

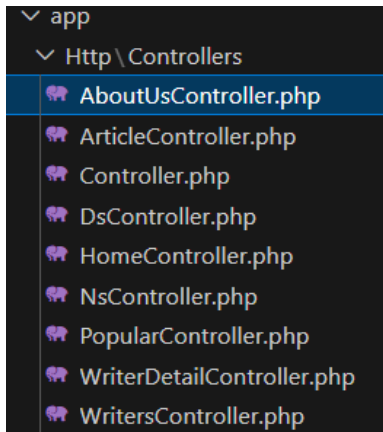
- Routing :

```
1 <?php
2
3 use App\Http\Controllers\AboutUsController;
4 use App\Http\Controllers\ArticleController;
5 use App\Http\Controllers\DsController;
6 use App\Http\Controllers\HomeController;
7 use App\Http\Controllers\NsController;
8 use App\Http\Controllers\PopularController;
9 use App\Http\Controllers\WriterDetailController;
10 use App\Http\Controllers\WritersController;
11 use Illuminate\Support\Facades\Route;
12
13 Route::get(uri: '/', action: function (): Factory|View {
14     return view(view: 'welcome');
15 });
16
17 Route::prefix(prefix: '/user')->group(callback: function(): void{
18     Route::get(uri: '/home', action: [HomeController::class, 'index'])->name(name: 'user.home');
19     Route::get(uri: '/datascience', action: [DsController::class, 'index'])->name(name: 'user.datascience');
20     Route::get(uri: '/networksecurity', action: [NsController::class, 'index'])->name(name: 'user.networksecurity');
21     Route::get(uri: '/writers', action: [WritersController::class, 'index'])->name(name: 'user.writers');
22     Route::get(uri: '/aboutus', action: [AboutUsController::class, 'index'])->name(name: 'user.aboutus');
23     Route::get(uri: '/popular', action: [PopularController::class, 'index'])->name(name: 'user.popular');
24     Route::get(uri: 'article/{id}', action: [ArticleController::class, 'index'])->name(name: 'user.article');
25     Route::get(uri: 'writerdetail/{id}', action: [WriterDetailController::class, 'index'])->name(name: 'user.writerdetail');
26 });
```

Terdapat 8 bagian routing yang di-grouping dalam route ‘/user’ :

- Route pertama (‘/user/home’) akan menampilkan tampilan home yang dibantu oleh ‘HomeController’ dengan nama ‘user.home’.
- Route ke-dua (‘/user/datascience’) akan menampilkan tampilan category data science yang dibantu oleh ‘DsController’ dengan nama ‘user.datascience’.
- Route ke-tiga (‘/user/networksecurity’) akan menampilkan tampilan category network security yang dibantu oleh ‘NsController’ dengan nama ‘user.networksecurity’.
- Route ke-empat (‘/user/writers’) akan menampilkan tampilan writers yang dibantu oleh ‘WritersController’ dengan nama ‘user.writers’.
- Route ke-lima (‘/user/aboutus’) akan menampilkan tampilan about us yang dibantu oleh ‘AboutUsController’ dengan nama ‘user.aboutus’.
- Route ke-enam (‘/user/popular’) akan menampilkan tampilan popular (artikel yang sering user kunjungi) yang dibantu oleh ‘PopularController’ dengan nama ‘user.popular’.
- Route ke-tujuh (‘/user/article/{id}’) akan menampilkan tampilan article yang dipilih oleh user yang dibantu oleh ‘ArticleController’ dengan nama ‘user.article’. Route ‘/{id}’ digunakan untuk melakukan passing variable ‘id’.
- Route terakhir (‘/user/writerdetail/{id}’) akan menampilkan tampilan artikel apa saja yang dibuat oleh writer dengan bantuan controller ‘WriterDetailController’ dan dinamai ‘user.writerdetail’. Route ‘/{id}’ digunakan untuk melakukan passing variable ‘id’.

- **Controller :**



Berikut penjelasan setiap controller yang saya gunakan :

- HomeController :

```
<?php

namespace App\Http\Controllers;

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

2 references | 0 implementations
class HomeController extends Controller
{
    1 reference | 0 overrides
    public function index(): Factory|View{
        $writer = Writer::all();
        $article = Article::latest()->limit(value: 3)->get();
        return view(view: 'page.home', data: compact(var_name: 'article', var_names: 'writer'));
    }
}
```

Pada HomeController, terdapat function index yang digunakan untuk me-return tampilan view and blade yang diarahkan ke page home. Selain itu, model writer dan article juga dipanggil agar database bisa diakses pada view and blade. Untuk database writer, data yang diambil adalah semua data sehingga menggunakan function all(). Sementara itu, data article yang diambil adalah data terbaru yang di-limit sebanyak 3 data.

- DsController :

```
class DsController extends Controller
{
    1 reference | 0 overrides
    public function index(): Factory|View{
        $writer = Writer::all();
        $article = Article::where(column: 'category', operator: 'Data Science')->get();
        return view(view: 'page.ds', data: compact(var_name: 'article', var_names: 'writer'));
    }
}
```

Sama seperti HomeController, DsController juga menggunakan function index untuk me-return tampilan view, namun data article yang digunakan memiliki constraint dimana data yang diambil adalah data article dengan category 'Data Science' saja.

- NsController :

```
class NsController extends Controller
{
  1 reference | 0 overrides
  public function index(): Factory|View{
    $writer = Writer::all();
    $article = Article::where(column: 'category', operator: 'Network Security')->get();
    return view(view: 'page.ns', data: compact(var_name: 'article', var_names: 'writer'));
  }
}
```

Sama seperti DsController, tetapi data article yang diambil adalah data article dengan category 'Network Security'.

- WritersController :

```
class WritersController extends Controller
{
  1 reference | 0 overrides
  public function index(): Factory|View{
    $writer = Writer::all();
    return view(view: 'page.writers', data: compact(var_name: 'writer'));
  }
}
```

Pada WritersController, function index me-return tampilan page writers dengan membawa database writer melalui model.

- AboutUsController :

```
class AboutUsController extends Controller
{
  1 reference | 0 overrides
  public function index(): Factory|View{
    return view(view: 'page.aboutus');
  }
}
```

AboutUsController hanya me-return tampilan page about us melalui function index.

- PopularController :

```
class PopularController extends Controller
{
  1 reference | 0 overrides
  public function index(): Factory|View{
    $article = Article::with(relations: 'writer')->orderBy(column: 'click_count', direction: 'desc')->paginate(perPage: 3);
    $writer = Writer::all();
    return view(view: 'page.popular', data: compact(var_name: 'article', var_names: 'writer'));
  }
}
```

Pada PopularController, model Article dipanggil untuk me-return database article yang diorder berdasarkan click_count (kolom pada database article untuk melihat seberapa sering user melakukan 'click' terhadap artikel yang mereka kunjungi) dan ditampilkan secara descending. Kemudian, function paginate() juga digunakan agar setiap page pada halaman popular hanya menampilkan 3 data article saja.

- ArticleController :

```
class ArticleController extends Controller
{
    1 reference | 0 overrides
    public function index($id): mixed|View{
        $article = Article::all();
        $articles = Article::findOrFail(id: $id);
        $articles->click_count += 1;
        $articles->save();
        $writer = Writer::all();
        return view('page.articledetail', data: compact('article', 'writer'))->with('id', value:$id);
    }
}
```

Pada ArticleController, dilakukan operasi increment dimana jika user mengunjungi article yang mereka pilih, maka secara otomatis kolom click_count pada database article akan bertambah. Kemudian, passing variable '\$id' juga digunakan untuk merecord article apa yang dipilih oleh user berdasarkan id article tersebut. Sehingga click_count yang ditambah akan menyesuaikan article yang dipilih.

- WriterDetailController :

```
class WriterDetailController extends Controller
{
    1 reference | 0 overrides
    public function index($id): mixed|View{
        $article = Article::all();
        $writer = Writer::all();
        return view('page.writerdetail', data: compact('article', 'writer'))->with('id', value:$id);
    }
}
```

ArticleController juga melakukan passing variable '\$id' agar user bisa melihat article sesuai dengan writer yang mereka pilih.

- Model :

Terdapat 2 model yaitu Article dan Writer :

- Article :

```
<?php

namespace App\Models;

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

21 references | 0 implementations
class Article extends Model
{
    use HasFactory;
    0 references
    protected $table = 'article';
    0 references
    protected $guarded = [];

    0 references | 0 overrides
    public function writer(): BelongsTo{
        return $this->belongsTo(related: Writer::class);
    }
}
```

Pada model Article, terdapat function writer() untuk menyatakan hubungan atau relasi terhadap model database Writer. Dalam hal ini, hubungan antara database Writer dan Article adalah one to many. Ini menyiratkan bahwa setiap artikel dikaitkan dengan satu penulis.

- Writer :

```
<?php

namespace App\Models;

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

20 references | 0 implementations
class Writer extends Model
{
    use HasFactory;
    0 references
    protected $table = 'writer';
    0 references
    protected $guarded = [];

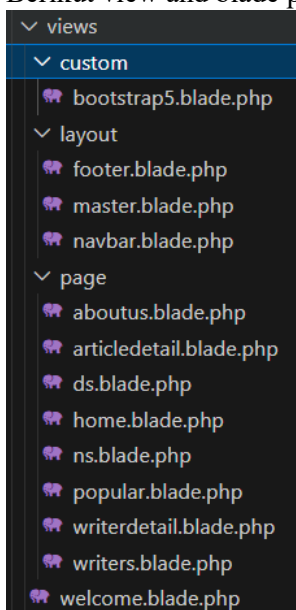
    0 references | 0 overrides
    public function articles(): HasMany{
        return $this->hasMany(related: Article::class);
    }
}
```

Model Writer mendefinisikan sebuah hubungan antara article dan writer melalui metode writer, yang menggunakan relasi hasMany untuk menunjukkan bahwa setiap writer terkait bisa membuat banyak article.

Properti \$table digunakan untuk menentukan nama tabel yang digunakan, sementara \$guarded = [] menunjukkan bahwa semua atribut pada model ini dapat di-mass assign tanpa adanya pembatasan. Dengan demikian, model ini memungkinkan akses dan manipulasi data article dan writer dengan mudah, serta menjaga fleksibilitas dalam pengelolaan data yang terkait.

- View and Blade :

Berikut view and blade pada website ini :



- Custom blade :

```
<link rel="stylesheet" href="{{ asset(path: 'bootstrap/css/bootstrap.min.css')}}">
```

Berisikan code link dimana letak asset bootstrap disimpan, sehingga penggunaan bootstrap bisa lebih mudah untuk diakses.

- Layout Master :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0 maximum-scale=1, user-scalable=no">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>{{config(key: 'app.name')}}</title>
  @include(view: 'custom.bootstrap5')
</head>
<body>
  <div class="">
    @include(view: 'layout.navbar')
    @yield(section: 'content')
    @include(view: 'layout.footer')
  </div>

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965Dz00rT7abK41JStQ" ></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.6/dist/umd/popper.min.js" integrity="sha384-wHAiF" ></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.2.1/dist/js/bootstrap.min.js" integrity="sha384-B0Ug" ></script>
</body>
</html>
```

Master Blade adalah file template yang digunakan sebagai template dasar untuk mengakses halaman - halaman lainnya di dalam website. Blade ini berfungsi untuk mendefinisikan struktur umum halaman web, seperti header, footer, dll. Pada code diatas, template yang digunakan adalah navbar, content, dan footer.

- Layout Navbar :

```
<nav class="navbar navbar-expand-sm navbar-dark bg-white shadow sticky-top" style="height: 6rem;">
  <a href="{{ route(name: 'user.home') }}" class="navbar-brand mx-5">
    
  </a>
  <ul class="navbar-nav ms-auto gap-5 fw-bold mx-5 align-items-center">
    <li class="nav-item">
      <a class="nav-link text-dark" href="{{ route(name: 'user.home') }}">Home</a>
    </li>
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle text-dark" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
        Category
      </a>
      <div class="dropdown-menu">
        <a class="dropdown-item text-dark fw-bold" href="{{ route(name: 'user.datascience') }}">Data Science</a>
        <a class="dropdown-item text-dark fw-bold" href="{{ route(name: 'user.networksecurity') }}">Network Security</a>
      </div>
    </li>
    <li class="nav-item">
      <a class="nav-link text-dark" href="{{ route(name: 'user.writers') }}">Writers</a>
    </li>
    <li class="nav-item">
      <a class="nav-link text-dark" href="{{ route(name: 'user.aboutus') }}">About Us</a>
    </li>
    <li class="nav-item">
      <a class="nav-link text-dark" href="{{ route(name: 'user.popular') }}">Popular</a>
    </li>
  </ul>
</nav>
```

Layout Navbar digunakan untuk membuat tampilan navbar pada website.

- Layout Footer :

```
<footer class="bg-body-tertiary text-center text-lg-start">
  <div class="text-center p-5 text-white" style="background-color: black;">
    | © EduFun2024 | Web Programming | Vincent Nicholas Hie | 2602063365
  </div>
</footer>
```

Layout Footer digunakan untuk membuat tampilan footer pada website.

- Page Home :

```
@extends(view: 'layout.master')

@section(section: 'content')

<div class="container-fluid mt-4 mb-4">
  @foreach ($article as $art)
    <div class="card border-0 d-flex flex-row p-4">
      <div class="flex-shrink-0" style="width: 30rem; height: 15rem;">
        
      </div>
      <div class="ms-3">
        <h2 class="fw-bold mb-1 mt-2">{{ $art->title }}</h2>
        @foreach ($writer as $writers)
          @if ($writers->id == $art->writer_id)
            <p class="mb-2 small">{{ date(format: 'd F Y', timestamp: strtotime(datetime: $writers->created_at)) }}</p>
          @endif
        @endforeach
        <p class="mb-3">{{ Str::words(value: $art->content, words: 70) }}</p>
        <div class="mt-auto text-end">
          <a href="/user/article/{{ $art->id }}" class="btn btn-dark rounded-pill px-5 py-2 text-white">Read More</a>
        </div>
      </div>
    </div>
  @endforeach
</div>
@endsection
```

Page Home adalah tampilan utama pada website.

- Page About Us :

```
@extends(view: 'layout.master')

@section(section: 'content')
<div class="container my-5">
  <div class="text-center">
    <h1 class="fw-bold">About EduFun</h1>
  </div>
  <div class="row justify-content-center">
    <div class="col-md-8">
      <h3 class="text-center mt-4">EduFun adalah perusahaan</h3>
      <h3 class="text-center mt-4">Hingga Juni 2024, EduFun</h3>
    </div>
  </div>
</div>
@endsection
```

Page Home adalah tampilan About Us ketika user meng-click 'About Us' pada navbar.

- Page Article Detail :

```
@extends(view: 'layout.master')

@section(section: 'content')
@foreach ($article as $art)
    @if ($art->id == $id)
        <div class="m-4 mb-0">
            <h1 class="fw-bold">{{ $art->category }}</h1>
        </div>
        <div class="container-fluid mb-4">
            <div class="card border-0 shadow-sm p-4">
                <div class="mb-1">
                    
                </div>
                @foreach ($writer as $writers)
                    @if ($writers->id == $art->writer_id)
                        <p class="mb-4">{{ date(format: 'd F Y', time: $writers->created_at) }}</p>
                    @endif
                @endforeach
                <p class="">
                    {{ $art->content }}
                </p>
            </div>
        </div>
    @endif
@endforeach
@endsection
```

Page Article Detail merupakan tampilan ketika user meng-click button 'read more' pada article yang mereka pilih.

- Page Data Science :

```
@extends(view: 'layout.master')

@section(section: 'content')
<div class="m-4 mb-0">
    <h1 class="fw-bold">Data Science</h1>
</div>
<div class="container-fluid mt-4 mb-4">
    @foreach ($article as $art)
        <div class="card border-0 d-flex flex-row p-4">
            <div class="flex-shrink-0" style="width: 30rem; height: 15rem">
                
            </div>
            <div class="ms-3">
                <h2 class="fw-bold mb-1 mt-2">{{ $art->title }}</h2>
                @foreach ($writer as $writers)
                    @if ($writers->id == $art->writer_id)
                        <p class="mb-2 small">{{ date(format: 'd F Y', time: $writers->created_at) }}</p>
                    @endif
                @endforeach
                <p class="mb-3">{{ Str::words(value: $art->content, word_count: 10) }}</p>
                <div class="mt-auto text-end">
                    <a href="/user/article/{{ $art->id }}" class="btn btn-primary">Read More</a>
                </div>
            </div>
        </div>
    @endforeach
</div>
@endsection
```

Page Data Science adalah tampilan website yang muncul ketika user memilih category Data Science pada navbar. Pada page ini, tampilan yang muncul adalah data kumpulan artikel mengenai data science.

- Page Network Security :

```
1 @extends(view: 'layout.master')
2
3 @section(section: 'content')
4 <div class="m-4 mb-0">
5   <h1 class="fw-bold">Network Security</h1>
6 </div>
7 <div class="container-fluid mt-4 mb-4">
8   @foreach ($article as $art)
9     <div class="card border-0 d-flex flex-row p-4">
10       <div class="flex-shrink-0" style="width: 30rem; height: 150px;">
11         
12       </div>
13       <div class="ms-3">
14         <h2 class="fw-bold mb-1 mt-2">{{ $art->title }}</h2>
15         @foreach ($writer as $writers)
16           @if ($writers->id == $art->writer_id)
17             <p class="mb-2 small">{{ date(format: 'd-m-Y', $art->published_date) }}</p>
18           @endif
19         @endforeach
20         <p class="mb-3">{{ Str::words(value: $art->content, words: 10) }}</p>
21         <div class="mt-auto text-end">
22           <a href="/user/article/{{ $art->id }}" class="btn btn-primary">Read More</a>
23         </div>
24       </div>
25     </div>
26   @endforeach
27 </div>
28 @endsection
```

Page Network Security adalah tampilan website yang muncul ketika user memilih category Network Security pada navbar. Pada page ini, tampilan yang muncul adalah data kumpulan artikel mengenai network security.

- Page Popular :

```
@extends(view: 'layout.master')

@section(section: 'content')
<div class="m-4 mb-0">
  <h1 class="fw-bold">Popular</h1>
</div>
<div class="container-fluid mt-4 mb-4">
  @foreach ($article as $art)
    <div class="card border-0 d-flex flex-row p-4">
      <div class="flex-shrink-0" style="width: 30rem; height: 150px;">
        
      </div>
      <div class="ms-3">
        <h2 class="fw-bold mb-1 mt-2">{{ $art->title }}</h2>
        @foreach ($writer as $writers)
          @if ($writers->id == $art->writer_id)
            <p class="mb-2 small">{{ $art->published_date }}</p>
          @endif
        @endforeach
        <p class="mb-3">{{ Str::words(value: $art->content, words: 10) }}</p>
        <div class="mt-auto text-end">
          <a href="/user/article/{{ $art->id }}" class="btn btn-primary">Read More</a>
        </div>
      </div>
    </div>
  @endforeach
</div>
<div class="d-flex justify-content-center p-4">
  {{ $article->links() }}
</div>
@endsection
```

Page Popular digunakan untuk menampilkan kumpulan article yang sering user kunjungi.

- Page Writer Detail :

```
@extends(view: 'layout.master')

@section(section: 'content')
@foreach ($writer as $writers)
    @if ($writers->id == $id)
        <div class="m-4 mb-0 p-4 d-flex flex-row">
            <div style="width: 100px; height: 100px;">
                
            </div>
            <div class="m-4 mt-0 mr-0 mb-0 text-center d-flex flex-column">
                <h3 class="fw-bold">{{ $writers->name }}</h3>
                <h4>Spesialis {{ $writers->category }}</h4>
            </div>
        </div>
        <div class="container-fluid mb-4">
            @foreach ($article as $art)
                @if ($art->writer_id == $id)
                    <div class="card border-0 d-flex flex-row p-4">
                        <div class="flex-shrink-0" style="width: 30rem; height: 100px;">
                            
                        </div>
                        <div class="ms-3">
                            <h2 class="fw-bold mb-1 mt-2">{{ $art->title }}</h2>
                            @foreach ($writer as $writers)
                                @if ($writers->id == $art->writer_id)
                                    <p class="mb-2 small">{{ date(format: 'd-m-Y', strtotime($art->created_at)) }}</p>
                                @endif
                            @endforeach
                            <p class="mb-3">{{ Str::words(value: $art->content, limit: 150) }}</p>
                            <div class="mt-auto text-end">
                                <a href="/user/article/{{ $art->id }}" class="btn btn-primary">Read More</a>
                            </div>
                        </div>
                    </div>
                @endif
            @endforeach
        </div>
    @endif
@endforeach

```

Page Writer Detail digunakan untuk menampilkan article apa saja yang telah dibuat oleh writer/penulis.

- Page Writer :

```
@extends(view: 'layout.master')

@section(section: 'content')
<div class="container py-5 mb-5">
    <h2 class="fw-bold mb-5">Our Writers:</h2>
    <div class="row justify-content-start g-5">
        @foreach($writer as $writers)
            @csrf
            <div class="col-12 col-md-4">
                <div class="text-center">
                    <a href="{{ 'writerdetail/' . $writers->id }}" class="text-decoration: none; color: #007bff;">
                        <div class="position-relative" style="width: 100px; height: 100px; margin: 0 auto 10px auto;">
                            <div class="rounded-circle" style="width: 100%; height: 100%; background-color: #f0f0f0; display: flex; align-items: center; justify-content: center; position: relative; z-index: 1;">
                                
                            </div>
                        </div>
                    <h5 class="fw-bold mb-3">{{ $writers->name }}</h5>
                    <p>Spesialis {{ $writers->category }}</p>
                    </a>
                </div>
            </div>
        @endforeach
    </div>
</div>

```

Page Writer menampilkan profil mengenai writer/penulis dari database.

- Migration :

- Membuat migrasi tabel Writer :

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create(table: 'writer', callback: function (Blueprint $table): void {
            $table->id();
            $table->string(column: 'name');
            $table->enum(column: 'category', allowed: ['Data Science', 'Network Security']);
            $table->string(column: 'imageUrl');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists(table: 'writer');
    }
};
```

Untuk migrasi tabel Writer, data kolom yang dibuat adalah name (nama penulis), category (bidang yang dikuasai penulis), dan imageUrl (menyimpan data routing untuk image).

- Membuat migrasi tabel Article :

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create(table: 'article', callback: function (Blueprint $table): void {
            $table->id();
            $table->string(column: 'title');
            $table->foreignId(column: 'writer_id')->constrained(table: 'writer')->onDelete(action: 'cascade');
            $table->string(column: 'imageUrl');
            $table->enum(column: 'category', allowed: ['Data Science', 'Network Security']);
            $table->date(column: 'published_date');
            $table->text(column: 'content');
            $table->integer(column: 'click_count')->default(value: 0);
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists(table: 'article');
    }
};
```

Untuk migrasi tabel Article, data kolom yang dibuat adalah title (judul artikel), writer_id (id penulis sebagai foreign_id), imageUrl (menyimpan data routing untuk image), category (kategori artikel), published date (tanggal publish artikel), content (isi content pada artikel), dan click_count (data berupa angka seberapa banyak artikel dikunjungi).

- Database Seeder/Faker :

- Seeder Article :

```
class ArticleSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    0 references | 0 overrides
    public function run(): void
    {
        Article::create(attributes: [
            'title' => 'Machine Learning',
            'writer_id' => '1',
            'imageURL' => '/Images/ml.jpg',
            'category' => 'Data Science',
            'published_date' => '2024-01-30',
            'content' => 'Machine learning adalah cabang kecerdasan buatan yang mempelajari bagaimana membuat komputer melakukan tugas yang memerlukan kecerdasan manusia.',
            'click_count' => random_int(min: 1, max: 100)
        ]);
        Article::create(attributes: [
            'title' => 'Deep Learning',
            'writer_id' => '1',
            'imageURL' => '/Images/dl.jpg',
            'category' => 'Data Science',
            'published_date' => '2024-05-26',
            'content' => 'Deep learning adalah cabang dari machine learning yang menggunakan algoritma yang terinspirasi oleh struktur dan fungsi otak manusia.',
            'click_count' => random_int(min: 1, max: 100)
        ]);
        Article::create(attributes: [
            'title' => 'Natural Language Processing',
            'writer_id' => '1',
            'imageURL' => '/Images/nlp.jpg',
            'category' => 'Data Science',
            'published_date' => '2024-03-15',
            'content' => 'Natural Language Processing adalah cabang dari machine learning yang mempelajari bagaimana membuat komputer memahami dan menghasilkan bahasa manusia.',
            'click_count' => random_int(min: 1, max: 100)
        ]);
    }
}
```

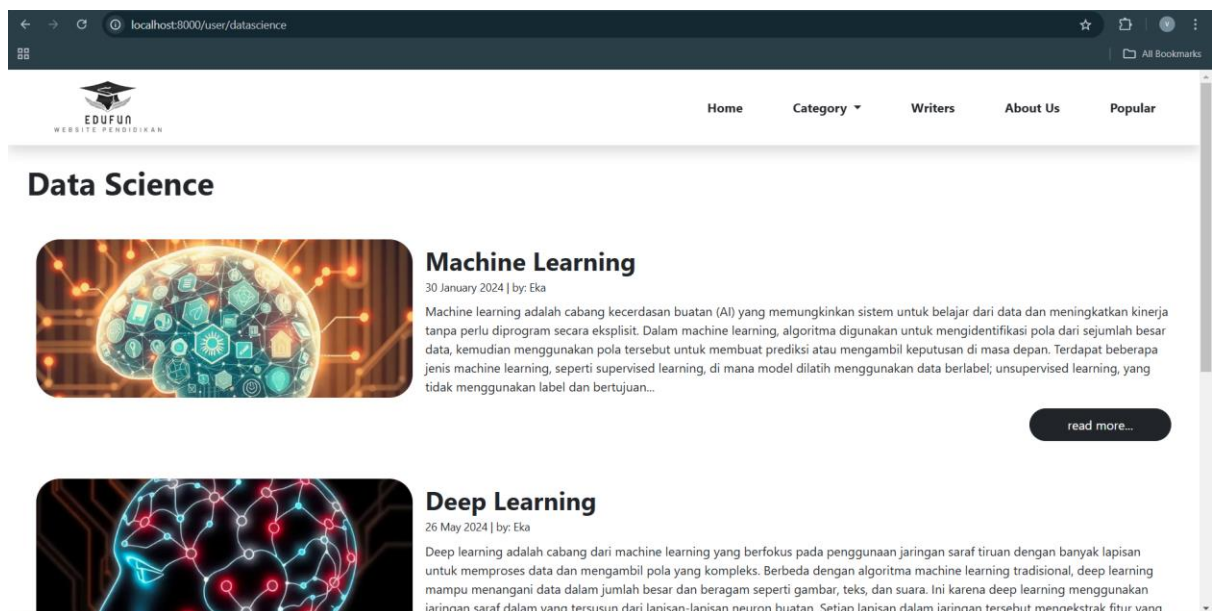
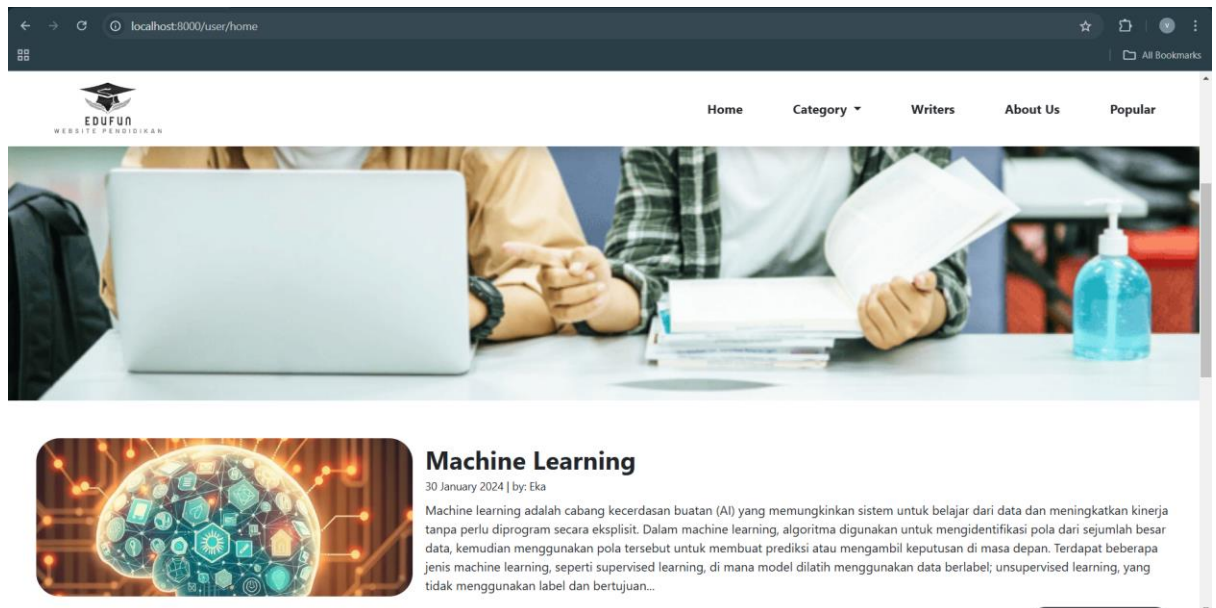
Digunakan untuk membuat data dummy article. Disini saya membuat data dummy sebanyak 6 artikel.

- Seeder Writer :


```
    /**
     * Run the database seeds.
     */
    0 references | 0 overrides
    public function run(): void
    {
        $faker = Faker::create(locale: 'id_ID');
        Writer::create(attributes: [
            'name' => $faker->name,
            'category' => 'Data Science',
            'imageURL' => '/Images/writer.jpg'
        ]);
        Writer::create(attributes: [
            'name' => $faker->name,
            'category' => 'Data Science',
            'imageURL' => '/Images/writer.jpg'
        ]);
        Writer::create(attributes: [
            'name' => $faker->name,
            'category' => 'Network Security',
            'imageURL' => '/Images/writer.jpg'
        ]);
    }
}
```

Digunakan untuk membuat data dummy writer. Saya membuat data dummy writer sebanyak 3 orang dan menggunakan faker name untuk membuat fake name dari writer/penulis.

Screenshoot tampilan website :




localhost:8000/user/networksecurity



EDUFUN
WEBSITE PENDIDIKAN

HomeCategoryWritersAbout UsPopular

Network Security




Apa itu Network Security

23 January 2024 | by: Laila

Network security adalah praktik untuk melindungi integritas, kerahasiaan, dan ketersediaan jaringan dan data yang ada di dalamnya dari ancaman atau akses yang tidak sah. Hal ini mencakup berbagai langkah, mulai dari perangkat keras hingga perangkat lunak, seperti firewall, enkripsi data, sistem deteksi intrusi, dan pemantauan aktivitas jaringan. Keamanan jaringan menjadi sangat penting dalam dunia digital saat ini, di mana ancaman seperti serangan malware, phishing, dan DDoS dapat merusak infrastruktur, mencuri...

[read more...](#)




Software Security

12 August 2024 | by: Laila

Keamanan perangkat lunak mengacu pada praktik, langkah-langkah, dan teknik yang dirancang untuk melindungi aplikasi dan sistem perangkat lunak dari ancaman dan kerentanan keamanan. Keamanan perangkat lunak melibatkan upaya memastikan bahwa perangkat lunak tahan terhadap serangan seperti akses tidak sah, pelanggaran data, dan manipulasi jahat. Komponen utama keamanan perangkat lunak meliputi praktik pengodean yang aman, penilaian kerentanan rutin, enkripsi, dan autentikasi pengguna. Dengan...


localhost:8000/user/writers




EDUFUN
WEBSITE PENDIDIKAN

HomeCategoryWritersAbout UsPopular


Our Writers:



Eka Tantri Kuswandari S.T.
Spesialis Data Science




Bagya Sitompul
Spesialis Data Science



Laila Agustina
Spesialis Network Security

localhost:8000/user/aboutus



EDUFUN
WEBSITE PENDIDIKAN

HomeCategoryWritersAbout UsPopular

About EduFun

EduFun adalah perusahaan pendidikan berbasis teknologi asal Indonesia. EduFun menyediakan layanan akses pendidikan dalam format tulisan berbahasa Indonesia yang disajikan secara online melalui website.

Hingga Juni 2024, EduFun memiliki lebih dari 10 ribu pengguna. EduFun hadir sebagai bentuk revolusi dari pendidikan di Indonesia dengan mengedepankan cara berpikir kritis, logis, rasional, dan sumber pengetahuan sains yang terintegrasi terhadap semua mahasiswa IT di Indonesia. EduFun bercita-cita mencetak generasi Indonesia yang memahami ilmu pengetahuan dan cinta belajar.

