Laporan

**Tugas Sistem Informasi**

Rekayasa Perangkat Lunak

****

**Disusun Oleh:**

**Nama : Muhamad Dwi Cahyo**

**Jurusan : Rekayasa Perangkat Lunak**

**Kelas : XI RPL**

# 

# E-undangan

E-undangan, sebuah web yang memungkinkan pengguna untuk dengan mudah membuat undangan pernikahan, ulang tahun, atau acara istimewa lainnya dalam bentuk elektronik. Repositori ini berisi kode sumber dari aplikasi E-undangan yang telah saya kembangkan menggunakan framework PHP CodeIgniter 4 sebagai dasar pengembangan.

## Tools yang digunakan

* vscode
* xampp
* php(codeigniter 4)
* composer
* stisla(admin template)
* myth/auth(Login library)

## Install Codeigniter 4

Berikut adalah langkah-langkah instalasi CodeIgniter 4 dengan ringkas dan jelas:

1. Unduh CodeIgniter 4: Kunjungi situs resmi CodeIgniter (https://codeigniter.com) dan unduh versi terbaru dari CodeIgniter 4.
2. Ekstrak Arsip: Setelah unduhan selesai, ekstrak arsip ZIP yang telah diunduh ke direktori lokal di komputer Anda.
3. Konfigurasi: Buka file .env yang ada di direktori root CodeIgniter 4 dan konfigurasikan pengaturan database sesuai dengan pengaturan Anda. Juga, pastikan bahwa CI\_ENVIRONMENT diatur dengan benar untuk lingkungan Anda (misalnya, development, production, atau testing).
4. Buka Terminal/Command Prompt: Buka terminal atau command prompt di direktori root CodeIgniter 4.
5. Jalankan Composer: Jalankan perintah composer install untuk menginstal dependensi yang diperlukan oleh CodeIgniter 4. Pastikan Anda memiliki Composer terinstal di komputer Anda sebelum menjalankan perintah ini.
6. Jalankan Perintah PHP: Jalankan perintah php spark serve di terminal atau command prompt untuk menjalankan server pengembangan lokal CodeIgniter 4. Ini akan memulai server lokal di http://localhost:8080 secara default.
7. Uji Instalasi: Buka browser web dan kunjungi http://localhost:8080 atau URL yang ditampilkan di terminal. Jika Anda melihat halaman selamat datang CodeIgniter, berarti instalasi telah berhasil.

## Integrasi Template Stisla

Berikut adalah langkah-langkah untuk mengintegrasikan template Stisla ke dalam proyek CodeIgniter 4:

1. Unduh Template Stisla: Kunjungi situs resmi Stisla (https://getstisla.com) dan unduh versi terbaru dari template Stisla.
2. Ekstrak Arsip: Setelah unduhan selesai, ekstrak arsip ZIP yang telah diunduh ke dalam direktori proyek CodeIgniter 4.
3. Pindahkan File: Pindahkan semua file dan folder dari template Stisla yang diperlukan ke direktori public di dalam direktori proyek CodeIgniter 4.
4. Buat Layout Template: Di dalam direktori app/Views, buat file baru dengan nama layout.php. File ini akan menjadi layout utama yang akan digunakan untuk semua halaman.
5. Perbarui Layout Template: Buka file layout.php dan salin konten template dari file index.html yang ada di template Stisla. Pastikan untuk mengatur path yang benar untuk asset-asset seperti CSS, JavaScript, dan gambar.
6. Buat Halaman: Di dalam direktori app/Views, buat file-file halaman yang ingin Anda buat dengan menggunakan template Stisla. Contohnya, buat file home.php untuk halaman beranda.
7. Perbarui Halaman: Buka file halaman yang telah Anda buat, misalnya home.php, dan salin konten dari template halaman yang sesuai di dalam folder template Stisla. Sesuaikan konten halaman sesuai dengan kebutuhan proyek Anda.
8. Menggunakan Layout Template: Di setiap file halaman yang telah Anda buat, tambahkan baris berikut pada bagian atas file untuk menggunakan layout template:

<?= $this->extend('layout') ?>

1. Tampilkan Konten Halaman: Di dalam layout template layout.php, tambahkan kode berikut pada bagian yang sesuai untuk menampilkan konten halaman yang telah Anda buat:

<?= $this->renderSection('content') ?>

1. Uji Integrasi: Buka browser web dan kunjungi halaman yang telah Anda buat, misalnya http://localhost:8080/home. Jika tampilan halaman sesuai dengan template Stisla dan integrasi berhasil, berarti integrasi template Stisla dengan CodeIgniter 4 telah selesai.

## install myth/auth

Berikut adalah langkah-langkah instalasi singkat dan jelas untuk Myth/Auth:

1. Pastikan lingkungan pengembangan PHP Anda sudah siap, termasuk web server dan MySQL/MariaDB.
2. Pastikan Composer terinstal di sistem Anda.
3. Buka terminal atau command prompt, lalu navigasikan ke direktori proyek Anda.
4. Jalankan perintah composer create-project myth/auth untuk membuat proyek baru menggunakan Composer.
5. Buka file .env dan atur parameter database sesuai dengan pengaturan Anda.
6. Jalankan perintah php spark migrate -all untuk menjalankan migrasi database.
7. Jalankan perintah php spark auth:create-user untuk membuat akun admin.
8. Jalankan server pengembangan PHP dengan perintah php spark serve.
9. Buka browser dan akses http://localhost:8080 (sesuaikan dengan port yang digunakan).
10. Masukkan kredensial admin yang telah Anda buat sebelumnya pada halaman login.

## beberapa tampilan E-Undangan

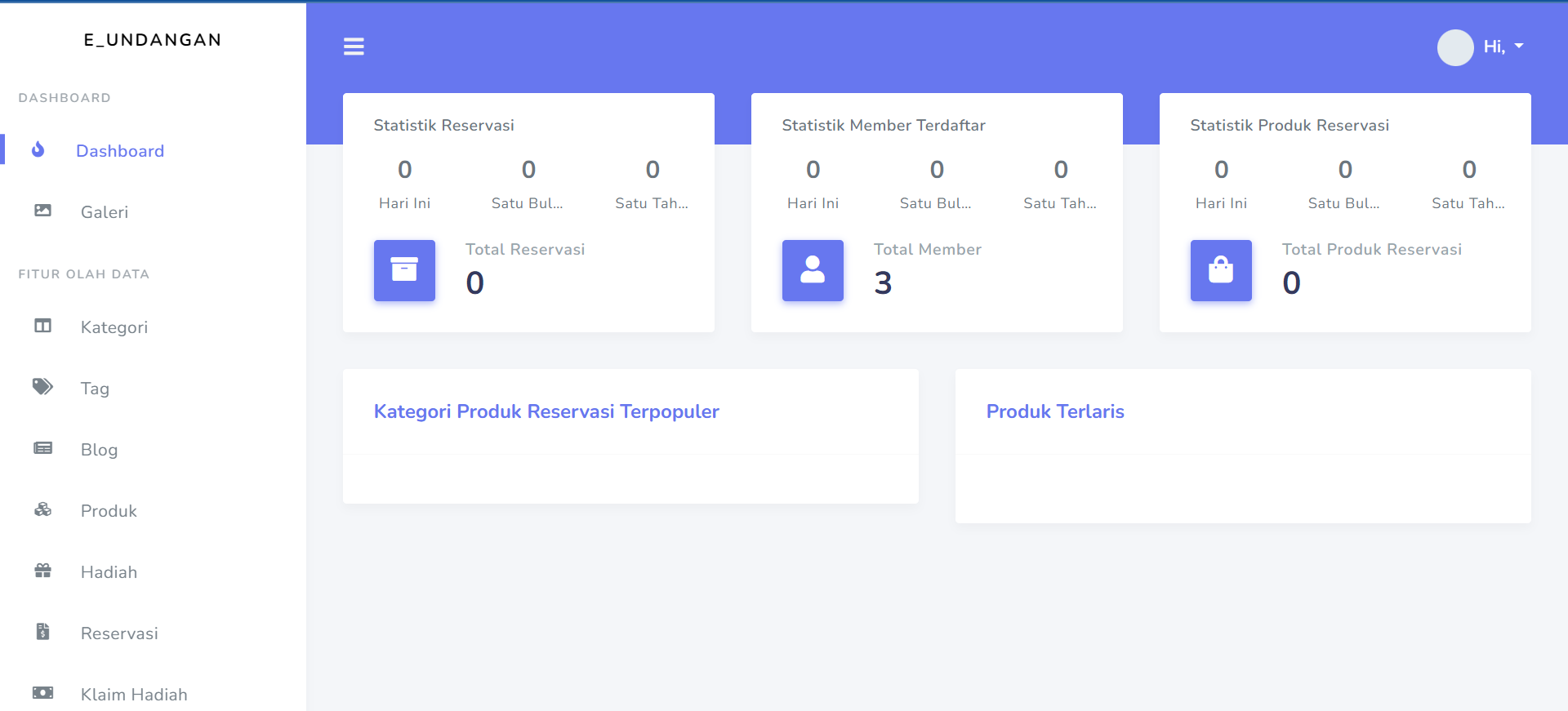
landing page



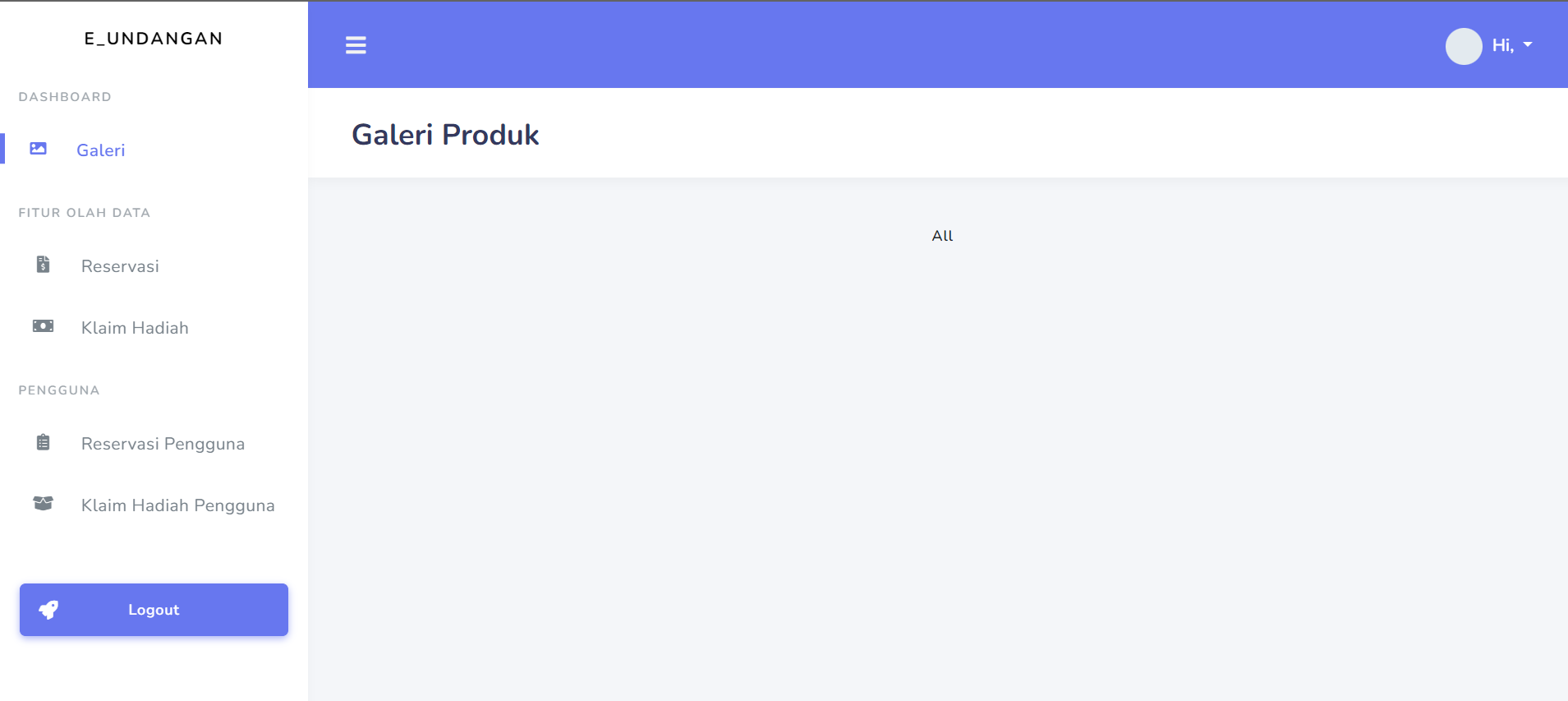
login



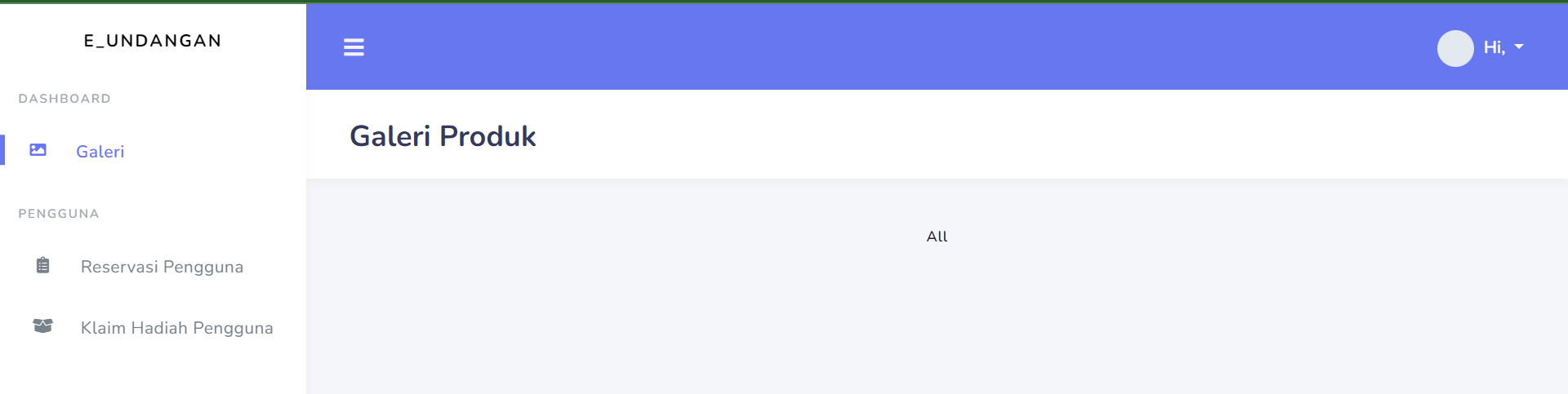
tampilan superadmin



tampilan admin



tampilan user



## Penjelasan Crud simple di E-Undangan

### **buat routes**



### **buat controller kategori**

<?php

namespace App\Controllers;

use App\Models\KategoriModel;

use CodeIgniter\RESTful\ResourcePresenter;

class Kategori extends ResourcePresenter

{

protected $kategori;

public function \_\_construct()

{

$this->kategori = new KategoriModel();

}

/\*\*

\* Present a view of resource objects

\*

\* @return mixed

\*/

public function index()

{

$data['title'] = 'Kategori Produk';

$data['kategori'] = $this->kategori->findAll();

return view('kategori/index', $data);

}

/\*\*

\* Present a view to present a specific resource object

\*

\* @param mixed $id

\*

\* @return mixed

\*/

public function show($id = null)

{

//

}

/\*\*

\* Present a view to present a new single resource object

\*

\* @return mixed

\*/

public function new()

{

$data['title'] = 'Tambah Kategori Produk';

return view('kategori/new', $data);

}

/\*\*

\* Process the creation/insertion of a new resource object.

\* This should be a POST.

\*

\* @return mixed

\*/

public function create()

{

// lakukan validasi

$rules = $this->kategori->validationRules;

// cek ada validasi tidak

if (!$this->validate($rules)) {

return redirect()->back()->withInput()->with('errors', $this->validator->getErrors());

}

$allowedPostFields = [

'nama\_kategori' => $this->request->getPost('nama\_kategori'),

'deskripsi\_kategori' => $this->request->getPost('deskripsi\_kategori')

];

if (!$this->kategori->save($allowedPostFields)) {

return redirect()->back()->withInput()->with('errors', $this->kategori->errors());

}

// Success!

session()->setFlashdata('message', 'Berhasil menambahkan kategori produk');

return redirect()->to(base\_url('superadmin/kategori'));

}

/\*\*

\* Present a view to edit the properties of a specific resource object

\*

\* @param mixed $id

\*

\* @return mixed

\*/

public function edit($id\_kategori = null)

{

$data['title'] = 'Edit Kategori Produk';

$data['kategori'] = $this->kategori->where('id\_kategori', $id\_kategori)->first();

return view('kategori/edit', $data);

}

/\*\*

\* Process the updating, full or partial, of a specific resource object.

\* This should be a POST.

\*

\* @param mixed $id

\*

\* @return mixed

\*/

public function update($id\_kategori = null)

{

// lakukan validasi

$rules = $this->kategori->validationRules;

// cek ada validasi tidak

if (!$this->validate($rules)) {

return redirect()->back()->withInput()->with('errors', $this->validator->getErrors());

}

$allowedPostFields = [

'nama\_kategori' => $this->request->getPost('nama\_kategori'),

'deskripsi\_kategori' => $this->request->getPost('deskripsi\_kategori')

];

if (!$this->kategori->update($id\_kategori, $allowedPostFields)) {

return redirect()->back()->withInput()->with('errors', $this->kategori->errors());

}

session()->setFlashdata('message', 'Berhasil mengubah kategori produk');

return redirect()->to(base\_url('superadmin/kategori'));

}

/\*\*

\* Present a view to confirm the deletion of a specific resource object

\*

\* @param mixed $id

\*

\* @return mixed

\*/

public function remove($id\_kategori = null)

{

if ($id\_kategori != null && $id\_kategori > 0) {

return $this->delete($id\_kategori);

}

}

/\*\*

\* Process the deletion of a specific resource object

\*

\* @param mixed $id

\*

\* @return mixed

\*/

public function delete($id\_kategori = null)

{

$this->kategori->where('id\_kategori', $id\_kategori)->delete();

session()->setFlashdata('message', 'Berhasil menghapus kategori produk');

return redirect()->to(base\_url('superadmin/kategori'));

}

}

### **buat model kategori**

<?php

namespace App\Models;

use CodeIgniter\Model;

class KategoriModel extends Model

{

protected $DBGroup = 'default';

protected $table = 'kategori';

protected $primaryKey = 'id\_kategori';

protected $useAutoIncrement = true;

protected $insertID = 0;

protected $returnType = 'array';

protected $useSoftDeletes = false;

protected $protectFields = true;

protected $allowedFields = ['id\_kategori', 'nama\_kategori', 'deskripsi\_kategori', 'created\_at', 'updated\_at'];

// Dates

protected $useTimestamps = true;

protected $dateFormat = 'datetime';

protected $createdField = 'created\_at';

protected $updatedField = 'updated\_at';

protected $deletedField = false;

// Validation

protected $validationRules = [

'nama\_kategori' => [

'label' => 'Nama Kategori',

'rules' => 'required'

],

'deskripsi\_kategori' => [

'label' => 'Deskripsi Kategori',

'rules' => 'required'

]

];

protected $validationMessages = [];

protected $skipValidation = false;

protected $cleanValidationRules = true;

// Callbacks

protected $allowCallbacks = true;

protected $beforeInsert = [];

protected $afterInsert = [];

protected $beforeUpdate = [];

protected $afterUpdate = [];

protected $beforeFind = [];

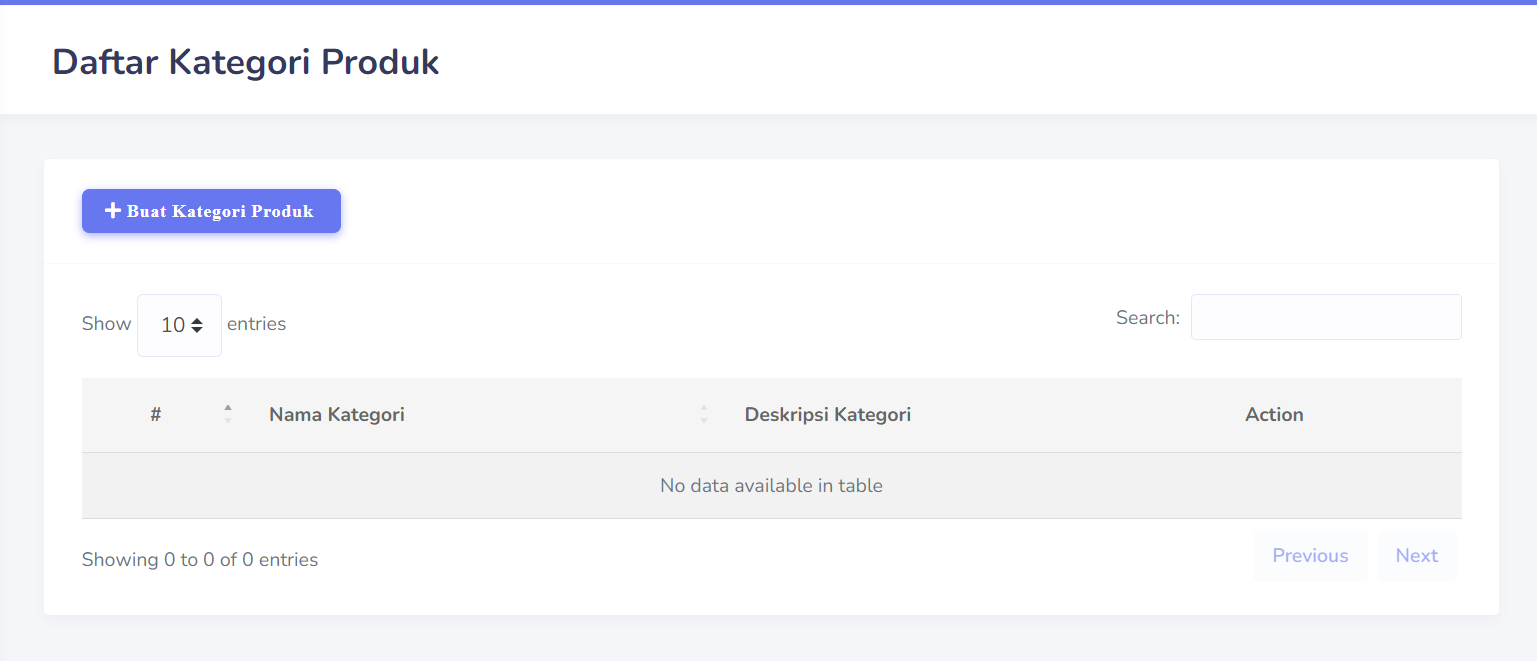
protected $afterFind = [];

protected $beforeDelete = [];

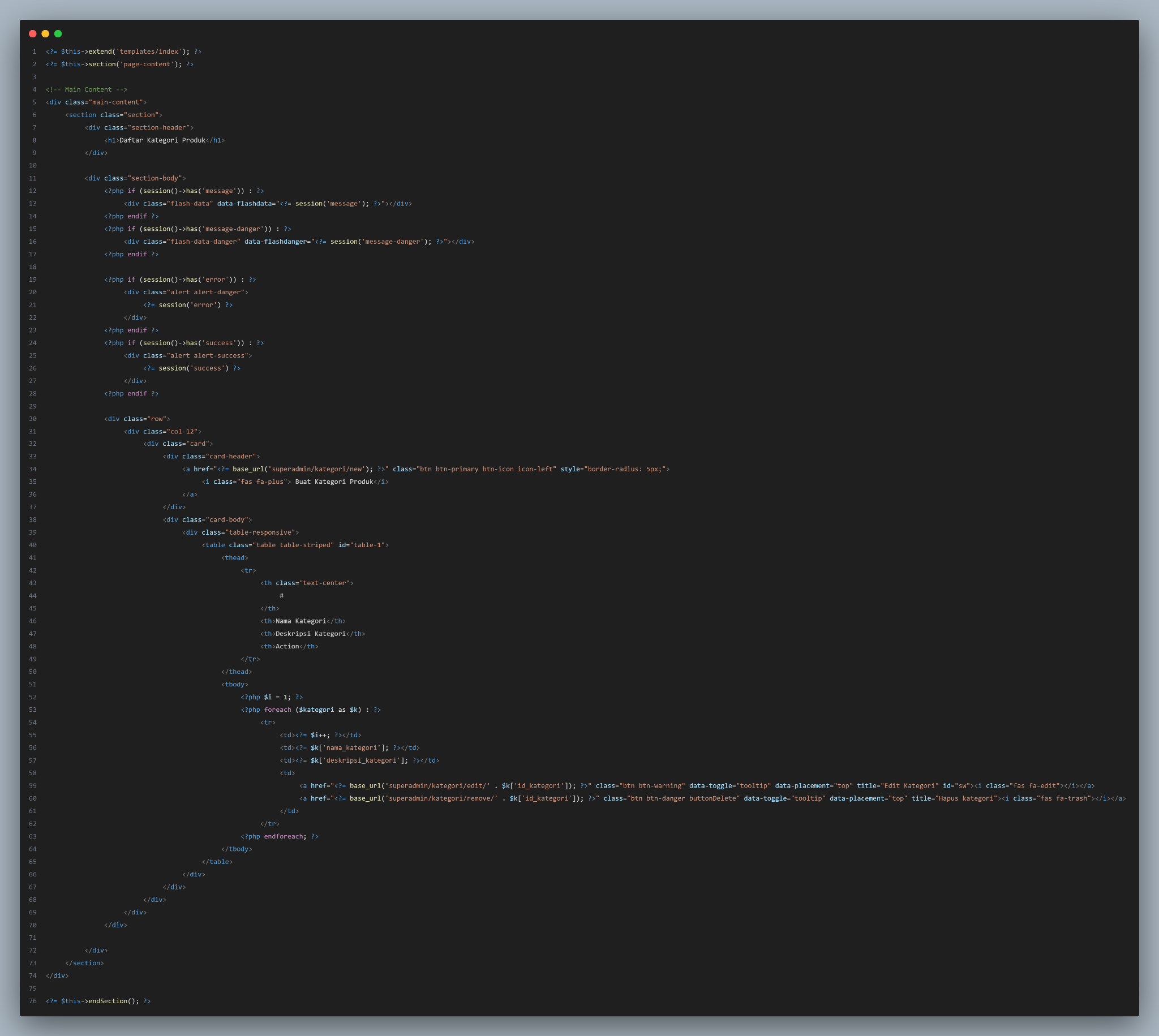
protected $afterDelete = [];

}

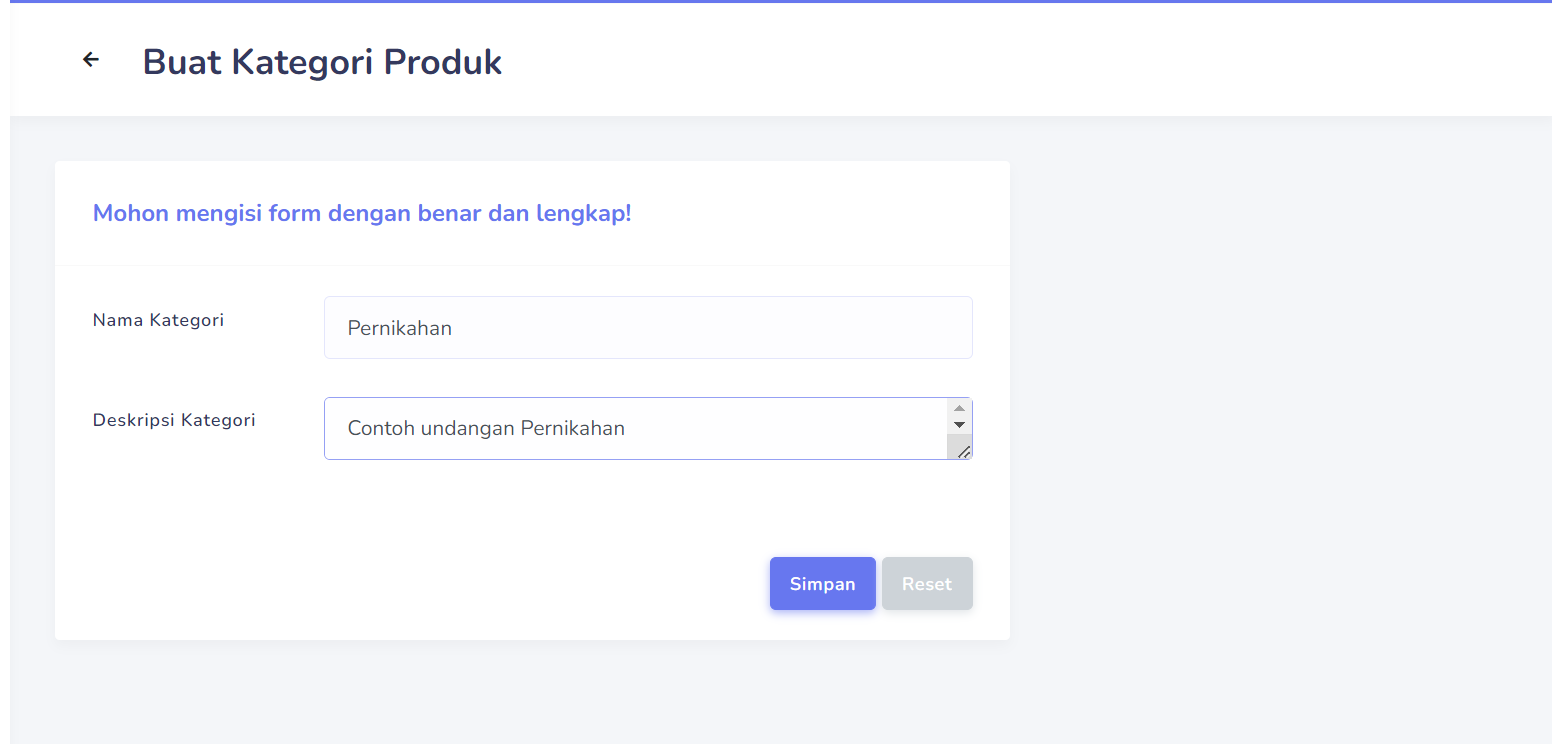
### **buat views kategori**



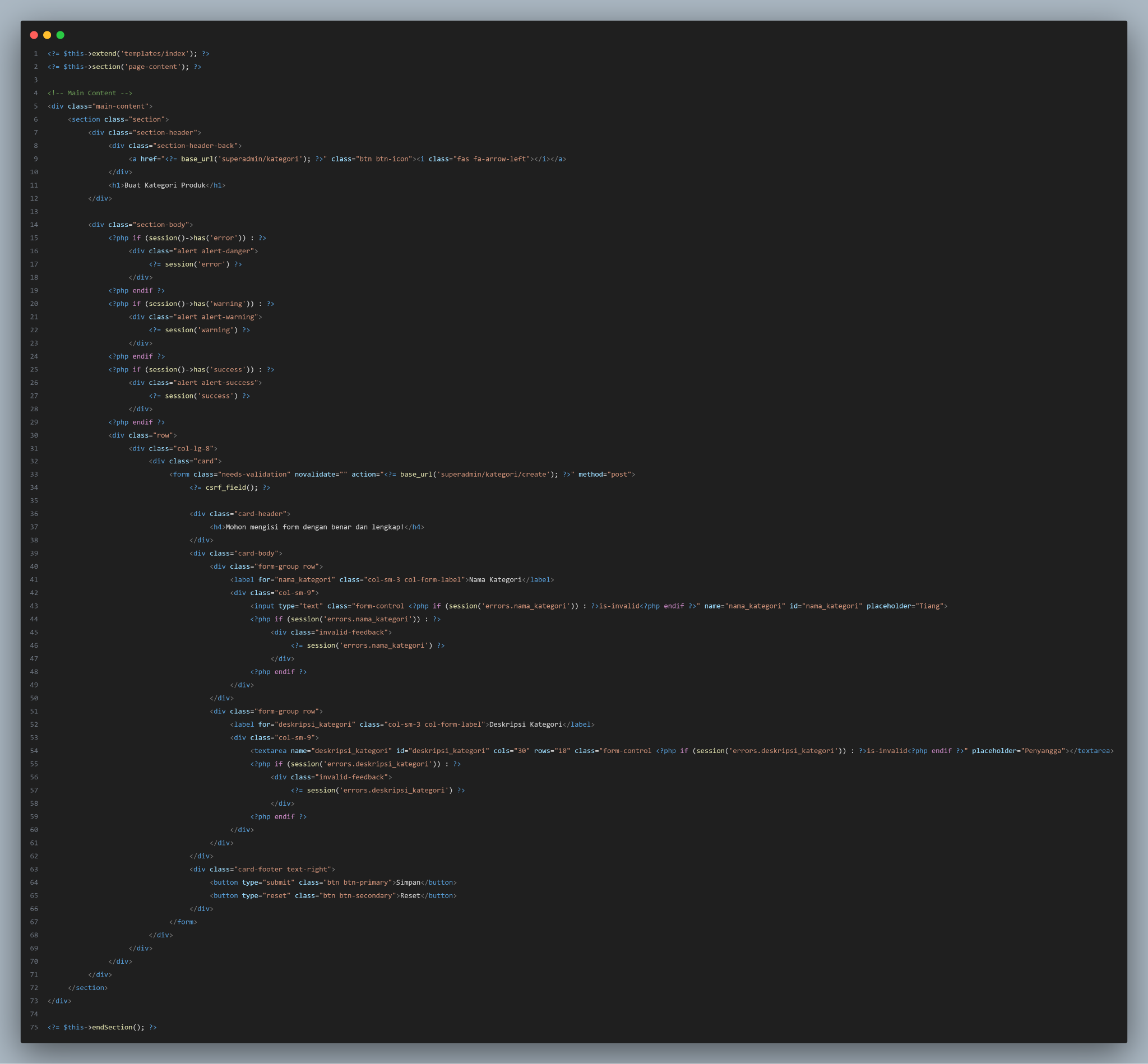
kode tampilan seperti dibawah



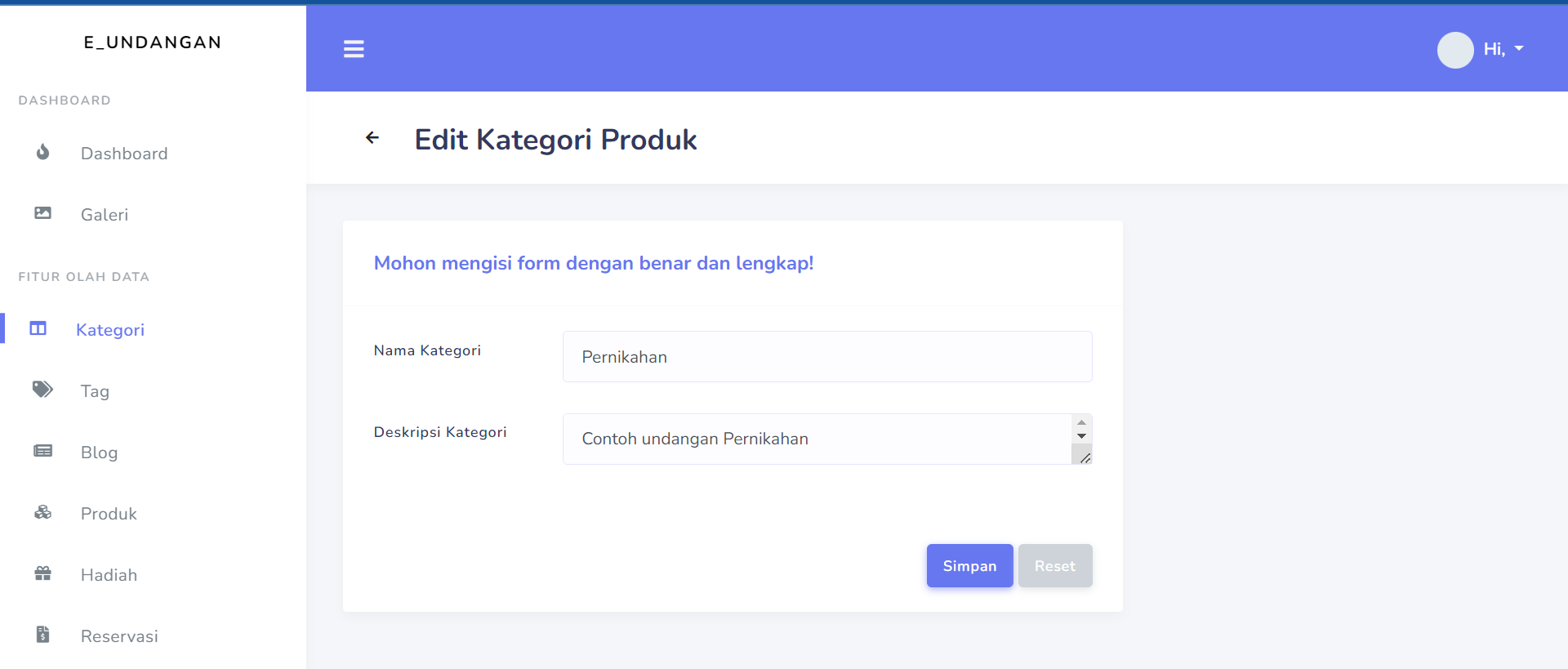
### **buat create views kategori**



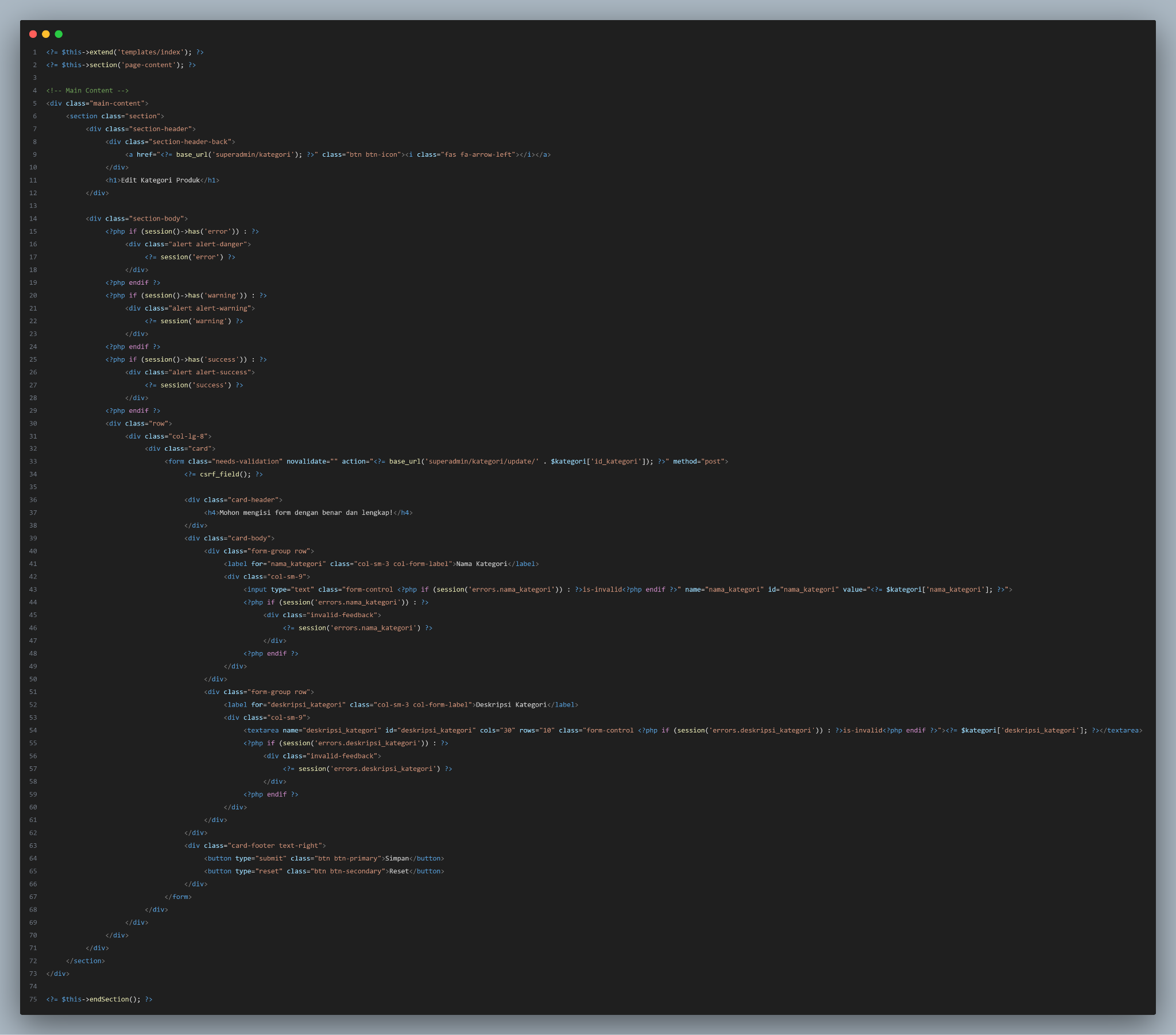
kode tampilan seperti dibawah ini



### **buat edit views kategori**



kode tampilan seperti dibawah ini



# source code lainnya ada di github <https://github.com/dwicahyo1512/elektronik-undangan>