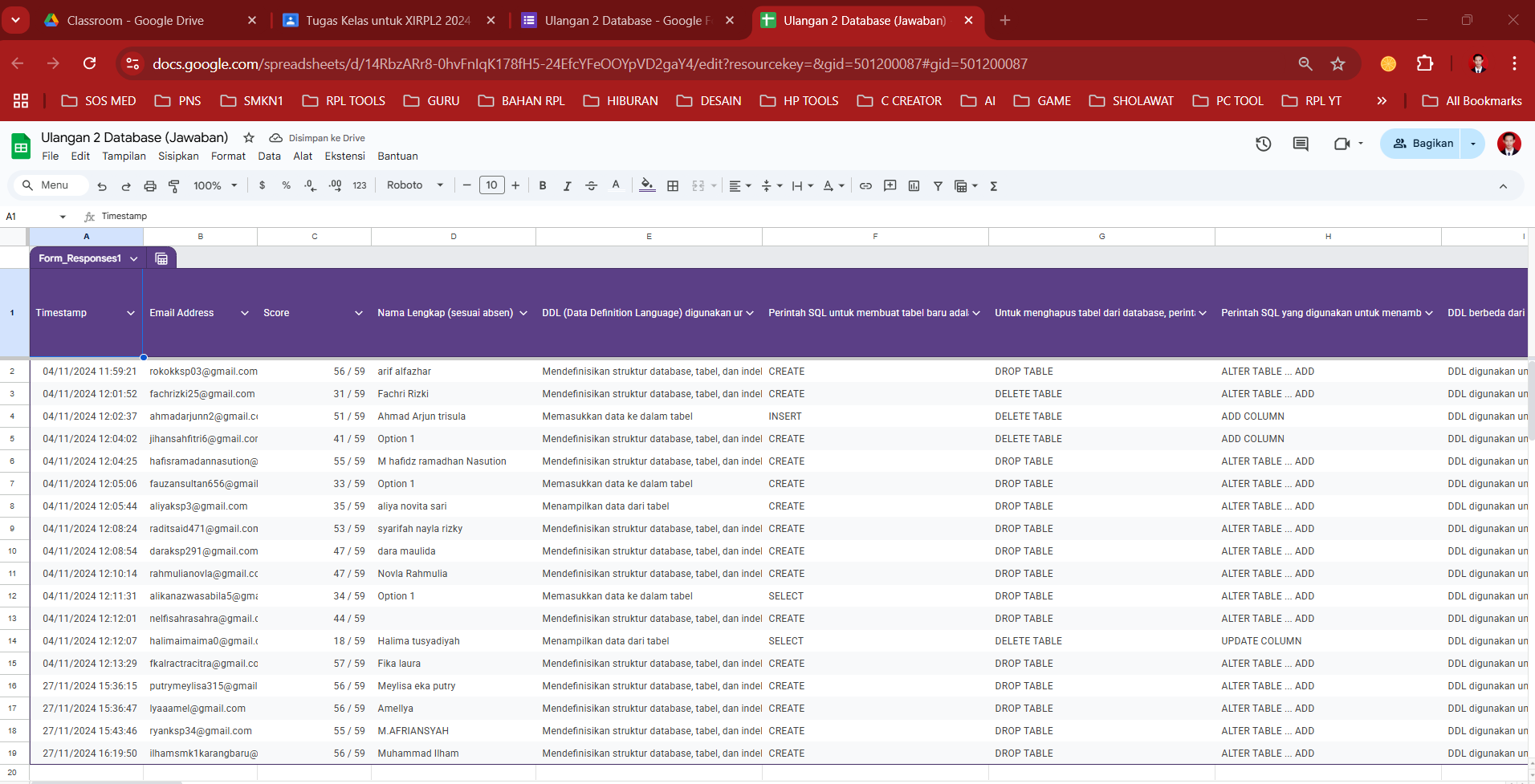


1. **Dokumentasi pengisian formulir**





1. **Hasil pengisian formulir**



**Rekapitulasi Observasi**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Daftar Pertanyaan** | **Jawaban** | | | |
| **Responden A (pemilik)** | **Responden B (admin)** | **Responden C (pelanggan)** | **Responden D (random)** |
| **Bagian 2: Penggunaan Teknologi** |  |  |  |  |
| - Gawai/gadget apakah yang Anda miliki (bisa memilih lebih dari satu)? | Smartphone | Smartphone | Smartphone | Smartphone |
| - Seberapa sering Anda menggunakan laptop/PC? | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
| - Seberapa sering Anda menggunakan smartphone? | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
| - Hal apakah yang anda lakukan dengan gawai anda (bisa memilih lebih dari satu)? | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film |
| - Aplikasi/software apa saja yang pernah anda gunakan sehari-hari dengan gawai anda (bisa memilih lebih dari satu)? | Smartphone | Smartphone | Smartphone | Smartphone |
|  | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
| **Bagian 3: Proses Pendataan kasir Saat Ini** | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
| - Hal apa saja yang Anda lakukan di lokasi kerja anda sehari hari (bisa memilih lebih dari satu)? | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | ain game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film |
| - Berapa volume/trafik data proses transaksi penjualan di toko anda per hari? | Smartphone | Smartphone | Smartphone | Smartphone |
| - Siapa saja yang terlibat dalam proses penjualan/kasir toko Anda? | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
| - Bagaimana cara pelayanan transaksi kasir di toko anda yang dilakukan saat ini? | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
| - Seberapa nyaman Anda dalam menggunakan model layanan tersebut? | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film |
| - Apa saja kesulitan yang Anda hadapi dalam proses kasir/penjualan toko anda saat ini (bisa memilih lebih dari satu)? | Smartphone | Smartphone | Smartphone | Smartphone |
|  | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
| **Bagian 4: Kebutuhan Fitur Aplikasi** | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
| - Menurut Anda, fitur apa saja yang seharusnya ada dalam aplikasi kasir toko komputer? (bisa memilih lebih dari satu) | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film |
|  | Smartphone | Smartphone | Smartphone | Smartphone |
| **Bagian 5: Saran dan Masukan** | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
| - Apakah Anda memiliki saran atau masukan lainnya terkait pengembangan aplikasi kasir toko komputer ini? | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
|  | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film |
| **A.    Daftar Pertanyaan Wawancara** | Smartphone | Smartphone | Smartphone | Smartphone |
| **Bagian 1: Pengalaman dan Kebutuhan Umum** | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
| - Apakah Anda sudah pernah menggunakan aplikasi kasir toko komputer sebelumnya? | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
| - Bagaimana proses pelayanan kasir toko komputer di tempat Anda bekerja saat ini? | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film |
| - Kegiatan apa saja yang anda lakukan di toko komputer pada saat ini? | Smartphone | Smartphone | Smartphone | Smartphone |
| - Menurut Anda, seberapa penting toko anda untuk memiliki aplikasi kasir/POS? | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
|  | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
| **Bagian 2: Fitur dan Fungsi yang Diinginkan** | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film |
| - Apa saja fitur yang Anda anggap penting dalam aplikasi kasir toko komputer? | Smartphone | Smartphone | Smartphone | Smartphone |
| - Bagaiman fitur laporan di dalam aplikasi yang anda harapkan? | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
| - Bagaimana fitur transaksi/layanan penjualan yang anda harapkan? | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
| - Hal apa saja yang anda harapkan pada fitur barang? | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film |
| - Bagaimana proses pencetakan bukti pembayaran yang diberikan ke customer yang anda harapkan? | Smartphone | Smartphone | Smartphone | Smartphone |
| - Bagiamana proses pemberitahuan (notifikasi) transaksi penjualan yang anda inginkan? | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
|  | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
| **Bagian 3: Antarmuka Pengguna dan Pengalaman** | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film |
| - Bagaimana tampilan login yang anda inginkan (boleh digambarkan)? | Smartphone | Smartphone | Smartphone | Smartphone |
| - Bagaimana tampilan dashboard/halaman depan yang anda inginkan (boleh digambarkan)? | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
| - Bagaimana tampilan tampil barang yang anda inginkan (boleh digambarkan)? | Setiap hari | Setiap hari | Setiap hari | Setiap hari |
| - Bagaimana tampilan tambah barang depan yang anda inginkan (boleh digambarkan)? | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film | Main game, Membuat laporan dan kerjaan kantor lainnya, Nonton film |
| - Bagaimana tampilan transaksi yang anda inginkan (boleh digambarkan)? | Smartphone | Smartphone | Smartphone | Smartphone |
| - Bagaimana tampilan cetak/struk yang anda inginkan (boleh digambarkan)? |  |  |  |  |
|  |  |  |  |  |
| **Bagian 4: Masukan dan Saran** |  |  |  |  |
| - Apa saran atau masukan tambahan yang Anda ingin berikan untuk pengembangan aplikasi ini? |  |  |  |  |

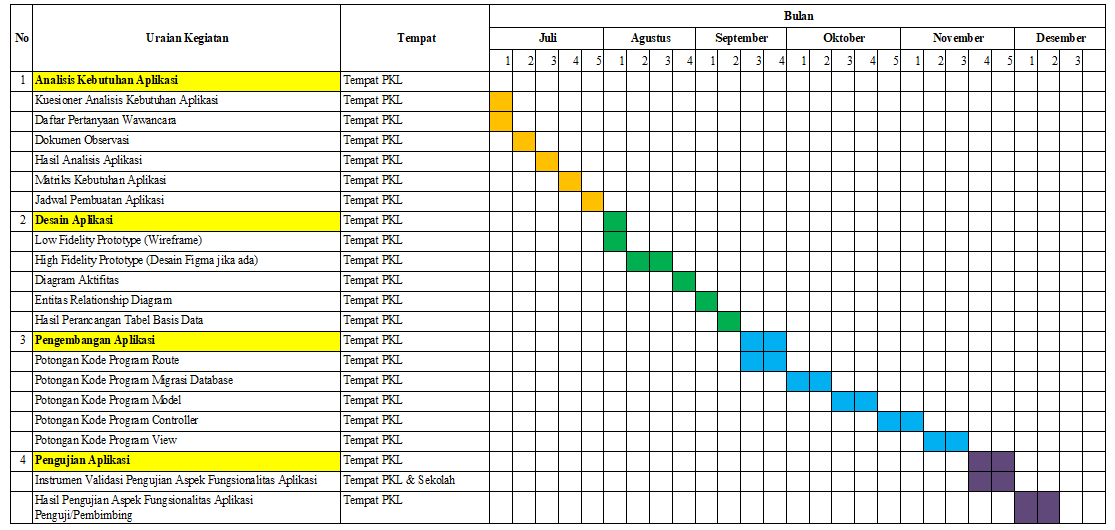
## Hasil Analisis Aplikasi

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Fitur** | **Pendukung** | **Urgensi/Tingkat Kepentingan** | | | | | |
| **Create** | **Read (tampil)** | **Update** | **Delete** | **Printout** | **Pencarian** |
| Login | username |  | perlu |  |  |  |  |
|  | password |  | perlu |  |  |  |  |
| Dashboard | admin |  | perlu |  |  |  |  |
|  | pelanggan |  | perlu |  |  |  |  |
|  | barang |  | perlu |  |  |  |  |
|  | penjualan |  | perlu |  |  |  |  |
| Pengguna/Admin | namaadmin | perlu | perlu | perlu | perlu |  | perlu |
|  | username | perlu | perlu | perlu | perlu |  | perlu |
|  | password | perlu | perlu | perlu | perlu |  | perlu |
|  | alamat | perlu | perlu | perlu | perlu |  | perlu |
| Pelanggan | namapelanggan | perlu | perlu | perlu | perlu |  | perlu |
|  | username | perlu | perlu | perlu | perlu |  | perlu |
|  | password | perlu | perlu | perlu | perlu |  | perlu |
|  | alamat | perlu | perlu | perlu | perlu |  | perlu |
| Barang | namabarang | perlu | perlu | perlu | perlu |  | perlu |
|  | kategori | perlu | perlu | perlu | perlu |  | perlu |
|  | merk | perlu | perlu | perlu | perlu |  | perlu |
|  | suplier | perlu | perlu | perlu | perlu |  | perlu |
|  | stok | perlu | perlu | perlu | perlu |  | perlu |
|  | harga | perlu | perlu | perlu | perlu |  | perlu |
| Barang->Kategori | namakategori | perlu | perlu | perlu | perlu |  | perlu |
| Barang->Suplier | namakategori | perlu | perlu | perlu | perlu |  | perlu |
|  | alamat | perlu | perlu | perlu | perlu |  | perlu |
| Transaksi | nomor | perlu | perlu | perlu | perlu |  | perlu |
|  | namapelanggan | perlu | perlu | perlu | perlu |  | perlu |
|  | namaadmin | perlu | perlu | perlu | perlu |  | perlu |
|  | namabarang | perlu | perlu | perlu | perlu |  | perlu |
|  | jumlahbarang | perlu | perlu | perlu | perlu |  | perlu |
|  | total | perlu | perlu |  |  |  |  |
| Rekap/laporan | rekap-harian |  | perlu |  |  |  |  |
|  | rekap-bulanan |  | perlu |  |  |  |  |
|  | rekap-tahunan |  | perlu |  |  |  |  |
| Cetak | cetak |  |  |  |  | perlu | perlu |
|  |  |  |  |  |  |  |  |

## Matriks Kebutuhan Aplikasi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Fitur/Fungsionalitas** | **Deskripsi Singkat** | **Prioritas** | **Tingkat Kepentingan** | **Keterangan** |
|  | Login | Melindungi data, membatasi hak akses. | Tinggi | Sangat Penting | Keamanan harus diterapkan untuk memastikan data penjualan tetap rahasia dan hanya diakses oleh pihak yang berwenang. |
|  | Pengelolaan Data barang, admin dan pelanggan | Menginput, mengedit, dan menghapus data barang, admin dan pelanggan | Tinggi | Sangat Penting | Fitur inti untuk pengelolaan data seluruh barang, admin dan pelanggan. Harus ada validasi data dan hak akses pengguna tertentu. |
|  | Transaksi Penjualan | Menampilkan proses transaksi lengkap penjualan | Tinggi | Sangat Penting | Fitur inti untuk pengelolaan data seluruh penjualan. Harus ada validasi data dan hak akses pengguna tertentu. |
|  | Pelaporan Data Penjualan | Membuat laporan data penjualan | Sedang | Penting | Laporan harus dapat disesuaikan dengan filter tertentu sesuai kebutuhan pengguna (admin). |

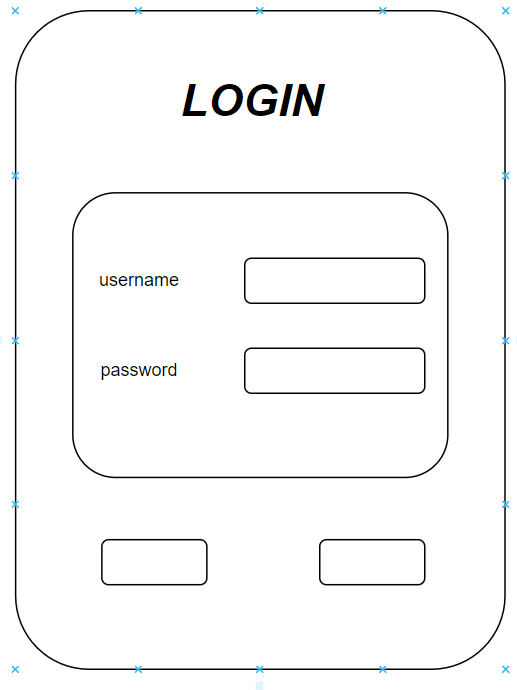
## Jadwal Pembuatan Aplikasi



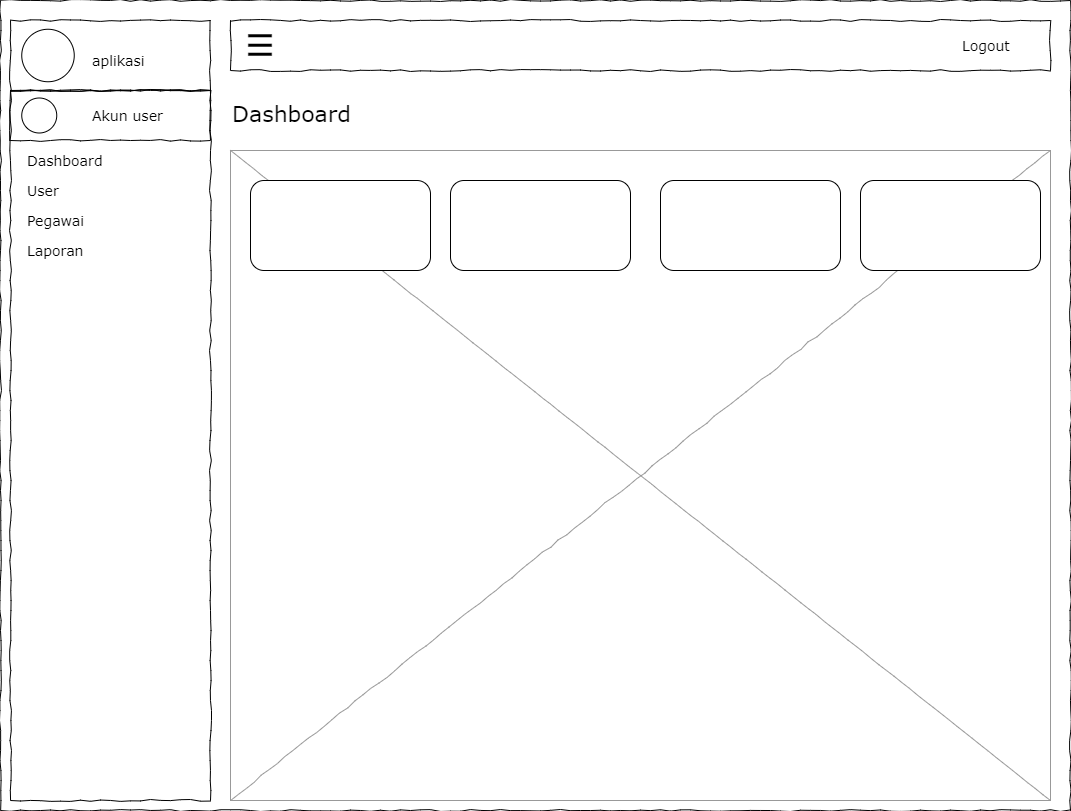
# BAB II DESAIN APLIKASI

## Low Fidelity Prototype (Wireframe)

* 1. **Low Fidelity Prototype Login**



* 1. **Low Fidelity Prototype Dashboard**

****

* 1. **Low Fidelity Prototype User**

|  |  |
| --- | --- |
| Tampil | Tambah |
| Edit | Show |

* 1. **Low Fidelity Prototype Pelanggan**

|  |  |
| --- | --- |
| Tampil | Tambah |
| Edit | Show |

* 1. **Low Fidelity Prototype Produk**

|  |  |
| --- | --- |
| Tampil | Tambah |
| Edit | Show |

* 1. **Low Fidelity Prototype Penjualan**

|  |  |
| --- | --- |
| Tampil | Tambah |
| Edit | Show |

* 1. **Low Fidelity Prototype laporan**

|  |  |
| --- | --- |
| Tampil | Tambah |
| Edit | Show |

## High Fidelity Prototype (Desain Figma jika ada)

1. **High** **Fidelity Prototype Dashboard**

****

1. High **Fidelity Prototype User**

|  |  |
| --- | --- |
| Tampil | Tambah |
| Edit | Show |

1. **High** **Fidelity Prototype Pelanggan**

|  |  |
| --- | --- |
| Tampil | Tambah |
| Edit | Show |

1. High **Fidelity Prototype Layanan**

|  |  |
| --- | --- |
| Tampil | Tambah |
| Edit | Show |

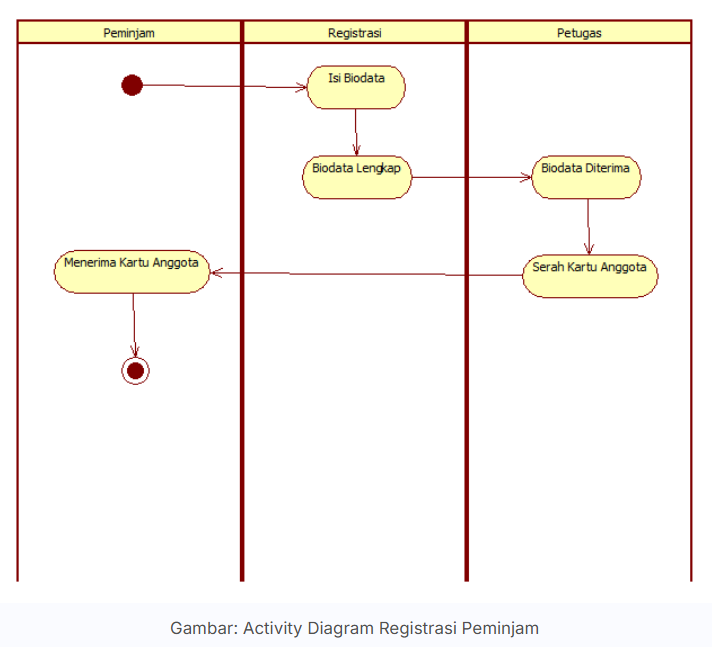
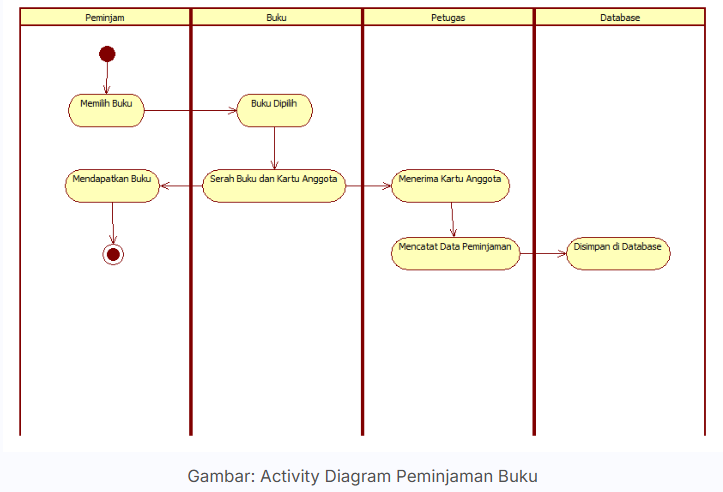
1. High **Fidelity Prototype Penjualan**

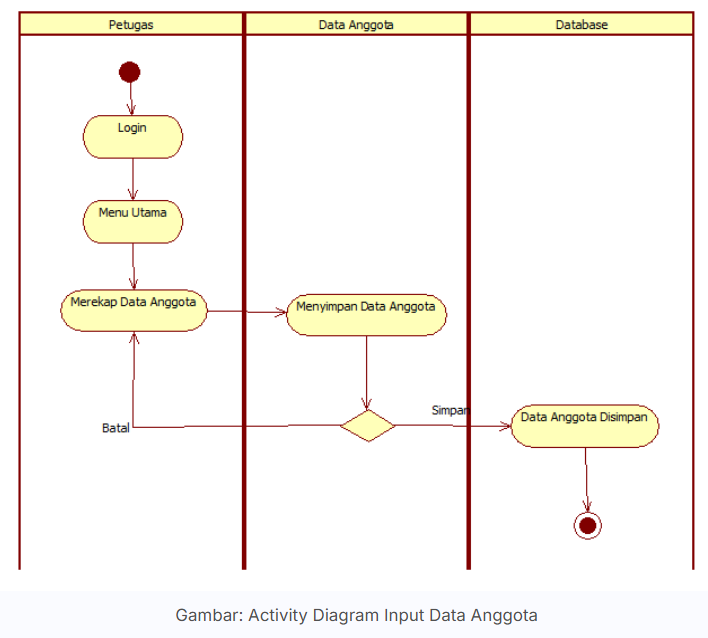
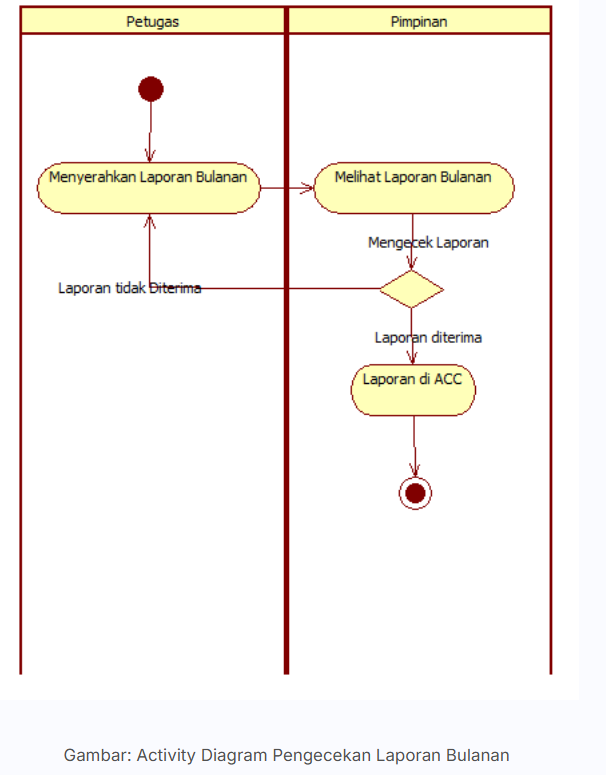
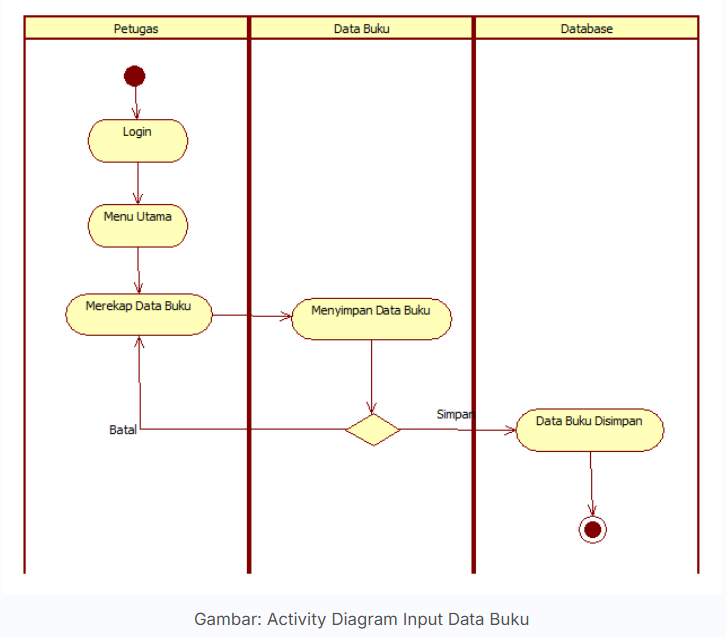
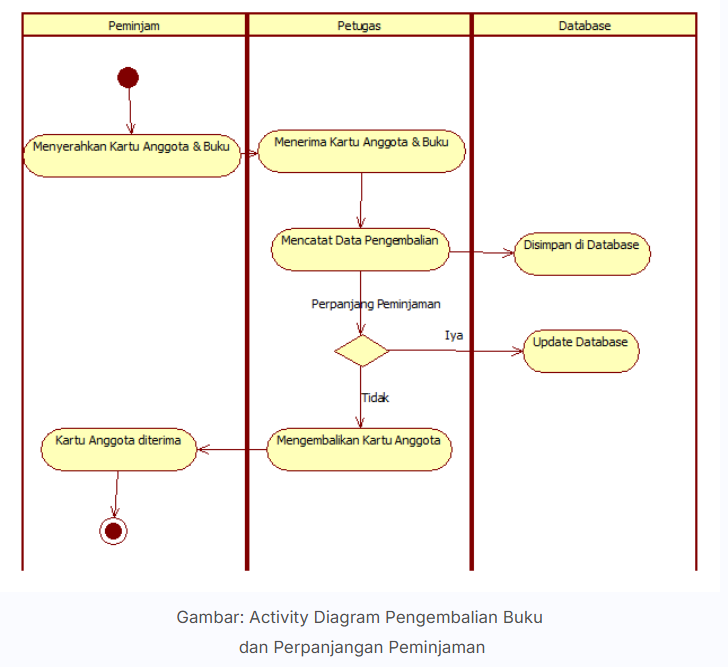
|  |  |
| --- | --- |
| Tampil | Tambah |
| Edit | Show |

1. High **Fidelity Prototype laporan**

|  |  |
| --- | --- |
| Tampil | Tambah |

## Diagram Aktifitas





# BAB IV DESAIN DATABASE

## Entitas Relationship Diagram

Unf



1nf



2nf



3nf



## Penggaturan Enviroment (file .env)

Hilangkan hastag pada bagian koneksi database yang akan dirancang di env sbb:

DB\_CONNECTION=sqlite

# DB\_HOST=127.0.0.1

# DB\_PORT=3306

# DB\_DATABASE=laravel

# DB\_USERNAME=root

# DB\_PASSWORD=

Sesuaikan dengan kebuatuhan aplikasi sehingga menjadi sbb:

DB\_CONNECTION=mysql

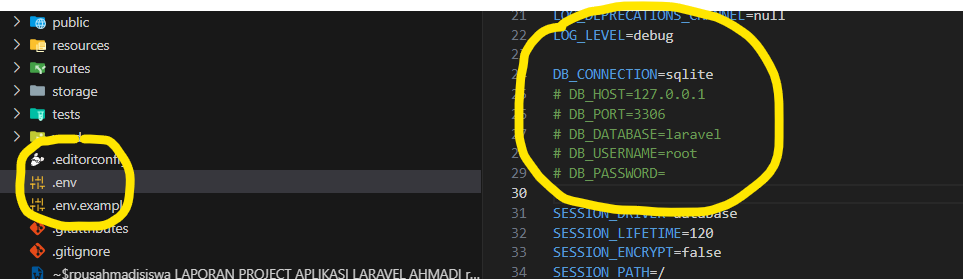
DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE= perpusahmadisiswa

DB\_USERNAME=root

DB\_PASSWORD=



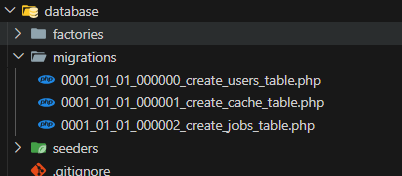
## Migrasi Database

**Perlu Diingat Sebelumnya…!**

Pembuatan tabel didatabase di mulai dari tabel induk (parent\_table) dahulu, baru kemudian tabel anak (child\_table)

Ada 2 cara membuat file migrasi, yaitu:

* Dibuat dalam 1 file dengan nama file bebas
* Migrasi dalam beberapa file yang masing-masing file sesuai nama tabelnya.

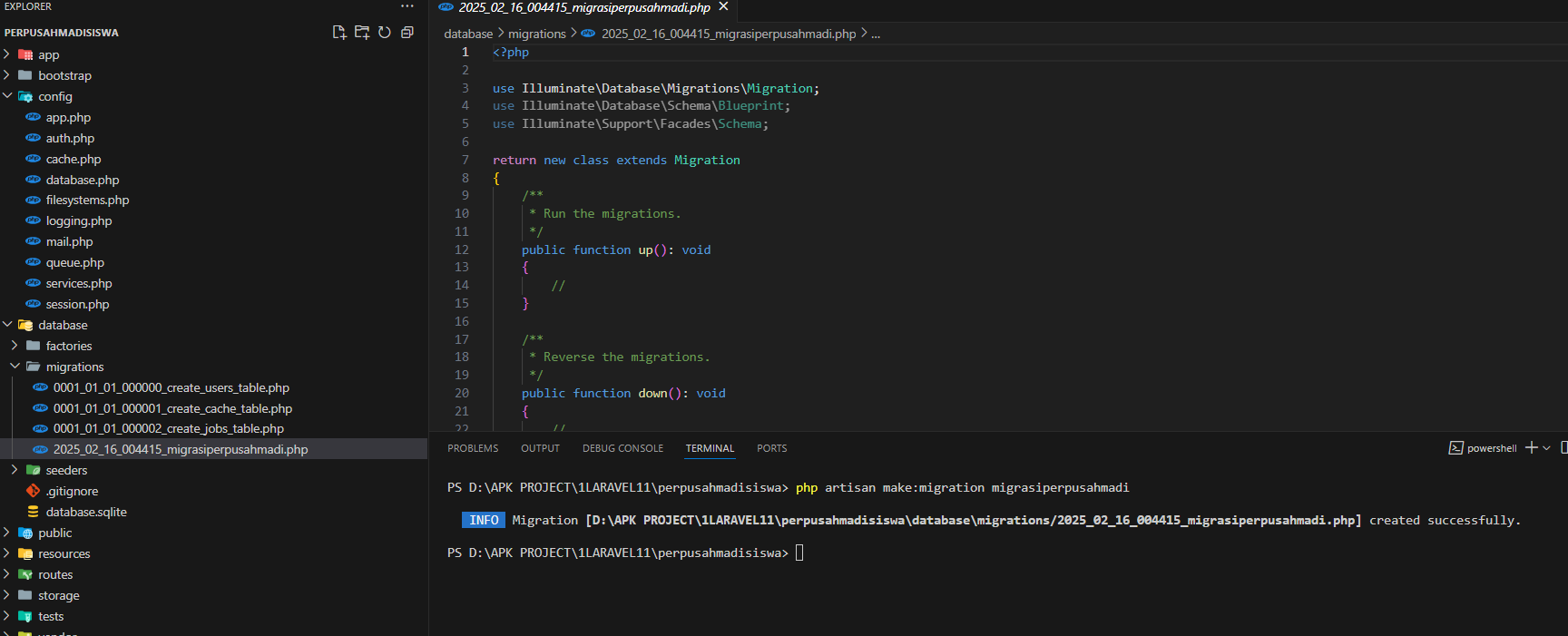
Namun sebelumnya, jika ingin menggunakan admin untuk Login, maka kita harus menghapus file migrasi user di folder migration sbb:  


1. **Model 1 file**

Biasanya, file migrasi ini bisa saja dibuat secara manual, namun bisa juga dibuat secara otomatis menggunakan perintah Artisan:

php artisan make:migration migrasiperpusahmadi

jika berhasil tampilannya akan sbb:



Perintah ini akan menghasilkan **file baru** di dalam direktori: **database/migrations/**

Buka file : migrasiperpusahmadi.php

Dan copykan file berikut:

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*/

    public function up(): void

    {

        Schema::create('kategori', function (Blueprint $table) {

            $table->id();

            $table->string('namakategori');

            $table->timestamps();

        });

        Schema::create('buku', function (Blueprint $table) {

            $table->id();

            $table->foreignId('idkategori')->constrained('kategori')->onDelete('cascade');

            $table->string('nomorseri');

            $table->string('judul');

            $table->string('penerbit');

            $table->string('pengarang');

            $table->year('tahun');

            $table->enum('kondisi', ['rusak', 'bagus'])->default('bagus');

            $table->string('rak');

            $table->string('foto')->nullable();

            $table->timestamps();

        });

        Schema::create('peminjam', function (Blueprint $table) {

            $table->id();

            $table->string('namapeminjam');

            $table->string('username')->unique();

            $table->string('password');

            $table->enum('status', ['pending', 'setujui', 'tolak'])->default('pending');

            $table->enum('keterangan', ['siswa', 'guru', 'umum'])->default('siswa');

            $table->text('alamat');

            $table->timestamp('setujui')->nullable();

            $table->string('foto')->nullable();

            $table->timestamps();

        });

        Schema::create('admin', function (Blueprint $table) {

            $table->id();

            $table->string('namaadmin');

            $table->string('username')->unique();

            $table->string('password');

            $table->enum('status', ['pending', 'setujui', 'tolak'])->default('pending');

            $table->enum('role', ['admin', 'petugas'])->default('petugas');

            $table->string('foto')->nullable();

            $table->timestamp('setujui')->nullable();

            $table->timestamps();

        });

        Schema::create('denda', function (Blueprint $table) {

            $table->id();

            $table->string('namadenda');

            $table->decimal('denda', 10, 2);

            $table->timestamps();

        });

        Schema::create('peminjaman', function (Blueprint $table) {

            $table->id();

            $table->foreignId('idpeminjam')->constrained('peminjam')->onDelete('cascade');

            $table->foreignId('idadmin')->constrained('admin')->onDelete('cascade');

            $table->decimal('totaldenda', 10, 2)->default(0);

            $table->timestamps();

        });

        Schema::create('detilpeminjaman', function (Blueprint $table) {

            $table->id();

            $table->foreignId('idpeminjaman')->constrained('peminjaman')->onDelete('cascade');

            $table->foreignId('idbuku')->constrained('buku')->onDelete('cascade');

            $table->foreignId('iddenda')->nullable()->constrained('denda')->onDelete('set null');

            $table->enum('status', ['terlambat', 'tidakterlambat'])->default('tidakterlambat');

            $table->date('tanggalpinjam');

            $table->integer('durasipeminjaman')->default(5);

            $table->date('tanggalkembali');

            $table->date('tanggalkembalikan')->nullable();

            $table->integer('jumlahharitelat')->default(0);

            $table->decimal('denda', 10, 2)->default(0);

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*/

    public function down(): void

    {

        Schema::dropIfExists('detilpeminjaman');

        Schema::dropIfExists('peminjaman');

        Schema::dropIfExists('denda');

        Schema::dropIfExists('admin');

        Schema::dropIfExists('peminjam');

        Schema::dropIfExists('buku');

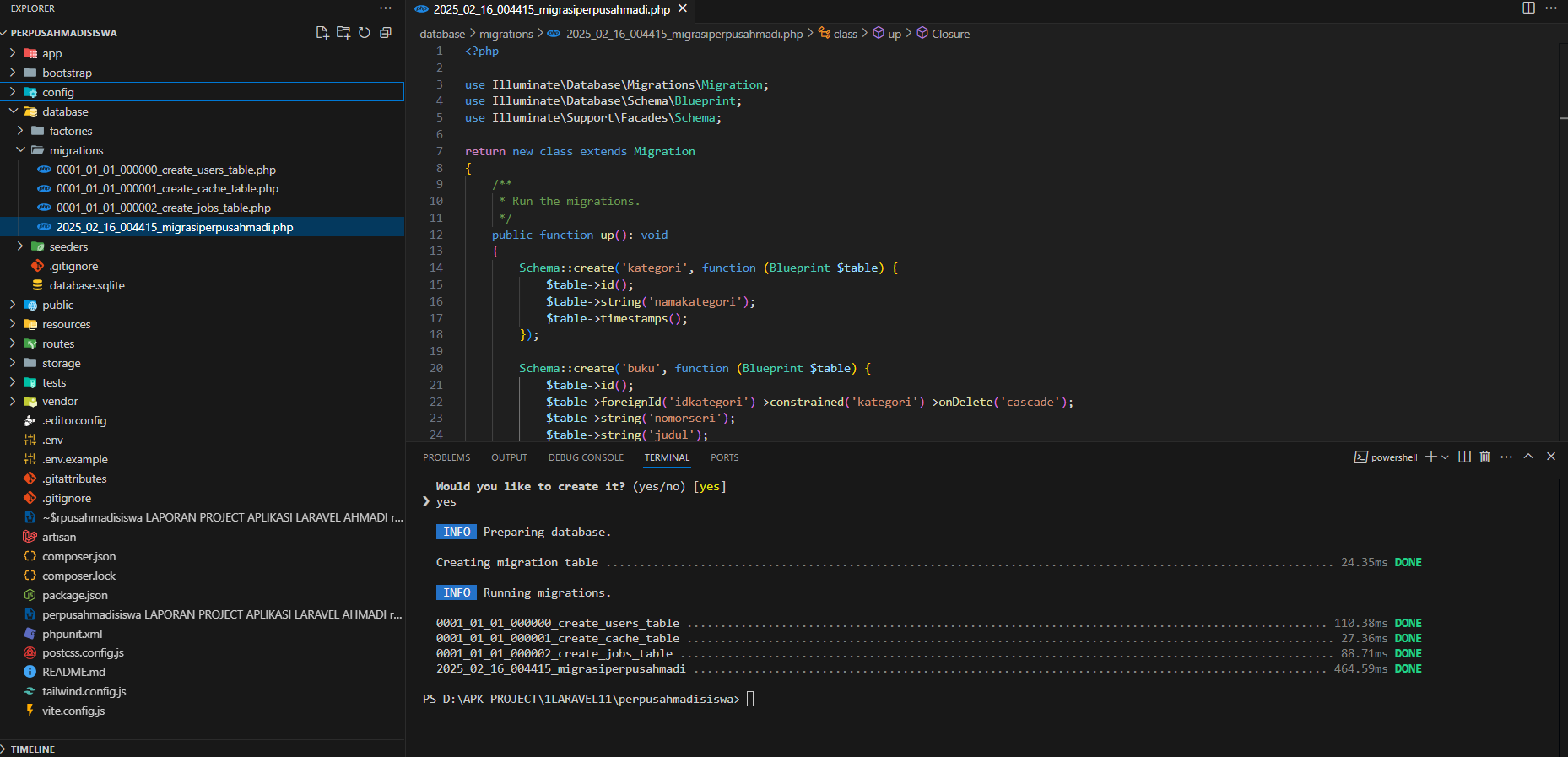
        Schema::dropIfExists('kategori');

    }

};

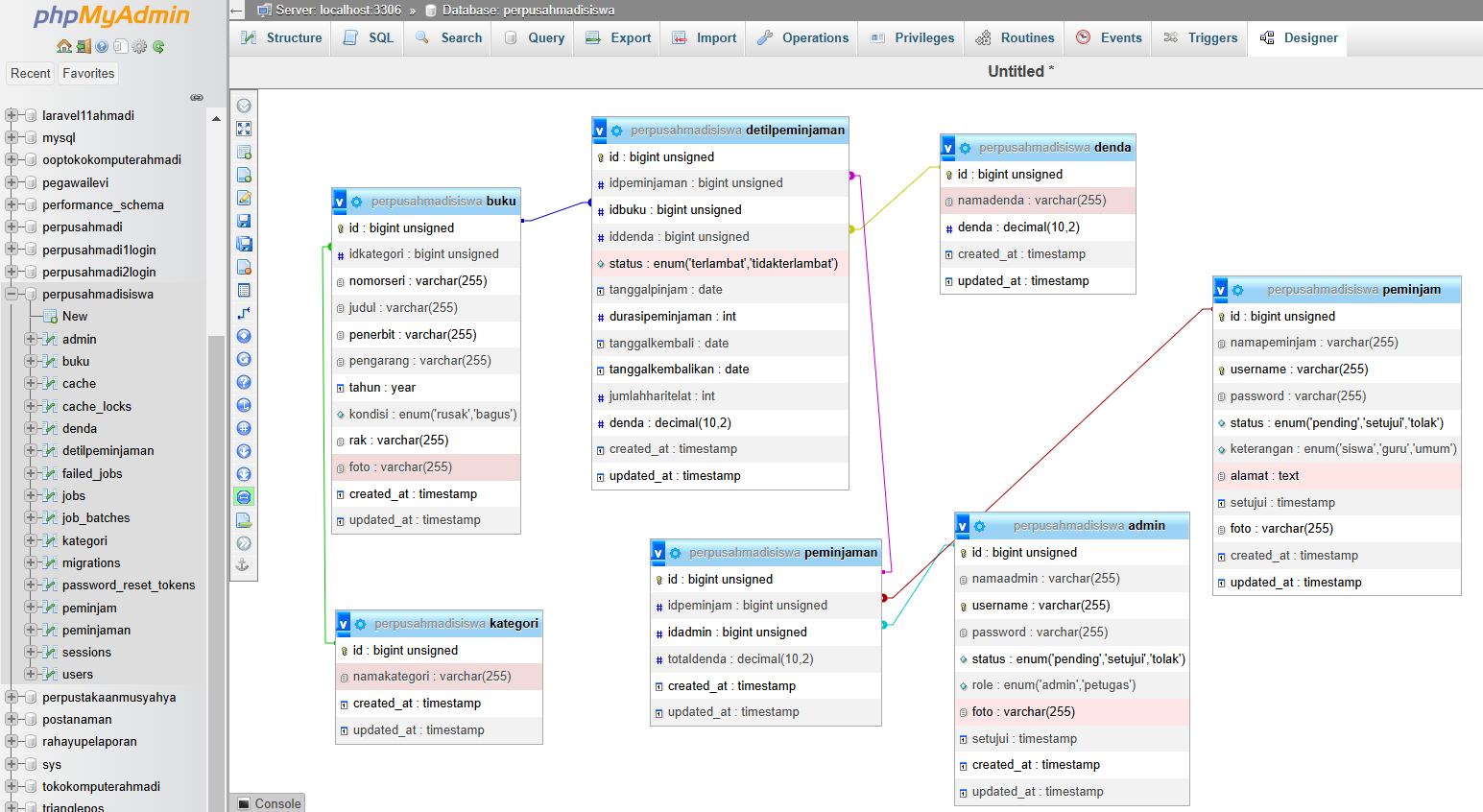
Jalankan perintah berikut di terminal untuk mengunggah migrasi ke database:

php artisan migrate

jika berhasil, akan muncul sbb:  


Cek database perpus di phpmyadmin, dan periksa di bagian design skema database apakah sudah sesuai dengan modeling yang telah dibuat di visio, jika berhasil maka database perpus akan terlihat sebagai berikut:

## Hasil Perancangan Tabel Basis Data



## Seeder Database

1. **Cara Membuat dan Menjalankan Seeder di Laravel**

1. Membuat File Seeder

Untuk mengisi tabel-tabel pada database, kita akan membuat seeder pada tabel tabel berikut:

Admin :

php artisan make:seeder AdminSeeder

Peminjam :

php artisan make:seeder PeminjamSeeder

Kategori :

php artisan make:seeder KategoriSeeder

Buku :

php artisan make:seeder BukuSeeder

Denda :

php artisan make:seeder DendaSeeder

Peminjam :

php artisan make:seeder PeminjamanSeeder

Peminjam :

php artisan make:seeder DetilpeminjamanSeeder

Ini akan membuat file masing-masing tabel di dalam folder database/seeders/.

Admin :

php artisan make:seeder AdminSeeder

php artisan make:seeder PeminjamSeeder

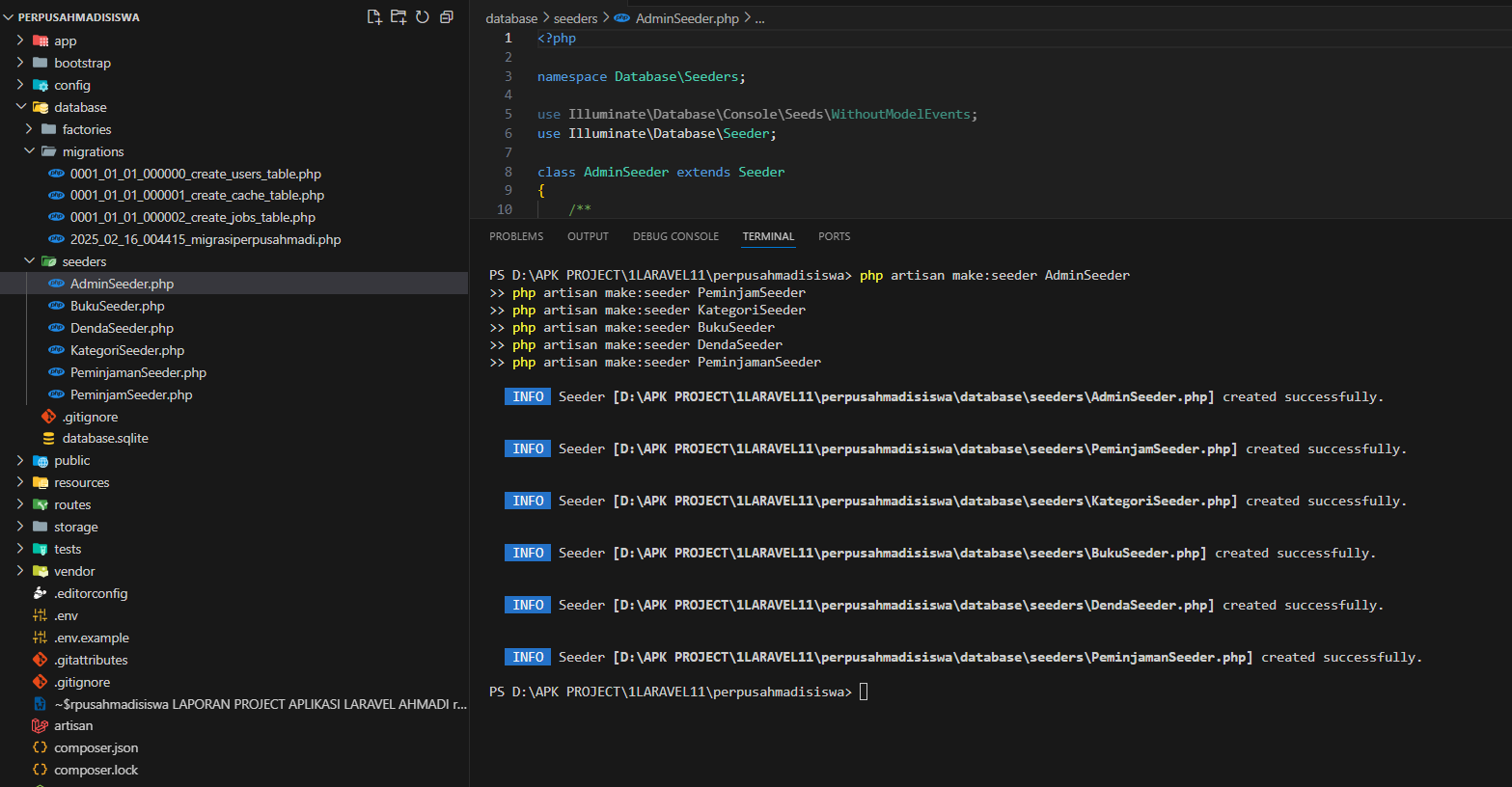
php artisan make:seeder KategoriSeeder

php artisan make:seeder BukuSeeder

php artisan make:seeder DendaSeeder

php artisan make:seeder PeminjamanSeeder

php artisan make:seeder DetilpeminjamanSeeder



2. Edit File Seeder

* AdminSeeder.php

Buka file AdminSeeder.php di dalam folder database/seeders/, lalu tambahkan kode :

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

use Illuminate\Support\Facades\DB;

use Illuminate\Support\Facades\Hash;

class AdminSeeder extends Seeder

{

    public function run(): void

    {

        DB::table('admin')->insert([

            ['namaadmin' => 'Admin1', 'username' => 'admin1', 'password' => Hash::make('admin123'), 'status' => 'setujui', 'role' => 'admin', 'foto' => null, 'setujui' => now()],

            ['namaadmin' => 'Admin2', 'username' => 'admin2', 'password' => Hash::make('admin123'), 'status' => 'setujui', 'role' => 'petugas', 'foto' => null, 'setujui' => now()],

            ['namaadmin' => 'Admin3', 'username' => 'admin3', 'password' => Hash::make('admin123'), 'status' => 'pending', 'role' => 'admin', 'foto' => null, 'setujui' => null],

            ['namaadmin' => 'Admin4', 'username' => 'admin4', 'password' => Hash::make('admin123'), 'status' => 'tolak', 'role' => 'petugas', 'foto' => null, 'setujui' => null],

            ['namaadmin' => 'Admin5', 'username' => 'admin5', 'password' => Hash::make('admin123'), 'status' => 'setujui', 'role' => 'admin', 'foto' => null, 'setujui' => now()],

        ]);

    }

}

* PeminjamSeeder.php

Buka file PeminjamSeeder.php di dalam folder database/seeders/, lalu tambahkan kode :

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

use Illuminate\Support\Facades\DB;

use Illuminate\Support\Facades\Hash;

class PeminjamSeeder extends Seeder

{

    public function run(): void

    {

        DB::table('peminjam')->insert([

            ['namapeminjam' => 'Ali', 'username' => 'ali123', 'password' => Hash::make('password'), 'status' => 'setujui', 'keterangan' => 'siswa', 'alamat' => 'Jakarta', 'setujui' => now(), 'foto' => null],

            ['namapeminjam' => 'Budi', 'username' => 'budi456', 'password' => Hash::make('password'), 'status' => 'setujui', 'keterangan' => 'guru', 'alamat' => 'Bandung', 'setujui' => now(), 'foto' => null],

            ['namapeminjam' => 'Citra', 'username' => 'citra789', 'password' => Hash::make('password'), 'status' => 'pending', 'keterangan' => 'umum', 'alamat' => 'Surabaya', 'setujui' => null, 'foto' => null],

            ['namapeminjam' => 'Dika', 'username' => 'dika101', 'password' => Hash::make('password'), 'status' => 'tolak', 'keterangan' => 'siswa', 'alamat' => 'Yogyakarta', 'setujui' => null, 'foto' => null],

            ['namapeminjam' => 'Eka', 'username' => 'eka202', 'password' => Hash::make('password'), 'status' => 'setujui', 'keterangan' => 'guru', 'alamat' => 'Medan', 'setujui' => now(), 'foto' => null],

        ]);

    }

}

* KategoriSeeder.php

Buka file KategoriSeeder.php di dalam folder database/seeders/, lalu tambahkan kode :

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

use Illuminate\Support\Facades\DB;

class KategoriSeeder extends Seeder

{

    public function run(): void

    {

        DB::table('kategori')->insert([

            ['namakategori' => 'Fiksi'],

            ['namakategori' => 'Non-Fiksi'],

            ['namakategori' => 'Komik'],

            ['namakategori' => 'Ensiklopedia'],

            ['namakategori' => 'Biografi'],

        ]);

    }

}

* BukuSeeder.php

Buka file BukuSeeder.php di dalam folder database/seeders/, lalu tambahkan kode :

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

use Illuminate\Support\Facades\DB;

class BukuSeeder extends Seeder

{

    public function run(): void

    {

        DB::table('buku')->insert([

            ['idkategori' => 1, 'nomorseri' => 'BK001', 'judul' => 'Laskar Pelangi', 'penerbit' => 'Bentang Pustaka', 'pengarang' => 'Andrea Hirata', 'tahun' => 2005, 'kondisi' => 'bagus', 'rak' => 'A1', 'foto' => null],

            ['idkategori' => 2, 'nomorseri' => 'BK002', 'judul' => 'Sapiens', 'penerbit' => 'Gramedia', 'pengarang' => 'Yuval Noah Harari', 'tahun' => 2011, 'kondisi' => 'bagus', 'rak' => 'B1', 'foto' => null],

            ['idkategori' => 3, 'nomorseri' => 'BK003', 'judul' => 'Naruto', 'penerbit' => 'Shonen Jump', 'pengarang' => 'Masashi Kishimoto', 'tahun' => 1999, 'kondisi' => 'bagus', 'rak' => 'C1', 'foto' => null],

            ['idkategori' => 4, 'nomorseri' => 'BK004', 'judul' => 'Ensiklopedia Hewan', 'penerbit' => 'Erlangga', 'pengarang' => 'Tim Penulis', 'tahun' => 2018, 'kondisi' => 'bagus', 'rak' => 'D1', 'foto' => null],

            ['idkategori' => 5, 'nomorseri' => 'BK005', 'judul' => 'Biografi Steve Jobs', 'penerbit' => 'Gramedia', 'pengarang' => 'Walter Isaacson', 'tahun' => 2011, 'kondisi' => 'bagus', 'rak' => 'E1', 'foto' => null],

        ]);

    }

}

* DendaSeeder.php

Buka file DendaSeeder.php di dalam folder database/seeders/, lalu tambahkan kode :

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

use Illuminate\Support\Facades\DB;

class DendaSeeder extends Seeder

{

    public function run(): void

    {

        DB::table('denda')->insert([

            ['namadenda' => 'Terlambat 1-3 hari', 'denda' => 5000.00, 'created\_at' => now(), 'updated\_at' => now()],

            ['namadenda' => 'Terlambat 4-7 hari', 'denda' => 15000.00, 'created\_at' => now(), 'updated\_at' => now()],

            ['namadenda' => 'Terlambat lebih dari 7 hari', 'denda' => 30000.00, 'created\_at' => now(), 'updated\_at' => now()],

            ['namadenda' => 'Buku rusak', 'denda' => 50000.00, 'created\_at' => now(), 'updated\_at' => now()],

            ['namadenda' => 'Buku hilang', 'denda' => 100000.00, 'created\_at' => now(), 'updated\_at' => now()],

        ]);

    }

}

* PeminjamanSeeder.php

Buka file PeminjamanSeeder.php di dalam folder database/seeders/, lalu tambahkan kode :

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

use Illuminate\Support\Facades\DB;

class PeminjamanSeeder extends Seeder

{

    public function run(): void

    {

        DB::table('peminjaman')->insert([

            ['idpeminjam' => 1, 'idadmin' => 1, 'totaldenda' => 0, 'created\_at' => now(), 'updated\_at' => now()],

            ['idpeminjam' => 2, 'idadmin' => 2, 'totaldenda' => 5000, 'created\_at' => now(), 'updated\_at' => now()],

            ['idpeminjam' => 3, 'idadmin' => 1, 'totaldenda' => 0, 'created\_at' => now(), 'updated\_at' => now()],

            ['idpeminjam' => 4, 'idadmin' => 2, 'totaldenda' => 15000, 'created\_at' => now(), 'updated\_at' => now()],

            ['idpeminjam' => 5, 'idadmin' => 1, 'totaldenda' => 0, 'created\_at' => now(), 'updated\_at' => now()],

        ]);

    }

}

* DetilpeminjamanSeeder.php

Buka file DetilpeminjamanSeeder.php di dalam folder database/seeders/, lalu tambahkan kode :

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

use Illuminate\Support\Facades\DB;

class DetilpeminjamanSeeder extends Seeder // Perbaiki nama class

{

    public function run(): void

    {

        DB::table('detilpeminjaman')->insert([

            ['idpeminjaman' => 1, 'idbuku' => 1, 'iddenda' => null, 'status' => 'tidakterlambat', 'tanggalpinjam' => now(), 'durasipeminjaman' => 5, 'tanggalkembali' => now()->addDays(5), 'tanggalkembalikan' => null, 'jumlahharitelat' => 0, 'denda' => 0, 'created\_at' => now(), 'updated\_at' => now()],

            ['idpeminjaman' => 2, 'idbuku' => 2, 'iddenda' => 1, 'status' => 'terlambat', 'tanggalpinjam' => now()->subDays(10), 'durasipeminjaman' => 5, 'tanggalkembali' => now()->subDays(5), 'tanggalkembalikan' => now(), 'jumlahharitelat' => 5, 'denda' => 5000, 'created\_at' => now(), 'updated\_at' => now()],

            ['idpeminjaman' => 3, 'idbuku' => 3, 'iddenda' => null, 'status' => 'tidakterlambat', 'tanggalpinjam' => now(), 'durasipeminjaman' => 5, 'tanggalkembali' => now()->addDays(5), 'tanggalkembalikan' => null, 'jumlahharitelat' => 0, 'denda' => 0, 'created\_at' => now(), 'updated\_at' => now()],

            ['idpeminjaman' => 4, 'idbuku' => 4, 'iddenda' => 2, 'status' => 'terlambat', 'tanggalpinjam' => now()->subDays(15), 'durasipeminjaman' => 5, 'tanggalkembali' => now()->subDays(10), 'tanggalkembalikan' => now(), 'jumlahharitelat' => 10, 'denda' => 15000, 'created\_at' => now(), 'updated\_at' => now()],

            ['idpeminjaman' => 5, 'idbuku' => 5, 'iddenda' => null, 'status' => 'tidakterlambat', 'tanggalpinjam' => now(), 'durasipeminjaman' => 5, 'tanggalkembali' => now()->addDays(5), 'tanggalkembalikan' => null, 'jumlahharitelat' => 0, 'denda' => 0, 'created\_at' => now(), 'updated\_at' => now()],

        ]);

    }

}

3. Jalankan Seeder

Untuk menjalankan seeder, gunakan perintah berikut:

php artisan db:seed

Jika ingin menjalankan AdminSeeder saja, gunakan:

php artisan db:seed --class=AdminSeeder

php artisan db:seed --class=PeminjamSeeder

php artisan db:seed --class=KategoriSeeder

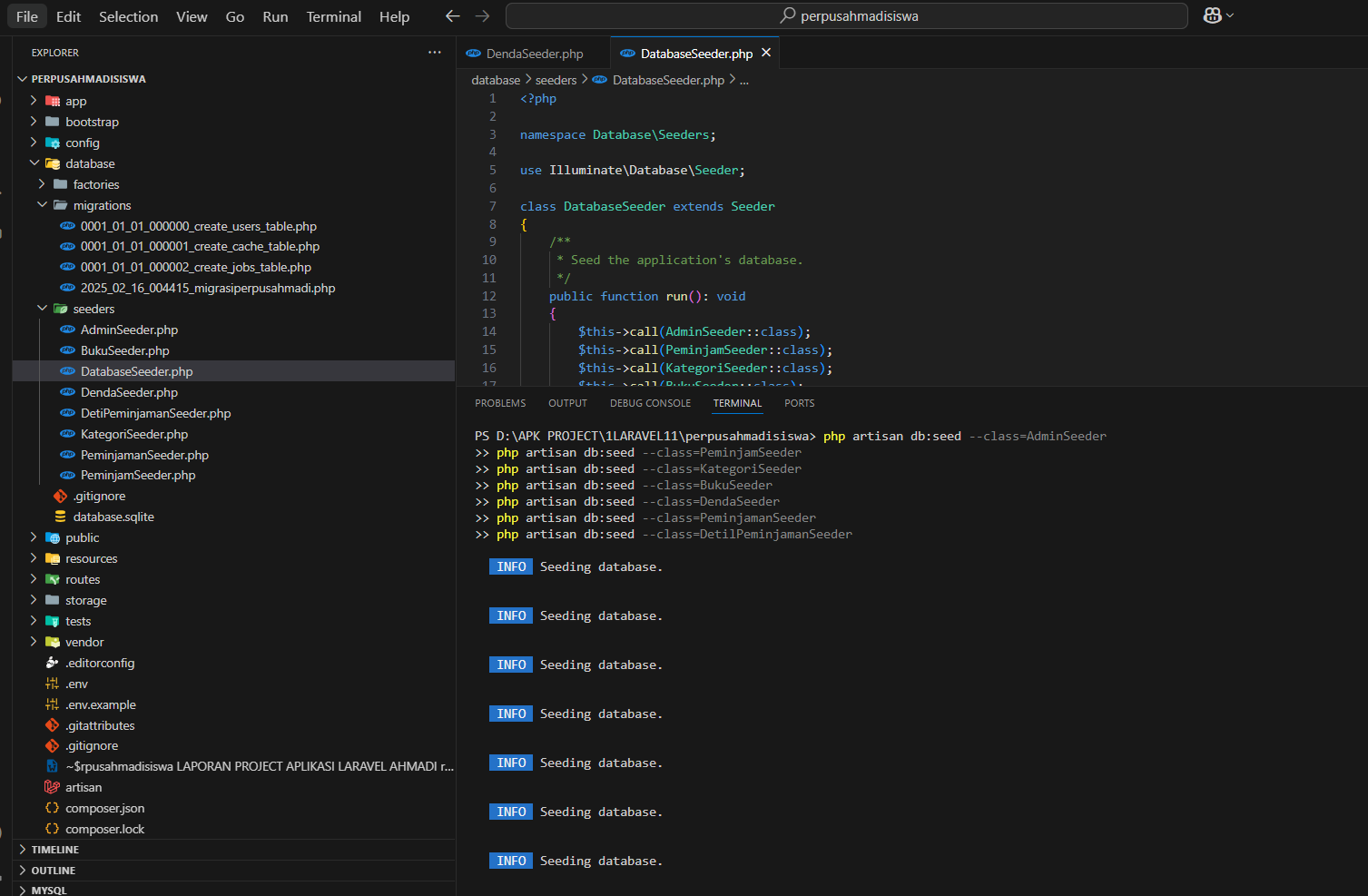
php artisan db:seed --class=BukuSeeder

php artisan db:seed --class=DendaSeeder

php artisan db:seed --class=PeminjamanSeeder

php artisan db:seed --class=DetilPeminjamanSeeder

Dengan cara ini, setiap kali php artisan db:seed dijalankan, data admin akan otomatis dimasukkan ke database sbb:



4. Menjalankan Semua Seeder Sekaligus

Jika ingin menjalankan semua seeder yang terdaftar dalam DatabaseSeeder.php, tambahkan seeder AdminSeeder ke dalamnya:

Buka file DatabaseSeeder.php di dalam folder database/seeders/, lalu ubah menjadi :

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder

{

/\*\*

\* Seed the application's database.

\*/

public function run(): void

{

$this->call(AdminSeeder::class);

$this->call(PeminjamSeeder::class);

$this->call(KategoriSeeder::class);

$this->call(BukuSeeder::class);

$this->call(DendaSeeder::class);

$this->call(PeminjamanSeeder::class);

$this->call(DetilPeminjamanSeeder::class);

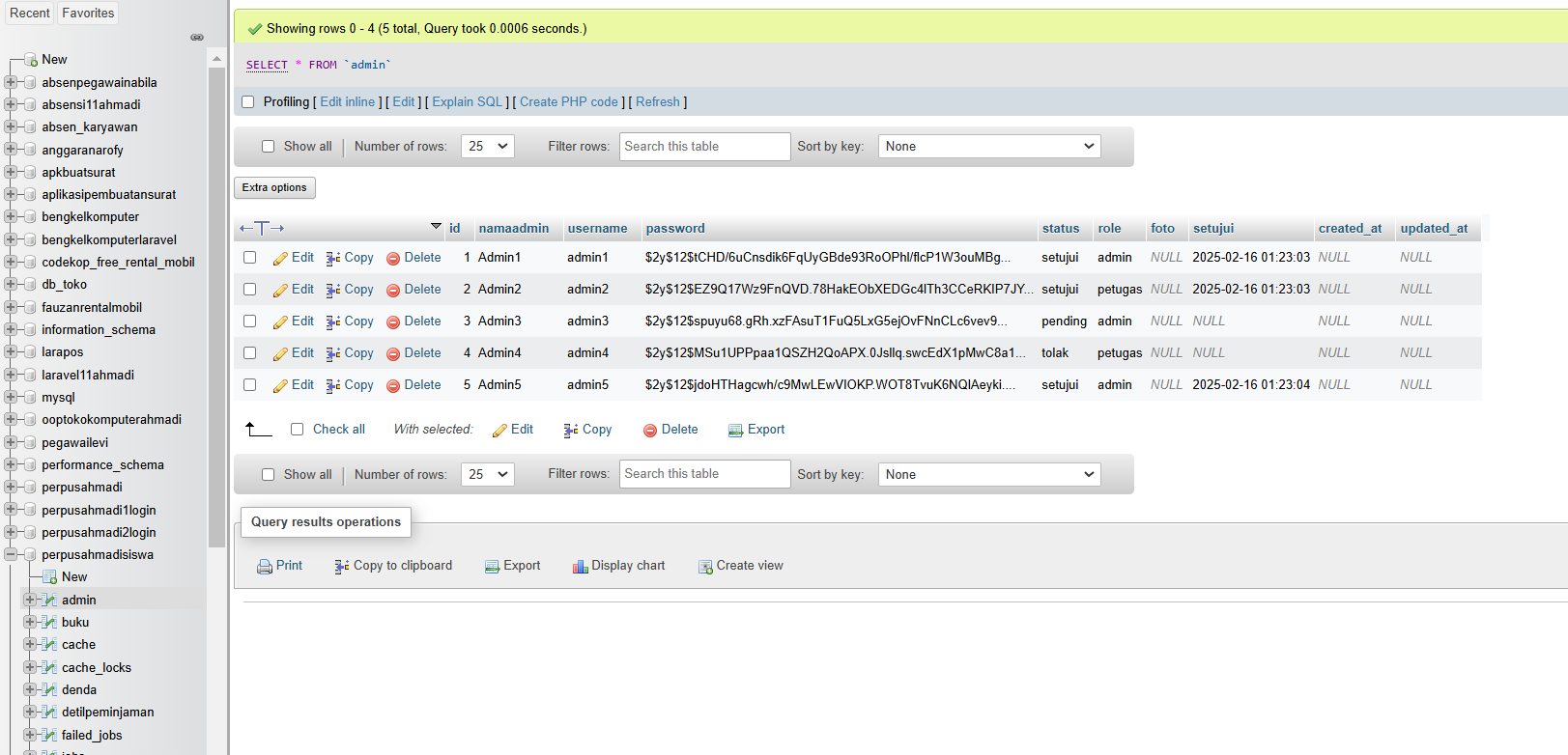
}

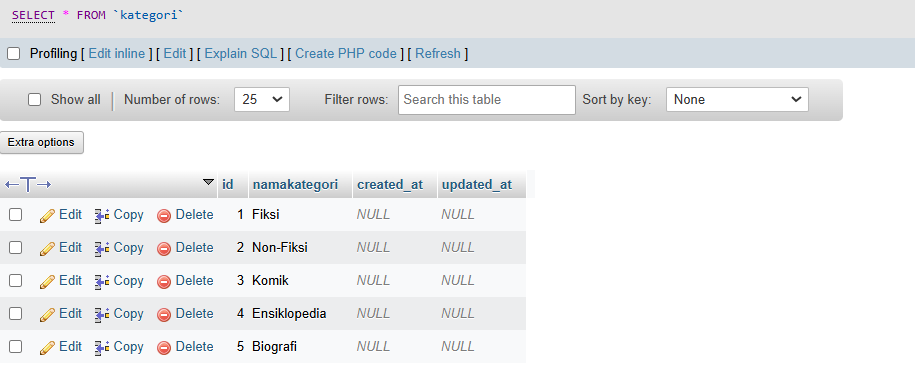
}

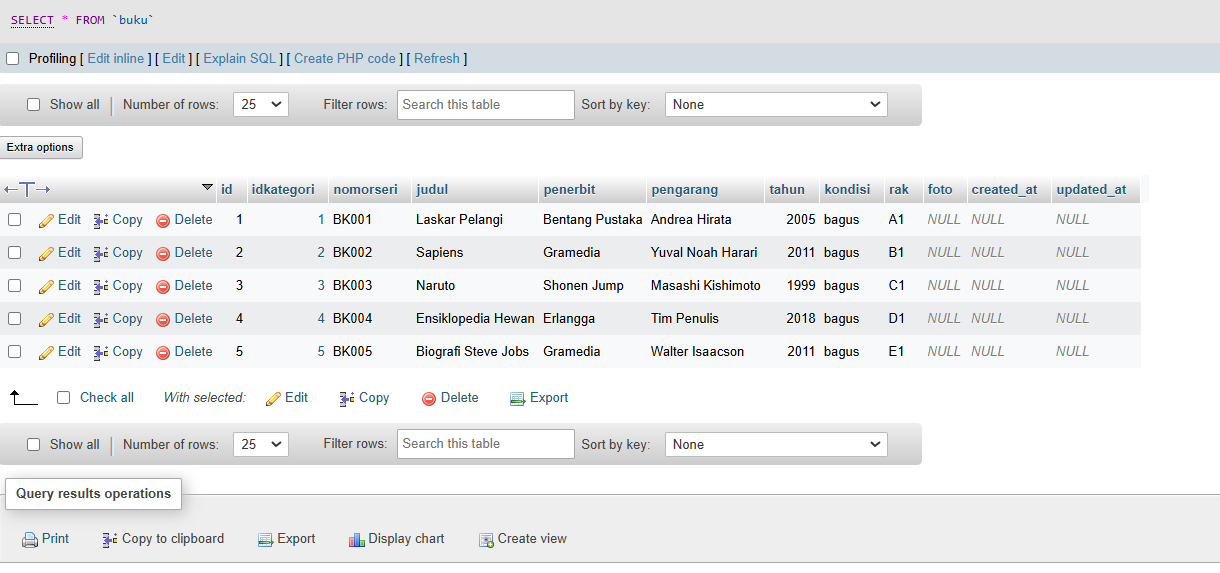
Lalu jalankan:

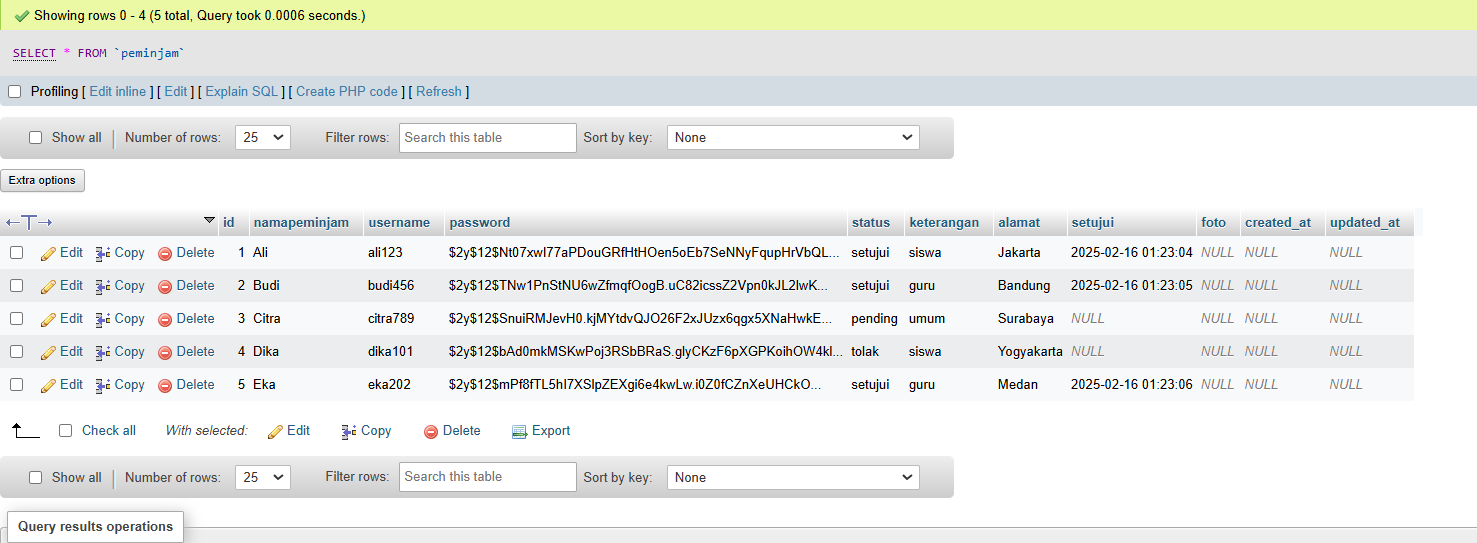
php artisan db:seed

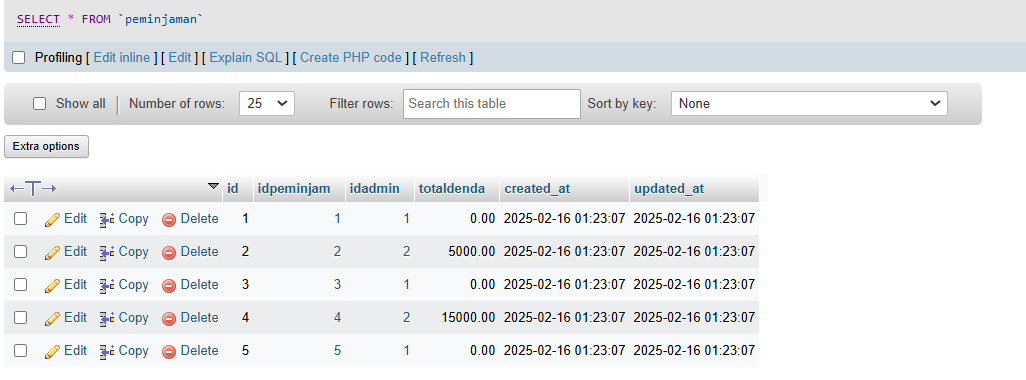
jika berhasil seeder akan mengisi kolom di tabel db kita sbb:

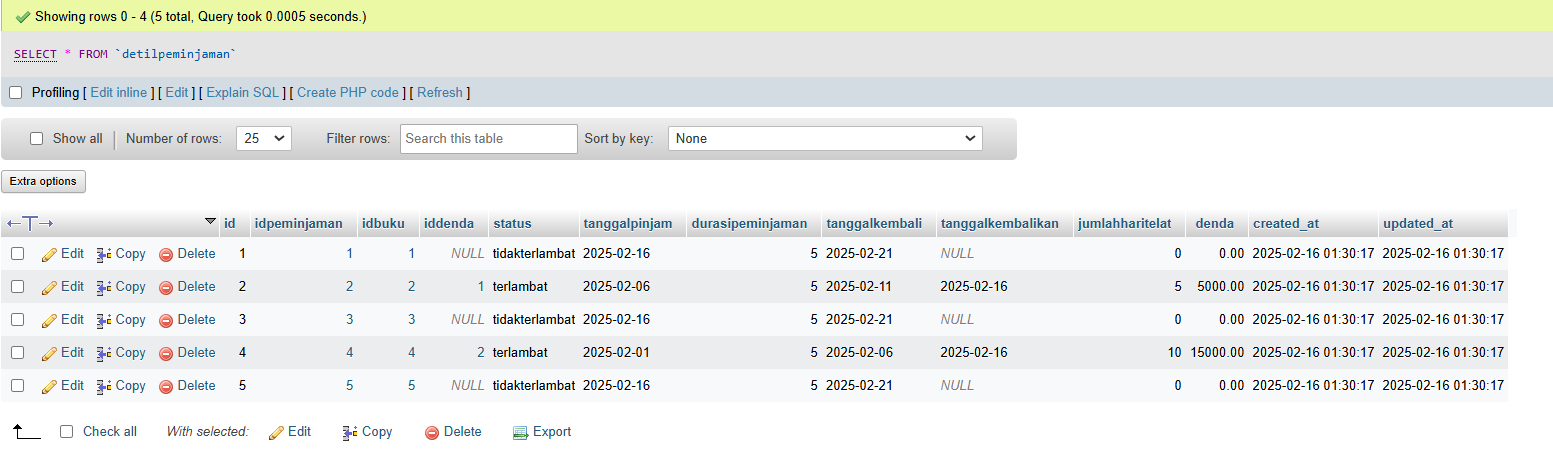












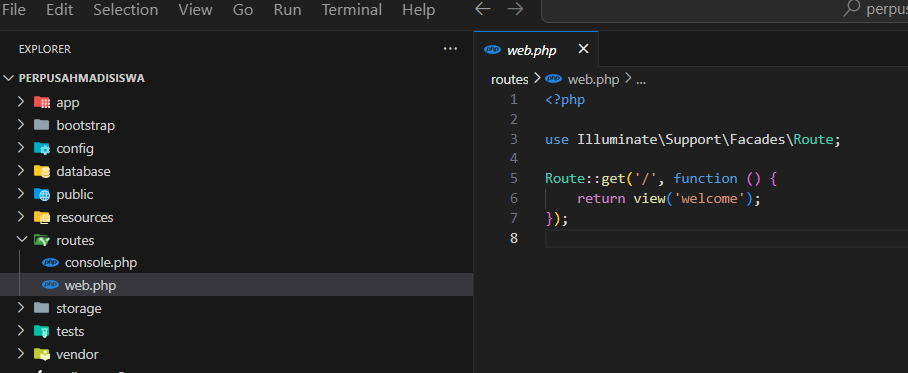
5. Refresh Database dan Jalankan Seeder

Jika ingin menghapus semua data dalam database dan menjalankan migrasi serta seeder sekaligus, jalankan:

php artisan migrate:refresh –seed

Itu akan menghapus semua tabel, menjalankan ulang migrasi, dan menjalankan seeder kembali.

# BAB III PENGEMBANGAN APLIKASI



Kode di atas adalah definisi route dalam Laravel yang mengatur bagaimana aplikasi menangani permintaan ke URL tertentu.

**Penjelasan:**

use Illuminate\Support\Facades\Route;

* Baris ini mengimpor *facade* Route, yang digunakan untuk mendefinisikan route dalam Laravel.

Route::get('/', function () {

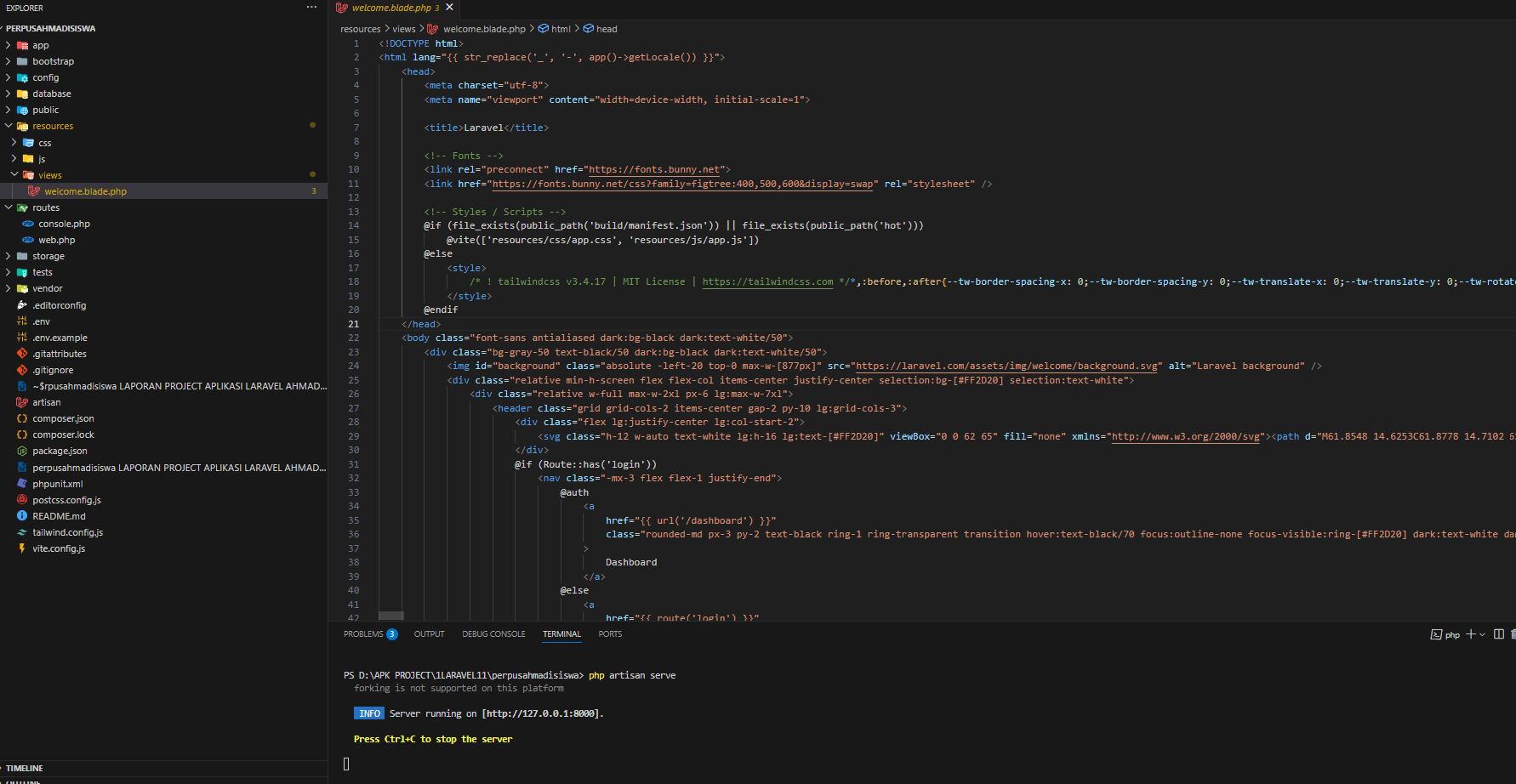
return view('welcome');

});

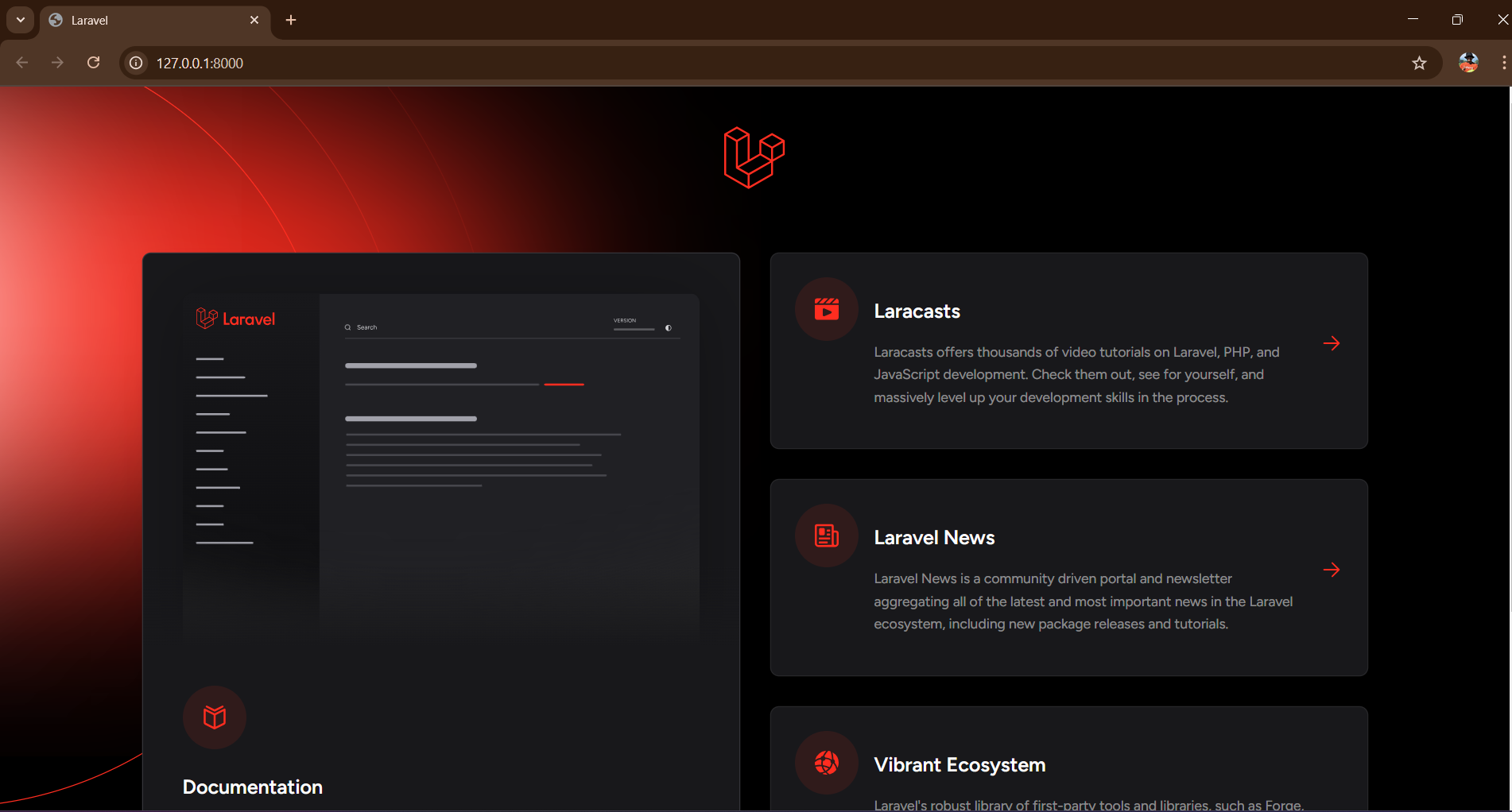
* Route::get('/'): Mendefinisikan route untuk permintaan **HTTP GET** ke URL **"/"** (halaman utama / root).
* function () { return view('welcome'); }: Ketika pengguna mengakses **"/"**, Laravel akan menjalankan fungsi anonim yang mengembalikan tampilan welcome.blade.php.
* view('welcome'): Menunjukkan bahwa file **resources/views/welcome.blade.php** akan dirender sebagai tampilan halaman utama.

Route ini mengatur agar ketika pengguna membuka [**http://yourdomain.com/**](http://yourdomain.com/) (atau **localhost:8000/** saat menggunakan Artisan server), Laravel akan menampilkan halaman **welcome.blade.php**.

Berikut adalah halaman welcomenya:



Dan berikut halaman tampil welcome di browser



Lalu bagaimana jika ingin membuat Login?

Berikut adalah contoh route untuk login:

Route::middleware(['guest'])->group(function () { Route::get('/admin/login', [ControllerLoginAdmin::class, 'showLoginForm'])->name('admin.login'); Route::post('/admin/login', [ControllerLoginAdmin::class, 'login']); Route::get('/admin/register', [ControllerLoginAdmin::class, 'showRegisterForm'])->name('admin.register'); Route::post('/admin/register', [ControllerLoginAdmin::class, 'register']); });

Kode di atas adalah konfigurasi route dalam Laravel yang mengelompokkan rute terkait **login dan registrasi admin**, serta menerapkan middleware **guest** untuk memastikan hanya pengguna yang belum login yang dapat mengaksesnya.

**Penjelasan Kode:**

**1. Middleware guest**

Route::middleware(['guest'])->group(function () { ... });

* Middleware guest memastikan bahwa route di dalamnya hanya dapat diakses oleh pengguna yang **belum login**.
* Jika pengguna sudah login, mereka akan dialihkan (biasanya ke dashboard atau halaman utama).

**2. Route untuk Halaman Login Admin**

Route::get('/admin/login', [ControllerLoginAdmin::class, 'showLoginForm'])->name('admin.login');

* Route ini menangani permintaan **GET** ke /admin/login.
* Memanggil method showLoginForm() dalam ControllerLoginAdmin, yang bertugas menampilkan halaman login admin.
* name('admin.login'): Memberikan nama **admin.login** untuk route ini, sehingga bisa dipanggil dengan route('admin.login') di dalam kode Laravel.

**3. Route untuk Proses Login Admin**

Route::post('/admin/login', [ControllerLoginAdmin::class, 'login']);

* Route ini menangani permintaan **POST** ke /admin/login.
* Memanggil method login() dalam ControllerLoginAdmin, yang menangani proses autentikasi admin berdasarkan input login (email/password).

**4. Route untuk Halaman Registrasi Admin**

Route::get('/admin/register', [ControllerLoginAdmin::class, 'showRegisterForm'])->name('admin.register');

* Route ini menangani permintaan **GET** ke /admin/register.
* Memanggil method showRegisterForm() dalam ControllerLoginAdmin, yang bertugas menampilkan halaman registrasi admin.
* name('admin.register'): Memberikan nama **admin.register** untuk route ini, sehingga bisa dipanggil dengan route('admin.register').

**5. Route untuk Proses Registrasi Admin**

Route::post('/admin/register', [ControllerLoginAdmin::class, 'register']);

* Route ini menangani permintaan **POST** ke /admin/register.
* Memanggil method register() dalam ControllerLoginAdmin, yang menangani proses pendaftaran admin baru.

**Kesimpulan**

1. **Kelompok Route**:
   * Semua route di dalam Route::middleware(['guest'])->group(function () { ... }) hanya dapat diakses oleh pengguna **yang belum login**.
2. **Login Admin**:
   * GET /admin/login: Menampilkan form login.
   * POST /admin/login: Memproses autentikasi admin.
3. **Registrasi Admin**:
   * GET /admin/register: Menampilkan form registrasi admin.
   * POST /admin/register: Memproses pendaftaran admin.
4. **Keamanan**:
   * Middleware guest memastikan hanya pengguna yang belum login yang bisa mengakses halaman login/registrasi.

Selanjutnya kita akan membuat routes untuk menangani pemanggilan fitur kita. Pembuatan fitur ini tentunya dilakukan dengan melihat analisis kebutuhan fitur yang sudah disusun di bab sebelumnya.

## Analisis Fitur

Fitur yang dibutuhkan:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Fitur** | **Pendukung** | **Urgensi/Tingkat Kepentingan** | | | | | |
| **Create** | **Read (tampil)** | **Update** | **Delete** | **Pencarian** | **Cetak** |
| Login | username |  |  |  |  |  |  |
|  | password |  |  |  |  |  |  |
| Dashboard | admin |  |  |  |  |  |  |
|  | pelanggan |  |  |  |  |  |  |
|  | barang |  |  |  |  |  |  |
|  | penjualan |  |  |  |  |  |  |
| Pengguna/Admin | namaadmin |  |  |  |  |  |  |
|  | username |  |  |  |  |  |  |
|  | password |  |  |  |  |  |  |
|  | alamat |  |  |  |  |  |  |
| peminjam | namapelanggan |  |  |  |  |  |  |
|  | username |  |  |  |  |  |  |
|  | password |  |  |  |  |  |  |
|  | alamat |  |  |  |  |  |  |
| buku | namabarang |  |  |  |  |  |  |
|  | kategori |  |  |  |  |  |  |
|  | merk |  |  |  |  |  |  |
|  | suplier |  |  |  |  |  |  |
|  | stok |  |  |  |  |  |  |
|  | harga |  |  |  |  |  |  |
| Barang->Kategori | namakategori |  |  |  |  |  |  |
| Barang->Suplier | namakategori |  |  |  |  |  |  |
|  | alamat |  |  |  |  |  |  |
| Transaksi | nomor |  |  |  |  |  |  |
|  | namapelanggan |  |  |  |  |  |  |
|  | namaadmin |  |  |  |  |  |  |
|  | namabarang |  |  |  |  |  |  |
|  | jumlahbarang |  |  |  |  |  |  |
|  | total |  |  |  |  |  |  |
| Rekap/laporan | rekap-harian |  |  |  |  |  |  |
| Cetak | cetak |  |  |  |  |  |  |
| **Total** | | |  |  |  |  |  |

## Membuat Layout Blade:

Pembuatan layouts blade dapat dilakukan dengna 2 cara, yaitu secara manual dan dengan terminal.

Namun jangan lupa, karena kita akan membuat aplikasi dengan 2 level user, maka kita perlu membuat 2 jenis layout blade, yaitu untuk level admin dan peminjam.

* Level admin layout/appadmin.blade.php
* Level peminjam layout/apppeminjam.blade.php
* **Layout blade admin**

Gunakan Perintah untuk membuat folder sbb:

New-Item -ItemType File resources/views/layouts

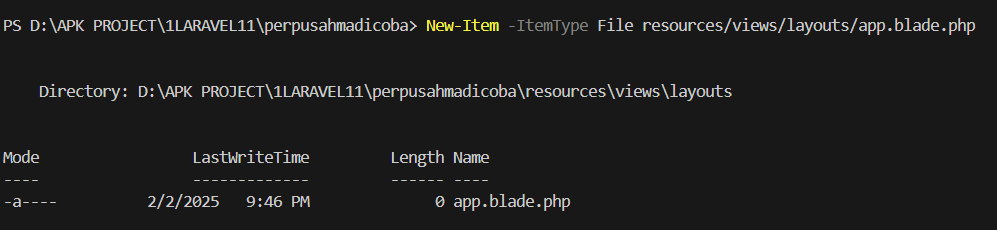
Gunakan Perintah yang Sesuai di Terminal Windows sbb:

New-Item -ItemType File resources/views/layouts/appadmin.blade.php

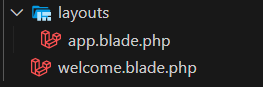
Gunakan Command Prompt (CMD):

echo. > resources/views/layouts/appadmin.blade.php

jika berhasil akan tampil sebagai berikut:



Atau jika gagal buat saja dengan cara manual, dan hasil akhirnya adalah sbb:



Ada 2 cara untuk menambahkan script di layout/app.blade.php, yaitu dengan CDN dan ASET

**Dengan CDN**

<!DOCTYPE html>

<html lang="id">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Admin Panel</title>

<!-- AdminLTE CSS via CDN -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/admin-lte/3.2.0/css/adminlte.min.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.3.0/css/bootstrap.min.css">

</head>

<body class="hold-transition sidebar-mini layout-fixed">

<div class="wrapper">

<!-- Navbar -->

<nav class="main-header navbar navbar-expand navbar-white navbar-light">

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" data-widget="pushmenu" href="#" role="button"><i class="fas fa-bars"></i></a>

</li>

</ul>

</nav>

<!-- Sidebar -->

<aside class="main-sidebar sidebar-dark-primary elevation-4">

<a href="#" class="brand-link">

<span class="brand-text font-weight-light">AdminLTE</span>

</a>

<div class="sidebar">

<nav class="mt-2">

<ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu">

<li class="nav-item">

<a href="{{ route('admin.index') }}" class="nav-link">

<i class="nav-icon fas fa-users"></i>

<p>Kelola Admin</p>

</a>

</li>

</ul>

</nav>

</div>

</aside>

<!-- Content Wrapper -->

<div class="content-wrapper">

<div class="content-header">

<div class="container-fluid">

<div class="row mb-2">

<div class="col-sm-6">

<h1 class="m-0">@yield('title')</h1>

</div>

</div>

</div>

</div>

<section class="content">

<div class="container-fluid">

@yield('content')

</div>

</section>

</div>

<!-- Footer -->

<footer class="main-footer">

<strong>Copyright &copy; 2024 <a href="#">AdminLTE</a>.</strong>

All rights reserved.

</footer>

</div>

<!-- AdminLTE & Bootstrap JS -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.3.0/js/bootstrap.bundle.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/admin-lte/3.2.0/js/adminlte.min.js"></script>

</body>

</html>

## Dengan ASSET

**1. Download AdminLTE**

1. **Download AdminLTE 3** dari GitHub:

[**https://github.com/ColorlibHQ/AdminLTE/releases**](https://github.com/ColorlibHQ/AdminLTE/releases)

1. Pilih **AdminLTE v3.2.0** (ZIP) dan ekstrak.
2. Pindahkan folder **dist** dan **plugins** ke dalam **public/** proyek Laravel.

**2. Struktur Folder Laravel Setelah Penyesuaian**

/public

/dist

/css

adminlte.min.css

/js

adminlte.min.js

/plugins

/fontawesome-free

/jquery

/bootstrap

**3. Update layouts/app.blade.php**

Pastekan kode berikut:

<!DOCTYPE html>

<html lang="id">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Admin Panel</title>

    <!-- AdminLTE & Bootstrap CSS -->

    <link rel="stylesheet" href="{{ asset('dist/css/adminlte.min.css') }}">

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

    <link rel="stylesheet" href="{{ asset('plugins/bootstrap/css/bootstrap.min.css') }}">

    <!-- SWEETALERT -->

    <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>

    <!-- DataTables -->

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

    <link rel="stylesheet" href="{{ asset('plugins/datatables-responsive/css/responsive.bootstrap4.min.css') }}">

    <link rel="stylesheet" href="{{ asset('plugins/datatables-buttons/css/buttons.bootstrap4.min.css') }}">

</head>

<body class="hold-transition sidebar-mini layout-fixed">

    <div class="wrapper">

        <!-- Navbar -->

        <nav class="main-header navbar navbar-expand navbar-white navbar-light">

            <ul class="navbar-nav">

                <li class="nav-item">

                    <a class="nav-link" data-widget="pushmenu" href="#" role="button"><i class="fas fa-bars"></i></a>

                </li>

            </ul>

        </nav>

        <!-- Sidebar -->

        <aside class="main-sidebar sidebar-dark-primary elevation-4">

            <a href="#" class="brand-link">

                <span class="brand-text font-weight-light">AdminLTE</span>

            </a>

            <div class="sidebar">

                <nav class="mt-2">

                    <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu">

                        <li class="nav-item">

                            <a href="{{ route('admin.index') }}" class="nav-link">

                                <i class="nav-icon fas fa-users"></i>

                                <p>Kelola Admin</p>

                            </a>

                        </li>

                    </ul>

                </nav>

            </div>

        </aside>

        <!-- Content Wrapper -->

        <div class="content-wrapper mt-3">

            <div class="content-header">

                <div class="container-fluid">

                    <div class="row mb-2">

                        <div class="col-sm-6">

                            <h1 class="m-0">@yield('title')</h1>

                        </div>

                    </div>

                </div>

            </div>

            <section class="content">

                <div class="container-fluid">

                    <div class="card">

                        <div class="card-body">

                            @yield('content')

                        </div>

                    </div>

                </div>

            </section>

        </div>

        <!-- Footer -->

        <footer class="main-footer">

            <strong>Copyright &copy; 2024 <a href="#">AdminLTE</a>.</strong>

            All rights reserved.

        </footer>

    </div>

<!-- Scripts -->

<script src="{{ asset('plugins/jquery/jquery.min.js') }}"></script>

    <script src="{{ asset('plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>

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

        $(document).ready(function() {

            $('body').addClass('sidebar-collapse'); // Collapse sidebar jika masih aktif

            $('.main-sidebar').remove(); // Hapus elemen sidebar jika ada

            $('.content-wrapper').css('margin-left', '0'); // Pastikan tidak ada margin kiri

        });

    </script>

    <!-- DataTables & Plugins -->

    <script src="{{ asset('plugins/datatables/jquery.dataTables.min.js') }}"></script>

    <script src="{{ asset('plugins/datatables-bs4/js/dataTables.bootstrap4.min.js') }}"></script>

    <script src="{{ asset('plugins/datatables-responsive/js/dataTables.responsive.min.js') }}"></script>

    <script src="{{ asset('plugins/datatables-responsive/js/responsive.bootstrap4.min.js') }}"></script>

    <script src="{{ asset('plugins/datatables-buttons/js/dataTables.buttons.min.js') }}"></script>

    <script src="{{ asset('plugins/datatables-buttons/js/buttons.bootstrap4.min.js') }}"></script>

    <script src="{{ asset('plugins/datatables-buttons/js/buttons.html5.min.js') }}"></script>

    <script src="{{ asset('plugins/datatables-buttons/js/buttons.print.min.js') }}"></script>

    <script src="{{ asset('plugins/datatables-buttons/js/buttons.colVis.min.js') }}"></script>

    <!-- DataTables & Plugins Bootstrap 5-->

    <script src="{{ asset('plugins/datatables-bs5/js/dataTables.bootstrap5.min.js') }}"></script>

    <script src="{{ asset('plugins/datatables-responsive/js/responsive.bootstrap5.min.js') }}"></script>

    <script src="{{ asset('plugins/datatables-buttons/js/buttons.bootstrap5.min.js') }}"></script>

    <!-- Tambahkan script SweetAlert -->

    <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>

    <script>

        document.addEventListener("DOMContentLoaded", function() {

            document.querySelectorAll('.delete-btn').forEach(button => {

                button.addEventListener('click', function(event) {

                    event.preventDefault();

                    let form = this.closest('form');

                    Swal.fire({

                        title: "Apakah Anda yakin?",

                        text: "Data akan dihapus secara permanen!",

                        icon: "warning",

                        showCancelButton: true,

                        confirmButtonColor: "#d33",

                        cancelButtonColor: "#3085d6",

                        confirmButtonText: "Ya, Hapus!",

                        cancelButtonText: "Batal"

                    }).then((result) => {

                        if (result.isConfirmed) {

                            form.submit(); // Jika dikonfirmasi, form akan dikirim

                        }

                    });

                });

            });

        });

    </script>

    <!-- Tambahkan script Dropdown Navbar -->

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

    <script>

        $(document).ready(function() {

            $('.dropdown-submenu a.dropdown-toggle').on("click", function(e) {

                if (!$(this).next('div').hasClass('show')) {

                    $(this).parents('.dropdown-menu').first().find('.show').removeClass("show");

                }

                var $subMenu = $(this).next("div");

                $subMenu.toggleClass('show');

                $(this).parents('li.nav-item.dropdown.show').on('hidden.bs.dropdown', function() {

                    $('.dropdown-submenu .dropdown-menu').removeClass("show");

                });

                return false;

            });

        });

    </script>

    <!-- Tambahkan script Data Table -->

    <script>

        $(document).ready(function() {

            let table = $('#example1').DataTable({

                "responsive": true,

                "lengthChange": true,

                "autoWidth": false,

                "paging": true,

                "ordering": true,

                "info": true,

                "searching": true,

                "buttons": ["copy", "csv", "excel", "pdf", "print", "colvis"], // Tombol Export

                "language": {

                    "lengthMenu": "Tampilkan \_MENU\_ entri",

                    "zeroRecords": "Data tidak ditemukan",

                    "info": "Menampilkan \_START\_ sampai \_END\_ dari \_TOTAL\_ data",

                    "infoEmpty": "Tidak ada data tersedia",

                    "search": "Cari:",

                    "paginate": {

                        "previous": "Sebelumnya",

                        "next": "Selanjutnya"

                    }

                }

            });

            // Tempatkan tombol di dalam container yang benar untuk Bootstrap 5

            table.buttons().container().appendTo('#example1\_wrapper .dt-buttons');

        });

    </script>

</body>

</html>

Kode di atas adalah bagian dari **template Blade** dalam Laravel yang digunakan untuk menampilkan konten halaman dalam layout utama.

**Penjelasan Kode**

<section class="content">

<div class="container-fluid">

@yield('content')

</div>

</section>

1. **<section class="content">**
   * Elemen <section> dengan class "content" digunakan sebagai pembungkus utama konten halaman.
   * Biasanya digunakan dalam template AdminLTE atau layout utama lainnya.
2. **<div class="container-fluid">**
   * **container-fluid** adalah class dari Bootstrap yang membuat elemen memiliki lebar penuh.
   * Memastikan bahwa konten dalam section tidak memiliki padding tetap, sehingga responsif di berbagai ukuran layar.
3. **@yield('content')**
   * **@yield('content')** adalah direktif Blade yang digunakan untuk menentukan **tempat** di mana konten dari halaman anak (child view) akan dimasukkan.
   * Ketika halaman yang menggunakan layout ini dirender, bagian ini akan **digantikan** oleh konten yang didefinisikan dalam @section('content') di dalam view anak.

**Contoh Penggunaan dalam Layout**

**1. Template Utama (layouts/app.blade.php)**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>@yield('title')</title>

<link rel="stylesheet" href="path/to/adminlte.css">

</head>

<body>

<div class="wrapper">

@include('partials.navbar') {{-- Navbar --}}

@include('partials.sidebar') {{-- Sidebar --}}

<div class="content-wrapper">

<section class="content">

<div class="container-fluid">

@yield('content') {{-- Tempat untuk konten halaman --}}

</div>

</section>

</div>

@include('partials.footer') {{-- Footer --}}

</div>

</body>

</html>

**2. View Anak (dashboard.blade.php)**

@extends('layouts.app')

@section('title', 'Dashboard Admin')

@section('content')

<div class="card">

<div class="card-header">

<h3>Dashboard</h3>

</div>

<div class="card-body">

<p>Selamat datang di dashboard admin!</p>

</div>

</div>

@endsection

**Bagaimana Cara Kerjanya?**

1. Saat **dashboard.blade.php** dirender:
   * @extends('layouts.app') menunjukkan bahwa view ini menggunakan **layouts/app.blade.php** sebagai layout utama.
   * @section('content') akan **mengisi** bagian @yield('content') dalam layout utama.
2. Hasil yang dirender:

<section class="content">

<div class="container-fluid">

<div class="card">

<div class="card-header">

<h3>Dashboard</h3>

</div>

<div class="card-body">

<p>Selamat datang di dashboard admin!</p>

</div>

</div>

</div>

</section>

**Kesimpulan**

* **@yield('content')** berfungsi sebagai **tempat konten** dalam layout utama.
* **View anak (child view)** menggunakan @section('content') untuk memasukkan kontennya ke dalam layout utama.
* **AdminLTE + Bootstrap** digunakan untuk styling agar tampilan lebih rapi dan responsif.

Dengan pendekatan ini, Laravel bisa **memisahkan struktur utama dan konten halaman**, sehingga lebih mudah dikelola dan digunakan kembali.

* **Layout blade peminjam**

Gunakan Perintah untuk membuat folder sbb:

New-Item -ItemType File resources/views/layouts

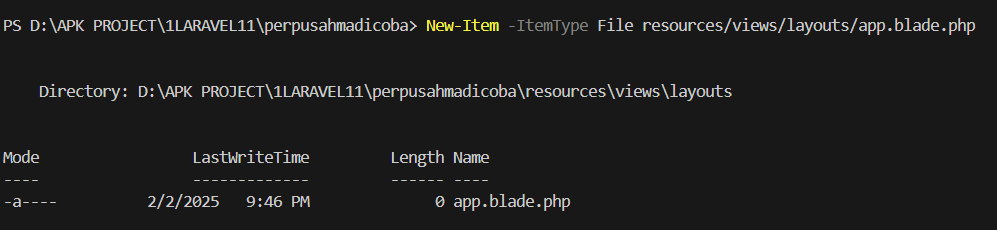
Gunakan Perintah yang Sesuai di Terminal Windows sbb:

New-Item -ItemType File resources/views/layouts/apppeminjam.blade.php

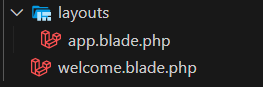
Gunakan Command Prompt (CMD):

echo. > resources/views/layouts/apppeminjam.blade.php

jika berhasil akan tampil sebagai berikut:



Atau jika gagal buat saja dengan cara manual, dan hasil akhirnya adalah sbb:



1. **Membuat Fitur Admin**:

Untuk membuat fitur admin yang menerapkan prinsip MODEL, VIEW, CONTROLLER (MVC), ketiganya harus dibuat berurutan dan bersamaan Namun sebelumnya kita harus Atur routingnya

**Mengatur Routing ADMIN di route/web.php**

use App\Http\Controllers\ControllerAdmin;

Route::middleware(['auth:admin'])->group(function () {

Route::get('/admin/dashboard', [ControllerAdmin::class, 'dashboard'])->name('admin.dashboard');

Route::resource('/admin', ControllerAdmin::class);

Route::post('/admin/{id}/approve', [ControllerAdmin::class, 'approve'])->name('admin.approve');

Route::post('/admin/{id}/reject', [ControllerAdmin::class, 'reject'])->name('admin.reject');

});

Nah, dari route yang telah dibuat: **Cara Mengakses Halaman Admin di Browser**

| **Halaman** | **URL** | **Keterangan** |
| --- | --- | --- |
| **Index (Daftar Admin)** | http://your-domain.com/admin | Menampilkan daftar admin (Admin Index). |
| **Create (Tambah Admin)** | http://your-domain.com/admin/create | Menampilkan form tambah admin baru. |
| **Edit (Ubah Admin)** | http://your-domain.com/admin/{id}/edit | Mengedit data admin dengan ID tertentu. |
| **Show (Detail Admin)** | http://your-domain.com/admin/{id} | Menampilkan detail admin dengan ID tertentu. |

**Contoh Penggunaan**

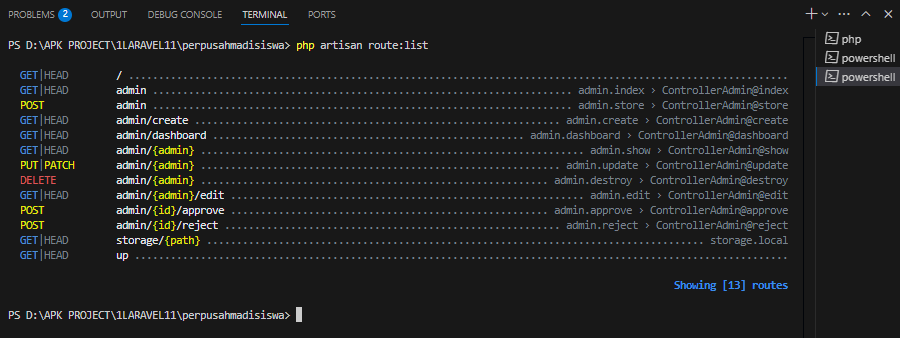
1. **Menampilkan Daftar Admin**
   * Buka di browser: http://localhost:8000/admin
   * Menampilkan daftar admin yang sudah terdaftar.
2. **Menambahkan Admin Baru**
   * Buka: http://localhost:8000/admin/create
   * Mengisi form untuk menambahkan admin baru.
3. **Mengedit Admin**
   * Buka: http://localhost:8000/admin/1/edit *(gantilah 1 dengan ID admin yang ingin diedit)*
   * Menampilkan form edit admin.
4. **Menampilkan Detail Admin**
   * Buka: http://localhost:8000/admin/1 *(gantilah 1 dengan ID admin yang ingin dilihat detailnya)*
   * Menampilkan informasi lengkap admin.

**Catatan:**

* **Pastikan login sebagai admin** sebelum mengakses route ini.
* Jika Laravel berjalan di **localhost**, gunakan http://localhost:8000/admin
* Jika di **hosting**, ganti your-domain.com dengan domain atau IP server Anda.

Dan jika dilihat di route list maka hasilnya akan sbb:

php artisan route:list



Namun sebelumnya, kita harus membuat Models, Views, dan Controller Admin terlebih dahulu.

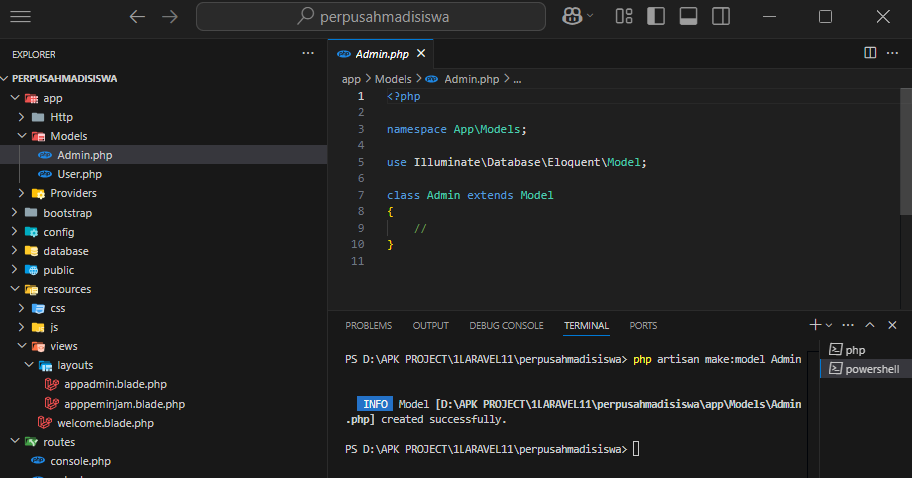
**Membuat Model dengan Migration**

Jalankan perintah berikut untuk membuat model Admin beserta migrasinya:

php artisan make:model Admin

Mengecek File yang Dibuat Model Admin:

app/Models/Admin.php



dan pastekan kode berikut:

<?php

namespace App\Models;

use Illuminate\Foundation\Auth\User as Authenticatable;

use Illuminate\Notifications\Notifiable;

use Illuminate\Database\Eloquent\Factories\HasFactory;

class Admin extends Authenticatable

{

use HasFactory, Notifiable;

protected $table = 'admin';

protected $fillable = [

'namaadmin',

'username',

'password',

'status',

'role',

'foto',

'setujui'

];

protected $hidden = [

'password',

];

protected $casts = [

'setujui' => 'datetime',

];

}

**Membuat Controller**

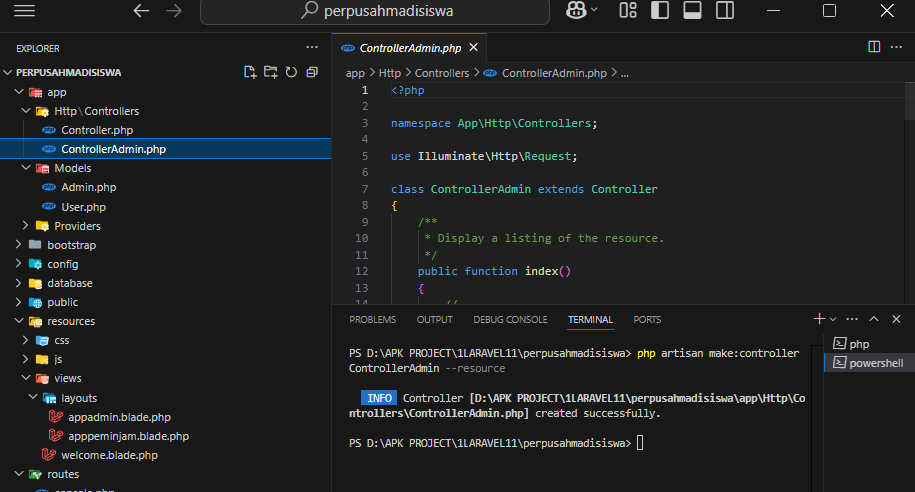
Gunakan perintah berikut untuk membuat controller AdminController dengan resource (CRUD):

php artisan make:controller ControllerAdmin --resource

**resource** akan membuat metode CRUD (index, create, store, show, edit, update, destroy) otomatis dalam controller.

Mengecek File yang Dibuat Controller AdminController:

app/Http/Controllers/ControllerAdmin.php



dan pastekan kode berikut:

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Hash;

use Illuminate\Support\Facades\Auth;

use App\Models\Admin;

use Illuminate\Support\Facades\Storage;

class ControllerAdmin extends Controller

{

    // Menampilkan halaman login

    public function showLoginForm()

    {

        return view('admin.login');

    }

    // Proses login

    public function login(Request $request)

    {

        $request->validate([

            'username' => 'required',

            'password' => 'required',

        ]);

        if (Auth::guard('admin')->attempt(['username' => $request->username, 'password' => $request->password])) {

            return redirect()->route('admin.dashboard');

        }

        return back()->with('error', 'Username atau password salah.');

    }

    // Menampilkan halaman register

    public function showRegisterForm()

    {

        return view('admin.register');

    }

    // Proses registrasi admin baru

    public function register(Request $request)

    {

        $request->validate([

            'namaadmin' => 'required|string|max:255',

            'username' => 'required|string|unique:admin',

            'password' => 'required|min:6|confirmed',

            'role' => 'required|in:admin,petugas',

            'foto' => 'nullable|image|max:2048',

        ]);

        $fotoPath = null;

        if ($request->hasFile('foto')) {

            $fotoPath = $request->file('foto')->store('admin\_fotos', 'public');

        }

        Admin::create([

            'namaadmin' => $request->namaadmin,

            'username' => $request->username,

            'password' => Hash::make($request->password),

            'role' => $request->role,

            'status' => 'pending',

            'foto' => $fotoPath,

        ]);

        return redirect()->route('admin.login')->with('success', 'Registrasi berhasil, silakan login.');

    }

    // Logout

    public function logout()

    {

        Auth::guard('admin')->logout();

        return redirect()->route('admin.login');

    }

    // Menampilkan daftar admin

    public function index()

    {

        $admins = Admin::all();

        return view('admin.index', compact('admins'));

    }

    // Menampilkan halaman tambah admin

    public function create()

    {

        return view('admin.create');

    }

    // Proses penyimpanan admin baru

    public function store(Request $request)

    {

        $request->validate([

            'namaadmin' => 'required|string|max:255',

            'username' => 'required|string|unique:admin',

            'password' => 'required|min:6|confirmed',

            'role' => 'required|in:admin,petugas',

            'foto' => 'nullable|image|max:2048',

        ]);

        $fotoPath = $request->file('foto') ? $request->file('foto')->store('admin\_fotos', 'public') : null;

        Admin::create([

            'namaadmin' => $request->namaadmin,

            'username' => $request->username,

            'password' => Hash::make($request->password),

            'role' => $request->role,

            'status' => 'pending',

            'foto' => $fotoPath,

        ]);

        return redirect()->route('admin.index')->with('success', 'Admin berhasil ditambahkan.');

    }

    // Menampilkan detail admin

    public function show($id)

    {

        $admin = Admin::findOrFail($id);

        return view('admin.show', compact('admin'));

    }

    // Menampilkan halaman edit admin

    public function edit($id)

    {

        $admin = Admin::findOrFail($id);

        return view('admin.edit', compact('admin'));

    }

    // Proses update admin

    public function update(Request $request, $id)

    {

        $request->validate([

            'namaadmin' => 'required|string|max:255',

            'username' => 'required|string|max:255|unique:admin,username,' . $id,

            'password' => 'nullable|min:6|confirmed',

            'foto' => 'nullable|image|max:2048',

            'status' => 'required|in:pending,setujui,tolak',

            'role' => 'required|in:admin,petugas',

        ]);

        $admin = Admin::findOrFail($id);

        $admin->namaadmin = $request->namaadmin;

        $admin->username = $request->username;

        if ($request->filled('password')) {

            $admin->password = Hash::make($request->password);

        }

        if ($request->hasFile('foto')) {

            // Hapus foto lama jika ada

            if ($admin->foto) {

                Storage::disk('public')->delete($admin->foto);

            }

            // Simpan foto baru

            $admin->foto = $request->file('foto')->store('admin\_fotos', 'public');

        }

        $admin->status = $request->status;

        $admin->role = $request->role;

        $admin->save();

        return redirect()->route('admin.index')->with('success', 'Admin berhasil diperbarui.');

    }

    // Menghapus admin

    public function destroy($id)

    {

        $admin = Admin::findOrFail($id);

        // Hapus foto jika ada

        if ($admin->foto) {

            Storage::disk('public')->delete($admin->foto);

        }

        $admin->delete();

        return redirect()->route('admin.index')->with('success', 'Admin berhasil dihapus.');

    }

    // Menyetujui admin

    public function approve($id)

    {

        $admin = Admin::findOrFail($id);

        $admin->update(['status' => 'setujui']);

        return redirect()->route('admin.index')->with('success', 'Admin berhasil disetujui.');

    }

    // Menolak admin

    public function reject($id)

    {

        $admin = Admin::findOrFail($id);

        $admin->update(['status' => 'tolak']);

        return redirect()->route('admin.index')->with('success', 'Admin ditolak.');

    }

    // Menampilkan halaman dashboard admin

    public function dashboard()

    {

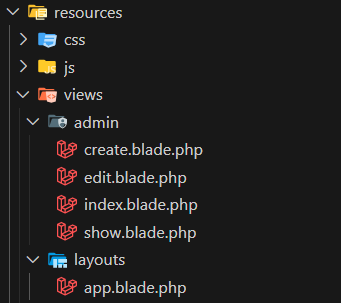
        return view('admin.dashboard');

    }

}

**Views:**

Buat Folder **resources/views/admin/** lalu tambahkan file berikut didalam folder dengan struktur berikut:



**a. index.blade.php**

@extends('layouts.appadmin')

@section('title', 'Manajemen Admin')

@section('content')

<div class="card">

    <div class="card-header" style="background-color: #1B03A3; color: white; font-weight: bold;">

        <h3 class="card-title">Manajemen Admin</h3>

        <div class="card-tools">

            <a href="{{ route('admin.create') }}" class="btn btn-primary btn-sm">

                <i class="fas fa-plus"></i> Tambah Admin

            </a>

        </div>

    </div>

    <!-- /.card-header -->

    <div class="card-body">

        <table id="adminTable" class="table table-bordered table-striped">

            <thead>

                <tr>

                    <th>No</th>

                    <th>Nama</th>

                    <th>Username</th>

                    <th>Status</th>

                    <th>Role</th>

                    <th>Foto</th>

                    <th>Confirm</th>

                    <th>Aksi</th>

                </tr>

            </thead>

            <tbody>

                @foreach ($admins as $index => $admin)

                <tr>

                    <td>{{ $index + 1 }}</td>

                    <td>{{ $admin->namaadmin }}</td>

                    <td>{{ $admin->username }}</td>

                    <td>

                        @if ($admin->status == 'pending')

                        <span class="badge badge-warning">Pending</span>

                        @elseif ($admin->status == 'setujui')

                        <span class="badge badge-success">Disetujui</span>

                        @elseif ($admin->status == 'tolak')

                        <span class="badge badge-danger">Ditolak</span>

                        @endif

                    </td>

                    <td>

                        @if($admin->role == 'admin')

                        <span class="badge badge-danger">{{ ucfirst($admin->role) }}</span>

                        @elseif($admin->role == 'petugas')

                        <span class="badge badge-warning">{{ ucfirst($admin->role) }}</span>

                        @else

                        <span class="badge badge-secondary">{{ ucfirst($admin->role) }}</span>

                        @endif

                    </td>

                    <td>

                        @if($admin->foto)

                        <img src="{{ asset('storage/' . $admin->foto) }}" class="img-thumbnail" width="50">

                        @else

                        <span class="text-muted">Tidak ada foto</span>

                        @endif

                    </td>

                    <td>

                        @if ($admin->status == 'pending')

                        <form action="{{ route('admin.approve', $admin->id) }}" method="POST" style="display:inline;">

                            @csrf

                            @method('PUT')

                            <button type="submit" class="btn btn-success btn-sm">

                                <i class="fas fa-check"></i>

                            </button>

                        </form>

                        <form action="{{ route('admin.reject', $admin->id) }}" method="POST" style="display:inline;">

                            @csrf

                            @method('PUT')

                            <button type="submit" class="btn btn-danger btn-sm">

                                <i class="fas fa-times"></i>

                            </button>

                        </form>

                        @endif

                    </td>

                    <td>

                        <a href="{{ route('admin.show', $admin->id) }}" class="btn btn-info btn-sm">

                            <i class="fas fa-eye"></i>

                        </a>

                        <a href="{{ route('admin.edit', $admin->id) }}" class="btn btn-warning btn-sm">

                            <i class="fas fa-edit"></i>

                        </a>

                        <form id="delete-form-{{ $admin->id }}" action="{{ route('admin.destroy', $admin->id) }}" method="POST" class="d-inline">

                            @csrf

                            @method('DELETE')

                            <button type="button" class="btn btn-danger btn-sm delete-btn" data-id="{{ $admin->id }}">

                                <i class="fas fa-trash"></i>

                            </button>

                        </form>

                    </td>

                </tr>

                @endforeach

            </tbody>

        </table>

    </div>

</div>

@endsection

@section('js')

<script>

    $(document).ready(function () {

        $('.delete-btn').click(function () {

            let id = $(this).data('id');

            if (confirm("Apakah Anda yakin ingin menghapus admin ini?")) {

                document.getElementById('delete-form-' + id).submit();

            }

        });

        // DataTables (Jika ingin menggunakan DataTables, aktifkan ini)

        // $('#adminTable').DataTable();

    });

</script>

@endsection

**b. create.blade.php**

@extends('layouts.appadmin')

@section('title', 'Tambah Admin')

@section('content')

<div class="card">

    <div class="card-header">

        <h3 class="card-title">Tambah Admin</h3>

    </div>

    <div class="card-body">

        @if ($errors->has('username'))

        <div class="alert alert-danger">

            <strong>Error!</strong> Username sudah terdaftar.

        </div>

        @endif

        <form action="{{ route('admin.store') }}" method="POST" enctype="multipart/form-data">

            @csrf

            <div class="row">

                <div class="col-md-6">

                    <!-- Nama Admin -->

                    <div class="form-group">

                        <label for="namaadmin">Nama</label>

                        <input type="text" name="namaadmin" id="namaadmin" class="form-control" required>

                    </div>

                    <!-- Username -->

                    <div class="form-group">

                        <label for="username">Username</label>

                        <input type="text" name="username" id="username" class="form-control" required>

                    </div>

                    <!-- Password -->

                    <div class="form-group">

                        <label for="password">Password</label>

                        <input type="password" name="password" id="password" class="form-control" required>

                    </div>

                    <div class="form-group">

                        <label for="password\_confirmation">Konfirmasi Password</label>

                        <input type="password" name="password\_confirmation" id="password\_confirmation" class="form-control" required>

                    </div>

                </div>

                <div class="col-md-6">

                    <!-- Foto -->

                    <div class="form-group">

                        <label for="foto">Foto</label>

                        <input type="file" name="foto" id="foto" class="form-control-file">

                    </div>

                    <!-- Status -->

                    <div class="form-group">

                        <label for="status">Status</label>

                        <select name="status" id="status" class="form-control">

                            <option value="pending" selected>Pending</option>

                            <option value="setujui">Disetujui</option>

                            <option value="tolak">Ditolak</option>

                        </select>

                    </div>

                    <!-- Role -->

                    <div class="form-group">

                        <label for="role">Role</label>

                        <select name="role" id="role" class="form-control">

                            <option value="admin">Admin</option>

                            <option value="petugas" selected>Petugas</option>

                        </select>

                    </div>

                </div>

            </div>

            <div class="form-group mt-3 text-right">

                <button type="submit" class="btn btn-primary">

                    <i class="fas fa-save"></i> Simpan

                </button>

                <a href="{{ route('admin.index') }}" class="btn btn-secondary">

                    <i class="fas fa-arrow-left"></i> Kembali

                </a>

            </div>

        </form>

    </div>

</div>

@endsection

**c. edit.blade.php**

@extends('layouts.appadmin')

@section('content')

<div class="card">

    <div class="card-header">

        <h3 class="card-title">Edit Admin</h3>

    </div>

    <div class="card-body">

        <form action="{{ route('admin.update', $admin->id) }}" method="POST" enctype="multipart/form-data">

            @csrf

            @method('PUT')

            <div class="row">

                <div class="col-md-6">

                    <div class="form-group">

                        <label for="namaadmin">Nama</label>

                        <input type="text" name="namaadmin" id="namaadmin" class="form-control" value="{{ $admin->namaadmin }}" required>

                    </div>

                    <div class="form-group">

                        <label for="username">Username</label>

                        <input type="text" name="username" id="username" class="form-control" value="{{ $admin->username }}" required>

                    </div>

                    <div class="form-group">

                        <label for="password">Password (Kosongkan jika tidak ingin mengubah)</label>

                        <input type="password" name="password" id="password" class="form-control">

                    </div>

                    <div class="form-group">

                        <label for="password\_confirmation">Konfirmasi Password</label>

                        <input type="password" name="password\_confirmation" id="password\_confirmation" class="form-control">

                    </div>

                </div>

                <div class="col-md-6">

                    <div class="form-group">

                        <label for="foto">Foto</label>

                        <div class="d-flex align-items-center">

                            <div class="me-3">

                                <input type="file" name="foto" id="foto" class="form-control-file">

                            </div>

                            <div>

                                @if ($admin->foto)

                                <img src="{{ asset('storage/' . $admin->foto) }}" alt="Foto Admin" class="img-thumbnail" width="100">

                                @endif

                            </div>

                        </div>

                    </div>

                    <div class="form-group">

                        <label for="status">Status</label>

                        <select name="status" id="status" class="form-control">

                            <option value="pending" {{ $admin->status == 'pending' ? 'selected' : '' }}>Pending</option>

                            <option value="setujui" {{ $admin->status == 'setujui' ? 'selected' : '' }}>Disetujui</option>

                            <option value="tolak" {{ $admin->status == 'tolak' ? 'selected' : '' }}>Ditolak</option>

                        </select>

                    </div>

                    <div class="form-group">

                        <label for="role">Role</label>

                        <select name="role" id="role" class="form-control">

                            <option value="admin" {{ $admin->role == 'admin' ? 'selected' : '' }}>Admin</option>

                            <option value="petugas" {{ $admin->role == 'petugas' ? 'selected' : '' }}>Petugas</option>

                        </select>

                    </div>

                </div>

            </div>

            <div class="form-group mt-3 text-right">

                <button type="submit" class="btn btn-primary"><i class="fas fa-save"></i> Update</button>

                <a href="{{ route('admin.index') }}" class="btn btn-secondary"><i class="fas fa-arrow-left"></i> Kembali</a>

            </div>

        </form>

    </div>

</div>

@endsection

**d. show.blade.php**

@extends('layouts.appadmin')

@section('title', 'Detail Admin')

@section('content')

<div class="container-fluid">

    <div class="row justify-content-center">

        <div class="col-lg-8">

            <div class="card shadow">

                <div class="card-header bg-primary text-white">

                    <h5 class="mb-0">Detail Admin</h5>

                </div>

                <div class="card-body">

                    <div class="row">

                        <!-- Kolom Kiri (Foto Profil) -->

                        <div class="col-md-4 text-center">

                            <img src="{{ $admin->foto ? asset('storage/' . $admin->foto) : asset('img/default-profile.png') }}"

                                 class="img-thumbnail rounded-circle shadow" width="150" alt="Foto Admin">

                        </div>

                        <!-- Kolom Kanan (Detail Admin) -->

                        <div class="col-md-8">

                            <table class="table table-borderless">

                                <tr>

                                    <th>Nama</th>

                                    <td>: {{ $admin->namaadmin }}</td>

                                </tr>

                                <tr>

                                    <th>Username</th>

                                    <td>: {{ $admin->username }}</td>

                                </tr>

                                <tr>

                                    <th>Role</th>

                                    <td>: {{ ucfirst($admin->role) }}</td>

                                </tr>

                                <tr>

                                    <th>Status</th>

                                    <td>:

                                        @if($admin->status == 'setujui')

                                            <span class="badge bg-success">Disetujui</span>

                                        @elseif($admin->status == 'pending')

                                            <span class="badge bg-warning">Pending</span>

                                        @else

                                            <span class="badge bg-danger">Ditolak</span>

                                        @endif

                                    </td>

                                </tr>

                            </table>

                        </div>

                    </div>

                </div>

                <div class="card-footer text-end">

                    <a href="{{ route('admin.index') }}" class="btn btn-secondary">Kembali</a>

                </div>

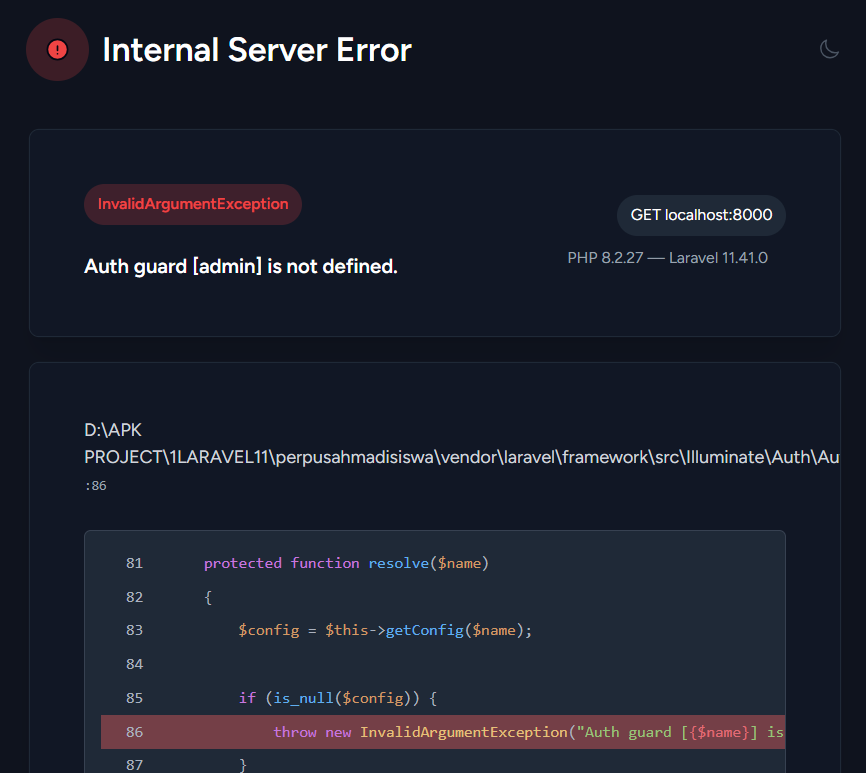
            </div>

        </div>

    </div>

</div>

@endsection



Dan jika mendapatkan pesan error seperti diatas, jangan panik.

Error **"InvalidArgumentException: Auth guard [admin] is not defined."** terjadi karena Laravel tidak menemukan guard admin yang didefinisikan dalam konfigurasi autentikasi.

**Solusi: Tambahkan Guard admin di config/auth.php**

1. Buka file **config/auth.php**
2. Cari bagian **guards**, lalu tambahkan konfigurasi admin:

'guards' => [

'web' => [

'driver' => 'session',

'provider' => 'users',

],

'admin' => [ // Tambahkan guard admin

'driver' => 'session',

'provider' => 'admins',

],

],

1. Cari bagian **providers**, lalu tambahkan provider admins:

'providers' => [

'users' => [

'driver' => 'eloquent',

'model' => App\Models\User::class,

],

'admins' => [ // Tambahkan provider untuk admin

'driver' => 'eloquent',

'model' => App\Models\Admin::class,

],

],

1. Tambahkan middleware untuk admin di bagian **config/auth.php**, jika belum ada:

'passwords' => [

'users' => [

'provider' => 'users',

'table' => 'password\_resets',

'expire' => 60,

'throttle' => 60,

],

'admins' => [ // Tambahkan ini

'provider' => 'admins',

'table' => 'password\_resets',

'expire' => 60,

'throttle' => 60,

],

],

1. **Simpan file** dan jalankan perintah berikut untuk memuat ulang konfigurasi:

php artisan config:clear

php artisan cache:clear

atau kode diatas secara keseluruhan adalah sbb:

<?php

return [

    'defaults' => [

        'guard' => 'web', // Gunakan default 'web' atau sesuaikan jika perlu

        'passwords' => 'users',

    ],

    'guards' => [

        'web' => [

            'driver' => 'session',

            'provider' => 'users',

        ],

        'admin' => [

            'driver' => 'session',

            'provider' => 'admins',

        ],

        'peminjam' => [

            'driver' => 'session',

            'provider' => 'peminjams',

        ],

    ],

    'providers' => [

        'users' => [

            'driver' => 'eloquent',

            'model' => App\Models\User::class,

        ],

        'admins' => [

            'driver' => 'eloquent',

            'model' => App\Models\Admin::class,

        ],

        'peminjams' => [

            'driver' => 'eloquent',

            'model' => App\Models\Peminjam::class,

        ],

    ],

    'passwords' => [

        'users' => [

            'provider' => 'users',

            'table' => 'password\_reset\_tokens',

            'expire' => 60,

            'throttle' => 60,

        ],

        'admins' => [

            'provider' => 'admins',

            'table' => 'password\_reset\_tokens',

            'expire' => 60,

            'throttle' => 60,

        ],

        'peminjams' => [

            'provider' => 'peminjams',

            'table' => 'password\_reset\_tokens',

            'expire' => 60,

            'throttle' => 60,

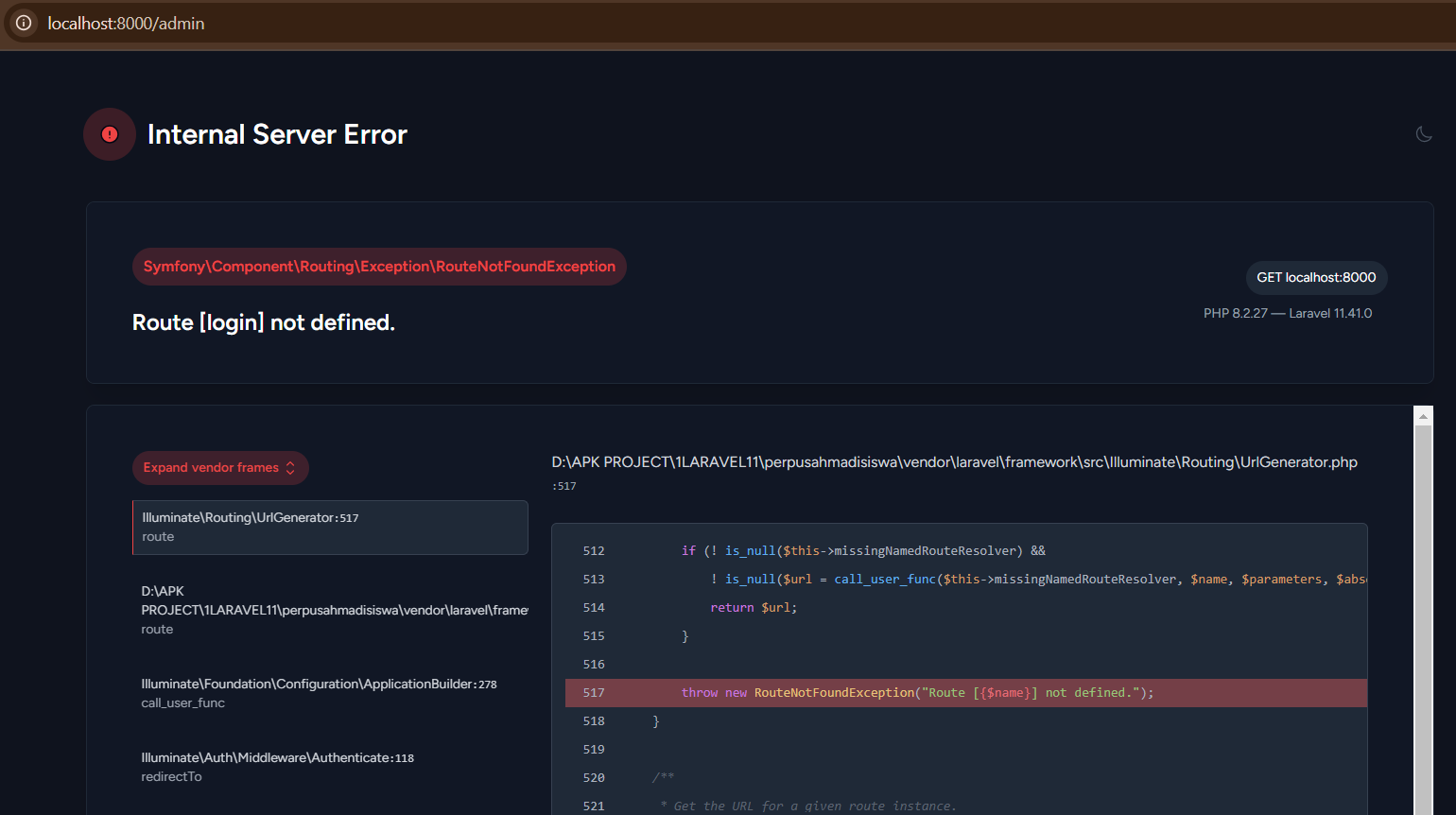
        ],

    ],

    'password\_timeout' => 10800,

];

jika muncul error ketika mengakses halaman admin seperti berikut, itu terjadi karena kita belum menambahkan route loginnya, padahal aturan login sudah kita terapkan melalui midleware.



Solusinya:

Buat route untuk login admin secara lengkap

Namun jika belum ingin membuatnya, cukup komentari route yang ada saat ini, dan pastekan route yang belum ada middlewarenya berikut ini:



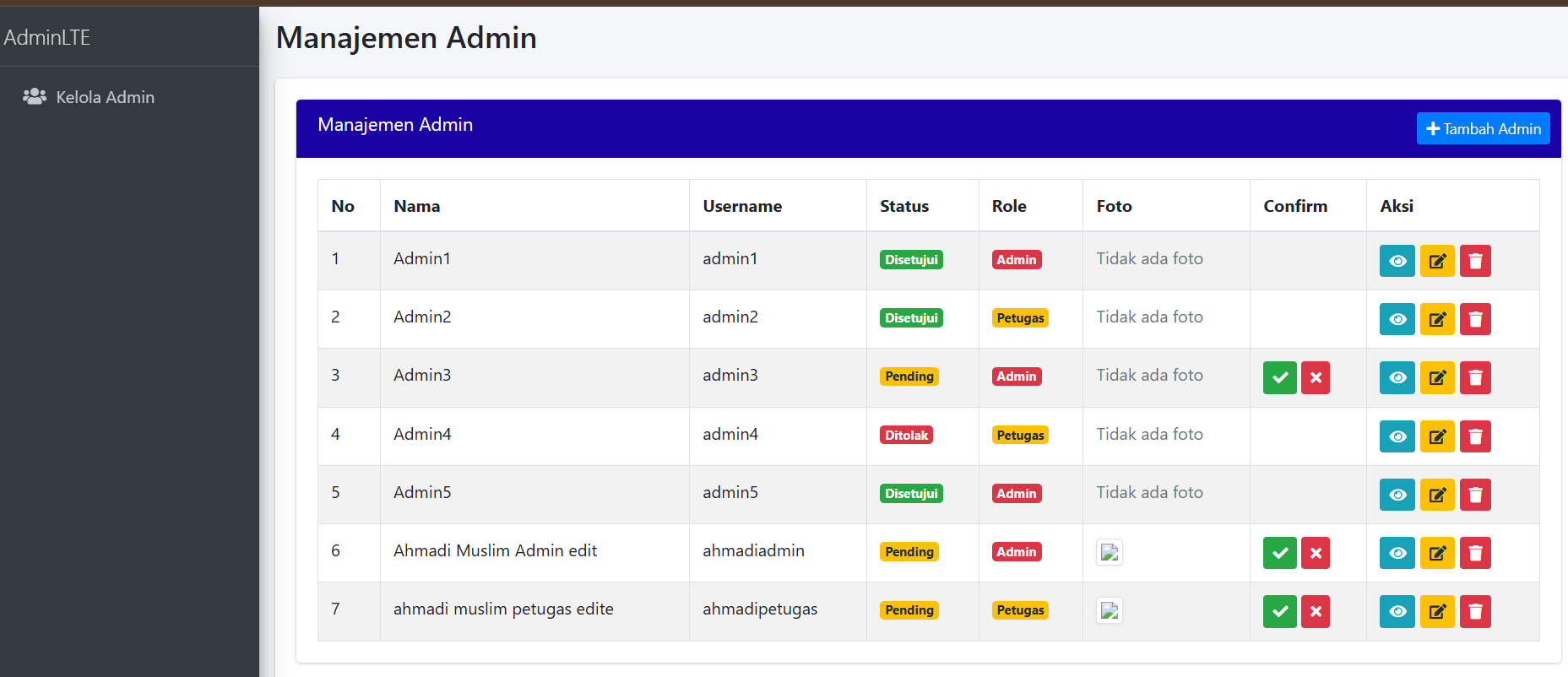
Route::get('/admin/dashboard', [ControllerAdmin::class, 'dashboard'])->name('admin.dashboard');

Route::resource('/admin', ControllerAdmin::class);

Route::post('/admin/{id}/approve', [ControllerAdmin::class, 'approve'])->name('admin.approve');

Route::post('/admin/{id}/reject', [ControllerAdmin::class, 'reject'])->name('admin.reject');

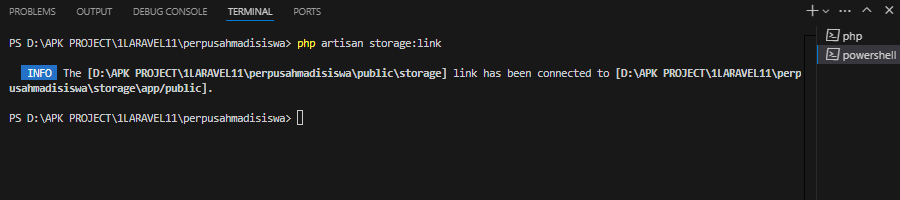
Pada saat ini mungkin foto belum dapat ditampilkan

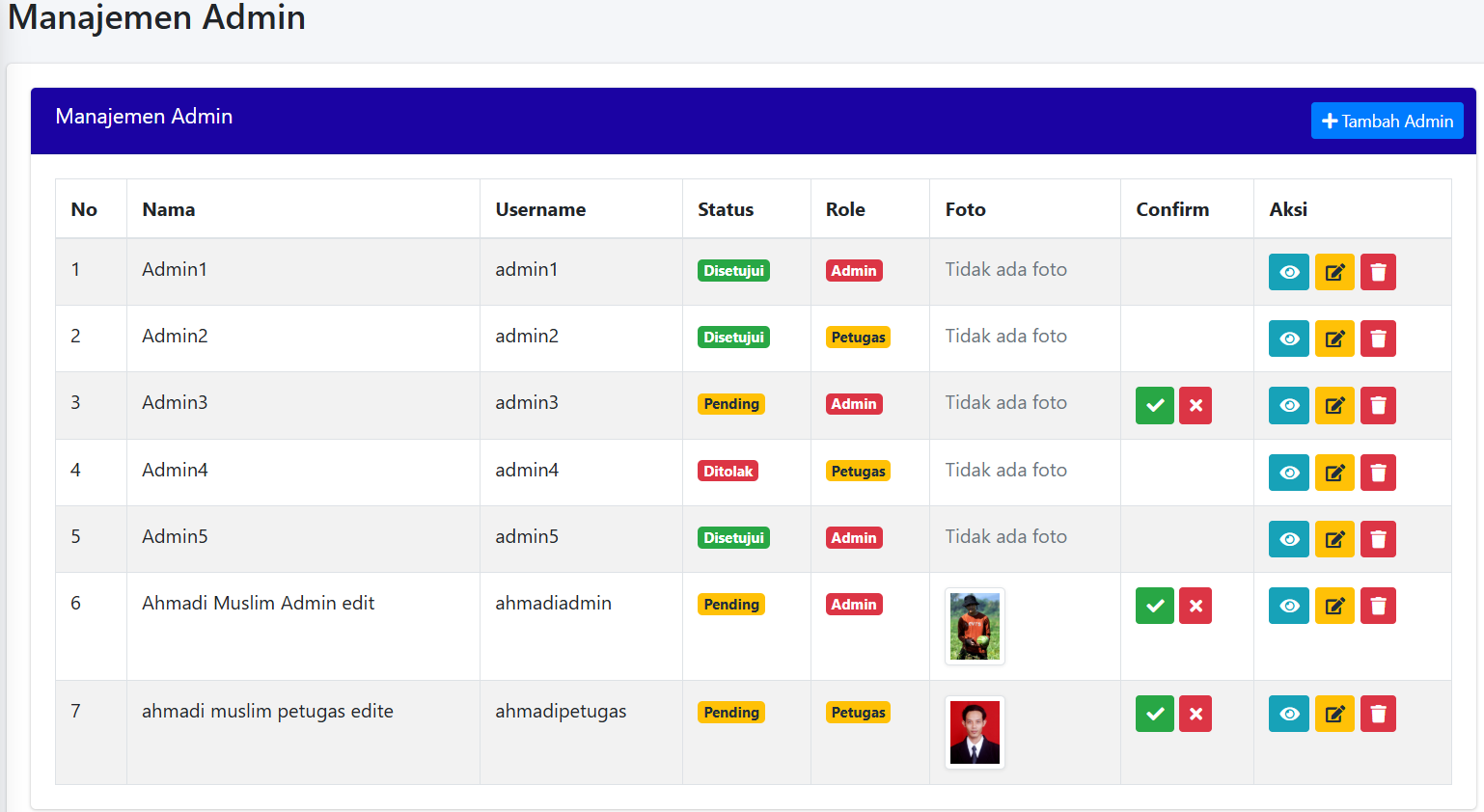


Pastikan bahwa file foto benar-benar ada di dalam direktori storage/app/public/ dan bahwa Anda telah menjalankan perintah berikut:

php artisan storage:link

Perintah ini membuat symbolic link antara storage/app/public/ dengan public/storage/ sehingga bisa diakses melalui browser.





1. **Membuat Fitur LOGIN** admin

Untuk membuat fitur login kita juga harus menerapkan prinsip MODEL, VIEW, CONTROLLER (MVC) juga, sama seperti membuat fitur admin. Akan tetapi karena kita akan menggunakan data dari tabel Admin, maka kita cukup membuat view dan controllernya saja, sedangkan model kita gunakan dari admin. Namun, sebelumnya, kita perlu pula mengatur routenya di web.php.

**Mengatur Routing LOGIN di route/web.php**

Tambahkan route di routes/web.php:

use App\Http\Controllers\ControllerLoginAdmin;

Route::middleware(['guest'])->group(function () {

    Route::get('/admin/login', [ControllerLoginAdmin::class, 'showLoginForm'])->name('admin.login');

    Route::post('/admin/login', [ControllerLoginAdmin::class, 'login']);

    Route::get('/admin/register', [ControllerLoginAdmin::class, 'showRegisterForm'])->name('admin.register');

    Route::post('/admin/register', [ControllerLoginAdmin::class, 'register']);

});

**Penjelasan Route Web**

Kode di atas adalah sekumpulan **route** dalam Laravel yang digunakan untuk **autentikasi admin** (login dan registrasi). Semua route ini dilindungi oleh middleware guest, sehingga **hanya pengguna yang belum login** yang dapat mengaksesnya.

**1. Middleware guest**

Route::middleware(['guest'])->group(function () { ... });

* **Middleware guest** memastikan bahwa route di dalamnya **hanya bisa diakses oleh pengguna yang belum login**.
* Jika pengguna sudah login, Laravel biasanya akan mengarahkan mereka ke halaman lain (misalnya dashboard).

**2. Route Login Admin**

Route::get('/admin/login', [ControllerLoginAdmin::class, 'showLoginForm'])->name('admin.login');

* **Method**: GET
* **URL**: /admin/login
* **Controller**: ControllerLoginAdmin@showLoginForm
* **Fungsi**: Menampilkan halaman login admin (views/admin/login.blade.php).
* **Akses di browser**:

http://localhost:8000/admin/login

Route::post('/admin/login', [ControllerLoginAdmin::class, 'login']);

* **Method**: POST
* **URL**: /admin/login
* **Controller**: ControllerLoginAdmin@login
* **Fungsi**: Menerima input login (email & password), memverifikasi, dan mengarahkan admin ke dashboard jika berhasil.
* **Tidak bisa diakses langsung di browser** karena ini adalah request **POST**, harus dikirim melalui form.

**3. Route Registrasi Admin**

Route::get('/admin/register', [ControllerLoginAdmin::class, 'showRegisterForm'])->name('admin.register');

* **Method**: GET
* **URL**: /admin/register
* **Controller**: ControllerLoginAdmin@showRegisterForm
* **Fungsi**: Menampilkan halaman registrasi admin (views/admin/register.blade.php).
* **Akses di browser**:

http://localhost:8000/admin/register

Route::post('/admin/register', [ControllerLoginAdmin::class, 'register']);

* **Method**: POST
* **URL**: /admin/register
* **Controller**: ControllerLoginAdmin@register
* **Fungsi**: Menerima data pendaftaran admin baru, memprosesnya, dan menyimpannya ke database.
* **Tidak bisa diakses langsung di browser**, harus dikirim melalui form **POST**.

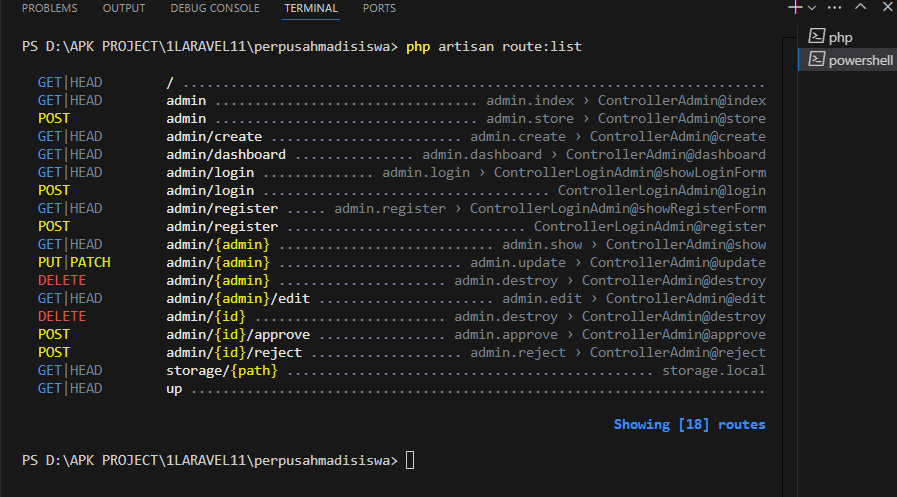
**Bagaimana Cara Mengaksesnya di Browser?**

| **Route** | **URL untuk diakses di browser** | **Method** | **Fungsi** |
| --- | --- | --- | --- |
| Login (Form) | http://localhost:8000/admin/login | GET | Menampilkan halaman login |
| Login (Proses) | http://localhost:8000/admin/login | POST | Memproses login (dikirim melalui form) |
| Register (Form) | http://localhost:8000/admin/register | GET | Menampilkan halaman registrasi |
| Register (Proses) | http://localhost:8000/admin/register | POST | Memproses registrasi admin (dikirim melalui form) |

**Kesimpulan**

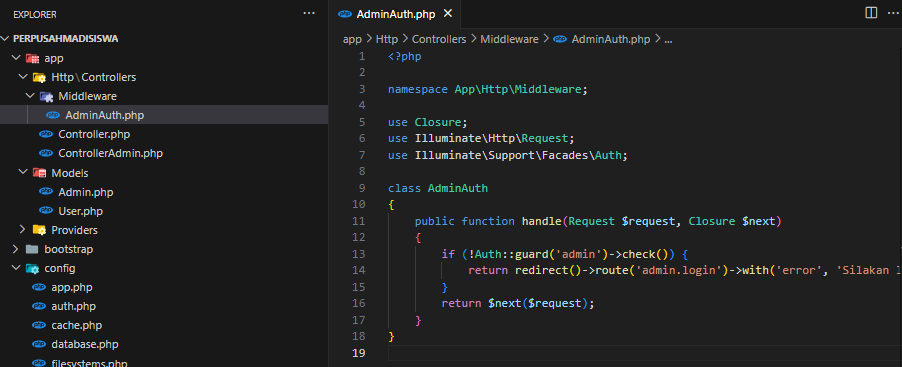
* **Route GET** bisa diakses langsung dari browser (/admin/login dan /admin/register).
* **Route POST** tidak bisa diakses langsung di browser, tetapi digunakan saat form dikirimkan.
* Middleware guest memastikan route ini hanya bisa diakses oleh pengguna yang belum login.
* Jika admin sudah login, Laravel akan **mengalihkan mereka ke halaman lain** (biasanya dashboard).

Berikut daftar Route yang terlist di artisan:



**2. Buat Middleware untuk Admin**

Buat secara manual saja dengan kenampakan struktur folder sbb:



Copykan kode berikut ke file app/Http/Middleware/AdminAuth.php:

<?php

namespace App\Http\Middleware;

use Closure;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

class AdminAuth

{

    public function handle(Request $request, Closure $next)

    {

        if (!Auth::guard('admin')->check()) {

            return redirect()->route('admin.login')->with('error', 'Silakan login terlebih dahulu.');

        }

        return $next($request);

    }

}

**2. Buat Kernel untuk Admin**

Buat file app/Http/Kernel.php secara manual sehingga kenampakannya sbb:



Copykan kode berikut ke app/Http/Kernel.php:

<?php

namespace App\Http;

use Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel

{

    protected $middlewareAliases = [

        'auth' => \Illuminate\Auth\Middleware\Authenticate::class,

    ];

    protected $routeMiddleware = [

        'auth.admin' => \App\Http\Middleware\AdminAuth::class,

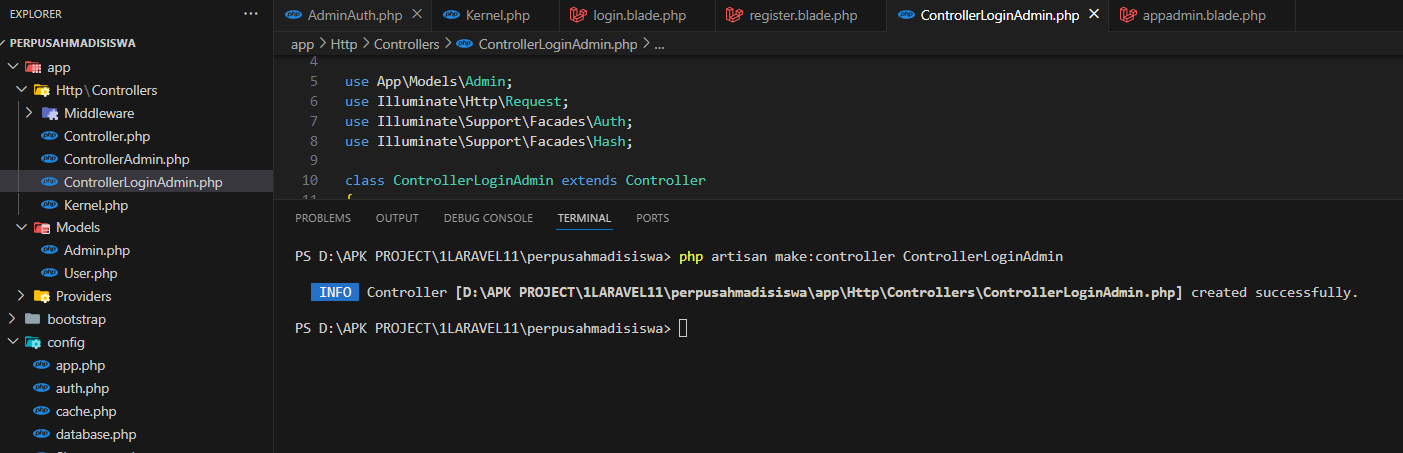
    ];

}

**3. Buat Controller**

Jalankan perintah:

php artisan make:controller ControllerLoginAdmin



Edit file app/Http/Controllers/ControllerLoginAdmin.php:

<?php

namespace App\Http\Controllers;

use App\Models\Admin;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Hash;

class ControllerAdmin extends Controller

{

    // Menampilkan halaman login

    public function showLoginForm()

    {

        return view('admin.login');

    }

    // Proses login admin

    public function login(Request $request)

    {

        $credentials = $request->validate([

            'username' => 'required|string',

            'password' => 'required|string',

        ]);

        $admin = Admin::where('username', $request->username)->first();

        if (!$admin || !Hash::check($request->password, $admin->password)) {

            return back()->withErrors(['username' => 'Username atau password salah.']);

        }

        if ($admin->status !== 'setujui') {

            return back()->withErrors(['username' => 'Akun belum disetujui oleh admin.']);

        }

        Auth::guard('admin')->login($admin);

        return redirect()->route('admin.dashboard')->with('success', 'Login berhasil!');

    }

    // Menampilkan halaman register

    public function showRegisterForm()

    {

        return view('admin.register');

    }

    // Proses registrasi admin

    public function register(Request $request)

    {

        $request->validate([

            'namaadmin' => 'required|string|max:255',

            'username' => 'required|string|unique:admin',

            'password' => 'required|min:6|confirmed',

            'role' => 'required|in:admin,petugas',

            'foto' => 'nullable|image|max:2048'

        ]);

        $fotoPath = $request->file('foto') ? $request->file('foto')->store('admin\_photos', 'public') : null;

        Admin::create([

            'namaadmin' => $request->namaadmin,

            'username' => $request->username,

            'password' => Hash::make($request->password),

            'role' => $request->role,

            'status' => 'pending', // Status default "pending" sampai disetujui

            'foto' => $fotoPath

        ]);

        return redirect()->route('admin.login')->with('success', 'Registrasi berhasil! Menunggu persetujuan admin.');

    }

    // Logout admin

    public function logout()

    {

        Auth::guard('admin')->logout();

        return redirect()->route('admin.login')->with('success', 'Anda telah logout.');

    }

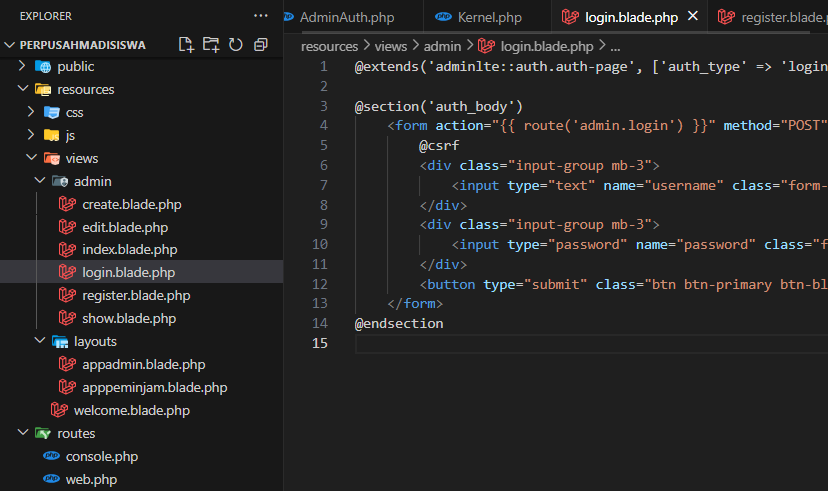
}

**5. Buat Halaman VIEWS => Login dan Register**

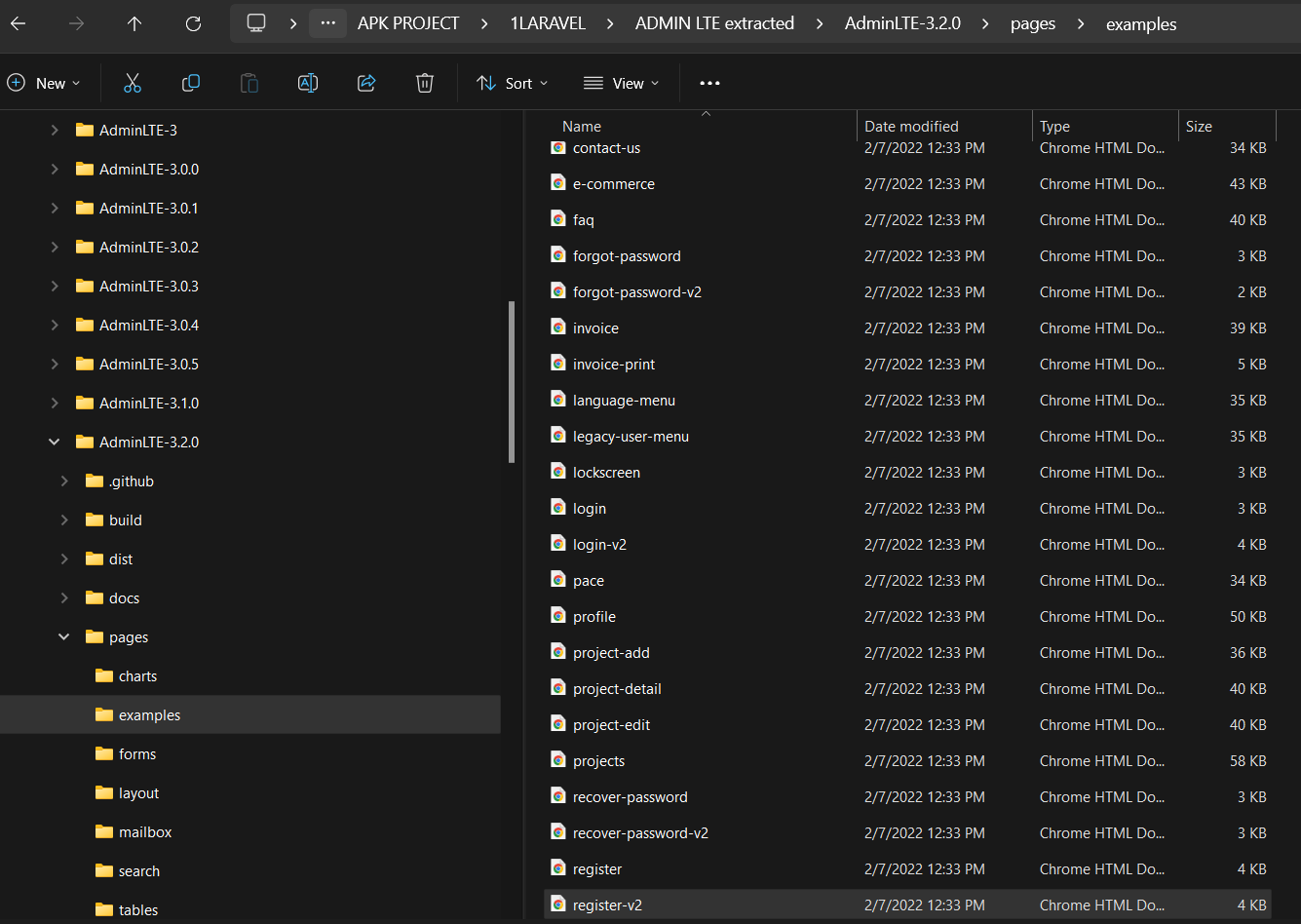
Buat 2 buah file di views/admin sbb

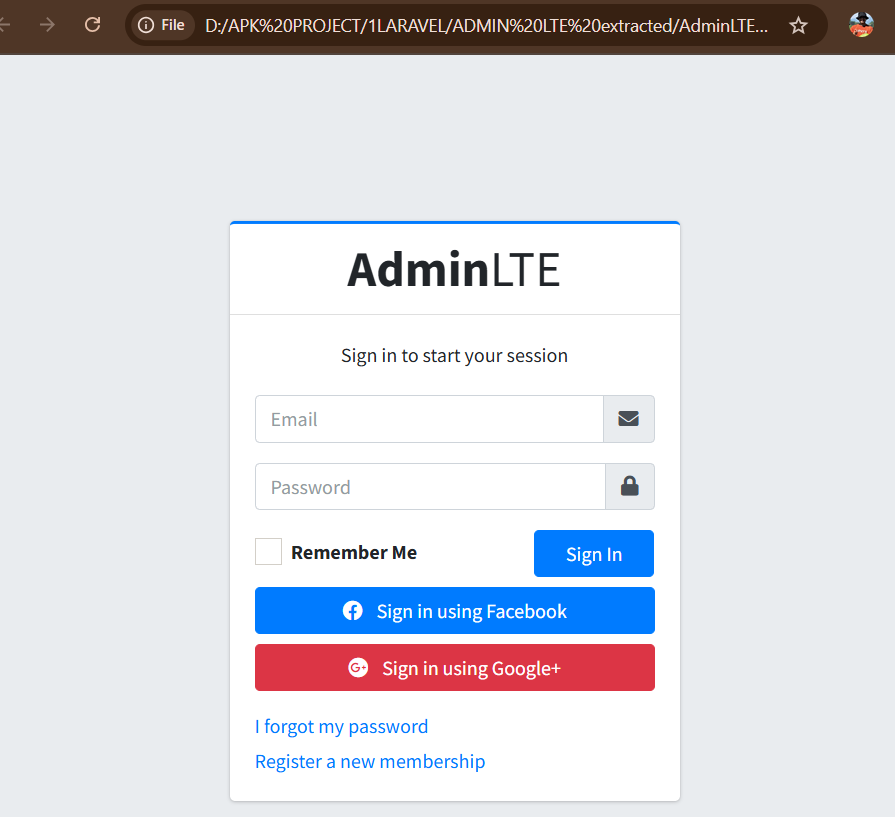
Login : resources/views/admin/login.blade.php

Register : resources/views/admin/register.blade.php



Gunakan standar halaman login dari Admin LTE





Copykan kodenya di admin/login.blade.php sbb:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Login Admin</title>

    <!-- AdminLTE & Bootstrap CSS -->

    <link rel="stylesheet" href="{{ asset('dist/css/adminlte.min.css') }}">

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

    <link rel="stylesheet" href="{{ asset('plugins/bootstrap/css/bootstrap.min.css') }}">

</head>

<body class="hold-transition login-page">

<div class="login-box">

    <!-- Login Logo -->

    <div class="card card-outline card-primary">

        <div class="card-header text-center">

            <a href="#" class="h1"><b>Admin</b>LTE</a>

        </div>

        <div class="card-body">

            <p class="login-box-msg">Sign in to start your session</p>

            @if(session('error'))

                <div class="alert alert-danger">

                    {{ session('error') }}

                </div>

            @endif

            <form action="{{ route('admin.login.post') }}" method="POST">

                @csrf

                <div class="input-group mb-3">

                    <input type="text" name="username" class="form-control" placeholder="Username" required autofocus>

                    <div class="input-group-append">

                        <div class="input-group-text">

                            <span class="fas fa-user"></span>

                        </div>

                    </div>

                </div>

                <div class="input-group mb-3">

                    <input type="password" name="password" class="form-control" placeholder="Password" required>

                    <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-8">

                        <div class="icheck-primary">

                            <input type="checkbox" id="remember" name="remember">

                            <label for="remember">

                                Remember Me

                            </label>

                        </div>

                    </div>

                    <div class="col-4">

                        <button type="submit" class="btn btn-primary btn-block">Sign In</button>

                    </div>

                </div>

            </form>

            <p class="mb-1">

                <a href="{{ route('admin.password.request') }}">I forgot my password</a>

            </p>

            <p class="mb-0">

                <a href="{{ route('admin.register') }}" class="text-center">Register a new membership</a>

            </p>

        </div>

    </div>

</div>

<!-- jQuery -->

<script src="{{ asset('plugins/jquery/jquery.min.js') }}"></script>

<!-- Bootstrap -->

<script src="{{ asset('plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>

<!-- AdminLTE App -->

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

</body>

</html>

Copykan kodenya di admin/register.blade.php sbb:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Register Admin</title>

    <!-- AdminLTE & Bootstrap CSS -->

    <link rel="stylesheet" href="{{ asset('dist/css/adminlte.min.css') }}">

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

    <link rel="stylesheet" href="{{ asset('plugins/icheck-bootstrap/icheck-bootstrap.min.css') }}">

</head>

<body class="hold-transition register-page">

<div class="register-box">

    <div class="card card-outline card-primary">

        <div class="card-header text-center">

            <a href="#" class="h1"><b>Admin</b>LTE</a>

        </div>

        <div class="card-body">

            <p class="login-box-msg">Register a new membership</p>

            @if(session('error'))

                <div class="alert alert-danger">

                    {{ session('error') }}

                </div>

            @endif

            <form action="{{ route('admin.register.post') }}" method="POST" enctype="multipart/form-data">

                @csrf

                <div class="input-group mb-3">

                    <input type="text" name="namaadmin" class="form-control" placeholder="Nama Lengkap" required>

                    <div class="input-group-append">

                        <div class="input-group-text">

                            <span class="fas fa-user"></span>

                        </div>

                    </div>

                </div>

                <div class="input-group mb-3">

                    <input type="text" name="username" class="form-control" placeholder="Username" required>

                    <div class="input-group-append">

                        <div class="input-group-text">

                            <span class="fas fa-user"></span>

                        </div>

                    </div>

                </div>

                <div class="input-group mb-3">

                    <input type="password" name="password" class="form-control" placeholder="Password" required>

                    <div class="input-group-append">

                        <div class="input-group-text">

                            <span class="fas fa-lock"></span>

                        </div>

                    </div>

                </div>

                <div class="input-group mb-3">

                    <input type="password" name="password\_confirmation" class="form-control" placeholder="Konfirmasi Password" required>

                    <div class="input-group-append">

                        <div class="input-group-text">

                            <span class="fas fa-lock"></span>

                        </div>

                    </div>

                </div>

                <div class="input-group mb-3">

                    <select name="role" class="form-control" required>

                        <option value="petugas" selected>Petugas</option>

                        <option value="admin">Admin</option>

                    </select>

                    <div class="input-group-append">

                        <div class="input-group-text">

                            <span class="fas fa-user-tag"></span>

                        </div>

                    </div>

                </div>

                <div class="input-group mb-3">

                    <input type="file" name="foto" class="form-control">

                    <div class="input-group-append">

                        <div class="input-group-text">

                            <span class="fas fa-image"></span>

                        </div>

                    </div>

                </div>

                <div class="row">

                    <div class="col-8">

                        <div class="icheck-primary">

                            <input type="checkbox" id="agreeTerms" name="terms" value="agree" required>

                            <label for="agreeTerms">

                                I agree to the <a href="#">terms</a>

                            </label>

                        </div>

                    </div>

                    <div class="col-4">

                        <button type="submit" class="btn btn-primary btn-block">Register</button>

                    </div>

                </div>

            </form>

            <p class="mt-3 mb-1">

                <a href="{{ route('admin.login') }}">I already have a membership</a>

            </p>

        </div>

    </div>

</div>

<!-- jQuery -->

<script src="{{ asset('plugins/jquery/jquery.min.js') }}"></script>

<!-- Bootstrap -->

<script src="{{ asset('plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>

<!-- AdminLTE App -->

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

</body>

</html>

**Mengatasi kendala reset password dan tetap login**

**Solusi: Tambahkan Route Reset Password**

Tambahkan route ini di **routes/web.php**, dalam **middleware guest**:

Route::get('/admin/forgot-password', function () {

return view('admin.forgot-password'); // Buat halaman reset password

})->name('admin.password.request');

**📌 Jika Ingin Menambahkan Reset Password Secara Penuh**

1. **Buat Controller Reset Password** (jika ingin fitur lengkap):

sh

SalinEdit

php artisan make:controller ControllerResetPasswordAdmin

1. **Tambahkan Logika Reset Password di ControllerResetPasswordAdmin.php**:

<?php

namespace App\Http\Controllers;

use App\Models\Admin;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Hash;

use Illuminate\Support\Facades\Password;

class ControllerResetPasswordAdmin extends Controller

{

public function showForgotPasswordForm()

{

return view('admin.forgot-password');

}

public function sendResetLink(Request $request)

{

$request->validate(['username' => 'required|exists:admin,username']);

// Proses pengiriman token reset password (custom)

return back()->with('success', 'Link reset password telah dikirim!');

}

public function showResetForm($token)

{

return view('admin.reset-password', ['token' => $token]);

}

public function resetPassword(Request $request)

{

$request->validate([

'username' => 'required|exists:admin,username',

'password' => 'required|min:6|confirmed',

]);

$admin = Admin::where('username', $request->username)->first();

$admin->password = Hash::make($request->password);

$admin->save();

return redirect()->route('admin.login')->with('success', 'Password berhasil direset.');

}

}

1. **Tambahkan Route Reset Password di routes/web.php**:

Route::get('/admin/forgot-password', [ControllerResetPasswordAdmin::class, 'showForgotPasswordForm'])->name('admin.password.request');

Route::post('/admin/forgot-password', [ControllerResetPasswordAdmin::class, 'sendResetLink'])->name('admin.password.email');

Route::get('/admin/reset-password/{token}', [ControllerResetPasswordAdmin::class, 'showResetForm'])->name('admin.password.reset');

Route::post('/admin/reset-password', [ControllerResetPasswordAdmin::class, 'resetPassword'])->name('admin.password.update');

1. **Buat View untuk Reset Password**
   * **resources/views/admin/forgot-password.blade.php**

SalinEdit

@extends('layouts.appadmin')

@section('content')

<div class="login-box">

<div class="card card-outline card-primary">

<div class="card-header text-center">

<a href="#" class="h1"><b>Admin</b>LTE</a>

</div>

<div class="card-body">

<p class="login-box-msg">Masukkan username Anda untuk reset password</p>

@if (session('success'))

<div class="alert alert-success">{{ session('success') }}</div>

@endif

@if ($errors->any())

<div class="alert alert-danger">

<ul>

@foreach ($errors->all() as $error)

<li>{{ $error }}</li>

@endforeach

</ul>

</div>

@endif

<form action="{{ route('admin.password.email') }}" method="post">

@csrf

<div class="input-group mb-3">

<input type="text" name="username" class="form-control" placeholder="Username" required>

<div class="input-group-append">

<div class="input-group-text">

<span class="fas fa-user"></span>

</div>

</div>

</div>

<div class="row">

<div class="col-12">

<button type="submit" class="btn btn-primary btn-block">Kirim Link Reset</button>

</div>

</div>

</form>

<p class="mt-3 mb-1">

<a href="{{ route('admin.login') }}">Kembali ke Login</a>

</p>

</div>

</div>

</div>

@endsection

* + **resources/views/admin/reset-password.blade.php**

SalinEdit

@extends('layouts.appadmin')

@section('content')

<div class="login-box">

<div class="card card-outline card-primary">

<div class="card-header text-center">

<a href="#" class="h1"><b>Admin</b>LTE</a>

</div>

<div class="card-body">

<p class="login-box-msg">Masukkan password baru Anda</p>

@if ($errors->any())

<div class="alert alert-danger">

<ul>

@foreach ($errors->all() as $error)

<li>{{ $error }}</li>

@endforeach

</ul>

</div>

@endif

<form action="{{ route('admin.password.update') }}" method="post">

@csrf

<input type="hidden" name="token" value="{{ $token }}">

<div class="input-group mb-3">

<input type="text" name="username" class="form-control" placeholder="Username" required>

<div class="input-group-append">

<div class="input-group-text">

<span class="fas fa-user"></span>

</div>

</div>

</div>

<div class="input-group mb-3">

<input type="password" name="password" class="form-control" placeholder="Password Baru" required>

<div class="input-group-append">

<div class="input-group-text">

<span class="fas fa-lock"></span>

</div>

</div>

</div>

<div class="input-group mb-3">

<input type="password" name="password\_confirmation" class="form-control" placeholder="Konfirmasi Password" required>

<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">Reset Password</button>

</div>

</div>

</form>

<p class="mt-3 mb-1">

<a href="{{ route('admin.login') }}">Kembali ke Login</a>

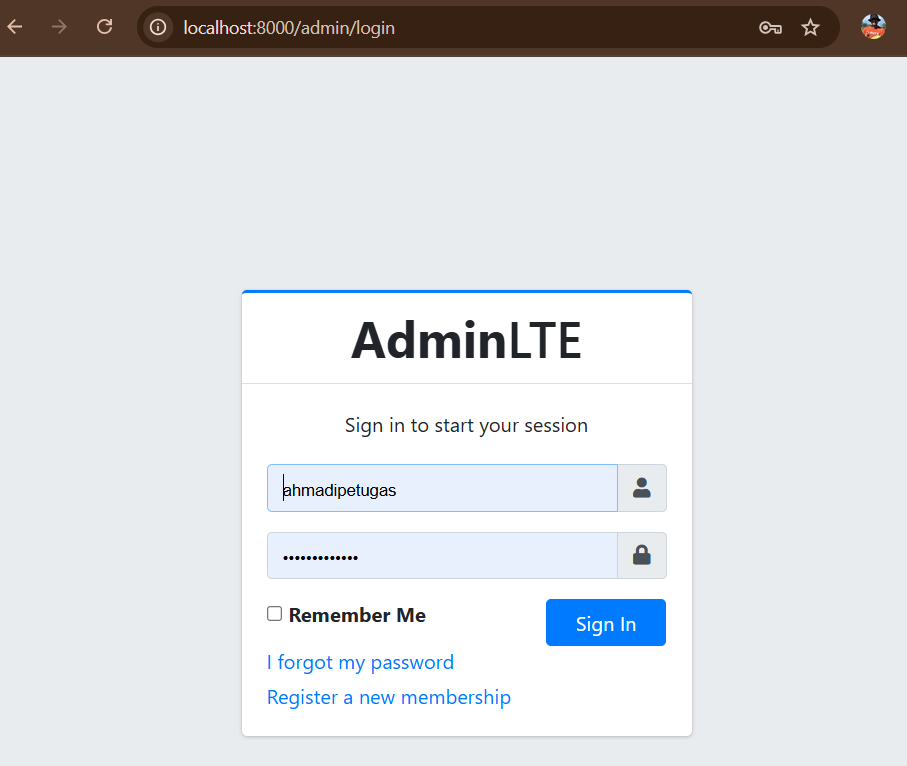
</p>

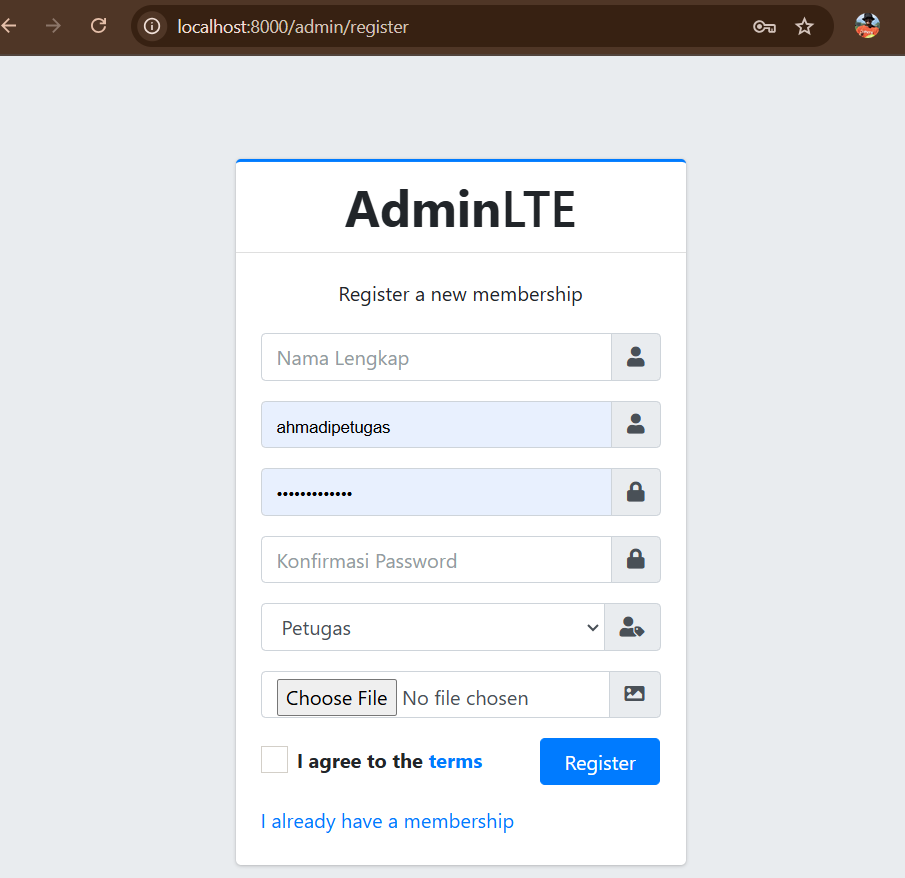
</div>

</div>

</div>

@endsection

****

****

**View Dashboard (resources/views/admin/dashboard.blade.php)**

blade

SalinEdit

@extends('layouts.appadmin')

@section('content')

<div class="content-header">

    <div class="container-fluid">

        <div class="row mb-2">

            <div class="col-sm-6">

                <h1 class="m-0">Dashboard Admin</h1>

            </div>

            <div class="col-sm-6">

                <ol class="breadcrumb float-sm-right">

                    <li class="breadcrumb-item"><a href="#">Home</a></li>

                    <li class="breadcrumb-item active">Dashboard</li>

                </ol>

            </div>

        </div>

    </div>

</div>

<div class="content">

    <div class="container-fluid">

        <div class="row">

            <!-- Selamat Datang -->

            <div class="col-md-12">

                <div class="alert alert-primary">

                    <h5>👋 Selamat datang, {{ auth()->user()->namaadmin }}!</h5>

                    <p>Anda masuk sebagai <strong>{{ ucfirst(auth()->user()->role) }}</strong>. Gunakan menu di sidebar untuk mengelola sistem.</p>

                </div>

            </div>

            <!-- Statistik Admin -->

            <div class="col-lg-4 col-6">

                <div class="small-box bg-info">

                    <div class="inner">

                        <h3>150</h3>

                        <p>Total Peminjaman</p>

                    </div>

                    <div class="icon">

                        <i class="fas fa-book"></i>

                    </div>

                    <a href="#" class="small-box-footer">

                        Lihat Detail <i class="fas fa-arrow-circle-right"></i>

                    </a>

                </div>

            </div>

            <div class="col-lg-4 col-6">

                <div class="small-box bg-success">

                    <div class="inner">

                        <h3>75</h3>

                        <p>Admin/Petugas</p>

                    </div>

                    <div class="icon">

                        <i class="fas fa-users"></i>

                    </div>

                    <a href="#" class="small-box-footer">

                        Lihat Detail <i class="fas fa-arrow-circle-right"></i>

                    </a>

                </div>

            </div>

            <div class="col-lg-4 col-6">

                <div class="small-box bg-warning">

                    <div class="inner">

                        <h3>20</h3>

                        <p>Peminjam Baru</p>

                    </div>

                    <div class="icon">

                        <i class="fas fa-user-plus"></i>

                    </div>

                    <a href="#" class="small-box-footer">

                        Lihat Detail <i class="fas fa-arrow-circle-right"></i>

                    </a>

                </div>

            </div>

            <!-- Card Informasi -->

            <div class="col-md-12">

                <div class="card">

                    <div class="card-header">

                        <h3 class="card-title">Informasi Sistem</h3>

                    </div>

                    <div class="card-body">

                        <ul>

                            <li>Sistem ini digunakan untuk mengelola peminjaman buku.</li>

                            <li>Admin dapat menyetujui atau menolak pendaftaran peminjam.</li>

                            <li>Silakan gunakan menu di sidebar untuk mengakses fitur lainnya.</li>

                        </ul>

                    </div>

                </div>

            </div>

        </div>

    </div>

</div>

@endsection

Dengan ini, fitur login dan registrasi admin sudah lengkap.

Selanjutnya untuk membuat aplikasi kita menjadi lebih menarik, perbaiki appadmin.blade.php dengan kode sbb:

<!DOCTYPE html>

<html lang="id">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Admin Panel</title>

    <!-- AdminLTE & Bootstrap CSS -->

    <link rel="stylesheet" href="{{ asset('dist/css/adminlte.min.css') }}">

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

    <link rel="stylesheet" href="{{ asset('plugins/bootstrap/css/bootstrap.min.css') }}">

    <!-- SWEETALERT -->

    <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>

    <!-- DataTables -->

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

    <link rel="stylesheet" href="{{ asset('plugins/datatables-responsive/css/responsive.bootstrap4.min.css') }}">

    <link rel="stylesheet" href="{{ asset('plugins/datatables-buttons/css/buttons.bootstrap4.min.css') }}">

</head>

<body class="hold-transition sidebar-mini layout-fixed">

    <div class="wrapper">

        <!-- Navbar -->

        <nav class="main-header navbar navbar-expand navbar-white navbar-light">

            <ul class="navbar-nav">

                <li class="nav-item">

                    <a class="nav-link" data-widget="pushmenu" href="#" role="button"><i class="fas fa-bars"></i></a>

                </li>

            </ul>

            <!-- Navbar Right Menu -->

            <ul class="navbar-nav ml-auto">

                <!-- Dropdown Profil -->

                <li class="nav-item dropdown">

                    <a class="nav-link" data-toggle="dropdown" href="#">

                        <!-- Foto Pengguna -->

                        @php

                        $foto = auth()->user()->foto ? asset('storage/' . auth()->user()->foto) : asset('dist/img/default-avatar.png');

                        @endphp

                        <img src="{{ $foto }}" class="img-circle elevation-2" alt="User Image" width="30" height="30">

                        <span class="ml-2">{{ auth()->user()->namaadmin }}</span>

                    </a>

                    <div class="dropdown-menu dropdown-menu-lg dropdown-menu-right">

                        <div class="dropdown-header text-center">

                            <img src="{{ $foto }}" class="img-circle elevation-2" alt="User Image" width="60" height="60">

                            <p class="mt-2">{{ auth()->user()->namaadmin }} <br> <small>{{ auth()->user()->role }}</small></p>

                        </div>

                        <div class="dropdown-divider"></div>

                        <a href="{{ route('admin.profile') }}" class="dropdown-item">

                            <i class="fas fa-user mr-2"></i> Profil

                        </a>

                        <div class="dropdown-divider"></div>

                        <a href="#" class="dropdown-item" onclick="event.preventDefault(); document.getElementById('logout-form').submit();">

                            <i class="fas fa-sign-out-alt mr-2"></i> Logout

                        </a>

                        <form id="logout-form" action="{{ route('admin.logout') }}" method="POST" class="d-none">

                            @csrf

                        </form>

                    </div>

                </li>

            </ul>

        </nav>

        <!-- Sidebar -->

        <aside class="main-sidebar sidebar-dark-primary elevation-4">

            <a href="#" class="brand-link">

                <span class="brand-text font-weight-light">AdminLTE</span>

            </a>

            <div class="sidebar">

                <nav class="mt-2">

                    <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu">

                        <li class="nav-item">

                            <a href="{{ route('admin.dashboard') }}" class="nav-link">

                                <i class="nav-icon fas fa-users"></i>

                                <p>Dashboard</p>

                            </a>

                        </li>

                        <li class="nav-item">

                            <a href="{{ route('admin.index') }}" class="nav-link">

                                <i class="nav-icon fas fa-users"></i>

                                <p>Kelola Admin</p>

                            </a>

                        </li>

                    </ul>

                </nav>

            </div>

        </aside>

        <!-- Content Wrapper -->

        <div class="content-wrapper mt-3">

            <div class="content-header">

                <div class="container-fluid">

                    <div class="row mb-2">

                        <div class="col-sm-6">

                            <h1 class="m-0">@yield('title')</h1>

                        </div>

                    </div>

                </div>

            </div>

            <section class="content">

                <div class="container-fluid">

                    <div class="card">

                        <div class="card-body">

                            @yield('content')

                        </div>

                    </div>

                </div>

            </section>

        </div>

        <!-- Footer -->

        <footer class="main-footer">

            <strong>Copyright &copy; 2024 <a href="#">AdminLTE</a>.</strong>

            All rights reserved.

        </footer>

    </div>

    <!-- Scripts -->

    <script src="{{ asset('plugins/jquery/jquery.min.js') }}"></script>

    <script src="{{ asset('plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>

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

    <!-- DataTables & Plugins -->

    <script src="{{ asset('plugins/datatables/jquery.dataTables.min.js') }}"></script>

    <script src="{{ asset('plugins/datatables-bs4/js/dataTables.bootstrap4.min.js') }}"></script>

    <script src="{{ asset('plugins/datatables-responsive/js/dataTables.responsive.min.js') }}"></script>

    <script src="{{ asset('plugins/datatables-responsive/js/responsive.bootstrap4.min.js') }}"></script>

    <script src="{{ asset('plugins/datatables-buttons/js/dataTables.buttons.min.js') }}"></script>

    <script src="{{ asset('plugins/datatables-buttons/js/buttons.bootstrap4.min.js') }}"></script>

    <script src="{{ asset('plugins/datatables-buttons/js/buttons.html5.min.js') }}"></script>

    <script src="{{ asset('plugins/datatables-buttons/js/buttons.print.min.js') }}"></script>

    <script src="{{ asset('plugins/datatables-buttons/js/buttons.colVis.min.js') }}"></script>

    <!-- Tambahkan script SweetAlert -->

    <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>

    <!-- Tambahkan script Data Table -->

    <script>

        $(document).ready(function() {

            let table = $('#example1').DataTable({

                "responsive": true,

                "lengthChange": true,

                "autoWidth": false,

                "paging": true,

                "ordering": true,

                "info": true,

                "searching": true,

                "buttons": ["copy", "csv", "excel", "pdf", "print", "colvis"],

                "language": {

                    "lengthMenu": "Tampilkan \_MENU\_ entri",

                    "zeroRecords": "Data tidak ditemukan",

                    "info": "Menampilkan \_START\_ sampai \_END\_ dari \_TOTAL\_ data",

                    "infoEmpty": "Tidak ada data tersedia",

                    "search": "Cari:",

                    "paginate": {

                        "previous": "Sebelumnya",

                        "next": "Selanjutnya"

                    }

                }

            });

            table.buttons().container().appendTo('#example1\_wrapper .dt-buttons');

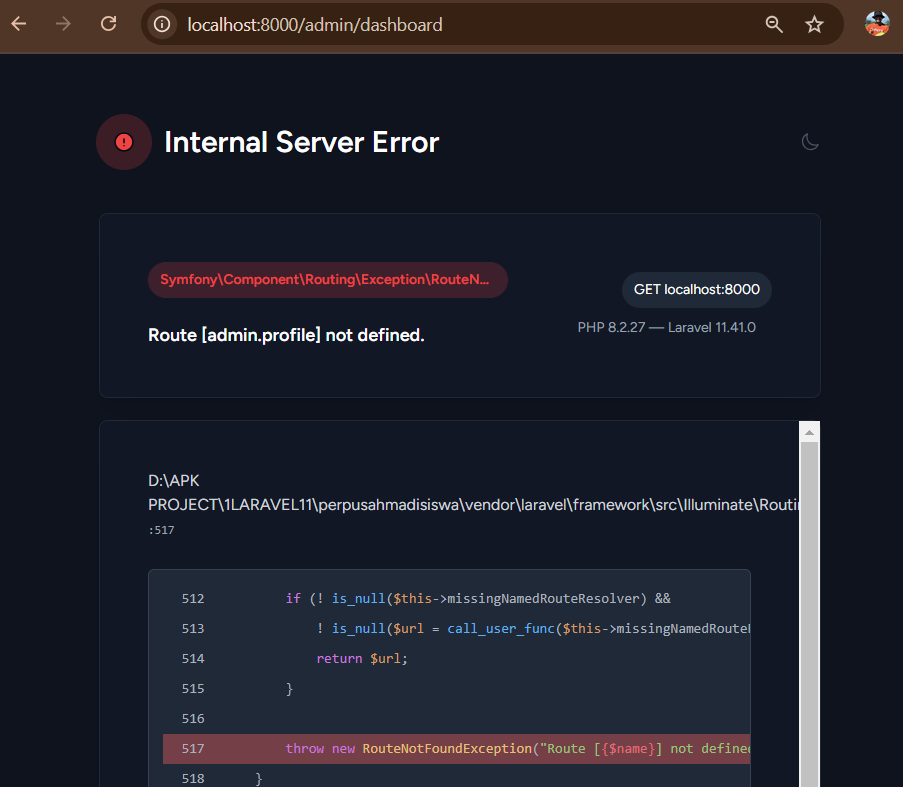
        });

    </script>

</body>

</html>

Setelah diupdate, tentunya akan ada error sbb karena kita belum mempunyai fitur profil seperti yang sudah kita cantumkan dalam komponen navbar admin.



**Solusi Perbaikan**

1. **Tambahkan route untuk profil admin** di routes/web.php.
2. **Pastikan route memiliki nama admin.profile**, agar sesuai dengan yang dipanggil di Blade.
3. **Buat controller & method untuk halaman profil** jika belum ada.

**📌 Langkah 1: Tambahkan Route di routes/web.php**

Tambahkan kode berikut di dalam file routes/web.php:

use App\Http\Controllers\ControllerAdmin;

Route::middleware(['auth:admin'])->group(function () {

Route::get('/admin/profile', [ControllerAdmin::class, 'profile'])->name('admin.profile');

});

📌 **Penjelasan:**

* **auth:admin** memastikan hanya admin yang bisa mengakses profil.
* **ControllerAdmin::class, 'profile'** akan memanggil method profile() di ControllerAdmin.
* **Nama route admin.profile** sesuai dengan yang digunakan di Blade.

**📌 Langkah 2: Tambahkan Method di ControllerAdmin.php**

Buka file **app/Http/Controllers/ControllerAdmin.php** lalu tambahkan method berikut:

public function profile()

{

return view('admin.profile'); // Pastikan ada file `profile.blade.php`

}

📌 **Penjelasan:**

* Method ini akan **menampilkan view admin.profile**.
* Pastikan ada file **resources/views/admin/profile.blade.php**.

**📌 Langkah 3: Buat File View profile.blade.php**

Buat file **resources/views/admin/profile.blade.php** lalu isi dengan:

@extends('layouts.appadmin')

@section('title', 'Profil Admin')

@section('content')

<div class="card">

<div class="card-header">

<h3 class="card-title">Profil Saya</h3>

</div>

<div class="card-body">

<div class="row">

<div class="col-md-4 text-center">

<img src="{{ asset(auth()->user()->foto ?? 'dist/img/default-avatar.png') }}" class="img-fluid img-circle" width="150">

</div>

<div class="col-md-8">

<table class="table table-bordered">

<tr><th>Nama</th><td>{{ auth()->user()->namaadmin }}</td></tr>

<tr><th>Email</th><td>{{ auth()->user()->email }}</td></tr>

<tr><th>Role</th><td>{{ auth()->user()->role }}</td></tr>

</table>

</div>

</div>

</div>

</div>

@endsection

# BAB IV PENGUJIAN APLIKASI

## Instrumen Validasi Pengujian Aspek Fungsionalitas Aplikasi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fitur** | **Pendukung** | **Urgensi/Tingkat Kepentingan** | **Hasil Pengujian** | |
| **Create** | **Lulus** | **Gagal** |
| Login | username | Read (tampil) |  |  |
|  | password |  |  |  |
| Dashboard | admin | Create |  |  |
|  | pelanggan | Read (tampil) |  |  |
|  | barang | Update |  |  |
|  | penjualan | Delete |  |  |
| Pengguna/Admin | namaadmin | Create |  |  |
|  | username | Read (tampil) |  |  |
|  | password | Update |  |  |
|  | alamat | Delete |  |  |
| Pelanggan | namapelanggan | Create |  |  |
|  | username | Read (tampil) |  |  |
|  | password | Update |  |  |
|  | alamat | Delete |  |  |
| Barang | namabarang | Create |  |  |
|  | kategori | Read (tampil) |  |  |
|  | merk | Update |  |  |
|  | suplier | Delete |  |  |
|  | stok |  |  |  |
|  | harga |  |  |  |
| Barang->Kategori | namakategori | Create |  |  |
|  |  | Read (tampil) |  |  |
|  |  | Update |  |  |
|  |  | Delete |  |  |
| Barang->Suplier | namakategori | Create |  |  |
|  |  | Read (tampil) |  |  |
|  |  | Update |  |  |
|  | alamat | Delete |  |  |
| Transaksi | nomor | Create |  |  |
|  | namapelanggan | Read (tampil) |  |  |
|  | namaadmin | Update |  |  |
|  | namabarang | Delete |  |  |
|  | jumlahbarang |  |  |  |
|  | total |  |  |  |
| Rekap/laporan | rekap-harian | Read (tampil) |  |  |
| Cetak | cetak | Read (tampil) |  |  |
| **Total** | | |  |  |

## Hasil Pengujian Aspek Fungsionalitas Aplikasi Penguji/Pembimbing

Berdasarkan pengujian aplikasi yang dilakukan oleh pembimbing maka didapatkan presentase sebagai berikut:

𝑗𝑢𝑚𝑙𝑎ℎ 𝑠𝑘𝑜𝑟 𝑦𝑎𝑛𝑔 𝑑𝑖𝑑𝑎𝑝𝑎𝑡

𝑃𝑟𝑒𝑠𝑒𝑛𝑡𝑎𝑠𝑒(%) = 𝑥 100

𝑗𝑢𝑚𝑙𝑎ℎ 𝑠𝑘𝑜𝑟 𝑚𝑎𝑘𝑠𝑖𝑚𝑎𝑙

= 31 / 31 𝑥 100 %

= 100 %

# BAB V PEMELIHARAAN APLIKASI

## Saran Pengembangan Aplikasi dari Penguji

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Fitur** | **Elemen** | **Saran / Pengembangan** | **Keterangan (diisi penguiji)** | |
| **Selesai** | **Belum** |
| *Contoh1* | *Login* | *Input Pasword* | *Perlu ada tombol tampil Password* | *Selesai* |  |
| *Contoh2* | *Absen* | *Input Absen* | *Tombol Absensi Kurang Besar* | *Selesai* |  |
| *Contoh3* | *Utama* | *-* | *Perlu Ada Fitur Rekap/Laporan* |  | *Belum* |
| *Contoh4* | *Laporan* | *Rekap/laporan* | *Fitur Laporan dibuat dengan priode tanggal ke tanggal* |  | *Belum* |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 7 |  |  |  |  |  |
| 8 |  |  |  |  |  |
| 9 |  |  |  |  |  |
| 10 |  |  |  |  |  |
| dst.. |  |  |  |  |  |

## Kesimpulan

Aplikasi Absensi berbasis web yang dikembangkan dengan menggunakan framework laravel Versi 11.2 dan PHP versi 8.2 telah selesai diuji dan siap digunakan sesuai dengan fungsinya sebagai instrumen absensi di lingkungan kantor Badan Kepegawaian Dan Pengembangan Sumber Daya Manusia (BKPSDM) Kabupaten Aceh Tamiang.

## Lampiran

## FORMULIR BIMBINGAN