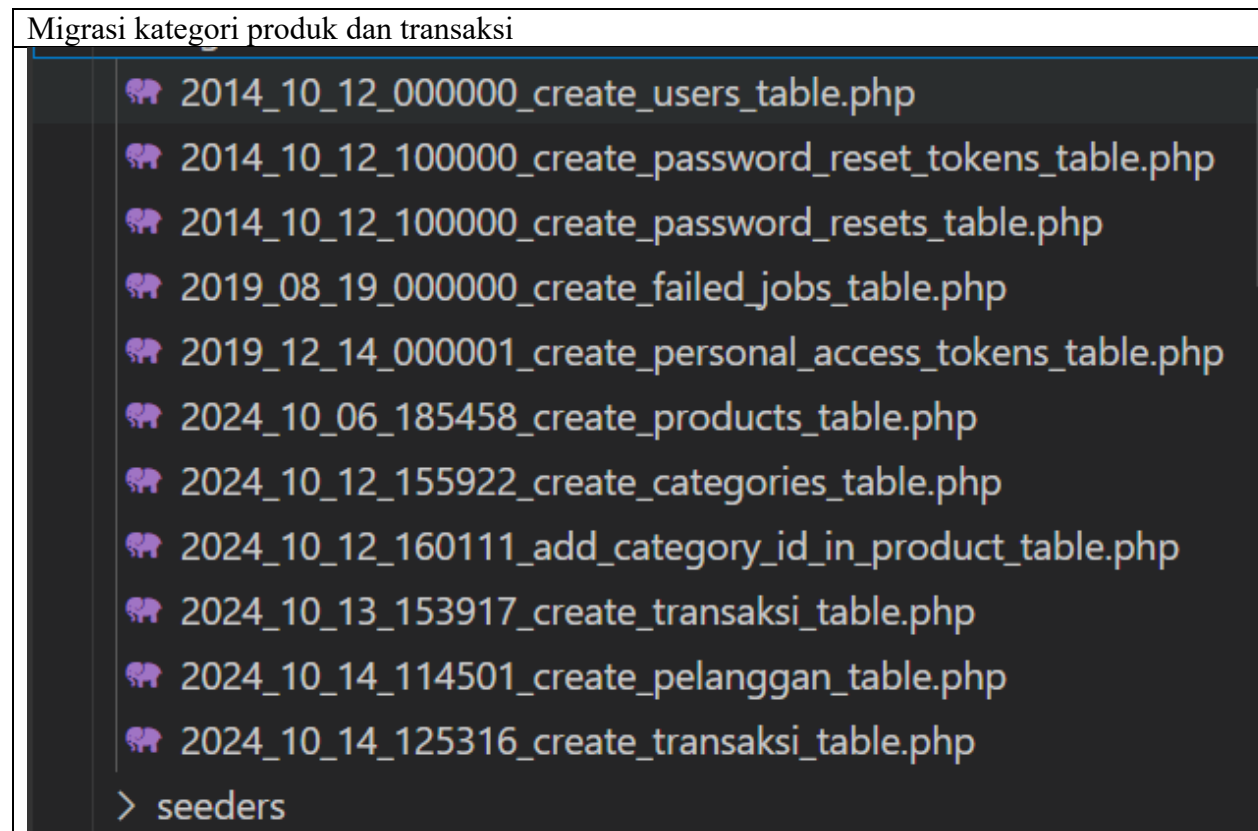


Nama : Misyhel Oktavia Br Nababan

Kelas : PW 1

Kampus : Universitas Methodist Indonesia Medan

Membuat migrasi kategori , produk dan transaksi



Dengan command terminal :

kategori :

```
php artisan make:migration create_kategori_table --create=kategori
```

produk:

```
php artisan make:migration create_produk_table --create=produk
```

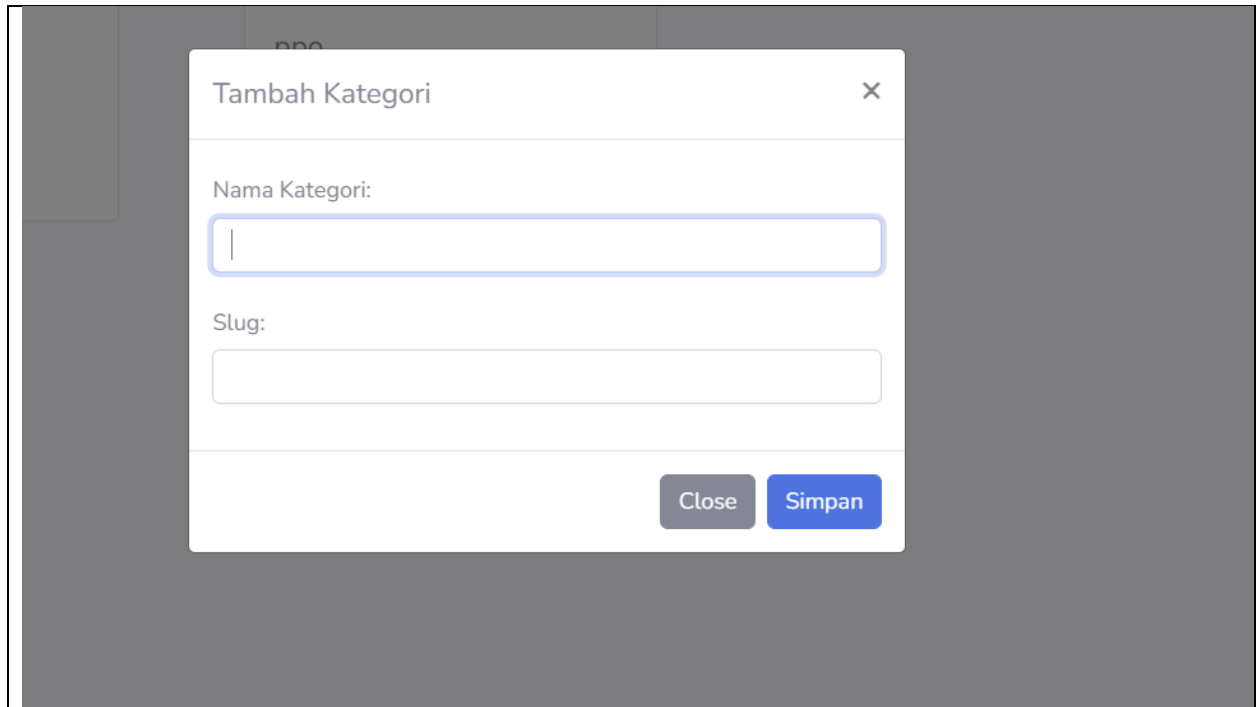
Transaksi :

```
php artisan make:migration create_transaksi_table --create=transaksi
```

Melakukan operasi database untuk kategori , produk dan pelanggan

Operasi kategori :

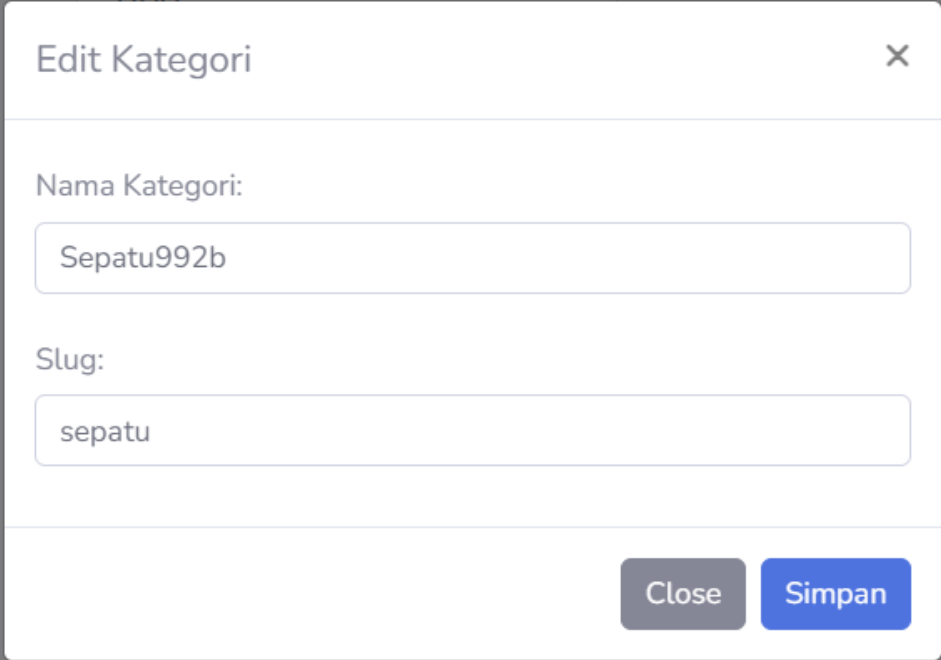
Tambah kategori :



The image shows a modal window titled "Tambah Kategori" (Add Category) with a close button (X) in the top right corner. The form contains two input fields: "Nama Kategori:" (Category Name) and "Slug:". The "Nama Kategori:" field is currently active, indicated by a blue border and a vertical cursor. At the bottom right of the modal, there are two buttons: "Close" (grey) and "Simpan" (blue).

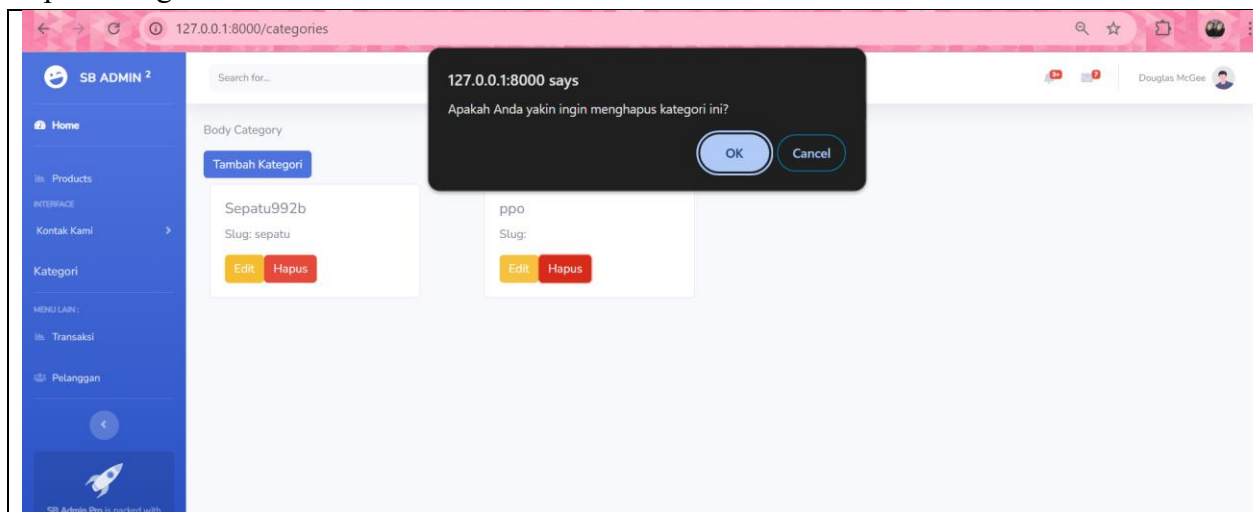
Tambah Kategori	
Nama Kategori:	<input type="text"/>
Slug:	<input type="text"/>
<div>Close Simpan</div>	

Edit Kategori :

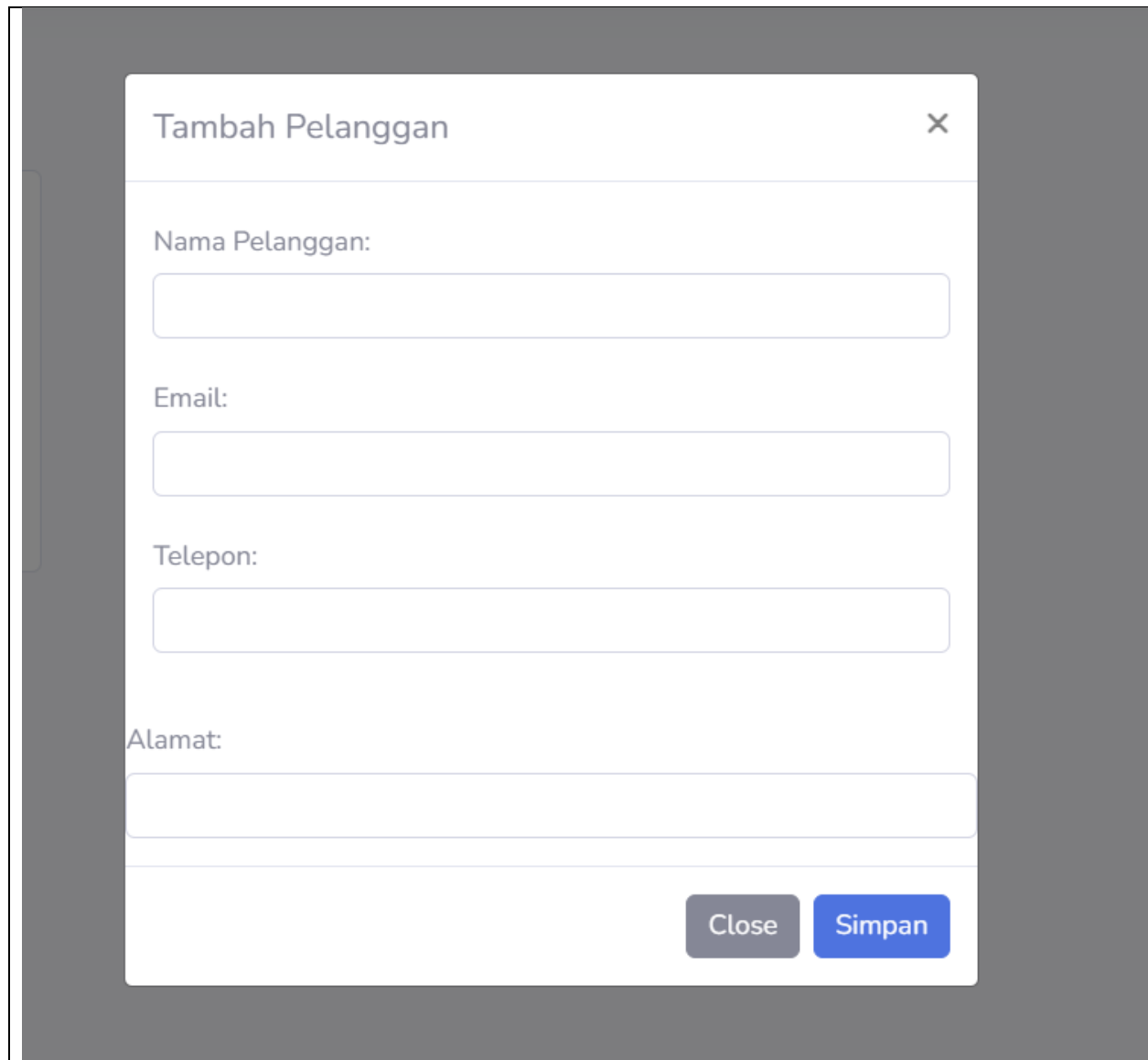


A modal window titled "Edit Kategori" with a close button (X) in the top right corner. It contains two input fields: "Nama Kategori:" with the value "Sepatu992b" and "Slug:" with the value "sepatu". At the bottom right, there are two buttons: "Close" (grey) and "Simpan" (blue).

Hapus Kategori :



Operasi pelanggan :
Tambah pelanggan



Tambah Pelanggan X

Nama Pelanggan:

Email:

Telepon:

Alamat:

Close Simpan

Edit Pelanggan :

Edit Pelanggan

Nama Pelanggan:

Misyhel oktavia br nababan

Email:

misyhelnbnn@gmail.com

Telepon:

0812930

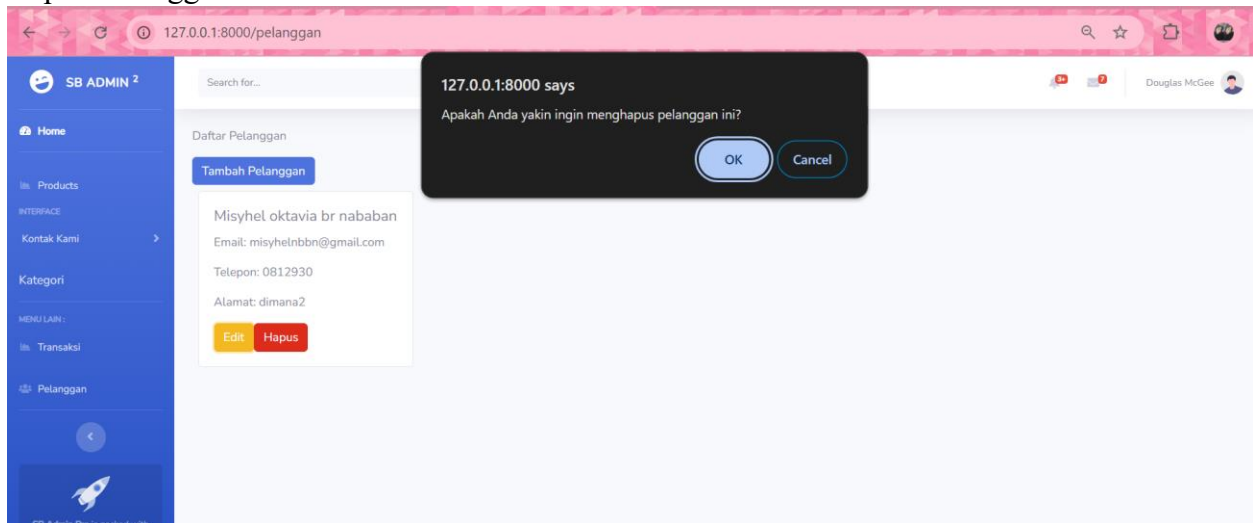
Telepon:

dimana2

Close

Simpan

Hapus Pelanggan :



Persiapan membuat migrasi

Migrasi table product

```
<?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('products', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('description');
            $table->string('price');
            $table->string('image')->nullable();

            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
}
```

```
*/  
public function down(): void  
{  
    Schema::dropIfExists('products');  
}  
};
```

Migrasi table category

```
<?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('categories', function (Blueprint $table) {  
            $table->id();  
            $table->string('name');  
            $table->string('slug')->unique();  
            $table->timestamps();  
        });  
    }  
  
    /**  
     * Reverse the migrations.  
     */  
    public function down(): void  
    {  
        Schema::dropIfExists('categories');  
    }  
};
```

Migrasi add category id in product table

```
<?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::table('products', function (Blueprint $table) {
            $table->foreignId('category_id')->after('id')->constrained('categories')->nullable();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::table('products', function (Blueprint $table) {
            $table->dropForeign(['category_id']);
            $table->dropColumn('category_id');
        });
    }
};
```

Migrasi transaksi

```
<?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()
{
    Schema::create('transaksi', function (Blueprint $table) {
        $table->id(); // Kolom ID sebagai primary key
        $table->unsignedBigInteger('produk_id'); // Foreign key untuk produk
        $table->integer('jumlah'); // Jumlah produk yang dibeli
        $table->decimal('total_harga', 10, 2); // Total harga transaksi
        $table->timestamps(); // Kolom created_at dan updated_at

        // Menambahkan foreign key constraint
        $table->foreign('produk_id')->references('id')->on('products')->onDelete('cascade');
    });
}

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

```

Migrasi table pelanggan

```

<?php

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

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('pelanggan', function (Blueprint $table) {
            $table->id();
            $table->string('nama');

```

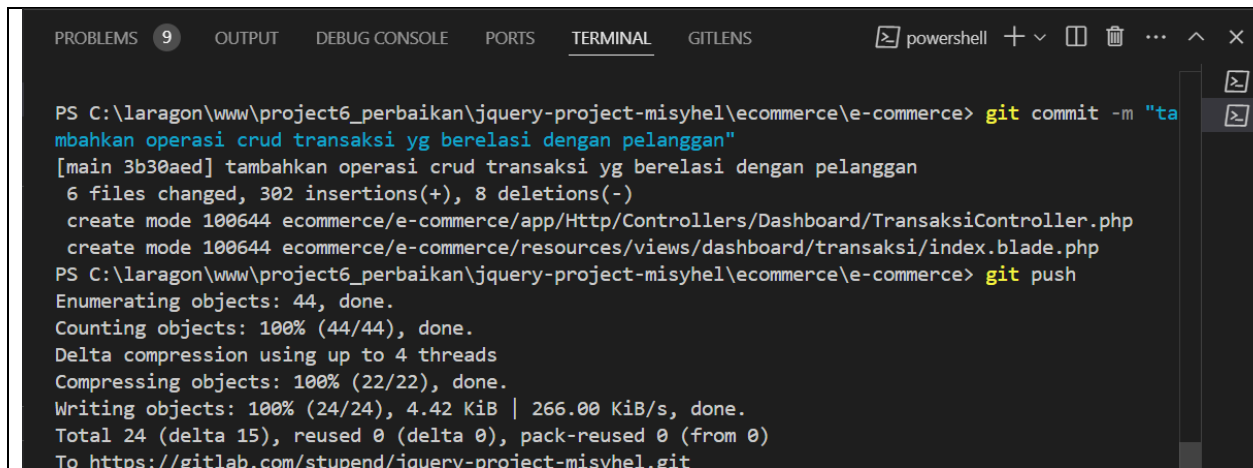
```

        $table->string('email')->unique();
        $table->string('telepon');
        $table->text('alamat')->nullable();
        $table->timestamps();
    });
}

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

```

Hasil commit dan push ke repo :

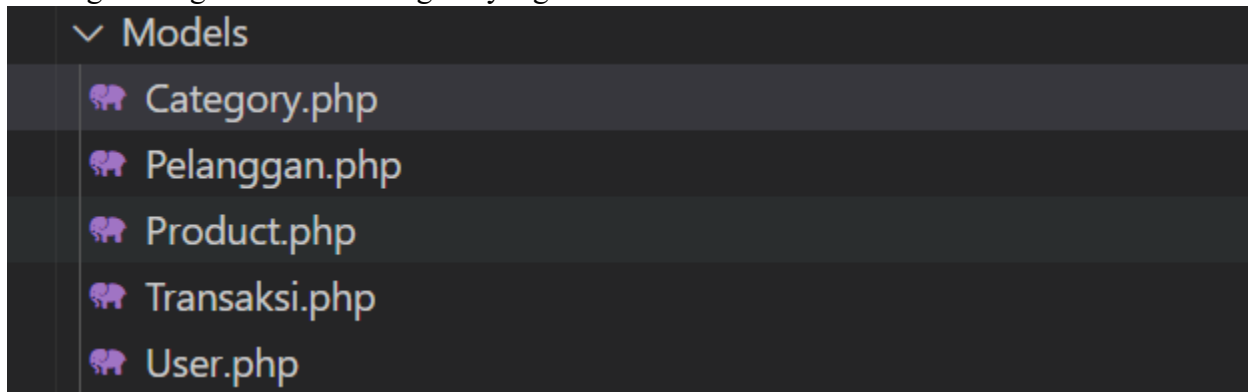


```

PS C:\laragon\www\project6_perbaikan\jquery-project-misyhel\ecommerce\e-commerce> git commit -m "tambahkan operasi crud transaksi yg berelasi dengan pelanggan"
[main 3b30aed] tambahkan operasi crud transaksi yg berelasi dengan pelanggan
6 files changed, 302 insertions(+), 8 deletions(-)
create mode 100644 ecommerce/e-commerce/app/Http/Controllers/Dashboard/TransaksiController.php
create mode 100644 ecommerce/e-commerce/resources/views/dashboard/transaksi/index.blade.php
PS C:\laragon\www\project6_perbaikan\jquery-project-misyhel\ecommerce\e-commerce> git push
Enumerating objects: 44, done.
Counting objects: 100% (44/44), done.
Delta compression using up to 4 threads
Compressing objects: 100% (22/22), done.
Writing objects: 100% (24/24), 4.42 KiB | 266.00 KiB/s, done.
Total 24 (delta 15), reused 0 (delta 0), pack-reused 0 (from 0)
To https://gitlab.com/stupend/iquerv-project-misvhel.git

```

Masing-masing model untuk migrasi yang dibuat :



Atribut fillable dan table masing-masing model

Model Category

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Category extends Model
{
    use HasFactory;

    protected $fillable = ['name', 'description'];
    protected $guarded = ['id'];

    // Relasi One-to-Many dengan Produk
    public function category()
    {
        return $this->hasMany(Product::class);
    }
}
```

Model pelanggan

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Pelanggan extends Model
{
    use HasFactory;

    protected $table = 'pelanggan';

    // Kolom yang dapat diisi
    protected $fillable = [
        'nama',
        'email',
        'telepon',
        'alamat',
    ];
}
```

Model product

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;

class Product extends Model
{
    use HasFactory;
    protected $guarded=['id'];
    public function category() :BelongsTo
    {
        return $this->belongsTo(Category::class,'category_id');
    }

    public function transaksi()
```

```
{
    return $this->hasMany(Transaksi::class);
}
}
```

Model transaksi

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Transaksi extends Model
{
    use HasFactory;

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

    protected $fillable = [
        'pelanggan_id',
        'total_harga',
        'tanggal_transaksi',
        'status',
    ];

    // Relasi dengan model Pelanggan
    public function pelanggan()
    {
        return $this->belongsTo(Pelanggan::class);
    }
}
```

Relasi tabel category dan product :

```
<?php

namespace App\Models;
```

```

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;

class Product extends Model
{
    use HasFactory;
    protected $guarded=['id'];
    public function category() :BelongsTo
    {
        return $this->belongsTo(Category::class,'category_id');
    }

    public function transaksi()
    {
        return $this->hasMany(Transaksi::class);
    }
}

```

Relasi transaksi dan pelanggan

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Transaksi extends Model
{
    use HasFactory;

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

    protected $fillable = [
        'pelanggan_id',
        'total_harga',
        'tanggal_transaksi',
        'status',
    ];

    // Relasi dengan model Pelanggan
    public function pelanggan()

```

```
{  
    return $this->belongsTo(Pelanggan::class);  
}  
}
```

Melakukan operasi produk

Tambah produk :

Tambah Produk

Nama Produk:

Nama Kategori:

Pilih Kategori Produk

Deskripsi:

Harga:

Close Save changes

Edit Produk

Edit Produk

Nama Produk:

Sepatu22

Nama Kategori:

Sepatu992b

Deskripsi:

sepatu gaming1

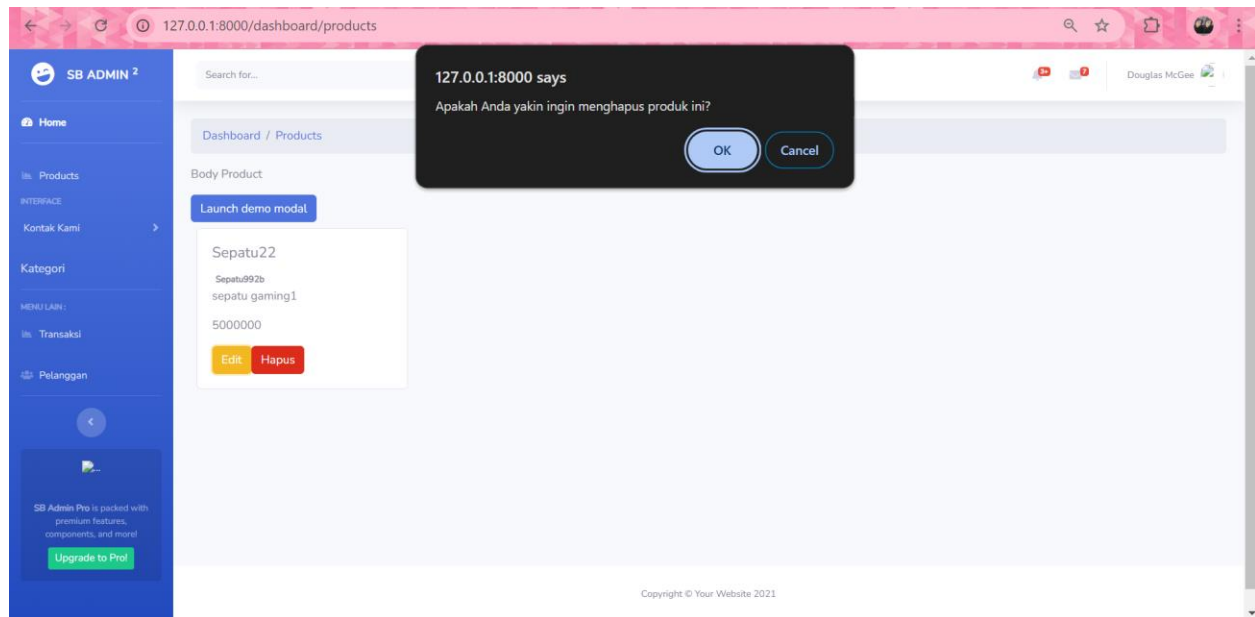
Harga:

5000000

Close

Save changes

Hapus Produk



Jelaskan istilah ORM dan fungsi nya secara singkat :

ORM (Object-Relational Mapping) adalah teknik yang menghubungkan objek dalam pemrograman berorientasi objek dengan database relasional. Fungsi utamanya adalah:

1. **Abstraksi Database:** Menghindari penggunaan query SQL langsung, memungkinkan interaksi melalui objek.
2. **Pengurangan Kode:** Mengurangi jumlah kode yang perlu ditulis untuk operasi database.
3. **Portabilitas:** Memudahkan pemindahan aplikasi antara berbagai jenis database.
4. **Pengelolaan Hubungan:** Menangani relasi antar tabel dengan lebih intuitif.
5. **Keamanan:** Mengurangi risiko serangan SQL injection.

Contoh ORM dalam Laravel adalah Eloquent, yang memungkinkan penggunaan sintaks berbasis objek untuk berinteraksi dengan database.

Jelaskan istilah Queri Builder dan fungsinya secara singkat

Query Builder adalah alat dalam framework atau bahasa pemrograman yang memungkinkan pengembang untuk membangun dan menjalankan kueri database menggunakan sintaks yang lebih mudah dan intuitif, biasanya berbasis objek atau fluent interface.

Fungsinya:

1. **Abstraksi SQL:** Memudahkan pembuatan kueri tanpa menulis SQL secara langsung.
2. **Keamanan:** Membantu mencegah SQL injection dengan mengamankan input pengguna.
3. **Fleksibilitas:** Memungkinkan pembentukan kueri yang kompleks dengan cara yang lebih terstruktur.
4. **Portabilitas:** Memungkinkan migrasi antar database dengan sedikit perubahan pada kode.

Contoh Query Builder terdapat pada Laravel, yang menyediakan berbagai metode untuk membangun kueri dengan cara yang bersih dan efisien.

Jelaskan istilah Collection dan fungsinya secara singkat

Collection adalah struktur data yang digunakan untuk menyimpan kumpulan elemen, biasanya objek atau array, dalam sebuah koleksi yang terorganisir. Dalam konteks framework seperti Laravel, Collection adalah kelas yang menyediakan berbagai metode untuk memanipulasi dan mengelola data yang diperoleh dari database atau array.

Fungsinya:

1. **Manipulasi Data:** Memudahkan pengolahan data, seperti penyaringan, pengurutan, dan pemetaan, melalui berbagai metode chainable.
2. **Fleksibilitas:** Dapat menangani data dalam format yang beragam dan mendukung operasi kompleks secara efisien.
3. **Iterasi:** Mempermudah iterasi melalui elemen dalam koleksi tanpa perlu menulis loop yang rumit.
4. **Integrasi:** Terintegrasi dengan baik dengan Eloquent ORM, memungkinkan pengembang untuk dengan mudah memanipulasi hasil kueri database.

Contoh penggunaan Collection sering ditemukan dalam pengambilan data menggunakan Eloquent, di mana hasil kueri secara otomatis diubah menjadi objek Collection.

Screenshoot isi routes di dalam file web.php

```
<?php

use App\Http\Controllers\Dashboard\ProductController;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\ParsingDataController;
use SebastianBergmann\CodeCoverage\Report\Html\Dashboard;
use App\Http\Controllers\Dashboard\CategoryController;
use App\Http\Controllers\Dashboard\PelangganController;
```

```

use App\Http\Controllers\Dashboard\TransaksiController;
/*
-----
| Web Routes
|-----
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider and all of them will
| be assigned to the "web" middleware group. Make something great!
|
*/

Route::get('/', function () {
    return redirect()->route('dashboard.index');
})->name('dashboard');
Auth::routes();
Route::resource('products',ProductController::class);
Route::resource('products',ProductController::class)->only([
    'index','store','destroy'
])->names('products');
Route::prefix('dashboard')->middleware(['auth'])->name('dashboard.')->group(function(){
    Route::get('/',function(){
        return view('dashboard.index',['title'=>'Dashboard']);
    }->name('index');
    Route::resource('products',ProductController::class)->only([
        'index','store','destroy'
    ]->names('products');
    Route::redirect('test','dashboard/products');
});
Route::get('/home', [App\Http\Controllers\HomeController::class, 'index']->name('home'));
// Route untuk halaman Kontak Kami
// Route group dengan prefix 'admin'
Route::prefix('admin')->group(function () {
    // Route untuk halaman Kontak Kami
    Route::get('/kontakkami', function () {
        return view('dashboard.kontakkami');
    }->name('kontakkami');
});
Route::get('/parse-data/{nama_lengkap}/{email}/{jenis_kelamin}',
[ParsingDataController::class, 'parseData'])
->name('parse.data');
Route::get('/kategori', function () {
    return view('kategori');
})->name('kategori');

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

```

```

    return view('transaksi');
  })->name('transaksi');

Route::resource('pelanggan', PelangganController::class);
Route::resource('categories', CategoryController::class);
Route::resource('transaksi', TransaksiController::class);

```

Kategori

Screenshoot source code simpan kategori

```

@extends('layouts.dashboard')

@section('content')
<p>Body Category</p>
<!-- Button trigger modal -->
<button type="button" class="btn btn-primary" data-toggle="modal" data-
target="#addCategoryModal">
    Tambah Kategori
</button>

<div id="category-list" class="row">
    @foreach ($categories as $category)
    <div class="col-md-3 m-2">
        <div class="card" style="width: 18rem;">
            <div class="card-body">
                <h5 class="card-title">{{ $category['name'] }}</h5>
                <p class="card-text">Slug: {{ $category['slug'] }}</p>
                <div class="d-flex">
                    <button class="btn btn-warning edit-button" data-toggle="modal" data-
target="#editCategoryModal" onclick="loadCategoryData({{ $category->id
}})">Edit</button>
                    <button class="btn btn-danger" onclick="deleteCategory('{{
route('categories.destroy', $category['id']) }}'">Hapus</button>
                </div>
            </div>
        </div>
    </div>
    @endforeach
</div>

<!-- Modal untuk menambah kategori -->
<div class="modal fade" id="addCategoryModal" tabindex="-1" role="dialog" aria-
labelledby="addCategoryModalLabel" aria-hidden="true">

```

```

<div class="modal-dialog modal-dialog-centered" role="document">
  <div class="modal-content">
    <div class="modal-header">
      <h5 class="modal-title" id="addCategoryModalLabel">Tambah Kategori</h5>
      <button type="button" class="close" data-dismiss="modal" aria-label="Close">
        <span aria-hidden="true">&times;</span>
      </button>
    </div>
    <form id="category-form" method="POST" action="{{ route('categories.store') }}">
      @csrf
      <div class="modal-body">
        <div class="form-group">
          <label for="category-name" class="col-form-label">Nama Kategori:</label>
          <input type="text" class="form-control" id="category-name" name="name"
required>
        </div>
        <div class="form-group">
          <label for="category-slug" class="col-form-label">Slug:</label>
          <input type="text" class="form-control" id="category-slug" name="slug"
required>
        </div>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary"
onclick="submitCategoryForm()">Simpan</button>
      </div>
    </form>
  </div>
</div>

<!-- Modal untuk edit kategori -->
<div class="modal fade" id="editCategoryModal" tabindex="-1" role="dialog" aria-
labelledby="editCategoryModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-dialog-centered" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="editCategoryModalLabel">Edit Kategori</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <form id="edit-category-form" method="POST">
        @csrf

```

```

        @method('PUT')
        <div class="modal-body">
            <div class="form-group">
                <label for="edit-category-name" class="col-form-label">Nama
Kategori:</label>
                <input type="text" class="form-control" id="edit-category-name"
name="name" required>
            </div>
            <div class="form-group">
                <label for="edit-category-slug" class="col-form-label">Slug:</label>
                <input type="text" class="form-control" id="edit-category-slug" name="slug"
required>
            </div>
        </div>
        <div class="modal-footer">
            <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
            <button type="button" class="btn btn-primary"
onclick="submitEditCategoryForm()">Simpan</button>
        </div>
    </form>
</div>
</div>
</div>

```

```
@endsection
```

```

@push('scripts')
<script>
function submitCategoryForm() {
    let form = document.getElementById('category-form');
    let data = new FormData(form);

    $.ajax({
        url: $(form).attr('action'),
        type: 'POST',
        data: data,
        contentType: false,
        processData: false,
        success: function(response) {
            $('#addCategoryModal').modal('hide');
            window.location.reload();
        },
        error: function(response) {
            if (response.status === 422) {
                let errors = response.responseJSON.errors;

```

```

        for (let key in errors) {
            $('[name=${key}]').addClass('is-invalid');
        }
    }
}
});
}

function loadCategoryData(id) {
    $.ajax({
        url: `/categories/${id}/edit`,
        type: 'GET',
        success: function(response) {
            $('#edit-category-name').val(response.name);
            $('#edit-category-slug').val(response.slug); // Menambahkan pengambilan slug
            $('#edit-category-form').attr('action', `/categories/${id}`);
            $('#editCategoryModal').modal('show');
        },
        error: function(error) {
            console.log(error);
        }
    });
}

function submitEditCategoryForm() {
    let form = document.getElementById('edit-category-form');
    let data = new FormData(form);

    $.ajax({
        url: $(form).attr('action'),
        type: 'POST',
        data: data,
        contentType: false,
        processData: false,
        success: function(response) {
            $('#editCategoryModal').modal('hide');
            window.location.reload();
        },
        error: function(response) {
            if (response.status === 422) {
                let errors = response.responseJSON.errors;
                for (let key in errors) {
                    $('[name=${key}]').addClass('is-invalid');
                }
            }
        }
    });
}

```

```

    });
}

function deleteCategory(url) {
  if (confirm('Apakah Anda yakin ingin menghapus kategori ini?')) {
    $.ajax({
      url: url,
      type: 'DELETE',
      data: {
        '_token': $('meta[name="csrf-token"]').attr('content')
      },
      success: function(response) {
        window.location.reload();
      },
      error: function(response) {
        console.log(response);
      }
    });
  }
}
</script>
@endpush

```

Screenshoot untuk hapus produk

```

@extends('layouts.dashboard')

@section('content')
<p>Body Product</p>
<!-- Button trigger modal -->
<button type="button" class="btn btn-primary" data-toggle="modal" data-
target="#exampleModal">
  Launch demo modal
</button>

<div id="product-list" class="row">
  @foreach ($products as $product)
    <div class="col-md-3 m-2">
      <div class="card" style="width: 18rem;">
        <div class="card-body">
          <h5 class="card-title">{{ $product['name'] }}</h5>
          <span class="badge text-bg-primary">{{ $product->category->name }}</span>
          <p class="card-text">{{ $product['description'] }}</p>
          <p class="card-text">{{ $product['price'] }}</p>

```



```

        <div class="d-flex">
            <button class="btn btn-warning" onclick="loadProductData({{ $product->id }})"
data-toggle="modal" data-target="#editProductModal">Edit</button>
            <button class="btn btn-danger" onclick="deleteProduct('{{
route('products.destroy', $product['id']) }}')">Hapus</button>
        </div>
    </div>
</div>
</div>
    @endforeach
</div>

<!-- Modal untuk menambah produk -->
<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-
labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel">Tambah Produk</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <form id="product-form" action="{{ route('products.store') }}" method="POST">
                @csrf
                <div class="modal-body">
                    <div class="form-group">
                        <label for="product-name" class="col-form-label">Nama Produk:</label>
                        <input type="text" class="form-control" id="product-name" name="name">

                        <label for="category" class="col-form-label">Nama Kategori:</label>
                        <select name="category_id" id="category" class="form-control">
                            <option value="selected disabled">Pilih Kategori Produk</option>
                            @foreach ($categories as $category)
                                <option value="{{ $category->id }}">{{ $category->name }}</option>
                            @endforeach
                        </select>

                        <label for="product-description" class="col-form-label">Deskripsi:</label>
                        <textarea type="text" class="form-control" id="product-description"
name="description"></textarea>

                        <label for="product-price" class="col-form-label">Harga:</label>
                        <input type="text" class="form-control" id="product-price" name="price">
                    </div>
                </div>
            </form>
        </div>
    </div>
</div>

```

```

        <div class="modal-footer">
            <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
            <button type="button" class="btn btn-primary" onclick="submitForm()">Save
changes</button>
        </div>
    </form>
</div>
</div>
</div>

<!-- Modal untuk edit produk -->
<div class="modal fade" id="editProductModal" tabindex="-1" role="dialog" aria-
labelledby="editProductModalLabel" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="editProductModalLabel">Edit Produk</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <form id="edit-product-form" method="POST">
                @csrf
                @method('PUT')
                <div class="modal-body">
                    <div class="form-group">
                        <label for="edit-product-name" class="col-form-label">Nama Produk:</label>
                        <input type="text" class="form-control" id="edit-product-name"
name="name">

                        <label for="edit-category" class="col-form-label">Nama Kategori:</label>
                        <select name="category_id" id="edit-category" class="form-control">
                            <option value="selected disabled">Pilih Kategori Produk</option>
                            @foreach ($categories as $category)
                                <option value="{{ $category->id }}">{{ $category->name }}</option>
                            @endforeach
                        </select>

                        <label for="edit-product-description" class="col-form-
label">Deskripsi:</label>
                        <textarea type="text" class="form-control" id="edit-product-description"
name="description"></textarea>

                        <label for="edit-product-price" class="col-form-label">Harga:</label>

```

```

        <input type="text" class="form-control" id="edit-product-price"
name="price">
    </div>
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
    <button type="button" class="btn btn-primary"
onclick="submitEditProductForm()">Save changes</button>
</div>
</form>
</div>
</div>
</div>

```

@endsection

@push('scripts')

```

<script>
    function loadProductData(id) {
        $.ajax({
            url: `/products/${id}/edit`,
            type: 'GET',
            success: function(response) {
                $('#edit-product-name').val(response.name);
                $('#edit-category').val(response.category_id);
                $('#edit-product-description').val(response.description);
                $('#edit-product-price').val(response.price);
                $('#edit-product-form').attr('action', `/products/${id}`);
                $('#editProductModal').modal('show');
            },
            error: function(error) {
                console.log(error);
            }
        });
    }
}

```

```

function submitEditProductForm() {
    let form = document.getElementById('edit-product-form');
    let data = new FormData(form);

    $.ajax({
        url: $(form).attr('action'),
        type: 'POST',
        data: data,
        contentType: false,

```

```

        processData: false,
        success: function(response) {
            $('#editProductModal').modal('hide');
            window.location.reload();
        },
        error: function(response) {
            if (response.status === 422) {
                let errors = response.responseJSON.errors;
                for (let key in errors) {
                    $('[name=${key}]').addClass('is-invalid');
                }
            }
        }
    });
}

```

```

function submitForm() {
    let form = document.getElementById('product-form');
    let data = new FormData(form);

```

```

    $.ajax({
        url: $(form).attr('action'),
        type: 'POST',
        data: data,
        contentType: false,
        processData: false,
        success: function(response) {
            $('#exampleModal').modal('hide');
            window.location.reload();
        },
        error: function(response) {
            if (response.status === 422) {
                let errors = response.responseJSON.errors;
                for (let key in errors) {
                    $('[name=${key}]').addClass('is-invalid');
                }
            }
        }
    });
}

```

```

function deleteProduct(url) {
    if (confirm('Apakah Anda yakin ingin menghapus produk ini?')) {
        $.ajax({
            url: url,
            type: 'DELETE',

```

```

        data: {
          '_token': $('meta[name="csrf-token"]').attr('content')
        },
        success: function(response) {
          window.location.reload();
        },
        error: function(response) {
          console.log(response);
        }
      });
    }
  }
}
</script>
@endpush

```

Screenshoot source untuk hapus pelanggan

```

@extends('layouts.dashboard')

@section('content')
<p>Daftar Pelanggan</p>
<!-- Button trigger modal -->
<button type="button" class="btn btn-primary" data-toggle="modal" data-
target="#addPelangganModal">
  Tambah Pelanggan
</button>

<div id="pelanggan-list" class="row">
  @foreach ($pelanggan as $item)
    <div class="col-md-3 m-2">
      <div class="card" style="width: 18rem;">
        <div class="card-body">
          <h5 class="card-title">{{ $item->nama }}</h5>
          <p class="card-text">Email: {{ $item->email }}</p>
          <p class="card-text">Telepon: {{ $item->telepon }}</p>
          <p class="card-text">Alamat: {{ $item->alamat }}</p>
          <div class="d-flex">
            <button class="btn btn-warning edit-button" data-toggle="modal" data-
target="#editPelangganModal" onclick="loadPelangganData({{ $item->id }})">Edit</button>
            <button class="btn btn-danger" onclick="deletePelanggan('{{
route('pelanggan.destroy', $item->id) }}')">Hapus</button>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

</div>
@endforeach
</div>

<!-- Modal untuk menambah pelanggan -->
<div class="modal fade" id="addPelangganModal" tabindex="-1" role="dialog" aria-
labelledby="addPelangganModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-dialog-centered" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="addPelangganModalLabel">Tambah Pelanggan</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <form id="pelanggan-form" method="POST" action="{{ route('pelanggan.store') }}">
        @csrf
        <div class="modal-body">
          <div class="form-group">
            <label for="pelanggan-nama" class="col-form-label">Nama
Pelanggan:</label>
            <input type="text" class="form-control" id="pelanggan-nama" name="nama"
required>
          </div>
          <div class="form-group">
            <label for="pelanggan-email" class="col-form-label">Email:</label>
            <input type="email" class="form-control" id="pelanggan-email" name="email"
required>
          </div>
          <div class="form-group">
            <label for="pelanggan-telepon" class="col-form-label">Telepon:</label>
            <input type="text" class="form-control" id="pelanggan-telepon"
name="telepon" required>
          </div>
          <div class="form-group">
            <label for="pelanggan-telepon" class="col-form-label">Alamat:</label>
            <input type="text" class="form-control" id="pelanggan-alamat"
name="alamat" required>
          </div>
          <div class="modal-footer">
            <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
            <button type="button" class="btn btn-primary"
onclick="submitPelangganForm()">Simpan</button>
          </div>
        </div>
      </form>
    </div>
  </div>
</div>

```

```

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

<!-- Modal untuk edit pelanggan -->
<div class="modal fade" id="editPelangganModal" tabindex="-1" role="dialog" aria-
labelledby="editPelangganModalLabel" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="editPelangganModalLabel">Edit Pelanggan</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <form id="edit-pelanggan-form" method="POST">
                @csrf
                @method('PUT')
                <div class="modal-body">
                    <div class="form-group">
                        <label for="edit-pelanggan-nama" class="col-form-label">Nama
Pelanggan:</label>
                        <input type="text" class="form-control" id="edit-pelanggan-nama"
name="nama" required>
                    </div>
                    <div class="form-group">
                        <label for="edit-pelanggan-email" class="col-form-label">Email:</label>
                        <input type="email" class="form-control" id="edit-pelanggan-email"
name="email" required>
                    </div>
                    <div class="form-group">
                        <label for="edit-pelanggan-telepon" class="col-form-label">Telepon:</label>
                        <input type="text" class="form-control" id="edit-pelanggan-telepon"
name="telepon" required>
                    </div>
                    <div class="form-group">
                        <label for="edit-pelanggan-telepon" class="col-form-label">Telepon:</label>
                        <input type="text" class="form-control" id="edit-pelanggan-alamat"
name="alamat" required>
                    </div>
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
                </div>
            </form>
        </div>
    </div>
</div>

```

```

        <button type="button" class="btn btn-primary"
onclick="submitEditPelangganForm()">Simpan</button>
    </div>
</form>
</div>
</div>
</div>

```

```
@endsection
```

```
@push('scripts')
```

```
<script>
```

```
function submitPelangganForm() {
    let form = document.getElementById('pelanggan-form');
    let data = new FormData(form);
```

```

    $.ajax({
        url: $(form).attr('action'),
        type: 'POST',
        data: data,
        contentType: false,
        processData: false,
        success: function(response) {
            $('#addPelangganModal').modal('hide');
            window.location.reload();
        },
        error: function(response) {
            if (response.status === 422) {
                let errors = response.responseJSON.errors;
                for (let key in errors) {
                    $('[name=${key}]').addClass('is-invalid');
                }
            }
        }
    });
}

```

```
function loadPelangganData(id) {
```

```

    $.ajax({
        url: `/pelanggan/${id}/edit`,
        type: 'GET',
        success: function(response) {
            // Cek respons
            console.log(response); // Menampilkan respons di konsol

```

```

            // Pastikan untuk menggunakan property yang tepat dari response

```



```

        $('#edit-pelanggan-nama').val(response.data.nama);
        $('#edit-pelanggan-email').val(response.data.email);
        $('#edit-pelanggan-telepon').val(response.data.telepon);
        $('#edit-pelanggan-alamat').val(response.data.alamat); // Pastikan alamat ada di
response
        $('#edit-pelanggan-form').attr('action', `/pelanggan/${id}`);
        $('#editPelangganModal').modal('show');
    },
    error: function(error) {
        console.log('Error:', error);
        alert('Gagal memuat data pelanggan.');// Notifikasi error
    }
});
}

function submitEditPelangganForm() {
    let form = document.getElementById('edit-pelanggan-form');
    let data = new FormData(form);

    $.ajax({
        url: $(form).attr('action'),
        type: 'POST',
        data: data,
        contentType: false,
        processData: false,
        success: function(response) {
            $('#editPelangganModal').modal('hide');
            window.location.reload();
        },
        error: function(response) {
            if (response.status === 422) {
                let errors = response.responseJSON.errors;
                for (let key in errors) {
                    $('[name=${key}]').addClass('is-invalid');
                }
            }
        }
    });
}

function deletePelanggan(url) {
    if (confirm("Apakah Anda yakin ingin menghapus pelanggan ini?")) {
        $.ajax({
            url: url,
            type: 'DELETE',
            data: {

```

```
        '_token': $('meta[name="csrf-token"]').attr('content')
    },
    success: function(response) {
        window.location.reload(); // Refresh halaman setelah berhasil dihapus
    },
    error: function(response) {
        console.log(response); // Log error di konsol
        alert('Gagal menghapus pelanggan.');// Notifikasi error
    }
    });
}
}

</script>
@endpush
```