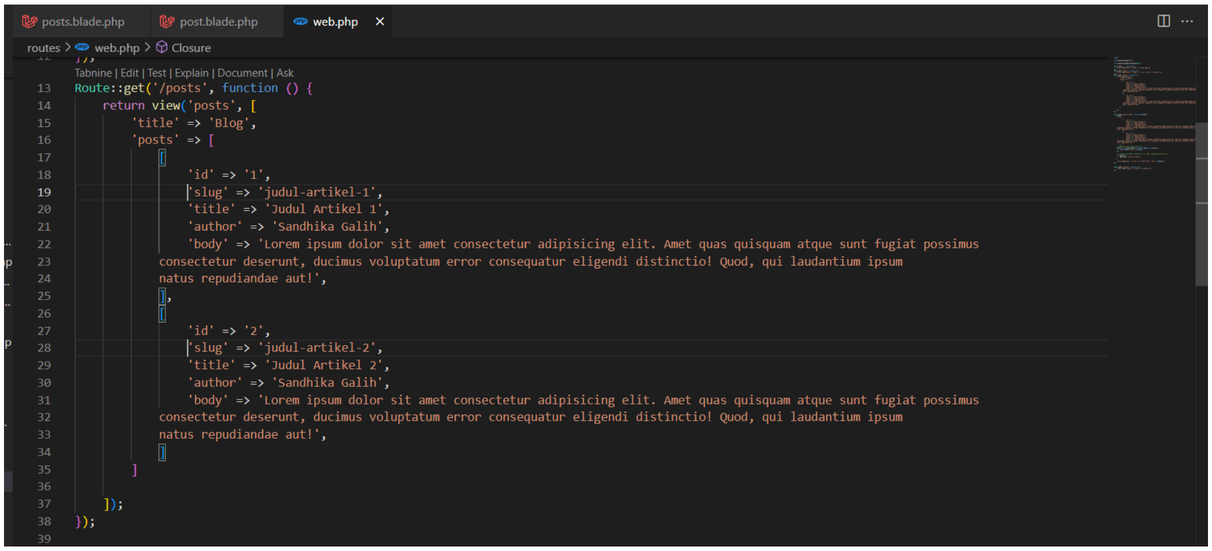
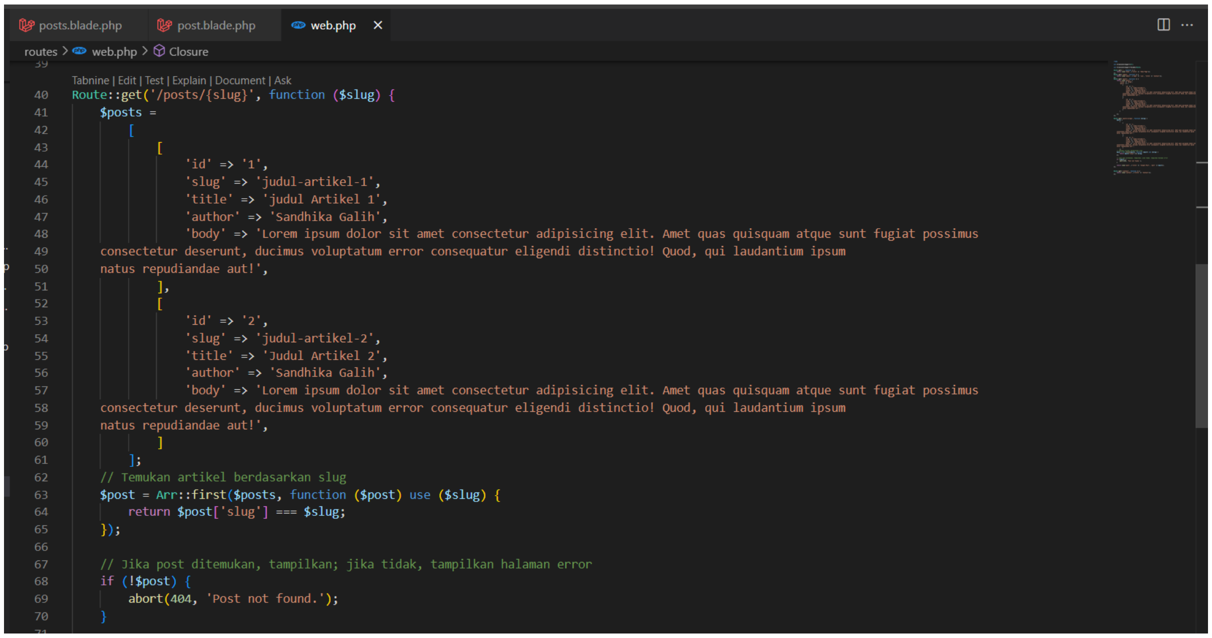
MODEL

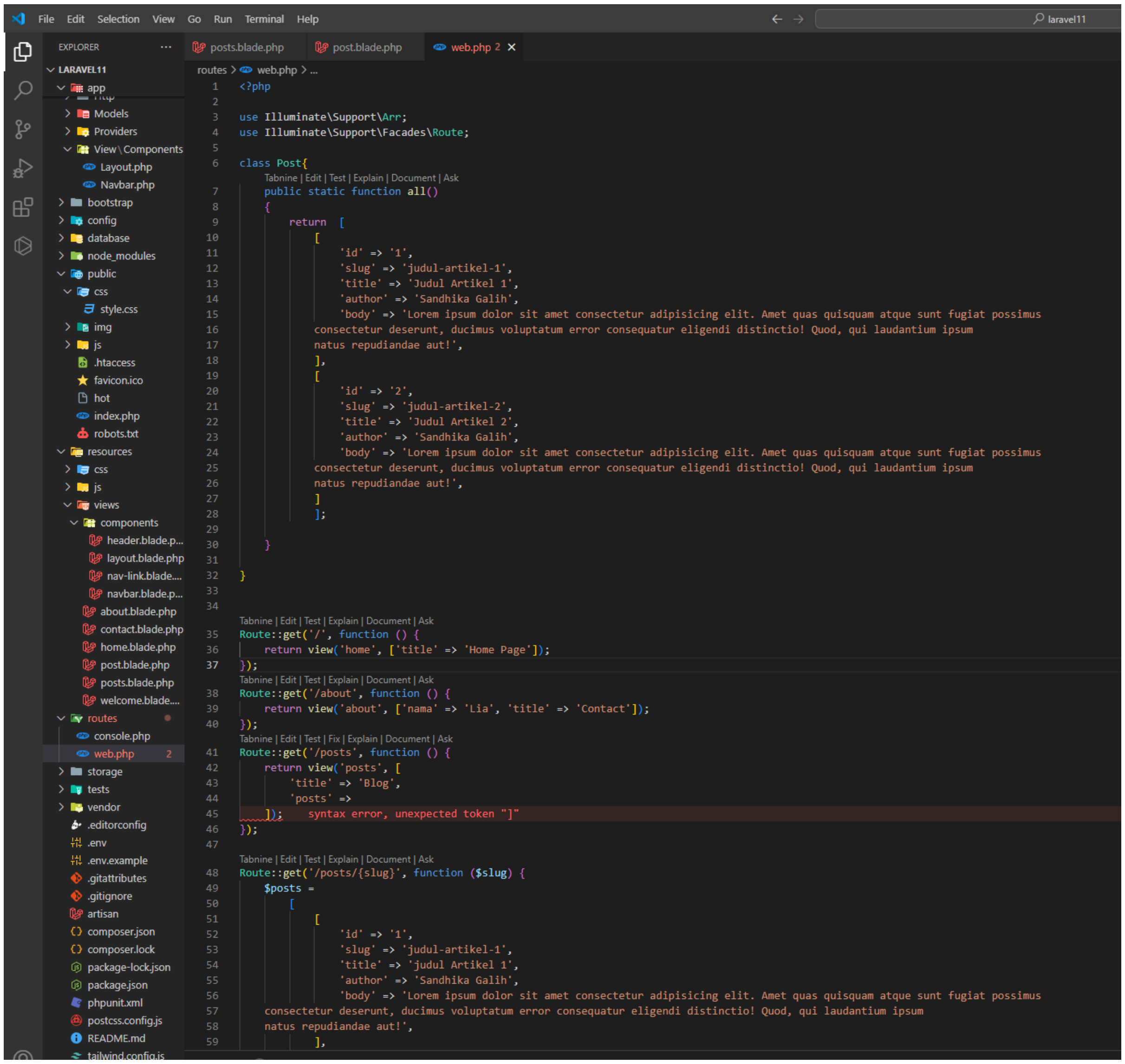

posts.blade.php
post.blade.php
web.php
routes > web.php > Closure
Tabnine | Edit | Test | Explain | Document | Ask

```php
Route::get('/posts', function () {
    return view('posts', [
        'title' => 'Blog',
        'posts' => [
            [
                'id' => '1',
                'slug' => 'judul-artikel-1',
                'title' => 'Judul Artikel 1',
                'author' => 'Sandhika Galih',
                'body' => 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possimus
            consectetur deserunt, ducimus voluptatum error consequatur eligendi distinctio! Quod, qui laudantium ipsum
            natus repudiandae aut!',
            ],
            [
                'id' => '2',
                'slug' => 'judul-artikel-2',
                'title' => 'Judul Artikel 2',
                'author' => 'Sandhika Galih',
                'body' => 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possimus
            consectetur deserunt, ducimus voluptatum error consequatur eligendi distinctio! Quod, qui laudantium ipsum
            natus repudiandae aut!',
            ]
        ]

    ]);
});
```

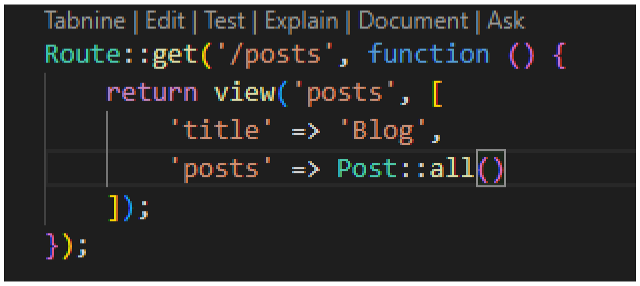

posts.blade.php
post.blade.php
web.php
routes > web.php > Closure
Tabnine | Edit | Test | Explain | Document | Ask

```php
Route::get('/posts/{slug}', function ($slug) {
    $posts =
        [
            [
                'id' => '1',
                'slug' => 'judul-artikel-1',
                'title' => 'judul Artikel 1',
                'author' => 'Sandhika Galih',
                'body' => 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possimus
    consectetur deserunt, ducimus voluptatum error consequatur eligendi distinctio! Quod, qui laudantium ipsum
    natus repudiandae aut!',
            ],
            [
                'id' => '2',
                'slug' => 'judul-artikel-2',
                'title' => 'Judul Artikel 2',
                'author' => 'Sandhika Galih',
                'body' => 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possimus
    consectetur deserunt, ducimus voluptatum error consequatur eligendi distinctio! Quod, qui laudantium ipsum
    natus repudiandae aut!',
            ]
        ];
    // Temukan artikel berdasarkan slug
    $post = Arr::first($posts, function ($post) use ($slug) {
        return $post['slug'] === $slug;
    });

    // Jika post ditemukan, tampilkan; jika tidak, tampilkan halaman error
    if (!$post) {
        abort(404, 'Post not found.');
    }
```

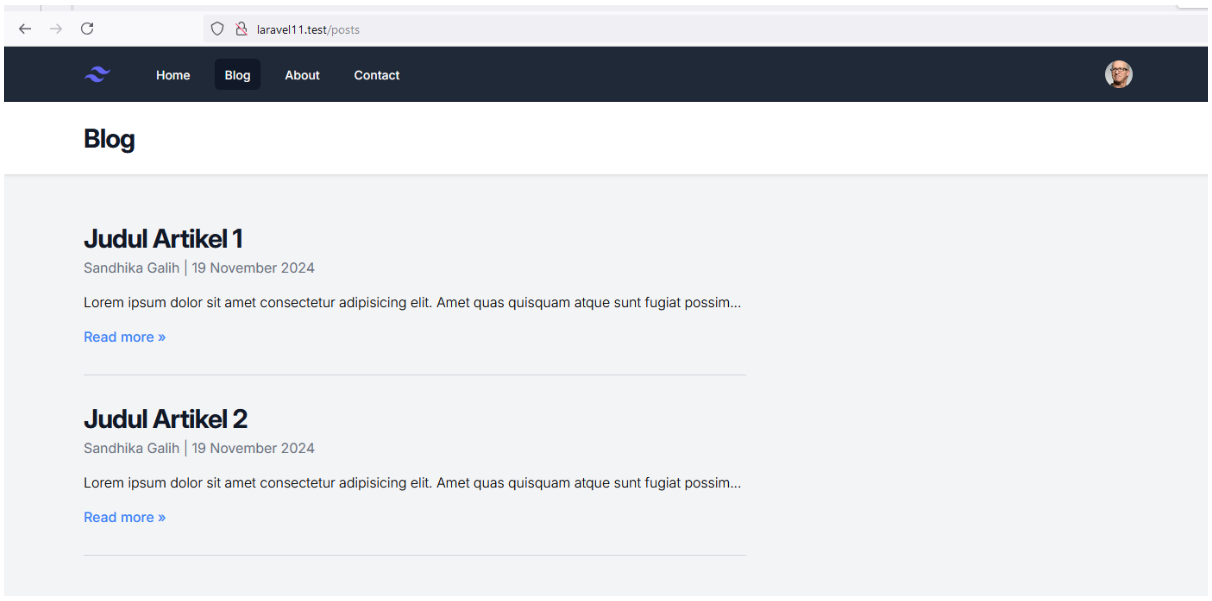
1. Memperbaiki route agar menggunakan 1 sumber data.


```php
<?php

use Illuminate\Support\Arr;
use Illuminate\Support\Facades\Route;

class Post{
    Tabnine | Edit | Test | Explain | Document | Ask
    public static function all()
    {
        return  [
            [
                'id' => '1',
                'slug' => 'judul-artikel-1',
                'title' => 'Judul Artikel 1',
                'author' => 'Sandhika Galih',
                'body' => 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possimus
            consectetur deserunt, ducimus voluptatum error consequatur eligendi distinctio! Quod, qui laudantium ipsum
            natus repudiandae aut!',
            ],
            [
                'id' => '2',
                'slug' => 'judul-artikel-2',
                'title' => 'Judul Artikel 2',
                'author' => 'Sandhika Galih',
                'body' => 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possimus
            consectetur deserunt, ducimus voluptatum error consequatur eligendi distinctio! Quod, qui laudantium ipsum
            natus repudiandae aut!',
            ]
            ];

    }

}


Tabnine | Edit | Test | Explain | Document | Ask
Route::get('/', function () {
    return view('home', ['title' => 'Home Page']);
});
Tabnine | Edit | Test | Explain | Document | Ask
Route::get('/about', function () {
    return view('about', ['nama' => 'Lia', 'title' => 'Contact']);
});
Tabnine | Edit | Test | Fix | Explain | Document | Ask
Route::get('/posts', function () {
    return view('posts', [
        'title' => 'Blog',
        'posts' =>
    ]);    syntax error, unexpected token "]"
});

Tabnine | Edit | Test | Explain | Document | Ask
Route::get('/posts/{slug}', function ($slug) {
    $posts =
        [
            [
                'id' => '1',
                'slug' => 'judul-artikel-1',
                'title' => 'judul Artikel 1',
                'author' => 'Sandhika Galih',
                'body' => 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possimus
    consectetur deserunt, ducimus voluptatum error consequatur eligendi distinctio! Quod, qui laudantium ipsum
    natus repudiandae aut!',
            ],
```

2. Dengan membuat sebuah class yang nanti akan mengembalikan data array posts.


```php
Tabnine | Edit | Test | Explain | Document | Ask
Route::get('/posts', function () {
    return view('posts', [
        'title' => 'Blog',
        'posts' => Post::all()
    ]);
});
```

3. Data yang dikirim kita ambil dari class Post dengan memanggil method staticnya ::all().



laravel11.test/posts

Blog

Judul Artikel 1

Sandhika Galih | 19 November 2024

Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possim...

Read more »

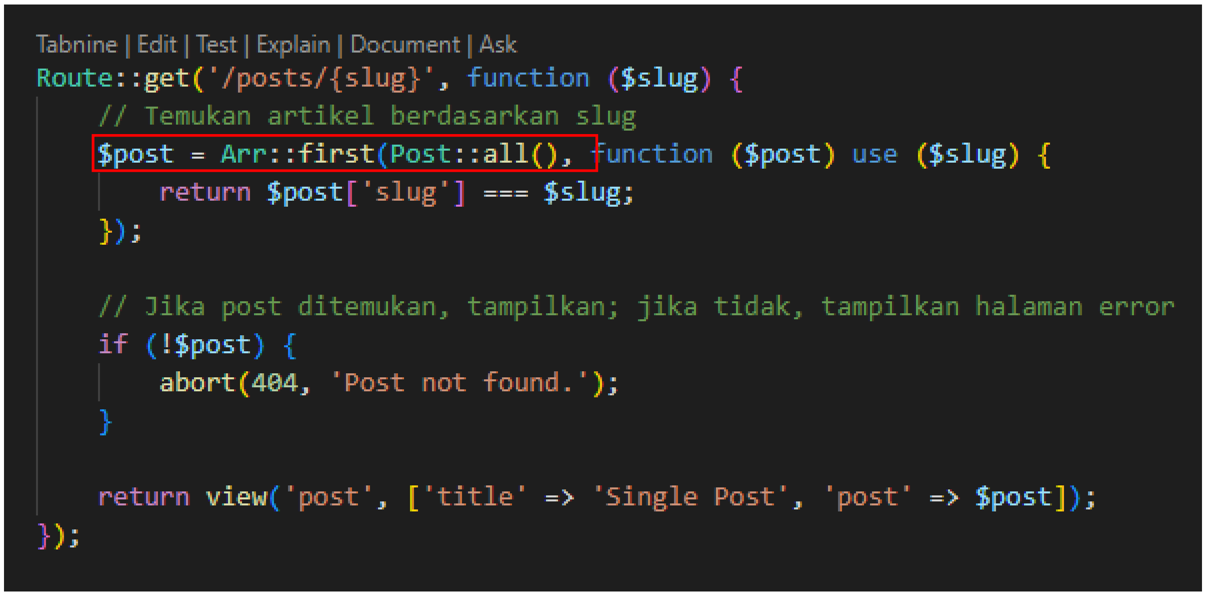
Judul Artikel 2

Sandhika Galih | 19 November 2024

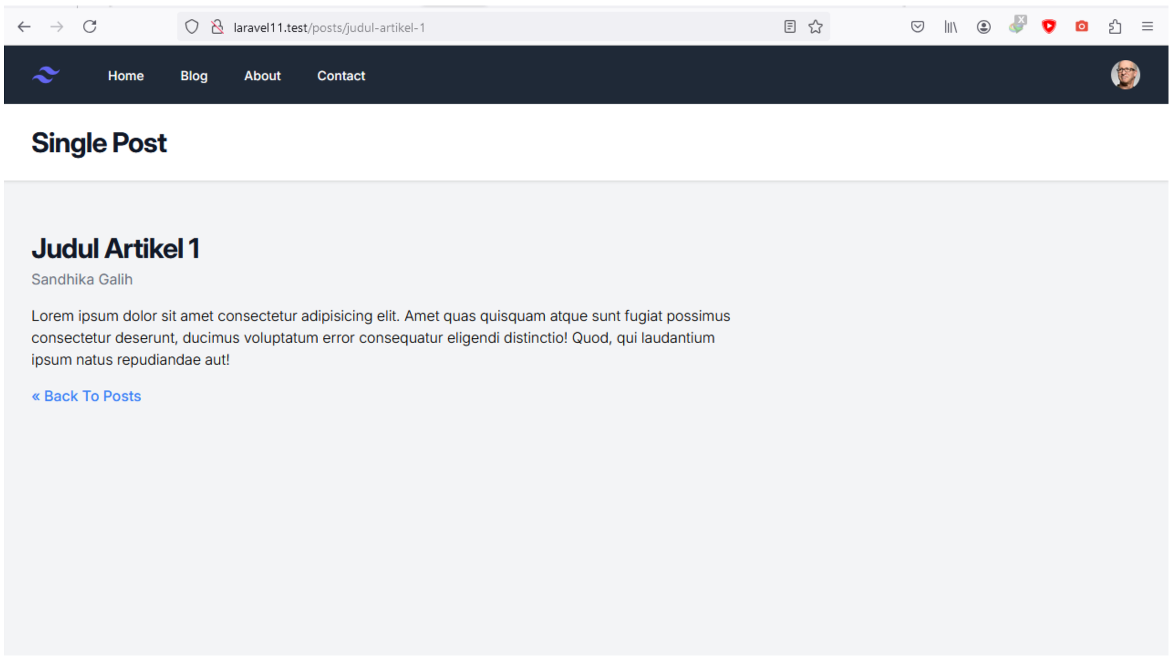
Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possim...

Read more »

4. Yang mana kalau dijalankan masih sama.



```
Tabnine | Edit | Test | Explain | Document | Ask
Route::get('/posts/{slug}', function ($slug) {
    // Temukan artikel berdasarkan slug
    $post = Arr::first(Post::all(), function ($post) use ($slug) {
        return $post['slug'] === $slug;
    });

    // Jika post ditemukan, tampilkan; jika tidak, tampilkan halaman error
    if (!$post) {
        abort(404, 'Post not found.');
    }

    return view('post', ['title' => 'Single Post', 'post' => $post]);
});
```



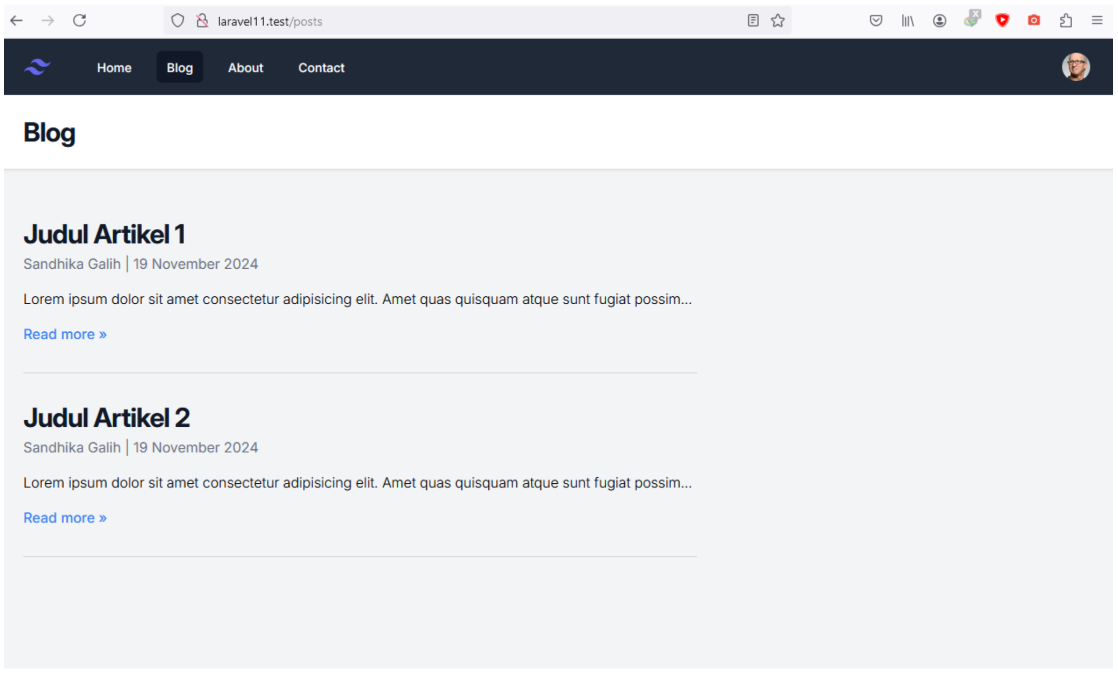
Single Post

Judul Artikel 1

Sandhika Galih

Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possimus consectetur deserunt, ducimus voluptatum error consequatur eligendi distinctio! Quod, qui laudantium ipsum natus repudiandae aut!

« Back To Posts



Blog

Judul Artikel 1

Sandhika Galih | 19 November 2024

Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possim...

Read more »

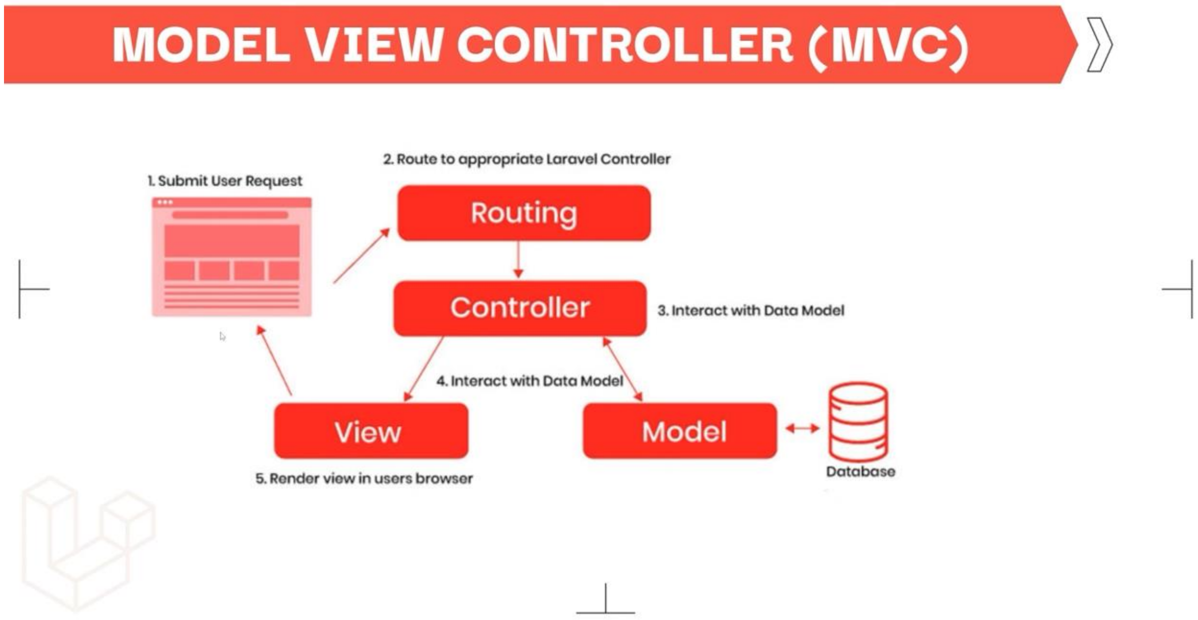
Judul Artikel 2

Sandhika Galih | 19 November 2024

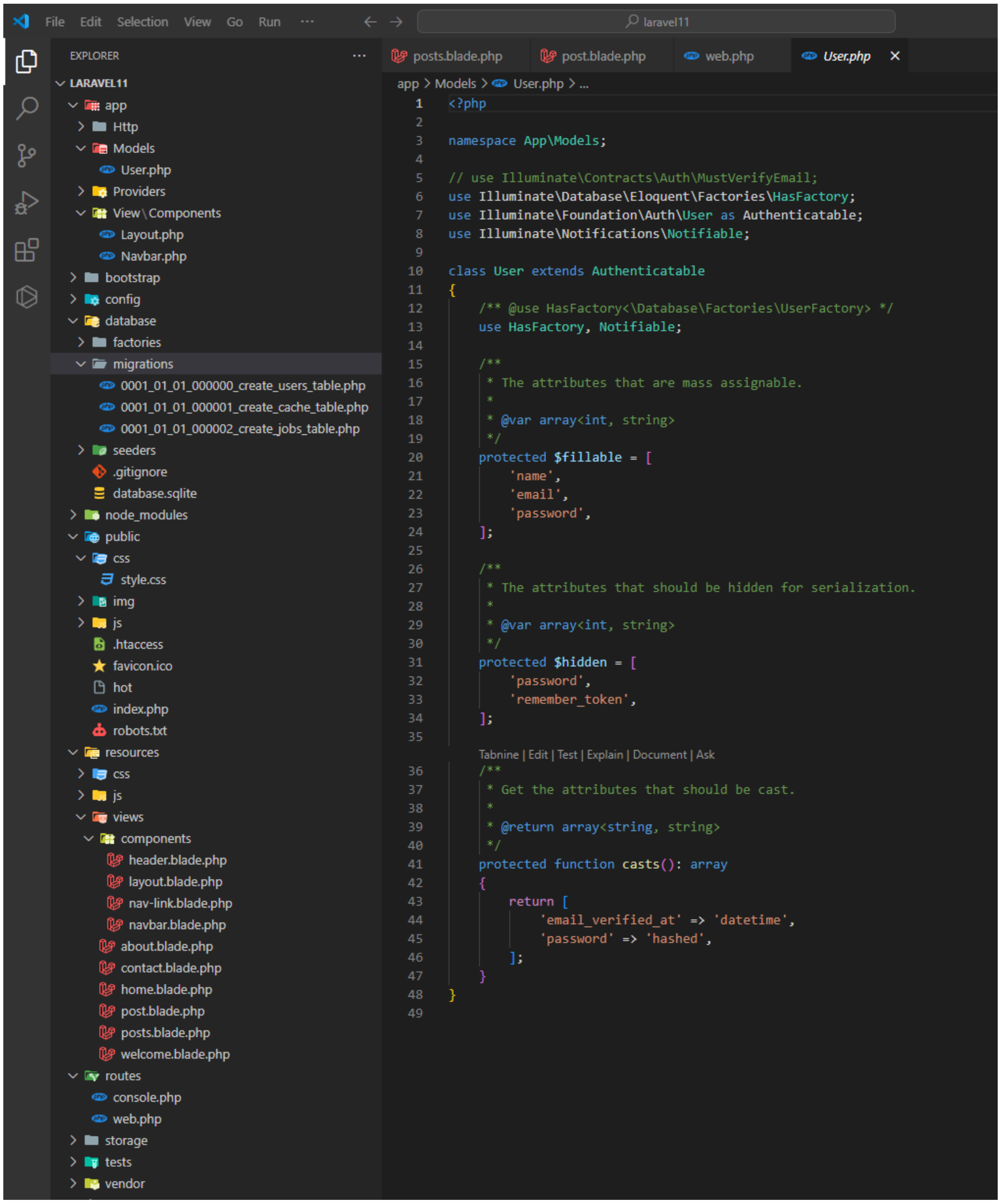
Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possim...

Read more »

5. Kita lakukan hal yang sama jika mau mengambil data di single post nya. Namun ada 1 masalah, yaitu kita tidak mungkin menaruh class post di route yang mana peruntukannya juga sudah berbeda. Route digunakan untuk penjaluran dari request.

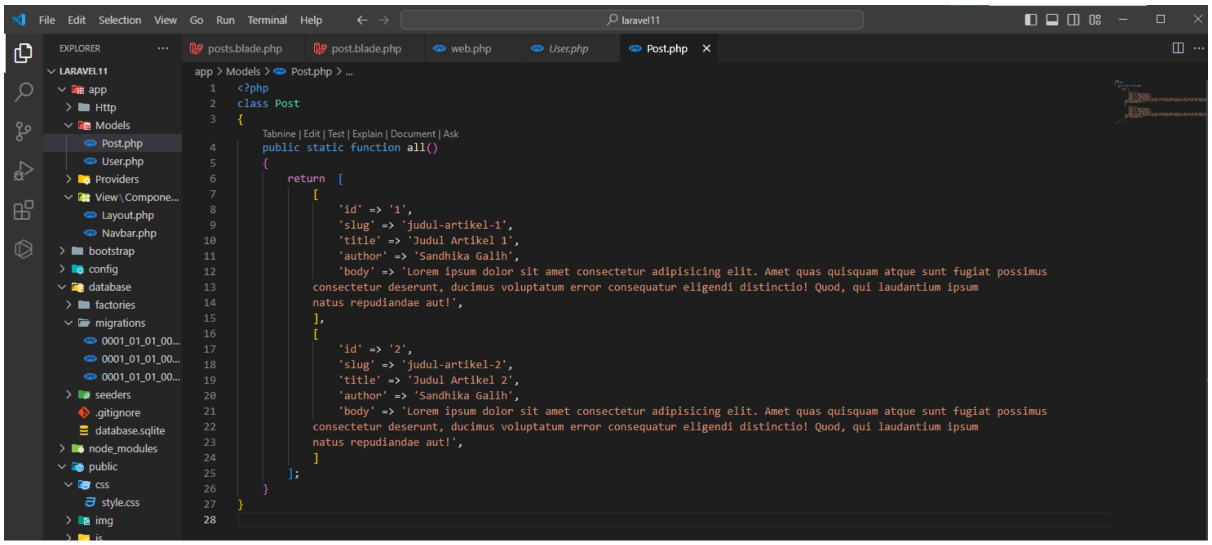

MODEL VIEW CONTROLLER (MVC)
1. Submit User Request
2. Route to appropriate Laravel Controller
Routing
Controller
3. Interact with Data Model
4. Interact with Data Model
View
Model
Database
5. Render view in users browser

6. Berdasarkan konsep MVC, Class post kita pindahkan ke folder model.



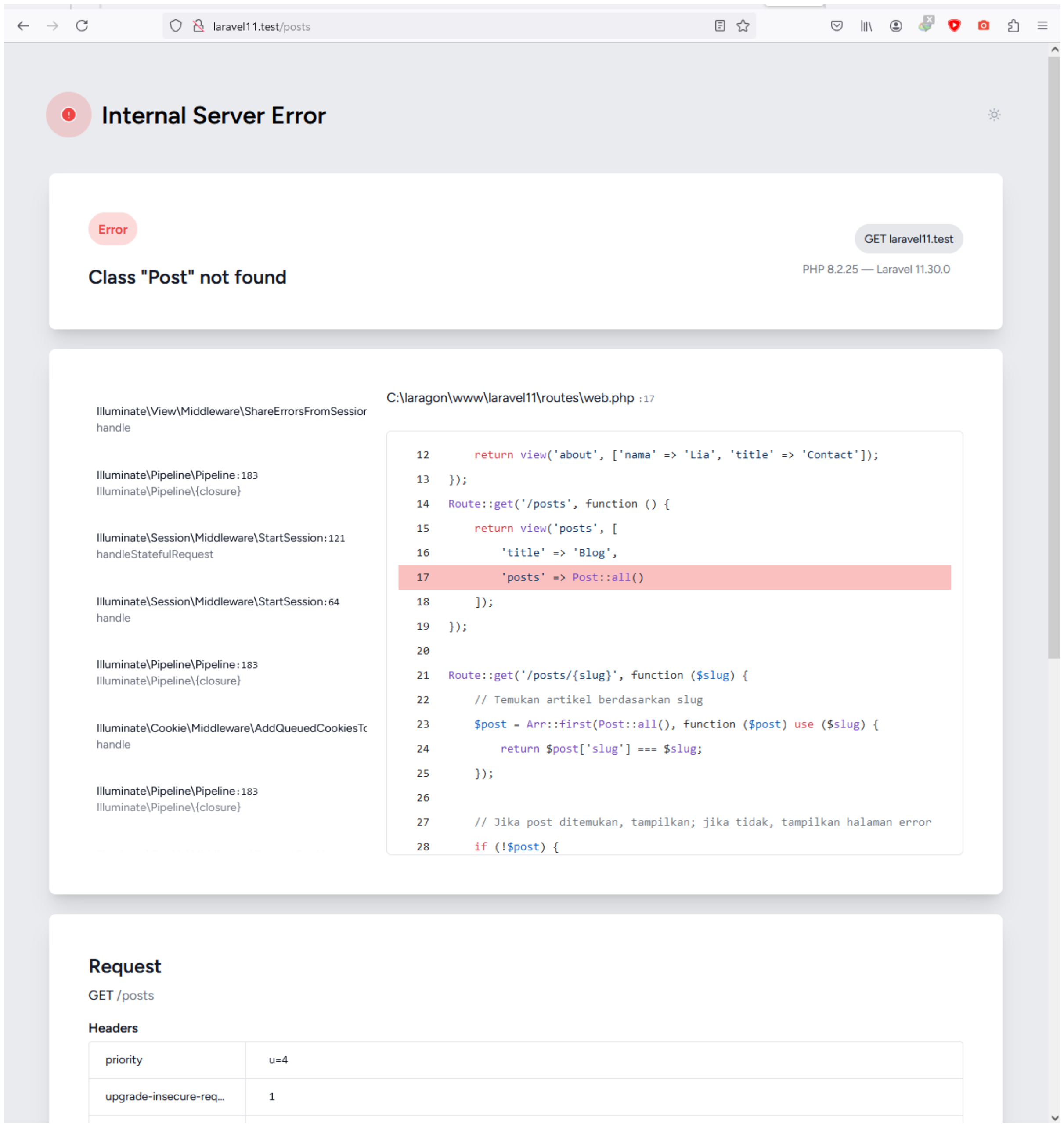
```php
<?php

namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;

class User extends Authenticatable
{
    /** @use HasFactory<\Database\Factories\UserFactory> */
    use HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'email',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * Get the attributes that should be cast.
     *
     * @return array<string, string>
     */
    protected function casts(): array
    {
        return [
            'email_verified_at' => 'datetime',
            'password' => 'hashed',
        ];
    }
}
```

7. Pada class ini akan me-representasikan tabel yang ada di database. Class ini singular dari tabel yang biasa punya pasangan di migration. Ada migration default yang namanya users. Migration ini biasanya untuk membuat tabel users dan pasangan modelnya adalah user.



```php
<?php
class Post
{
    public static function all()
    {
        return  [
            [
                'id' => '1',
                'slug' => 'judul-artikel-1',
                'title' => 'Judul Artikel 1',
                'author' => 'Sandhika Galih',
                'body' => 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possimus
            consectetur deserunt, ducimus voluptatum error consequatur eligendi distinctio! Quod, qui laudantium ipsum
            natus repudiandae aut!',
            ],
            [
                'id' => '2',
                'slug' => 'judul-artikel-2',
                'title' => 'Judul Artikel 2',
                'author' => 'Sandhika Galih',
                'body' => 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possimus
            consectetur deserunt, ducimus voluptatum error consequatur eligendi distinctio! Quod, qui laudantium ipsum
            natus repudiandae aut!',
            ]
        ];
    }
}
```

8. Dengan contoh tadi jika kita membuat class post yang isinya diambil dari class post yang tadi di route. Namun route kita tidak kenal model post yang sudah kita buat. Jadi model post nya tidak otomatis bisa langsung kita gunakan.



laravel11.test/posts

Internal Server Error

Error

Class "Post" not found

GET laravel11.test

PHP 8.2.25 — Laravel 11.30.0

Illuminate\View\Middleware\ShareErrorsFromSessior
handle

Illuminate\Pipeline\Pipeline:183
Illuminate\Pipeline\{closure}

Illuminate\Session\Middleware\StartSession:121
handleStatefulRequest

Illuminate\Session\Middleware\StartSession:64
handle

Illuminate\Pipeline\Pipeline:183
Illuminate\Pipeline\{closure}

Illuminate\Cookie\Middleware\AddQueuedCookiesTc
handle

Illuminate\Pipeline\Pipeline:183
Illuminate\Pipeline\{closure}

C:\laragon\www\laravel11\routes\web.php :17

```
    return view('about', ['nama' => 'Lia', 'title' => 'Contact']);
});
Route::get('/posts', function () {
    return view('posts', [
        'title' => 'Blog',
        'posts' => Post::all()
    ]);
});

Route::get('/posts/{slug}', function ($slug) {
    // Temukan artikel berdasarkan slug
    $post = Arr::first(Post::all(), function ($post) use ($slug) {
        return $post['slug'] === $slug;
    });

    // Jika post ditemukan, tampilkan; jika tidak, tampilkan halaman error
    if (!$post) {
```

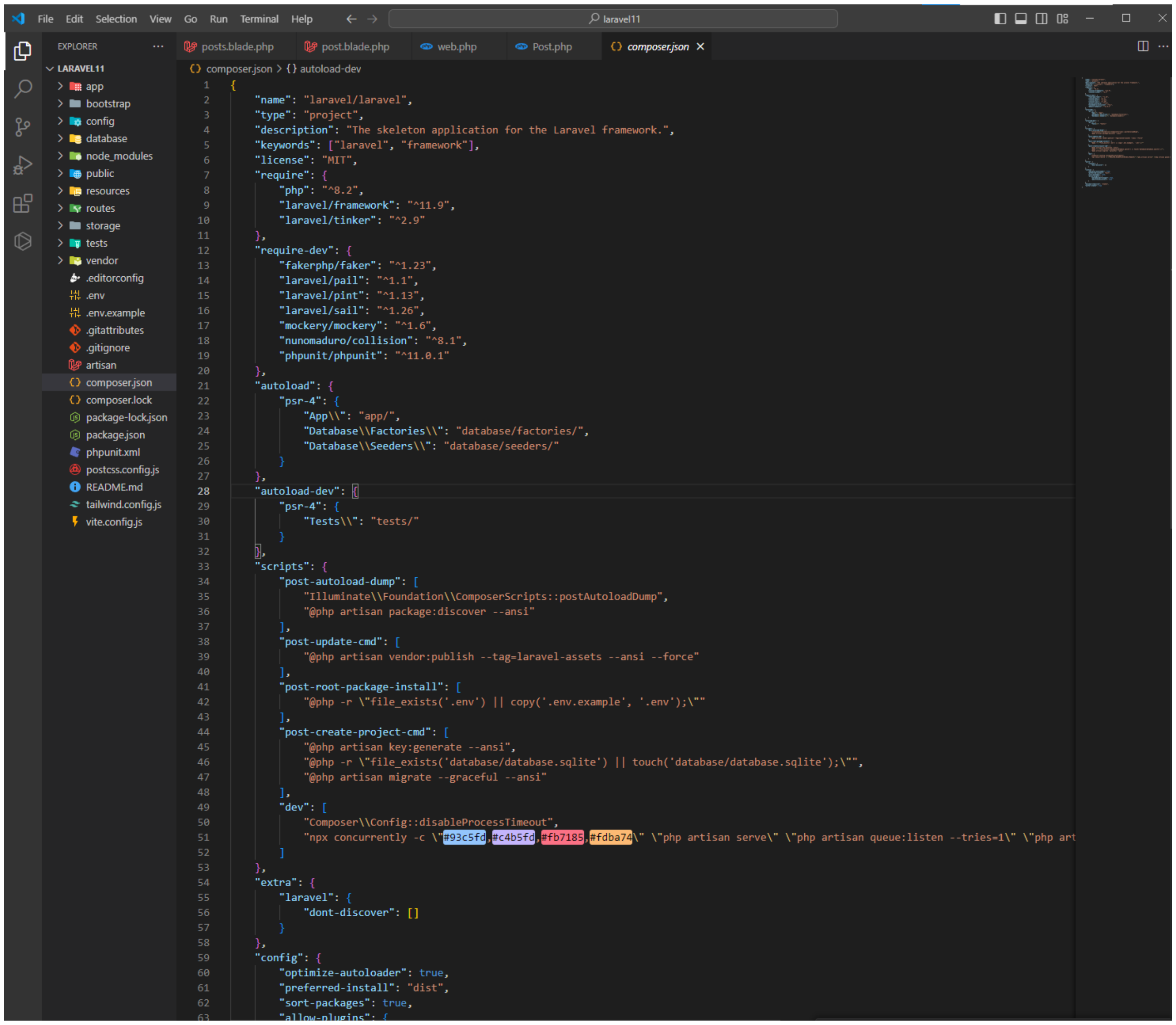
Request

GET /posts

Headers

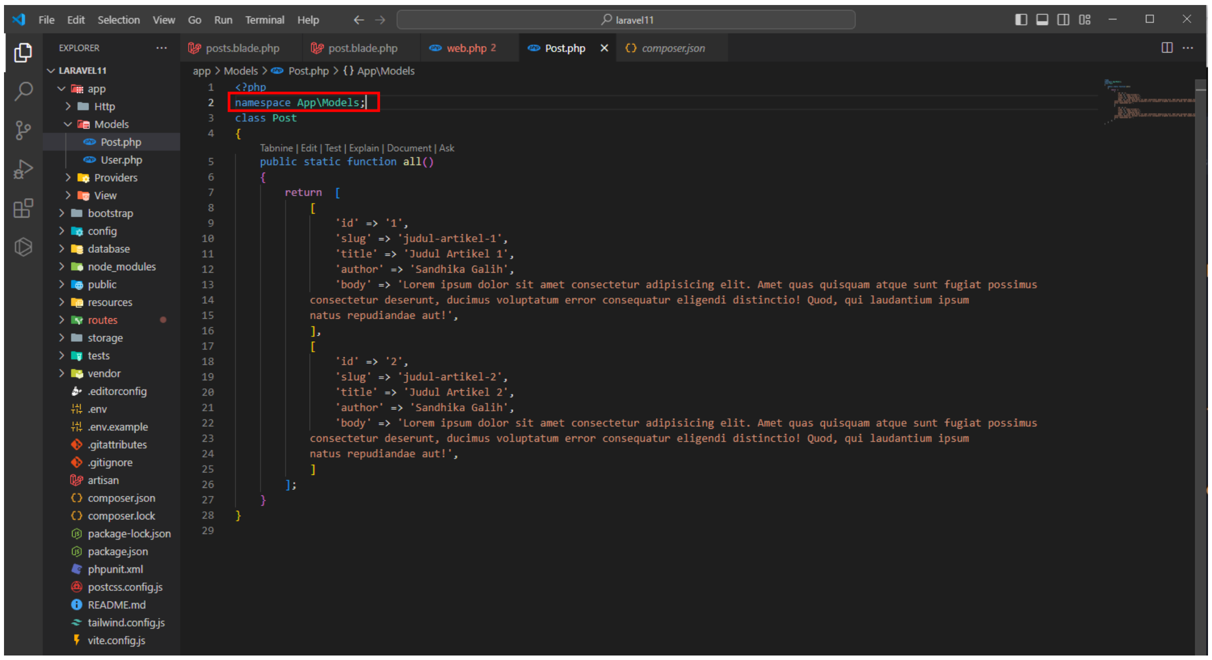
priority	u=4
upgrade-insecure-req...	1

9. Jika kita balik ke halaman web, maka akan error.



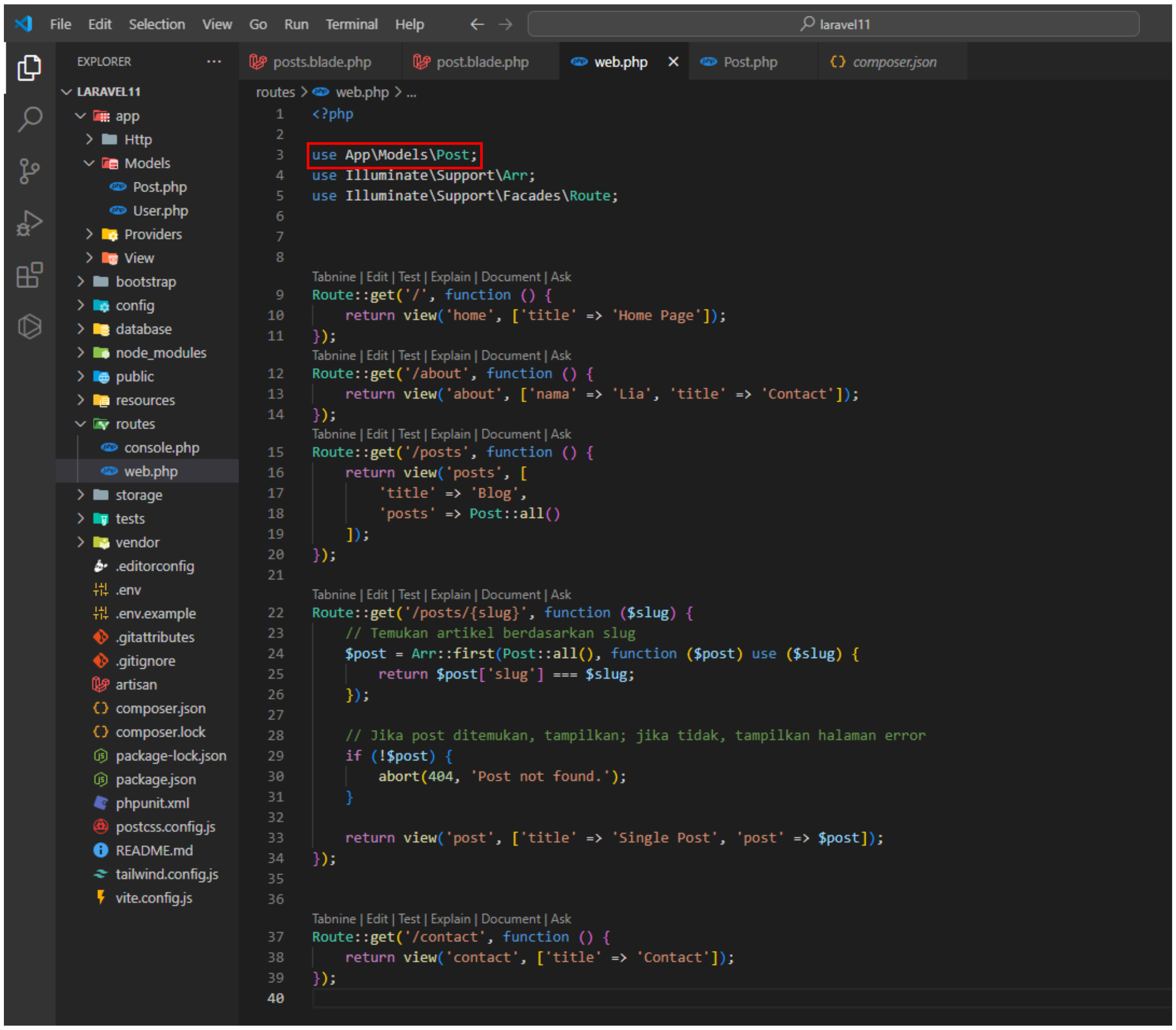
```json
{
    "name": "laravel/laravel",
    "type": "project",
    "description": "The skeleton application for the Laravel framework.",
    "keywords": ["laravel", "framework"],
    "license": "MIT",
    "require": {
        "php": "^8.2",
        "laravel/framework": "^11.9",
        "laravel/tinker": "^2.9"
    },
    "require-dev": {
        "fakerphp/faker": "^1.23",
        "laravel/pail": "^1.1",
        "laravel/pint": "^1.13",
        "laravel/sail": "^1.26",
        "mockery/mockery": "^1.6",
        "nunomaduro/collision": "^8.1",
        "phpunit/phpunit": "^11.0.1"
    },
    "autoload": {
        "psr-4": {
            "App\\": "app/",
            "Database\\Factories\\": "database/factories/",
            "Database\\Seeders\\": "database/seeders/"
        }
    },
    "autoload-dev": {
        "psr-4": {
            "Tests\\": "tests/"
        }
    },
    "scripts": {
        "post-autoload-dump": [
            "Illuminate\\Foundation\\ComposerScripts::postAutoloadDump",
            "@php artisan package:discover --ansi"
        ],
        "post-update-cmd": [
            "@php artisan vendor:publish --tag=laravel-assets --ansi --force"
        ],
        "post-root-package-install": [
            "@php -r \"file_exists('.env') || copy('.env.example', '.env');\""
        ],
        "post-create-project-cmd": [
            "@php artisan key:generate --ansi",
            "@php -r \"file_exists('database/database.sqlite') || touch('database/database.sqlite');\"",
            "@php artisan migrate --graceful --ansi"
        ],
        "dev": [
            "Composer\\Config::disableProcessTimeout",
            "npx concurrently -c \"#93c5fd,#c4b5fd,#fb7185,#fdba74\" \"php artisan serve\" \"php artisan queue:listen --tries=1\" \"php art
        ]
    },
    "extra": {
        "laravel": {
            "dont-discover": []
        }
    },
    "config": {
        "optimize-autoloader": true,
        "preferred-install": "dist",
        "sort-packages": true,
        "allow-plugins": {
```

10. Kita akan memanfaatkan proses autoloading yang ada di laravel. Class post kita tidak terbaca karena belum tersimpan di dalam sebuah namespace. Namespace adalah sebuah teknik untuk menyimpan class ke dalam folder yang spesifik agar tidak bentrok dengan class lain yang namanya sama.



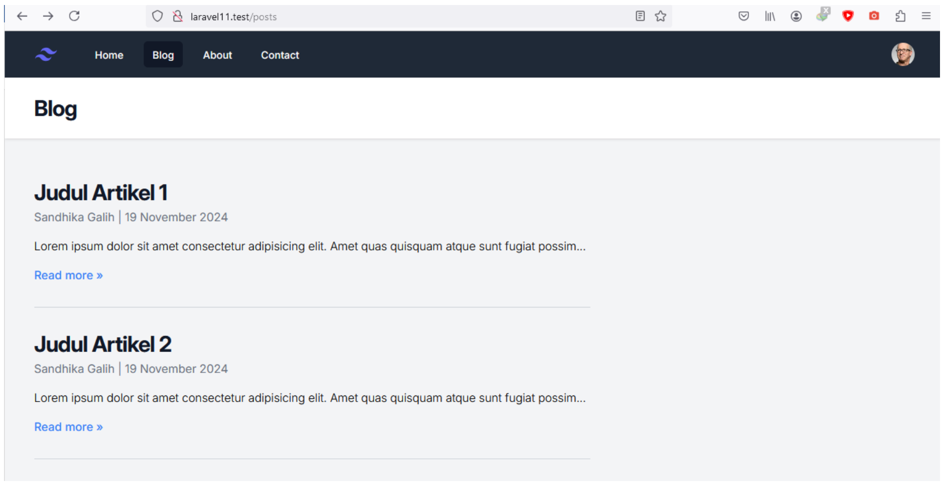
```php
<?php
namespace App\Models;
class Post
{
    public static function all()
    {
        return  [
            [
                'id' => '1',
                'slug' => 'judul-artikel-1',
                'title' => 'Judul Artikel 1',
                'author' => 'Sandhika Galih',
                'body' => 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possimus
            consectetur deserunt, ducimus voluptatum error consequatur eligendi distinctio! Quod, qui laudantium ipsum
            natus repudiandae aut!',
            ],
            [
                'id' => '2',
                'slug' => 'judul-artikel-2',
                'title' => 'Judul Artikel 2',
                'author' => 'Sandhika Galih',
                'body' => 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possimus
            consectetur deserunt, ducimus voluptatum error consequatur eligendi distinctio! Quod, qui laudantium ipsum
            natus repudiandae aut!',
            ]
        ];
    }
}
```

11. Namspace ini untuk memberitahu bahwa class post berada di dalam folder App\folder Model.



```php
<?php

use App\Models\Post;
use Illuminate\Support\Arr;
use Illuminate\Support\Facades\Route;



Route::get('/', function () {
    return view('home', ['title' => 'Home Page']);
});
Route::get('/about', function () {
    return view('about', ['nama' => 'Lia', 'title' => 'Contact']);
});
Route::get('/posts', function () {
    return view('posts', [
        'title' => 'Blog',
        'posts' => Post::all()
    ]);
});

Route::get('/posts/{slug}', function ($slug) {
    // Temukan artikel berdasarkan slug
    $post = Arr::first(Post::all(), function ($post) use ($slug) {
        return $post['slug'] === $slug;
    });

    // Jika post ditemukan, tampilkan; jika tidak, tampilkan halaman error
    if (!$post) {
        abort(404, 'Post not found.');
    }

    return view('post', ['title' => 'Single Post', 'post' => $post]);
});


Route::get('/contact', function () {
    return view('contact', ['title' => 'Contact']);
});
```

12. Tinggal kita panggil di file web.



laravel11.test/posts

Home Blog About Contact

Blog

Judul Artikel 1

Sandhika Galih | 19 November 2024

Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possim...

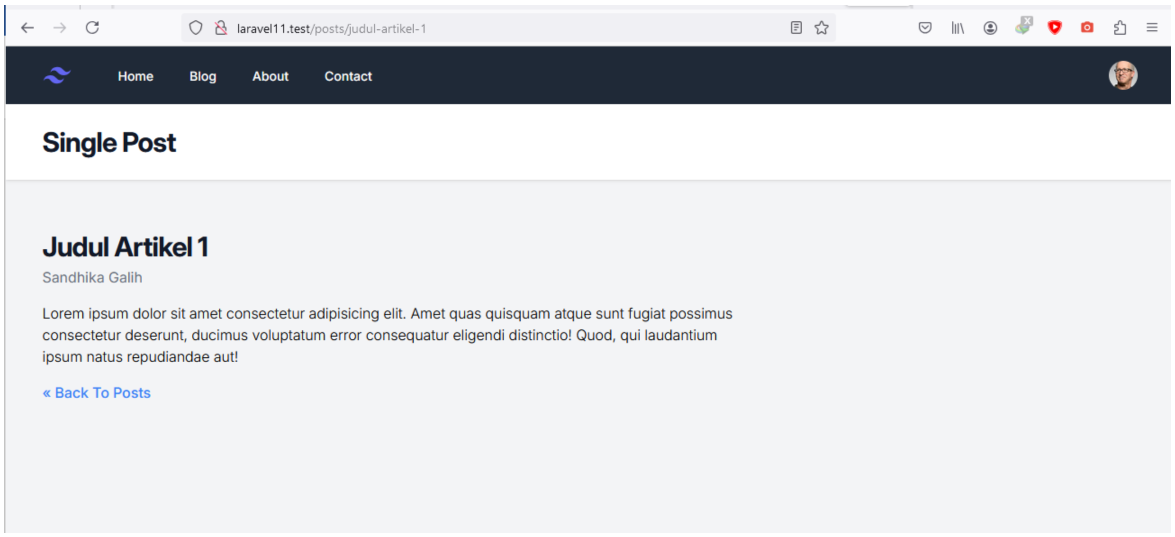
Read more »

Judul Artikel 2

Sandhika Galih | 19 November 2024

Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possim...

Read more »



Single Post

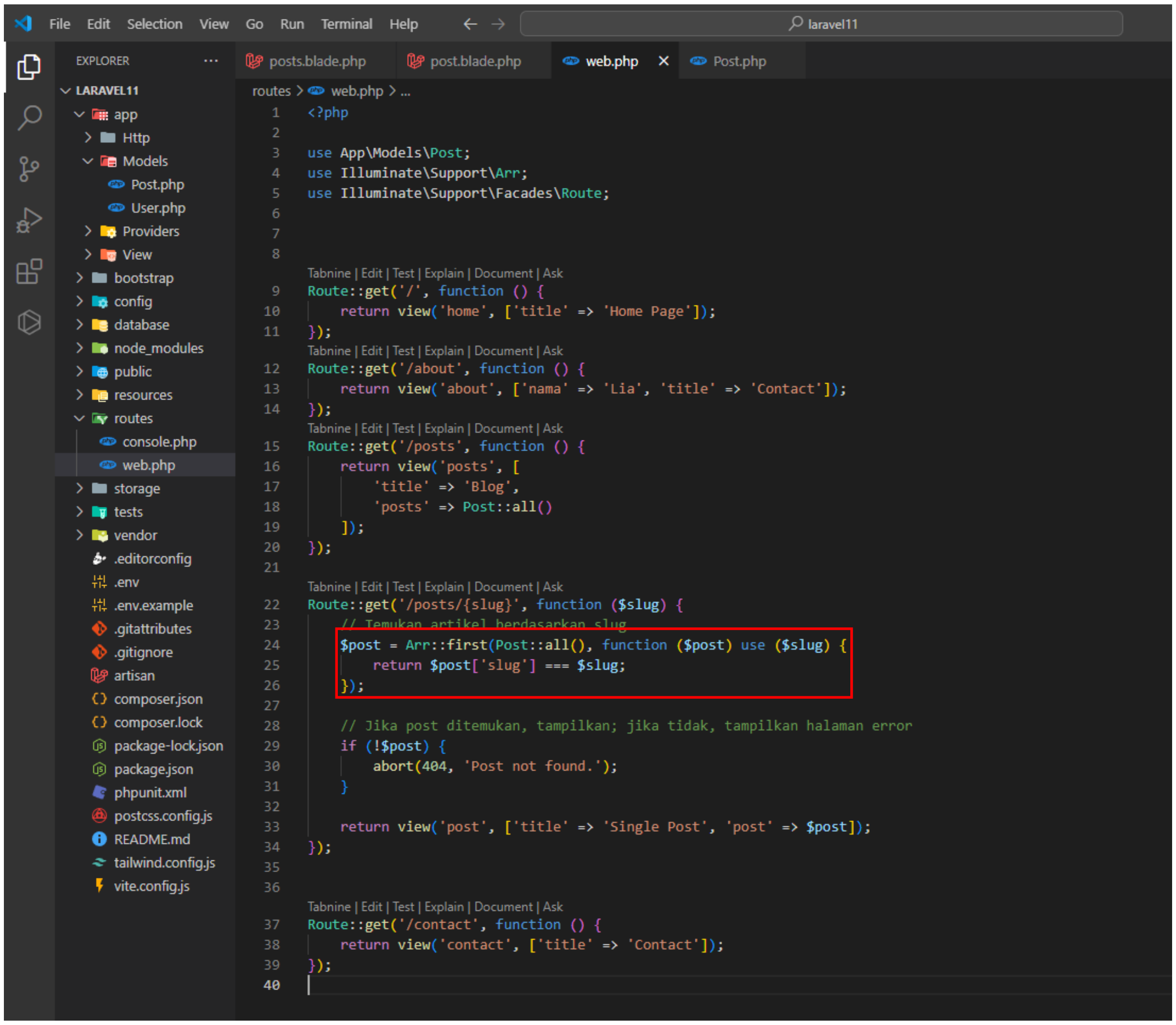
Judul Artikel 1

Sandhika Galih

Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possimus consectetur deserunt, ducimus voluptatum error consequatur eligendi distinctio! Quod, qui laudantium ipsum natus repudiandae aut!

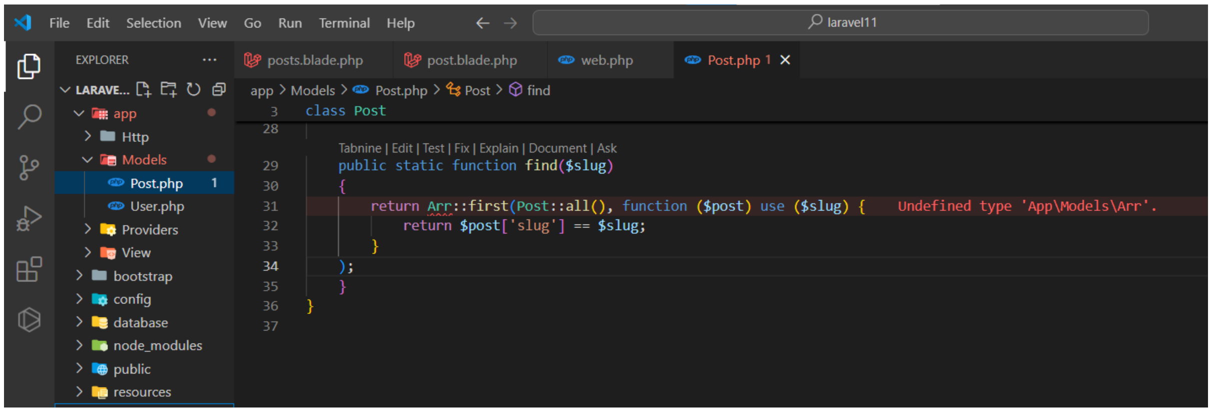
« Back To Posts

13. Hasilnya jika kita kembali ke halaman web.

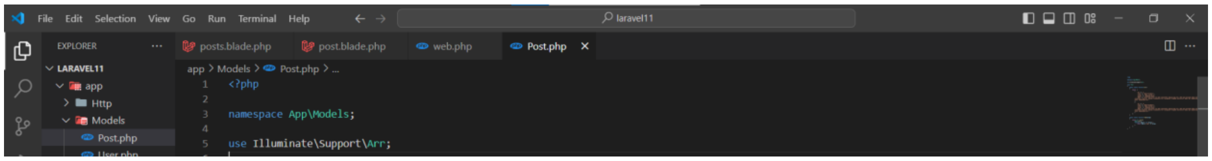


```php
<?php

use App\Models\Post;
use Illuminate\Support\Arr;
use Illuminate\Support\Facades\Route;



Tabnine | Edit | Test | Explain | Document | Ask
Route::get('/', function () {
    return view('home', ['title' => 'Home Page']);
});
Tabnine | Edit | Test | Explain | Document | Ask
Route::get('/about', function () {
    return view('about', ['nama' => 'Lia', 'title' => 'Contact']);
});
Tabnine | Edit | Test | Explain | Document | Ask
Route::get('/posts', function () {
    return view('posts', [
        'title' => 'Blog',
        'posts' => Post::all()
    ]);
});

Tabnine | Edit | Test | Explain | Document | Ask
Route::get('/posts/{slug}', function ($slug) {
    // Temukan artikel berdasarkan slug
    $post = Arr::first(Post::all(), function ($post) use ($slug) {
        return $post['slug'] === $slug;
    });

    // Jika post ditemukan, tampilkan; jika tidak, tampilkan halaman error
    if (!$post) {
        abort(404, 'Post not found.');
    }

    return view('post', ['title' => 'Single Post', 'post' => $post]);
});


Tabnine | Edit | Test | Explain | Document | Ask
Route::get('/contact', function () {
    return view('contact', ['title' => 'Contact']);
});

```

14. Sekarang kita perbaiki model agar dapat melakukan pencarian secara spesifik atau pencarian single post menggunakan static method telah kita buat. Untuk sekarang proses pencarian dilakukan di route. Pada kotak merah seharusnya itu tugas dari model kasrena melakukan pencarian atau merubah data masuk ke kategori bisnis logic yang seharusnya dibebankan pada model.

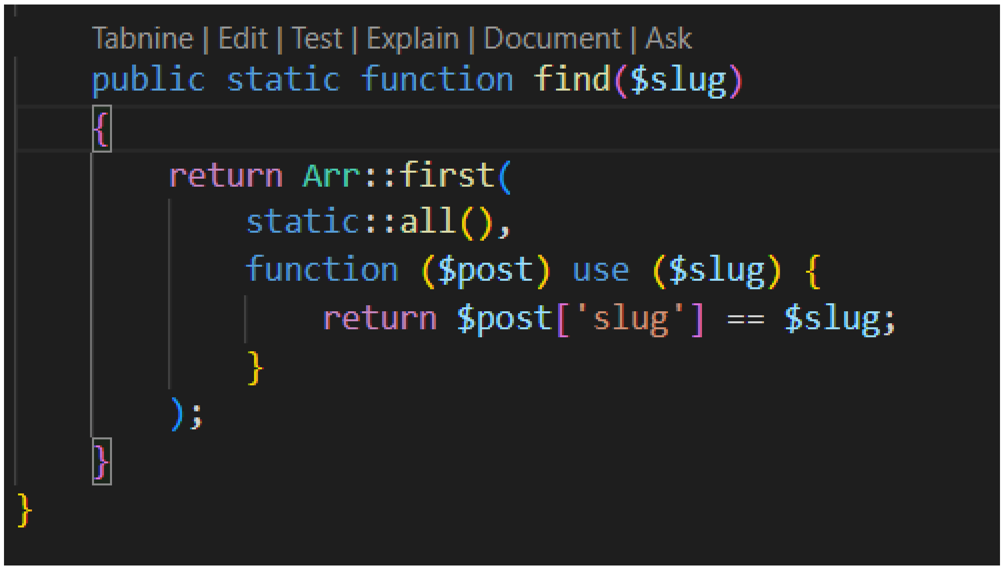


```
class Post

    Tabnine | Edit | Test | Fix | Explain | Document | Ask
    public static function find($slug)
    {
        return Arr::first(Post::all(), function ($post) use ($slug) {    Undefined type 'App\Models\Arr'.
            return $post['slug'] == $slug;
        }
    );
    }
}
```



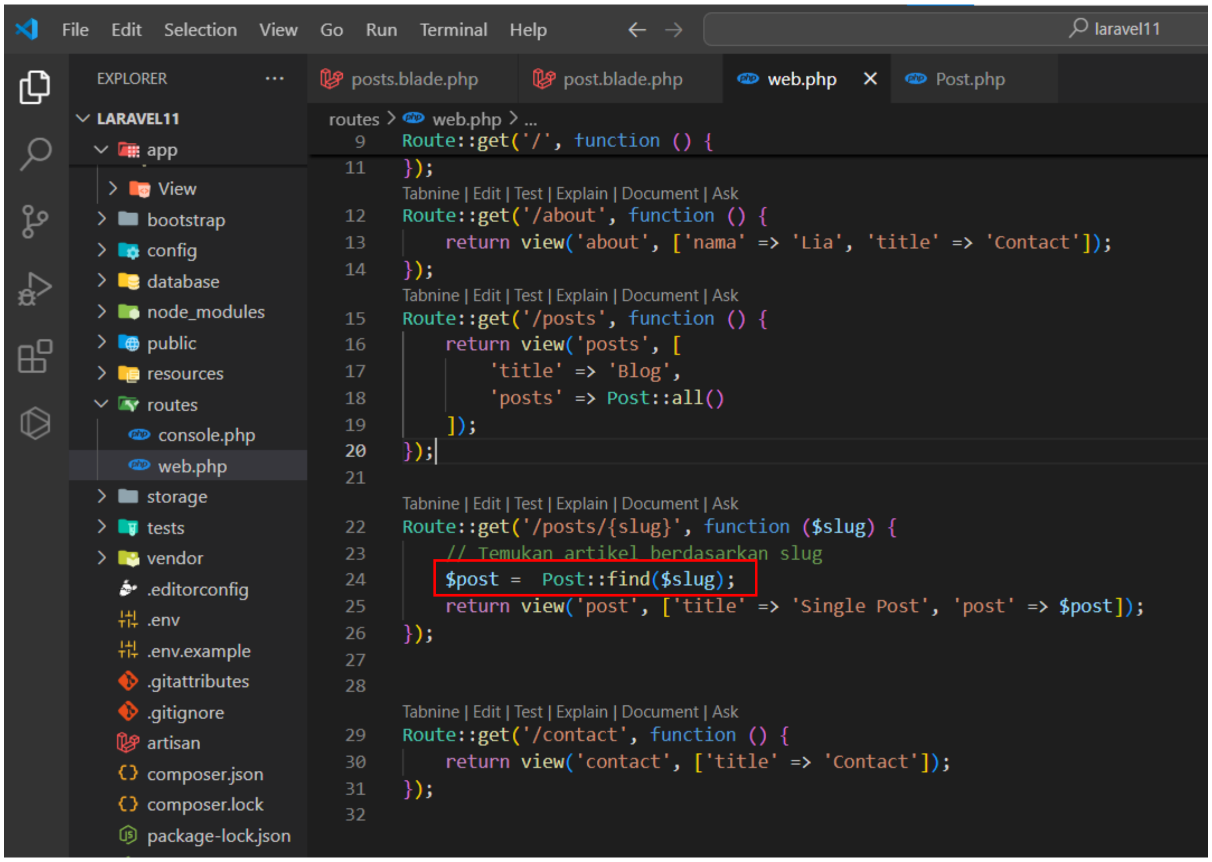
```
<?php

namespace App\Models;

use Illuminate\Support\Arr;
```

15. Kita buat funct baru atau funct static.



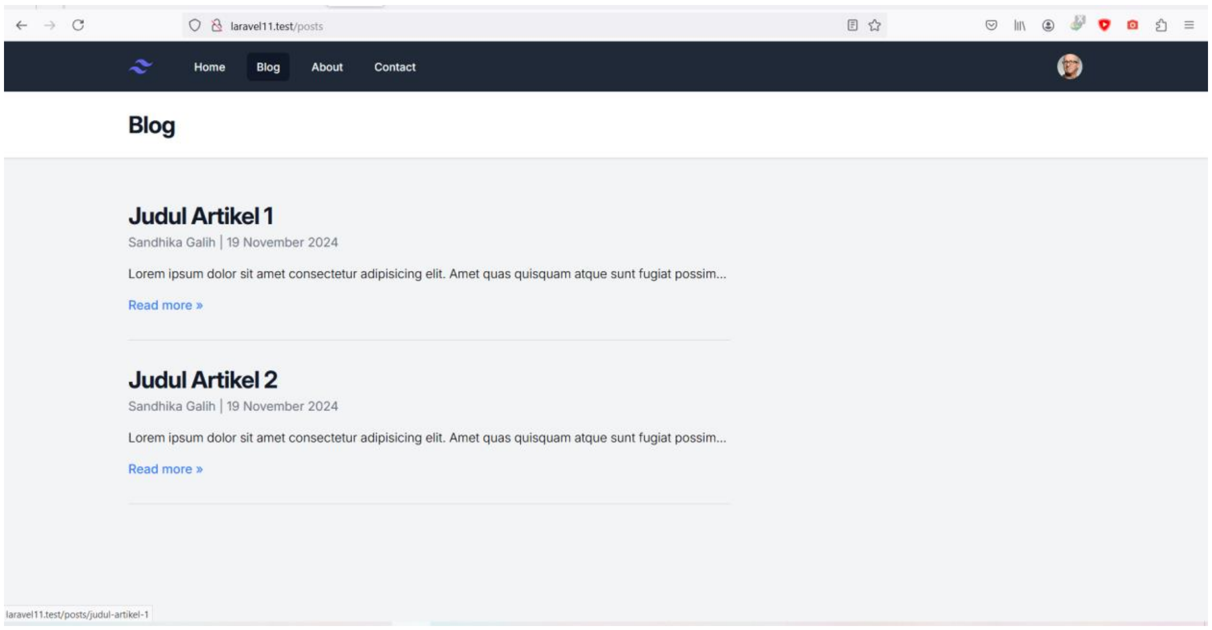
```
    Tabnine | Edit | Test | Explain | Document | Ask
    public static function find($slug)
    {
        return Arr::first(
            static::all(),
            function ($post) use ($slug) {
                return $post['slug'] == $slug;
            }
        );
    }
}
```

16. Kita ganti post nya menjadi static, karena ada di class yang sama. Kalau dalam sebuah method memanggil method yang lain di dalam class yang sama itu pake **this.**



```php
Route::get('/', function () {
});
Route::get('/about', function () {
    return view('about', ['nama' => 'Lia', 'title' => 'Contact']);
});
Route::get('/posts', function () {
    return view('posts', [
        'title' => 'Blog',
        'posts' => Post::all()
    ]);
});

Route::get('/posts/{slug}', function ($slug) {
    // Temukan artikel berdasarkan slug
    $post =  Post::find($slug);
    return view('post', ['title' => 'Single Post', 'post' => $post]);
});


Route::get('/contact', function () {
    return view('contact', ['title' => 'Contact']);
});
```

17. Jadi kita tinggal memanggil class Post method find.



laravel11.test/posts

Home Blog About Contact

Blog

Judul Artikel 1

Sandhika Galih | 19 November 2024

Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possim...

Read more »

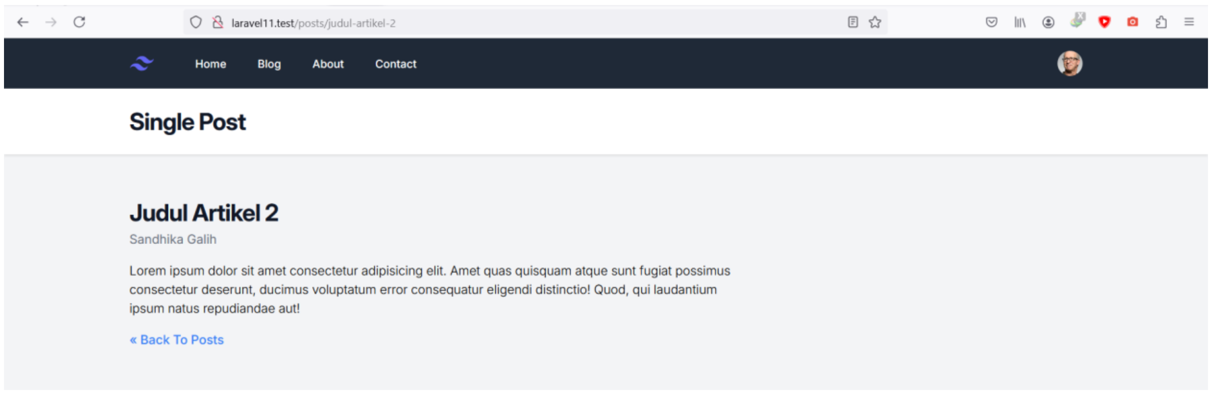
Judul Artikel 2

Sandhika Galih | 19 November 2024

Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possim...

Read more »

laravel11.test/posts/judul-artikel-1



Single Post

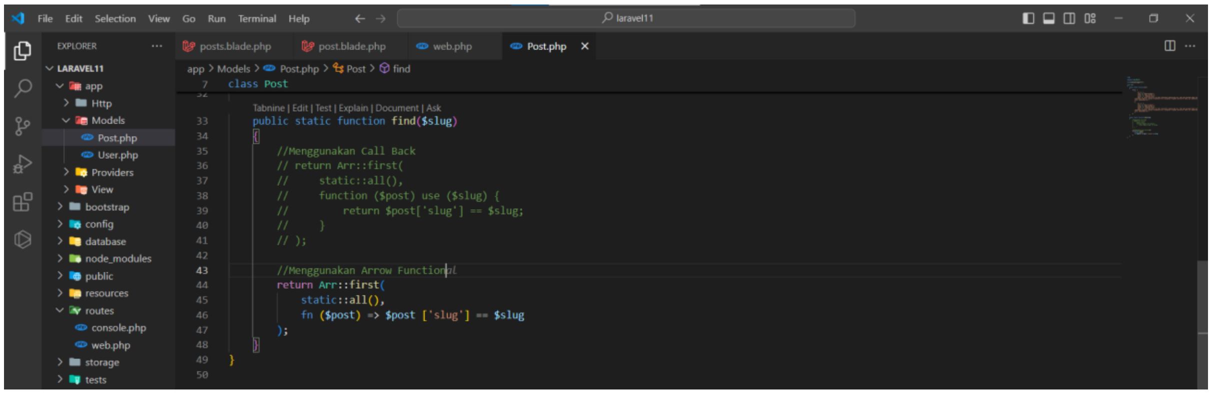
Judul Artikel 2

Sandhika Galih

Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possimus consectetur deserunt, ducimus voluptatum error consequatur eligendi distinctio! Quod, qui laudantium ipsum natus repudiandae aut!

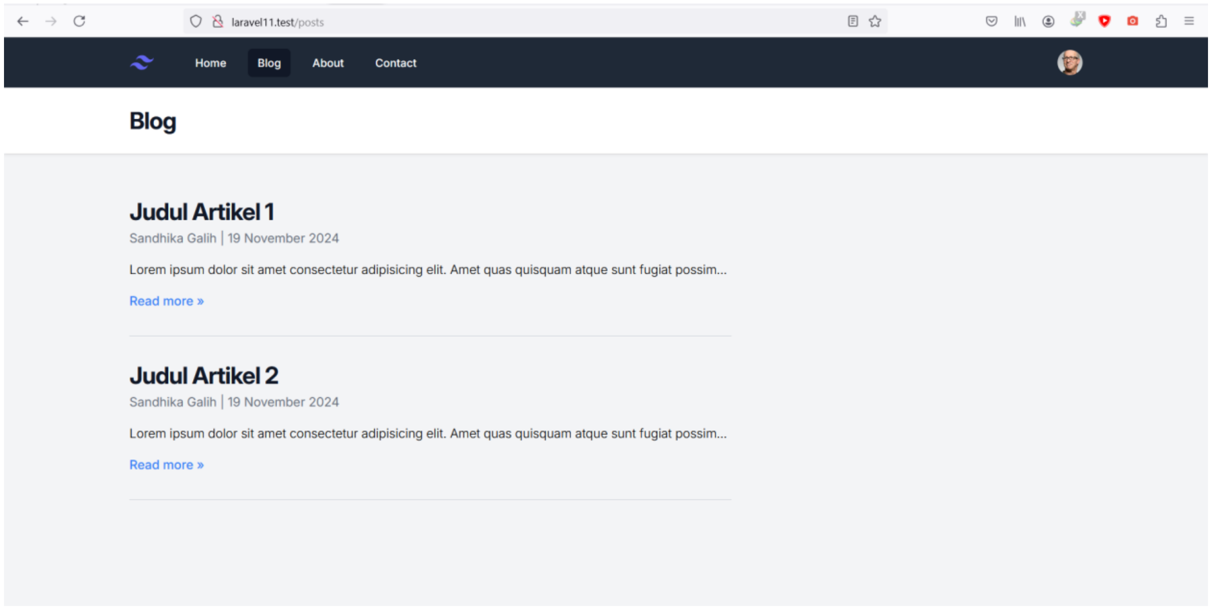
« Back To Posts

18. Hasil run nya.



```php
class Post

    public static function find($slug)
    {
        //Menggunakan Call Back
        // return Arr::first(
        //     static::all(),
        //     function ($post) use ($slug) {
        //         return $post['slug'] == $slug;
        //     }
        // );

        //Menggunakan Arrow Function
        return Arr::first(
            static::all(),
            fn ($post) => $post ['slug'] == $slug
        );
    }
}
```

19. Kita menggunakan teknik **arrow funct** jika tidak ingin menggunakan **use**.



Blog

Judul Artikel 1

Sandhika Galih | 19 November 2024

Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possim...

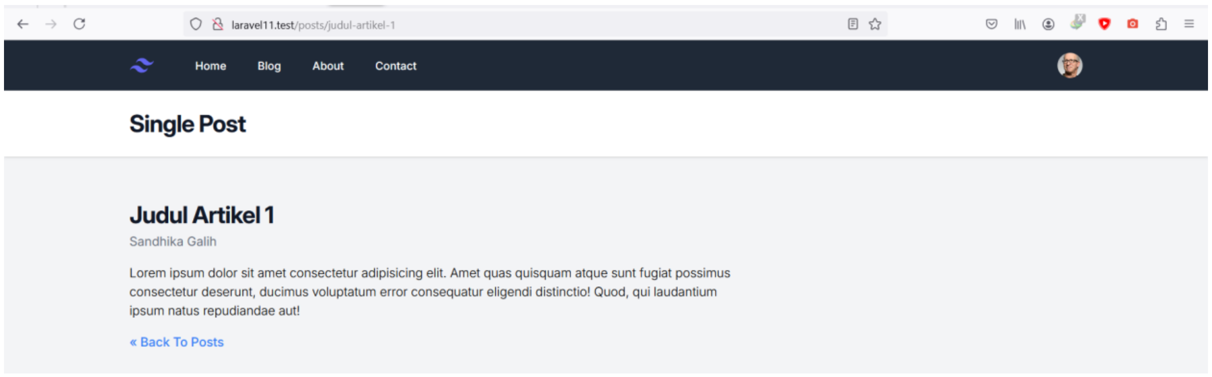
Read more »

Judul Artikel 2

Sandhika Galih | 19 November 2024

Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possim...

Read more »



Single Post

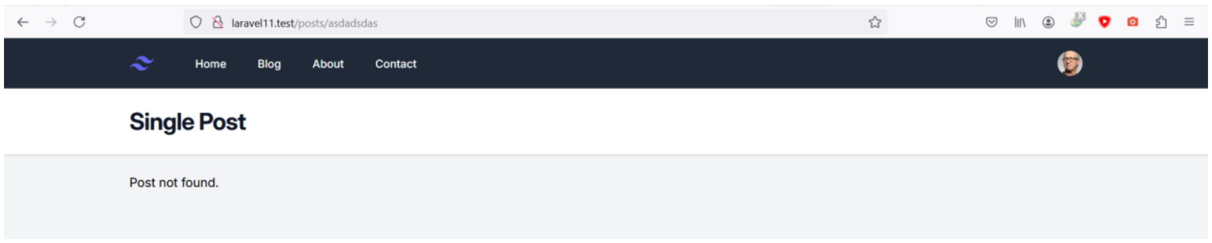
Judul Artikel 1

Sandhika Galih

Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possimus consectetur deserunt, ducimus voluptatum error consequatur eligendi distinctio! Quod, qui laudantium ipsum natus repudiandae aut!

« Back To Posts

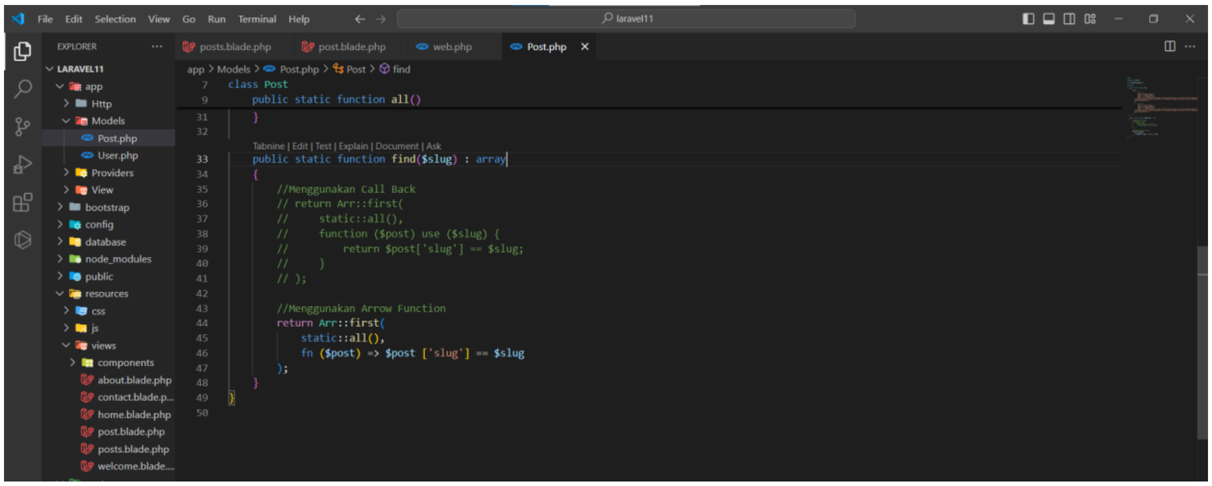
20. Hasil saat di run masih sama. Kenapa kita buat manual pakai class terus namanya find dan all ini supaya merepresentasikan ke depannya kita akan menggunakan fungsi laravel dengan nama method yang sama.



Single Post

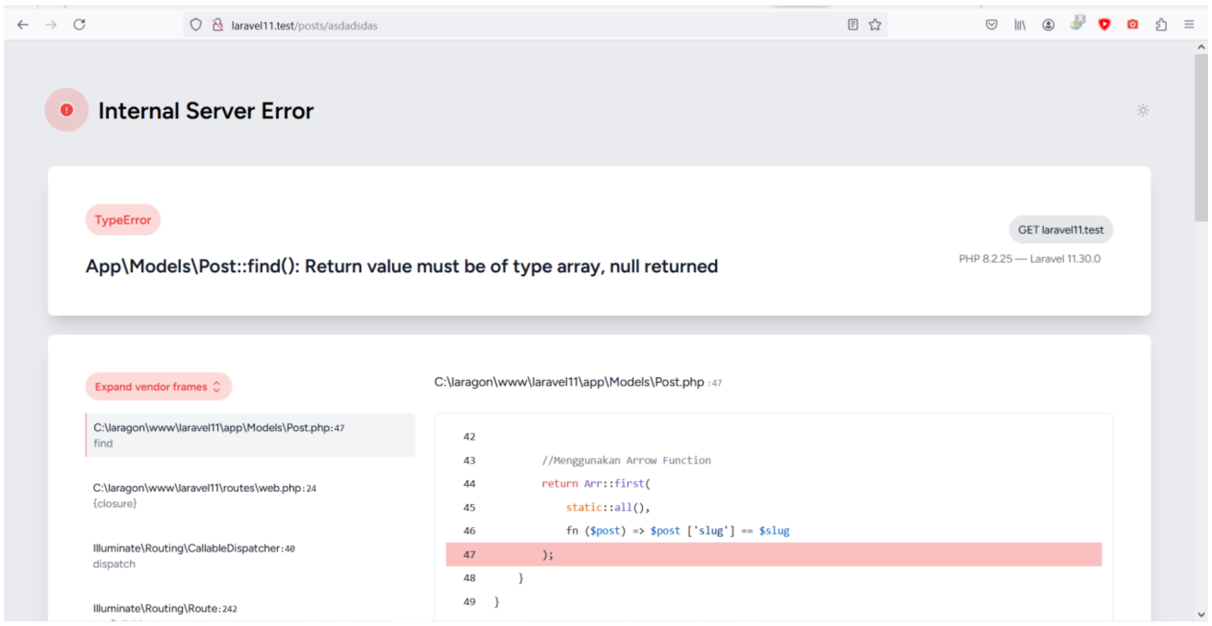
Post not found.

21. Gimana kalau misalnya postnya nggak ada, di url kita ketik asal maka akan terjadi error.



```php
class Post
    public static function all()
    }

    public static function find($slug) : array
    {
        //Menggunakan Call Back
        // return Arr::first(
        //     static::all(),
        //     function ($post) use ($slug) {
        //         return $post['slug'] == $slug;
        //     }
        // );

        //Menggunakan Arrow Function
        return Arr::first(
            static::all(),
            fn ($post) => $post ['slug'] == $slug
        );
    }
}
```

22. Kita kasih alamat 404 aja ketika posnya nggak ketemu. Kalau misalkan mau sedikit memperbaiki supaya lebih yakin dengan apa yang dikembalikan oleh method, Kita bisa ngasih return type berupa array.



laravel11.test/posts/asdadsdas

Internal Server Error

TypeError

App\Models\Post::find(): Return value must be of type array, null returned

GET laravel11.test

PHP 8.2.25 — Laravel 11.30.0

Expand vendor frames

C:\laragon\www\laravel11\app\Models\Post.php:47
find

C:\laragon\www\laravel11\routes\web.php:24
{closure}

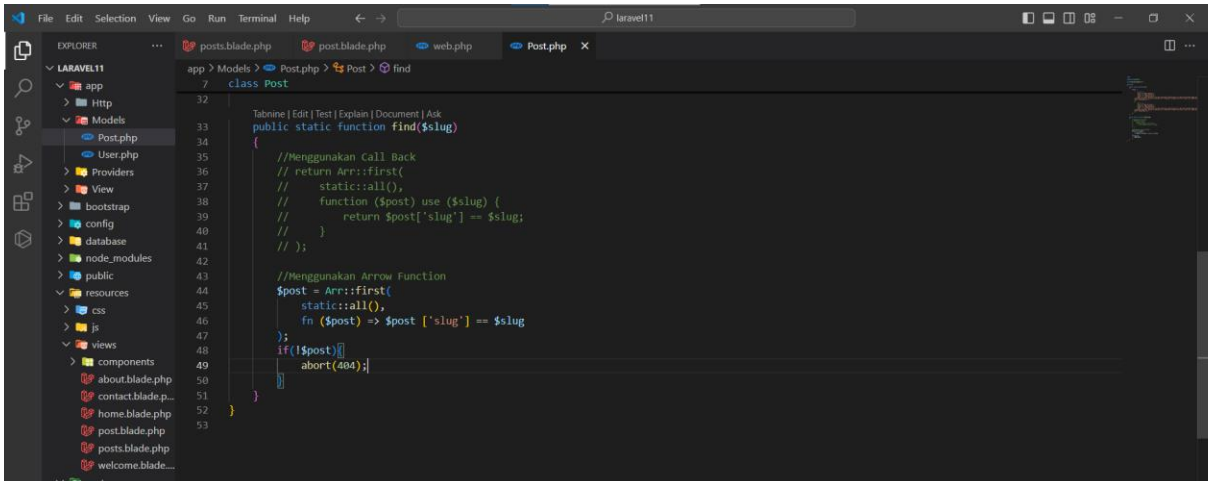
Illuminate\Routing\CallableDispatcher:40
dispatch

Illuminate\Routing\Route:242

C:\laragon\www\laravel11\app\Models\Post.php :47

```
42
43        //Menggunakan Arrow Function
44        return Arr::first(
45            static::all(),
46            fn ($post) => $post ['slug'] == $slug
47        );
48    }
49 }
```

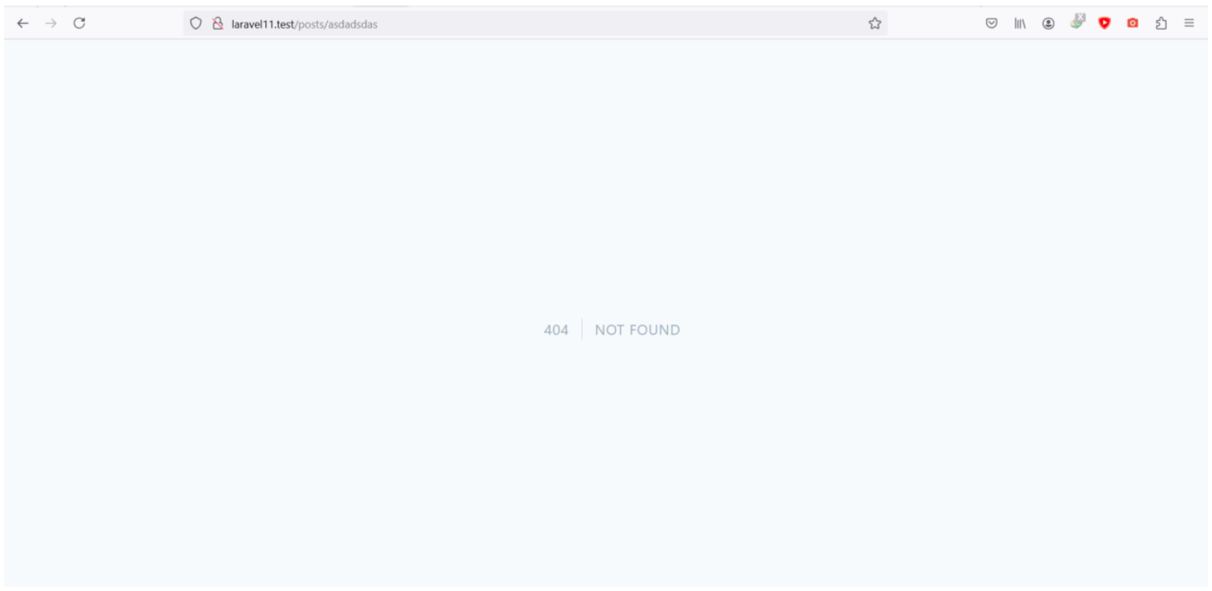
23. Langsung ketahuan salahnya di mana gitu di sini



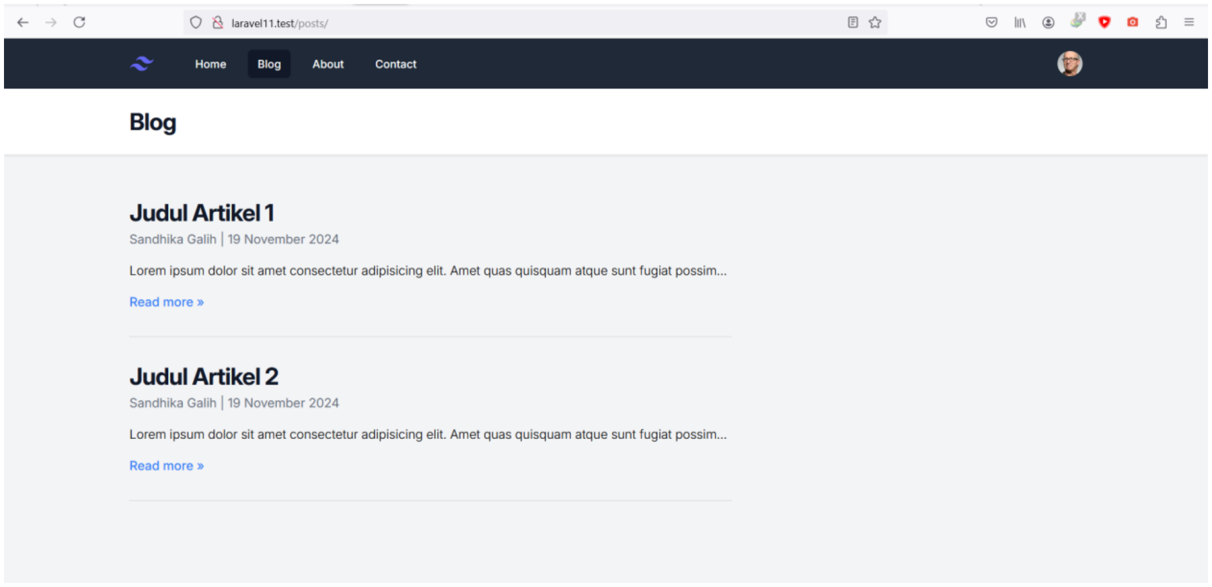
app > Models > Post.php > Post > find

```php
class Post

    public static function find($slug)
    {
        //Menggunakan Call Back
        // return Arr::first(
        //     static::all(),
        //     function ($post) use ($slug) {
        //         return $post['slug'] == $slug;
        //     }
        // );

        //Menggunakan Arrow Function
        $post = Arr::first(
            static::all(),
            fn ($post) => $post ['slug'] == $slug
        );
        if(!$post){
            abort(404);
        }
    }
}
```

24. Pertama kita akan Simpan dulu ke dalam sebuah variabel, jika posnya kosong bisa tulis kasih halaman 404.



404 | NOT FOUND



Blog

Judul Artikel 1

Sandhika Galih | 19 November 2024

Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possim...

Read more »

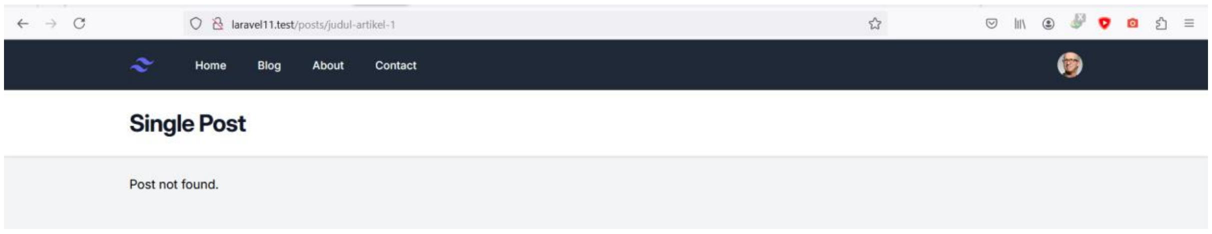
Judul Artikel 2

Sandhika Galih | 19 November 2024

Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possim...

Read more »

25. Jika url kosong maka halaman tetap berjalan.

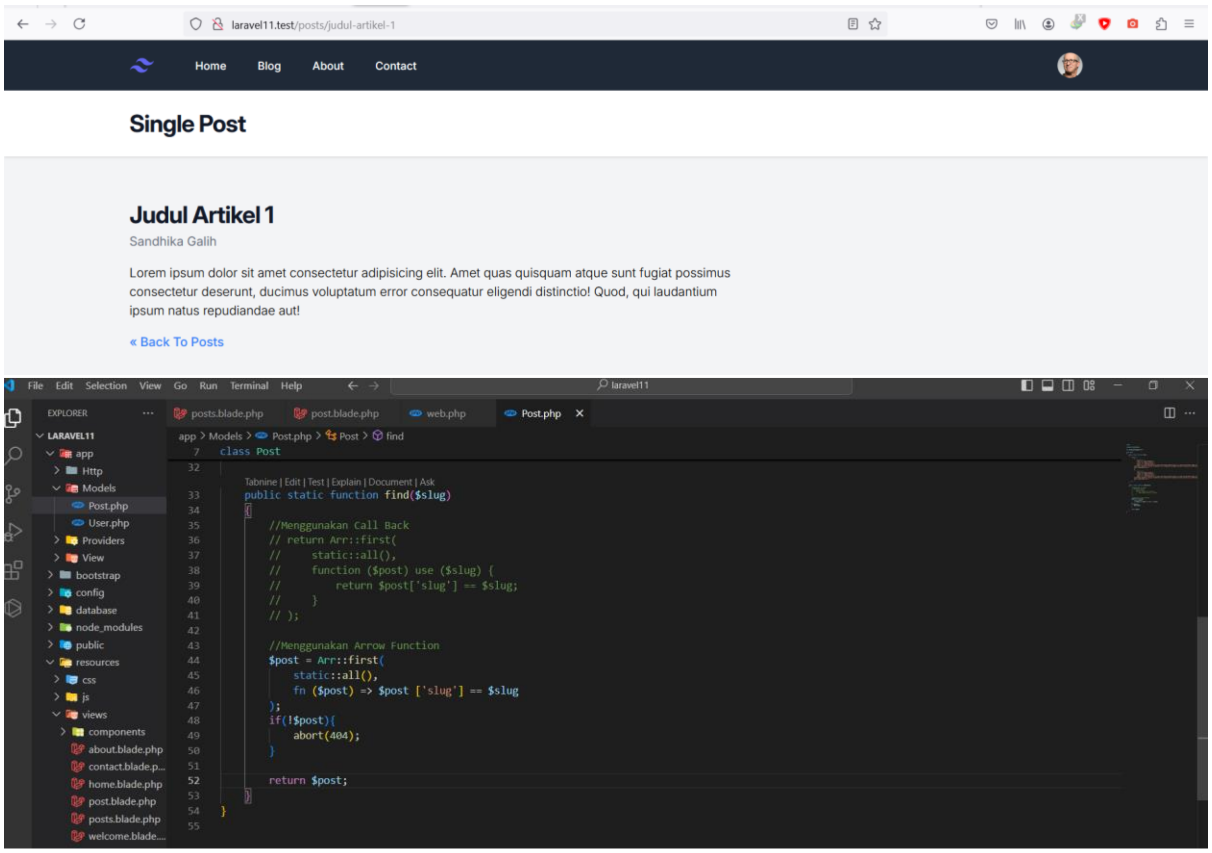


Single Post

Post not found.

26. Namun jika read more di klik halaman akan error karena belum dilakukan return.

27. Lakukan return $post



Single Post

Judul Artikel 1

Sandhika Galih

Lorem ipsum dolor sit amet consectetur adipisicing elit. Amet quas quisquam atque sunt fugiat possimus consectetur deserunt, ducimus voluptatum error consequatur eligendi distinctio! Quod, qui laudantium ipsum natus repudiandae aut!

« Back To Posts

```php
class Post

    public static function find($slug)
    {
        //Menggunakan Call Back
        // return Arr::first(
        //     static::all(),
        //     function ($post) use ($slug) {
        //         return $post['slug'] == $slug;
        //     }
        // );

        //Menggunakan Arrow Function
        $post = Arr::first(
            static::all(),
            fn ($post) => $post ['slug'] == $slug
        );
        if(!$post){
            abort(404);
        }

        return $post;
    }
}
```

28. Hasil setelah dilakukan return $post