

Nama: Dea Syahbilah Amanda

NIM: 2602193862

EduFun

1. Routing

Routing di Laravel berfungsi untuk mendefinisikan URL yang dapat diakses oleh pengguna dan mengarahkan permintaan tersebut ke controller dan method yang sesuai.

a. Definisi Route

Routing berfungsi untuk menangani permintaan HTTP dari pengguna.

```
3 use App\Http\Controllers\PostController;
4 use Illuminate\Support\Facades\Route;
5
6 Route::get('/', function () {
7     return view('welcome');
8 });
9 Route::get('/', [PostController::class, 'index']);
10 Route::get('/category/{category}', [PostController::class, 'showCategory']->name('category.show'));
11 Route::get('/detail/{id}', [PostController::class, 'show']->name('detail.show'));
12 Route::get('/writers', [PostController::class, 'showWriters']->name('writers.index'));
13 Route::get('/writer/{author}', [PostController::class, 'showByWriter']->name('writer.show'));
14 Route::get('/about', function () {
15     return view('about');
16 })->name('about');
```

- Route GET /: Menampilkan halaman utama, di dalamnya juga berada snippet dari beberapa artikel pembelajaran.
- Route GET /category/{category}: Menampilkan materi pembelajaran berdasarkan kategori (Interactive Multimedia atau Software Engineering).
- Route GET /detail/{id}: Menampilkan artikel pembelajaran yang telah di-select pengguna, datanya di fetch berdasarkan ID.
- Route GET /writers: Menampilkan daftar penulis yang ada.
- Route GET /writer/{author}: Menampilkan artikel yang ditulis oleh penulis tertentu.
- Route GET /about: Menampilkan halaman "About Us" tentang EduFun.

2. Controller

Controller bertanggung jawab untuk mengatur logika aplikasi dan mengarahkan data ke view.

a. Controller

```

3 namespace App\Http\Controllers;
4
5 use App\Models\Post;
6
7 abstract class Controller
8 {
9     public function index()
10     {
11         $posts = Post::latest()->take(3)->get();
12         return view('home', compact('posts'));
13     }
14 }

```

- index(): Menampilkan 3 artikel untuk ditampilkan di home page.

b. PostController

```

8 class PostController extends Controller
9 {
10     public function showCategory($category)
11     {
12         $posts = Post::where('category', $category)->get();
13         return view('category', compact('posts', 'category'));
14     }
15
16     public function show($id)
17     {
18         $post = Post::findOrFail($id);
19         return view('detail', compact('post'));
20     }
21
22     public function showWriters()
23     {
24         $writers = Post::select('author')->distinct()->get();
25         return view('writers.index', compact('writers'));
26     }
27
28     public function showByWriter($author)
29     {
30         $posts = Post::where('author', $author)->get();
31         return view('writers.show', compact('posts', 'author'));
32     }
33 }

```

- showCategory(\$category): Menampilkan artikel berdasarkan kategori.
- show(\$id): Menampilkan detail artikel berdasarkan ID, function ini digunakann oleh beberapa page, seperti pada category page dan home page ketika user meng-klik read more.
- showWriters(): Menampilkan daftar penulis.
- showByWriter(\$author): Menampilkan setiap artikel yang ditulis oleh penulis.

3. Model

Model di Laravel digunakan untuk berinteraksi dengan database. Setiap model biasanya terkait dengan satu tabel di database dan memungkinkan kita untuk melakukan operasi CRUD dengan mudah

a. Post Model

Model Post berfungsi untuk menangani data artikel.

```

3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Post extends Model
9 {
10     use HasFactory;
11     protected $fillable = [
12         'title', 'category', 'content', 'author', 'post_date',
13     ];
14     protected $dates = ['post_date'];
15 }

```

Model ini mengatur kolom pada table 'posts' yang dapat diisi (fillable) untuk menghindari mass-assignment vulnerability dan mendeklarasikan kolom post_date sebagai tanggal agar bisa diformat dengan mudah.

4. Database Migration



	id	title	category	content	author	post_date	created_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Kenapa memilih Multimedia Interaktif?	Interactive Multimedia	Ini adalah artikel tentang multimedia interaktif....	Multimedia guru	2024-11-07	2024-11-13 25:0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	Code Smells	Software Engineering	Mari pelajari macam-macam code smells dan bagaman...	Smelly Stinks	2024-11-08	2024-11-13 37:4
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	Software engineering basics	Software Engineering	Fondasi untuk kamu-kamu yang tertarik belajar Soft...	Sucipto S.E	2024-11-07	2024-11-13 37:4
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	Course Multimedia Interaktif from zero to hero	Interactive Multimedia	Ini adalah course tentang multimedia interaktif da...	Multmed M.D	2024-11-07	2024-11-13 37:4

```

public function up(): void
{
    Schema::create('posts', function (Blueprint $table) {
        $table->id();
        $table->string('title');
        $table->string('category');
        $table->text('content');
        $table->string('author');
        $table->date('post_date');
        $table->timestamps();
    });
}

```

up: Menentukan struktur tabel, kolom-kolom yang dibutuhkan seperti title, category, content, author, dan post_date, beserta data type yang digunakan.

5. Database dan Seeder

Seeder digunakan untuk mengisi database dengan data dummy.

```

9 class PostSeeder extends Seeder
10 {
11     public function run(): void
12     {
13         Post::create([
14             'title' => 'Kenapa memilih Multimedia Interaktif?',
15             'category' => 'Interactive Multimedia',
16             'content' => 'Ini adalah artikel tentang multimedia interaktif...',
17             'author' => 'Multimedia guru',
18             'post_date' => '2024-11-07',
19         ]);
20
21         Post::create([
22             'title' => 'Code Smells',
23             'category' => 'Software Engineering',
24             'content' => 'Mari pelajari macam-macam code smells dan bagaimana cara menghindarnya di code
25             kamu...',
26             'author' => 'Smelly Stinks',
27             'post_date' => '2024-11-08',
28         ]);
29         Post::create([
30             'title' => 'Software engineering basics',
31             'category' => 'Software Engineering',
32             'content' => 'Fondasi untuk kamu-kamu yang tertarik belajar Software Engineering...',
33             'author' => 'Sucipto S.E',

```

Setelah menulis seeder, untuk memasukan dummy data ke database dapat digunakan command: php artisan db:seed.

6. View

a. Home Page

Fungsi: Halaman utama yang dilihat oleh pengunjung saat mengunjungi website. Terdapat beberapa featured artikel pada home page.

Code:

```

25 <header>
26     <p>BANNER EDUFUN</p>
27     <h1>Welcome to EduFun</h1>
28     <p>Your free education platform!</p>
29 </header>
30
31 <section>
32     <h2>Popular Articles</h2>
33     @foreach ($posts as $post)
34         <article>
35             <h3>{{ $post->title }}</h3>
36             <p>Category: {{ $post->category }}</p>
37             <p>Posted on: {{ $post->post_date }} by {{ $post->author }}</p>
38             <p>{{ Str::limit($post->content, 100) }}</p>
39         </article>
40     @endforeach
41 </section>

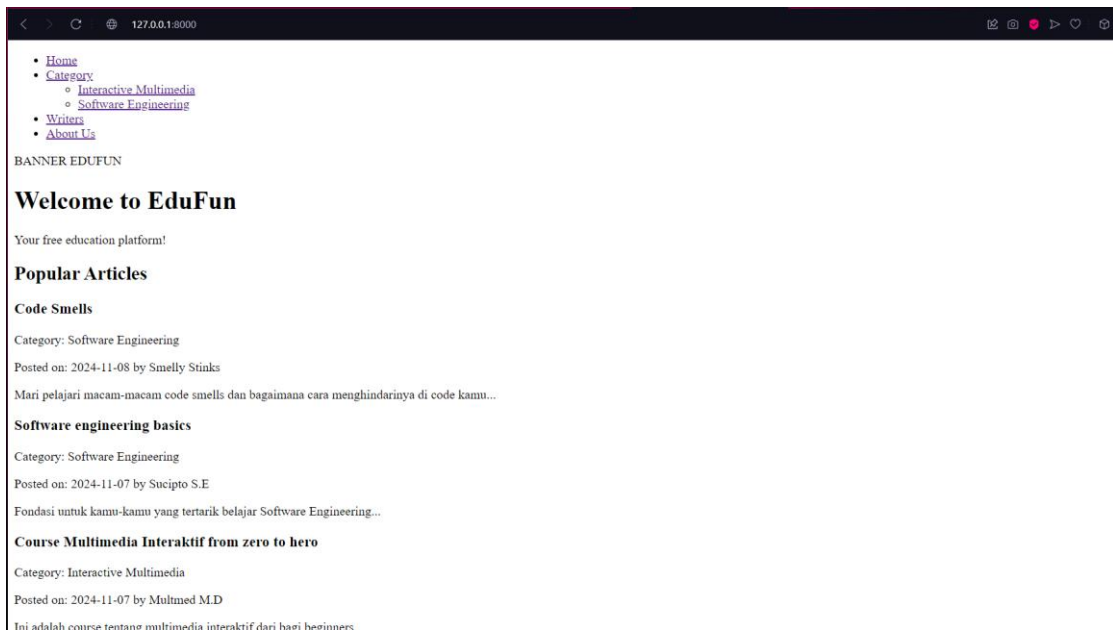
```

- @foreach: Menampilkan setiap artikel yang diambil dari controller (\$posts).
- Str::limit(): Untuk membatasi panjang konten artikel yang ditampilkan di halaman utama.

Data yang diterima:

- \$posts (koleksi artikel yang dikirim dari controller).

Output



b. Category Page

Fungsi: Menampilkan daftar artikel berdasarkan kategori yang dipilih oleh pengguna

Code

```

25     <header>
26         <h1>Category: {{ ucfirst($category) }}</h1>
27     </header>
28
29     <section>
30         @if ($posts->isEmpty())
31             <p>No posts available in this category.</p>
32         @else
33             @foreach ($posts as $post)
34                 <article>
35                     <h2>{{ $post->title }}</h2>
36                     <p>Posted on: {{ $post->post_date }} by {{ $post->author }}</p>
37                     <p>{{ Str::limit($post->content, 100) }}</p>
38                     <a href="{{ route('detail.show', $post->id) }}">Read More</a>
39                 </article>
40             @endforeach
41         @endif
42     </section>

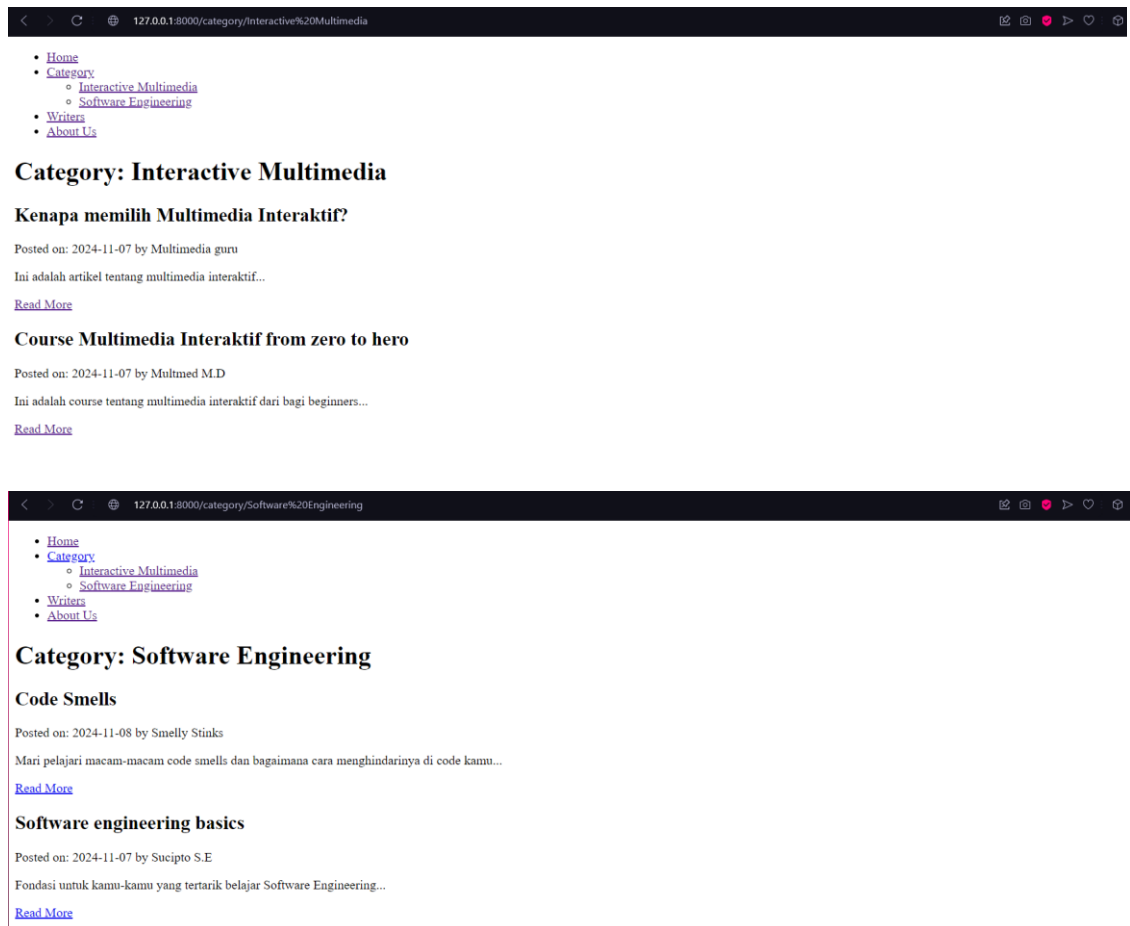
```

- @foreach: Menampilkan setiap daftar artikel sesuai dengan kategori yang dipilih.
- @if: Untuk memeriksa apakah ada artikel dalam kategori tersebut atau tidak, dan menampilkan pesan jika tidak ada artikel.

Data yang diterima:

- \$posts (koleksi artikel dalam kategori tertentu).
- \$category (nama kategori yang dipilih oleh pengguna).

Output



c. Detail Page

Fungsi: Halaman detail artikel yang menampilkan informasi lengkap tentang artikel yang dipilih. Halaman ini menampilkan konten lengkap dari materi pembelajaran

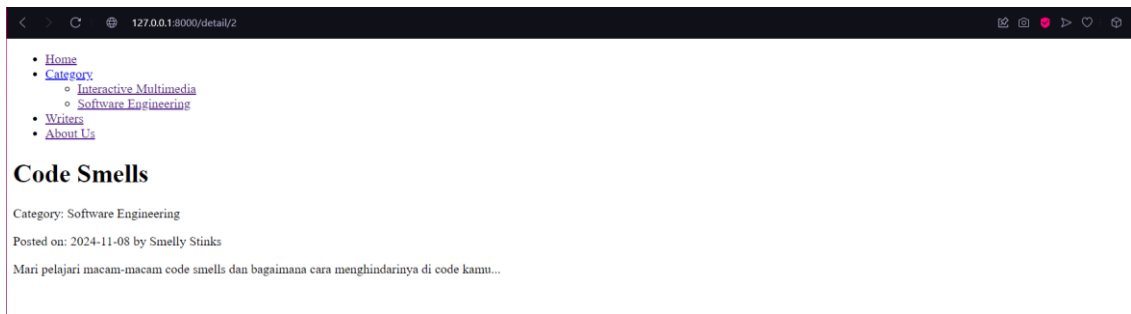
Code

```
25 | <header>
26 |   <h1>{{ $post->title }}</h1>
27 |   <p>Category: {{ $post->category }}</p>
28 |   <p>Posted on: {{ $post->post_date }} by {{ $post->author }}</p>
29 | </header>
30 |
31 | <article>
32 |   <p>{{ $post->content }}</p>
33 | </article>
```

Data yang diterima:

- \$post (artikel yang dipilih).

Output



d. Writers Page

Fungsi: Halaman yang menampilkan daftar penulis. Pengguna dapat mengklik nama penulis untuk melihat artikel-artikel yang telah mereka tulis.

Code

```

25 | <header>
26 |   <h1>Writers</h1>
27 | </header>
28 |
29 | <section>
30 |   <h2>Choose a Writer</h2>
31 |   <ul>
32 |     @foreach ($writers as $writer)
33 |       <li><a href="{{ route('writer.show', $writer->author) }}">{{ $writer->author }}</a></li>
34 |     @endforeach
35 |   </ul>
36 | </section>

```

- @foreach: Menampilkan daftar seluruh penulis yang unik.
- route(): Digunakan untuk membuat URL untuk melihat artikel berdasarkan penulis.

Data yang diterima:

- \$writers (daftar penulis unik yang diambil dari database).

Output



e. Writer's Post Page

Fungsi: Halaman ini menampilkan artikel-artikel yang ditulis oleh penulis tertentu yang dipilih pengguna. Artikel-artikel ini akan diambil berdasarkan nama penulis.

Code

```

25 <header>
26 |   <h1>Posts by {{ $author }}</h1>
27 </header>
28
29 <section>
30 |   @if ($posts->isEmpty())
31 |     <p>No posts available from this writer.</p>
32 |   @else
33 |     @foreach ($posts as $post)
34 |       <article>
35 |         <h2>{{ $post->title }}</h2>
36 |         <p>Category: {{ $post->category }}</p>
37 |         <p>Posted on: {{ $post->post_date }}</p>
38 |         <p>{{ Str::limit($post->content, 100) }}</p>
39 |         <a href="{{ route('detail.show', $post->id) }}">Read More</a>
40 |       </article>
41 |     @endforeach
42 |   @endif
43 </section>

```

- @foreach: Digunakan untuk menampilkan artikel-artikel yang ditulis oleh penulis tersebut.
- route(): Digunakan untuk membuat URL yang me-redirect pengguna ke artikel yang mereka inginkan.

Data yang diterima:

- \$posts (koleksi artikel yang ditulis oleh penulis tertentu).
- \$author (nama penulis yang dipilih).

Output



f. About Us Page

Fungsi: Halaman yang memberikan informasi tentang perusahaan EduFun

Code


```
25 <header>
26 |   <h1>About Us</h1>
27 </header>
28
29 <section>
30   <h2>About EduFun</h2>
31   <p>EduFun adalah perusahaan pendidikan berbasis teknologi asal
    Indonesia. EduFun menyediakan layanan akses pendidikan dalam
    format tulisan berbahasa Indonesia yang disajikan secara online
    melalui website</p>
32   <p>Hingaa Juni 2024, Edufun memiliki lebih dari 10 ribu
    pengguna. EduFun hadir sebagai bentuk revolusi pendididkan
    Indonesia dengan mengedepankan cara berpikir kiritis, logis,
    rasional, dan sumber pengetahuan sains yang terintegrasi
    terhadap semua mahasiswa IT di Indonesia. </p>
33 </section>
```

Output

