

LAPORAN PRAKTIKUM
MATA KULIAH PEMROGRAMAN WEB LANJUT

Dosen Pengampu : Dimas Wahyu Wibowo, ST., MT.

**JOBSHEET 5: BLADE VIEW, WEB TEMPLATING (ADMIN LTE),
DATATABLES**



Nama : Yonanda Mayla Rusdiaty

NIM : 2341760184

Prodi : D-IV Sistem Informasi Bisnis

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

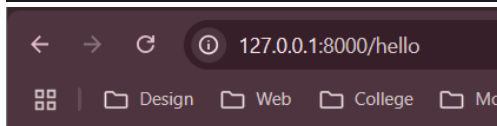
2024

A. VIEW

Karena Laravel menggunakan templating engine bawaan Blade, maka setiap file View diakhiri dengan `.blade.php`. Misal: `index.blade.php`, `home.blade.php`, `product.blade.php`. Contoh sederhana dari view dengan nama file `hello.blade.php` adalah sebagai berikut.

```
resources > views > hello.blade.php > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>View</title>
8 </head>
9 <body>
10  <h1>Hello, {{$name}}</h1>
11 </body>
12 </html>

Route::get(uri: '/hello', action: function(): Factory|View{
  return view(view: 'hello', data: ['name' => 'Andi']);
});
```



Hello, Andi

Setiap kita membuat view, kita mungkin akan menampilkannya melalui router atau controller menggunakan global view helper. Selain menggunakan global view helper kita juga dapat menggunakan view facade untuk menampilkan view

```
use Illuminate\Support\Facades\view;
return view::make(view: 'hello', data: ['name' => 'Andi']);
```

1. View di dalam direktori

Jika di dalam direktori resources/views terdapat direktori lagi untuk menyimpan file view, sebagai contoh hello.blade.php ada di dalam direktori blog, maka kita bisa menggunakan “dot” notation untuk mereferensikan direktori, sehingga syntax dalam route akan seperti berikut

```
Route::get(uri: '/hello', action: function(): Factory|View{
    return view (view: 'blog.hello', data: ['name' => 'Andi']);
});
```

2. Menampilkan view dari controller

View dapat dipanggil melalui Controller. Sehingga Routing akan memanggil Controller terlebih dahulu, dan Controller akan me-return view yang dimaksud

```
Route::get(uri: '/hello', action: [WelcomeController::class, 'hello']);
```

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
2 references | 0 implementations
class WelcomeController extends Controller
{
    1 reference | 0 overrides
    public function hello(): Factory|View
    {
        return view(view: 'blog.hello', data: ['name' => 'Andi']);
    }
}
```

3. Meneruskan data ke view

Seperti yang anda lihat di contoh sebelumnya, anda dapat meneruskan data array ke view agar data tersebut tersedia untuk view

```
return view(view: 'blog.hello', data: ['name' => 'Andi']);
```

Saat meneruskan informasi dengan cara ini, data harus berupa array dengan pasangan kunci / nilai. Setelah memberikan data ke view, anda kemudian dapat mengakses setiap nilai dalam view menggunakan kunci data seperti: <?php echo \$name; ?> atau {{ \$name }}.

Sebagai alternatif untuk meneruskan array data lengkap ke fungsi view helper, anda dapat menggunakan metode with untuk menambahkan bagian data individual ke view. Metode with mengembalikan instance view objek sehingga anda dapat melanjutkan rangkaian metode sebelum mengembalikan tampilan

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
2 references | 0 implementations
class WelcomeController extends Controller
{
    1 reference | 0 overrides
    public function hello(): View
    {
        return view(view: 'blog.hello')
            ->with(key: 'name', value: 'Andi')
            ->with(key: 'occupation', value: 'Programmer');
    }
}
```

```
<!DOCTYPE html>
<html>
<head>
    <title>Hello</title>
</head>
<body>
    <h1>Hello, {{ $name }}!</h1>
    <h1>Your occupation is: {{ $occupation }}</h1>
</body>
</html>
```

← → ↻ ⓘ 127.0.0.1:8000/hello

🗑️ | 📁 Design 📁 Web 📁 College 📁 Mobile

Hello, Andi!

Your occupation is: Programmer

4. Berbagi data dengan semua view

Terkadang, anda mungkin perlu berbagi data dengan semua tampilan yang dirender oleh aplikasi anda. anda dapat melakukannya dengan menggunakan metode berbagi tampilan fasad. Biasanya, anda harus melakukan panggilan ke

share methodi dalam metode boot penyedia layanan. anda bebas menambahkannya ke kelas App\Providers\AppServiceProvider atau membuat penyedia layanan terpisah untuk menampungnya

```
app > Providers > AppServiceProvider.php > ...
1  <?php
2
3  namespace App\Providers;
4
5  use Illuminate\Support\ServiceProvider;
6  use Illuminate\Support\Facades\View;
7
8  1 reference | 0 implementations
9  class AppServiceProvider extends ServiceProvider
10 {
11     /**
12      * Register any application services.
13      * @return void
14      */
15     public function register(): void
16     {
17         //
18     }
19
20     /**
21      * Bootstrap any application services.
22      * @return void
23      */
24     public function boot(): void
25     {
26         View::share(key: 'name', value: 'Andi');
27     }
28 }
```

B. BLADE LAYOUT, SECTION, DAN COMPONENT

Dalam membuat suatu tampilan, seringkali ada beberapa bagian yang sama dan selalu ditampilkan di setiap halaman view. Bagian-bagian ini dapat dibuat template sehingga tidak perlu dibuat berulang kali di setiap halaman view.

1) Layout dan Section

Berikut adalah contoh dari sebuah layout dari master/parent view.

```
resources > views > layouts > app.blade.php > html
1  <html>
2  <head>
3      <title>Halaman @yield(section: 'title')</title>
4  </head>
5  <body>
6      @section(section: 'sidebar')
7          Ini adalah master sidebar.
8      @show
9
10     <div class="container">
11         @yield(section: 'content')
12     </div>
13 </body>
14 </html>
```

Kode yang merupakan layout global, mendefinisikan title, sidebar, dan content yang dapat diisi oleh child view yang menggunakannya.

Berikut adalah contoh child view yang menggunakan layout app.blade.php.

```
resources > views > layouts > child.blade.php > ...
1  @extends('layouts.app')
2
3  @section('title', 'Profil')
4
5  @section('sidebar')
6      @parent
7
8      <p>Sidebar halaman Profil.</p>
9  @endsection
10
11 @section('content')
12     <p>Ini adalah bagian konten. NIM - Nama</p>
13 @endsection
```

Keterangan: @extends(“layouts.app”) digunakan untuk menjadikan file view app.blade.php. sebagai master view. Kemudian kita isi section title dengan “Profil” yang akan dirender sebagai <title> Halaman Profil </title>. Kemudian kita juga mengisi section sidebar menggunakan directive @parent. Sehingga ditampilkan konten sidebar parent/master serta menampilkan kalimat “Sidebar halaman Profil”. Setelah itu kita juga isi section “content” dengan tulisan “Ini adalah bagian konten. NIM - Nama”.

2) Component

Component berfungsi untuk membuat view yang dapat kita gunakan berulang kali. Berbeda dengan layout yang bertindak sebagai master, component dapat dianggap sebagai child view yang bisa kita pakai di view lain yang membutuhkannya. Misalnya dalam pengembangan sebuah aplikasi kita akan membutuhkan view untuk alert yang memberikan notifikasi kepada pengguna aplikasi terkait informasi, peringatan ataupun pesan error. Alert ini akan digunakan berulang kali di aplikasi. Oleh karena itu kita bisa membuatnya sebagai component yang bisa digunakan di view lainnya. Berikut adalah contoh component untuk membuat alert

```
resources > views > components > alert.blade.php > ...
1 <div class="alert alert-danger">
2     {!! $slot !!}
3 </div>
4 @extends(view: 'layouts.app')
5 {{-- kode --}}
6 @component("alert")
7     <b>Tulisan ini akan mengisi variable $slot</b>
8 @endcomponent
```

C. BOOTSTRAP PADA LARAVEL

Secara default, Laravel menggunakan NPM untuk menginstal paket frontend ini. Untuk menggunakannya, dapat dilakukan langkah-langkah berikut:

- 1) Instalasi Node.js
- 2) Memeriksa instalasi Node.js dan NPM

```
C:\Users\YONANDA MAYLA>node -v
v20.11.0
```

```
C:\Users\YONANDA MAYLA>npm -v
10.2.4
```

- 3) Kemudian melalui command prompt, ubah direktori ke dalam direktori project Laravel yang telah dibuat sebelumnya.

```
D:\>cd \Apps\laragon\www\PWL2025\week 5\jobsheet
```

```
D:\Apps\laragon\www\PWL2025\week 5\jobsheet>|
```

- 4) Scaffolding Bootstrap yang disediakan oleh Laravel terletak di dalam paket Composer, yang dapat diinstal menggunakan Composer:laravel/ui.

composer require laravel/ui

```
D:\Apps\laragon\www\PWL2025\week 5\jobsheet>composer require laravel/ui
./composer.json has been updated
Running composer update laravel/ui
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
- Locking laravel/ui (v4.6.1)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
- Downloading laravel/ui (v4.6.1)
- Installing laravel/ui (v4.6.1): Extracting archive
Generating optimized autoload files
```

```

Installing additional autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

[INFO] Discovering packages.

laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
laravel/ui ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

16 packages you are using are looking for funding
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

[INFO] No publishable resources for tag [laravel-assets].

[WARNING] security vulnerability advisory affecting 1 package
Run "composer audit" for a full list of advisories.
Using version "4.6" for laravel/ui

```

- 5) Setelah paket diinstal, kita dapat menginstal scaffolding frontend menggunakan perintah Artisan:laravel/ui

```

The [Controller.php] file already exists. Do you want to replace it? (yes/no) [yes]
> n

[INFO] Authentication scaffolding generated successfully.

[INFO] Bootstrap scaffolding installed successfully.

[WARN] Please run [npm install && npm run dev] to compile your fresh scaffolding.

```

- 6) Sebelum mengompilasi CSS, install dependensi frontend proyek Anda menggunakan Node package manager (NPM) :

```

D:\Apps\laragon\www\PWL2025\week 5\jobsheet>npm install

added 51 packages, and audited 52 packages in 3m

16 packages are looking for funding
  run `npm fund` for details

3 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

```

- 7) Setelah paket diinstal, Anda dapat menggunakan perintah berikut untuk mengompilasi aset Anda

```

D:\Apps\laragon\www\PWL2025\week 5\jobsheet>npm run dev

> dev
> vite

VITE v4.5.9 ready in 349 ms

→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h to show help

LARAVEL v10.48.28 plugin v0.7.8

→ APP_URL: http://localhost

```


D. MENGGUNAKAN TEMPLATE BOOTSTRAP DI LARAVEL

Sebenarnya pada project laravel, sudah ada file css bootstrap secara default pada pertama kali kita menginstall laravel. Letaknya ada pada file app.css dalam folder css. Anda dapat langsung menggunakannya dengan menghubungkan file app.css tersebut dengan file view anda

Untuk menghubungkan file css ke laravel anda dapat menggunakan syntax berikut:

```
resources > css > app.css
You, 24 seconds ago | 1 author (You)
1 <link rel="stylesheet" type="text/css" href="/css/style.css">
2
```

Dan file JSnya:

```
resources > js > app.js
You, 1 second ago | 1 author (You)
1 import './bootstrap';
2 <script type="text/javascript" src="/js/app.js"></script>
3
```

PRAKTIKUM 1- INTEGRASI LARAVEL DENGAN ADMIN LTE 3

1. Dalam root folder project lakukan command berikut, untuk mendefinisikan requirement project. composer require jeroennoten/laravel-adminlte

```
D:\Apps\laragon\www\PWL2025\week 5\jobsheet>composer require jeroennoten/laravel-adminlte
./composer.json has been updated
Running composer update jeroennoten/laravel-adminlte
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 2 installs, 0 updates, 0 removals
  - Downloading almasaeed2018/adminlte (v3.2.0)
  - Installing almasaeed2018/adminlte (v3.2.0): Extracting archive
  - Installing jeroennoten/laravel-adminlte (v3.15.0): Extracting archive
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

 INFO  Discovering packages.

jeroennoten/laravel-adminlte ..... DONE
laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
laravel/ui ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

$! packages you are using are looking for funding.
Use the 'composer fund' command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

 INFO  No publishable resources for tag [laravel-assets].

Found 1 security vulnerability advisory affecting 1 package.
Run "composer audit" for a full list of advisories.
Using version "3.15" for jeroennoten/laravel-adminlte
```

2. Melakukan instalasi requirement project di atas dengan command berikut: php artisan adminlte:install

```
D:\Apps\laragon\www\PWL2025\week 5\jobsheet>php artisan adminlte:install
AdminLTE assets files published successfully
Configuration file published successfully
Translation files published successfully
The installation is complete.
```

3. Buat file resources/views/layout/app.blade.php. Isi dengan kode berikut.

```

resources > views > layouts > app.blade.php > ...
You, 58 seconds ago | 1 author (You)
1  @extends(view: 'adminlte:page')
2  {{-- Extend and customize the browser title --}}
3  @section(section: 'title')
4  {{ config(key: 'adminlte.title') }}
5  @hasSection(name: 'subtitle') | @yield(section: 'subtitle') @endif
6  @stop
7  {{-- Extend and customize the page content header --}}
8  @section(section: 'content_header')
9  @hasSection(name: 'content_header_title')
10     <h1 class="text-muted">
11         @yield(section: 'content_header_title')
12         @hasSection(name: 'content_header_subtitle')
13             <small class="text-dark">
14                 <i class="fas fa-xs fa-angle-right text-muted"></i>
15                 @yield(section: 'content_header_subtitle')
16             </small>
17         @endif
18     </h1>
19 @endif
20 @stop
21 {{-- Rename section content to content_body --}}
22 @section(section: 'content')
23 @yield(section: 'content_body')
24 @stop
25 {{-- Create a common footer --}}
26 @section(section: 'footer')
27 <div class="float-right">
28     Version: {{ config(key: 'app.version', default: '1.0.0') }}
29 </div>
30 <strong>

```

4. Edit resources/views/welcome.blade.php, kemudian replace seluruh kodenya dengan kode berikut

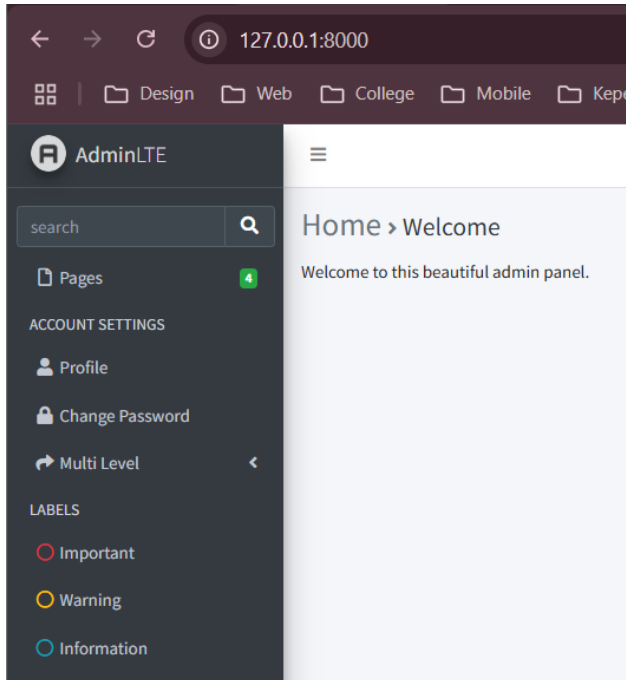
```

resources > views > welcome.blade.php > ...
You, 1 minute ago | 1 author (You)
1  @extends(view: 'layouts.app')
2
3  {{-- Customize layout sections --}}
4
5  @section(section: 'subtitle', content: 'Welcome')
6  @section(section: 'content_header_title', content: 'Home')
7  @section(section: 'content_header_subtitle', content: 'Welcome')
8
9  {{-- Content body: main page content --}}
10
11 @section(section: 'content_body')
12     <p>Welcome to this beautiful admin panel.</p>
13 @stop
14
15 {{-- Push extra CSS --}}
16
17 @push('css')
18     {{-- Add here extra stylesheets --}}
19     {{-- <link rel="stylesheet" href="/css/admin_custom.css"> --}}
20 @endpush
21
22 {{-- Push extra scripts --}}
23 You, 24 hours ago • feat: View js 5
24 @push('js')
25     <script>console.log("Hi, I'm using the Laravel-AdminLTE package!");</script>
26 @endpush

```

```
Route::get(uri: '/', action: function (): Factory|View {  
    return view(view: 'welcome');  
});
```

5. Kembali ke browser, menuju ke halaman awal



PRAKTIKUM 2 – INTEGRASI DENGAN DATA TABLES

Datatables biasa digunakan untuk menampilkan list data dengan banyak fitur seperti searching, pagination, sorting dan lain-lainnya. Dalam laravel biasa digunakan yajra datatables karena didesain agar sesuai dengan Laravel

1. Install Laravel DataTables

```
D:\Apps\laragon\www\PWL2025\week 5\jobsheet>composer require laravel/ui --dev
laravel/ui is currently present in the require key and you ran the command with the --dev flag, which will move it to the require-dev key.
Do you want to move this requirement? [no]? yes
./composer.json has been updated
Running composer update laravel/ui
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Writing lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

 INFO  Discovering packages.

jeroennoten/laravel-adminlte ..... DONE
laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
laravel/ui ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

$! packages you are using are looking for funding.
Use the 'composer fund' command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

 INFO  No publishable resources for tag [laravel-assets].

Found 1 security vulnerability advisory affecting 1 package.
Run "composer audit" for a full list of advisories.
Using version "4.0" for laravel/ui
```

```
D:\Apps\laragon\www\PWL2025\week 5\jobsheet>composer require yajra/laravel-datatables:*10.0
The "10.0" constraint for "yajra/laravel-datatables" appears too strict and will likely not match what you want. See https://getcomposer.org/constraints
./composer.json has been updated
Running composer update yajra/laravel-datatables
Loading composer repositories with package information
Updating dependencies
Lock file operations: 7 installs, 0 updates, 0 removals
- Locking league/fractal (0.20.2)
- Locking yajra/laravel-datatables (v10.0.0)
- Locking yajra/laravel-datatables-buttons (v10.0.9)
- Locking yajra/laravel-datatables-editor (v1.25.4)
- Locking yajra/laravel-datatables-fractal (v10.0.0)
- Locking yajra/laravel-datatables-html (v10.12.0)
- Locking yajra/laravel-datatables-oracle (v10.11.4)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 7 installs, 0 updates, 0 removals
- Downloading yajra/laravel-datatables-oracle (v10.11.4)
- Downloading yajra/laravel-datatables-html (v10.12.0)
- Downloading league/fractal (0.20.2)
- Downloading yajra/laravel-datatables-fractal (v10.0.0)
- Downloading yajra/laravel-datatables-editor (v1.25.4)
- Downloading yajra/laravel-datatables-buttons (v10.0.9)
- Downloading yajra/laravel-datatables (v10.0.0)
- Installing yajra/laravel-datatables-oracle (v10.11.4): Extracting archive
- Installing yajra/laravel-datatables-html (v10.12.0): Extracting archive
- Installing league/fractal (0.20.2): Extracting archive
- Installing yajra/laravel-datatables-fractal (v10.0.0): Extracting archive
- Installing yajra/laravel-datatables-editor (v1.25.4): Extracting archive
- Installing yajra/laravel-datatables-buttons (v10.0.9): Extracting archive
- Installing yajra/laravel-datatables (v10.0.0): Extracting archive
$ package suggestions were added by new dependencies, use 'composer suggest' to see details.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

 INFO  Discovering packages.
```

2. Pastikan nodejs sudah terinstall, dengan perintah `npm -v`. Npm digunakan untuk manajemen package javascript.

```
D:\Apps\laragon\www\PWL2025\week 5\jobsheet>npm -v
10.2.4
```

3. Install Laravel DataTables Vite dan sass

```
D:\Apps\laragon\www\PWL2025\week 5\jobsheet>npm i laravel-datatables-vite --save-dev
added 9 packages, and audited 61 packages in 41s

17 packages are looking for funding
  run 'npm fund' for details

3 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run 'npm audit' for details.
```

```
D:\Apps\laragon\www\PWL2025\week 5\jobsheet>npm install -D sass
up to date, audited 61 packages in 2s

17 packages are looking for funding
  run 'npm fund' for details

3 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run 'npm audit' for details.
```

4. Edit file resources/js/app.js

```
resources > js > app.js
You, 1 second ago | 1 author (You)
1  import './bootstrap';
2  import './sass/app.scss';
3  import 'laravel-datatables-vite';
4
```

5. Buatlah file resources/sass/app.scss

```
resources > sass > app.scss
You, 4 seconds ago | 1 author (You)
1  // Fonts
2  @import url('https://fonts.bunny.net/css?family=Nunito');
3
4  // Variables
5  @import 'variables';
6
7  // Bootstrap
8  @import 'bootstrap/scss/bootstrap';
9
10 // DataTables
11 @import "bootstrap-icons/font/bootstrap-icons.css";
12 @import "datatables.net-bs5/css/dataTables.bootstrap5.min.css";
13 @import "datatables.net-buttons-bs5/css/buttons.bootstrap5.min.css";
14 @import "datatables.net-select-bs5/css/select.bootstrap5.css";
15
```

6. Jalankan dengan npm run dev

```
D:\Apps\laragon\www\PWL2025\week 5\jobsheet>npm run dev

> dev
> vite

VITE v4.5.9 ready in 407 ms
  → Local:   http://localhost:5173/
  → Network: use --host to expose
  → press h to show help

LARAVEL v10.48.28 plugin v0.7.8
  → APP_URL: http://localhost
```

7. Selanjutnya kita akan buat datatables untuk kategori

```
D:\Apps\laragon\www\PWL2025\week 5\jobsheet>php artisan datatables:make Kategori

INFO DataTable [D:\Apps\laragon\www\PWL2025\week 5\jobsheet\app\DataTables\KategoriDataTable.php]
created successfully.
```

8. Kita edit KategoriDataTable untuk mengatur kolom apasaja yang ingin ditampilkan
Sesuaikan dengan kodeprogram berikut

```
app > DataTables > KategoriDataTable.php > PHP Intelephense > KategoriDataTable > dataTable
1 <?php
2
3 namespace App\DataTables;
4
5 use App\Models\Kategori;
6 use Illuminate\Database\Eloquent\Builder as QueryBuilder;
7 use Yajra\DataTables\EloquentDataTable;
8 use Yajra\DataTables\Html\Builder as HtmlBuilder;
9 use Yajra\DataTables\Html\Button;
10 use Yajra\DataTables\Html\Column;
11 use Yajra\DataTables\Html\Editor\Editor;
12 use Yajra\DataTables\Html\Editor\Fields;
13 use Yajra\DataTables\Services\DataTable;
14
15 0 references | 0 implementations
16 class KategoriDataTable extends DataTable
17 {
18     /**
19      * Build the DataTable class.
20      *
21      * @param QueryBuilder $query Results from query() method.
22      */
23     0 references | 0 overrides
24     public function dataTable(QueryBuilder $query): EloquentDataTable
25     {
26         return (new EloquentDataTable($query))
27             /*->addColumn('action', 'kategori.action') */
28             ->setRowId('id');
29     }
30 }
```

9. Ubah kategori model, sesuaikan seperti berikut

```
app > Models > KategoriModel.php > ...
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Relations\HasMany;
6 use Illuminate\Database\Eloquent\Model;
7 use App\Models\BarangModel;
8
9 4 references | 0 implementations
10 class KategoriModel extends Model
11 {
12     0 references
13     protected $table = 'm_kategori';
14     0 references
15     protected $primaryKey = 'kategori_id';
16     0 references
17     protected $fillable = ['kategori_kode', 'kategori_nama'];
18
19     0 references | 0 overrides
20     public function barang(): HasMany
21     {
22         return $this->hasMany(related: BarangModel::class, foreignKey: 'barang_id', localKey: '
23     }
24 }
```

10. Ubah Kategori Controller sesuaikan dengan skrip berikut:

```
1 reference | 0 overrides
public function index (KategoriDataTable $dataTable): mixed
{
    return $dataTable->render('kategori.index'); Argument '1'
}
```

11. Buat folder kategori di dalam resources/view, kemudian buat view blade index untuk kategori resources/views/kategori/index.blade.php

```
resources > views > kategori > index.blade.php > ...
1 @extends(view: 'layouts.app')
2
3 @section(section: 'subtitle', content: 'Kategori')
4 @section(section: 'content-header-title', content: 'Home')
5 @section(section: 'content-header-subtitle', content: 'Kategori')
6
7 @section(section: 'content')
8     <div class="container">
9         <div class="card">
10             <div class="card-header">Manage Kategori</div>
11             <div class="card-body">
12                 {{ $dataTable->table() }}
13             </div>
14         </div>
15     </div>
16 @endsection
17
18 @push('scripts')
19     {{ $dataTable->scripts() }}
20 @endpush
```

12. Pastikan route untung kategori sudah tersedia

```
Route::get(uri: '/kategori', action: [KategoriController::class, 'index']);
```


13. Sesuaikan app layout seperti kode berikut

```
resources > views > layouts > app.blade.php > ...
You 3 seconds ago | 1 author (You)
1 @extends(view: 'adminlte::page')
2 {{-- Extend and customize the browser title --}}
3 @section(section: 'title')
4 {{ config(key: 'adminlte.title') }}
5 @hasSection(name: 'subtitle') | @yield(section: 'subtitle') @endif
6 @stop
7 @vite(entrypoints: 'resources/js/app.js')
8 {{-- Extend and customize the page content header --}}
9 @section(section: 'content_header')
10 @hasSection(name: 'content_header_title')
11 <h1 class="text-muted">
12     @yield(section: 'content_header_title')
13     @hasSection(name: 'content_header_subtitle')
14     <small class="text-dark">
15         <i class="fas fa-xs fa-angle-right text-muted"></i>
16         @yield(section: 'content_header_subtitle')
17     </small>
18     @endif
19 </h1>
20 @endif
21 @stop
22 {{-- Rename section content to content_body --}}
23 @section(section: 'content')
24 @yield(section: 'content_body')
25 @stop
26 {{-- Create a common footer --}}
27 @section(section: 'footer')
28 <div class="float-right">
29     Version: {{ config(key: 'app.version', default: '1.0.0') }}
```

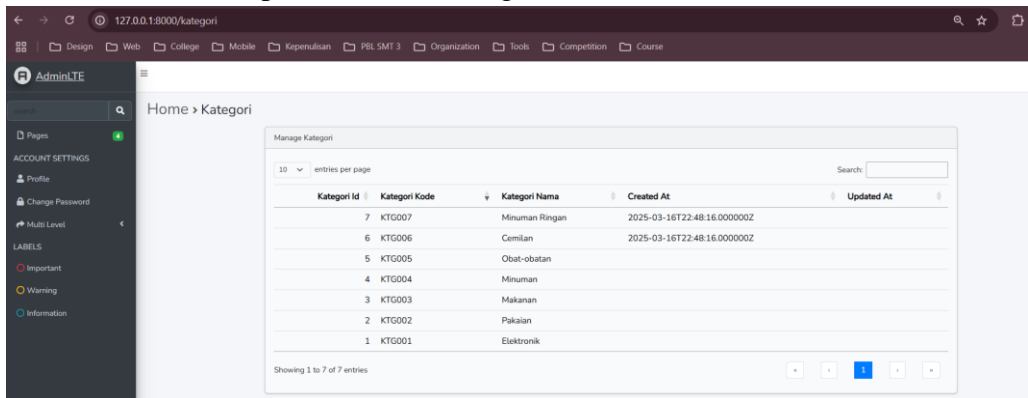
14. Menset ViteJs / script type defaults

```
app > Providers > AppServiceProvider.php > ...
4
5 use Illuminate\Support\ServiceProvider;
6 use Illuminate\Support\Facades\View;
7 use Yajra\DataTables\Html\Builder;
8
9 1 reference | 0 implementations | You, 3 seconds ago | 1 author (You)
10 class AppServiceProvider extends ServiceProvider
11 {
12     /**
13      * Register any application services.
14      * @return void
15      */
16     0 references | 0 overrides
17     public function register(): void
18     {
19         //
20     }
21
22     /**
23      * Bootstrap any application services.
24      * @return void
25      */
26     0 references | 0 overrides
27     public function boot(): void
28     {
29         Builder::useVite();
30     }
31 }
```

15. . Isikan beberapa data ke table kategori

	kategori_id	kategori_kode	kategori_nama	created_at	updated_at
Edit Copy Delete	1	KTG001	Elektronik	NULL	NULL
Edit Copy Delete	2	KTG002	Pakaian	NULL	NULL
Edit Copy Delete	3	KTG003	Makanan	NULL	NULL
Edit Copy Delete	4	KTG004	Minuman	NULL	NULL
Edit Copy Delete	5	KTG005	Obat-obatan	NULL	NULL
Edit Copy Delete	6	KTG006	Cemilan	2025-03-16 22:48:16	NULL
Edit Copy Delete	7	KTG007	Minuman Ringan	2025-03-16 22:48:16	NULL

16. Datatables sudah dapat di load di /kategori



The screenshot shows a web application interface with a sidebar on the left containing navigation links like 'Pages', 'ACCOUNT SETTINGS', 'Profile', 'Change Password', 'Multi Level', and 'LABELS'. The main content area displays a 'Manage Kategori' table with columns for 'Kategori Id', 'Kategori Kode', 'Kategori Nama', 'Created At', and 'Updated At'. The table contains 7 entries, with the first entry (ID 7) highlighted in blue. The table is paginated, showing 'Showing 1 to 7 of 7 entries'.

Manage Kategori				
10	entries per page	Search:		
Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At
7	KTG007	Minuman Ringan	2025-03-16T22:48:16.000000Z	
6	KTG006	Cemilan	2025-03-16T22:48:16.000000Z	
5	KTG005	Obat-obatan		
4	KTG004	Minuman		
3	KTG003	Makanan		
2	KTG002	Pakaian		
1	KTG001	Elektronik		
Showing 1 to 7 of 7 entries				

PRAKTIKUM 3 – MEMBUAT FORM KEMUDIAN MENYIMPAN DATA DALAM DATABASE

1. Menyesuaikan routing, tambahkan dua routing berikut

```
konfig::bootstrap(nama: '\kategoribaru', action: [kategoribaru::cJ922', '2f06,]);  
konfig::bootstrap(nama: '\kategoribaru\cJ922', action: [kategoribaru::cJ922', 'cJ922,]);
```

2. Tambahkan dua function berikut dalam KategoriController

```
app > Http > Controllers > KategoriController.php > ...  
You 4 seconds ago | 1 author (You)  
1 <?php  
2  
3 namespace App\Http\Controllers;  
4  
5 use App\Models\KategoriModel;  
6 use Illuminate\Http\Request;  
7 use App\DataTables\KategoriDataTable;  
8  
9 5 references | 0 implementations | You 4 seconds ago | 1 author (You)  
10 class KategoriController extends Controller  
11 {  
12     2 references | 0 overrides  
13     public function index(KategoriDataTable $dataTable): mixed  
14     {  
15         return $dataTable->render('kategori.index'); Argument '1' passed  
16     }  
17  
18     1 reference | 0 overrides  
19     public function create(): Factory|View  
20     {  
21         return view(view: 'kategori.create');  
22     }  
23  
24     1 reference | 0 overrides  
25     public function store(Request $request): Redirector|RedirectResponse  
26     {  
27         KategoriModel::create(attributes: [  
28             'kategori_kode' => $request->kodeKategori,  
29             'kategori_nama' => $request->namaKategori,  
30         ]);  
31     }  
32 }
```

3. Dalam folder views/kategori, buatlah file dengan nama create.blade.php

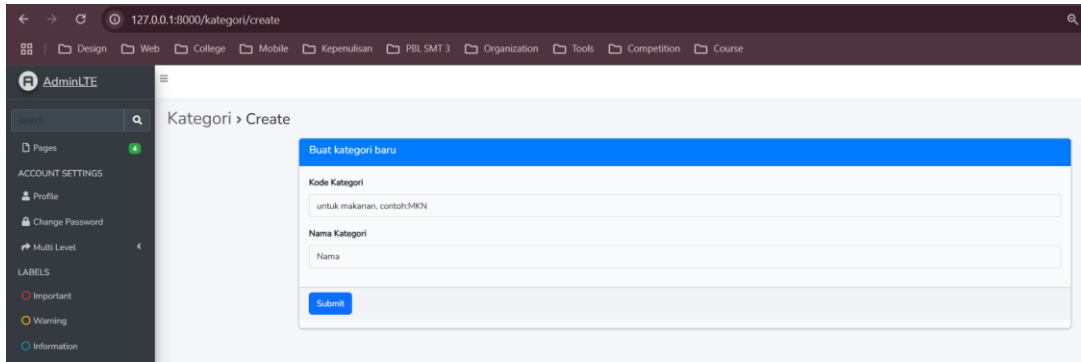
```
resources > views > kategori > create.blade.php > ...  
1 @extends('layouts.app')  
2  
3 {{-- Customize layout sections --}}  
4 @section(section: 'subtitle', content: 'Kategori')  
5 @section(section: 'content_header_title', content: 'Kategori')  
6 @section(section: 'content_header_subtitle', content: 'Create')  
7  
8 {{-- Content body: main page content --}}  
9 @section(section: 'content')  
10 <div class="container">  
11     <div class="card card-primary">  
12         <div class="card-header">  
13             <h3 class="card-title">Buat kategori baru</h3>  
14         </div>  
15  
16         <form method="post" action="..../kategori">  
17             <div class="card-body">  
18                 <div class="form-group">  
19                     <label for="kodeKategori">Kode Kategori</label>  
20                     <input type="text" class="form-control" id="kodeKategori" name="kodeKategori" />  
21                 </div>  
22  
23                 <div class="form-group">  
24                     <label for="namaKategori">Nama Kategori</label>  
25                     <input type="text" class="form-control" id="namaKategori" name="namaKategori" />  
26                 </div>  
27             </div>  
28  
29             <div class="card-footer">  
30                 <button type="submit" class="btn btn-primary">Submit</button>  
31             </div>  
32         </div>  
33     </div>  
34 </section>  
35 </div>
```

4. Kita lakukan pengecualian proteksi CsrfToken. Karena kita belum melakukan autentikasi. Kita edit dalam file berikut

```
app > http > Middleware > VerifyCsrfToken.php > ...
You, 1 second ago | 1 author (You)

1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Illuminate\Foundation\Http\Middleware\VerifyCsrfToken as Middleware;
6
7  2 references | 0 implementations | You, 1 second ago | 1 author (You)
8  class VerifyCsrfToken extends Middleware
9  {
10     /**
11      * The URIs that should be excluded from CSRF verification.
12      *
13      * @var array<int, string>
14      */
15     protected $except = [
16         '/kategori'
17     ];
18 }
```

5. Akses kategori/create



Kategori > Create

Buat kategori baru

Kode Kategori

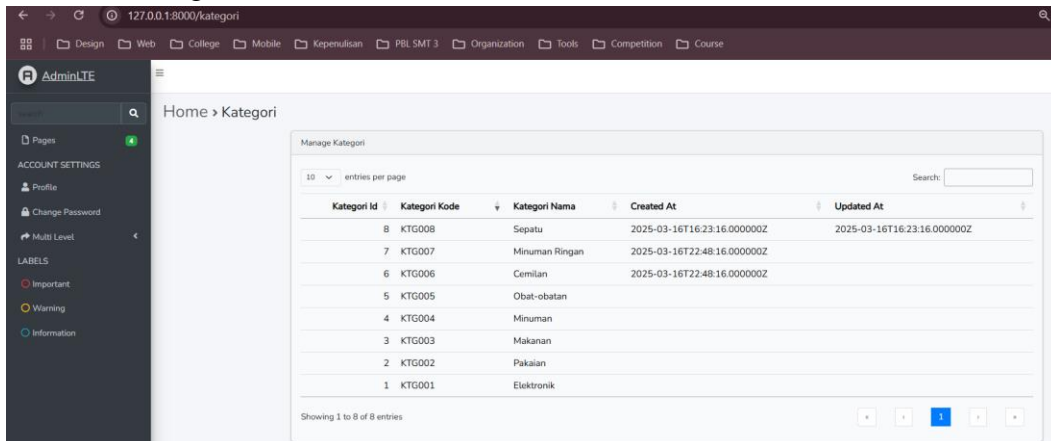
untuk makanan, contoh: MKN

Nama Kategori

Nama

Submit

6. Halaman Kategori



Home > Kategori

Manage Kategori

10 entries per page

Search:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At
8	KTG008	Sepatu	2025-03-16T16:23:16.000000Z	2025-03-16T16:23:16.000000Z
7	KTG007	Minuman Ringan	2025-03-16T22:48:16.000000Z	
6	KTG006	Cemilan	2025-03-16T22:48:16.000000Z	
5	KTG005	Obat-obatan		
4	KTG004	Minuman		
3	KTG003	Makanan		
2	KTG002	Pakaian		
1	KTG001	Elektronik		

Showing 1 to 8 of 8 entries

TUGAS PRAKTIKUM

1. Tambahkan button Add di halam manage kategori, yang mengarah ke create kategori baru

Jawab:

- a. Modifikasi index.blade.php untuk menambahkan button add

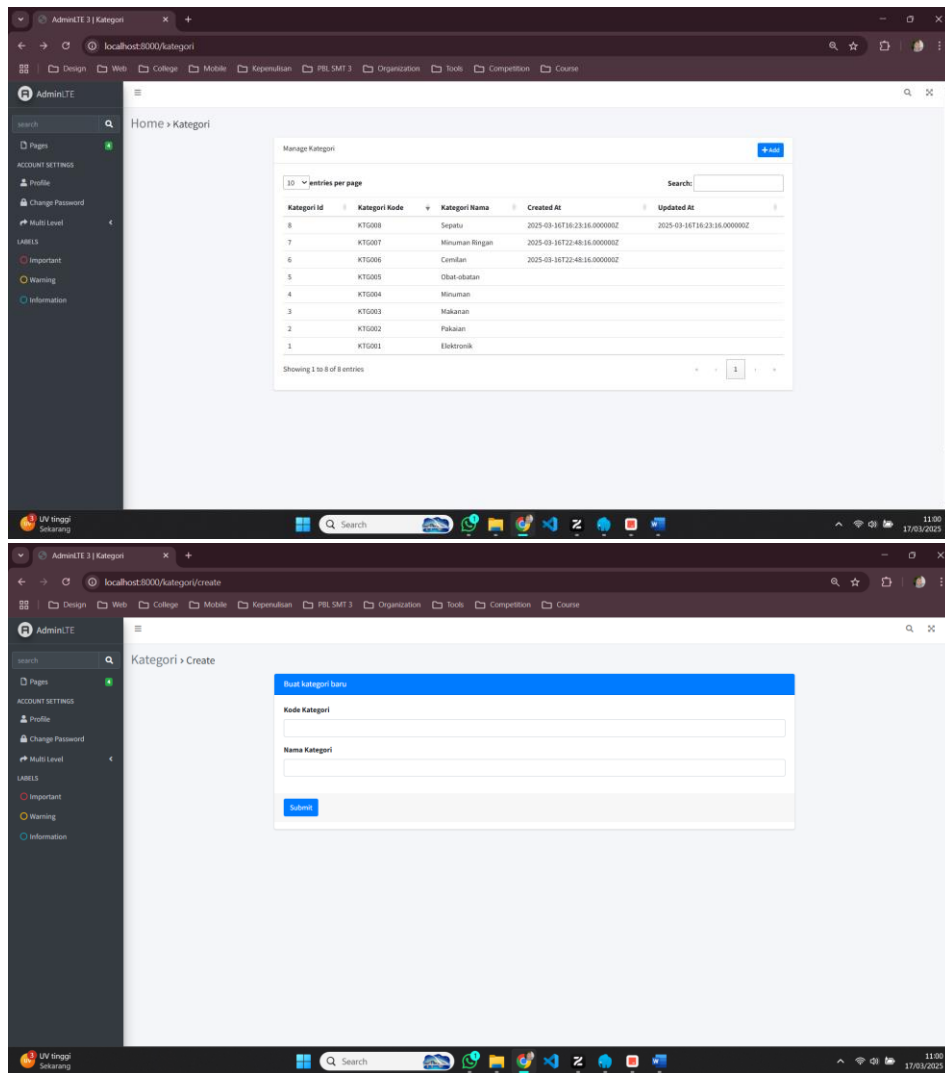
```
index.blade.php x web.php
resources / views / kategori / index.blade.php

1  @extends('layouts.app')
2
3  @section('subtitle', 'Kategori')
4  @section('content_header_title', 'Home')
5  @section('content_header_subtitle', 'Kategori')
6
7  @section('content')
8      <div class="container">
9          <div class="card">
10             <div class="card-header">
11                 Manage Kategori
12             <div class="float-right">
13                 <a href="{{ route('kategori.create') }}" class="btn btn-primary btn-sm">
14                     <i class="fa fa-plus"></i> Add
15                 </a>
16             </div>
17             </div>
18             <div class="card-body">
19                 {{ $dataTable->table() }}
20             </div>
21         </div>
22     </div>
23 @endsection
24
25 @push('scripts')
26     {{ $dataTable->scripts() }}
27 @endpush
```

- b. Modifikasi routingnya pada web.php

```
Route::get('/kategori/create', [KategoriController::class, 'create'])->name('kategori.create');
```

Ketika button add diklik pada /kategori, dia akan merouting ke /kategori/create



2. Tambahkan menu untuk halaman manage kategori, di daftar menu navbar

Jawab:

- Untuk menambahkan menu item pada Manage Kategori, kita perlu memodifikasi layout file dimana navbar menu didefinisikan, yaitu main layoutnya ada di `app.blade.php`, yg mengextend `adminlte::page`.

```

indexblade.php  app.blade.php x  web.php
resources / views / layouts / app.blade.php

1  @extends('adminlte::page')
2
3  {{-- Extend and customize the browser title --}}
4  @section('title')
5      {{ config('adminlte.title') }} @hassection('subtitle') | @yield('subtitle') @endif
6  @stop
7
8  {{-- Extend and customize the page content header --}}
9  @section('content_header')
10     @hassection('content_header_title')
11         <h1 class="text-muted">
12             @yield('content_header_title')
13             @hassection('content_header_subtitle')
14                 <small class="text-dark">
15                     <i class="fas fa-xs fa-angle-right text-muted"></i>
16                     @yield('content_header_subtitle')
17                 </small>
18             @endif
19         </h1>
20     @endif
21 @stop
22
23 {{-- Rename section content to content_body --}}
24 @section('content')
25     @yield('content_body')
26 @stop
27
28 {{-- Create a common footer --}}
29 @section('footer')
30     <div class="float-right">
31         Version: {{ config('app.version', '1.0.0') }}
32     </div>
33     <strong>

```

b. Modifikasi /config/adminlte.php karena konfigurasi item menunya ada disini

```

indexblade.php  app.blade.php  adminlte.php x  web.php
config / adminlte.php

289  /*
301  'menu' => [
322      ],
323      [
324          'text' => 'Manage Kategori',
325          'url' => '/kategori',
326          'icon' => 'fas fa-list',
327      ],
328      [
329          'text' => 'blog',
330          'url' => 'admin/blog',
331          'can' => 'manage-blog',
332      ],
333      [
334          'text' => 'pages',
335          'url' => 'admin/pages',
336          'icon' => 'far fa-fw fa-file',
337          'label' => 4,
338          'label_color' => 'success',
339      ],
340      ['header' => 'account_settings'],
341      [
342          'text' => 'profile',
343          'url' => 'admin/settings',
344          'icon' => 'fas fa-fw fa-user',
345      ],
346      [
347          'text' => 'change_password',
348          'url' => 'admin/settings',
349          'icon' => 'fas fa-fw fa-lock',
350      ],
351      [

```

3. Tambahkan action edit di datatables dan buat halaman edit serta controllernya

Jawab:

- a. Kita tambahkan kolom “action” di datatables yg berisi edit untuk setiap baris data di table kategori

```
class KategoriDataTable extends DataTable
{
    /**
     * Build the DataTable class.
     *
     * @param QueryBuilder $query Results from query() method.
     */
    public function dataTable($query)
    {
        return (new EloquentDataTable($query))
            ->addColumn('action', 'kategori.action') // Tambahkan kolom action
            ->setRowId('id');
    }
}
```

```
public function getColumns(): array
{
    return [
        Column::computed('action') // Tambahkan kolom action
            ->exportable(false)
            ->printable(false)
            ->width(60)
            ->addClass('text-center'),
        Column::make('kategori_id'),
        Column::make('kategori_kode'),
        Column::make('kategori_nama'),
        Column::make('created_at'),
        Column::make('updated_at'),
    ];
}
```

- b. Membuat view untuk menampilkan tombol edit di kolom action

```
index.blade.php  app.blade.php  adminlte.php  KategoriDataTable.php  action.blade.php x  web.php

resources / views / kategori / action.blade.php

1  <a href="{{ url('/kategori/' . $kategori_id . '/edit') }}" class="btn btn-sm btn-primary">Edit</a>
```

- c. Menambahkan routing

```
Route::get('/kategori/create', [KategoriController::class, 'create'])->name('kategori.create');
Route::get('/kategori/{id}/edit', [KategoriController::class, 'edit']);
Route::put('/kategori/{id}', [KategoriController::class, 'update']);
```


- d. Menambahkan method edit dan update pada KategoriController untuk menampilkan form edit, dan update untuk menyimpan perubahan.

```
// Method untuk menampilkan form edit
public function edit($id)
{
    $kategori = KategoriModel::findOrFail($id);
    return view('kategori.edit', compact('kategori'));
}

// Method untuk menyimpan perubahan
public function update(Request $request, $id)
{
    $kategori = KategoriModel::findOrFail($id);
    $kategori->update([
        'kategori_kode' => $request->kodeKategori,
        'kategori_nama' => $request->namaKategori,
    ]);
    return redirect('/kategori');
}
```

- e. Buat view edit.blade.php yang berisi form untuk mengedit data kategori

```
resources / views / kategori / edit.blade.php
1 @extends('layouts.app')
2
3 @section('subtitle', 'Kategori')
4 @section('content_header_title', 'Kategori')
5 @section('content_header_subtitle', 'Edit')
6
7 @section('content')
8 <div class="container">
9     <div class="card">
10         <div class="card-header">
11             <h3 class="card-title">Edit Kategori</h3>
12         </div>
13         <form method="post" action="{{ url('/kategori/' . $kategori->kategori_id) }}">
14             @method('PUT')
15             @csrf
16             <div class="card-body">
17                 <div class="form-group">
18                     <label for="kodeKategori">Kode Kategori</label>
19                     <input type="text" class="form-control" id="kodeKategori" name="kodeKategori" value="{{ $kategori->kategori_kode }}" placeholder="Masukkan kode kate
20                 </div>
21                 <div class="form-group">
22                     <label for="namaKategori">Nama Kategori</label>
23                     <input type="text" class="form-control" id="namaKategori" name="namaKategori" value="{{ $kategori->kategori_nama }}" placeholder="Masukkan nama kate
24                 </div>
25             </div>
26             <div class="card-footer">
27                 <button type="submit" class="btn btn-primary">Simpan</button>
28                 <a href="{{ url('/kategori/' . $kategori->kategori_id) }}" class="btn btn-secondary">Batal</a>
29             </div>
30         </form>
31     </div>
32 </div>
33 @endsection
```

- f. Karena kita menggunakan method PUT, kita perlu memastikan route ini dikecualikan dari verifikasi CSRF (sementara, karena belum ada autentikasi). Kita akan menambahkan /kategori/* ke array \$except:

```

1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Illuminate\Foundation\Http\Middleware\VerifyCsrfToken as Middleware;
6
7  class VerifyCsrfToken extends Middleware
8
9
10     /**
11      * The URIs that should be excluded from CSRF verification.
12      *
13      * @var array<int, string>
14      */
15     protected $except = [
16         '/kategori',
17         '/kategori/*' //tambahkan ini untuk route PUT
18     ];

```

g. Hasilnya

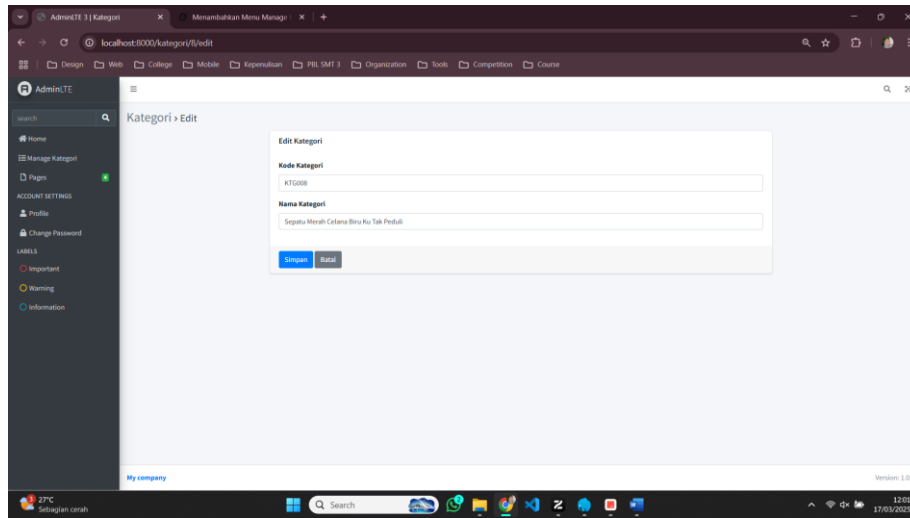
Manage Kategori

10 entries per page

Search:

Action	Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At
Edit	8	KTG008	Sepatu Merah Celana Biru Kuy Tak Peduli	2025-03-16T16:23:16.000000Z	2025-03-17T04:57:23.000000Z
Edit	7	KTG007	Minuman Ringan	2025-03-16T22:48:16.000000Z	
Edit	6	KTG006	Cemilan	2025-03-16T22:48:16.000000Z	
Edit	5	KTG005	Obat-obatan		
Edit	4	KTG004	Minuman		
Edit	3	KTG003	Makanan		
Edit	2	KTG002	Pakaian		
Edit	1	KTG001	Elektronik		

Showing 1 to 8 of 8 entries



4. Tambahkan action delete di datatables serta controllernya

Jawab:

- a. Kita akan memperbarui kolom "Action" di DataTables untuk menambahkan tombol "Delete" di samping tombol "Edit" yang sudah ada.

```

app.blade.php  VerifyCsrfToken.php  adminlte.php  KategoriDataTable.php  action.blade.php
app / DataTables / KategoriDataTable.php

1  <?php
2
3  namespace App\DataTables;
4
5  use App\Models\KategoriModel;
6  use Illuminate\Database\Eloquent\Builder as QueryBuilder;
7  use Yajra\DataTables\EloquentDataTable;
8  use Yajra\DataTables\Html\Builder as HtmlBuilder;
9  use Yajra\DataTables\Html/Button;
10 use Yajra\DataTables\Html\Column;
11 use Yajra\DataTables\Services\DataTable;
12
13 class KategoriDataTable extends DataTable
14 {
15     /**
16      * Build the DataTable class.
17      *
18      * @param QueryBuilder $query Results from query() method.
19      */
20     public function dataTable($query)
21     {
22         return (new EloquentDataTable($query))
23             ->addColumn('action', 'kategori.action') // Tetap menggunakan view yang sama
24             ->setRowId('id');
25     }
26
27     /**
28      * Get the query source of dataTable.
29      */
30     public function query(KategoriModel $model): QueryBuilder
31     {
32         return $model->newQuery();
33     }

```

- b. Tambahkan tombol "Delete" di samping tombol "Edit" dalam file view. Karena link <a> tidak mendukung method DELETE, kita akan menggantinya dengan form kecil yang menggunakan method DELETE

```

app.blade.php  VerifyCsrfToken.php  adminlte.php  KategoriDataTable.php  action.blade.php  edit.blade.php  web.php  KategoriController.php
resources / views / kategori / action.blade.php

1  <div class="action-buttons">
2      <a href="{{ url('/kategori/' . $kategori_id . '/edit') }}" class="btn btn-sm btn-primary">Edit</a>
3      <form action="{{ url('/kategori/' . $kategori_id . '/delete') }}" method="POST" style="display:inline;">
4          @method('DELETE')
5          <button type="submit" class="btn btn-sm btn-danger" onclick="return confirm('Apakah Anda yakin ingin menghapus kategori ini?')">Delete</button>
6      </form>
7  </div>

```

- c. Tambahkan route untuk menghapus data kategori menggunakan method DELETE

```
Route::delete('/kategori/{id}', [KategoriController::class, 'destroy']);
```

- d. Tambahkan method destroy di KategoriController untuk menangani penghapusan data

```
// Method untuk menghapus data
public function destroy($id)
{
    $kategori = Kategorimodel::findOrFail($id);
    $kategori->delete();
    return redirect('/kategori')->with('success', 'Kategori berhasil dihapus!');
}
```

- e. Untuk memberikan feedback kepada pengguna setelah penghapusan berhasil, tambahkan pesan sukses di halaman index.

```
resources / views / kategori / index.blade.php

1  @extends('layouts.app')
2
3  @section('subtitle', 'Kategori')
4  @section('content_header_title', 'Home')
5  @section('content_header_subtitle', 'Kategori')
6
7  @section('content')
8      <div class="container">
9          @if (session('success'))
10             <div class="alert alert-success">
11                 {{ session('success') }}
12             </div>
13          @endif
14          <div class="card">
15             <div class="card-header">Manage Kategori</div>
16             <div class="card-body">
17                 {{ $dataTable->table() }}
18             </div>
19          </div>
20      </div>
21  @endsection
22
23  @push('scripts')
24      {{ $dataTable->scripts() }}
25  @endpush
```

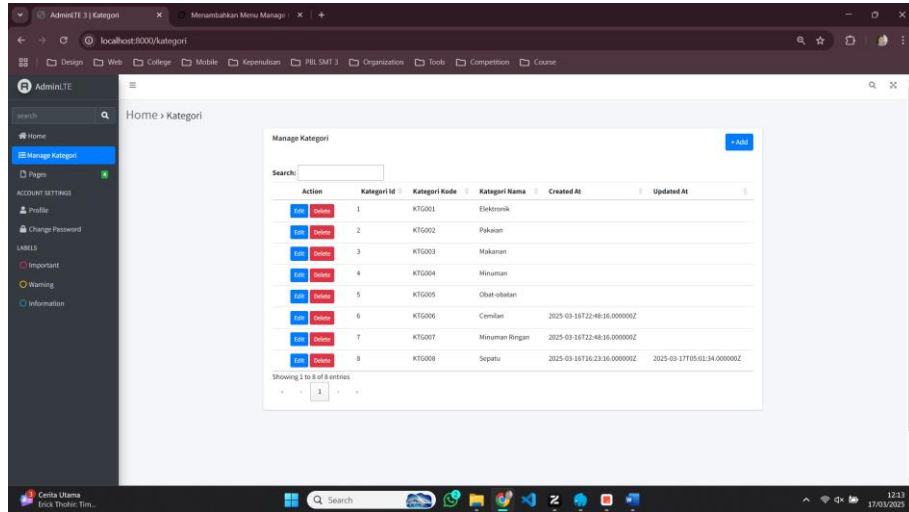
- f. Karena form HTML tidak mendukung method DELETE secara langsung, kita akan menggunakan @method('DELETE') di dalam link penghapusan. Namun, karena tombol "Delete" adalah link (<a>), kita perlu menangani ini melalui JavaScript atau memastikan route dikecualikan dari verifikasi CSRF sementara

```

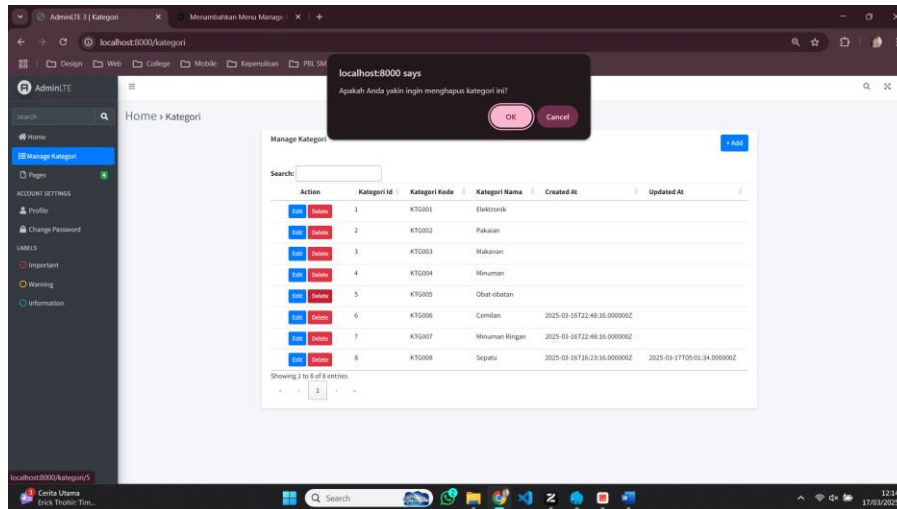
1 <?php
2
3 namespace App\Http\Middleware;
4
5 use Illuminate\Foundation\Http\Middleware\VerifyCsrfToken as Middleware;
6
7 class VerifyCsrfToken extends Middleware
8
9 /**
10  * The URIs that should be excluded from CSRF verification.
11  *
12  * @var array<int, string>
13  */
14 protected $except = [
15     '/kategori',
16     '/kategori/*' //tambahkan ini untuk route PUT
17 ];
18

```

g. Hasilnya



Saya ingin menghapus kategori id 8



Maka tampilannya akan seperti ini

