



## Route

```
Route::get(uri: '/', action: function (): Factory|View {
    return view(view: 'welcome');
});

Route::get(uri: '/', action: [HomeController::class, 'index']);
Route::get(uri: '/category/{category}', action: [CategoryController::class, 'show'])->name(name: 'category.show');
Route::get(uri: '/writers/{id}', action: [WriterController::class, 'show']);
Route::get(uri: '/about', action: [HomeController::class, 'about']);
Route::get(uri: '/article/{id}', action: [ArticleController::class, 'show']);
```

Code ini untuk menampilkan Halaman-halaman yang terdapat pada website, seperti category, about dan lainnya. Category yang ditampilkan akan sesuai dengan konten yang tertera seperti software engineering. Begitu pula dengan halaman article, yang akan menampilkan id dari artikel yang tertera.

## Controller (Home)

```
class HomeController extends Controller
{
    public function index()
    {
        $articles = Article::latest()->take(5)->get();
        return view('home', compact('articles'));
    }

    public function about()
    {
        return view('about');
    }
}
```

Controller ini digunakan untuk menampilkan 5 article terbaru yang ada di database dan menampilkannya pada halaman home.

## Controller(Writer)

```

use Illuminate\Http\Request;
use App\Models\Writer;
use App\Models\Article;

2 references | 0 implementations
class WriterController extends Controller
{
    1 reference | 0 overrides
    public function index(): Factory|View
    {
        $writers = Writer::all();
        return view(view: 'writers', data: compact(var_name: 'writers'));
    }

    0 references | 0 overrides
    public function show($id): Factory|View
    {
        $writer = Writer::findOrFail(id: $id);

        $articles = Article::where(column: 'writer_id', operator: $id)->get();

        return view(view: 'writer_detail', data: compact(var_name: 'writer', var_names: 'articles'));
    }
}

```

Controller ini untuk menampilkan daftar penulis dari database dan menampilkan detail penulis dan artikel yang ditulis oleh penulis.

### Controller(Article)

```

use App\Models\Article;

2 references | 0 implementations
class ArticleController extends Controller
{
    1 reference | 0 overrides
    public function show($id): Factory|View
    {
        $article = Article::findOrFail(id: $id);

        return view(view: 'article_detail', data: compact(var_name: 'article'));
    }

    0 references | 0 overrides
    public function popular(): Factory|View
    {
        $popularArticles = Article::orderBy(column: 'views', direction: 'desc')->paginate(perPage: 3);

        return view(view: 'article_popular', data: compact(var_name: 'popularArticles'));
    }
}

```

Controller ini untuk menampilkan halaman detail article dengan mengambil article berdasarkan id nya, serta untuk menampilkan halaman article yang populer berdasarkan view dan description nya.

## Controller(Category)

```
2 references | 0 implementations
class CategoryController extends Controller
{
    1 reference | 0 overrides
    public function show($category): Factory|View
    {
        $category = Category::where(column: 'name', operator: $category)->firstOrFail();

        $articles = Article::where(column: 'category_id', operator: $category->id)->get();

        return view(view: 'category.show', data: compact(var_name: 'category', var_names: 'articles'));
    }
}
```

Controller ini untuk mencari kategori berdasarkan namanya, serta mengambil artikel yang sesuai dengan kategori ini. Untuk ditampilkan pada halaman website.

## Controller(About Us)

```
0 references | 0 implementations
class AboutController extends Controller
{
    0 references | 0 overrides
    public function index(): Factory|View
    {
        return view(view: 'about');
    }
}
```

Controller untuk menampilkan halaman about us.

## View(Home)

```

<div class="container-fluid dashboard">
  <div class="row">
    <div class="col-1 logo">
      Logo
    </div>
    <div class="col-6">
    </div>
    <div class="col-1 button1">
      <a class="nav-link" href="/">Home</a>
    </div>
    <div class="col-1 button2">
      <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">Catego
      <div class="dropdown-menu" aria-labelledby="navbarDropdown">
        <a class="dropdown-item" href="{{ route(name: 'category.show', parameters: 'interactive-multimedia') }}">Interactive Multimedia</a>
        <a class="dropdown-item" href="{{ route(name: 'category.show', parameters: 'software-engineering') }}">Software Engineering</a>
      </div>
    </div>
    <div class="col-1 button3">
      <a class="nav-link" href="/writers">Writers</a>
    </div>
    <div class="col-1 button4">
      <a class="nav-link" href="/about">About Us</a>
    </div>
    <div class="col-1 button5">
      <a class="nav-link" href="/popular">Popular</a>
    </div>
  </div>
</div>

```

Code untuk tampilan Home bagian Dashboard dengan satu fitur dropdown pada tombol category

```

<div class="container my-4">
  <div class="row">
    @foreach($articles as $article)
      <div class="container-fluid mb-4">
        <div class="row">
          <div class="col-5 gambar">
            
          </div>
          <div class="col-7 description">
            <div class="row">
              <div class="title">
                <h5 class="card-title">{{ $article->title }}</h5>
              </div>
              <div class="container-fluid">
                <div class="row">
                  <div class="col-3 date">
                    <h5 class="card-title">{{ $article->posting_date }}</h5>
                  </div>
                  <div class="col-5 author">
                    <h5 class="card-title">by :{{ $article->writer_id }}</h5>
                  </div>
                </div>
              </div>
              <div class="description">
                <p class="card-text">{{ Str::limit(value: $article->content, limit: 100) }}</p>
              </div>
              <div class="more">
                <a href="/article/{{ $article->id }}" class="btn btn-primary">Read more...</a>
              </div>
            </div>
          </div>
        </div>
      </div>
    @endforeach
  </div>
</div>

```

Code untuk menampilkan article berdasarkan banyaknya article yang dikirim dari controller. Code ini juga menunjukkan detail-detail article seperti deskripsi, date, author, dan lainnya.

View(About Us)

```

<div class="container-fluid dashboard">
  <div class="row">
    <div class="col-1 logo">
      Logo
    </div>
    <div class="col-6">
    </div>
    <div class="col-1 button1">
      <a class="nav-link" href="/">Home</a>
    </div>
    <div class="col-1 button2">
      <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">Catego
      <div class="dropdown-menu" aria-labelledby="navbarDropdown">
        <a class="dropdown-item" href="{{ route(name: 'category.show', parameters: 'interactive-multimedia') }}">Interactive Multimedia</a>
        <a class="dropdown-item" href="{{ route(name: 'category.show', parameters: 'software-engineering') }}">Software Engineering</a>
      </div>
    </div>
    <div class="col-1 button3">
      <a class="nav-link" href="/writers">Writers</a>
    </div>
    <div class="col-1 button4">
      <a class="nav-link" href="/about">About Us</a>
    </div>
    <div class="col-1 button5">
      <a class="nav-link" href="/popular">Popular</a>
    </div>
  </div>
</div>

```

Code untuk tampilan Home bagian Dashboard dengan satu fitur dropdown pada tombol category

```

<div class="container mt-5 AboutUs">
  <div class="row">
    <div class="col-md-12 text-center AboutUs">
      <h1>About Us</h1>
      <p>EduFun adalah perusahaan pendidikan berbasis teknologi asal Indonesia. EduFun menyediakan layanan akses pendidikan dalam format tulisan berbahasa Ir
      <p>Hingga Juni 2024, EduFun memiliki lebih dari 10 ribu pengguna. EduFun hadir sebagai bentuk revolusi dari pendidikan di Indonesia dengan mengedepankan
    </div>
  </div>
</div>

```

Code untuk menampilkan about Us

View(Category)

```

<div class="container my-4">
  <h2>Kategori: {{ $category->name }}</h2>
  <p>Daftar artikel di bawah kategori {{ $category->name }}:</p>

  <div class="row">
    @forelse($articles as $article)
      <div class="col-md-4">
        <div class="card mb-4">
          
          <div class="card-body">
            <h5 class="card-title">{{ $article->title }}</h5>
            <p class="card-text">{{ Str::limit(value: $article->content, limit: 100) }}</p>
            <a href="{{ route(name: 'article.show', parameters: $article->id) }}" class="btn btn-primary">Baca Selengkapnya...</a>
          </div>
        </div>
      </div>
    @empty
      <p class="text-muted">Tidak ada artikel di kategori ini.</p>
    @endforelse
  </div>
</div>

```

Code untuk menampilkan article berdasarkan category yang dipilih dan menampilkan detail-detail article seperti judul dan gambar

View(writers)

```

<div class="container my-4">
  <h2>{{ $writer->name }}</h2>
  <p>Artikel yang ditulis oleh {{ $writer->name }}:</p>
  <ul class="list-group">
    @foreach($articles as $article)
      <li class="list-group-item">
        <a href="/article/{{ $article->id }}">{{ $article->title }}</a>
      </li>
    @endforeach
  </ul>
</div>

```

Code untuk menampilkan penulis dan article-article yang ditulisnya

### Views(Popular)

```

<div class="container">
  <h1>Popular Articles</h1>

  @if($popularArticles->count())
    <ul class="list-group">
      @foreach($popularArticles as $article)
        <li class="list-group-item">
          <a href="{{ route(name: 'article.show', parameters: $article->id) }}">
            <h4>{{ $article->title }}</h4>
          </a>
          <p>{{ Str::limit(value: $article->content, limit: 100) }}</p>
          <small>Views: {{ $article->views }}</small>
        </li>
      @endforeach
    </ul>
  @else
    <p>No popular articles found.</p>
  @endif

```

Code untuk menampilkan popular article serta deskripsinya

### Migration(Writers)

```

class CreateWritersTable extends Migration
{
    0 references | 0 overrides
    public function up(): void
    {
        Schema::create(table: 'writers', callback: function (Blueprint $table): void {
            $table->id();
            $table->string(column: 'name');
            $table->string(column: 'email')->unique();
            $table->timestamps();
        });
    }

    0 references | 0 overrides
    public function down(): void
    {
        Schema::dropIfExists(table: 'writers');
    }
}

```

Migrate attribute-attribute tersebut ke dalam database

### Migration(Article)

```
class CreateArticlesTable extends Migration
{
    0 references | 0 overrides
    public function up(): void
    {
        Schema::create(table: 'articles', callback: function (Blueprint $table): void {
            $table->id();
            $table->string(column: 'title');
            $table->text(column: 'content');
            $table->date(column: 'posting_date');
            $table->foreignId(column: 'writer_id')->constrained(table: 'writers')->onDelete(action: 'cascade');
            $table->string(column: 'image_url')->nullable();
            $table->integer(column: 'views')->default(value: 0);
            $table->timestamps();
        });
    }

    0 references | 0 overrides
    public function down(): void
    {
        Schema::dropIfExists(table: 'articles');
    }
}
```

Migrate attribute-attribute ke dalam database

### Migration(Category)

```
class CreateCategoriesTable extends Migration
{
    0 references | 0 overrides
    public function up(): void
    {
        Schema::create(table: 'categories', callback: function (Blueprint $table): void {
            $table->id();
            $table->string(column: 'name')->unique();
            $table->string(column: 'description')->nullable();
            $table->timestamps();
        });
    }

    0 references | 0 overrides
    public function down(): void
    {
        Schema::dropIfExists(table: 'categories');
    }
}
```

Migrate attribute-attribute ke dalam database

### Seeder(Article)



```

class ArticleSeeder extends Seeder
{
    0 references | 0 overrides
    public function run(): void
    {
        Article::truncate();
        $faker = Faker::create(locale: 'id_ID');

        $writerIds = Writer::pluck(column: 'id')->toArray();

        foreach (range(start: 1, end: 50) as $index) {
            Article::create(attributes: [
                'title' => $faker->sentence,
                'content' => $faker->paragraphs(nb: 5, asText: true),
                'posting_date' => Carbon::now()->subDays(value: rand(min: 1, max: 365))->format(format: 'Y-m-d'),
                'writer_id' => $faker->randomElement(array: $writerIds),
                'image_url' => $faker->imageUrl(width: 400, height: 300, category: 'computer', randomize: true, word: 'Gambar'),
                'views' => $faker->numberBetween(int1: 0, int2: 1000),
            ]);
        }
    }
}

```

Untuk memasukkan data random menggunakan bahasa Indonesia ke dalam attribute-attribute yang sudah di deklarasi serta menambahkan format bila diperlukan

Seeder(Writer)

```

class WriterSeeder extends Seeder
{
    0 references | 0 overrides
    public function run(): void
    {
        $faker = Faker::create(locale: 'id_ID');

        foreach (range(start: 1, end: 10) as $index) {
            Writer::create(attributes: [
                'name' => $faker->name,
                'email' => $faker->unique()->safeEmail,
            ]);
        }
    }
}

```

Untuk memasukkan data random menggunakan bahasa Indonesia ke dalam attribute-attribute yang sudah di deklarasi serta menambahkan format bila diperlukan

## Seeder(Category)

```
0 references | 0 implementations
class CategorySeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    0 references | 0 overrides
    public function run(): void
    {
        Category::create(attributes: ['name' => 'interactive-multimedia', 'description' => 'Kategori untuk Interactive Multimedia']);
        Category::create(attributes: ['name' => 'software-engineering', 'description' => 'Kategori untuk Software Engineering']);
    }
}
```