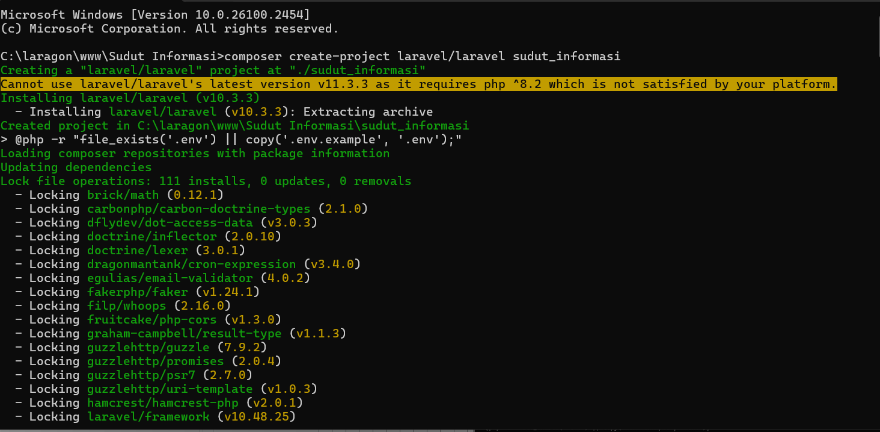
**Pertemuan 9 : Laravel**

**Tugas 9 : Sudut Informasi**

|  |  |
| --- | --- |
| **Nama / Nim** | **: Zayyana Ahya Shifa (2318002)** |
| **Kelas** | **: A** |
| **Pengajar** | **: Alrizky Putra Dhandi (2318116)** |

* 1. **Tujuan**

1. Praktikan dapat memahami dasar – dasar PHP.
2. Praktikan dapat mengimplementasikan PHP.
   1. **Alat dan Bahan**
3. Laptop/pc
4. Modul Praktikum Web 2024
5. TextEditor (Visual Studio Code / Sublime Text)
   1. **DFD Level 1**
6. Buka cmd atau gitbash kemudian masuk ke directori untuk menyimpan project laravel kita, kemudian create-project via composer seperti gambar dibawah ini. Tunggu hingga selesai menginstal laravelnya



Gambar 9.1 Tampilan CMD

1. Buka project Laravel yang sudah dibuat pada text editor. Kemudian silahkan buka file .env lalu cari baris code berikut

|  |
| --- |
| DB\_DATABASE=laravel  DB\_USERNAME=root  DB\_PASSWORD= |

Jika sudah menemukan baris code diatas ubah menjadi seperti berikut

|  |
| --- |
| DB\_DATABASE=crud\_laravel\_sudut\_informasi  DB\_USERNAME=root  DB\_PASSWORD= |

1. Buat model dan migration untuk table categories dengan menggunakan perintah artisan berikut

|  |
| --- |
| php artisan make:model Jurnalis |

Setelah perintah di atas dijalankan maka kita akan mendapatkan 2 buah file baru yaitu file model yang terletak di app/Models/Category.php dan juga file migration yang terletak di database\migrations/2024\_01\_20\_140624\_create\_categories\_table.php.

1. Buka file migration dari table categories kemudian isikan source code berikut pada function up

|  |
| --- |
| public function up()  {  Schema::create('jurnalis', function (Blueprint $table) {  $table->id();  $table->string('nama\_jurnalis');  $table->string('nama\_berita');  $table->date('tanggal');  $table->timestamps();  });  } |

Setelah itu jalankan perintah migrate supaya table categories diambahkan ke database kita

|  |
| --- |
| Php artisan migrate |

1. Buka file model Jurnalis.php kemudian tambahkan source code berikut

|  |
| --- |
| <?php  namespace App\Models;  use Illuminate\Database\Eloquent\Factories\HasFactory;  use Illuminate\Database\Eloquent\Model;  class Jurnalis extends Model  {  protected $fillable = [  'nama\_jurnalis',  'nama\_berita',  'tanggal',  ];    } |

1. Buat controller baru dengan menggunakan perintah artisan berikut

|  |
| --- |
| php artisan make:controller JurnalisController |

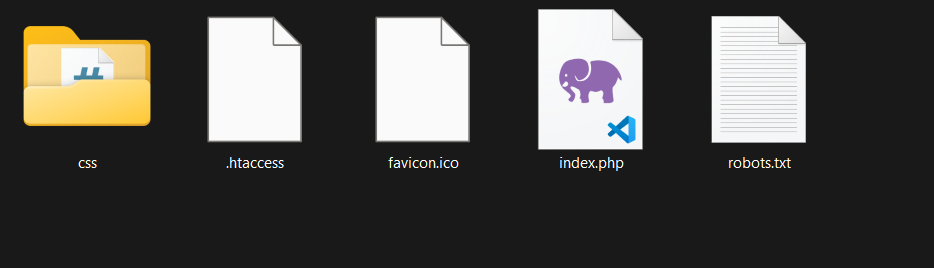
Setelah berhasil menjalankan perintah di atas kita akan mendapatkan file baru dengan nama CategoryController.php di dalam folder app/http/controllers. Buka file tersebut kemudian isikan fungsi fungsi berikut

|  |
| --- |
| <?php  namespace App\Http\Controllers;  use App\Models\Jurnalis;  use Illuminate\Http\Request;  use Barryvdh\DomPDF\Facade\Pdf;  class JurnalisController extends Controller  {  public function index()  {  $jurnalis = Jurnalis::all();  return view('jurnalis.index', compact('jurnalis'));  }  public function store(Request $request)  {  Jurnalis::create($request->all());  return redirect()->route('jurnalis.index')->with('success', 'Data berhasil ditambahkan');  }  public function edit($id)  {  $jurnalis = Jurnalis::findOrFail($id); // Retrieve the journalist data  return view('jurnalis.edit', compact('jurnalis'));  }        public function update(Request $request, Jurnalis $jurnalis)  {  $request->validate([  'nama\_jurnalis' => 'required',  'nama\_berita' => 'required',  'tanggal' => 'required|date',  ]);    $jurnalis->update($request->all());    return view('jurnalis.edit', compact('jurnalis'));  }        public function destroy(Jurnalis $jurnalis)  {  $jurnalis->delete();  return redirect()->route('jurnalis.index')->with('success', 'Data berhasil dihapus!');  }    public function cetakPdf()  {  $jurnalis = Jurnalis::all();  $pdf = Pdf::loadView('jurnalis.pdf', compact('jurnalis'));  return $pdf->stream('daftar\_jurnalis.pdf');  }  } |

1. Buka file web.php pada folder routes lalu tambahkan route baru untuk mengakses halaman halaman category

|  |
| --- |
| <?php  use Illuminate\Support\Facades\Route;  use App\Http\Controllers\JurnalisController;  Route::resource('jurnalis', JurnalisController::class);  Route::get('jurnalis-pdf', [JurnalisController::class, 'cetakPdf'])->name('jurnalis.pdf');  Route::get('jurnalis/{jurnalis}/edit', [JurnalisController::class, 'edit'])->name('jurnalis.edit');  Route::put('/jurnalis/{jurnalis}', [JurnalisController::class, 'update'])->name('jurnalis.update');  Route::delete('jurnalis/{jurnalis}', [JurnalisController::class, 'destroy'])->name('jurnalis.destroy'); |

1. Copy kan folder assets dan css pada project catshop ke dalam folder public di project Laravel kita.



Gambar 9.2 Tampilan Memindahkan Folder *Assets* Dan CSS

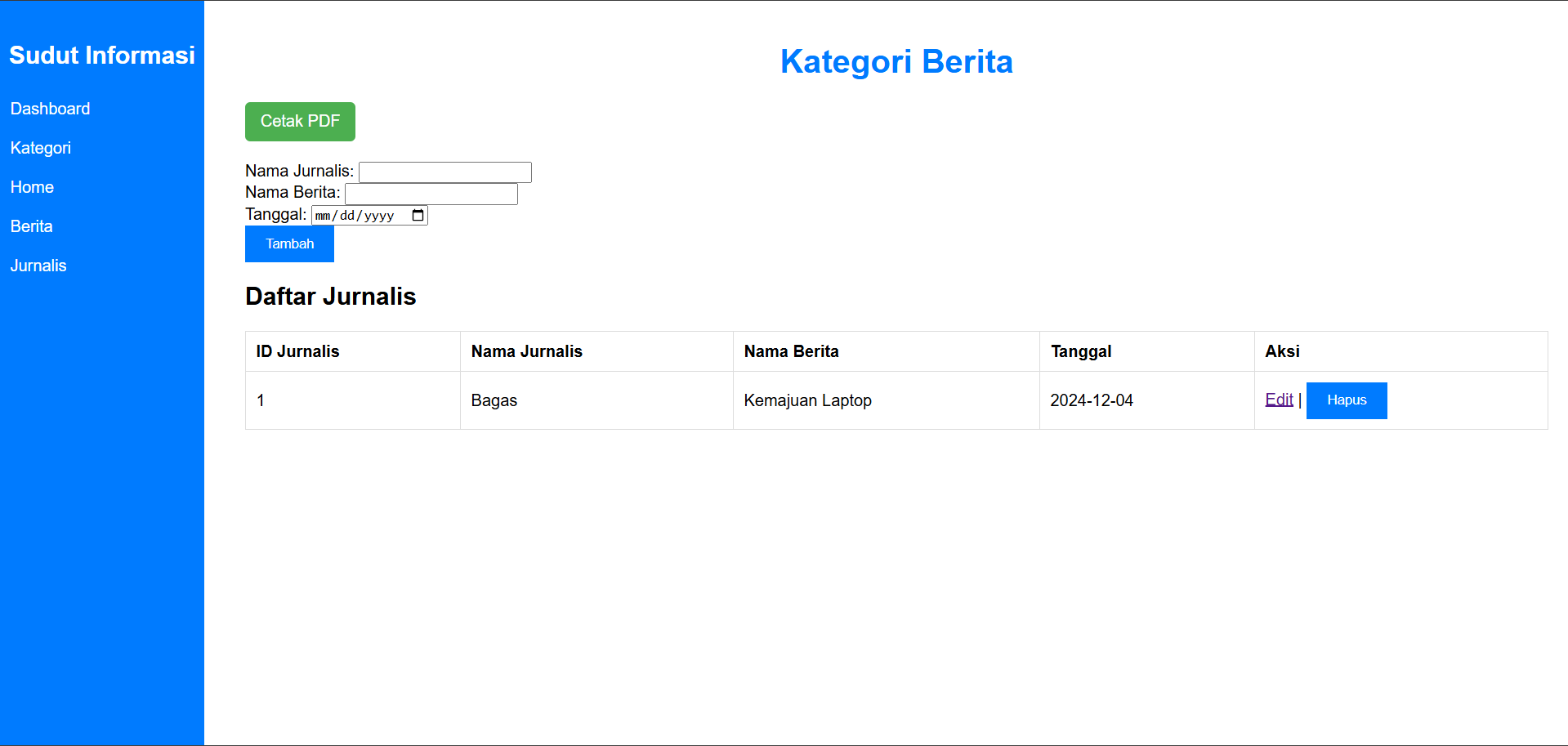
1. Buat file index.blade.php dan tambahkan source code berikut

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Jurnalis</title>  <link rel="stylesheet" href="{{ asset('css/style.css') }}">  </head>  <body>  <div class="sidebar">  <h2>Sudut Informasi</h2>  <ul>  <li><a href="#">Dashboard</a></li>  <li><a href="#">Kategori</a></li>  <li><a href="#">Home</a></li>  <li><a href="#">Berita</a></li>  <li><a href="#">Jurnalis</a></li>  </ul>  </div>  <div class="content">  <h1>Kategori Berita</h1>  <a href="{{ route('jurnalis.pdf') }}" class="btn btn-pdf" target="\_blank">Cetak PDF</a>  <!-- Form -->  <form action="{{ route('jurnalis.store') }}" method="POST">  @csrf  <label>Nama Jurnalis:</label>  <input type="text" name="nama\_jurnalis" required><br>  <label>Nama Berita:</label>  <input type="text" name="nama\_berita" required><br>  <label>Tanggal:</label>  <input type="date" name="tanggal" required><br>  <button type="submit">Tambah</button>  </form>  <!-- Tabel -->  <h2>Daftar Jurnalis</h2>  <table border="1">  <tr>  <th>ID Jurnalis</th>  <th>Nama Jurnalis</th>  <th>Nama Berita</th>  <th>Tanggal</th>  <th>Aksi</th>  </tr>  @foreach ($jurnalis as $item)  <tr>  <td>{{ $item->id }}</td>  <td>{{ $item->nama\_jurnalis }}</td>  <td>{{ $item->nama\_berita }}</td>  <td>{{ $item->tanggal }}</td>  <td>  <!-- Tombol Edit -->  <a href="{{ route('jurnalis.edit', $item->id) }}">Edit</a>  |  <!-- Tombol Hapus -->  <form action="{{ route('jurnalis.destroy', $item->id) }}" method="POST" style="display:inline;">  @csrf  @method('DELETE')  <button type="submit" onclick="return confirm('Yakin ingin menghapus?')">Hapus</button>  </form>  </td>  </tr>  @endforeach  </table>  </div>  </body>  </html> |

1. Buat file edit.blade.php dan tambahkan source code berikut

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Edit Jurnalis</title>  <link rel="stylesheet" href="{{ asset('css/styles.css') }}">  </head>  <body>  <div class="form-container">  <h1>Edit Data Jurnalis</h1>  <form action="{{ route('jurnalis.update', $jurnalis->id ?? 0) }}" method="POST">  @csrf  @method('PUT')  <label>Nama Jurnalis:</label>  <input type="text" name="nama\_jurnalis" value="{{ $jurnalis->nama\_jurnalis }}" required><br>  <label>Nama Berita:</label>  <input type="text" name="nama\_berita" value="{{ $jurnalis->nama\_berita }}" required><br>  <label>Tanggal:</label>  <input type="date" name="tanggal" value="{{ $jurnalis->tanggal }}" required><br>  <button type="submit">Simpan Perubahan</button>  </form>  <a href="{{ route('jurnalis.index') }}" class="back-link">Kembali</a>  </div>  </body>  </html> |

Hasil Tampilan:



Gambar 9.1 Tampilan Menambahkan Kategori

Analisa :

Pada gambar diatas menunjukkan halaman manajemen kategori yang memiliki form input untuk menambahkan data baru, seperti nama jurnalis, nama berita, dan tanggal. Selain itu, terdapat tabel yang berisi data kategori yang sudah ada dengan fitur edit dan hapus di kolom "Action". Antarmuka terlihat cukup rapi dengan navigasi di sisi kiri, serta form dan tabel yang terorganisir.