



Mata Kuliah : Pemrograman Web Lanjut (PWL)
Program Studi : D4 – Sistem Informasi Bisnis
Semester : 4 (empat)
Pertemuan ke- : 5 (lima)
Nama : Aqila Nur Azza

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 methodi 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.



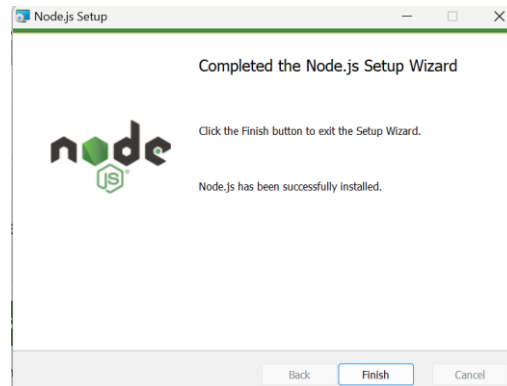
```
@extends('layouts.app')  
// kode..  
@component("alert")  
<b>Tulisan ini akan mengisi variabel $slot</b>  
@endcomponent
```

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

```
C:\Users\Aqila Nur>node -v  
v22.14.0
```

npm -v

```
C:\Users\Aqila Nur>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

```
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

51 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

Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext/sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0

Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext/pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext\php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0

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 for laravel/ui
```

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



```
php artisan ui bootstrap
```

```
php artisan ui bootstrap --auth
```

```
INFO Bootstrap scaffolding installed successfully.  
WARN Please run [npm install && npm run dev] to compile your fresh scaffolding.
```

```
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, install dependensi frontend proyek Anda menggunakan Node package manager (NPM) :

```
npm install
```

```
C:\laragon\www\PWL_2025\PWL_WEEK5\PWL_POS>npm install  
added 53 packages, and audited 54 packages in 21s  
  
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
```

```
VITE v5.4.14 ready in 569 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 menghubungkan 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`

```
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

Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext/sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext/php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0

Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext/pdo_sqlsrv (The specified module could not be found), C:/laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext/php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0

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`

```
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`



```
{-} 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         "laravel/ui": "^4.6"
15     },
16     "require-dev": {
17         "fakerphp/faker": "^1.9.1",
18         "laravel/pint": "^1.0",
19         "laravel/sail": "^1.18",
20         "mockery/mockery": "^1.4.4",
21         "nunomaduro/collision": "^7.0",
22         "phpunit/phpunit": "^10.1",
23         "spatie/laravel-ignition": "^2.0"
24     },
25     "autoload": {
26         "psr-4": {
27             "App\\": "app/",
28             "Database\\Factories\\": "database/factories/",
29             "Database\\Seeders\\": "database/seeders/"
30         }
31     },
32     "autoload-dev": {
33         "psr-4": {
34             "Tests\\": "tests/"
35         }
36     },
37 }
```



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>
    @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>

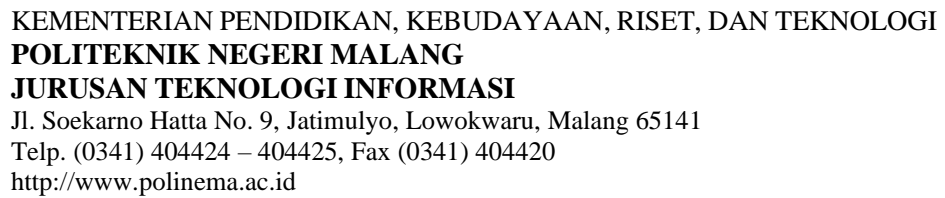
    $(document).ready(function() {
        // Add your common script logic here...
    });

</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
```



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

Hal. 13 / 26



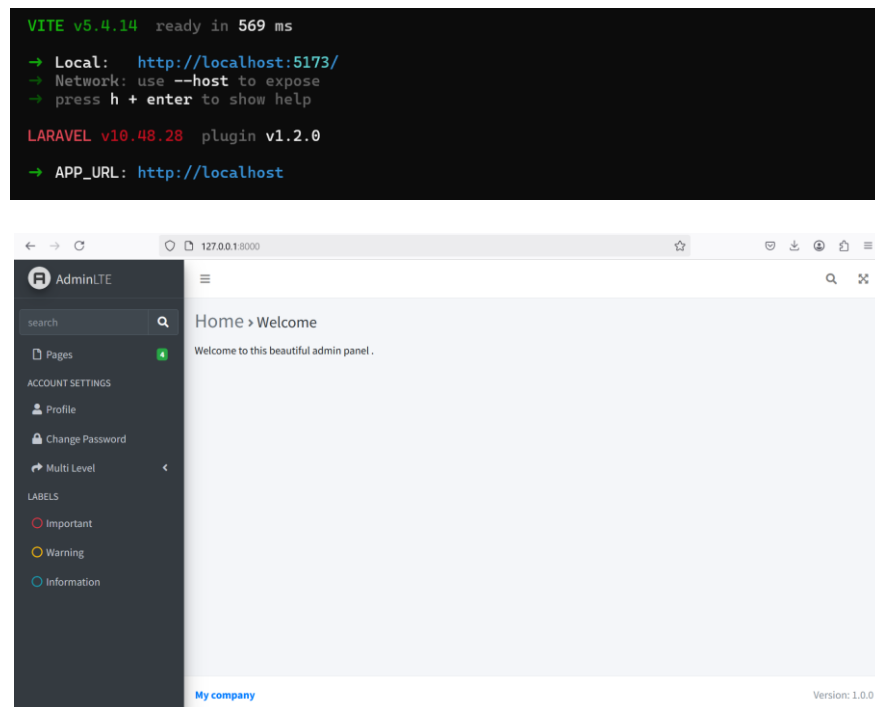
Untuk menggunakan blade template ini cukup dengan extend **AdminLTE layout** dengan cara `@extends('adminlte::page')`.

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.

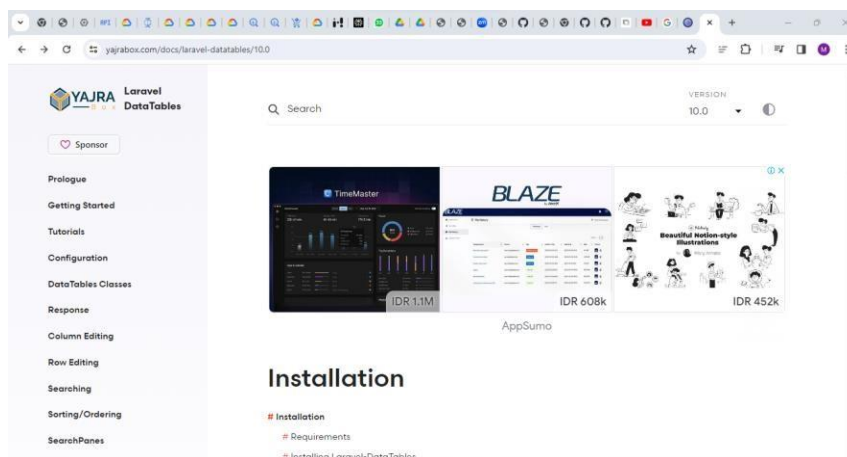




Praktikum 2 – Integrasi dengan DataTables

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`

```
Found 1 security vulnerability advisory affecting 1 package.  
Run "composer audit" for a full list of advisories.  
Using version ^4.6 for laravel/ui
```

`composer require yajra/laravel-datatables:^10.0`

```
INFO: Discovering packages.  
jeroennoten/laravel-adminlte ..... DONE  
laravel/sail ..... DONE  
laravel/sanctum ..... DONE  
laravel/ui ..... DONE  
mesobot/carbon ..... DONE  
nunomaduro/collision ..... DONE  
nunomaduro/termwind ..... DONE  
spatie/laravel-ignition ..... DONE  
yajra/laravel-datatables-buttons ..... DONE  
yajra/laravel-datatables-editor ..... DONE  
yajra/laravel-datatables-fractal ..... DONE  
yajra/laravel-datatables-html ..... DONE  
yajra/laravel-datatables-oracle ..... DONE  
  
No 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  
  
Warning: PHP Startup: Unable to load dynamic library 'sqlsrv' (tried: C:/Laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext/sqlsrv (The specified module could not be found), C:/Laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext/php_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0  
  
Warning: PHP Startup: Unable to load dynamic library 'pdo_sqlsrv' (tried: C:/Laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext/pdo_sqlsrv (The specified module could not be found), C:/Laragon/bin/php/php-8.1.10-Win32-vs16-x64/ext/php_pdo_sqlsrv.dll (The specified module could not be found)) in Unknown on line 0  
  
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.
```

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



```
C:\laragon\www\PWL_2025\PWL_WEEK5\PWL_POS>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:

access, adduser, audit, bugs, cache, ci, completion,
config, dedupe, deprecate, diff, dist-tag, docs, doctor,
edit, exec, explain, explore, find-dupes, fund, get, help,
help-search, hook, init, install, install-ci-test,
install-test, link, ll, login, logout, ls, org, outdated,
owner, pack, ping, pkg, prefix, profile, prune, publish,
query, rebuild, repo, restart, root, run-script, sbom,
search, set, shrinkwrap, star, stars, start, stop, team,
test, token, uninstall, unpublish, unstar, update, version,
view, whoami

Specify configs in the ini-formatted file:
  C:\Users\Aqila Nur\.npmrc
or on the command line via: npm <command> --key=value

More configuration info: npm help config
Configuration fields: npm help 7 config

npm@10.9.2 C:\Program Files\nodejs\node_modules\npm
```

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

Npm digunakan untuk manajemen package javascript.

3. Install Laravel DataTables Vite dan sass

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

```
C:\laragon\www\PWL_2025\PWL_WEEK5\PWL_POS>npm i laravel-datatables-vite --save-dev
added 9 packages, and audited 63 packages in 16s

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\PWL_WEEK5\PWL_POS>npm install -D sass
up to date, audited 63 packages in 3s

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



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

5. Buatlah file resources/saas/app.scss

```
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';  
12
```

6. Jalankan dengan `npm run dev`

```
VITE v5.4.14 ready in 569 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`

```
INFO DataTable [C:\laragon\www\PWL_2025\PWL_WEEK5\PWL_POS\ap  
p\DataTables\KategoriDataTable.php] created successfully.
```

8. Kita edit KategoriDatable untuk mengatur kolom apasaja 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\Fields;
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([
```



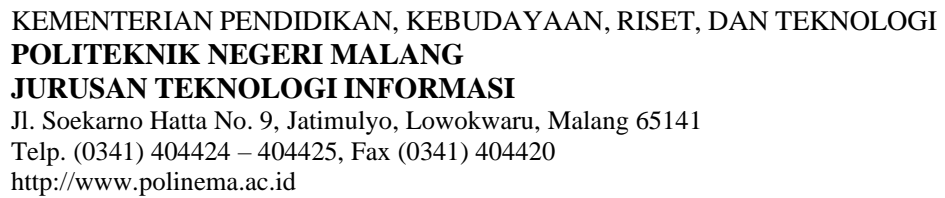
```
        Button::make('excel'),
        Button::make('csv'),
        Button::make('pdf'),
        Button::make('print'),
        Button::make('reset'),
        Button::make('reload')

    });

}

/**
 * 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');
}
}
```



9. Ubah kategori model, sesuaikan seperti berikut

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

```
app > Models > BarangModel.php > BarangModel
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class BarangModel extends Model
9  {
10     use HasFactory;
11 }
```



10. Ubah Kategori Controller sesuaikan dengan skrip berikut:

```
app > Http > Controllers > KategoriController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\DataTables\KategoriDataTable;
7
8  class KategoriController extends Controller
9  {
10     public function index(KategoriDataTable $dataTable)
11     {
12         return $dataTable->render('kategori.index');
13     }
14 }
```

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('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

```
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
```

```
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') }}
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-xs 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         Version: {{ config('app.version', '1.0.0') }}
35     </div>
36     <strong>
37         <a href="{{ config('app.company_url', '#') }}">
38             {{ config('app.company_name', 'My Company') }}
39         </a>
40     </strong>
41 @stop
42
43 {{-- Add common Javascript/JQuery 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         /* You can add AdminLTE customizations here */
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

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

1. Isikan beberapa data ke table kategori

	kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	FB	Food & Beverage	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	BH	Beauty & Health	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	HC	Home Care	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	BK	Baby & Kids	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	EL	Electronics	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	SNK	Camilan	2025-03-04 03:44:11	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	CML	Cemilan	2024-03-09 10:30:00	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	MNR	Minuman Ringan	2024-03-09 13:01:39	NULL

```
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);
5
```

2. Datatables sudah dapat di load di /kategori



Praktikum 3 – Membuat form kemudian menyimpan data dalam database

Langkah-langkahnya adalah sebagai berikut:

1. Menyesuaikan routing, tambahkan dua routing berikut

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

2. Tambahkan dua function berikut dalam KategoriController

```
app > Http > Controllers > KategoriController.php > ...  
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  class KategoriController extends Controller  
10 {  
11     public function index(KategoriDataTable $dataTable)  
12     {  
13         return $dataTable->render('kategori.index');  
14     }  
15  
16     public function create()  
17     {  
18         return view('kategori.create');  
19     }  
20  
21     public function store(Request $request)  
22     {  
23         KategoriModel::create([  
24             'kategori_kode' => $request->kodeKategori,  
25             'kategori_nama' => $request->namaKategori,  
26         ]);  
27  
28         return redirect('/kategori');  
29     }  
30 }  
31
```



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('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
38
```

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

```
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  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       ];
17  }
```



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		
6	SNK	Camilan	2025-03-04T03:44:11.000000Z	
7	CML	Cemilan	2024-03-09T10:30:00.000000Z	
8	MNR	Minuman Ringan	2024-03-09T13:01:39.000000Z	
9	SPT	Sepatu	2024-03-09T07:55:38.000000Z	2024-03-09T07:55:38.000000Z

Showing 1 to 9 of 9 entries



Tugas Praktikum

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

- Menambahkan button pada resource/view/kategori/index.blade.php

```
resources > views > kategori > index.blade.php > div.container > div.card > div.card-body
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() }}
16             </div>
17         </div>
18     </div>
19 @endsection
20
21 @push('scripts')
22 | {{ $dataTable->scripts() }}
23 @endpush
24
```

- Menambahkan route yang mengarah ke resource/view/kategori/create.blade.php

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

- Tampilan Halaman Kategori

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At
9	SPT	Sepatu	2024-03-09T07:55:38.000000Z	2024-03-09T07:55:38.000000Z
6	SNK	Camilan	2025-03-04T03:44:11.000000Z	
8	MNR	Minuman Ringan	2024-03-09T13:01:39.000000Z	
3	HC	Home Care		
1	FB	Food & Beverage		
5	EL	Electronics		
7	CML	Camilan	2024-03-09T10:30:00.000000Z	
4	BK	Baby & Kids		
2	BH	Beauty & Health		

- Tampilan ketika button tambah kategori di pilih

Kategori > Create

Suat kategori baru

Kode Kategori: FSH

Nama Kategori: Fashion

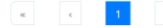
Submit

- Menyimpan kategori baru (Submit)



8	MNR	Minuman Ringan	2024-03-09T13:01:39.000000Z	
9	SPT	Sepatu	2024-03-09T07:55:38.000000Z	2025-03-15T15:21:02.000000Z
10	FSH	Fashion	2025-03-15T15:23:09.000000Z	2025-03-15T15:23:09.000000Z

Showing 1 to 10 of 10 entries

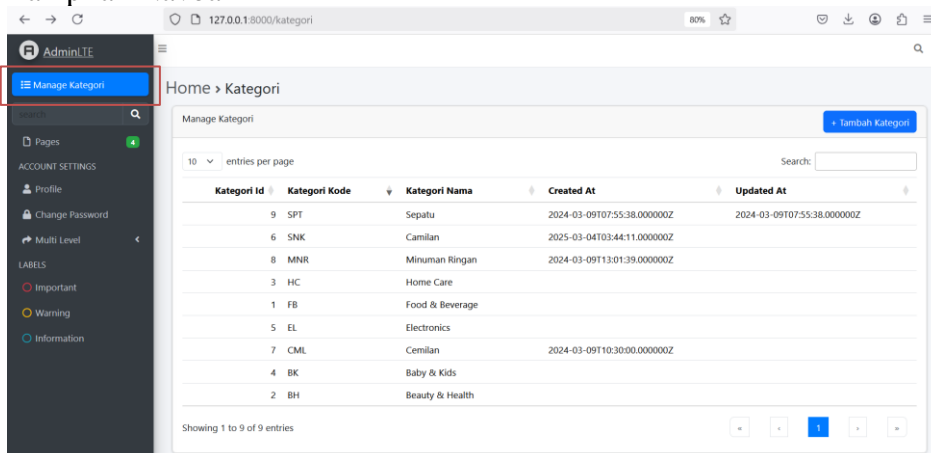


2. Tambahkan menu untuk halaman manage kategori, di daftar menu navbar

- Menambahkan navbar yang diatur pada config/adminlte

```
config > adminlte.php
3 return [
296     // For detailed instructions you can look here:
297     // https://github.com/jeroennoten/Laravel-AdminLTE/wiki/Menu-Configuration
298     //
299     //
300     'menu' => [
301         // Navbar items:
302         [
303             'text' => 'Manage Kategori',
304             'url' => 'kategori',
305             'icon' => 'fas fa-list',
306         ],
307     ],
308     'type' => 'navbar-search',
309     'text' => 'search',
310     'topnav_right' => true,
311 ],
312     'type' => 'fullscreen-widget',
313     'topnav_right' => true,
314 ],
315 ],
316 ],
317 ]
```

- Tampilan Navbar



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

- Menambahkan Fungsi edit dan update pada KategoriController

```
30
31 public function edit($id)
32 {
33     $data = KategoriModel::findOrFail($id);
34     return view('kategori.edit', ['kategori' => $data]);
35 }
36
37 public function update(Request $request, $id)
38 {
39     // Validasi input
40     $request->validate([
41         'kodeKategori' => 'required',
42         'namaKategori' => 'required'
43     ]);
44
45     // Cari data berdasarkan ID
46     $kategori = KategoriModel::findOrFail($id);
47
48     // Update data
49     $kategori->kode_kategori = $request->kodeKategori;
50     $kategori->nama_kategori = $request->namaKategori;
51
52     $kategori->save();
53
54     return redirect('/kategori');
55 }
56
57
58 }
```

- Menambahkan dan mengatur action pada Kontroler Datatables



```
public function dataTable(QueryBuilder $query): EloquentDataTable
{
    return (new EloquentDataTable($query))
        // ->addColumn('action', 'kategori.action')
        ->addColumn('action', function($id){
            $edit = route('kategori.edit', $id);
            return 'a href="'.$edit.'" class="btn btn-warning btn-sm">Edit</a>;
        })
        ->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('BfTtip')
        ->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'),

        Column::computed('action')
        ->exportable(false)
        ->printable(false)
        ->width(100)
        ->addClass('text-center'),
    ];
}
```

➤ Menambahkan edit.blade.php (tampilan form edit)

```
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', $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
```



- Menambahkan action colom datatable pada index.blade.php

```
20
21 @push('scripts')
22 {{ $dataTable->scripts() }}
23
24 {{-- Tambahkan script untuk kolom Action di DataTables --}}
25 <script>
26 $(document).ready(function () {
27     $('#kategoriTable').on('draw.dt', function () {
28         $('.btn-edit').on('click', function () {
29             var id = $(this).data('id');
30             window.location.href = "{{ url('kategori') }}" + id + "/edit";
31         });
32     });
33 }
34 </script>
35 @endpush
```

- Menambahkan route pada web.php

```
39
40 Route::get('/kategori/edit/{id}', [KategoriController::class, 'edit'])->name('kategori.edit');
41 Route::put('/kategori/{id}', [KategoriController::class, 'update'])->name('kategori.update');
42
```

Output :

- Tampilan sebelum diubah :

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At	Action
1	FB	Food & Beverage			Edit
2	BH	Beauty & Health			Edit
3	HC	Home Care			Edit
4	BK	Baby & Kids			Edit
5	EL	Electronics			Edit
6	SNK	Camilan	2025-03-04T03:44:11.000000Z		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	2024-03-09T07:55:38.000000Z	2025-03-15T15:21:02.000000Z	Edit
10	FSH	Fashion	2025-03-15T15:23:09.000000Z	2025-03-15T15:23:09.000000Z	Edit

- Setelah klik button edit akan diarahkan pada edit.blade.php berdasarkan id = 10

Edit Kategori

Kode Kategori: FSH

Nama kategori: Fashion

Simpan

- Tampilan setelah di ubah/edit



Manage Kategori [+ Tambah Kategori](#)

10 entries per page Search:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At	Action
1	FB	Food & Beverage			Edit
2	BH	Beauty & Health			Edit
3	HC	Home Care			Edit
4	BK	Baby & Kids			Edit
5	EL	Electronics			Edit
6	SNK	Camilan	2025-03-04T03:44:11.000000Z		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	2024-03-09T07:55:38.000000Z	2025-03-15T15:21:02.000000Z	Edit
10	FS	Fashion	2025-03-15T15:23:09.000000Z	2025-03-15T15:25:02.000000Z	Edit

Showing 1 to 10 of 10 entries

4. Tambahkan action delete di datatables serta controllernya

➤ Menambahkan action pada KategoriDatatables

```
public function dataTable(QueryBuilder $query): EloquentDataTable
{
    return (new EloquentDataTable($query))
        // >>>addColumn('action', 'kategori.action')
        >>>addColumn('action', function($id){
            $edit = route('kategori.edit', $id);
            $delete = route('kategori.delete', $id);
            return '
                <a href="'.$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('id');
}
```

```
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'),

        Column::computed('action')
            >>exportable(false)
            >>printable(false)
            >>width(150)
            >>addClass('text-center'),
    ];
}
```

➤ Menambahkan Fungsi pada KategoriController

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

➤ Menambahkan route pada web.php

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

Output :

➤ Tampilan sebelum dilakukan penghapusan data :



Manage Kategori + Tambah 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
6	SNK	Camilan	2025-03-04T03:44:11.000000Z		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	2024-03-09T07:55:38.000000Z	2025-03-15T15:21:02.000000Z	Edit Delete
10	FS	Fashion	2025-03-15T15:23:09.000000Z	2025-03-15T15:25:02.000000Z	Edit Delete

Showing 1 to 10 of 10 entries

- Tampilan setelah klik button delete :

Manage Kategori + Tambah 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
6	SNK	Camilan	2025-03-04T03:44:11.000000Z		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	2024-03-09T07:55:38.000000Z	2025-03-15T15:21:02.000000Z	Edit Delete
10	FS	Fashion	2025-03-15T15:23:09.000000Z	2025-03-15T15:25:02.000000Z	Edit Delete

Showing 1 to 10 of 10 entries

- Tampilan setelah dilakukan penghapusan data berdasarkan Id = 10

Manage Kategori + Tambah Kategori

10 entries per page Search:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At	Action
1	FB	Kategori Nama			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
6	SNK	Camilan	2025-03-04T03:44:11.000000Z		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	2024-03-09T07:55:38.000000Z	2025-03-15T15:21:02.000000Z	Edit Delete

Showing 1 to 9 of 9 entries

References

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