

NIM: 2602110986

Nama: Anthonio Obert Lais

Kelas: LJ01

## Dokumentasi Quiz Before UTS

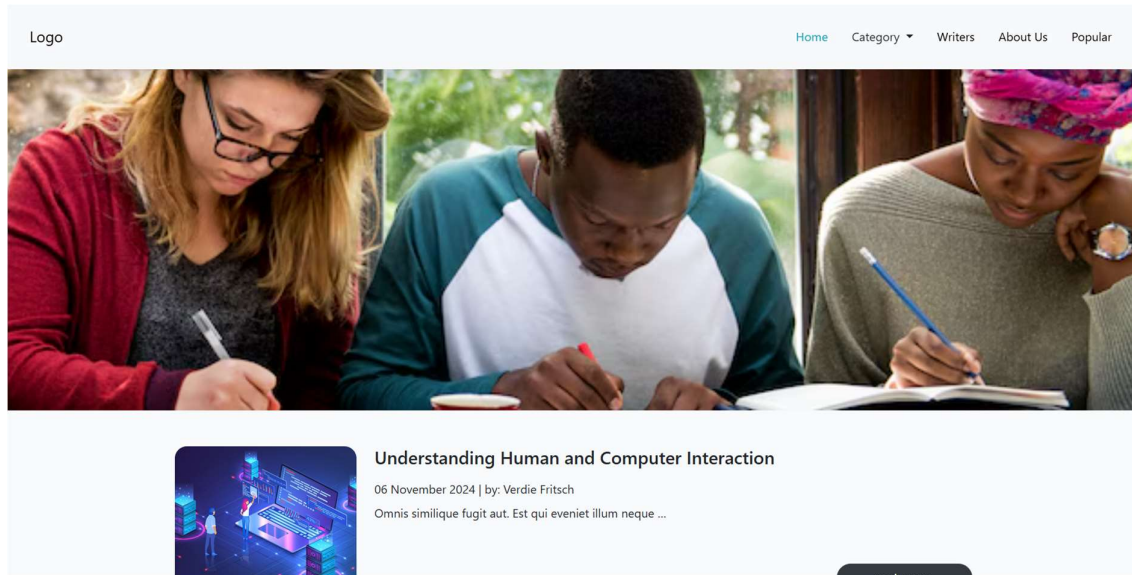


Figure 1. Home Page

Pembuatan semua page dilakukan dengan menggunakan layout seperti pada code berikut:

```
main-layout.blade.php X article-card.blade.php article.blade.php category.blade.php Article
resources > views > layouts > main-layout.blade.php > html > body.bg-body > x-navbar
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>EduFun</title>
7   @vite(['resources/sass/app.scss',
8     'resources/js/app.js', 'public/bootstrap/bootstrap-4.5.0-dist/css/bootstrap.css'])
9 </head>
10 <body class="bg-body">
11
12   <x-navbar/>
13
14   @yield('content')
15
16   @include('components.footer')
17
18 </body>
19 </html>
20
```

Sehingga navbar dan footer selalu ada pada tiap page. Untuk navbar dibuat menggunakan component dari Laravel, dikarenakan melakukan fetching untuk memunculkan setiap category dari article yang ada.



Figure 2. Article Card

Dibuatkannya Article Card yang berguna untuk menghilangkan redundansi ketika suatu page dibutuhkan untuk menampilkan article yang ada.

Article terdiri dari data berikut:

```
Schema::create('articles', function (Blueprint $table) {  
    $table->id();  
    $table->timestamps();  
    $table->string('title');  
    $table->text('body');  
    $table->string('image_path');  
    $table->foreignId('article_category_id')->constrained('article_categories')->onDelete('cascade');  
    $table->foreignId('writer_id')->constrained('writers')->onDelete('cascade');  
});
```

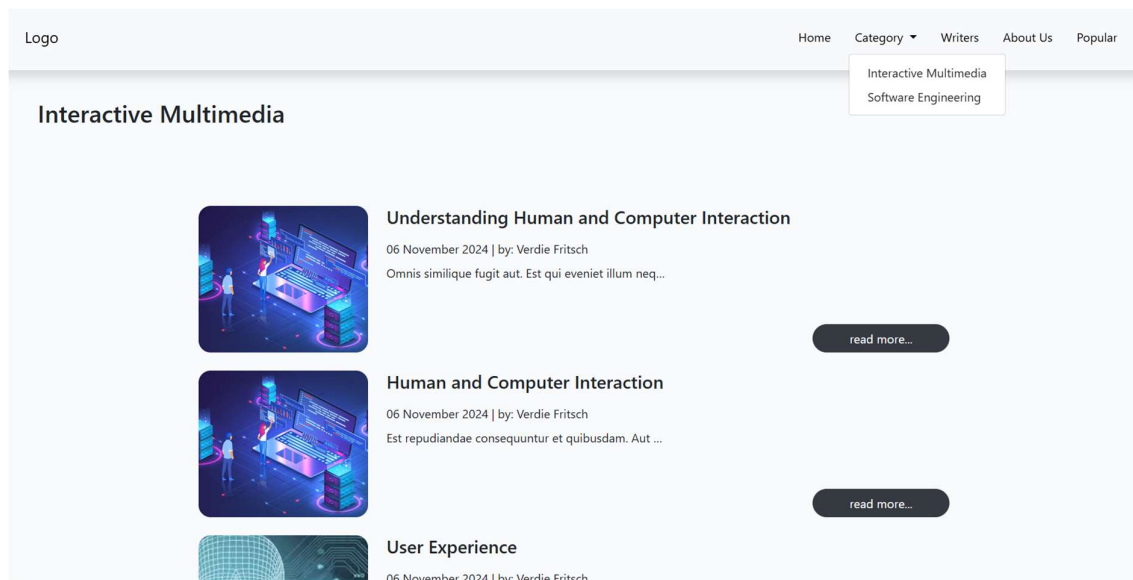


Figure 3. Category Page

Category dibuat menjadi suatu table agar dapat melakukan filtering melalui id dan untuk menerapkan normalisasi. Article telah dilakukan seeding dengan memastikan kesamaan job dari writer dan category article yang dibuat.

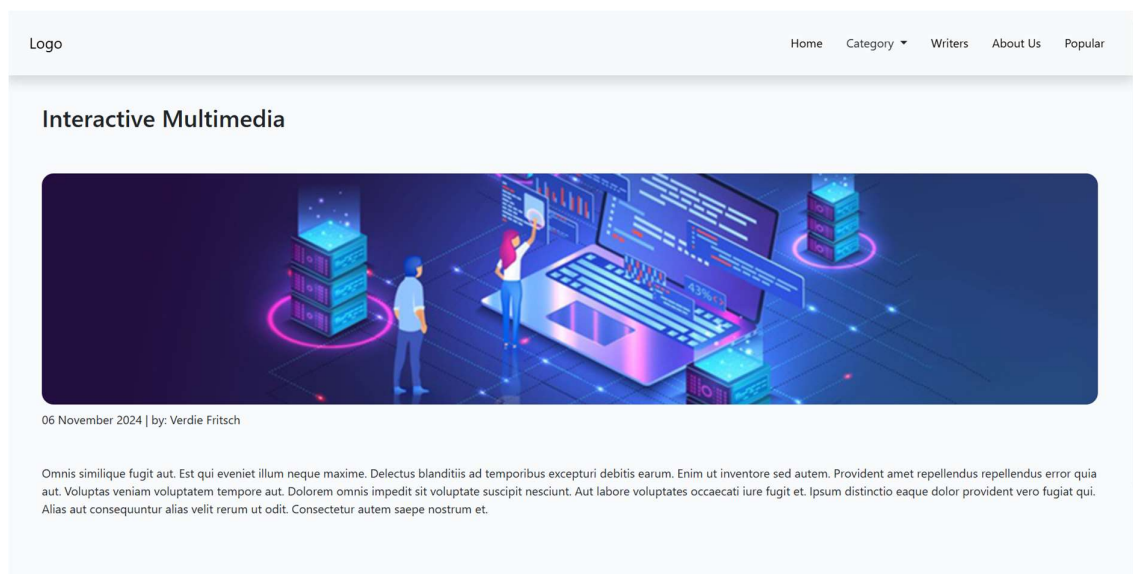


Figure 4. Article Detail

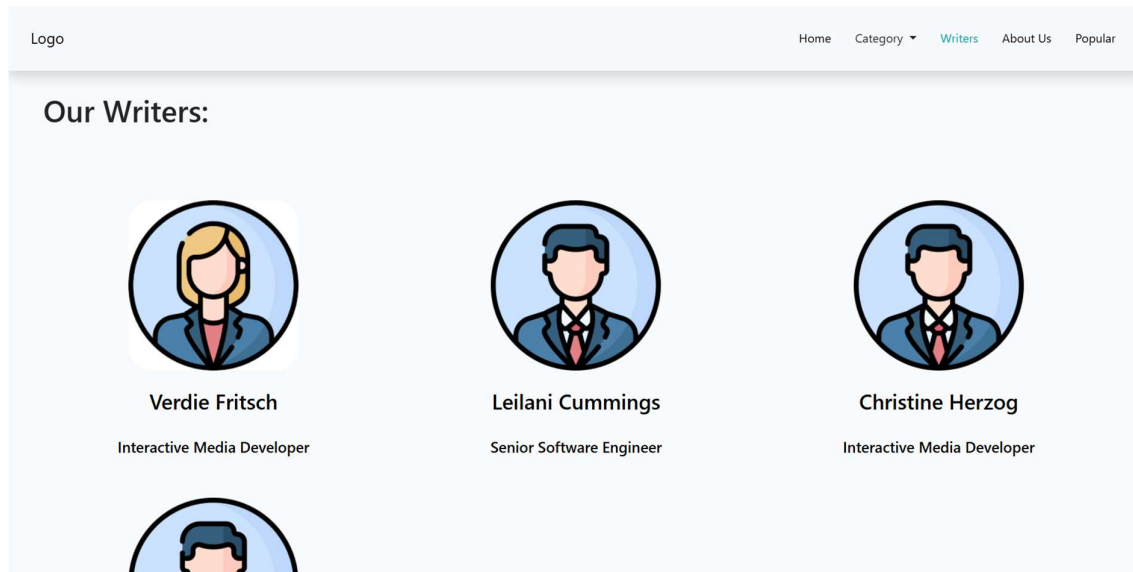


Figure 5. Writers Page

Halaman Writers yang ketika diklik akan memunculkan article yang dibuat oleh masing-masing writer. Data writer diseed dengan faker, berikut untuk detail code.

```
class WriterFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    protected $model = Writer::class;

    public function definition(): array
    {
        return [
            //
            'name' => $this->faker->name(),
            'job' => $this->faker->randomElement(['Multimedia Designer', 'Interactive Media Developer', 'Senior Software Engineer', 'Full Stack Developer']),
            'image_path' => $this->faker->randomElement(['/assets/seed/writers/1.png', '/assets/seed/writers/2.png', '/assets/seed/writers/3.png', '/assets/seed/writers/4.png', '/assets/seed/writers/5.png', '/assets/seed/writers/6.png', '/assets/seed/writers/7.png', '/assets/seed/writers/8.png', '/assets/seed/writers/9.png', '/assets/seed/writers/10.png']),
        ];
    }
}
```

```
class WriterSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        //

        # Sequence to make sure each category has the fair amount of writers
        Writer::factory()->count(4)
            ->sequence(
                ['job' => fake()->randomElement(['Multimedia Designer', 'Interactive Media Developer'])],
                ['job' => fake()->randomElement(['Senior Software Engineer', 'Full Stack Developer'])],
            )
            ->create();
    }
}
```

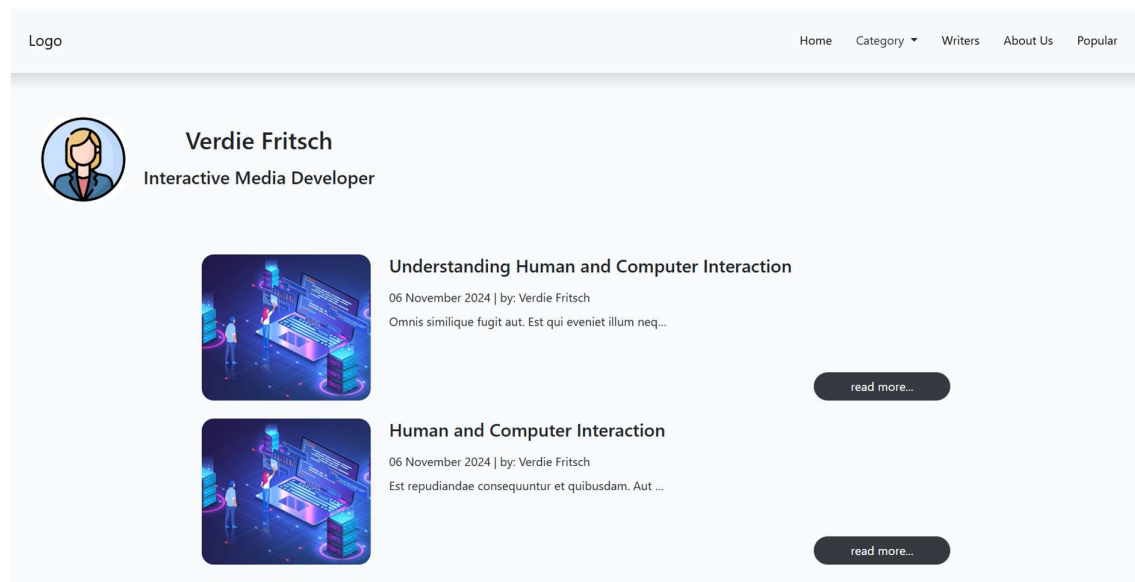


Figure 6. Writer Articles Page

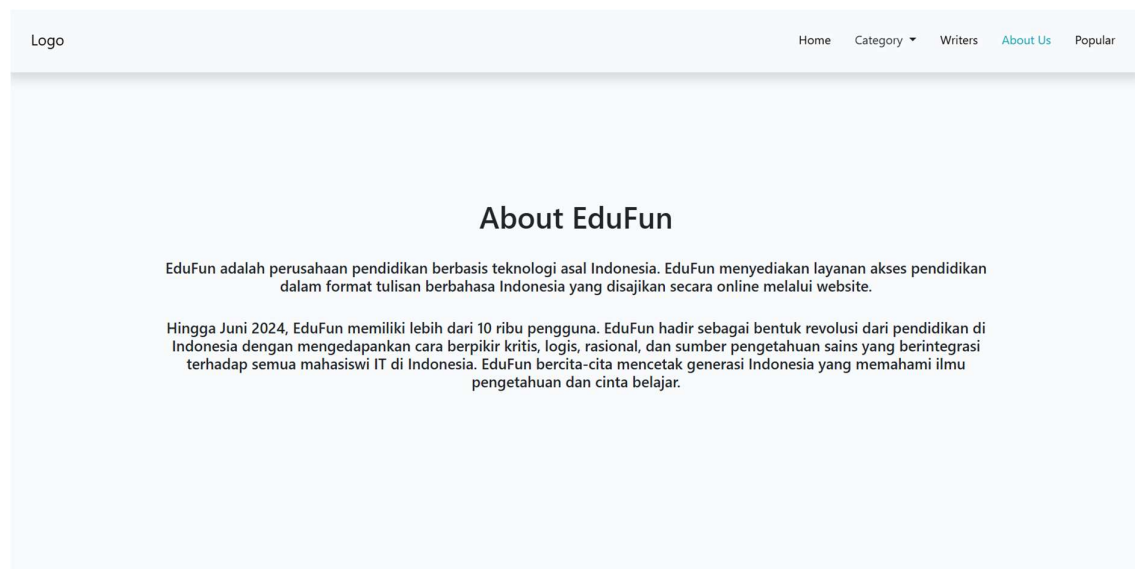


Figure 7. About Us Page

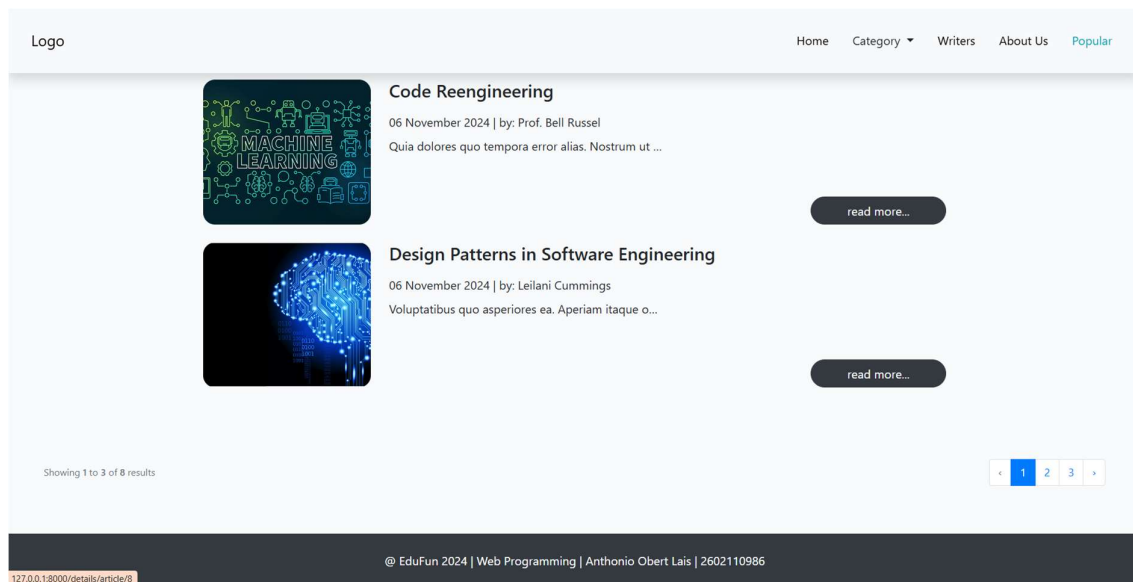


Figure 8. Popular Page

Untuk Popular page telah dibuat dengan pagination dari bootstrap dan berisi 3 article per page-nya. Article ditampilkan semua dengan disort berdasarkan title nya (ascending).

```
public function show_popular()
{
    $articles = Article::orderBy('title', 'asc')->paginate(3);

    return view('pages.popular', compact('articles'));
}
```

Berikut detail routing:

```
Route::get('/', HomeController::class)->name('home');
Route::get('/writers', [WriterController::class, 'index']->name('writers');
Route::get('/about-us', AboutUsPageController::class)->name('about-us');

Route::prefix('details')->group(function () {
    Route::get('/article/{article}', [ArticleController::class, 'show']->name('details.article');
    Route::get('/writer-articles/{id}', [WriterController::class, 'show']->name('details.writer-articles');
});

Route::controller(ArticleController::class)->group(function () {
    Route::get('/category/{category_id}', 'show_by_category')->name('category');
    Route::get('/popular', 'show_popular')->name('popular');
});
```