



Mata Kuliah : Pemrograman Web Lanjut (PWL)
Program Studi : D4 – Teknik Informatika / D4 – Sistem Informasi Bisnis
Semester : 4 (empat) / 6 (enam)
Pertemuan ke- : 5 (satu)
Nama : Karina Ika Indasa

JOBSHEET 05

Blade View, Web Templating(AdminLTE), Datatables

View merupakan tempat bagi kita untuk meletakkan kode-kode HTML. Sehingga dalam laravel tidak menggunakan lagi file berformat .html. Kita masih menggunakan tagging HTML, tapi kita perlu handle tampilan dengan lebih canggih. Menampilkan data yang diberikan oleh controller. Untuk itu kita akan menggunakan templating engine, yaitu Blade.

Blade merupakan templating engine bawaan Laravel. Berguna untuk mempermudah dalam menulis kode tampilan. Dan juga memberikan fitur tambahan untuk memanipulasi data di view yang dilempar dari controller. Blade juga memungkinkan penggunaan plain PHP pada kode View.

Sesuai dengan **Studi Kasus PWL.pdf**.

Jadi project Laravel 10 kita masih sama dengan menggunakan repositori **PWL_POS**.

Project PWL_POS akan kita gunakan sampai pertemuan 12 nanti, sebagai project yang akan kita pelajari

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.

```
<!-- View pada resources/views/hello.blade.php -->
<html>
    <body>
        <h1>Hello, {{ $name }}</h1>
    </body>
</html>
```



View tersebut dapat dijalankan melalui Routing, dimana route akan memanggil View sesuai dengan nama file tanpa 'blade.php'

```
Route::get('/hello', function () {  
    return view('hello', ['name' => '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('hello', ['name' => 'Andi']);
```

Seperti yang kita lihat, argumen pertama yang diteruskan ke view helper sesuai dengan nama file view di direktori resources / views. Argumen kedua adalah larik data yang harus tersedia untuk tampilan. Dalam hal ini, laravel meneruskan variabel nama yang ditampilkan dalam tampilan menggunakan sintaks Blade.

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('/hello', function () {  
    return view('blog.hello', ['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('/hello,[WelcomeController::class, 'hello']);
```

```
<?php  
namespace App\Http\Controllers;  
use Illuminate\Http\Request;  
class WelcomeController extends Controller  
{  
    public function hello() {  
        return view('blog.hello', ['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('hello', ['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

```
return view('hello')
    ->with('name', 'Andi')
    ->with('occupation', 'Astronaut');
```

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

```
<?php
namespace App\Providers;

use Illuminate\Support\Facades\View;

class AppServiceProvider extends ServiceProvider
{
    /**
     * Register any application services.
     *
     * @return void
     */
    public function register()
    {
        //
    }
}
```



```
}

/**
 * Bootstrap any application services.
 *
 * @return void
 */
public function boot()
{
    View::share('key', 'value');
}
}
```

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

Pada laravel, layout digunakan untuk membuat master view yang akan selalu ditampilkan oleh view-view child yang menggunakannya. Dalam sebuah layout kita bisa memberikan tempat-tempat yang bisa digunakan oleh child view. Tempat-tempat tersebut adalah section. Misalnya, dalam layout utama, kita definisikan section sidebar, main_content, dan footer. Selanjutnya, setiap child view yang menggunakan layout utama dapat menempatkan kode view di masing-masing section yang tersedia di layout utama. Pada layout, setiap kode html akan digunakan oleh child view. Sehingga child view tidak perlu mendefinisikan tag html, head, title, dll pada tiap-tiap view. Terdapat beberapa istilah yang digunakan untuk menerapkan layout.

@yield

@yield("nama_section") digunakan untuk mendefinisikan bagian dari layout yang akan digunakan dan diisi oleh child view.

@section

@section digunakan selain untuk mendefinisikan sebuah section, juga bisa untuk mengisi section yang diharapkan oleh parent view / layout melalui @yield. Diakhiri dengan @endsction.



@parent

Dalam child view kita bisa menampilkan juga konten yang ada pada parent dalam section tertentu, hal tersebut dilakukan dengan @parent.

@extends

Extends digunakan pada setiap child view yang ingin menggunakan sebuah view sebagai parent / layout.

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

```
<!-- Disimpan di resources/views/layouts/app.blade.php -->
<html>
  <head>
    <title> Halaman @yield('title')</title>
  </head>
  <body>
    @section('sidebar')
      Ini adalah master sidebar.
    @show
    <div
      class="container
      ">
      @yield('content'
      )
    </div>
  </body>
</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.



```
<!--Disimpan di resources/views/child.blade.php -->
@extends('layouts.app')
@section('title', 'Profil')
@section('sidebar')
    @parent
    <p>Sidebar halaman Profil.</p>
@endsection
@section('content')
    <p>Ini adalah bagian konten. NIM - Nama</p>
@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.

```
<!-- Disimpan di resources/views/components/alert.blade.php -->
<div class="alert alert-danger">
{{ $slot }}
</div>
```

Di bawah ini adalah contoh view yang menerapkan component alert.



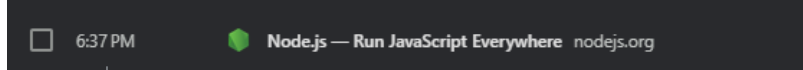
```
@extends('layouts.app')  
// kode..  
@component("alert")  
<b>Tulisan ini akan mengisi variabel $slot</b>  
@endcomponent
```

C. Bootstrap pada Laravel

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

Dalam jobsheet ini instalasi bootstrap dilakukan dibagian praktikum. Berikut ini hanya pengetahuan cara instalasi umumnya.

1. Melakukan instalasi Node.js. Installer dapat diperoleh dari <https://nodejs.org/en/>. Sesuaikan dengan sistem operasi yang digunakan.



2. Untuk memastikan instalasi Node.js dan NPM berjalan lancar, Anda dapat memeriksanya dengan menjalankan dua perintah berikut melalui Command Prompt:

```
node -v  
npm -v
```

```
C:\laragon\www\PWL_2025\Week5\JS5>node -v  
v22.14.0  
  
C:\laragon\www\PWL_2025\Week5\JS5>npm -v  
10.9.2
```

Setelah Anda mengetikkan kedua perintah tersebut, Command Prompt akan menunjukkan versi Node.js dan NPM yang ter-install di komputer.

3. Kemudian melalui command prompt, ubah direktori ke dalam direktori project laravel yang telah dibuat sebelumnya.
4. Scaffolding Bootstrap yang disediakan oleh Laravel terletak di dalam paket Composer, yang dapat diinstal menggunakan Composer:laravel/ui.

```
composer require laravel/ui
```

```
C:\laragon\www\PWL_2025\Week5\JS5>composer require laravel/ui  
laravel/ui is currently present in the require-dev key and you ran the command without the --dev flag, which will move it to the require 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
```



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

```
php artisan ui bootstrap

php artisan ui bootstrap --auth
```

```
C:\laragon\www\PWL_2025\Week5\JS5>php artisan ui bootstrap
[INFO] Bootstrap scaffolding installed successfully.
[WARN] Please run [npm install && npm run dev] to compile your fresh scaffolding.

C:\laragon\www\PWL_2025\Week5\JS5>php artisan ui bootstrap --auth
The [layouts/app.blade.php] view already exists. Do you want to replace it? (yes/no) [no]
> yes
The [Controller.php] file already exists. Do you want to replace it? (yes/no) [yes]
> yes
[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, instal dependensi frontend proyek Anda menggunakan Node package manager (NPM) :

```
npm install
```

```
C:\laragon\www\PWL_2025\Week5\JS5>npm install
added 53 packages, and audited 54 packages in 2m
16 packages are looking for funding
  run 'npm fund' for details
2 moderate severity vulnerabilities
To address all issues (including breaking changes), run:
  npm audit fix --force
Run 'npm audit' for details.
npm notice
npm notice New major version of npm available! 10.9.2 -> 11.2.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.2.0
npm notice To update run: npm install -g npm@11.2.0
npm notice
```

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

```
npm run dev
```

```
C:\laragon\www\PWL_2025\Week5\JS5>npm run dev
> dev
> vite

VITE v5.4.14 ready in 774 ms
→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help

LARAVEL v10.48.28 plugin v1.2.0
→ APP_URL: http://localhost
```




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 mengubungkan file css ke laravel anda dapat menggunakan syntax berikut:

```
<link rel="stylesheet" type="text/css" href="/css/style.css">
```

Atau

```
<link rel="stylesheet" type="text/css" href="{{ asset('/css/app.css') }}">
```

Dan file JSnya:

```
<script type="text/javascript" src="/js/app.js"></script>
```

Atau

```
<script type="text/javascript" src="{{ asset('/js/app.js') }}"></script>
```

Secara default linknya dimulai dari folder public. Jadi anda bisa meletakkan file css dan js didalam folder public



Praktikum 1 – Integrasi Laravel dengan AdminLTE3

Cara paling cepat dan clean dapat dilakukan menggunakan repository dalam tautan berikut <https://github.com/jeroennoten/Laravel-AdminLTE/wiki/Installation>.

1. Dalam root folder project lakukan command berikut, untuk mendefinisikan requirement project.

`composer require jeroennoten/laravel-adminlte`

```
C:\Windows\System32\cmd.exe X + v
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\laragon\www\PWL_2025\Week5\JS5>composer require jeroennoten/laravel-adminlte
./composer.json has been updated
Running composer update jeroennoten/laravel-adminlte
Loading composer repositories with package information
Updating dependencies
Lock file operations: 2 installs, 0 updates, 0 removals
  - Locking almasaeed2010/adminlte (v3.2.0)
  - Locking jeroennoten/laravel-adminlte (v3.15.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 2 installs, 0 updates, 0 removals
  - Downloading almasaeed2010/adminlte (v3.2.0)
  - Downloading jeroennoten/laravel-adminlte (v3.15.0)
  - Installing almasaeed2010/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
```

2. Melakukan instalasi requirement project di atas dengan command berikut:

`php artisan adminlte:install`

```
C:\laragon\www\PWL_2025\Week5\JS5>php artisan adminlte:install
AdminLTE assets files published successfully
Configuration file published successfully
Translation files published successfully
The installation is complete.
```

Perintah di atas akan meng-install:

- **AdminLTE distribution files** dan dependensinya (Bootstrap, jQuery, etc.) dalam folder public/vendor .
- Konfigurasi package di file config/adminlte.php
- Paket translasi di folder lang/vendor/adminlte/
- Dalam composer.json akan otomatis ditambahkan require untuk laravel-adminlte

```
Week5 > JS5 > {} composer.json > ...
1  {
2      "name": "laravel/laravel",
3      "type": "project",
4      "description": "The skeleton application for the Laravel framework.",
5      "keywords": ["laravel", "framework"],
6      "license": "MIT",
7      "require": {
8          "php": "^8.1",
9          "guzzlehttp/guzzle": "^7.2",
10         "jeroennoten/laravel-adminlte": "^3.15",
11         "laravel/framework": "^10.10",
12         "laravel/sanctum": "^3.3",
13         "laravel/tinker": "^2.8"
14     },
15     "require-dev": {
16         "fakerphp/faker": "^1.9.1",
17         "laravel/pint": "^1.0",
18         "laravel/sail": "^1.18",
19         "mockery/mockery": "^1.4.4",
20         "nunomaduro/collision": "^7.0",
21         "phpunit/phpunit": "^10.1",
22         "spatie/laravel-ignition": "^2.0"
23     },
24     "autoload": {
25         "psr-4": {
```



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

```
@extends('adminlte::page')

{{-- Extend and customize the browser title --}} @section('title')
    {{ config('adminlte.title') }} @hasSection('subtitle') | @yield('subtitle') @endif
@stop

{{-- Extend and customize the page content header --}}

@section('content_header') @hasSection('content_header_title')
    <h1 class="text-muted"> @yield('content_header_title')

    @hasSection('content_header_subtitle')
        <small class="text-dark">
            <i class="fas fa-xs fa-angle-right text-muted"></i> @yield('content_header_subtitle')
        </small> @endif
    </h1>
@elseif
@stop

{{-- Rename section content to content_body --}}

@section('content') @yield('content_body')
@stop

{{-- Create a common footer --}} @section('footer')
    <div class="float-right">
        Version: {{ config('app.version', '1.0.0') }}
    </div>

    <strong>
        <a href="{{ config('app.company_url', '#') }}">
            {{ config('app.company_name', 'Mycompany') }}
        </a>
    </strong> @stop

{{-- Add common Javascript/Jquery code --}} @push('js')

<script>

    $(document).ready(function() {
        // Add your common script logic here...
    });

</script> @endpush

{{-- Add common CSS customizations --}}

@push('css')
<style type="text/css">

    {{-- You can add AdminLTE customizations here --}}
    /*
    .card-header {
        border-bottom: none;
    }
    .card-title {
        font-weight: 600;
    }
    */
</style> @endpush
```



```
Week5 > JSS > resources > views > layouts > app.blade.php > ...
1 @extends('adminlte::page')
2
3 {{-- Extend and customize the browser title --}}
4 @section('title')
5 {{ config(key: 'adminlte.title') }}
6 @hasSection('subtitle') | @yield('subtitle') @endif
7 @stop
8
9 {{-- Extend and customize the page content header --}}
10 @section('content_header')
11 @hasSection('content_header_title')
12 <h1 class="text-muted">
13     @yield('content_header_title')
14
15     @hasSection('content_header_subtitle')
16     <small class="text-dark">
17         <i class="fas fa-xs fa-angle-right text-muted"></i>
18         @yield('content_header_subtitle')
19     </small>
20     @endif
21 </h1>
22 @endif
23 @stop
24
25 {{-- Rename section content to content_body --}}
26 @section('content')
27 @yield('content_body')
28 @stop
29
30 {{-- Create a common footer --}}
31 @section('footer')
32 <div class="float-right">
33     Version: {{ config(key: 'app.version', default: '1.0.0') }}
34 </div>
35
36 <strong>
37     <a href="{{ config(key: 'app.company_url', default: '#') }}">
38         {{ config(key: 'app.company_name', default: 'My company') }}
39     </a>
40 </strong>
41 @stop
42
43 {{-- Add common Javascript/Jquery code --}}
44 @push('js')
45 <script>
46     $(document).ready(function () {
47         // Add your common script logic here...
48     });
49 </script>
50 @endpush
51
52 {{-- Add common CSS customizations --}}
53 @push('css')
54 <style type="text/css">
55     /* You can add AdminLTE customizations here */
56     /*
57     .card-header {
58         border-bottom: none;
59     }
60     .card-title {
61         font-weight: 600;
62     }
63     */
64 </style>
65 @endpush
```

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



```
resources > views > welcome.blade.php > ...
1 @extends('layouts.app')
2
3 {{-- Customize layout sections --}}
4
5 @section('subtitle', 'Welcome')
6 @section('content_header_title', 'Home')
7 @section('content_header_subtitle', 'Welcome')
8
9 {{-- Content body: main page content --}}
10
11 @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
24 @push('js')
25 <script> console.log("Hi, I'm using the Laravel-AdminLTE package!"); </script>
26 @endpush
```

```
Week5 > JSS > resources > views > welcome.blade.php > ...
1 @extends('layouts.app')
2
3 {{-- Customize layout sections --}}
4 @section('subtitle', 'Welcome')
5 @section('content_header_title', 'Home')
6 @section('content_header_subtitle', 'Welcome')
7
8 {{-- Content body: main page content --}}
9 @section('content_body')
10 <p>Welcome to this beautiful admin panel.</p>
11 @stop
12
13 {{-- Push extra CSS --}}
14 @push('css')
15 {{-- Add here extra stylesheets --}}
16 <link rel="stylesheet" href="/css/admin_custom.css" --> --}}
17 @endpush
18
19 {{-- Push extra scripts --}}
20 @push('js')
21 <script> console.log("Hi, I'm using the Laravel-AdminLTE package!"); </script>
22 @endpush
23
```

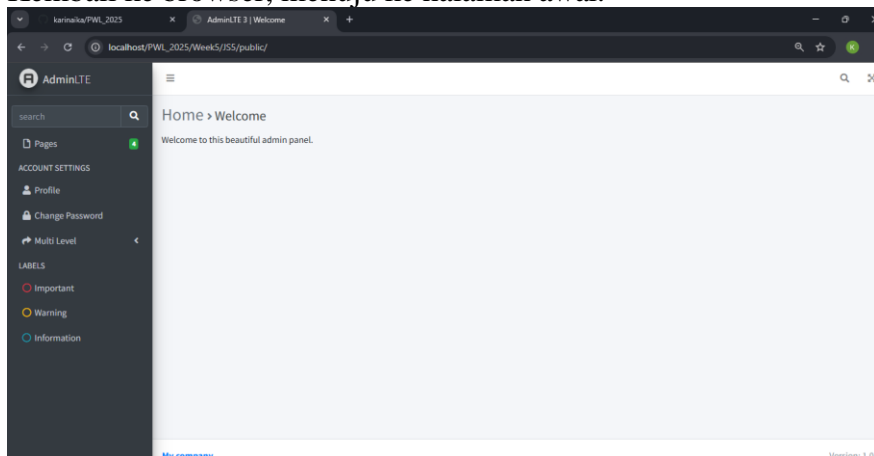
Untuk menggunakan blade template ini cukup dengan extend **AdminLTE layout** dengan cara `@extends('adminlte::page')`.

Template yields di beberapa bagian terklasifikasi dalam 2 yield:

- **main:** Biasa digunakan untuk extending the layout.
- **misc:** untuk kasus yang tidak biasa, atau hanya situasi tertentu.

Dokumentasi lebih detail terdapat di link berikut: <https://github.com/jeroennoten/Laravel-AdminLTE/wiki/Usage>

5. Kembali ke browser, menuju ke halaman awal.

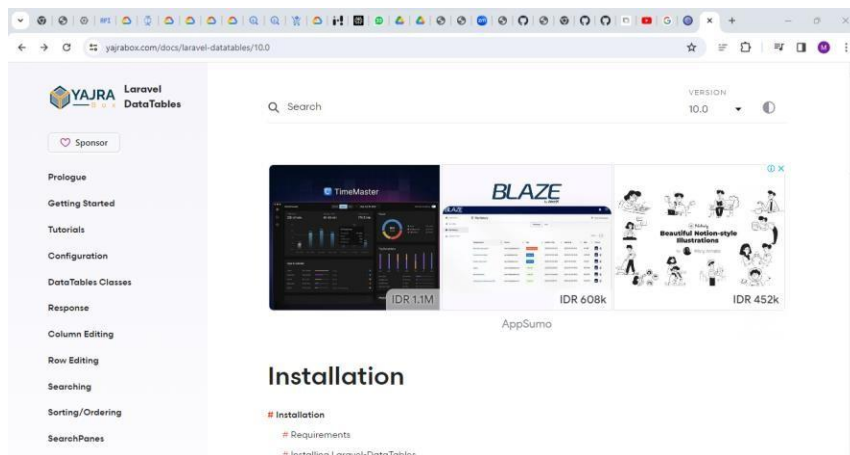




Praktikum 2 – Integrasi dengan DataTables

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.

<https://yajrabox.com/docs/laravel-datatables/10.0>



Berikut langkah-langkah praktikum

1. Install Laravel DataTables

`composer require laravel/ui --dev`

```
C:\Windows\System32\cmd.e X + v
C:\laragon\www\PWL_2025\Week5\JS5>composer require laravel/ui --dev
./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
> 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

81 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.
```



composer require yajra/laravel-datatables:^10.0

```
C:\Windows\System32\cmd.exe X + v
Using version ^4.6 for laravel/ui

C:\laragon\www\PWL_2025\Week5\JS5>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
8 package suggestions were added by new dependencies, use 'composer suggest' to see details.
```

2. Pastikan nodejs sudah terinstall, dengan perintah `npm -v`.

Jika belum install nodejs package manager dari <https://nodejs.org/dist/v20.11.1/node-x64.msi>

Npm digunakan untuk manajemen package javascript.

```
C:\laragon\www\PWL_2025\Week5\JS5>npm -v.
npm <command>

Usage:
npm install          install all the dependencies in your project
npm install <foo>    add the <foo> dependency to your project
npm test             run this project's tests
npm run <foo>        run the script named <foo>
npm <command> -h     quick help on <command>
npm -l              display usage info for all commands
npm help <term>      search for help on <term> (in a browser)
npm help npm         more involved overview (in a browser)

All commands:
```

3. Install Laravel DataTables Vite dan sass

npm i laravel-datatables-vite --save-dev

```
C:\laragon\www\PWL_2025\Week5\JS5>npm i laravel-datatables-vite --save-dev
npm warn idealTree Removing dependencies.laravel-datatables-vite in favor of devDependencies.laravel-datatables-vite

up to date, audited 63 packages in 6s

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

2 moderate severity vulnerabilities

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

Run 'npm audit' for details.
```

npm install -D sass

```
C:\laragon\www\PWL_2025\Week5\JS5>npm install -D sass

up to date, audited 63 packages in 2s

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

2 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



```
Week5 > JS5 > resources > js > JS app.js
1  import './bootstrap';
2  import '../sass/app.scss';
3  import 'laravel-datatables-vite';
```

5. Buatlah file resources/sass/app.scss

```
Week5 > JS5 > resources > sass > app.scss
1  // Fonts
2  @import url('https://fonts.bunny.net/css?family=Nunito');
3
4  // Bootstrap
5  @import 'bootstrap/scss/bootstrap';
6
7  // DataTables
8  @import 'bootstrap-icons/font/bootstrap-icons.css';
9  @import 'datatables.net-bs5/css/dataTables.bootstrap5.min.css';
10 @import 'datatables.net-buttons-bs5/css/buttons.bootstrap5.min.css';
11 @import 'datatables.net-select-bs5/css/select.bootstrap5.css';
```

6. Jalankan dengan `npm run dev`

```
C:\laragon\www\PWL_2025\Week5\JS5>npm run dev
> dev
> vite

VITE v5.4.14 ready in 774 ms
→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help

LARAVEL v10.48.28 plugin v1.2.0
→ APP_URL: http://localhost
```

7. Selanjutnya kita akan buat datatables untuk kategori

`php artisan datatables:make Kategori`

```
C:\laragon\www\PWL_2025\Week5\JS5>php artisan datatables:make Kategori
```

```
INFO DataTable [C:\laragon\www\PWL_2025\Week5\JS5\app\DataTables\KategoriDataTable.php] created successfully
```

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

```
<?php
namespace App\DataTables;

use App\Models\KategoriModel;
use Illuminate\Database\Eloquent\Builder as QueryBuilder;
use Yajra\DataTables\EloquentDataTable;
use Yajra\DataTables\Html\Builder as HtmlBuilder;
use Yajra\DataTables\Html\Column;
use Yajra\DataTables\Html\Editor\Editor;
use Yajra\DataTables\Html\Editor\Field;
use Yajra\DataTables\Services\DataTable;
class KategoriDataTable extends DataTable
{
```




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

/**
 * Get the query source of dataTable.
 */
public function query(KategoriModel $model): QueryBuilder
{
    return $model->newQuery();
}

/**
 * Optional method if you want to use the html builder.
 */
public function html(): HtmlBuilder
{
    return $this->builder()
        ->setTableId('kategori-table')
        ->columns($this->getColumns())
        ->minifiedAjax()
        // ->dom('Bfrtip')
        ->orderBy(1)
        ->selectStyleSingle()
        ->buttons([

            Button::make('excel'),
            Button::make('csv'),
            Button::make('pdf'),
            Button::make('print'),
            Button::make('reset'), Button::make('reload')

        ]);
}
/**
 * Get the dataTable columns definition.
 */
public function getColumns(): array
{
    return [
/*
        Column::computed('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'),
    ];
}
/**
 * Get the filename for export.
 */
protected function filename(): string
{
    return 'Kategori_' . date('YmdHis');
}
```



```
Week5 > JS5 > 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\Html\Editor\Editor;
12 use Yajra\DataTables\Html\Editor\Field;
13 use Yajra\DataTables\Services\DataTable;
14
15 2 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(model: $query))
27             // ->addColumn('action', 'kategori.action')
28             ->setRowId(content: 'id');
29     }
30
31     /**
32      * Get the query source of dataTable.
33      */
34     0 references | 0 overrides
35     public function query(KategoriModel $model): QueryBuilder
36     {
37         return $model->newQuery();
38     }
39
40     /**
41      * Get the dataTable columns definition.
42      */
43     1 reference | 0 overrides
44     public function getColumns(): array
45     {
46         return [
47             // Column::computed('action')
48             // ->exportable(false)
49             // ->printable(false)
50             // ->width(60)
51             // ->addClass('text-center'),
52             Column::make('kategori_id'),
53             Column::make('kategori_kode'),
54             Column::make('kategori_nama'),
55             Column::make('created_at'),
56             Column::make('updated_at'),
57         ];
58     }
59
60     /**
61      * Get the filename for export.
62      */
63     0 references | 0 overrides
64     protected function filename(): string
65     {
66         return 'Kategori_' . date(format: 'YmdHis');
67     }
68 }
```

9. Ubah kategori model, sesuaikan seperti beriku

➤ KategoriModel.php

```
Week5 > JS5 > app > Models > KategoriModel.php > PHP Intelephense > KategoriModel
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Model;
6 use Illuminate\Database\Eloquent\Relations\HasMany;
7
8 2 references | 0 implementations
9 class KategoriModel extends Model
10 {
11     0 references
12     protected $table = 'm_kategori';
13     0 references
14     protected $primaryKey = 'kategori_id';
15
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(BarangModel::class, 'barang_id', 'barang_id');
23     }
24 }
```

➤ BarangModel.php

```
Week5 > JS5 > app > Models > BarangModel.php > PHP > BarangModel
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\Relations\HasMany;
8
9 1 reference | 0 implementations
10 class BarangModel extends Model
11 {
12     use HasFactory;
13 }
```



10. Ubah Kategori Controller sesuai dengan skrip berikut:

```
Week5 > JS5 > app > Http > Controllers > KategoriController.php > PHP Intelephense > KategoriController
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\DataTables\KategoriDataTable;
7
8  7 references | 0 implementations
9  class KategoriController extends Controller
10
11  3 references | 0 overrides
12  public function index(KategoriDataTable $dataTable): mixed
13  {
14      return $dataTable->render('kategori_index');
```

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

```
Week5 > JS5 > resources > views > kategori > index.blade.php > ...
1  @extends('layouts.app')
2
3  {{-- Customize layout sections --}}
4  @section('subtitle', 'Kategori')
5  @section('content_header_title', 'Home')
6  @section('content_header_subtitle', 'Kategori')
7
8  @section('content')
9      <div class="container">
10         <div class="card">
11             <div class="card-header">Manage Kategori</div>
12             <div class="card-body">
13                 {{ $dataTable->table() }}
14             </div>
15         </div>
16     </div>
17 @endsection
18
19 @push('scripts')
20     {{ $dataTable->scripts() }}
21 @endpush
```



12. Pastikan route untuk kategori sudah tersedia

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

13. Sesuaikan app layout seperti kode berikut

```
@extends('adminlte::page')

{{-- Extend and customize the browser title --}} @section('title')
    {{ config('adminlte.title') }} @hasSection('subtitle') |
    @yield('subtitle') @endif
@stop @vite('resources/js/app.js')

{{-- Extend and customize the page content header --}}

@section('content_header') @hasSection('content_header_title')
    <h1 class="text-muted">
        @yield('content_header_title')

        @hasSection('content_header_subtitle')
            <small class="text-dark">
                <i class="fas fa-xs fa-angle-right text-muted"></i> @yield('content_header_subtitle')
            </small> @endif
    </h1>
@endif
@stop

{{-- Rename section content to content_body --}}

@section('content')
    @yield('content_body')
@stop

{{-- Create a common footer --}} @section('footer')
    <div class="float-right">
        Version: {{ config('app.version', '1.0.0') }}
    </div>

    <strong>
        <a href="{{ config('app.company_url', '#') }}">
            {{ config('app.company_name', 'My company') }}
        </a>
    </strong>
@stop

{{-- Add common Javascript/Jquery code --}}

@push('js')
<script src="https://cdn.datatables.net/2.0.2/js/dataTables.js"></script> @endpush

@stack('scripts')

{{-- Add common CSS customizations --}}

@push('css')
```



```
<link rel="stylesheet"
href="https://cdn.datatables.net/2.0.2/css/dataTables.dataTables.css" />

<style type="text/css">

    {{-- You can add AdminLTE customizations here --}}
    /*
    .card-header {
        border-bottom: none;
    }
    .card-title {
        font-weight: 600;
    }
    */

</style>

@endpush
```

```
Week5 > JS > resources > views > layouts > app.blade.php > ...
1  @extends('adminlte::page')
2
3  {{-- Extend and customize the browser title --}}
4  @section('title')
5      {{ config(key: 'adminlte.title') }}
6  @hasSection('subtitle') | @yield('subtitle') @endif
7  @stop
8
9  @vite('resources/js/app.js')
10
11 {{-- Extend and customize the page content header --}}
12 @section('content_header')
13     @hasSection('content_header_title')
14         <h1 class="text-muted">
15             @yield('content_header_title')
16             @hasSection('content_header_subtitle')
17                 <small class="text-dark">
18                     <i class="fas fa-xx fa-angle-right text-muted"></i>
19                     @yield('content_header_subtitle')
20                 </small>
21             @endif
22         </h1>
23     @endif
24 @stop
25
26 {{-- Rename section content to content_body --}}
27 @section('content')
28     @yield('content_body')
29 @stop
30
31 {{-- Create a common footer --}}
32 @section('footer')
33     <div class="float-right">
34         Versions: {{ config(key: 'app.version', default: '1.0.0') }}
35     </div>
36     <strong>
37         <a href="{{ config(key: 'app.company_url', default: '#') }}">
38             {{ config(key: 'app.company_name', default: 'My company') }}
39         </a>
40     </strong>
41 @stop
42
43 {{-- Add common JavaScript/Query code --}}
44 @push('js')
45     <script src="https://cdn.datatables.net/2.0.2/js/dataTables.js"></script>
46 @endpush
47 @stack('scripts')
48
49 {{-- Add common CSS customizations --}}
50 @push('css')
51     <link rel="stylesheet" href="https://cdn.datatables.net/2.0.2/css/dataTables.dataTables.css" />
52     <style type="text/css">
53         /* Tambahkan kustomisasi AdminLTE di sini */
54         /*
55         .card-header {
56             border-bottom: none;
57         }
58         .card-title {
59             font-weight: 600;
60         }
61         */
62     </style>
63 @endpush
```



14. Menset ViteJs / script type defaults

```
Week5 > JS5 > app > Providers > AppServiceProvider.php > ...
1  <?php
2
3  namespace App\Providers;
4
5  use Illuminate\Support\ServiceProvider;
6  use Yajra\DataTables\Html\Builder;
7
8  6 references | 0 implementations
9  class AppServiceProvider extends ServiceProvider
10 {
11     /**
12      * Register any application services.
13      */
14     0 references | 0 overrides
15     public function register(): void
16     {
17         //
18     }
19
20     /**
21      * Bootstrap any application services.
22      */
23     0 references | 0 overrides
24     public function boot(): void
25     {
26         Builder::useVite();
27     }
28 }
```

15. Isikan beberapa data ke table kategori

```
1 INSERT INTO m_kategori (kategori_kode, kategori_nama, created_at, updated_at)
2 VALUES
3 ('CML','Cemilan', '2024-03-09 10:30:00', NULL),
4 ('MNR', 'Minuman Ringan', '2024-03-09 13:01:39', NULL);|
```

		kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	FB	Food & Beverage	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	BH	Beauty & Health	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	HC	Home Care	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	BK	Baby & Kids	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	EL	Electronics	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	7	CML	Cemilan	2024-03-09 10:30:00	NULL
<input type="checkbox"/>	Edit Copy Delete	8	MNR	Minuman Ringan	2024-03-09 13:01:39	NULL

⬅️ ➡️ ☐ Check all With selected: Edit Copy Delete Export

16. Datatables sudah dapat di load di /kategori

AdminLTE

search

Pages 4

ACCOUNT SETTINGS

Profile

Change Password

Multi Level

LABELS

Important

Warning

Information

Home > Kategori

Manage Kategori

10 entries per page Search:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At
1	FB	Food & Beverage		
2	BH	Beauty & Health		
3	HC	Home Care		
4	BK	Baby & Kids		
5	EL	Electronics		
7	CML	Cemilan	2024-03-09T10:30:00.000000Z	
8	MNR	Minuman Ringan	2024-03-09T13:01:39.000000Z	

Showing 1 to 7 of 7 entries



Praktikum 3 – Membuat form kemudian menyimpan data dalam database

Langkah-langkahnya adalah sebagai berikut:

1. Menyesuaikan routing, tambahkan dua routing berikut

```
37 Route::get('/kategori/create', [KategoriController::class, 'create']);  
38 Route::post('/kategori', [KategoriController::class, 'store']);
```

2. Tambahkan dua function berikut dalam KategoriController

```
Week5 > JS5 > app > Http > Controllers > KategoriController.php > PHP Intelephense > KategoriController  
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  
8 references | 0 implementations  
9 class KategoriController extends Controller  
10 {  
11     3 references | 0 overrides  
12     public function index(KategoriDataTable $dataTable): mixed  
13     {  
14         return $dataTable->render('kategori.index');  
15     }  
16  
16     1 reference | 0 overrides  
17     public function create(): Factory|View  
18     {  
19         return view(view: 'kategori.create');  
20     }  
21  
21     1 reference | 0 overrides  
22     public function store(Request $request): Redirector|RedirectResponse  
23     {  
24         KategoriModel::create([  
25             'kategori_kode' => $request->kodeKategori,  
26             'kategori_nama' => $request->namaKategori,  
27         ]);  
28         return redirect(to: '/kategori');  
29     }  
30  
31 }
```



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

```
Week5 > JS5 > resources > views > kategori > create.blade.php > ...
1  @extends('layouts.app')
2
3  {{-- Customize layout sections --}}
4  @section('subtitle', 'Kategori')
5  @section('content_header_title', 'Kategori')
6  @section('content_header_subtitle', 'Create')
7
8  {{-- Content body: main page content --}}
9  @section('content')
10
11  <div class="container">
12      <div class="card card-primary">
13          <div class="card-header">
14              <h3 class="card-title">Buat kategori baru</h3>
15          </div>
16
17          <form method="post" action="/kategori">
18              <div class="card-body">
19                  <div class="form-group">
20                      <label for="kodeKategori">Kode Kategori</label>
21                      <input type="text" class="form-control" id="kodeKategori" name="kodeKategori" placeholder="">
22                  </div>
23
24                  <div class="form-group">
25                      <label for="namaKategori">Nama Kategori</label>
26                      <input type="text" class="form-control" id="namaKategori" name="namaKategori" placeholder="">
27                  </div>
28              </div>
29
30              <div class="card-footer">
31                  <button type="submit" class="btn btn-primary">Submit</button>
32              </div>
33          </form>
34      </div>
35  </div>
36
37  @endsection
```

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

```
Week5 > JS5 > app > Http > Middleware > VerifyCsrfToken.php > ...
1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Illuminate\Foundation\Http\Middleware\VerifyCsrfToken as Middleware;
6
7  12 references | 0 implementations
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     0 references
16     protected $except = [
17         '/kategori'
18     ];
19 }
```




5. Akses kategori/create

Kategori > Create

Buat kategori baru

Kode Kategori

Nama Kategori

Submit

6. Halaman kategori

Home > Kategori

Manage Kategori

10 entries per page

Search:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At
1	FB	Food & Beverage		
2	BH	Beauty & Health		
3	HC	Home Care		
4	BK	Baby & Kids		
5	EL	Electronics		
7	CML	Cemilan	2024-03-09T10:30:00.000000Z	
8	MNR	Minuman Ringan	2024-03-09T13:01:39.000000Z	
9	SPT	Sepatu	2025-03-17T16:07:25.000000Z	2025-03-17T16:07:25.000000Z

Showing 1 to 8 of 8 entries

1



Tugas Praktikum

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

➤ Tambahkan route

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

➤ Tambahkan pada KategoriController

```
16 public function create(): Factory|View  
17 {  
18     return view(view: 'kategori.create');  
19 }  
20  
21 1 reference | 0 overrides  
22 public function store(Request $request): RedirectResponse  
23 {  
24     $request->validate([  
25         'kodeKategori' => 'required|max:10',  
26         'namaKategori' => 'required|max:100',  
27     ]);  
28     KategoriModel::create([  
29         'kategori_kode' => $request->kodeKategori,  
30         'kategori_nama' => $request->namaKategori,  
31     ]);  
32     return redirect()->route('kategori.index')->with('success', 'Kategori berhasil ditambahkan.');
```

➤ Tambahkan tombol Add(+ Tambah Kategori)

```
Week5 > JS5 > resources > views > kategori > index.blade.php > ...  
1 @extends('layouts.app')  
2  
3 {{-- Customize layout sections --}}  
4 @section('subtitle', 'Kategori')  
5 @section('content_header_title', 'Home')  
6 @section('content_header_subtitle', 'Kategori')  
7  
8 @section('content')  
9     <div class="container">  
10         <div class="card">  
11             <div class="card-header">Manage Kategori  
12             <a href="{{ route('kategori.create') }}" class="btn btn-primary float-right">+ Tambah Kategori /a>  
13             </div>  
14             <div class="card-body">  
15                 {{ $dataTable->table(['class' => 'table table-bordered table-striped']) }}  
16             </div>  
17         </div>  
18     </div>  
19 @endsection
```



➤ Halaman Create Kategori Baru

```
Week5 > JS5 > resources > views > kategori > create.blade.php > div.container > div.card.card-primary > form > div.card-footer
1 @extends('layouts.app')
2
3 @section('subtitle', 'Kategori')
4 @section('content_header_title', 'Kategori')
5 @section('content_header_subtitle', 'Create')
6
7 @section('content')
8 <div class="container">
9 <div class="card card-primary">
10 <div class="card-header">
11 <h3 class="card-title">Buat Kategori Baru</h3>
12 </div>
13
14 <form method="post" action="{{ route('kategori.store') }}">
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" required>
20 </div>
21
22 <div class="form-group">
23 <label for="namaKategori">Nama Kategori</label>
24 <input type="text" class="form-control" id="namaKategori" name="namaKategori" required>
25 </div>
26 </div>
27
28 <div class="card-footer">
29 <button type="submit" class="btn btn-primary">Simpan</button>
30 <a href="{{ route('kategori.index') }}" class="btn btn-secondary">Kembali</a>
31 </div>
32 </form>
33 </div>
34 </div>
35 @endsection
```

➤ Hasil

Home > Kategori

Manage Kategori

10 entries per page

Search:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At
1	FB	Food & Beverage		
2	BH	Beauty & Health		
3	HC	Home Care		
4	BK	Baby & Kids		
5	EL	Electronics		
7	CML	Cemilan	2024-03-09T10:30:00.000000Z	
8	MNR	Minuman Ringan	2024-03-09T13:01:39.000000Z	
9	SPT	Sepatu	2025-03-17T16:07:25.000000Z	2025-03-17T16:07:25.000000Z

Showing 1 to 8 of 8 entries

Kategori > Create

Buat Kategori Baru

Kode Kategori

ACS

Nama Kategori

Accessories

Simpan Kembali



Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At
1	FB	Food & Beverage		
2	BH	Beauty & Health		
3	HC	Home Care		
4	BK	Baby & Kids		
5	EL	Electronics		
7	CML	Cemilan	2024-03-09T10:30:00.000000Z	
8	MNR	Minuman Ringan	2024-03-09T13:01:39.000000Z	
9	SPT	Sepatu	2025-03-17T16:07:25.000000Z	2025-03-17T16:07:25.000000Z
10	ACS	Accessories	2025-03-18T03:46:17.000000Z	2025-03-18T03:46:17.000000Z

2. Tambahkan menu untuk halaman manage kategori, di daftar menu navbar
 - Tambahkan menu Manage Kategori di navbar (sidebar menu) menggunakan Laravel AdminLTE.

```
'menu' => [  
    // Navbar items:  
    [  
        'text' => 'Manage Kategori',  
        'url' => '/kategori',  
        'icon' => 'fas fa-tags', // Icon kategori  
    ],  
]
```

- Hasil

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At
1	FB	Food & Beverage		
2	BH	Beauty & Health		
3	HC	Home Care		
4	BK	Baby & Kids		
5	EL	Electronics		
7	CML	Cemilan	2024-03-09T10:30:00.000000Z	
8	MNR	Minuman Ringan	2024-03-09T13:01:39.000000Z	
9	SPT	Sepatu	2025-03-17T16:07:25.000000Z	2025-03-17T16:07:25.000000Z
10	ACS	Accessories	2025-03-18T03:46:17.000000Z	2025-03-18T03:46:17.000000Z

3. Tambahkan action edit di datatables dan buat halaman edit serta controllernya
 - Tambahkan fungsi pada KategoriController

```
0 references | 0 overrides  
public function edit($id): Factory|View  
{  
    $data = KategoriModel::findOrFail($id);  
    return view('kategori.edit', data: ['kategori' => $data]);  
}  
  
0 references | 0 overrides  
public function update(Request $request, $id): Redirector|RedirectResponse  
{  
    $request->validate([  
        'kodeKategori' => 'required',  
        'namaKategori' => 'required'  
    ]);  
  
    $kategori = KategoriModel::findOrFail($id);  
  
    // Update data  
    $kategori->kategori_kode = $request->kodeKategori;  
    $kategori->kategori_nama = $request->namaKategori;  
  
    $kategori->save();  
  
    return redirect(to: '/kategori');  
}
```



➤ Tambahkan kolom edit di datatable

```
Week5 > JSS > app > DataTables > KategoriDataTable.php > PHP IntelliSense > KategoriDataTable > getColumns
15 class KategoriDataTable extends DataTable
16 {
17     // references | 0 overrides
18     public function dataTable(QueryBuilder $query): EloquentDataTable
19     {
20         return (new EloquentDataTable(model: $query))
21             // ->addColumn('action', 'kategori.action')
22             ->addColumn(name: 'action', content: function($id): string {
23                 $edit = route('kategori.edit', parameters: $id);
24                 return '
25                     <a href="' . $edit . '" class="btn btn-warning btn-sm">Edit</a>
26                 ';
27             })
28             ->setRowId(content: 'id');
29     }
30
31     /**
32      * Get the query source of dataTable.
33      */
34     // references | 0 overrides
35     public function query(KategoriModel $model): QueryBuilder
36     {
37         return $model->newQuery();
38     }
39
40     /**
41      * Optional method if you want to use the html builder.
42      */
43     // references | 0 overrides
44     public function html(): HtmlBuilder
45     {
46         return $this->builder()
47             ->setTableId('kategori-table')
48             ->columns($this->getColumns())
49             ->minifiedAjax()
50             // ->dom('Bfrt')
51             ->orderBy(1)
52             ->selectStyleSingle()
53             ->buttons([
54                 Button::make('excel'),
55                 Button::make('csv'),
56                 Button::make('pdf'),
57                 Button::make('print'),
58                 Button::make('reset'),
59                 Button::make('reload')
60             ]);
61     }
62
63     /**
64      * Get the dataTable columns definition.
65      */
66     // references | 0 overrides
67     public function getColumns(): array
68     {
69         return [
70             // Column::computed('action')
71             // ->exportable(false)
72             // ->printable(false)
73             // ->width(60)
74             // ->addClass('text-center'),
75             Column::make('kategori_id'),
76             Column::make('kategori_kode'),
77             Column::make('kategori_nama'),
78             Column::make('created_at'),
79             Column::make('updated_at'),
80
81             Column::computed('action')
82             ->exportable(false)
83             ->printable(false)
84             ->width(100)
85             ->addClass('text-center'),
86         ];
87     }
88 }
```

➤ Tambahkan edit blade baru

```
Week5 > JSS > resources > views > kategori > edit.blade.php > _
1 @extends('layouts.app')
2
3 @section('content')
4     <div class="container-fluid">
5         <div class="row">
6             <div class="col-md-12">
7                 <div class="card card-primary">
8                     <div class="card-header">
9                         <h3 class="card-title">Edit Kategori</h3>
10                     </div>
11
12                     <form action="{{ route('kategori.update', parameters: $kategori->kategori_id) }}" method="POST">
13                         @csrf
14                         @method('PUT')
15
16                         <div class="card-body">
17                             <div class="form-group">
18                                 <label for="kodeKategori">Kode Kategori</label>
19                                 <input type="text" class="form-control" name="kodeKategori" id="kodeKategori"
20                                     value="{{ $kategori->kategori_kode }}">
21                             </div>
22
23                             <div class="form-group">
24                                 <label for="namaKategori">Nama Kategori</label>
25                                 <input type="text" class="form-control" name="namaKategori" id="namaKategori"
26                                     value="{{ $kategori->kategori_nama }}">
27                             </div>
28                         </div>
29
30                         <div class="card-footer">
31                             <button type="submit" class="btn btn-primary">Simpan</button>
32                         </div>
33                     </form>
34                 </div>
35             </div>
36         </div>
37     </div>
38 @endsection
```



- Tambahkan pada index.blade.php

```
21 @push('scripts')
22 {{ $dataTable->scripts() }}
23 <script>
24 $(document).ready(function () {
25     $('#kategoriTable').on('draw.dt', function () {
26         $('.btn-edit').on('click', function () {
27             var id = $(this).data('id');
28             window.location.href = "{{ url('kategori') }}/" + id + "/edit";
29         });
30     });
31 </script>
32 @endpush
```

- Tambahkan

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

- Hasil

Home > Kategori

Manage Kategori

10 entries per page

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At	Action
1	FB	Food & Beverage			Edit
2	BH	Beauty & Health			Edit
3	HIC	Home Care			Edit
4	BK	Baby & Kids			Edit
5	EL	Electronics			Edit
7	CML	Cemilan	2024-03-09T10:30:00.000000Z		Edit
8	MNR	Minuman Ringan	2024-03-09T13:01:39.000000Z		Edit
9	SPT	Sepatu	2025-03-17T16:07:25.000000Z	2025-03-17T16:07:25.000000Z	Edit
10	ACS	Accessories	2025-03-18T03:46:17.000000Z	2025-03-18T03:46:17.000000Z	Edit

Showing 1 to 9 of 9 entries

Edit Kategori

Kode Kategori: AS

Nama Kategori: Accessories

Simpan

Home > Kategori

Manage Kategori

10 entries per page

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At	Action
1	FB	Food & Beverage			Edit
2	BH	Beauty & Health			Edit
3	HIC	Home Care			Edit
4	BK	Baby & Kids			Edit
5	EL	Electronics			Edit
7	CML	Cemilan	2024-03-09T10:30:00.000000Z		Edit
8	MNR	Minuman Ringan	2024-03-09T13:01:39.000000Z		Edit
9	SPT	Sepatu	2025-03-17T16:07:25.000000Z	2025-03-17T16:07:25.000000Z	Edit
10	AS	Accessories	2025-03-18T03:46:17.000000Z	2025-03-18T03:46:00.000000Z	Edit

Showing 1 to 9 of 9 entries



4. Tambahkan action delete di datatables serta controllernya

➤ Tambahkan pada Datatable

```
public function dataTable(QueryBuilder $query): EloquentDataTable
{
    return (new EloquentDataTable(model: $query))
        // ->addColumn('action', 'kategori.action')
        ->addColumn(name: 'action', content: function ($id): string {
            $edit = route(name: 'kategori.edit', parameters: $id);
            $delete = route(name: 'kategori.delete', parameters: $id);
            return "
                <a href='\"$edit\"' class='\"btn btn-warning btn-sm\">Edit</a>
                <a href='\"$delete\"' class='\"btn btn-danger btn-sm\" onclick='\"return confirm(\\\"Apakah Anda yakin ingin menghapus kategori ini?\\\")\">Delete</a>
            ";
        })
        ->setRowId(content: 'id');
}
```

➤ Tambahkan pada KategoriController

```
public function delete($id): Redirector|RedirectResponse{
    KategoriModel::where('kategori_id',$id)->delete();
    return redirect(to: '/kategori');
}
```

➤ Tambahkan Route

```
43 Route::get('/kategori/delete/{id}', [KategoriController::class, 'delete']->name('kategori.delete'));
```

➤ Hasil

127.0.0.1:8000 says
Apakah Anda yakin ingin menghapus kategori ini?

OK Cancel

Home > Kategori

Manage Kategori

10 entries per page Search:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At	Action
1	FB	Food & Beverage			Edit Delete
2	BH	Beauty & Health			Edit Delete
3	HC	Home Care			Edit Delete
4	BK	Baby & Kids			Edit Delete
5	EL	Electronics			Edit Delete
7	CML	Cemilan	2024-03-09T10:30:00.000000Z		Edit Delete
8	MNR	Minuman Ringan	2024-03-09T13:01:39.000000Z		Edit Delete
9	SPT	Sepatu	2025-03-17T16:07:25.000000Z	2025-03-17T16:07:25.000000Z	Edit Delete
10	AS	Accessories	2025-03-18T03:46:17.000000Z	2025-03-18T05:03:46.000000Z	Edit Delete

Showing 1 to 9 of 9 entries

Home > Kategori

Manage Kategori

10 entries per page Search:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At	Action
1	FB	Food & Beverage			Edit Delete
2	BH	Beauty & Health			Edit Delete
3	HC	Home Care			Edit Delete
4	BK	Baby & Kids			Edit Delete
5	EL	Electronics			Edit Delete
7	CML	Cemilan	2024-03-09T10:30:00.000000Z		Edit Delete
8	MNR	Minuman Ringan	2024-03-09T13:01:39.000000Z		Edit Delete
9	SPT	Sepatu	2025-03-17T16:07:25.000000Z	2025-03-17T16:07:25.000000Z	Edit Delete

Showing 1 to 8 of 8 entries

References

1. <https://github.com/jeroennoten/Laravel-AdminLTE/wiki/Usage>
2. <https://yajrabox.com/docs/laravel-datatables/10.0/quick-starter>