

LAMPIRAN

Lampiran 1 Source Code

ArticleController.php

```
<?php
namespace App\Http\Controllers;
use App\Models\Article;
use App\Enums\ArticleStatus;
use Illuminate\Http\Request;
use App\Http\Resources;
use
Illuminate\Routing\Controllers\HasMiddlew
are;
use
Illuminate\Routing\Controllers\Middleware;

class ArticleController extends Controller
implements HasMiddleware
{
    public static function middleware()
    {
        return [
            new Middleware(
                middleware: ['auth'],
                except: ['index', 'show', 'search']
            )
        ];
    }
    /**
     * Display a listing of the resource.
     */
    public function index($key = 'latest')
    {
        $articles = Article::query()
            ->select(['id', 'category_id', 'user_id',
            'title', 'slug', 'thumbnail', 'teaser', 'status',
            'published_at'])
            ->with(['category:id,name,slug',
            'user:id,name'])
            ->where('status',
            ArticleStatus::Diterbitkan)
            ->when($key === 'latest', fn ($query)
            => $query->latest('published_at'))
            ->when($key === 'year', fn ($query)
            => $query->popularThisYear())
            ->when($key === 'month', fn
            ($query) => $query->popularThisMonth())
            ->when($key === 'week', fn ($query)
            => $query->popularThisWeek())

            ->when($key === 'all-time', fn
            ($query) => $query->popularAllTime())
            ->paginate(9);
            abort_unless($articles->count(), 404);
            $pageMeta = [
                'latest' => ['title' => $title = 'Arsip
                Kuliner', 'description' => $description =
                'Jelajahi kekayaan kuliner Indonesia dari
                berbagai penjuru Nusantara.'],
                'trending' => ['title' => 'Trending
                Articles', 'description' => 'The most trending
                articles.'],
                'most-likes' => ['title' => 'Most Likes
                Article', 'description' => 'The most likes
                articles.'],
                'year' => ['title' => 'Populer Tahun
                Ini', 'description' => 'Artikel populer tahun
                ini.'],
                'month' => ['title' => 'Populer Bulan
                Ini', 'description' => 'Artikel populer bulan
                ini.'],
                'week' => ['title' => 'Populer Minggu
                Ini', 'description' => 'Artikel populer minggu
                ini.'],
                'all-time' => ['title' => 'Populer
                Sepanjang Masa', 'description' => 'Artikel
                populer sepanjang masa.'],
                ][$key] ?? ['title' => $title, 'description'
                => $description];
            return inertia('articles/index', [
                'articles' => fn () =>
                Resources\ArticleBlockResource::collection
                ($articles)
                ->additional(['meta' =>
                ['has_pages' => $articles->hasPages()],
                'page_meta' => $pageMeta,
                ]);
            }
            /**
             * Search the specified resource.
             */
            public function search(Request $request)
            {
                return Article::query()
                    ->select('id', 'title', 'slug')
                    ->where('title', 'like', "%{$request-
                    >search}%")
            }
        }
    }
}
```

```

-
>whereStatus(ArticleStatus::Diterbitkan)
->get()
->map(fn ($article) => [
    'id' => $article->id,
    'title' => $article->title,
    'href' => route('articles.show',
$article->slug),
]);
}
/**
 * Display the specified resource.
 */
public function show(Article $article)
{
    $article->visit()->hourlyIntervals()-
>withIp()->withSession()->withUser();

    $relatedArticles = Article::query()-
>select(['id', 'title', 'slug', 'teaser', 'user_id'])
->where('category_id', $article-
>category_id)
->with('user:id,name')
->where('id', '!=', $article->id)
->latest('published_at')
->limit(4)
->get();
    return inertia('articles/show', [
        'article' => fn () => new
Resources\ArticleSingleResource(
            $article-
>load(['category:id,name,slug',
'user:id,name', 'tags:id,name,slug'])
        ),
        'articles' => fn () => $relatedArticles,
    ]);
}
}

```

CategoryController.php

```

<?php
namespace App\Http\Controllers;
use App\Http\Requests\CategoryRequest;
use App\Http\Resources;
use App\Models\Category;
use Illuminate\Http\Request;

class CategoryController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()

```

```

{
    $categories =
Resources\CategoryListResource::collection
(
        $self = Category::query()
        ->select(['id', 'name', 'slug'])
        ->withCount('articles')
        ->orderBy('name', 'asc')
        ->paginate(10)
    )->additional(['meta' => ['has_pages' =>
$self->hasPages()]]);

    return inertia('categories/index', [
        'categories' => fn () => $categories,
    ]);
}

/**
 * Show the form for creating a new
resource.
 */
public function create()
{
    return inertia('categories/form', [
        'category' => new Category,
        'page_meta' => [
            'title' => 'Tambah Daerah Kuliner',
            'description' => 'Tambah daerah
kuliner baru di bawah ini.',
            'url' => route('categories.store'),
            'method' => 'post',
        ],
    ]);
}

/**
 * Store a newly created resource in
storage.
 */
public function store(CategoryRequest
$request)
{
    Category::create([
        'name' => $request->name,
        'slug' => str($request->name)-
>slug(),
        'thumbnail' => $request-
>file('thumbnail') ? $request-
>file('thumbnail')->store('thumbnails',
'public') : null,
        'teaser' => $request->teaser,
    ]);

    return to_route('categories.index');
}

```

```

/**
 * Display the specified resource.
 */
public function show(Category $category)
{
    $articles =
Resources\ArticleBlockResource::collection
($self = $category->articles()
->select(['id', 'category_id', 'user_id',
'title', 'slug', 'teaser', 'thumbnail',
'published_at'])
->with(['category:id,name,slug',
'user:id,name']))
->where('status',
\App\Enums\ArticleStatus::Diterbitkan)
->latest('published_at')
->paginate(9))
->additional([
'meta' => ['has_pages' => $self-
>hasPages()]
]);

return inertia('articles/index', [
'articles' => fn () => $articles,
'page_meta' => [
'title' => $category->name,
'description' => $category->teaser,
],
]);
}

/**
 * Show the form for editing the specified
resource.
 */
public function edit(Category $category)
{
    return inertia('categories/form', [
'category' => $category,
'page_meta' => [
'title' => 'Edit Daerah Kuliner',
'description' => 'Edit daerah
kuliner di bawah ini.',
'url' => route('categories.update',
$category),
'method' => 'put',
],
]);
}

/**
 * Update the specified resource in
storage.
 */

```

```

public function update(Request $request,
Category $category)
{
    $category->update([
'name' => $request->name,
'slug' => str($request->name)-
>slug(),
'thumbnail' => $request-
>file('thumbnail') ? $request-
>file('thumbnail')->store('thumbnails',
'public') : '',
'teaser' => $request->teaser,
]);

    return to_route('categories.index');
}

/**
 * Remove the specified resource from
storage.
 */
public function destroy(Category
$category)
{
    if ($category->articles()->exists()) {
        return back();
    }

    $category->delete();

    return back();
}
}

```

HomeController.php

```

<?php
namespace App\Http\Controllers;
use App\Enums;
use
App\Http\Resources\ArticleBlockResource;
use App\Models\Article;
use Illuminate\Http\Request;

class HomeController extends Controller
{
    public function __invoke(Request
$request)
    {
        return inertia('home', [
'popular_articles' => fn () =>
ArticleBlockResource::collection(
Article::query()

```

```

        ->select(['id', 'category_id',
'user_id', 'title', 'slug', 'thumbnail', 'teaser',
'published_at'])
        ->with(['category:id,name,slug',
'user:id,name'])
        -
>whereStatus(Enums\ArticleStatus::Diterbitkan)
        ->popularThisWeek()
        ->limit(3)
        ->get()
    ),
    'articles' => fn () =>
ArticleBlockResource::collection(
    Article::query()
        ->select(['id', 'category_id',
'user_id', 'title', 'slug', 'thumbnail', 'teaser',
'published_at'])
        ->with(['category:id,name,slug',
'user:id,name'])
        ->where('status',
Enums\ArticleStatus::Diterbitkan)
        ->latest('published_at')
        ->limit(6)
        ->get()
    ),
]);
}
}

```

InternalArticleController.php

```

<?php
namespace App\Http\Controllers;
use App\Enums;
use
App\Http\Resources\ArticleBlockResource;
use App\Models\Article;
use Illuminate\Http\Request;

class HomeController extends Controller
{
    public function __invoke(Request
$request)
    {
        return inertia('home', [
            'popular_articles' => fn () =>
ArticleBlockResource::collection(
                Article::query()
                    ->select(['id', 'category_id',
'user_id', 'title', 'slug', 'thumbnail', 'teaser',
'published_at'])
                    ->with(['category:id,name,slug',
'user:id,name'])

```

```

        -
>whereStatus(Enums\ArticleStatus::Diterbitkan)
        ->popularThisWeek()
        ->limit(3)
        ->get()
    ),
    'articles' => fn () =>
ArticleBlockResource::collection(
    Article::query()
        ->select(['id', 'category_id',
'user_id', 'title', 'slug', 'thumbnail', 'teaser',
'published_at'])
        ->with(['category:id,name,slug',
'user:id,name'])
        ->where('status',
Enums\ArticleStatus::Diterbitkan)
        ->latest('published_at')
        ->limit(6)
        ->get()
    ),
]);
}
}

```

ProfileController.php

```

<?php
namespace App\Http\Controllers;
use
App\Http\Requests\ProfileUpdateRequest;
use
Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Redirect;
use Inertia\Inertia;
use Inertia\Response;

class ProfileController extends Controller
{
    /**
     * Display the user's profile form.
     */
    public function edit(Request $request):
Response
    {
        return inertia('profile/edit', [
            'mustVerifyEmail' => $request-
>user() instanceof MustVerifyEmail,
            'status' => session('status'),
        ]);
    }
}

```

```

    }

    /**
     * Update the user's profile information.
     */
    public function
    update(ProfileUpdateRequest $request):
    RedirectResponse
    {
        $request->user()->fill($request-
        >validated());

        if ($request->user()->isDirty('email'))
        {
            $request->user()->email_verified_at
            = null;
        }

        $request->user()->save();

        return Redirect::route('profile.edit');
    }

    /**
     * Delete the user's account.
     */
    public function destroy(Request
    $request): RedirectResponse
    {
        $request->validate([
            'password' => ['required',
            'current_password'],
        ]);

        $user = $request->user();
        Auth::logout();

        $user->articles()->delete();

        $user->delete();

        $request->session()->invalidate();
        $request->session()-
        >regenerateToken();

        return Redirect::to('/');
    }
}

```

TagController.php

```

<?php
namespace App\Http\Controllers;
use App\Http\Requests\TagRequest;

```

```

use App\Http\Resources;
use App\Models\Tag;
use Illuminate\Http\Request;

class TagController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        $tags =
        Resources\TagListResource::collection(
            $self = Tag::query()
                ->select(['id', 'name', 'slug'])
                ->withCount('articles')
                ->latest('updated_at')
                ->paginate(10)
            )->additional(['meta' => ['has_pages' =>
            $self->hasPages()]]);

        return inertia('tags/index', [
            'tags' => fn () => $tags,
        ]);
    }

    /**
     * Show the form for creating a new
    resource.
     */
    public function create()
    {
        return inertia('tags/form', [
            'tag' => new Tag,
            'page_meta' => [
                'title' => 'Tambah Jenis Kuliner',
                'description' => 'Tambah jenis
                kuliner baru di bawah ini.',
                'url' => route('tags.store'),
                'method' => 'post',
            ],
        ]);
    }

    /**
     * Store a newly created resource in
    storage.
     */
    public function store(TagRequest
    $request)
    {
        Tag::create([
            'name' => $request->name,
            'slug' => str($request->name)-
            >slug(),

```

```

        'thumbnail' => $request-
>file('thumbnail') ? $request-
>file('thumbnail')->store('thumbnails',
'public') : null,
        'teaser' => $request->teaser,
    ]);

    return to_route('tags.index');
}

/**
 * Display the specified resource.
 */
public function show(Tag $tag)
{
    $articles =
Resources\ArticleBlockResource::collection
($self = $tag->articles())
    ->select(['id', 'category_id', 'user_id',
'title', 'slug', 'thumbnail', 'teaser',
'published_at'])
    ->with(['category:id,name,slug',
'user:id,name'])
    ->where('status',
\App\Enums\ArticleStatus::Diterbitkan)
    ->latest('published_at')
    ->paginate(9))
    ->additional([
        'meta' => ['has_pages' => $self-
>hasPages()]
    ]);

    return inertia('articles/index', [
        'articles' => fn () => $articles,
        'page_meta' => [
            'title' => $tag->name,
            'description' => $tag->teaser,
        ],
    ]);
}

/**
 * Show the form for editing the specified
resource.
 */
public function edit(Tag $tag)
{
    return inertia('tags/form', [
        'tag' => $tag,
        'page_meta' => [
            'title' => 'Edit Jenis Kuliner',
            'description' => 'Edit jenis kuliner
dibawah ini.',
            'url' => route('tags.update', $tag),
            'method' => 'put',

```

```

    ],
    ]);
}
/**
 * Update the specified resource in
storage.
 */
public function update(Request $request,
Tag $tag)
{
    $tag->update([
        'name' => $request->name,
        'slug' => str($request->name)-
>slug(),
        'thumbnail' => $request-
>file('thumbnail') ? $request-
>file('thumbnail')->store('thumbnails',
'public') : "",
        'teaser' => $request->teaser,
    ]);

    return to_route('tags.index');
}

/**
 * Remove the specified resource from
storage.
 */
public function destroy(Tag $tag)
{
    if ($tag->articles()->exists()) {
        return back();
    }

    $tag->delete();

    return back();
}
}

```

Article.php

```

<?php
namespace App\Models;

use App\Observers\ArticleObserver;
use Coderflex\Laravisit\Concerns\CanVisit;
use Coderflex\Laravisit\Concerns\HasVisits;
use
Illuminate\Database\Eloquent\Attributes\Ob
servedBy;
use
Illuminate\Database\Eloquent\Factories\Has
Factory;

```

```

use Illuminate\Database\Eloquent\Model;
use
Illuminate\Database\Eloquent\Relations\BelongsToMany;
use
Illuminate\Database\Eloquent\Relations\HasMany;
use
Illuminate\Database\Eloquent\Relations\MorphMany;
use
Illuminate\Database\Eloquent\Relations\MorphToMany;

#[ObservedBy([ArticleObserver::class])]
class Article extends Model implements CanVisit
{
    use HasFactory;
    use HasVisits;

    protected $guarded = [];

    public function user(): BelongsToMany
    {
        return $this->belongsToMany(User::class);
    }

    public function tags(): MorphToMany
    {
        return $this->morphToMany(Tag::class, 'taggable');
    }

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

    protected function casts(): array
    {
        return [
            'status' => \App\Enums\ArticleStatus::class,
            'published_at' => 'datetime',
        ];
    }
}

```

Category.php

```

<?php
namespace App\Models;

```

```

use
Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use
Illuminate\Database\Eloquent\Relations\HasMany;
use App\Traits\HasLabelValue;

class Category extends Model
{
    use HasLabelValue;

    protected $guarded = [];

    public function articles(): HasMany
    {
        return $this->hasMany(Article::class, 'category_id');
    }
}

```

Tag.php

```

<?php
namespace App\Models;
use App\Traits\HasLabelValue;
use Illuminate\Database\Eloquent\Model;
use
Illuminate\Database\Eloquent\Relations\MorphToMany;

class Tag extends Model
{
    use HasLabelValue;

    protected $guarded = [];

    public function articles(): MorphToMany
    {
        return $this->morphedByMany(Article::class, 'taggable');
    }
}

```

User.php

```

<?php
namespace App\Models;
use
Illuminate\Contracts\Auth\MustVerifyEmail;

```

```

use
Illuminate\Database\Eloquent\Factories\Has
Factory;
use
Illuminate\Database\Eloquent\Relations\Has
Many;
use Illuminate\Foundation\Auth\User as
Authenticatable;
use Illuminate\Notifications\Notifiable;
use Spatie\Permission\Traits\HasRoles;

class User extends Authenticatable
{
    use HasFactory, Notifiable, HasRoles;

    /**
     * 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>
     */
    public function gravatar($size = 200):
    string
    {
        return
        'https://www.gravatar.com/avatar/' .
        md5(strtolower(trim($this->email))) . '?s=' .
        $size . '&d=mp';
    }

    public function articles(): HasMany
    {
        return $this->hasMany(Article::class,
        'user_id');
    }

```

```

}

protected function casts(): array
{
    return [
        'email_verified_at' => 'datetime',
        'password' => 'hashed',
    ];
}
}

```

app-layout.jsx

```

import { Toaster } from
'@/components/ui/sonner';
import { Footer } from
'@/components/footer';
import { Navigation } from
'@/layouts/navigation/index';

export function AppLayout({ children }) {
    return (
        <div>
            <Toaster />
            <Navigation />
            <main className="min-h-
            [calc(100vh - _theme(spacing.16))] flex-1
            flex-col gap-4 py-6 md:py-16">
                {children}
            </main>
            <Footer />
        </div>
    );
}

```

guest-layout.jsx

```

import { Logo } from '@/components/logo';
import { NewLogo } from
'@/components/logo-new';
import { Link } from '@inertiajs/react';

export function GuestLayout({ children }) {
    const appName =
    import.meta.env.VITE_APP_NAME;
    return (
        <div className="flex items-center
        justify-center sm:min-h-screen">
            <div className="w-full max-w-lg
            py-6">
                <Link href="/" className="mb-6
                flex items-center justify-center gap-x-2">

```



```

        <NewLogo className="size-
10" />
        <span className="sr-only">Go
to {appName} (Home)</span>
        </Link>
        {children}
      </div>
    </div>
  );
}

```

user-layout.jsx

```

import { Aside } from
'@/components/aside';
import { Head } from '@inertiajs/react';
import { Container } from
'@/components/container';
import { Toaster } from
'@/components/ui/sonner';
import { Navigation } from
'@/layouts/navigation/index';

export function UserLayout({ title, children
}) {
  return (
    <div className="flex min-h-screen w-
full flex-col">
      <Toaster />
      <Head title={title} />
      <Navigation />
      <main className="min-h-
[calc(100vh - _theme(spacing.16))] bg-
muted/40 py-6 lg:gap-8 lg:py-16">
        <div className="-mt-6 mb-6
border-b bg-background py-6 lg:-mt-16
lg:mb-12">
          <Container>
            <h1 className="text-xl font-
semibold lg:text-3xl">{title}</h1>
          </Container>
        </div>
        <Container>
          <div className="flex flex-col
items-start gap-6 lg:flex-row lg:gap-12">
            <Aside />
            <div className="mb-20 w-
full lg:mb-4 lg:w-4/5">{children}</div>
          </div>
        </Container>
      </main>
    </div>
  );
}

```

login.jsx

```

import { useEffect } from 'react';
import { Head, Link, useForm } from
'@inertiajs/react';
import { GuestLayout } from
'@/layouts/guest-layout';
import { Button } from
'@/components/ui/button';
import { Checkbox } from
'@/components/ui/checkbox';
import { Label } from
'@/components/ui/label';
import { Input } from
'@/components/ui/input';
import { InputErrorMessage } from
'@/components/input-error-message';
import { AuthenticatedCard } from
'@/components/authenticated-card';

export default function Login({ status,
canResetPassword }) {
  const { data, setData, post, processing,
errors, reset } = useForm({
    email: "",
    password: "",
    remember: false,
  });

  useEffect(() => {
    return () => {
      reset('password');
    };
  }, []);

  const submit = (e) => {
    e.preventDefault();

    post(route('login'));
  };

  return (
    <
      <Head title="Log in" />

      {status && <div className="mb-4
text-sm font-medium text-green-
600">{status}</div>}
      <AuthenticatedCard title="Masuk ke
akun anda" description="Silahkan masuk ke
akun yang telah terdaftar">
        <form onSubmit={submit}
className="space-y-6">
          <div>

```

```

    <Label
htmlFor="email">Email</Label>

    <Input
      id="email"
      type="email"
      name="email"
      value={data.email}
      className="mt-1"
      autoComplete="username"
      isFocused={true}
      onChange={(e) =>
setData('email', e.target.value)}
    />

    <InputErrorMessage
message={errors.email} className="mt-2"
/>
  </div>

  <div>
    <Label
htmlFor="password">Kata Sandi</Label>

    <Input
      id="password"
      type="password"
      name="password"
      value={data.password}
      className="mt-1"
      autoComplete="current-
password"
      onChange={(e) =>
setData('password', e.target.value)}
    />

    <InputErrorMessage
message={errors.password}
className="mt-2" />
  </div>

  <div className="flex items-
center justify-between">
    <label className="flex
items-center">
      <Checkbox
        name="remember"

checked={data.remember}
        onChange={e =>
=> setData('remember', e)}
      />

```

```

      <span className="ms-2
text-sm">Ingat saya</span>
    </label>
    {canResetPassword && (
      <Link
        href={route('password.request')}
        className="text-sm
text-muted-foreground transition duration-
200 hover:text-foreground"
      >
        Lupa kata sandi?
      </Link>
    )}
  </div>

  <div className="flex items-
center justify-end">
    <Button className="ms-4"
disabled={processing}>
      Masuk
    </Button>
  </div>
</form>
</AuthenticatedCard>
</>
);
}

```

```

Login.layout = (page) => <GuestLayout
children={page} />;

```

register.jsx

```

import { useEffect } from 'react';
import { Head, Link, useForm } from
'@inertiajs/react';
import { InputErrorMessage } from
'@/components/input-error-message';
import { Label } from
'@/components/ui/label';
import { Button } from
'@/components/ui/button';
import { Input } from
'@/components/ui/input';
import { GuestLayout } from
'@/layouts/guest-layout';
import { AuthenticatedCard } from
'@/components/authenticated-card';

export default function Register() {
  const { data, setData, post, processing,
errors, reset } = useForm({
    name: "",

```

```

    email: "",
    password: "",
    password_confirmation: "",
  });

  useEffect(() => {
    return () => {
      reset('password',
'password_confirmation');
    };
  }, []);

  const submit = (e) => {
    e.preventDefault();

    post(route('register'));
  };

  const onChange = (event) => {
    setData(event.target.name,
event.target.value);
  };

  return (
    <
      <Head title="Register" />

      <AuthenticatedCard title="Create a
new account" description="Or log in to your
existing account">
        <form onSubmit={submit}
className="space-y-6">
          <div>
            <Label
htmlFor="name">Name</Label>

            <Input
              type="text"
              name="name"
              value={data.name}
              className="mt-1"
              autoComplete="name"
              autoFocus
              onChange={onChange}
              required
            />

            <InputErrorMessage
message={errors.name} className="mt-2"
            />
          </div>

          <div>

```

```

            <Label
htmlFor="email">Email</Label>

            <Input
              type="email"
              name="email"
              value={data.email}
              className="mt-1"
              autoComplete="username"
              onChange={onChange}
              required
            />

            <InputErrorMessage
message={errors.email} className="mt-2"
            />
          </div>

          <div>
            <Label
htmlFor="password">Password</Label
            >

            <Input
              type="password"
              name="password"
              value={data.password}
              className="mt-1"
              autoComplete="new-
password"
              onChange={onChange}
              required
            />

            <InputErrorMessage
message={errors.password}
className="mt-2" />
          </div>

          <div>
            <Label
htmlFor="password_confirmation">Confir
m Password</Label>

            <Input
              type="password"

              name="password_confirmation"

              value={data.password_confirmation}
              className="mt-1"
              onChange={onChange}

```

```

        required
      />

      <InputErrorMessage
        message={errors.password_confirmation}
        className="mt-2" />
    </div>

    <div className="flex items-
center justify-between">
      <Button asChild
        variant="outline">
        <Link
          href="/login">Login</Link>
        </Button>

        <Button type="submit"
          className="ml-4" disabled={processing}>
          Register
        </Button>
      </div>
    </form>
  </AuthenticatedCard>
</>
);
}

```

```

Register.layout = (page) => <GuestLayout
  children={page} />;

```

article.form.jsx

```

import { UserLayout } from '@layouts/user-
layout';
import { Card, CardContent,
CardDescription, CardFooter, CardHeader,
CardTitle } from '@components/ui/card';
import React, { useEffect } from 'react';
import { useForm } from '@inertiajs/react';
import { Button } from
 '@components/ui/button';
import { Select, SelectContent, SelectItem,
SelectTrigger, SelectValue } from
 '@components/ui/select';
import { Label } from
 '@components/ui/label';
import { Input } from
 '@components/ui/input';
import { MultiSelect } from
 '@components/multi-select';
import { InputErrorMessage } from
 '@components/input-error-message';
import { Textarea } from
 '@components/ui/textarea';

```

```

import { FileUpload } from
 '@components/file-upload';
import { Editor } from
 '@components/editor';
import { flashMessage } from '@lib/utls';
import { toast } from 'sonner';

export default function Form({ auth,
page_meta, page_data }) {
  const { data, setData, post, errors,
processing } = useForm({
    thumbnail: null,
    category_id:
page_data.article?.category_id ?? "",
    title: page_data.article?.title ?? "",
    teaser: page_data.article?.teaser ?? "",
    tags: [],
    content: page_data.article?.content ?? "",
    published_at:
page_data.article?.published_at ?? "",
    status: page_data.article?.status ?? "",
    _method: page_meta.method,
  });

  const [selected, setSelected] =
React.useState(
  page_data.article.tags?.reduce((unique,
tag) => {
    if (unique.findIndex((item) =>
item.value === tag.id) === -1) {
      unique.push({ value: tag.id, label:
tag.name });
    }
    return unique;
  }, []) ?? [],
);

  function submit(e) {
    e.preventDefault();
    post(page_meta.url, {
      preserveScroll: true,
      onSuccess: (params) => {
        const flash =
flashMessage(params);
        if (flash) {
          toast[flash.type](flash.message);
        }
      },
    });
  }

  function onChange(e) {
    setData(e.target.name, e.target.value);
  }
}

```

```

useEffect(() => {
  function transformTags() {
    return selected.map((item) =>
item.value);
  }

  setData('tags', transformTags());
}, [selected]);

return (
  <Card>
    <CardHeader>

<CardTitle>{page_meta.title}</CardTitle>

<CardDescription>{page_meta.description}
</CardDescription>
    </CardHeader>
    <form onSubmit={submit}>
      <CardContent>
        <div className="space-y-6">
          <div className="max-w-2xl
space-y-1">

<Label>Thumbnail</Label>
            <FileUpload
              onChange={(file) => setData('thumbnail',
file)} />
            <InputErrorMessage
              message={errors.thumbnail} />
            </div>
            <div className="space-y-
1">
              <Label>Nama
Masakan</Label>
              <Input name="title"
                onChange={onChange} value={data.title}
              />
              <InputErrorMessage
                message={errors.title} />
              </div>
              <div className="space-y-
1">
                <Label>Teaser</Label>
                <Textarea name="teaser"
                  onChange={onChange} value={data.teaser}
                />
                <InputErrorMessage
                  message={errors.teaser} />
                </div>
                <div className="space-y-
1">
                  <Label>Isi</Label>

```

```

      <Editor
        value={data.content} onChange={(value)
=> setData('content', value)} />
        <InputErrorMessage
          message={errors.content} />
      </div>
      <div className="grid grid-
cols-2 gap-6">
        <div className="space-y-
1">
          <Label>Daerah
Kuliner</Label>
          <Select
            defaultValue={data.category_id}
            onValueChange={(value) =>
              setData('category_id', value)}
            >
            <SelectTrigger
              className="w-full">
              <SelectValue
                placeholder="Pilih daerah kuliner">

                {page_data.categories.find((category) =>
category.value === data.category_id)
                  ?.label ??
                'Pilih status'}
              </SelectValue>
            </SelectTrigger>
            <SelectContent>

              {page_data.categories
                .sort((a, b) =>
                  a.label.localeCompare(b.label))
                .map((category)
=> (
                  <SelectItem
                    key={category.value}
                    value={category.value}>

                    {category.label}
                  </SelectItem>
                ))}
            </SelectContent>
          </Select>
          <InputErrorMessage
            message={errors.category_id} />
        </div>
        {auth.user.is_admin && (
          <div>
            <Label
              htmlFor="status">Status</Label>
            <Select

```

```

defaultValue={data.status}

onValueChange={(value) => setData('status',
value)}
      >
        <SelectTrigger
className="mt-1 capitalize">
          <SelectValue>

{page_data.statuses.find((status) =>
status.value === data.status)
              ?.label ??
'Pilih status'}
          </SelectValue>
        </SelectTrigger>
      </SelectContent>

{page_data.statuses.map((status) => (
  <SelectItem

key={status.value}

value={status.value}

className="capitalize"
      >

{status.label}
    </SelectItem>
  )
)}
    </SelectContent>
  </Select>

  <InputErrorMessage
message={errors.status} />
</div>
    )}
  </div>
  <div className="space-y-
1">
    <Label>Jenis
Kuliner</Label>
    <MultiSelect
max={3}
items={page_data.tags}
selected={selected}

setSelected={setSelected}
placeholder="Pilih jenis
kuliner..."
    />
    <InputErrorMessage
message={errors.tags} />
  </div>

```

```

    </div>
  </CardContent>
  <CardFooter>
    <Button disabled={processing}
type="submit">
      Simpan
    </Button>
  </CardFooter>
</form>
</Card>
  );
}

```

```

Form.layout = (page) => <UserLayout
title={page.props.page_meta.title}
children={page} />;

```

article.index.jsx

```

import { AppLayout } from '@layouts/app-
layout';
import { Container } from
'@/components/container';
import { Head } from '@inertiajs/react';
import { ArticleCard } from
'@/pages/articles/partials/article-card';
import { Pagination } from
'@/components/pagination';
import { MetaTags } from
'@/components/meta-tags';

export default function Index({ page_meta,
...props }) {
  const { data: articles, meta, links } =
props.articles;
  return (
    <
      <Head title={page_meta.title} />
      <MetaTags title={page_meta.title}
description={page_meta.description}
url={route('articles.index')} />
      <div className="-mt-6 border-b bg-
background py-6 sm:-mt-16 md:py-16">
        <Container>
          <h1 className="text-2xl font-
semibold leading-none tracking-tight
md:text-3xl">
            {page_meta.title}
          </h1>
          <p className="mt-4 text-lg
text-muted-
foreground">{page_meta.description}</p>
        </Container>
      </div>
    <
  )
}

```

```

<Container>
  {articles.length > 0 ? (
    <div className="grid gap-y-16
pt-8 sm:grid-cols-2 sm:gap-x-12 lg:grid-
cols-3 lg:py-24">
      {articles.map((article) => (
        <ArticleCard
key={article.id} article={article} />
      ))}
    </div>
  ) : (
    <p>Tidak ada masakan yang
ditemukan.</p>
  )}
</Container>
{meta.has_pages && (
  <div className="mt-24">
    <Pagination meta={meta}
links={links} />
  </div>
)}
</>
);
}

```

```

Index.layout = (page) => <AppLayout
children={page} />;

```

article.list.jsx

```

import { UserLayout } from '@layouts/user-
layout';
import { Table, TableBody, TableCell,
TableHead, TableHeader, TableRow } from
'@/components/ui/table';
import { Card, CardContent,
CardDescription, CardFooter, CardHeader,
CardTitle } from '@components/ui/card';
import { Select, SelectContent, SelectItem,
SelectTrigger, SelectValue } from
'@/components/ui/select';
import {
  DropdownMenu,
  DropdownMenuContent,
  DropdownMenuGroup,
  DropdownMenuItem,
  DropdownMenuLabel,
  DropdownMenuSeparator,
  DropdownMenuTrigger,
} from '@components/ui/dropdown-menu';
import {
  IconChevronLeft,
  IconChevronRight,

```

```

  IconCirclePlusFill,
  IconDotsVertical,
  IconOpenLink,
} from '@irsyadadl/paranoid';
import { Badge } from
'@/components/ui/badge';
import { Button } from
'@/components/ui/button';
import React, { useState } from 'react';
import { Link, router } from
'@inertiajs/react';
import { Input } from
'@/components/ui/input';
import { AlertAction } from
'@/components/alert-action';
import { useFilter } from '@hooks/use-
filter';
import { flashMessage } from '@lib/utlis';
import { toast } from 'sonner';

```

```

export default function List({ auth, ...props
}) {
  const { data: articles, meta, links } =
props.articles;
  const [params, setParams] =
useState(props.state);
  useFilter({
    route: route('internal-articles.index'),
    values: params,
    only: ['articles'],
  });

```

```

  return (
    <div>
      <div className="mb-6 grid gap-6
sm:grid-cols-3">
        <Card>
          <CardHeader>

<CardTitle>{meta.total}</CardTitle>
          <CardDescription>Total
Masakan</CardDescription>
          </CardHeader>
        </Card>
        <Card>
          <CardHeader>

<CardTitle>{meta.total_visits}</CardTitle>
          <CardDescription>Total
Kunjungan</CardDescription>
          </CardHeader>
        </Card>
        <Card>
          <CardHeader>

```

```

<CardTitle>{meta.unpublished_count}</CardTitle>
  <CardDescription>Masakan
  Yang Belum Diterbitkan</CardDescription>
  </CardHeader>
</Card>
</div>
<Card>
  <div className="flex flex-col
  justify-between gap-y-6 p-6 md:flex-row
  md:items-center md:gap-y-0">
    <CardHeader className="p-
    0">
      <CardTitle>Daftar
      Masakan</CardTitle>
      <CardDescription>{meta.total} masakan
      yang terdapat pada
      aplikasi</CardDescription>
      </CardHeader>
      <div className="flex max-w-
      md flex-col gap-2 md:flex-row">
        <Input
          value={params?.search}
          onChange={(e) =>
            setParams((prev) => ({ ...prev, search:
            e.target.value })))}
          placeholder="Cari..."
        />
        <div className="grid grid-
        cols-2 gap-x-2 md:flex">
          <Select
            value={params?.status}
            onValueChange={(e) => setParams({
            ...params, status: e })}>
            <SelectTrigger
              className="md:w-40">
                <SelectValue
                  placeholder={params?.status ?? 'Status'} />
                </SelectTrigger>
                <SelectContent>
                  <SelectItem
                    value="Arsipkan">Diarsipkan</SelectItem>
                  <SelectItem
                    value="Ditunda">Ditunda</SelectItem>
                  <SelectItem
                    value="Diterbitkan">Diterbitkan</SelectItem>
                </SelectContent>
              </Select>
            <Button asChild>
              <Link
                href={route('internal-articles.create')}>
                  <IconCirclePlusFill
                    className="mr-2 size-4" />
                    Tambah
                  </Link>
                </Button>
              </div>
            </div>
            <div>
              <CardContent className="p-0
              [&_td]:whitespace-nowrap [&_td]:px-6
              [&_th]:px-6 [&_thead]:border-t">
                <Table>
                  <TableHeader>
                    <TableRow>
                      <TableHead>#</TableHead>
                      <TableHead>Nama
                      Masakan</TableHead>
                      {auth.user.is_admin
                      && <TableHead>Penulis</TableHead>}
                    <TableHead>Status</TableHead>
                    <TableHead>Dibuat
                    pada</TableHead>
                    <TableHead>Diterbitkan
                    pada</TableHead>
                    <TableHead />
                  </TableRow>
                  </TableHeader>
                  <TableBody>
                    {articles.map((article,
                    index) => (
                      <TableRow
                        key={article.id}>
                          <TableCell>{meta.from +
                          index}</TableCell>
                          <TableCell>
                            <a
                              href={route('articles.show', [article])}
                              target="_blank">
                                {article.title}
                              </a>
                            <IconOpenLink className="ml-1 inline
                            size-3" />
                          </TableCell>
                          {auth.user.is_admin
                          &&
                          <TableCell>{article.user?.name}</TableCell>
                          >}
                          <TableCell>
                            <Badge
                              variant={article.status === 'Diterbitkan' ?
                              'default' : 'outline'}>
                                {article.status}

```


<pre> </Badge> </TableCell> <TableCell>{article.created_at}</TableCell> > <TableCell>{article.published_at}</TableCe ll> <TableCell> <div className="flex justify-end"> <DropdownMenu> <DropdownMenuTrigger> <IconDotsVertical className="size-4" /> </DropdownMenuTrigger> <DropdownMenuContent align="end" className="w-48"> <DropdownMenuLabel>ID Masakan: {article.id}</DropdownMenuLabel> <DropdownMenuSeparator /> <DropdownMenuItem asChild> <Link href={route('internal-articles.edit', [article])}> Edit </Link> </DropdownMenuItem> {auth.user.is_admin && (<DropdownMenuGroup> <AlertAction trigger={ <DropdownMenuItem onSelect={(e) => e.preventDefault()} > {article.status !== 'Diterbitkan' ? 'Diterbitkan' : 'Belum Diterbitkan'} } </DropdownMenuSeparator /> </pre>	<pre> </DropdownMenuItem> } title={ article.status !== 'Diterbitkan' ? 'Terbitkan Masakan' : 'Batalkan Penerbitan' } description={ article.status !== 'Diterbitkan' ? 'Apakah Anda yakin ingin mempublikasikan masakan ini?' : 'Apakah Anda yakin ingin membatalkan publikasi masakan ini?' } action={() => router.put(route('internal-articles.approve', [article]), {}, { preserveScroll: true, onSuccess: (params) => { const flash = flashMessage(params); if (flash) { toast[flash.type](flash.message); } }, },) } </DropdownMenuSeparator /> </pre>
---	---

```

<AlertAction
trigger={
  <DropdownMenuItem
    onSelect={(e) => e.preventDefault()}
    Hapus
  </DropdownMenuItem>
    title="Hapus Masakan"
    description="Apakah Anda yakin ingin
menghapus artikel ini?"
    action={() =>
      router.delete(
        route('internal-articles.destroy', [article]),
        {
          preserveScroll: true,
          onSuccess: (params) => {
            const flash = flashMessage(params);
            if (flash) {
              toast[flash.type](flash.message);
            }
          },
        },
      )
    }
  </DropdownMenuItem>
}
</DropdownMenuContent>
</DropdownMenu>
</div>
</TableCell>

```

```

</TableRow>
  )}
</TableBody>
</Table>
</CardContent>
  <CardFooter className="justify-
between border-t pt-6 text-sm text-muted-
foreground">
    <span>
      Menampilkan {meta.from}
      dari {meta.total} masakan.
    </span>
    {meta.has_pages && (
      <div className="flex items-
center gap-x-1">
        <Button asChild size="sm"
          variant="outline">
          {links.prev ? (
            <Link
              href={links.prev}>
              <IconChevronLeft
                className="-ml-1 mr-1 size-4" />
                Prev
              </Link>
            ) : (
              <span>Prev</span>
            )}
          </Button>
          <Button asChild size="sm"
            variant="outline">
            {links.next ? (
              <Link
                href={links.next}>
                Next
                <IconChevronRight
                  className="-mr-1 ml-1 size-4" />
                </Link>
              ) : (
                <span>Next</span>
              )}
            </Button>
          </div>
        )}
      </CardFooter>
    </Card>
  </div>
);
}

List.layout = (page) => <UserLayout
  title="Dashboard" children={page} />;

```

article.show.jsx

```

import { AppLayout } from '@layouts/app-
layout';
import { Container } from
'@/components/container';
import { Head, Link } from
'@inertiajs/react';
import { limitChars } from '@lib/utls';
import { AspectRatio } from
'@/components/ui/aspect-ratio';
import { Badge } from
'@/components/ui/badge';
import { RelatedArticles } from
'@/pages/articles/partials/related-articles';
import { Author } from
'@/pages/articles/partials/author';
import { TableOfContents } from
'@/pages/articles/partials/table-of-contents';
import { Prose } from
'@/pages/articles/partials/prose';
import { Share } from
'@/pages/articles/partials/share';
import { MetaTags } from
'@/components/meta-tags';

export default function Show(props) {
  const { article } = props;
  return (
    <
      <Head title={article.title} />
      <MetaTags
        title={article.title}
        description={article.teaser}
        url={route('articles.show',
[article])}
        image={article.thumbnail}
      />
      <Container>
        <div className="relative flex-
col-reverse items-start gap-y-16 lg:grid
lg:grid-cols-3 lg:gap-x-16">
          <div className="space-y-12
lg:sticky lg:top-10 lg:col-span-1">
            { /* <Author
user={article.user} /> */ }

            <TableOfContents
articleId={article.id} />

            <RelatedArticles />
          </div>
          <div className="space-y-6
lg:col-span-2">
            <AspectRatio
className="overflow-hidden rounded-lg
border" ratio={1.91}>

```

```

      <img
        className="grid h-full
w-full place-content-center object-cover
object-center text-center font-mono text-xs"
        src={article.thumbnail}

alt={limitChars(article.title)}
width={1200}
height={630}
      />
    </AspectRatio>

    <h1 className="text-2xl
font-semibold md:text-
4xl">{article.title}</h1>

    <div className="flex items-
center justify-between">
      <div className="flex
items-center text-sm text-muted-
foreground">
        <time>Masakan
Nusantara</time>
        <span className="mx-
2">|</span>
        <Link
href={route('categories.show',
[article.category])}>
          <Badge
variant="default">{article.category.name}</
Badge>
        </Link>
        <span className="mx-
2">|</span>
        {article.tags.length > 0 ?
(
          <div className="flex
items-center gap-x-2">
            {article.tags.map((tag, i) => (
              <Link key={i}
href={route('tags.show', [tag])}>
                <Badge
variant="outline">{tag.name}</Badge>
              </Link>
            ))}
          </div>
        ) : null}
      </div>
      <div className="flex">
        <Share article={article}
      />
    </div>
  </div>

```

```

      <p className="text-muted-foreground">{article.teaser}</p>

```

```

      <Prose
        content={article.content} />
      </div>
    </div>
  </Container>
</>
);
}

```

```

Show.layout = (page) => <AppLayout
  children={page} />;

```

categories.form.jsx

```

import { UserLayout } from '@layouts/user-layout';
import { Card, CardContent, CardDescription, CardFooter, CardHeader, CardTitle } from '@components/ui/card';
import { useForm } from '@inertiajs/react';
import { Button } from '@components/ui/button';
import { Input } from '@components/ui/input';
import { Label } from '@components/ui/label';
import { FileUpload } from '@components/file-upload';
import { InputErrorMessage } from '@components/input-error-message';
import { Textarea } from '@components/ui/textarea';

export default function Form(props) {
  const { data, setData, post, errors, processing } = useForm({
    thumbnail: null,
    name: props.category?.name ?? '',
    teaser: props.category?.teaser ?? '',
    _method: props.page_meta.method,
  });

  function submit(e) {
    e.preventDefault();
    post(props.page_meta.url);
  }

  function onChange(e) {
    setData(e.target.name, e.target.value);
  }

```

```

  return (
    <Card>
      <CardHeader>

        <CardTitle>{props.page_meta.title}</CardTitle>

        <CardDescription>{props.page_meta.description}</CardDescription>
      </CardHeader>
      <form onSubmit={submit}>
        <CardContent>
          <div className="space-y-6">
            <div className="max-w-2xl space-y-1">

              <Label>Thumbnail</Label>
              <FileUpload
                onChange={(file) => setData('thumbnail', file)} />

              <InputErrorMessage
                message={errors.thumbnail} />
            </div>
            <div className="space-y-1">

              <Label
                htmlFor="name">Nama Daerah</Label>
              <Input
                label="Name"
                name="name"
                id="name"
                value={data.name}
                onChange={(e) => setData('name', e.target.value)}
                required
                error={errors.name}
              />
              <InputErrorMessage
                message={errors.name} />
            </div>
            <div className="space-y-1">

              <Label>Teaser</Label>
              <Textarea name="teaser"
                onChange={onChange} value={data.teaser}
              />

              <InputErrorMessage
                message={errors.teaser} />
            </div>
          </div>
        </CardContent>
        <CardFooter>
          <Button
            disabled={processing}>Simpan</Button>
        </CardFooter>
      </form>
    </Card>
  );

```

```

    </form>
  </Card>
);
}

```

```

Form.layout = (page) => <UserLayout
title={page.props.page_meta.title}>{page}<
/UserLayout>;

```

categories.index.jsx

```

import { UserLayout } from '@layouts/user-
layout';
import { Card, CardContent,
CardDescription, CardFooter, CardHeader,
CardTitle } from '@components/ui/card';
import { Table, TableBody, TableCell,
TableHead, TableHeader, TableRow } from
'@components/ui/table';
import { IconChevronLeft,
IconChevronRight, IconCirclePlusFill,
IconDotsVertical } from
'@irsyadadl/paranoid';
import {
  DropdownMenu,
  DropdownMenuContent,
  DropdownMenuGroup,
  DropdownMenuItem,
  DropdownMenuLabel,
  DropdownMenuSeparator,
  DropdownMenuTrigger,
} from '@components/ui/dropdown-menu';
import { Link, router } from
'@inertiajs/react';
import { AlertAction } from
'@components/alert-action';
import { Button } from
'@components/ui/button';

export default function Index(props) {
  const { data: categories, meta, links } =
  props.categories;
  return (
    <Card>
      <div className="flex flex-col
justify-between gap-y-6 p-6 md:flex-row
md:items-center md:gap-y-0">
        <CardHeader className="p-0">
          <CardTitle>Daftar Daerah
Kuliner</CardTitle>
          <CardDescription>{meta.total}
daerah kuliner ditemukan pada
aplikasi</CardDescription>
        </CardHeader>

```

```

        <Button asChild>
          <Link
href={route('categories.create')}>
            <IconCirclePlusFill
className="mr-2 size-4" />
            Tambah
          </Link>
        </Button>
      </div>
      <CardContent className="p-0
[&_td]:whitespace-nowrap [&_td]:px-6
[&_th]:px-6 [&_thead]:border-t">
        <Table>
          <TableHeader>
            <TableRow>

              <TableHead>#</TableHead>

              <TableHead>Name</TableHead>

              <TableHead>Slug</TableHead>

              <TableHead>Masakan</TableHead>
            </TableHeader>
            <TableBody>
              {categories.map((category,
index) => (
                <TableRow
key={category.id}>
                  <TableCell>{meta.from
+ index}</TableCell>
                  <TableCell>
                    <Link
href={route('categories.show',
[category])}>{category.name}</Link>
                  </TableCell>

                  <TableCell>{category.slug}</TableCell>

                  <TableCell>{category.articles_count}</Tabl
eCell>

                  <TableCell>
                    <div className="flex
justify-end">
                      <DropdownMenu>

                        <DropdownMenuTrigger>

                        <IconDotsVertical className="size-4" />

                      </DropdownMenuTrigger>

```

```

<DropdownMenuContent align="end"
  className="w-48">

  <DropdownMenuLabel>Article ID:
    {category.id}</DropdownMenuLabel>

  <DropdownMenuSeparator />

  <DropdownMenuItem asChild>
    <Link
      href={route('categories.edit',
        [category])}>Edit</Link>

  </DropdownMenuItem>

  <DropdownMenuGroup>

  <AlertAction
    trigger={

  <DropdownMenuItem onSelect={(e) =>
    e.preventDefault()}>

    Hapus

  </DropdownMenuItem>
    }

    title="Hapus Daerah Kuliner"

    description="Apakah anda yakin ingin
      menghapus daerah kuliner ini?"

    action={() =>

      router.delete(route('categories.destroy',
        [category]), {

        preserveScroll: true,

      })
    }
  </DropdownMenuItem>

  </DropdownMenuGroup>

  </DropdownMenuContent>
    </DropdownMenu>
  </div>
  </TableCell>
  </TableRow>
  </tbody>
  </TableBody>
  </Table>
  </CardContent>

```

```

    <CardFooter className="justify-
      between border-t pt-6 text-sm text-muted-
      foreground">
      <span>
        Showing {meta.from} of
        {meta.total} categories.
      </span>
      {meta.has_pages && (
        <div className="flex items-
          center gap-x-1">
            <Button asChild size="sm"
              variant="outline">
              {links.prev ? (
                <Link
                  href={links.prev}>
                    <IconChevronLeft
                      className="-ml-1 mr-1 size-4" />
                    Prev
                  </Link>
                ) : (
                  <span>Prev</span>
                )}
            </Button>
            <Button asChild size="sm"
              variant="outline">
              {links.next ? (
                <Link
                  href={links.next}>
                    Next
                    <IconChevronRight
                      className="-mr-1 ml-1 size-4" />
                  </Link>
                ) : (
                  <span>Next</span>
                )}
            </Button>
          </div>
        )}
    </CardFooter>
  </Card>
  );
}

```

```

Index.layout = (page) => <UserLayout
  title="Daerah Kuliner" children={page} />;

```

tags.form.jsx

```

import { UserLayout } from '@layouts/user-
  layout';
import { Card, CardContent,
  CardDescription, CardFooter, CardHeader,
  CardTitle } from '@components/ui/card';
import { useForm } from '@inertiajs/react';

```

```

import { Button } from
'@/components/ui/button';
import { Input } from
'@/components/ui/input';
import { Label } from
'@/components/ui/label';
import { InputErrorMessage } from
'@/components/input-error-message';
import { FileUpload } from
'@/components/file-upload';
import { Textarea } from
'@/components/ui/textarea';

export default function Form(props) {
  const { data, setData, post, errors,
processing } = useForm({
    thumbnail: null,
    name: props.tag?.name ?? "",
    teaser: props.tag?.teaser ?? "",
    _method: props.page_meta.method,
  });

  function submit(e) {
    e.preventDefault();
    post(props.page_meta.url);
  }

  function onChange(e) {
    setData(e.target.name, e.target.value);
  }

  return (
    <Card>
      <CardHeader>
        <CardTitle>{props.page_meta.title}</CardTitle>

        <CardDescription>{props.page_meta.description}</CardDescription>
        </CardHeader>
        <form onSubmit={submit}>
          <CardContent>
            <div className="space-y-6">
              <div className="max-w-2xl
space-y-1">

                <Label>Thumbnail</Label>
                <FileUpload
onChange={file => setData('thumbnail',
file)} />

                <InputErrorMessage
message={errors.thumbnail} />
              </div>
            </div>
          </CardContent>
        </form>
      </CardHeader>
      <CardFooter>
        <Button
disabled={processing}>Save</Button>
      </CardFooter>
    </Card>
  );
}

Form.layout = (page) => <UserLayout
title={page.props.page_meta.title}>{page}<
/UserLayout>;

```

```

    <div className="space-y-
1">
      <Label
htmlFor="name">Name</Label>
      <Input
        label="Name"
        name="name"
        id="name"
        value={data.name}
        onChange={(e) =>
setData('name', e.target.value)}
        required
        error={errors.name}
      />
      <InputErrorMessage
message={errors.name} />
    </div>
    <div className="space-y-
1">
      <Label>Teaser</Label>
      <Textarea name="teaser"
onChange={onChange} value={data.teaser}
/>
      <InputErrorMessage
message={errors.teaser} />
    </div>
  </CardContent>
  <CardFooter>
    <Button
disabled={processing}>Save</Button>
  </CardFooter>
</form>
</Card>
);
}

```

```

Form.layout = (page) => <UserLayout
title={page.props.page_meta.title}>{page}<
/UserLayout>;

```

index.jsx

```

import { UserLayout } from '@layouts/user-
layout';
import { Card, CardContent,
CardDescription, CardFooter, CardHeader,
CardTitle } from '@components/ui/card';
import { Table, TableBody, TableCell,
TableHead, TableHeader, TableRow } from
'@/components/ui/table';
import { IconChevronLeft,
IconChevronRight, IconCirclePlusFill,

```

```

IconDotsVertical } from
'@irsyadadl/paranoid';
import {
  DropdownMenu,
  DropdownMenuContent,
  DropdownMenuGroup,
  DropdownMenuItem,
  DropdownMenuLabel,
  DropdownMenuSeparator,
  DropdownMenuTrigger,
} from '@components/ui/dropdown-menu';
import { Link, router } from
'@inertiajs/react';
import { AlertAction } from
'@components/alert-action';
import { Button } from
'@components/ui/button';

export default function Index(props) {
  const { data: tags, meta, links } =
  props.tags;
  return (
    <Card>
      <div className="flex flex-col
justify-between gap-y-6 p-6 md:flex-row
md:items-center md:gap-y-0">
        <CardHeader className="p-0">
          <CardTitle>Daftar Jenis
Kuliner</CardTitle>
          <CardDescription>{meta.total}
jenis kuliner ditemukan pada
aplikasi</CardDescription>
        </CardHeader>
        <Button asChild>
          <Link
href={route('tags.create')}>
            <IconCirclePlusFill
className="mr-2 size-4" />
            Tambah
          </Link>
        </Button>
      </div>
      <CardContent className="p-0
[&_td]:whitespace-nowrap [&_td]:px-6
[&_th]:px-6 [&_thead]:border-t">
        <Table>
          <TableHeader>
            <TableRow>

              <TableHead>#</TableHead>

              <TableHead>Name</TableHead>

              <TableHead>Slug</TableHead>

```

```

            <TableHead>Masakan</TableHead>
          </TableRow>
        </TableHeader>
        <TableBody>
          {tags.map((tag, index) => (
            <TableRow key={tag.id}>
              <TableCell>{meta.from
+ index}</TableCell>
              <TableCell>
                <Link
href={route('tags.show',
[tag])}>{tag.name}</Link>
              </TableCell>

              <TableCell>{tag.slug}</TableCell>

              <TableCell>{tag.articles_count}</TableCell>
            </TableRow>
          )}
        </TableBody>
      </Table>
    </div className="flex
justify-end">
      <DropdownMenu>
        <DropdownMenuTrigger>
          <IconDotsVertical className="size-4" />
        </DropdownMenuTrigger>
        <DropdownMenuContent align="end"
className="w-48">
          <DropdownMenuLabel>ID Masakan:
{tag.id}</DropdownMenuLabel>
          <DropdownMenuSeparator />
          <DropdownMenuItem asChild>
            <Link
href={route('tags.edit', [tag])}>Edit</Link>
          </DropdownMenuItem>
          <DropdownMenuGroup>
            <AlertAction
trigger={
              <DropdownMenuItem onSelect={e =>
e.preventDefault()}>
                Hapus

```



```

</DropdownMenuItem>
    }

    title="Hapus Jenis Kuliner"

    description="Apakah anda yakin ingin
    menghapus jenis kuliner ini?"

    action={() =>

    router.delete(route('tags.destroy', [tag]), {

    preserveScroll: true,

    })
  }
/>

</DropdownMenuGroup>

</DropdownMenuContent>
  </DropdownMenu>
</div>
  </TableCell>
</TableRow>
))}
</TableBody>
</Table>
</CardContent>
  <CardFooter className="justify-
between border-t pt-6 text-sm text-muted-
foreground">
    <span>
      Showing {meta.from} of
{meta.total} tags.
    </span>
    {meta.has_pages && (
      <div className="flex items-
center gap-x-1">
        <Button asChild size="sm"
variant="outline">
          {links.prev ? (
            <Link
href={links.prev}>
              <IconChevronLeft
className="-ml-1 mr-1 size-4" />
              Prev
            </Link>
          ) : (
            <span>Prev</span>
          )}
        </Button>
        <Button asChild size="sm"
variant="outline">
          {links.next ? (

```

```

      <Link
href={links.next}>
        Next
        <IconChevronRight
className="-mr-1 ml-1 size-4" />
      </Link>
    ) : (
      <span>Next</span>
    )}
  </Button>
</div>
  )}
</CardFooter>
</Card>
);
}

```

```

Index.layout = (page) => <UserLayout
title="Jenis Kuliner" children={page} />;

```

dashboard.jsx

```

import { Container } from
'@/components/container';
import { Head } from '@inertiajs/react';
import { UserLayout } from '@layouts/user-
layout';

export default function Dashboard() {
  return (
    <div>
      <Head title="Dashboard" />
      <Container>Dashboard
page</Container>
    </div>
  );
}

```

```

Dashboard.layout = (page) => <UserLayout
title="Dashboard" children={page} />;

```

home.jsx

```

import { AppLayout } from '@layouts/app-
layout';
import { Container } from
'@/components/container';
import { Head, Link } from
'@inertiajs/react';
import { ArticleCard } from
'@/pages/articles/partials/article-card';
import { CardDescription, CardHeader,
CardTitle } from '@components/ui/card';

```

```

import { MetaTags } from
'@/components/meta-tags';
import { NewLogo } from
'@/components/logo-new';

export default function Home(props) {
  const { articles, popular_articles } =
  props;
  return (
    <div>
      <Head title="Home" />
      <MetaTags
        title="Ensiklopedia Masakan
Nusantara"
        description="Temukan Warisan
Kuliner dalam Satu Aplikasi!"
        url={route('home')}
      />
      <Hero />
      <Container>
        <div className="space-y-16
sm:space-y-24">
          <section id="populer">
            <CardHeader
              className="mb-6 p-0">
              <CardTitle>Masakan
Populer</CardTitle>

            <CardDescription>Kumpulan masakan
terpopuler dari aplikasi
kami.</CardDescription>
            </CardHeader>
            {popular_articles.length > 0 ?
(
              <div className="grid gap-
x-12 gap-y-16 lg:grid-cols-3">

                {popular_articles.map((article) => (
                  <ArticleCard
                    key={article.id} article={article} />
                  ))}
                </div>
              ) : (
                <p>Tidak ada masakan
yang ditemukan</p>
              )}
            </section>
            <section id="latest">
              <CardHeader
                className="mb-6 p-0">
                <CardTitle>Arsip
Kuliner</CardTitle>

              <CardDescription>Kumpulan kuliner

```

```

nusantara dari aplikasi
kami</CardDescription>
            </CardHeader>
            {articles.length > 0 ? (
              <div className="grid gap-
x-12 gap-y-16 lg:grid-cols-3">
                {articles.map((article)
=> (
                  <ArticleCard
                    key={article.id} article={article} />
                  ))}
                </div>
              ) : (
                <p>Tidak ada masakan
yang ditemukan</p>
              )}
            </section>
          </div>
        </Container>
      </div>
    );
  }

  function Hero() {
    return (
      <div className="relative isolate -mt-6
mb-6 overflow-hidden border-b bg-
background lg:-mt-16 lg:mb-16">
        <div
          className="absolute inset-0 -z-10
h-full w-full bg-cover stroke-foreground/10
[mask-image:radial-
gradient(100%_100%_at_top_right,white,tra
nsparent)]"
          aria-hidden="true"
        >
          
        </div>
        <div className="relative mx-auto
max-w-7xl items-center px-6 pb-16 sm:pt-6
md:grid md:h-[calc(100vh-74px)] md:items-
center md:pt-0">
          <div className="mx-auto max-
w-2xl flex-shrink-0 lg:mx-0 lg:max-w-2xl
lg:pt-8">
            <div className="hidden
md:block">
              <Link
                href={route('articles.index')}
                className="inline-flex space-x-6">
                <span
                  className="rounded-full bg-sky-500/10 px-

```

```

3 py-1 text-sm font-semibold leading-6 text-
sky-400 ring-1 ring-inset ring-sky-500/20">
    Apa yang baru ?
  </span>
</Link>
</div>
<h1 className="mt-10 flex
gap-10 text-4xl font-bold tracking-tight text-
foreground sm:text-5xl/[3.5rem] md:text-6xl
lg:text-7xl">
  Arsip Kuliner
</h1>
<p className="mt-6 text-base
leading-relaxed text-muted-foreground
sm:text-xl/[2rem] xl:text-2xl/[2.3rem]">
  <strong>Ensiklopedia
Masakan Nusantara:</strong> Jelajahi
Kelezatan Nusantara di Ujung Jari
  Anda. Temukan Warisan
Kuliner dalam Satu Aplikasi!
</p>
</div>
</div>
</div>
);
}

```

```

Home.layout = (page) => <AppLayout
children={page} />;

```

app.jsx

```

import './bootstrap';
import './css/app.css';

import { hydrateRoot } from 'react-
dom/client';
import { createInertiaApp } from
'@inertiajs/react';
import { resolvePageComponent } from
'laravel-vite-plugin/inertia-helpers';
import { ThemeProvider } from
'@/components/theme-provider';

const appName =
import.meta.env.VITE_APP_NAME ||
'ArsipKuliner';

createInertiaApp({
  title: (title) => `${title} - ${appName}`,
  resolve: (name) =>
resolvePageComponent(`./pages/${name}.js
x`, import.meta.glob('./pages/**/*.jsx')),
  setup({ el, App, props }) {

```

```

const root = hydrateRoot(el);

root.render(
  <ThemeProvider>
    <App {...props} />
  </ThemeProvider>,
);
},
progress: {
  color: '#4B5563',
},
});

```

app.blade.php

```

<!DOCTYPE html>
<html class='dark scroll-smooth' lang="{{
str_replace('_', '-', app()->getLocale()) }}">

<head>
  <meta charset="utf-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1">
  <link rel="apple-touch-icon"
href="/apple-touch-icon.png">
  <link rel="icon" href="/favicon-
32x32.png" type="image/png"
sizes="32x32">
  <link rel="icon" href="/favicon-
16x16.png" type="image/png"
sizes="16x16">
  <link rel="manifest"
href="/site.webmanifest">
  <meta name="theme-color"
content="#FFFFFF" media="(prefers-color-
scheme: light)">
  <meta name="theme-color"
content="#000000" media="(prefers-color-
scheme: dark)">

  <title inertia>{{ config('app.name',
'Laravel') }}</title>

  <!-- Fonts -->
  <link rel="preconnect"
href="https://fonts.bunny.net">
  <link
href="https://fonts.bunny.net/css?family=figt
ree:400,500,600&display=swap"
rel="stylesheet" />

  <!-- Scripts -->
  @routes

```

```

    @viteReactRefresh
    @vite(['resources/js/app.jsx',
"resources/js/pages/{ $page['component'] }.js
x"])
    @inertiaHead
    <script src="/theme.js"></script>
    <script src="/tab.js"></script>
</head>

<body class="font-walsheim antialiased">
    @inertia
</body>

</html>

```

welcome.blade.php

```

<!DOCTYPE html>
<html class='dark scroll-smooth' lang="{{
str_replace('_', '-', app()->getLocale()) }}">

<head>
    <meta charset="utf-8">
    <meta name="viewport"
content="width=device-width, initial-
scale=1">
    <link rel="apple-touch-icon"
href="/apple-touch-icon.png">
    <link rel="icon" href="/favicon-
32x32.png" type="image/png"
sizes="32x32">
    <link rel="icon" href="/favicon-
16x16.png" type="image/png"
sizes="16x16">
    <link rel="manifest"
href="/site.webmanifest">
    <meta name="theme-color"
content="#FFFFFF" media="(prefers-color-
scheme: light)">
    <meta name="theme-color"
content="#000000" media="(prefers-color-
scheme: dark)">

    <title inertia>{{ config('app.name',
'Laravel') }}</title>

    <!-- Fonts -->
    <link rel="preconnect"
href="https://fonts.bunny.net">
    <link
href="https://fonts.bunny.net/css?family=figt
ree:400,500,600&display=swap"
rel="stylesheet" />

```

```

<!-- Scripts -->
@routes
@viteReactRefresh
@vite(['resources/js/app.jsx',
"resources/js/pages/{ $page['component'] }.js
x"])
    @inertiaHead
    <script src="/theme.js"></script>
    <script src="/tab.js"></script>
</head>

<body class="font-walsheim antialiased">
    @inertia
</body>

</html>

```

Lampiran 2 Kartu Monitoring Proposal



KARTU MONITORING BIMBINGAN
 MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
 FAKULTAS TEKNIK
 UNIVERSITAS MUHAMMADIYAH PAREPARE

PROPOSAL

Mahasiswa : RAISUL AMIN R	Pembimbing I : Ade Hastuty, ST, S.Kom, MT
NIM : 220280042	Pembimbing II : Wahyuddin, S.Kom, M.Kom
Judul Skripsi : RANCANG BANGUN ENSIKLOPEDIA DIGITAL MASAKAN NUSANTARA MENGGUNAKAN TEKNOLOGI PROGRESSIVE WEB APPS	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 Bab 1	A	Konsultasi 1 Perbaiki Dokumen Ikuti pedoman	A
Konsultasi 2 Bab 2 Kerangka Fikir	A	Konsultasi 2 Desain sistem Cari bukuanya	A
Konsultasi 3 Bab 3 Desain logika	A	Konsultasi 3 Acc	A
Konsultasi 4 Acc	A	Konsultasi 4	
Konsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik

Lampiran 3 Kartu Monitoring Hasil



KARTU MONITORING BIMBINGAN
 MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
 FAKULTAS TEKNIK
 UNIVERSITAS MUHAMMADIYAH PAREPARE

HASIL

Mahasiswa : RAISUL AMIN R	Pembimbing I : Ade Hastuty, ST., S.Kom., MT.
NIM : 220250042	Pembimbing II : Wahyuddin, S.Kom., M.Kom
Judul Skripsi : RANCANG BANGUN ENSIKLOPEDIA DIGITAL MASAKAN NUSANTARA MENGGUNAKAN TEKNOLOGI PROGRESSIVE WEB APPS	

ARAHAN PEMBIMBING I	HASRATOL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HASRATOL & PARAF PEMBIMBING
Konsultasi 1 - Abstrak - Hasil Pengujian.	A	Konsultasi 1 - Gambar desain di perbesar	f
Konsultasi 2 Acc Ujian hasil	13/5/24 A	Konsultasi 2 - use case diagram - Tabel use case - Activity Diagram	f
Konsultasi 3		Konsultasi 3 - Abstrak - Appendix	f
Konsultasi 4		Konsultasi 4 Acc	f
Konsultasi 5		Konsultasi 5	

Lampiran ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa saat konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib diampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal skripsi
4. Kartu ini ditemak di atas kertas karton berwarna hijau muda dan dicetak lembar dua

Lampiran 4 Kartu Monitoring Skripsi



KARTU MONITORING BIMBINGAN
 MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
 FAKULTAS TEKNIK
 UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI

Mahasiswa : RAISUL AMIN R	Pembimbing I : Ade Hastuty, S.T., S.Kom, M.T
NIM : 220280042	Pembimbing II : Wahyuddin, S.Kom, M.Kom.
Judul Skripsi : RANCANG BANGUN ENSIKLOPEDIA DIGITAL MASAKAN NUSANTARA MENGGUNAKAN TEKNOLOGI PROGRESSIVE WEB APPS	

ARAHAN PEMBIMBING I	HASIL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HASIL & PARAF PEMBIMBING
Konsultasi 1 Revisi sesuai uraian hasil	Ar	Konsultasi 1 - Use case diagram - Activity diagram	b
Konsultasi 2 Siapkan artikel ut skripsi ma	A	Konsultasi 2 - Sequence diagram - Abstrak	b
Konsultasi 3 Lanjut Uraian Tutup	Ap	Konsultasi 3 - Katab ating & wiringkan - Tabel Usecase	b
Konsultasi 4		Konsultasi 4 - Kesimpulan simpkronkan dgn Abstrak	G
Konsultasi 5		Konsultasi 5 Ace	b

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa setiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib diserahkan pada kaprodi skripsi dan menjadi salah satu persyaratan untuk ikut seminar pengabdian skripsi
4. Kartu ini disimpan di atas kertas karton berwarna hijau muda dan dilubak tengah bali