# 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

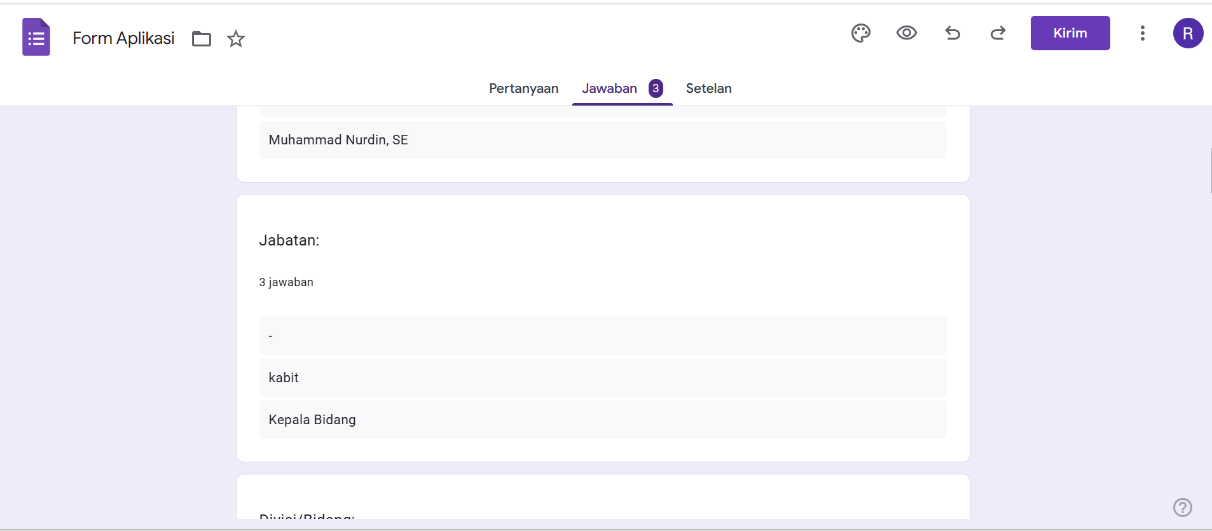
* Bagaimana tampilan login yang anda inginkan (boleh digambarkan)?
* Bagaimana tampilan dashboard/halaman depan yang anda inginkan (boleh digambarkan)?
* Bagaimana tampilan tampil barang yang anda inginkan (boleh digambarkan)?
* Bagaimana tampilan tambah barang depan yang anda inginkan (boleh digambarkan)?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

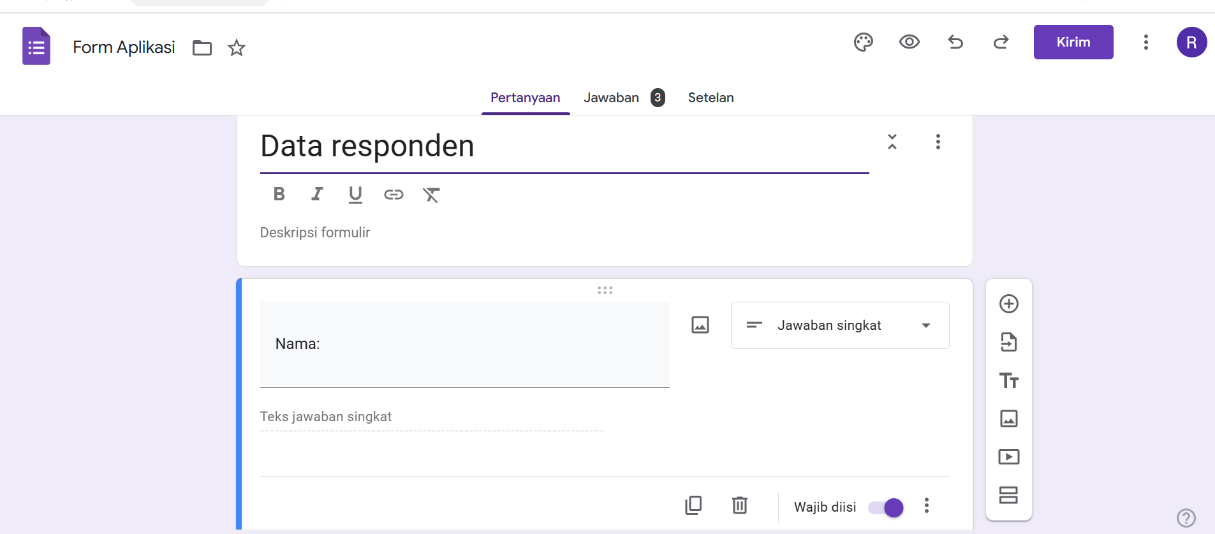
### Bagian 4: Masukan dan Saran

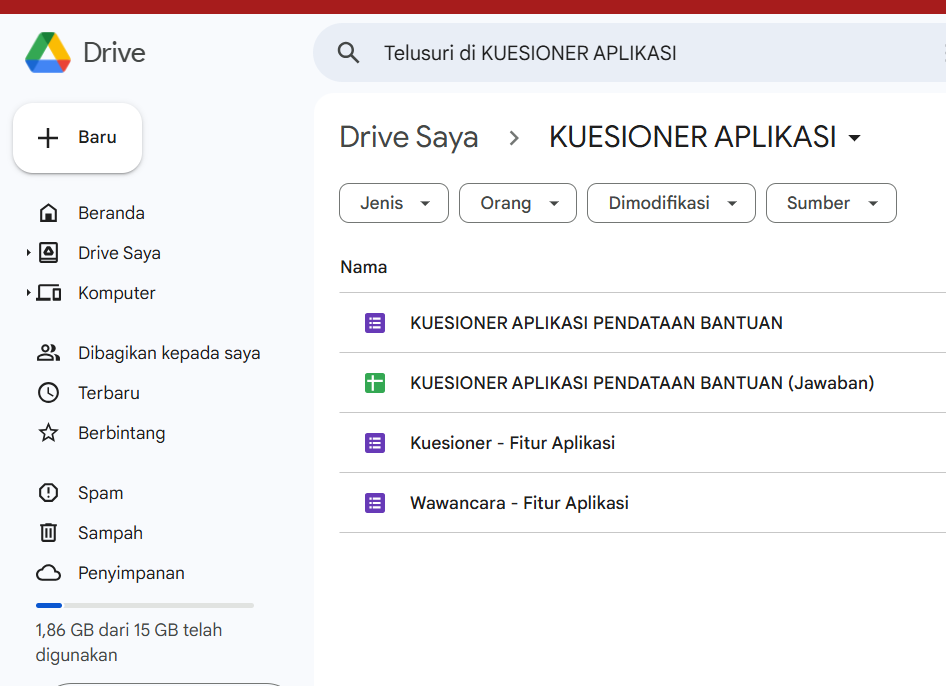
* Apa saran atau masukan tambahan yang Anda ingin berikan untuk pengembangan aplikasi ini?

## Dokumen Observasi

1. **Formulir dengan google form**





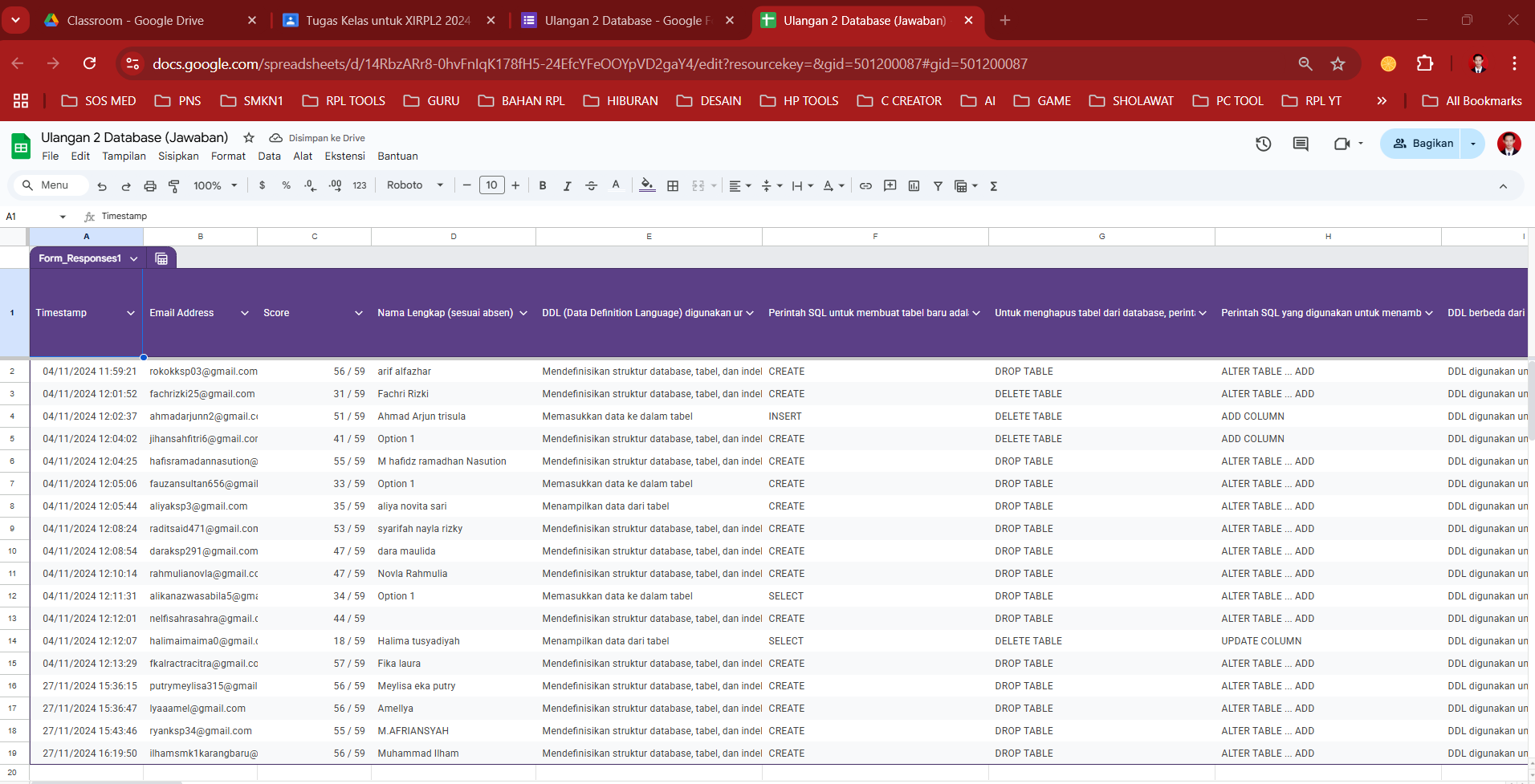


1. **Dokumentasi pengisian formulir**





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? |  |  |  |  |

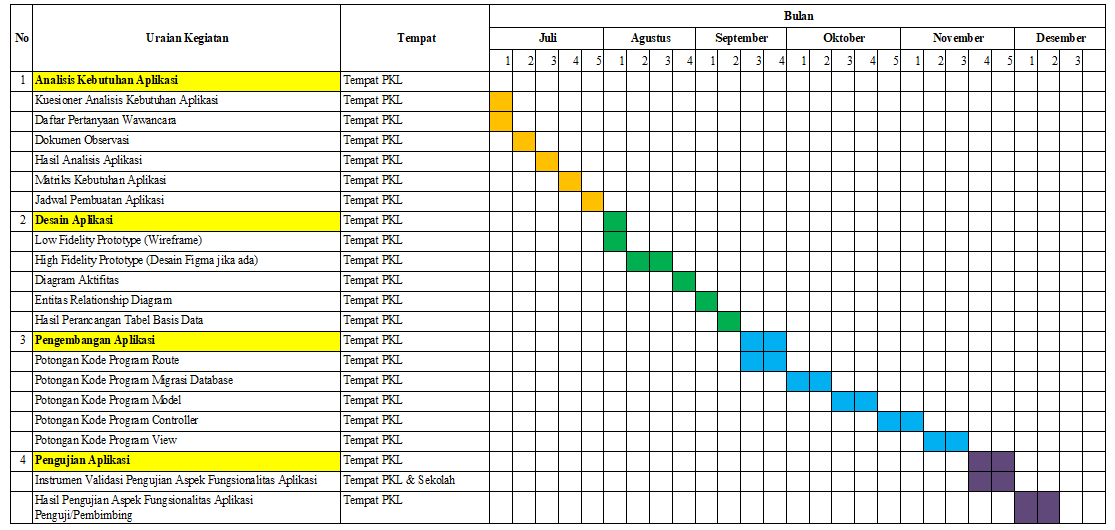
## Hasil Analisis Aplikasi

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **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). |

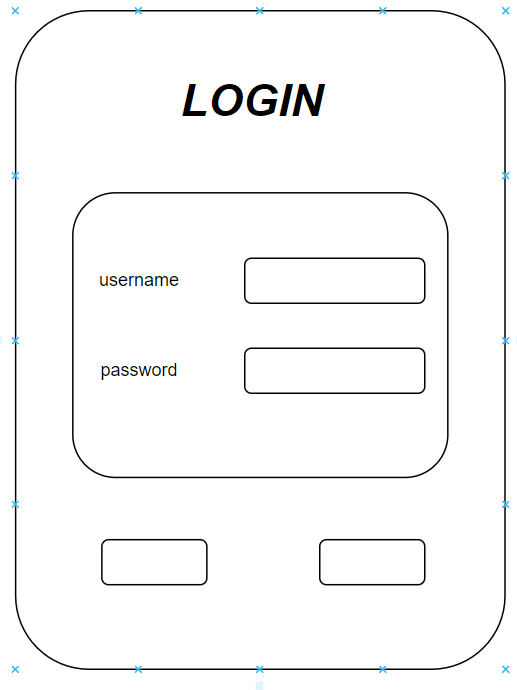
## Jadwal Pembuatan Aplikasi



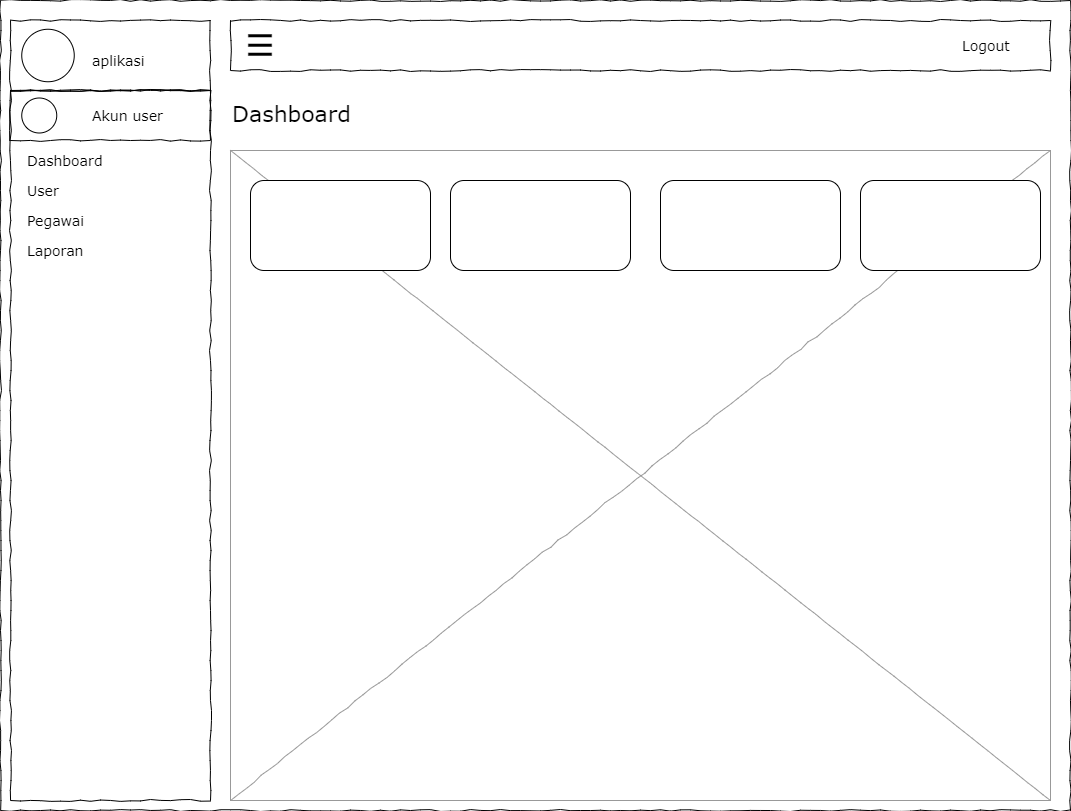
# BAB II Desain Aplikasi

## 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**

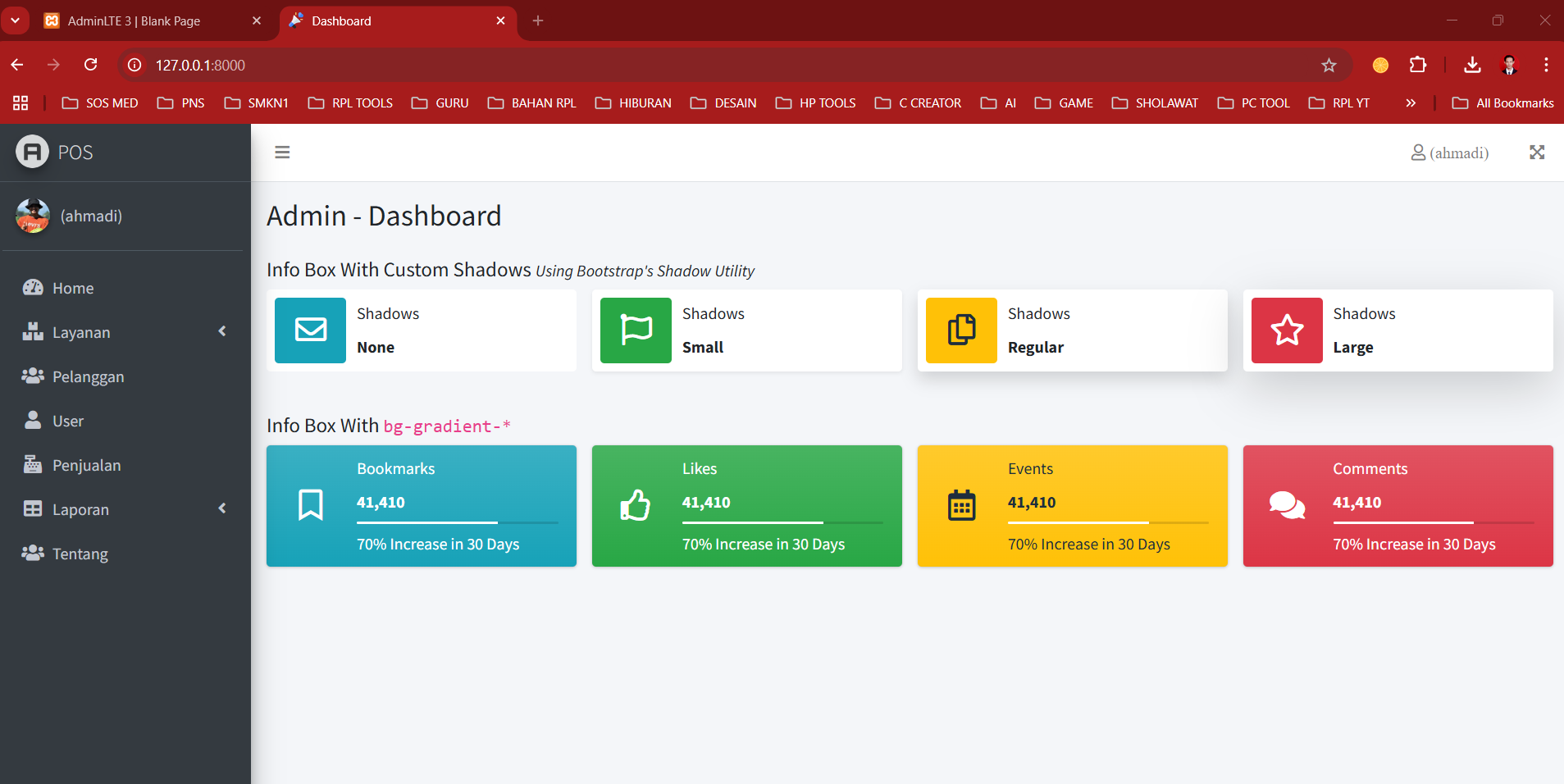
|  |  |
| --- | --- |
| Tampil | Tambah |
| Edit | Show |

* 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 |

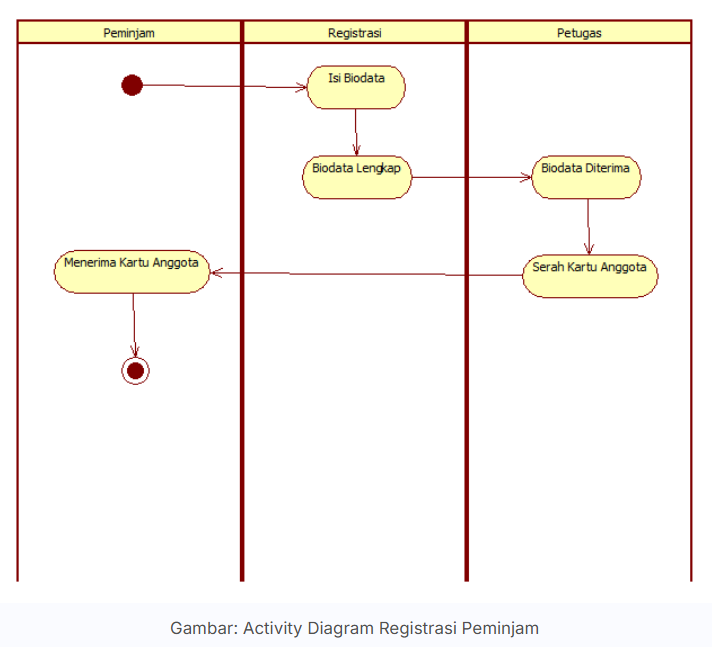
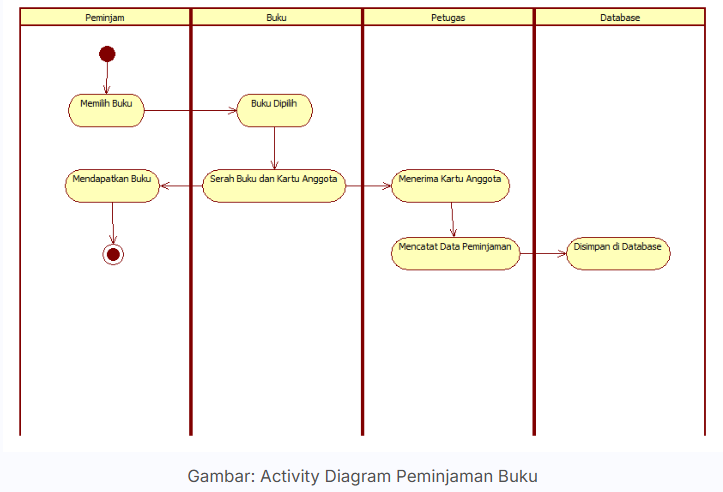
1. High **Fidelity Prototype Penjualan**

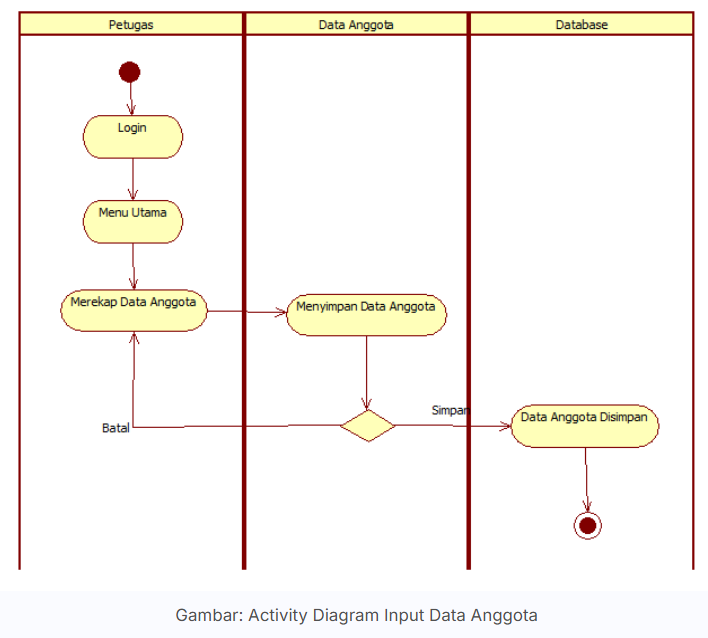
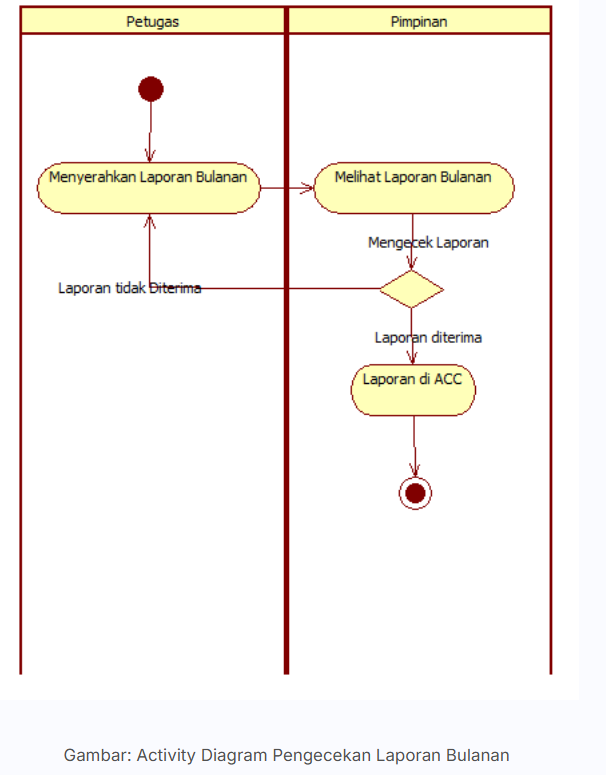
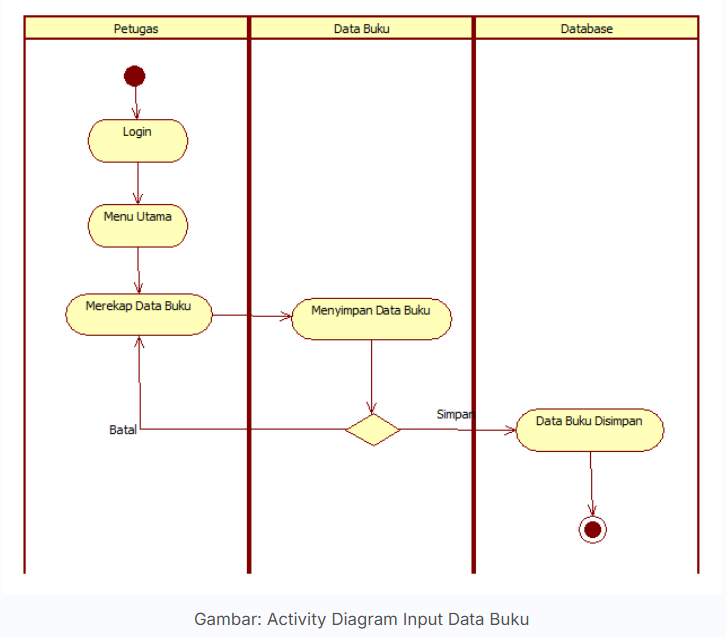
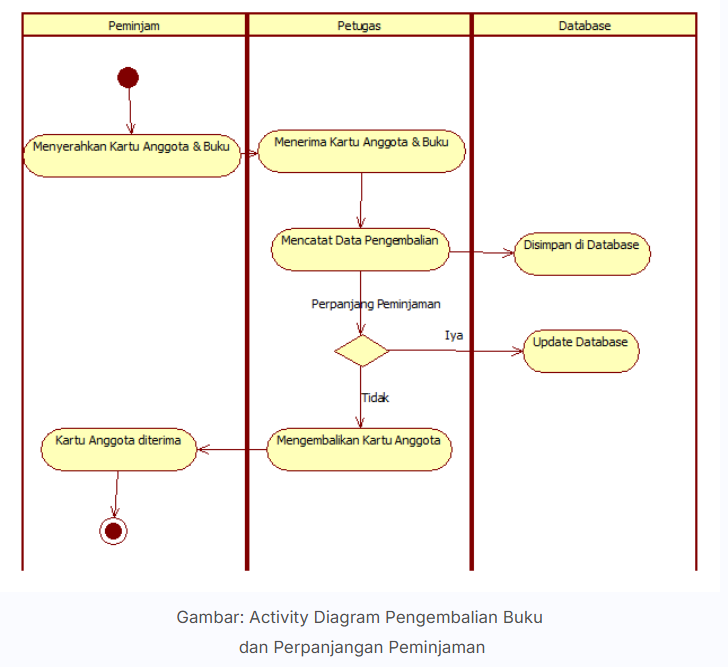
|  |  |
| --- | --- |
| Tampil | Tambah |
| Edit | Show |

1. High **Fidelity Prototype laporan**

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

## Diagram Aktifitas





## Entitas Relationship Diagram

Unf



1nf



2nf



3nf



## Migrasi Database

**Perlu Diingat Sebelumnya…!**

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

Ada 2 cara membuat file migrasi, yaitu dibuat dalam 1 file dan masing-masing file sesuai tabelnya.

1. **Model 1 file**

Biasanya, file migrasi ini dibuat secara otomatis menggunakan perintah Artisan:

php artisan make:migration create\_library\_tables

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

Buka file : create\_library\_tables.php

Dan copykan file berikut:

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateLibraryTables extends Migration

{

public function up()

{

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->string('role');

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

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

$table->timestamps();

});

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('judul');

$table->string('penerbit');

$table->string('pengarang');

$table->year('tahun');

$table->enum('status', ['tersedia', 'dipinjam'])->default('tersedia');

$table->timestamps();

});

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

$table->id();

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

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

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

$table->date('tanggalpinjam');

$table->date('tanggalkembali');

$table->integer('durasipeminjaman');

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

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

$table->timestamps();

});

}

public function down()

{

Schema::dropIfExists('peminjaman');

Schema::dropIfExists('buku');

Schema::dropIfExists('kategori');

Schema::dropIfExists('peminjam');

Schema::dropIfExists('admin');

}

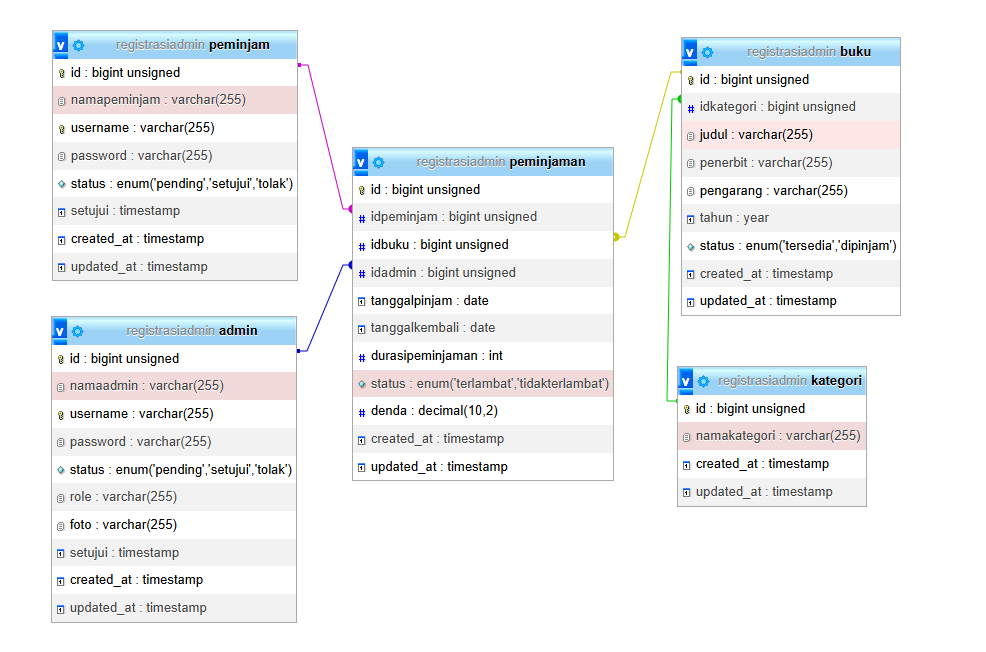
}

Jalankan perintah berikut di terminal untuk mengunggah migrasi ke database:

php artisan migrate

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

Jalankan perintah berikut di terminal untuk membuat file seeder:

php artisan make:seeder AdminSeeder

Ini akan membuat file AdminSeeder.php di dalam folder database/seeders/.

2. Edit File Seeder

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;

use Carbon\Carbon;

class AdminSeeder extends Seeder

{

/\*\*

\* Run the database seeds.

\*/

public function run(): void

{

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

[

'namaadmin' => 'Super Admin',

'username' => 'admin123',

'password' => Hash::make('password123'),

'status' => 'setujui',

'role' => 'admin',

'foto' => null,

'setujui' => Carbon::now(),

'created\_at' => Carbon::now(),

'updated\_at' => Carbon::now(),

],

[

'namaadmin' => 'Petugas Perpustakaan',

'username' => 'petugas01',

'password' => Hash::make('password123'),

'status' => 'setujui',

'role' => 'petugas',

'foto' => null,

'setujui' => Carbon::now(),

'created\_at' => Carbon::now(),

'updated\_at' => Carbon::now(),

],

]);

}

}

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);

}

}

3. Jalankan Seeder

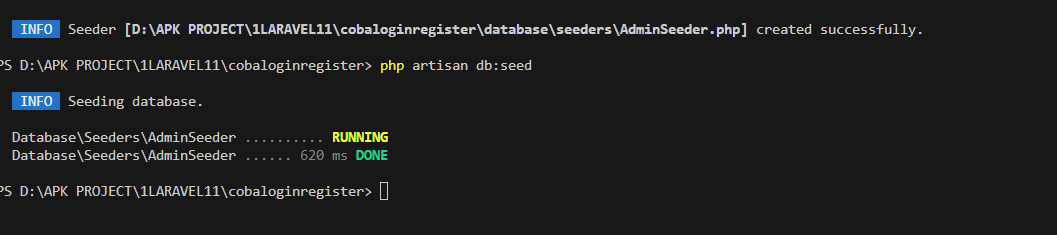
Untuk menjalankan seeder, gunakan perintah berikut:

php artisan db:seed

Jika ingin menjalankan AdminSeeder saja, gunakan:

php artisan db:seed --class=AdminSeeder

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



4. Menjalankan Semua Seeder Sekaligus

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

Buka file database/seeders/DatabaseSeeder.php, lalu tambahkan di dalam metode run():

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

Lalu jalankan:

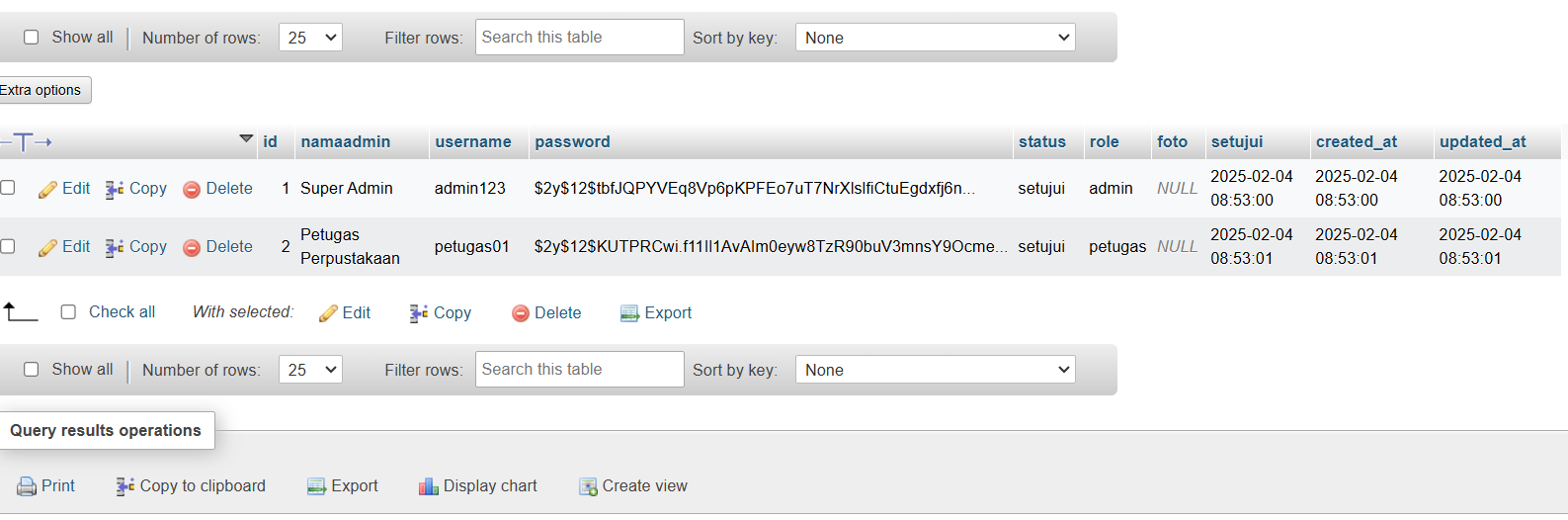
php artisan db:seed

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

## 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** | | |  |  |  |  |  |

1. **Membuat Layout Blade**:

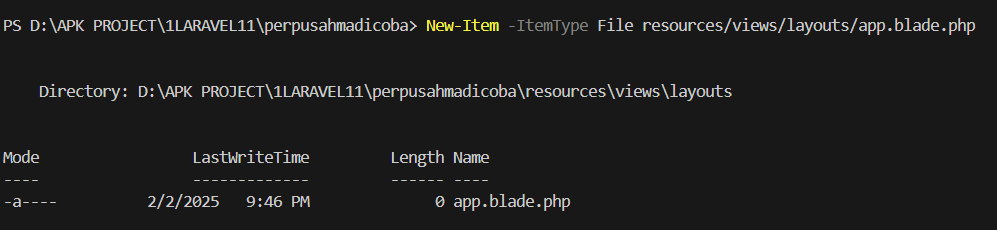
Gunakan Perintah yang Sesuai di Terminal Windows sbb:

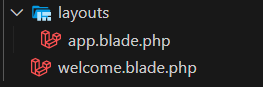
New-Item -ItemType File resources/views/layouts/app.blade.php

Gunakan Command Prompt (CMD):

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

jika berhasil akan tampil sebagai berikut:





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 dari Public -->

<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 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 dari Public -->

<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>

</body>

</html>

1. **Membuat Fitur Admin**:

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

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

<?php

use App\Http\Controllers\AdminController;

use Illuminate\Support\Facades\Route;

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

    return view('welcome');

});

Route::resource('admin', AdminController::class);

**Membuat Model dengan Migration**

Jalankan perintah berikut untuk membuat model Admin beserta migrasinya:

php artisan make:model Admin -m

Mengecek File yang Dibuat Model Admin:

app/Models/Admin.php

dan pastekan kode berikut:

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Admin extends Model

{

    use HasFactory;

    protected $table = 'admin'; // Nama tabel

    protected $fillable = [

        'namaadmin', 'username', 'password', 'role', 'foto'

    ];

    protected $hidden = [

        'password',

    ];

}

**Membuat Controller**

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

php artisan make:controller AdminController –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/AdminController.php

dan pastekan kode berikut:

<?php

namespace App\Http\Controllers;

use App\Models\Admin;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Hash;

use Illuminate\Support\Facades\Storage;

class AdminController extends Controller

{

    /\*\*

     \* Display a listing of the resource.

     \*/

    public function index()

    {

        $admins = Admin::all();

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

    }

    /\*\*

     \* Show the form for creating a new resource.

     \*/

    public function create()

    {

        return view('admin.create');

    }

    /\*\*

     \* Store a newly created resource in storage.

     \*/

    public function store(Request $request)

    {

        $request->validate([

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

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

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

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

            'foto' => 'nullable|image|mimes:jpeg,png,jpg|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,

            'foto' => $fotoPath

        ]);

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

    }

    /\*\*

     \* Display the specified resource.

     \*/

    public function show(Admin $admin)

    {

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

    }

    /\*\*

     \* Show the form for editing the specified resource.

     \*/

    public function edit(Admin $admin)

    {

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

    }

    /\*\*

     \* Update the specified resource in storage.

     \*/

    public function update(Request $request, Admin $admin)

    {

        $request->validate([

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

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

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

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

            'foto' => 'nullable|image|mimes:jpeg,png,jpg|max:2048'

        ]);

        $fotoPath = $admin->foto;

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

            if ($admin->foto) {

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

            }

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

        }

        $admin->update([

            'namaadmin' => $request->namaadmin,

            'username' => $request->username,

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

            'role' => $request->role,

            'foto' => $fotoPath

        ]);

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

    }

    /\*\*

     \* Remove the specified resource from storage.

     \*/

    public function destroy(Admin $admin)

    {

        if ($admin->foto) {

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

        }

        $admin->delete();

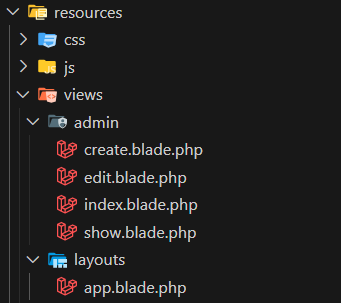
        return redirect()->route('admin.index')->with('success', 'Admin berhasil dihapus.');

    }

}

**Views:**

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



**a. index.blade.php**

@extends('layouts.app')

@section('content')

<div class="container">

<h2>Daftar Admin</h2>

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

<table class="table mt-3">

<tr>

<th>ID</th>

<th>Nama Admin</th>

<th>Username</th>

<th>Aksi</th>

</tr>

@foreach($admins as $admin)

<tr>

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

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

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

<td>

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

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

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

@csrf

@method('DELETE')

<button type="submit" class="btn btn-danger" onclick="return confirm('Yakin ingin menghapus?')">Hapus</button>

</form>

</td>

</tr>

@endforeach

</table>

</div>

@endsection

**b. create.blade.php**

@extends('layouts.app')

@section('content')

<div class="card">

    <div class="card-header">

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

    </div>

    <div class="card-body">

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

            @csrf

            <div class="form-group">

                <label>Nama Admin</label>

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

            </div>

            <div class="form-group">

                <label>Username</label>

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

            </div>

            <div class="form-group">

                <label>Password</label>

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

            </div>

            <div class="form-group">

                <label>Role</label>

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

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

                    <option value="user">User</option>

                </select>

            </div>

            <div class="form-group">

                <label>Foto</label>

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

            </div>

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

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

        </form>

    </div>

</div>

@endsection

**c. edit.blade.php**

@extends('layouts.app')

@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="form-group">

                <label>Nama Admin</label>

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

            </div>

            <div class="form-group">

                <label>Username</label>

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

            </div>

            <div class="form-group">

                <label>Password (Kosongkan jika tidak ingin mengubah)</label>

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

            </div>

            <div class="form-group">

                <label>Role</label>

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

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

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

                </select>

            </div>

            <div class="form-group">

                <label>Foto</label>

                @if($admin->foto)

                    <br>

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

                @endif

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

            </div>

            <button type="submit" class="btn btn-warning">Update</button>

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

        </form>

    </div>

</div>

@endsection

**d. show.blade.php**

@extends('layouts.app')

@section('content')

<div class="container">

<h2>Detail Admin</h2>

<p><strong>Nama Admin:</strong> {{ $admin->namaadmin }}</p>

<p><strong>Username:</strong> {{ $admin->username }}</p>

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

</div>

@endsection

Atau seperti ini:

@extends('layouts.app')

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

@section('content\_header')

    <h1>Detail Admin</h1>

@endsection

@section('content')

    <div class="card">

        <div class="card-header">

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

        </div>

        <div class="card-body">

            <div class="row">

                <!-- Kolom Foto di sebelah kiri -->

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

                    <strong>Foto:</strong><br>

                    @if($admin->foto)

                        <img src="{{ asset('storage/' . $admin->foto) }}" width="150" class="img-fluid">

                    @else

                        <p>Foto tidak tersedia</p>

                    @endif

                </div>

                <!-- Kolom Keterangan di sebelah kanan -->

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

                    <strong>Nama:</strong>

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

                    <strong>Username:</strong>

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

                    <strong>Role:</strong>

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

                </div>

            </div>

        </div>

        <div class="card-footer">

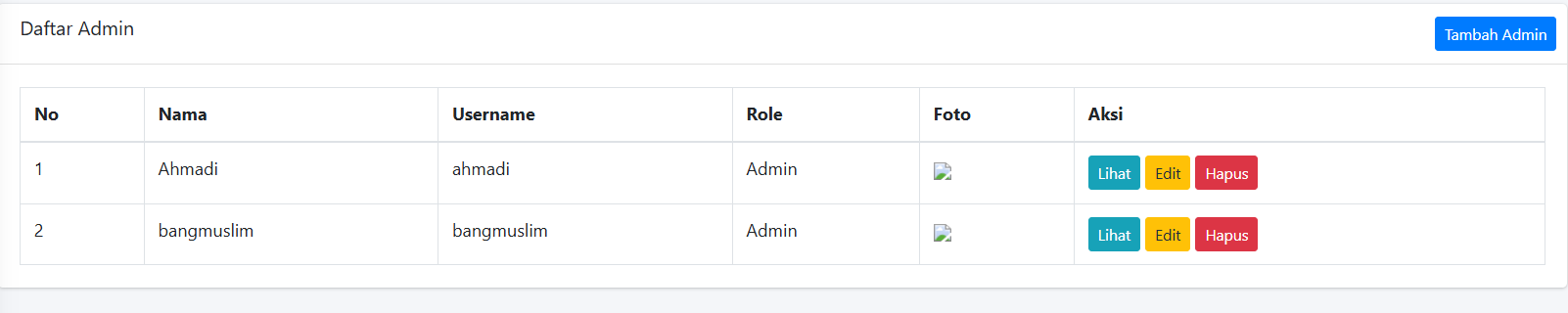
            <a href="{{ route('admin.index') }}" class="btn btn-primary">Kembali ke Daftar Admin</a>

        </div>

    </div>

@endsection

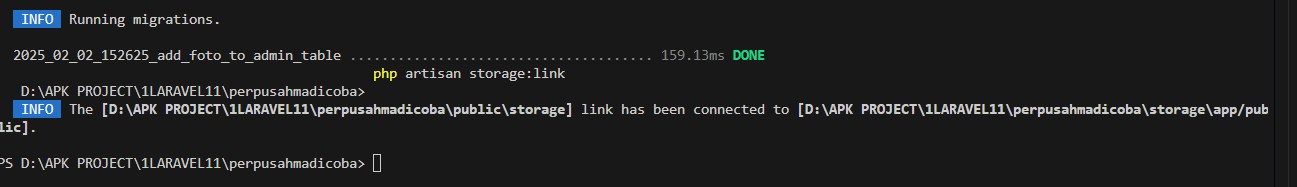
Pada saat ini mungkin foto belum dapat ditampilkan

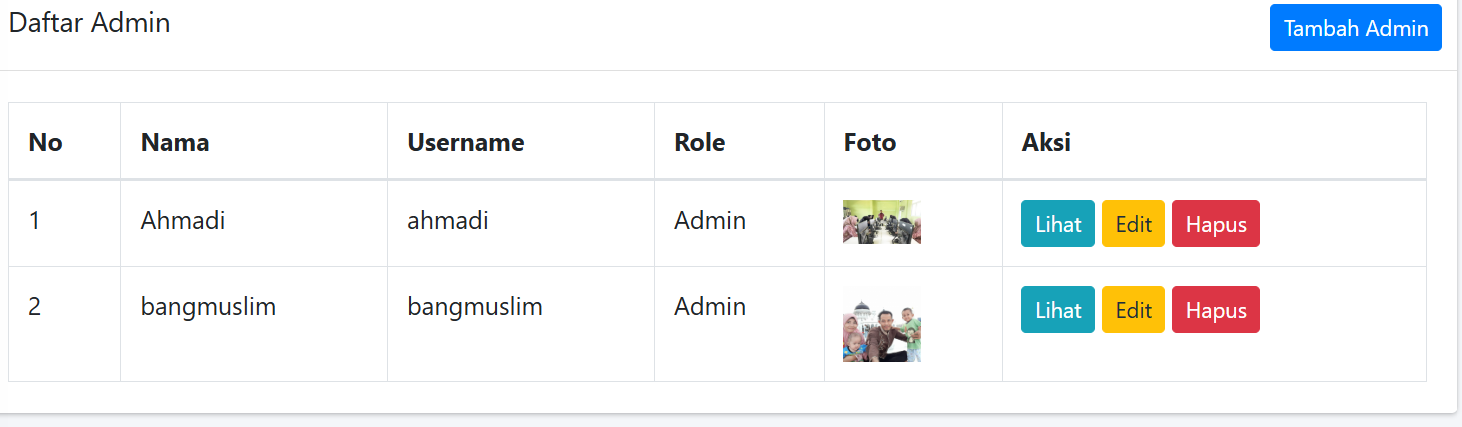


Pastikan bahwa file foto benar-benar ada di dalam direktori storage/app/public/ dan bahwa Anda telah menjalankan perintah berikut:

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**

Untuk membuat fitur login kita juga harus menerapkan prinsip MODEL, VIEW, CONTROLLER (MVC), 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:

php

SalinEdit

use App\Http\Controllers\AuthController;

Route::get('/login', [AuthController::class, 'showLogin'])->name('login');

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

Route::post('/logout', [AuthController::class, 'logout'])->name('logout');

Route::get('/register', [AuthController::class, 'showRegister'])->name('register');

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

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

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

return view('dashboard');

})->name('dashboard');

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

Route::get('/admin/users', [AuthController::class, 'showUsers'])->name('admin.users');

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

});

});

**2. Buat Middleware untuk Admin**

Jalankan perintah:

sh

SalinEdit

php artisan make:middleware AdminMiddleware

Edit file app/Http/Middleware/AdminMiddleware.php:

php

SalinEdit

namespace App\Http\Middleware;

use Closure;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

class AdminMiddleware

{

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

{

if (Auth::check() && Auth::user()->role === 'admin') {

return $next($request);

}

return redirect('/dashboard')->with('error', 'Akses hanya untuk admin.');

}

}

Tambahkan ke app/Http/Kernel.php:

php

SalinEdit

protected $routeMiddleware = [

'admin' => \App\Http\Middleware\AdminMiddleware::class,

];

**Membuat file menggunakan VS Code (Rekomendasi)**

Jika kamu sudah menginstal **Visual Studio Code**, gunakan perintah ini untuk membuka file:

sh

SalinEdit

code app/Http/Kernel.php

Jika perintah code tidak dikenali, buka **VS Code**, lalu tekan **CTRL + Shift + P**,

ketik **"Shell Command: Install 'code' command in PATH"**, lalu coba lagi.

**3. Buat Controller**

Jalankan perintah:

sh

SalinEdit

php artisan make:controller AuthController

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

php

SalinEdit

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Hash;

use App\Models\Admin;

class AuthController extends Controller

{

public function showLogin()

{

return view('auth.login');

}

public function login(Request $request)

{

$credentials = $request->validate([

'username' => 'required',

'password' => 'required',

]);

$user = Admin::where('username', $credentials['username'])->first();

if ($user && Hash::check($credentials['password'], $user->password)) {

if ($user->role === 'user' && !$user->approved) {

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

}

Auth::login($user);

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

}

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

}

public function logout()

{

Auth::logout();

return redirect('/login');

}

public function showRegister()

{

return view('auth.register');

}

public function register(Request $request)

{

$request->validate([

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

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

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

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

]);

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

Admin::create([

'namaadmin' => $request->namaadmin,

'username' => $request->username,

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

'role' => 'user', // Default sebagai user

'foto' => $fotoPath,

'approved' => false, // Perlu persetujuan admin

]);

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

}

public function showUsers()

{

$users = Admin::where('role', 'user')->where('approved', false)->get();

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

}

public function approveUser($id)

{

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

$user->approved = true;

$user->save();

return back()->with('success', 'User disetujui.');

}

}

**4. Buat Migration untuk Menambahkan Kolom "Approved"**

Jalankan perintah:

sh

SalinEdit

php artisan make:migration add\_approved\_to\_admin\_table --table=admin

Edit file migration di database/migrations:

php

SalinEdit

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration {

public function up()

{

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

$table->boolean('approved')->default(false);

});

}

public function down()

{

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

$table->dropColumn('approved');

});

}

};

Jalankan perintah:

sh

SalinEdit

php artisan migrate

**5. Buat Halaman Login (resources/views/auth/login.blade.php)**

Berikut ada beberapa cara membuat filenya di resources blade

**1. Buka Terminal di VS Code**

* **Cara 1**: Tekan **Ctrl + `** (tombol backtick, di bawah Esc) untuk membuka terminal.
* **Cara 2**: Klik **View** → **Terminal** di menu VS Code.

Pastikan terminal sudah berada di direktori root proyek Laravel. Jika belum, pindah ke folder proyek Laravel-mu:

sh

SalinEdit

cd /path/to/your-laravel-project

Ganti /path/to/your-laravel-project dengan path proyek Laravel yang benar.

**2. Buat Direktori auth (Jika Belum Ada)**

Jalankan perintah berikut untuk membuat folder auth di dalam resources/views:

sh

SalinEdit

mkdir -p resources/views/auth

Di Windows CMD, gunakan:

sh

SalinEdit

mkdir resources\views\auth

**3. Buat File login.blade.php**

Jalankan perintah berikut di terminal:

sh

SalinEdit

type nul > resources/views/auth/login.blade.php

Atau di PowerShell:

sh

SalinEdit

New-Item resources/views/auth/login.blade.php

Atau jika menggunakan **Git Bash**:

sh

SalinEdit

touch resources/views/auth/login.blade.php

**4. Edit File di VS Code**

Setelah file dibuat, buka di VS Code dengan perintah:

sh

SalinEdit

code resources/views/auth/login.blade.php

Jika tidak bisa, buka file langsung melalui **File Explorer** atau melalui VS Code dengan menavigasi ke **resources/views/auth/login.blade.php**.

SalinEdit

<!DOCTYPE html>

<html lang="id">

<head>

    <meta charset="UTF-8">

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

    <title>Login | AdminLTE</title>

    <!-- AdminLTE & Bootstrap CSS dari Public -->

    <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') }}">

    <style>

        body {

            font-family: 'Poppins', sans-serif;

        }

    </style>

</head>

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

    <div class="login-box">

        <div class="login-logo">

            <a href="#"><b>Perpustakaan</b>App</a>

        </div>

        <!-- /.login-logo -->

        <div class="card">

            <div class="card-body login-card-body">

                <p class="login-box-msg">Masukkan Username dan Password</p>

                @if ($errors->any())

                <div class="alert alert-danger">

                    <ul>

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

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

                        @endforeach

                    </ul>

                </div>

                @endif

                <form action="{{ url('/login') }}" 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="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-6">

                            <a href="{{ route('register') }}" class="btn btn-secondary btn-block">Daftar</a>

                        </div>

                        <div class="col-6">

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

                        </div>

                    </div>

                </form>

            </div>

        </div>

    </div>

    <!-- AdminLTE & Bootstrap JS dari Public -->

    <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>

</body>

</html>

**6. Buat Halaman Registrasi (resources/views/auth/register.blade.php)**

html

SalinEdit

<!DOCTYPE html>

<html lang="id">

<head>

    <meta charset="UTF-8">

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

    <title>Login | AdminLTE</title>

    <!-- AdminLTE & Bootstrap CSS dari Public -->

    <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') }}">

    <style>

        body {

            font-family: 'Poppins', sans-serif;

        }

    </style>

</head>

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

    <div class="register-box">

        <div class="register-logo">

            <a href="#"><b>Perpustakaan</b>App</a>

        </div>

        <div class="card">

            <div class="card-body register-card-body">

                <p class="login-box-msg">Buat Akun Baru</p>

                @if ($errors->any())

                <div class="alert alert-danger">

                    <ul>

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

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

                        @endforeach

                    </ul>

                </div>

                @endif

                <form action="{{ url('/register') }}" 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-circle"></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">

                        <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-6">

                            <a href="{{ route('login') }}" class="btn btn-secondary btn-block">Login</a>

                        </div>

                        <div class="col-6">

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

                        </div>

                    </div>

                </form>

            </div>

        </div>

    </div>

    <!-- AdminLTE & Bootstrap JS dari Public -->

    <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>

</body>

</html>

Tambahkan **implements Authenticatable** pada model Admin dan gunakan **Authenticatable dari Laravel**.

**1️⃣ Buka Model Admin.php dan Perbaiki**

🛠 **Edit file:** app/Models/Admin.php

php

SalinEdit

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Foundation\Auth\User as Authenticatable;

class Admin extends Authenticatable

{

use HasFactory;

protected $table = 'admin'; // Pastikan nama tabel sesuai

protected $fillable = ['namaadmin', 'username', 'password', 'role', 'foto'];

protected $hidden = ['password'];

}

📌 **Penjelasan Perbaikan:**

* **Menggunakan Authenticatable bawaan Laravel**, bukan Model biasa.
* **Menggunakan protected $hidden = ['password'];** agar password tidak bocor saat mengambil data.
* **Menggunakan protected $fillable** untuk menghindari Mass Assignment.

**2️⃣ Pastikan Guard & Provider di config/auth.php**

🛠 **Buka config/auth.php**

Cari bagian:

php

SalinEdit

'providers' => [

'users' => [

'driver' => 'eloquent',

'model' => App\Models\Admin::class, // Ubah ke model Admin

],

],

📌 **Pastikan model mengarah ke App\Models\Admin::class** karena kita ingin **Admin sebagai user login**.

**3️⃣ Jalankan php artisan config:clear**

Setelah mengedit config/auth.php, **bersihkan cache konfigurasi** agar perubahan diterapkan:

sh

SalinEdit

php artisan config:clear

Jika masih tidak bisa, jalankan juga:

sh

SalinEdit

php artisan cache:clear

php artisan config:cache

**🔥 Hasil Perbaikan**

✅ **Login sekarang berfungsi dengan model Admin.**  
✅ **Laravel bisa mengenali model Admin sebagai pengguna yang bisa login.**  
✅ **Tidak ada lagi error must be of type Illuminate\Contracts\Auth\Authenticatable**

# 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