

## EduFun

### 1. Hasil Website

#### A. Home

EduFun

Home Category Writers About Us Popular



Article



**Human and Computer Interaction**  
04 Nov 2024 | by: Qori Maimunah Halimah  
Human-Computer Interaction (HCI) focuses on the design and use of computer technology, particularly the interfaces between humans and computers. This...

[read more...](#)



**User Experience**  
04 Nov 2024 | by: Dimas Lasmono Nababan S.Sos  
User Experience (UX) encompasses all aspects of end-user interaction with a company, its services, and products. It goes beyond mere usability to incl...

[read more...](#)

#### B. Category

##### - Interactive Multimedia

EduFun

Home Category Writers About Us Popular

Interactive Multimedia



**Human and Computer Interaction**  
04 Nov 2024 | by: Qori Maimunah Halimah  
Human-Computer Interaction (HCI) focuses on the design and use of computer technology, particularly the interfaces between humans and computers. This...

[read more...](#)



**User Experience**  
04 Nov 2024 | by: Dimas Lasmono Nababan S.Sos  
User Experience (UX) encompasses all aspects of end-user interaction with a company, its services, and products. It goes beyond mere usability to incl...

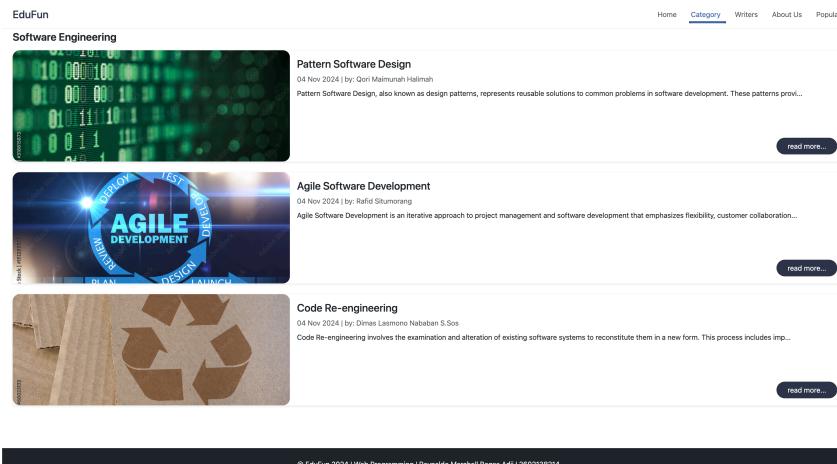
[read more...](#)



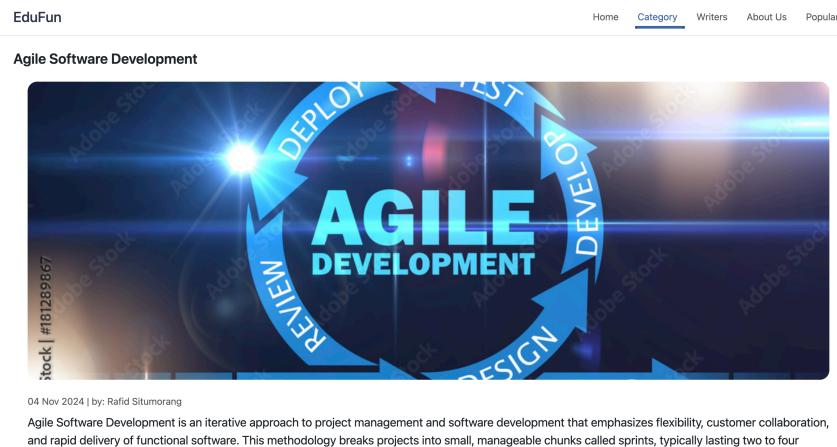
**User Experience for Digital Immersive Technology**  
04 Nov 2024 | by: Dimas Lasmono Nababan S.Sos  
User Experience in Digital Immersive Technology focuses on creating engaging and natural interactions in virtual and augmented reality environments. T...

[read more...](#)

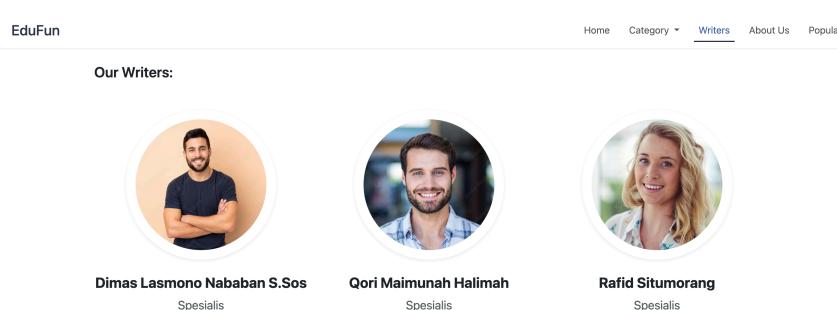
##### - Software Engineering



### C. Article Detail



### D. Writers



### E. Writer Detail

EduFun

Home Category Writers About Us Popular

**Qori Maimunah Halimah**  
Spesialis

**Human and Computer Interaction**  
04 Nov 2024 | by: Qori Maimunah Halimah  
Human-Computer Interaction (HCI) focuses on the design and use of computer technology, particularly the interfaces between humans and computers. This...

[read more...](#)

**Pattern Software Design**

© EduFun 2024 | Web Programming | Reynaldo Marchell Bagas Adji | 2602138214

## F. About Us

EduFun

Home Category Writers **About Us** Popular

## 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 mengejarkan 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.

© EduFun 2024 | Web Programming | Reynaldo Marchell Bagas Adji | 2602138214

## G. Popular

EduFun

Home Category Writers About Us Popular

This...

[read more...](#)

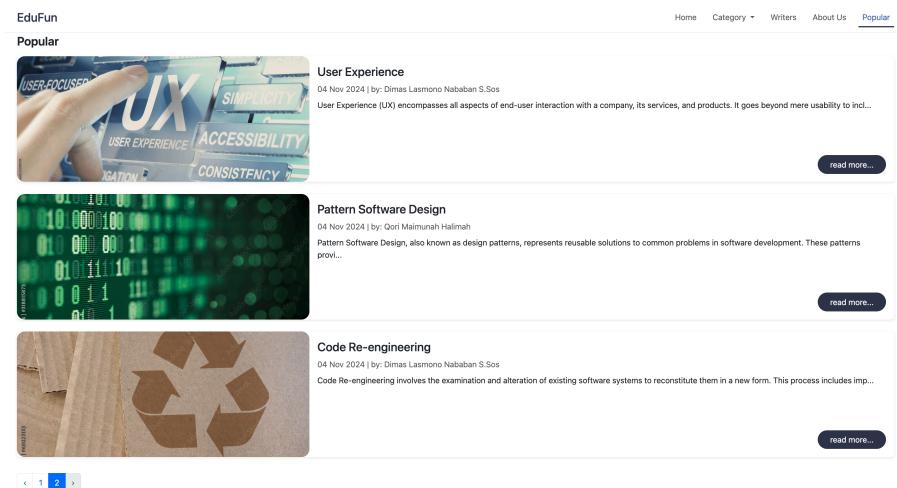
**User Experience for Digital Immersive Technology**  
04 Nov 2024 | by: Dimas Lasmono Nababan S.Sos  
User Experience in Digital Immersive Technology focuses on creating engaging and natural interactions in virtual and augmented reality environments. T...

[read more...](#)

**Agile Software Development**  
04 Nov 2024 | by: Rafid Situmorang  
Agile Software Development is an iterative approach to project management and software development that emphasizes flexibility, customer collaboration...

[read more...](#)

© EduFun 2024 | Web Programming | Reynaldo Marchell Bagas Adji | 2602138214



The screenshot shows a website layout with a header 'EduFun' and a navigation bar with links for Home, Category, Writers, About Us, and Popular. Below the header, there's a section titled 'Popular' with three article cards. The first card is titled 'User Experience' with a thumbnail showing a hand interacting with a digital interface. The second card is titled 'Pattern Software Design' with a thumbnail showing binary code. The third card is titled 'Code Re-engineering' with a thumbnail showing a recycling symbol. Each card has a 'read more...' button.

## 2. Penjelasan Code

### A. Routing

```
<?php

use App\Http\Controllers\AboutController;
use App\Http\Controllers\HomeController;
use App\Http\Controllers\ArticleController;
use App\Http\Controllers\PopularController;
use App\Http\Controllers\WriterController;
use Illuminate\Support\Facades\Route;

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

Saya membuat 7 route yang semuanya merupakan page route untuk menampilkan tampilan menggunakan method get(). Setiap route saya berikan name/alias untuk memudahkan navigasi di controller dan view.

### B. Controller

Saya membuat 5 file controller yang berkorespondensi dengan route yang sudah dibuat sebelumnya.

- **HomeController**

```
2 references | 0 implementations | You, 2 days ago | 1 author (You)
class HomeController extends Controller
{
    1 reference | 0 overrides
    public function index(): Factory|View
    {
        $articles = Article::with(relations: 'writer')->get();
        return view(view: 'home.index', data: ['articles' => $articles]);
    }
}
```

Data articles diambil dari database dengan menyertakan writer yang berelasi dengan article tersebut kemudian dikirimkan ke view.

- **ArticleController**

```
class ArticleController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    1 reference | 0 overrides
    public function index(string $category): Factory|View
    {
        $articles = Article::with(relations: 'writer')->where(column: "category", operator: $category)->get();

        return view(view: 'article.index', data: ['articles' => $articles, 'category'=> $category]);
    }
}
```

Di method index(), data articles diambil dari database dengan menyertakan writer dan category yang sesuai kemudian dikirimkan ke view.

```
public function show(string $id): Factory|View
{
    $article = Article::with(relations: 'writer')->where(column: "id", operator: $id)->first();

    // update the view_count column +1
    $article->increment(column: 'view_count');

    return view(view: 'article.show', data: ['article' => $article]);
}
```

Di method show(), data article diambil dari database dengan menyertakan writer dan id yang sesuai kemudian dikirimkan ke view. Setiap kali user mengklik article tersebut, maka column ‘view\_count’ pada table articles akan bertambah 1.

- **WriterController**

```
class WriterController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    1 reference | 0 overrides
    public function index(): Factory|View
    {
        $writers = Writer::all();
        return view(view: 'writer.index', data: ['writers' => $writers]);
    }
}
```

Di method index(), data seluruh writers diambil dari database kemudian dikirimkan ke view.

```
public function show(string $id): Factory|View
{
    $writer = Writer::find(id: $id);
    $articles = $writer->articles;

    return view(view: 'writer.show', data: ['writer'=> $writer, 'articles' => $articles]);
}
```

Pada method show(), data writer diambil berdasarkan id nya. Data article diambil dari article yang dimiliki oleh writer tersebut, kemudian kedua variable tersebut dikirimkan ke view.

- **AboutController**

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

Controller ini hanya mengirimkan view yang akan ditampilkan.

- **PopularController**

```
class PopularController extends Controller
{
    1 reference | 0 overrides
    public function index(): Factory|View
    {
        $articles = Article::with(relations: 'writer')->orderBy(column: 'view_count', direction: 'desc')->paginate
            (perPage: 3);
        return view(view: "popular.index", data: ['articles' => $articles]);
    }
}
```

Data articles beserta writer yang bersesuaian diambil berdasarkan column ‘view\_count’ dan diurutkan secara descending, kemudian dikirimkan ke view. Data tersebut juga menggunakan method paginate() dengan setiap page memiliki 3 halaman.

### C. Model

Saya membuat 2 model yaitu Writer dan Article dengan relasi 1 to many seperti berikut:

```
class Writer extends Model
{
    use HasFactory;
    0 references
    protected $guarded = [];

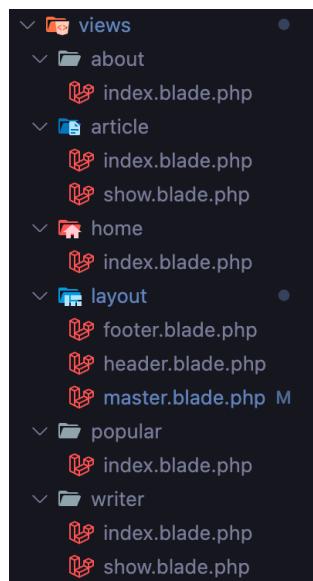
    0 references | 0 overrides
    public function articles(): HasMany{
        return $this->hasMany(related: Article::class);
    }
}

class Article extends Model
{
    use HasFactory;
    0 references
    protected $guarded = [];

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

### D. View and Blade

Berikut adalah layout dari view menggunakan blade template.



Setiap view bersesuaian dengan route dan controller yang sudah dideklarasikan sebelumnya.

### E. Migration

Saya membuat 2 migration yaitu 'create\_writers\_table' dan 'create\_articles\_table'.

```
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create(table: 'writers', callback: function (Blueprint $table): void {
            $table->id();
            $table->string(column: 'name');
            $table->string(column: 'position');
            $table->string(column: 'image');
            $table->timestamps();
        });
    }

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

Table ‘writers’ memiliki column id, name, position, image, dan timestamp (created\_at & updated\_at).

```
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create(table: 'articles', callback: function (Blueprint $table): void {
            $table->id();
            $table->string(column: 'title');
            $table->longText(column: 'content');
            $table->enum(column: 'category', allowed: ['IM', 'SE']);
            $table->string(column: 'image');
            $table->bigInteger(column: 'view_count')->default(value: 0);
            $table->foreignId(column: 'writer_id')->constrained(table: 'writers')->onDelete(action: 'cascade');
            $table->timestamps();
        });
    }

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

Table ‘articles’ memiliki column id, title, content, category (IM untuk Interactive Multimedia & SE untuk Software Engineering), image, view\_count, dan timestamp (created\_at & updated\_at). Terdapat juga foreignId yang mengarah ke table ‘writers’ dengan column id, yang saya beri nama writer\_id.

## F. Database Seeder/Faker

Saya membuat 2 seeder yaitu ‘ArticleSeeder’ dan ‘WriterSeeder’, masing-masing menggunakan Faker untuk membuat data palsu. Saya juga membuat array of object yang sesuai untuk data pada masing-masing table tersebut.

```
class ArticleSeeder extends Seeder
{
    public function run(): void
    {
        $articles = [
            [
                'title' => 'Code Re-engineering',
                'content' => 'Code Re-engineering involves the examination and alteration of existing software systems to reconstitute them in a new form. This process includes improving the systems functionality, performance, maintainability while preserving its essential functions. The practice encompasses various activities such as reverse engineering, code refactoring, data restructuring, and forward engineering. Engineering involves the current system architecture, identify problematic areas, and implement modern design patterns and best practices. Key aspects include improving code readability, reducing technical debt, optimizing performance, and ensuring compatibility with modern technologies. The process often involves refactoring code, updating deprecated functions, and implementing automated testing. Successful re-engineering requires careful planning, thorough documentation, and strategic implementation to minimize risks while maximizing system improvements. This practice is essential for modernizing legacy systems and extending their operational lifespan.',
                'image' => 'https://as2.ftcdn.net/v2/jpg/00/68/31/1000_E_68023133.AP2ehRaCtuShYJV8059F0L0BxbVedh6.jpg'
            ],
            [
                'category' => 'SE',
                'title' => 'Code Re-engineering',
                'content' => 'Code Re-engineering involves the examination and alteration of existing software systems to reconstitute them in a new form. This process includes improving the systems functionality, performance, maintainability while preserving its essential functions. The practice encompasses various activities such as reverse engineering, code refactoring, data restructuring, and forward engineering. Engineering involves the current system architecture, identify problematic areas, and implement modern design patterns and best practices. Key aspects include improving code readability, reducing technical debt, optimizing performance, and ensuring compatibility with modern technologies. The process often involves refactoring code, updating deprecated functions, and implementing automated testing. Successful re-engineering requires careful planning, thorough documentation, and strategic implementation to minimize risks while maximizing system improvements. This practice is essential for modernizing legacy systems and extending their operational lifespan.',
                'image' => 'https://as1.ftcdn.net/v2/jpg/01/81/28/98/1000_F_181289867_9kJKVvLUShaM6GB7jlpfVQMaedUS16nK.jpg'
            ],
            [
                'category' => 'SE',
                'title' => 'Code Re-engineering',
                'content' => 'Code Re-engineering involves the examination and alteration of existing software systems to reconstitute them in a new form. This process includes improving the systems functionality, performance, maintainability while preserving its essential functions. The practice encompasses various activities such as reverse engineering, code refactoring, data restructuring, and forward engineering. Engineering involves the current system architecture, identify problematic areas, and implement modern design patterns and best practices. Key aspects include improving code readability, reducing technical debt, optimizing performance, and ensuring compatibility with modern technologies. The process often involves refactoring code, updating deprecated functions, and implementing automated testing. Successful re-engineering requires careful planning, thorough documentation, and strategic implementation to minimize risks while maximizing system improvements. This practice is essential for modernizing legacy systems and extending their operational lifespan.',
                'image' => 'https://as2.ftcdn.net/v2/jpg/00/68/31/1000_E_68023133.AP2ehRaCtuShYJV8059F0L0BxbVedh6.jpg'
            ]
        ];
        foreach ($articles as $article) {
            Article::create(attributes: [
                'writer_id' => $writers->random()->id,
                'category' => $article->category,
                'title' => $article->title,
                'content' => $article->content,
                'image' => $article->image,
            ]);
        }
    }
}
```

```
class UserSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    0 references | 0 overrides
    public function run(): void
    {
        $faker = Faker::create(locale: 'id_ID');

        for ($i = 0; $i < 5; $i++) {
            User::create(attributes: [
                'name' => $faker->name,
                'email' => $faker->unique()->email,
                'password' => bcrypt(value: 'password'),
            ]);
        }
    }
}
```

Kedua seeder tersebut saya panggil dalam ‘DatabaseSeeder’ seperti berikut:

```
class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     */
    0 references | 0 overrides
    public function run(): void
    {
        // User::factory(10)->create();

        User::factory()->create(attributes: [
            'id' => 1,
            'name' => 'Test User',
            'email' => 'test@example.com',
            'password' => bcrypt(value: 'password'),
        ]);

        $this->call(class: [
            UserSeeder::class,
            WriterSeeder::class,
            ArticleSeeder::class,
        ]);
    }
}
```

**Link GitHub:** <https://github.com/reynaldomarchell/edufun.git>