Quiz Web Programming
Pembuatan Website “EduFun”

Disusun oleh:

Willsens Kiren Alexander / 2702234112 / LA01

Dosen Pengampu:

Reinert Yosua Rumagit, S.Kom., M.TI

School Of Computer Science
Binus University

2025/2026

No WA: 087793762813 | ID Line: will3ns

Pendahuluan

Dokumen ini dibuat untuk memenuhi tugas Ujian Tengah Semester mata kuliah Web Programming. Project ini adalah membuat website "EduFun" menggunakan framework Laravel, Bootstrap, dan database MySQL (via phpMyAdmin).

Sesuai dengan ketentuan NIM Genap, website ini akan menampilkan 2 kategori utama, yaitu Interactive Multimedia dan Software Engineering.

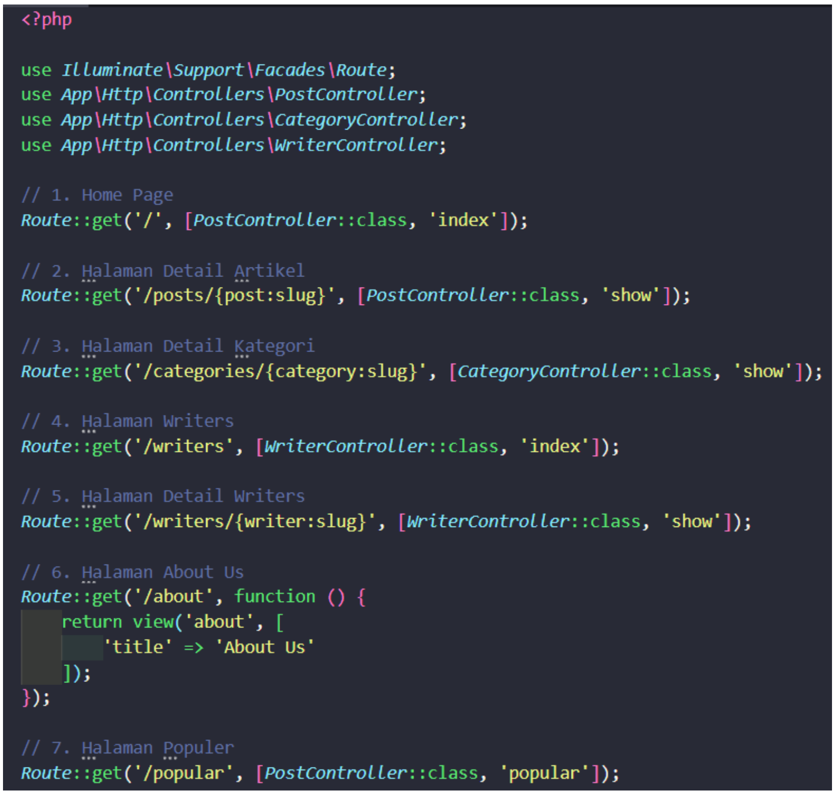
Website ini mengimplementasikan semua komponen utama Laravel yang diwajibkan, yaitu:

1. Routing
2. Model (dan Relasinya)
3. View and Blade
4. Controller
5. Migration
6. Database Seeder

Routing

Routing digunakan untuk mendefinisikan URL website dan menghubungkannya ke Controller.

Snippet code:



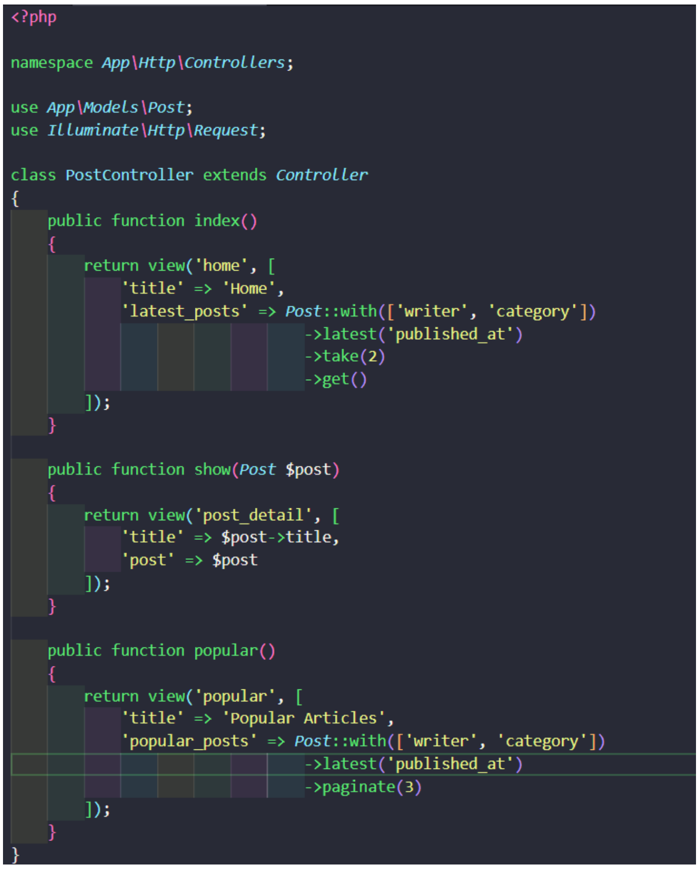
```php
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\PostController;
use App\Http\Controllers\CategoryController;
use App\Http\Controllers\WriterController;

// 1. Home Page
Route::get('/', [PostController::class, 'index']);

// 2. Halaman Detail Artikel
Route::get('/posts/{post:slug}', [PostController::class, 'show']);

// 3. Halaman Detail Kategori
Route::get('/categories/{category:slug}', [CategoryController::class, 'show']);

// 4. Halaman Writers
Route::get('/writers', [WriterController::class, 'index']);

// 5. Halaman Detail Writers
Route::get('/writers/{writer:slug}', [WriterController::class, 'show']);

// 6. Halaman About Us
Route::get('/about', function () {
    return view('about', [
        'title' => 'About Us'
    ]);
});

// 7. Halaman Populer
Route::get('/popular', [PostController::class, 'popular']);
```

Controllers

Controller bertugas mengambil data dari Model dan mengirimkannya ke View. Project ini memiliki 3 Controller utama: PostController, CategoryController, dan WriterController.

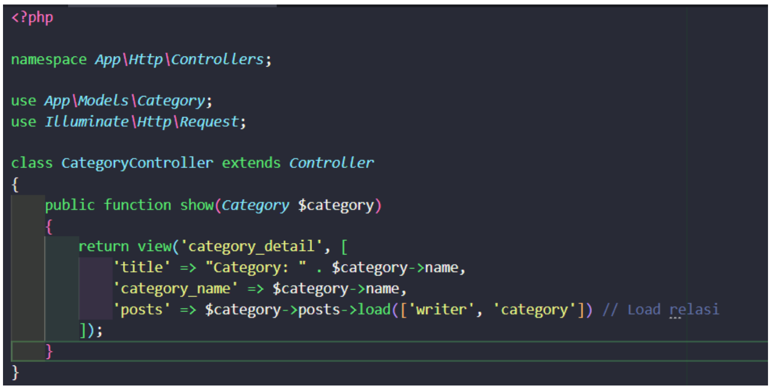
Snippet code masing-masing controller:

Post controller:



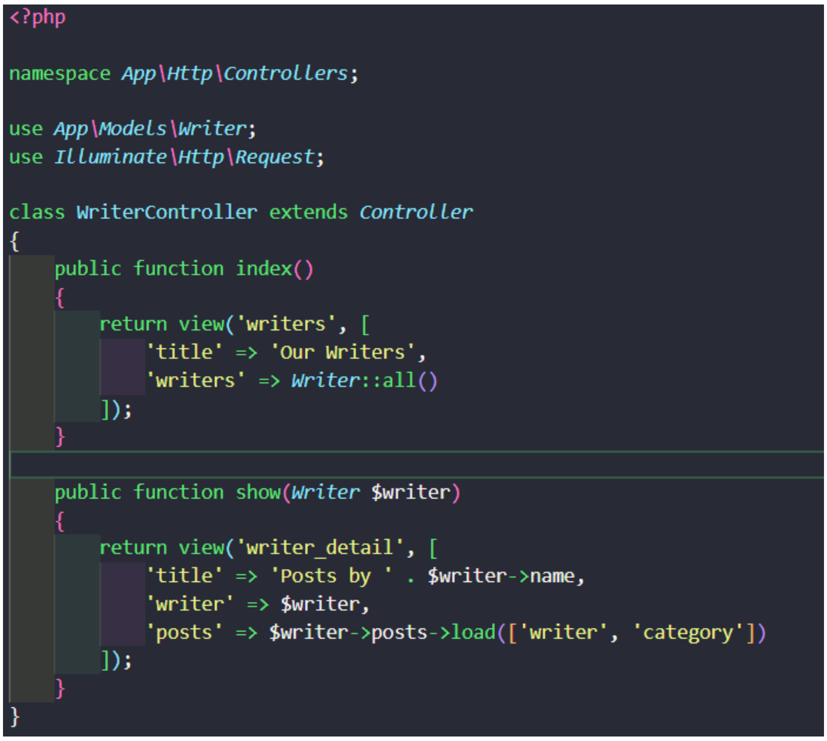
```php
<?php

namespace App\Http\Controllers;

use App\Models\Post;
use Illuminate\Http\Request;

class PostController extends Controller
{
    public function index()
    {
        return view('home', [
            'title' => 'Home',
            'latest_posts' => Post::with(['writer', 'category'])
                                ->latest('published_at')
                                ->take(2)
                                ->get()
        ]);
    }

    public function show(Post $post)
    {
        return view('post_detail', [
            'title' => $post->title,
            'post' => $post
        ]);
    }

    public function popular()
    {
        return view('popular', [
            'title' => 'Popular Articles',
            'popular_posts' => Post::with(['writer', 'category'])
                                ->latest('published_at')
                                ->paginate(3)
        ]);
    }
}
```

Category controller



```php
<?php

namespace App\Http\Controllers;

use App\Models\Category;
use Illuminate\Http\Request;

class CategoryController extends Controller
{
    public function show(Category $category)
    {
        return view('category_detail', [
            'title' => "Category: " . $category->name,
            'category_name' => $category->name,
            'posts' => $category->posts->load(['writer', 'category']) // Load relasi
        ]);
    }
}
```

Writer controller:



```php
<?php

namespace App\Http\Controllers;

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

class WriterController extends Controller
{
    public function index()
    {
        return view('writers', [
            'title' => 'Our Writers',
            'writers' => Writer::all()
        ]);
    }

    public function show(Writer $writer)
    {
        return view('writer_detail', [
            'title' => 'Posts by ' . $writer->name,
            'writer' => $writer,
            'posts' => $writer->posts->load(['writer', 'category'])
        ]);
    }
}
```

Model dan relasi

Model bertugas mengelola data dan relasi antar tabel di database. Terdapat 3 Model:

- Category.php
- Writer.php
- Post.php

Relasi yang digunakan:

One-to-Many:

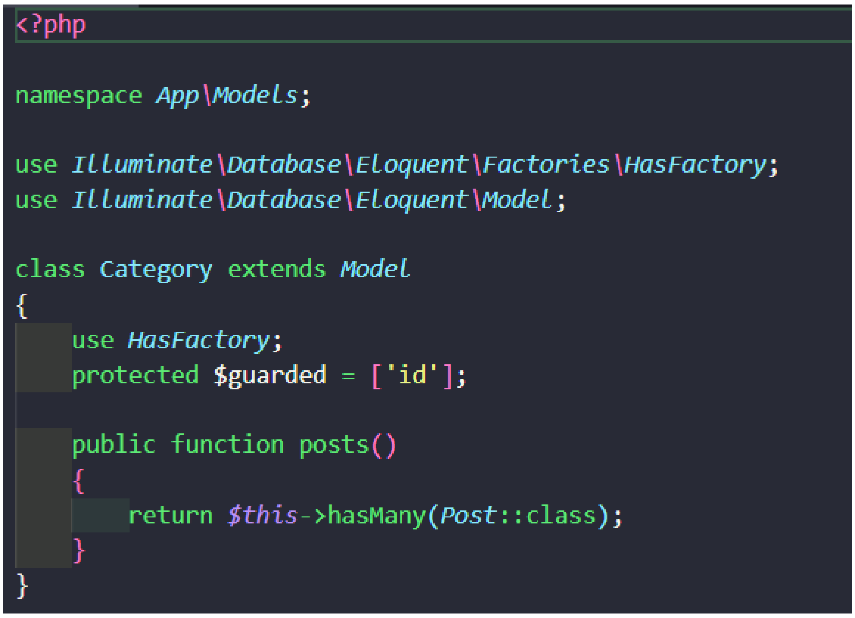
- Satu **Category** memiliki banyak **Post** (**hasMany**).
- Satu **Writer** memiliki banyak **Post** (**hasMany**).

Many-to-One (Inverse):

- Satu **Post** dimiliki oleh satu **Category** (**belongsTo**).
- Satu **Post** dimiliki oleh satu **Writer** (**belongsTo**).

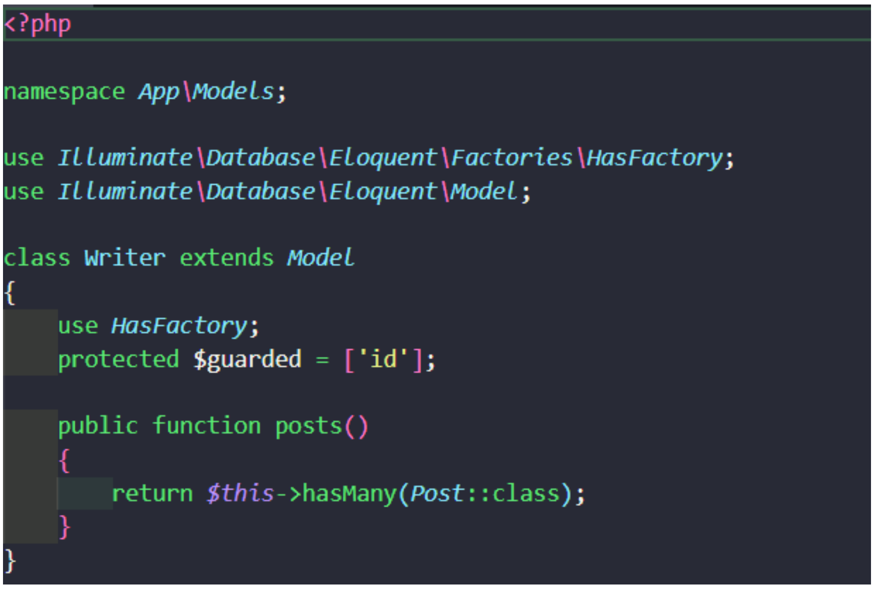
Snippet code untuk masing-masing model:

Category.php



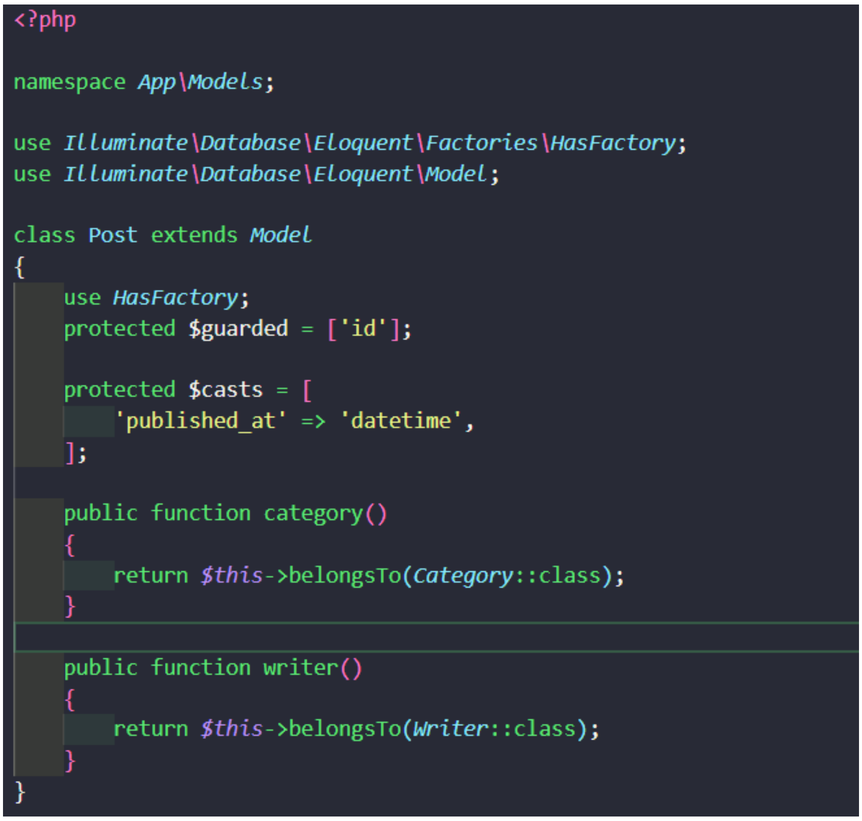
```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Category extends Model
{
    use HasFactory;
    protected $guarded = ['id'];

    public function posts()
    {
        return $this->hasMany(Post::class);
    }
}
```

Writer.php



```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Writer extends Model
{
    use HasFactory;
    protected $guarded = ['id'];

    public function posts()
    {
        return $this->hasMany(Post::class);
    }
}
```

Post.php



```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Post extends Model
{
    use HasFactory;
    protected $guarded = ['id'];

    protected $casts = [
        'published_at' => 'datetime',
    ];

    public function category()
    {
        return $this->belongsTo(Category::class);
    }

    public function writer()
    {
        return $this->belongsTo(Writer::class);
    }
}
```

View dan blade

View dan Blade digunakan untuk menampilkan User Interface. Project ini menggunakan sistem template inheritance:

- layouts/main.blade.php: Sebagai template utama yang berisi Navbar dan Footer.
- home.blade.php, about.blade.php, dll: Sebagai child view yang di-@yield ke dalam main.blade.php.
- components/post-card.blade.php: Sebagai komponen Blade yang reusable untuk menampilkan card artikel di berbagai halaman.
- Data kategori di Navbar dibagikan ke semua view melalui AppServiceProvider.php.

Snippet code: (terdapat pada edufun\resources\views) 🡠 terdapat di link github nantinya

Database (Migration and seeder)

- Migration: Digunakan untuk membuat struktur tabel. Terdapat 3 file migrasi untuk membuat tabel categories, writers, dan posts (lengkap dengan foreign key).
- Seeder: Digunakan untuk mengisi data awal (dummy) ke database.
  - CategorySeeder: Mengisi 2 kategori (Interactive Multimedia & Software Engineering).
  - WriterSeeder: Mengisi 2 data penulis yang relevan.
  - PostSeeder: Mengisi 6 artikel (3 per kategori) lengkap dengan judul, slug, excerpt, dan body.

Snippet code: (terdapat pada edufun\database\migrations dan edufun\database\seeders) 🡠 terdapat di link github nantinya

Lampiran

Untuk link dokumentasi, link soal, dan link githubnya akan berupa drive

Link video: https://drive.google.com/file/d/1x4BXfi_qnASl4u5KJ--dr3UuhmTXxZLK/view?usp=sharing (dalam bentuk video)

Link soal ujian: https://drive.google.com/file/d/1_YLHOLk-c5p9VToS9OvcM6fhTtFDVD0H/view?usp=sharing (dalam bentuk pdf)

Link github: https://github.com/willsens25/Project-EduFun