**Pertemuan 9 : FRAMEWORK LARAVEL I**

**Tugas 9 : HariamFakta**

|  |  |
| --- | --- |
| **Nama / Nim** | **:Dicki Ibnu Al Karim / 2318118** |
| **Kelas** | **:E** |
| **Pengajar** | **:** **Alrizky Putra Dhandi / 2318116** |

* 1. **Tujuan**

1. Mampu memahami konsep MVC pada framework Laravel.
2. Mampu mengkofiguransi framework Laravel.
   1. **Alat dan Bahan**
3. Laptop/pc
4. Modul Praktikum Web 2024
5. *Visual Studio Code*
   1. **Langkah-langkah**
6. env**.**

|  |
| --- |
| DB\_CONNECTION=mysql  DB\_HOST=127.0.0.1  DB\_PORT=3306  DB\_DATABASE=db\_harianfakta  DB\_USERNAME=root  DB\_PASSWORD= |

1. create\_products\_table.php

|  |
| --- |
| public function up(): void      {          Schema::create('categories', function (Blueprint $table) {              $table->id('id\_categories');              $table->string('nama',50);              $table->text('deskripsi');              $table->string('gambar', 100);              $table->timestamps();          });      } |

1. web.php

|  |
| --- |
| Route::get('/', function () {      return view('welcome'); // Halaman utama website Laravel Anda  });  Route::get('/BERITA', function () {      return response()->file(public\_path('categories-entry.php'));  }); |

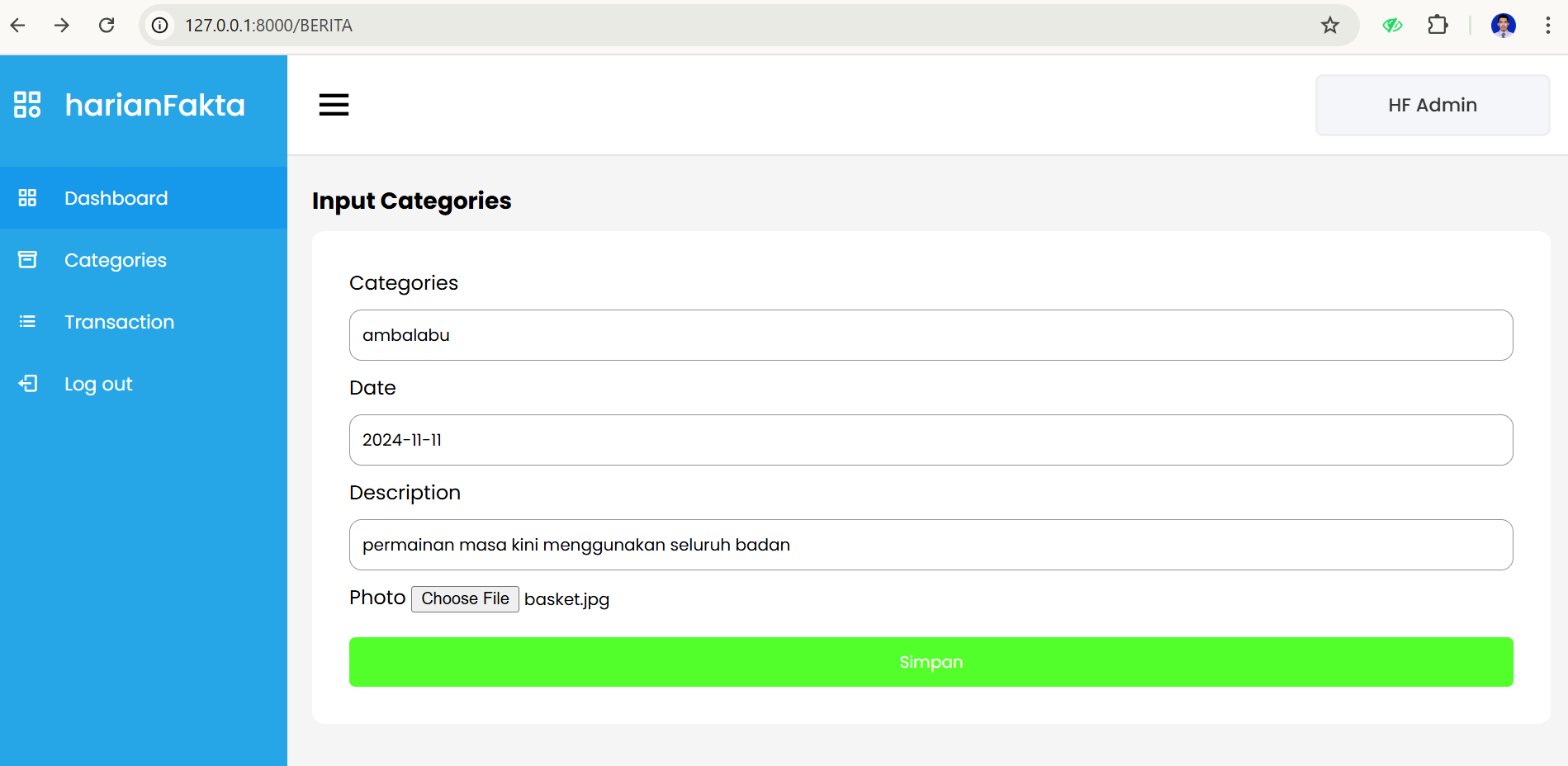
1. categoryControl.php

|  |
| --- |
| <?php  namespace App\Http\Controllers;  use App\Models\Category;  use Illuminate\Http\Request;  use Illuminate\Support\Facades\File;  class CategoryController extends Controller  {      // Menampilkan daftar category      public function index()      {          $categories = Category::all();          return view('categories.categories', compact('categories'));      }      // Menampilkan form untuk menambah category      public function create()      {          return view('categories.categories-entry');      }      // Mengirim data category ke database      public function store(Request $request)      {          // Validasi input data dari user          $request->validate([              'nama' => 'required',              'tanggal' => 'required|date',              'deskripsi' => 'required',              'gambar' => 'required|file|mimes:png,jpg,jpeg|max:2048',          ]);          try {              // Mengupload gambar ke folder tujuan              $gambar = $request->file('gambar');              $nama\_gambar = time() . '\_' . $gambar->getClientOriginalName();              $tujuan\_upload = 'uploads/';              $gambar->move($tujuan\_upload, $nama\_gambar);              // Membuat data Category di database              Category::create([                  'nama' => $request->nama,                  'tanggal' => $request->tanggal,                  'deskripsi' => $request->deskripsi,                  'gambar' => $nama\_gambar,              ]);              return redirect('/category')->with('success', 'Category berhasil ditambahkan');          } catch (\Exception $e) {              return back()->withErrors(['gambar' => 'Gagal mengunggah gambar: ' . $e->getMessage()]);          }      }      // Mengedit data category      public function edit($id\_categories)      {          $category = Category::find($id\_categories);          if (!$category) {              return redirect('/category')->withErrors(['error' => 'Category tidak ditemukan']);          }          return view('categories.categories-edit', compact('category'));      }      // Memperbarui data category      public function update(Request $request, $id\_categories)      {          $request->validate([              'nama' => 'required',              'tanggal' => 'required|date',              'deskripsi' => 'required',              'gambar' => 'file|mimes:png,jpg,jpeg|max:2048',          ]);          $category = Category::find($id\_categories);          if (!$category) {              return redirect('/category')->withErrors(['error' => 'Category tidak ditemukan']);          }          try {              if ($request->hasFile('gambar')) {                  // Menghapus gambar lama jika ada                  if (file\_exists('uploads/' . $category->gambar)) {                      File::delete('uploads/' . $category->gambar);                  }                  $gambar = $request->file('gambar');                  $nama\_gambar = time() . '\_' . $gambar->getClientOriginalName();                  $gambar->move('uploads/', $nama\_gambar);                  $category->gambar = $nama\_gambar;              }              // Memperbarui data category di database              $category->update([                  'nama' => $request->nama,                  'tanggal' => $request->tanggal,                  'deskripsi' => $request->deskripsi,              ]);              return redirect('/category')->with('success', 'Category berhasil diperbarui');          } catch (\Exception $e) {              return back()->withErrors(['gambar' => 'Gagal memperbarui gambar: ' . $e->getMessage()]);          }      }      // Menghapus category dari database      public function delete($id\_categories)      {          $category = Category::find($id\_categories);          if (!$category) {              return redirect('/category')->withErrors(['error' => 'Category tidak ditemukan']);          }          return view('categories.categories-hapus', compact('category'));      }      // Menghapus data category secara permanen      public function destroy($id\_categories)      {          $category = Category::find($id\_categories);          if (!$category) {              return redirect('/category')->withErrors(['error' => 'Category tidak ditemukan']);          }          try {              if (file\_exists('uploads/' . $category->gambar)) {                  File::delete('uploads/' . $category->gambar);              }              $category->delete();              return redirect('/category')->with('success', 'Category berhasil dihapus');          } catch (\Exception $e) {              return back()->withErrors(['error' => 'Gagal menghapus category: ' . $e->getMessage()]);          }      }  } |

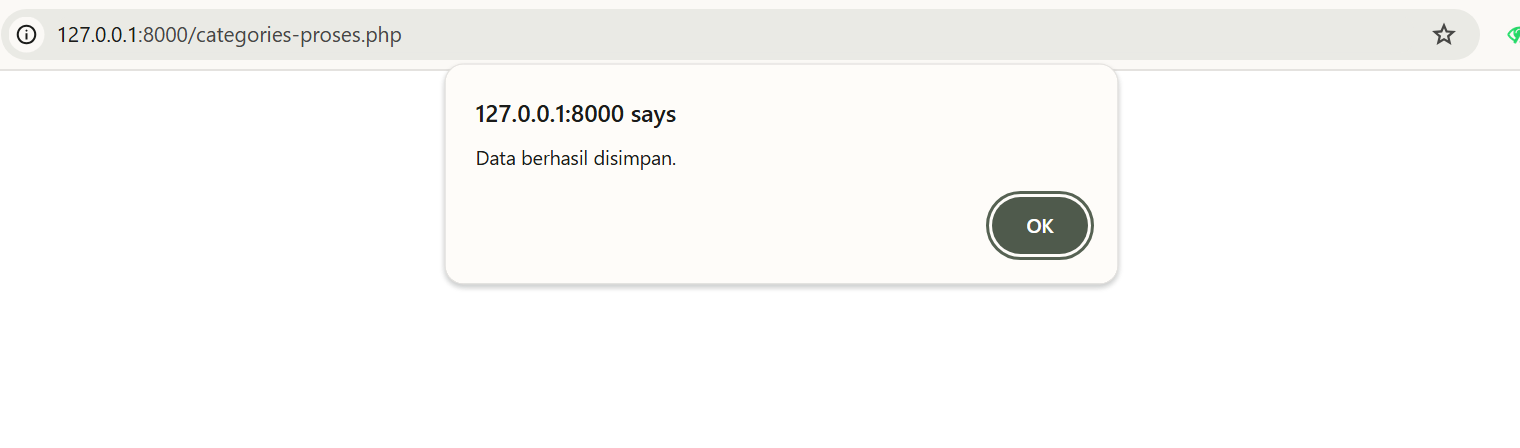
1. Create.blade.php

|  |
| --- |
| @extends('layouts.app')  @section('title')  Tambah Category | HF Admin  @endsection  @section('content')  <h3>Input Categories</h3>  <div class="form-login">  <form action="{{ url('/category/store') }}" method="post" enctype="multipart/form-data">  @csrf  <label for="categories">Categories</label>  <input class="input" type="text" name="nama" id="categories" placeholder="Categories" value="{{ old('nama') }}" />  @error('nama')  <p style="font-size: 10px; color: red">{{ $message }}</p>  @enderror  <label for="price">Price</label>  <input class="input" type="text" name="harga" id="price" placeholder="Price" value="{{ old('harga') }}" />  @error('harga')  <p style="font-size: 10px; color: red">{{ $message }}</p>  @enderror  <label for="description">Description</label>  <textarea class="input" name="deskripsi" id="description" placeholder="Description">{{ old('deskripsi') }}</textarea>  @error('deskripsi')  <p style="font-size: 10px; color: red">{{ $message }}</p>  @enderror  <label for="photo">Photo</label>  <input type="file" name="gambar" id="photo" />  @error('gambar')  <p style="font-size: 10px; color: red">{{ $message }}</p>  @enderror  <button type="submit" class="btn btn-simpan" name="simpan" style="margin-top: 50px">  Simpan  </button>  </form>  </div>  @endsection |

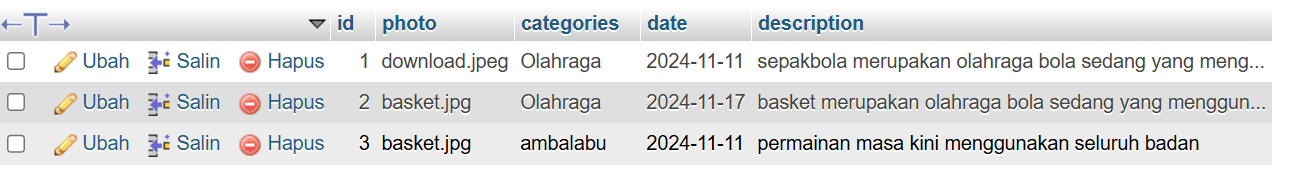
Hasil Tampilan



Gambar 9.1 Tampilan Menambahkan Produk



Gambar 9.2 Tampilan Produk Yang Ditambahkan



Gambar 9.3 Tampilan Database

Analisa :

Dalam implementasi fitur CRUD pada proyek Laravel ini, terdapat lima komponen utama yang telah dibuat dan saling berhubungan. Pertama, konfigurasi .env memastikan koneksi ke database MySQL berjalan dengan benar, menggunakan database db\_komotoys. Kedua, migrasi create\_products\_table.php menciptakan tabel products dengan kolom untuk name, image, dan stock sebagai elemen utama. Ketiga, ProductController menangani logika CRUD, termasuk validasi, penyimpanan file gambar di direktori storage, dan penghapusan file saat data dihapus. Keempat, tampilan Blade seperti index.blade.php dan create.blade.php memberikan antarmuka pengguna untuk menampilkan, menambahkan, dan mengedit data produk, dengan gambar ditampilkan dari direktori storage. Terakhir, proses upload dan penanganan file dilakukan dengan benar, tetapi perlu dipastikan bahwa gambar tersimpan dan URL sesuai dengan direktori yang digunakan untuk menghindari masalah tampilan.