

LAPORAN PRAKTIKUM
JOBSHEET V : BLADE VIEW, WEB TEMPLATING (ADMINLTE),
DATATABLES

MATA KULIAH PEMROGRAMAN WEB LANJUT

Dosen Pengampu : Dimas Wahyu Wibowo, S.T., M.T.



Disusun oleh :

Nama : Vita Eka Saraswati

NIM : 2341760082

No. Absen : 29

JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS
POLITEKNIK NEGERI MALANG

2025



Nama : Vita Eka Saraswati
NIM / Kelas : 2341760082 / SIB 2A
No. Absen : 29
Mata Kuliah : Pemrograman Web Lanjut

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
```

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
```

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



```
php artisan ui bootstrap
```

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

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

```
npm install
```

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

```
npm run dev
```

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



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
<http://www.polinema.ac.id>

```
<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
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.
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`

```
PS F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5\Jobsheet5> 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 U x package.json U
Minggu5 > Jobsheet5 > 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     },
```



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

```
app.blade.php U X
Minggus > Jobsheet5 > resources > views > layout > 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             @hasSection('content_header_subtitle')
15                 <small class="text-dark">
16                     <i class="fas fa-xs fa-angle-right text-muted"></i>
17                     @yield('content_header_subtitle')
18                 </small>
19             @endif
20         </h1>
21     @endif
22 @stop
23
24 {{-- Rename section content to content_body --}}
25 @section('content')
26     @yield('content_body')
27 @stop
28
29 {{-- Create a common footer --}}
30 @section('footer')
31     <div class="float-right">
32         Version: {{ config(key: 'app.version', default: '1.0.0') }}
33     </div>
34     <strong>
35         <a href="{{ config(key: 'app.company_url', default: '#') }}">
36             {{ config(key: 'app.company_name', default: 'My company') }}
37         </a>
38     </strong>
39 @stop
40
41 {{-- Add common Javascript/Jquery code --}}
42 @push('js')
43     <script>
44         $(document).ready(function() {
45             // Add your common script logic here...
46         });
47     </script>
48 @endpush
49
50 {{-- Add common CSS customizations --}}
51 @push('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
```



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

```
app.blade.php U welcome.blade.php X
Minggu5 > Jobsheet5 > 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
```

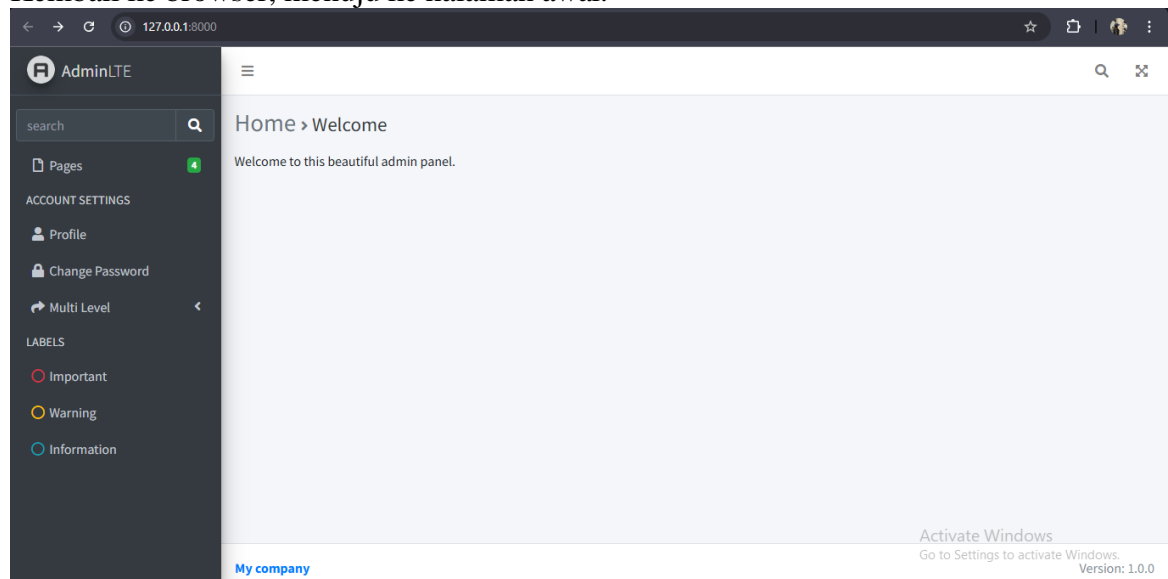
Untuk menggunakan blade template ini cukup dengan extend **AdminLTE layout** dengan cara `@extends('adminlte::page')`.

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.

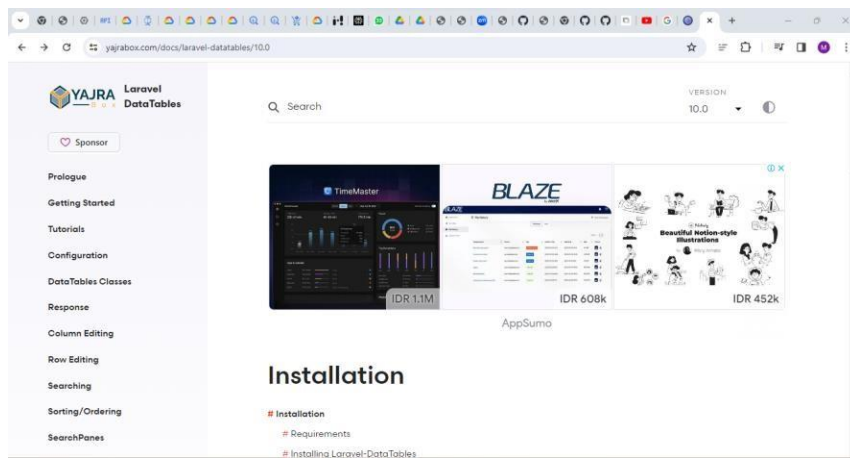




Praktikum 2 – Integrasi dengan DataTables

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`

```
PS F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PM125\Wingus\Jobsheet5> 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)
INFO: Discovering packages.

jeroennoten/laravel-adminlte ..... 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!
```

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

```
PS F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PM125\Wingus\Jobsheet5> composer require yajra/laravel-datatables:^10.0
./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)
```



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

```
PS F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5\Jobsheet5> npm i laravel-datatables-vite --save-dev
added 47 packages, and audited 48 packages in 54s

13 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! 10.8.2 -> 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
```

`npm install -D sass`

```
PS F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5\Jobsheet5> npm install -D sass
added 15 packages, and audited 63 packages in 9s

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

```
JS app.js U X
Minggu5 > Jobsheet5 > 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

```
JS app.js U app.scss U X
Minggu5 > Jobsheet5 > 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`

```
PS F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5\Jobsheet5> npm run dev

> dev
> vite

VITE v5.4.14 ready in 1499 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`

```
PS F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5\Jobsheet5> php artisan datatables:make Kategori

INFO DataTable [F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5\Jobsheet5\app\DataTables\KategoriDataTable.php] created successfully.
```

8. Kita edit KategoriDataTable untuk mengatur kolom apa saja yang ingin ditampilkan

```
KategoriModel.php U KategoriDataTable.php U X
Minggu5 > Jobsheet5 > app > DataTables > KategoriDataTable.php > PHP Intelephense > KategoriDataTable > dataTable

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\Column;
10 use Yajra\DataTables\Services\DataTable;
11
12
13 0 references | 0 implementations
14 class KategoriDataTable extends DataTable
15 {
16     /**
17      * Build the DataTable class.
18      * @param QueryBuilder $query Results from query() method.
19      */
20     0 references | 0 overrides
21     public function dataTable(QueryBuilder $query): EloquentDataTable
22     {
23         return (new EloquentDataTable($query))
24             // ->addColumn('action', 'kategori.action')
25             ->setRowId('id');
26     }
27 }
```




```
27  /**
28   * Get the query source of dataTable.
29   */
30  0 references | 0 overrides
31  public function query(KategoriModel $model): QueryBuilder
32  {
33      return $model->newQuery();
34  }
35
36  /**
37   * Optional method if you want to use the html builder.
38   */
39  0 references | 0 overrides
40  public function html(): HtmlBuilder
41  {
42      return $this->builder()
43          ->setTableId('kategori-table')
44          ->columns($this->getColumns())
45          ->minifiedAjax()
46          // ->dom('Bfritip')
47          ->orderBy(1)
48          ->selectStyleSingle()
49          ->buttons([
50              Button::make('excel'),
51              Button::make('csv'),
52              Button::make('pdf'),
53              Button::make('print'),
54              Button::make('reset'),
55              Button::make('reload'),
56          ]);
57  }
58
59  /**
60   * Get the dataTable columns definition.
61   */
62  1 reference | 0 overrides
63  public function getColumns(): array
64  {
65      return [
66          // Column::computed('action')
67          // ->exportable(false)
68          // ->printable(false)
69          // ->width(60)
70          // ->addClass('text-center'),
71          Column::make('kategori_id'),
72          Column::make('kategori_kode'),
73          Column::make('kategori_nama'),
74          Column::make('created_at'),
75          Column::make('updated_at'),
76      ];
77  }
78
79  /**
80   * Get the filename for export.
81   */
82  0 references | 0 overrides
83  protected function filename(): string
84  {
85      return 'Kategori_' . date(format: 'YmdHis');
86  }
```

9. Ubah kategori model, sesuaikan seperti berikut.

Karena file KategoriModel.php dan BarangModel.php belum ada, buat menggunakan perintah di bawah ini:

```
PS F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5\Jobsheet5> php artisan make:model KategoriModel
>>
[INFO] Model [F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5\Jobsheet5\app\Models\KategoriModel.php] created successfully.
```

```
PS F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5\Jobsheet5> php artisan make:model BarangModel
[INFO] Model [F:\KULIAH\SEMESTER 4\3. PRAK JOBSHEET\PWL25\Minggu5\Jobsheet5\app\Models\BarangModel.php] created successfully.
```



Modifikasi isi dile KategoriModel.php seperti di bawah ini

```
KategoriModel.php U X
Minggu5 > Jobsheet5 > app > Models > KategoriModel.php > PHP > 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     2 references
12     protected $table = 'm_kategori';
13     2 references
14     protected $primaryKey = 'kategori_id';
15
16     4 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, 'kategori_id', 'kategori_id');
23     }
24 }
```

10. Ubah Kategori Controller sesuaikan dengan skrip berikut:

```
KategoriModel.php U KategoriController.php 1 U X
Minggu5 > Jobsheet5 > 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 8 references | 0 implementations
9 class KategoriController extends Controller
10 {
11     4 references | 0 overrides
12     public function index(KategoriDataTable $dataTable): mixed
13     {
14         return $dataTable->render('kategori.index');
15     }
16 }
```

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

```
KategoriModel.php U KategoriController.php 1 U index.blade.php U X
Minggu5 > Jobsheet5 > resources > views > kategori > index.blade.php
1 @extends('layouts.app')
2
3 {{-- Customize layout sections --}}
4
5 @section('subtitle', 'Kategori')
6 @section('content_header_title', 'Home')
7 @section('content_header_subtitle', 'Kategori')
8
9 @section('content')
10     <div class="container">
11         <div class="card">
12             <div class="card-header">Manage Kategori</div>
13             <div class="card-body">
14                 {{ $dataTable->table() }}
15             </div>
16         </div>
17     </div>
18 @endsection
19
20 @push('scripts')
21     {{ $dataTable->scripts() }}
22 @endpush
```



12. Pastikan route untuk kategori sudah tersedia

```
38 //JOBSHEET 5
39 //PRAKTIKUM 2 - NO 12
40 Route::get('/kategori', [HomeController::class, 'index']);
```

13. Sesuaikan app layout seperti kode berikut

```
app.blade.php U X
Minggu5 > Jobsheet5 > 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-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
28 @section('content')
29     @yield('content_body')
30 @stop
31
32 {{-- Create a common footer --}}
33
34 @section('footer')
35     <div class="float-right">
36         Version: {{ config(key: 'app.version', default: '1.0.0') }}
37     </div>
38
39     <strong>
40         <a href="{{ config(key: 'app.company_url', default: '#') }}">
41             {{ config(key: 'app.company_name', default: 'My company') }}
42         </a>
43     </strong>
44 @stop
45
46 {{-- Add common Javascript/Jquery code --}}
47
48 @push('js')
49     <script src="https://cdn.datatables.net/2.0.2/js/dataTables.js"></script>
50 @endpush
51
52 @stack('scripts')
53
54 {{-- Add common CSS customizations --}}
55 @push('css')
56     <link href="https://cdn.datatables.net/2.0.2/css/dataTables.dataTables.css" />
57
58     <style type="text/css">
59         {{-- You can add AdminLTE customizations here --}}
60         /*
61         .card-header {
62             border-bottom: none;
63         }
64         .card-title {
65             font-weight: 600;
66         }
67         */
68     </style>
69 @endpush
70
```



14. Menset ViteJs / script type defaults

```
AppServiceProvider.php U X
Minggus > Jobsheet5 > 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  7 references | 0 implementations
9  class AppServiceProvider extends ServiceProvider
10 {
11     /**
12      * Register any application services.
13      */
14     2 references | 0 overrides | prototype
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

		kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	KTG001	Elektronik	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	KTG002	Pakaian	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	KTG003	Makanan	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	4	KTG004	Minuman	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	KTG005	Obat-obatan	NULL	NULL

16. Datatables sudah dapat di load di /kategori

AdminLTE 3 | Kategori

127.0.0.1:8000/kategori

Home > Kategori

Manage Kategori

Search:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At
1	KTG001	Elektronik		
2	KTG002	Pakaian		
3	KTG003	Makanan		
4	KTG004	Minuman		
5	KTG005	Obat-obatan		

Showing 1 to 5 of 5 entries

1



Praktikum 3 – Membuat form kemudian menyimpan data dalam database

Langkah-langkahnya adalah sebagai berikut:

1. Menyesuaikan routing, tambahkan dua routing berikut

```
//PRAKTIKUM 3 - NO 1
Route::get(uri: '/kategori/create', action: [KategoriController::class, 'create']);
Route::post(uri: '/kategori', action: [KategoriController::class, 'store']);
```

2. Tambahkan dua function berikut dalam KategoriController

```
Minggu5 > Jobsheet5 > app > Http > Controllers > KategoriController.php > PHP > KategoriController

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



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

```
Minggu5 > Jobsheet5 > resources > views > kategori > create.blade.php
1  @extends('layouts.app')
2  {{-- Customize layout sections --}}
3  @section('subtitle', 'Kategori')
4  @section('content_header_title', 'Kategori')
5  @section('content_header_subtitle', 'Create')
6  {{-- Content body: main page content --}}
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="..../kategori">
15             <div class="card-body">
16                 <div class="form-group">
17                     <label for="kodeKategori">Kode Kategori</label>
18                     <input type="text" class="form-control" id="kodeKategori" name="kodeKategori" placeholder="Masukkan kode kategori">
19                 </div>
20                 <div class="form-group">
21                     <label for="namaKategori">Nama Kategori</label>
22                     <input type="text" class="form-control" id="namaKategori" name="namaKategori" placeholder="Masukkan nama kategori">
23                 </div>
24             </div>
25
26             <div class="card-footer">
27                 <button type="submit" class="btn btn-primary">Submit</button>
28             </div>
29         </form>
30     </div>
31 </div>
32 @endsection
```

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

```
Minggu5 > Jobsheet5 > app > Http > Middleware > VerifyCsrfToken.php > PHP Intelephense > VerifyCsrfToken
1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Illuminate\Foundation\Http\Middleware\VerifyCsrfToken as Middleware;
6
7  14 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

KTG006

Nama Kategori

Skincare

Submit

6. Halaman kategori

Home > Kategori

Manage Kategori

Search:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At
1	KTG001	Elektronik		
2	KTG002	Pakaian		
3	KTG003	Makanan		
4	KTG004	Minuman		
5	KTG005	Obat-obatan		
7	KTG006	Skincare	2025-03-15T14:59:45.000000Z	2025-03-15T14:59:45.000000Z

Showing 1 to 6 of 6 entries

<div><div><div>←</div><div>→</div></div></div>				kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<div><div><div></div></div></div>	<div><div><div></div></div></div> Edit	<div><div><div></div></div></div> Copy	<div><div><div></div></div></div> Delete	1	KTG001	Elektronik	NULL	NULL
<div><div><div></div></div></div>	<div><div><div></div></div></div> Edit	<div><div><div></div></div></div> Copy	<div><div><div></div></div></div> Delete	2	KTG002	Pakaian	NULL	NULL
<div><div><div></div></div></div>	<div><div><div></div></div></div> Edit	<div><div><div></div></div></div> Copy	<div><div><div></div></div></div> Delete	3	KTG003	Makanan	NULL	NULL
<div><div><div></div></div></div>	<div><div><div></div></div></div> Edit	<div><div><div></div></div></div> Copy	<div><div><div></div></div></div> Delete	4	KTG004	Minuman	NULL	NULL
<div><div><div></div></div></div>	<div><div><div></div></div></div> Edit	<div><div><div></div></div></div> Copy	<div><div><div></div></div></div> Delete	5	KTG005	Obat-obatan	NULL	NULL
<div><div><div></div></div></div>	<div><div><div></div></div></div> Edit	<div><div><div></div></div></div> Copy	<div><div><div></div></div></div> Delete	7	KTG006	Skincare	2025-03-15 14:59:45	2025-03-15 14:59:45



Tugas Praktikum

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

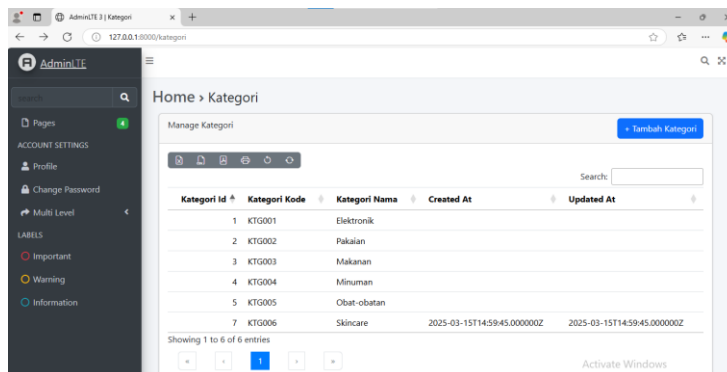
- Untuk menambahkan button “Add” di halaman kategori, edit file `/kategori/index.blade.php`

```
Minggu5 > Jobsheet5 > resources > views > kategori > index.blade.php
1  @extends('layouts.app')
2
3  {{-- Customize layout sections --}}
4
5  @section('subtitle', 'Kategori')
6  @section('content_header_title', 'Home')
7  @section('content_header_subtitle', 'Kategori')
8
9  @section('content')
10     <div class="container">
11         <div class="card-header">Manage Kategori
12         <div class="card-body">
13             <a href="{{ route('kategori.create') }}" class="btn btn-primary float-right">+ Tambah Kategori</a>
14         </div>
15         <div class="card-body">
16             {{ $dataTable->table() }}
17         </div>
18     </div>
19 @endsection
20
21 @push('scripts')
22     {{ $dataTable->scripts() }}
23 @endpush
```

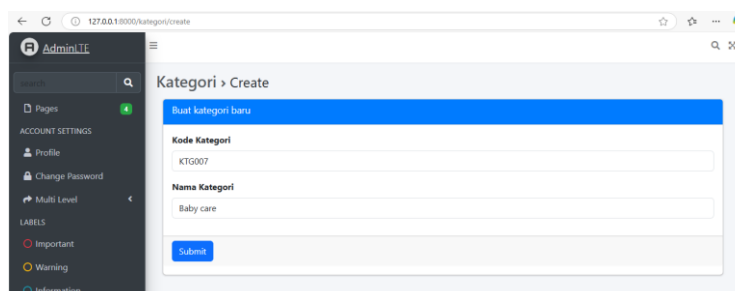
- Menambahkan route pada file route/web.php

```
//TUGAS - NO 1
Route::get('/kategori/create', [KategoriController::class, 'create'])->name('kategori.create');
```

- Tampilan pada web browser

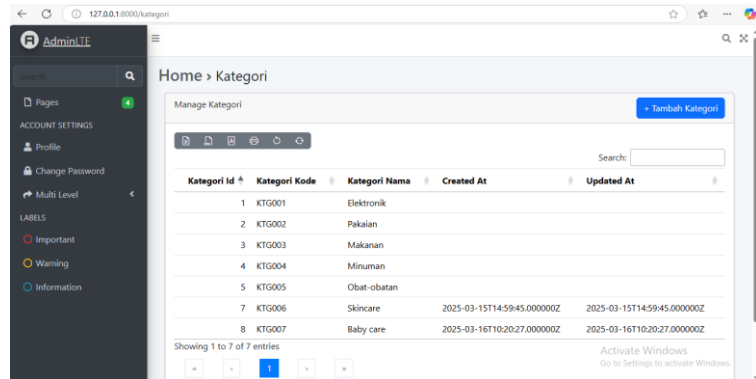


Untuk menambahkan kategori baru, user dapat mengklik button “Tambah Kategori” dan mengisi form yang tersedia.





Data berhasil di tambahkan ke database dan ditampilkan ke user



Home > Kategori

Manage Kategori

+ Tambah Kategori

Search:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At
1	KTG001	Elektronik		
2	KTG002	Pakaian		
3	KTG003	Makanan		
4	KTG004	Minuman		
5	KTG005	Obat-obatan		
7	KTG006	Skincare	2025-03-15T14:59:45.000000Z	2025-03-15T14:59:45.000000Z
8	KTG007	Baby care	2025-03-16T10:20:27.000000Z	2025-03-16T10:20:27.000000Z

Showing 1 to 7 of 7 entries

Activate Windows
Go to Settings to activate Windows.

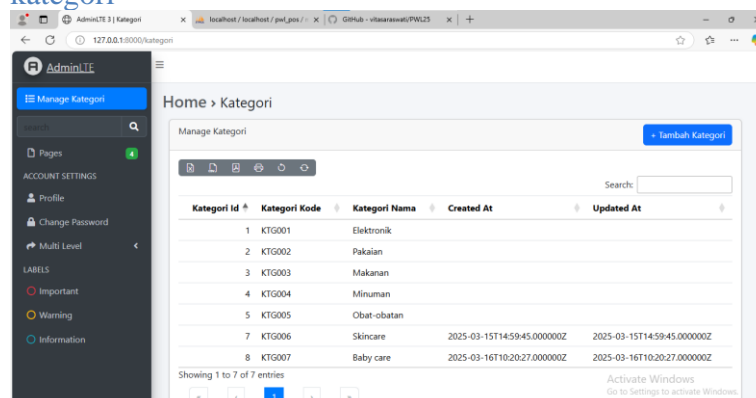
	kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/> Edit Copy Delete	1	KTG001	Elektronik	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	2	KTG002	Pakaian	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	3	KTG003	Makanan	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	4	KTG004	Minuman	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	5	KTG005	Obat-obatan	NULL	NULL
<input type="checkbox"/> Edit Copy Delete	7	KTG006	Skincare	2025-03-15 14:59:45	2025-03-15 14:59:45
<input type="checkbox"/> Edit Copy Delete	8	KTG007	Baby care	2025-03-16 10:20:27	2025-03-16 10:20:27

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

- Modifikasi file config/adminlte.php pada bagian navbar seperti di bawah ini

```
301     'menu' => [  
302         // Navbar items:  
303         [  
304             'text' => 'Manage Kategori',  
305             'url' => 'kategori',  
306             'icon' => 'fas fa-list',  
307         ],  
308         [  
309             'type' => 'navbar-search',  
310             'topnav_right' => true,  
311         ],  
312     ],
```

- Tampilan web pada web browser setelah menambahkan navigation bar manage kategori



Home > Kategori

Manage Kategori

+ Tambah Kategori

Search:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At
1	KTG001	Elektronik		
2	KTG002	Pakaian		
3	KTG003	Makanan		
4	KTG004	Minuman		
5	KTG005	Obat-obatan		
7	KTG006	Skincare	2025-03-15T14:59:45.000000Z	2025-03-15T14:59:45.000000Z
8	KTG007	Baby care	2025-03-16T10:20:27.000000Z	2025-03-16T10:20:27.000000Z

Showing 1 to 7 of 7 entries

Activate Windows
Go to Settings to activate Windows.



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

- Menambahkan function edit pada file **KategoriController.php**

```
31 public function edit($id): Factory\View {
32     $data = KategoriModel::findOrFail($id);
33     return view('kategori.edit', data: ['kategori' => $data]);
34 }
35
36
37 1 reference | 0 overrides
38 public function update(Request $request, $id): Redirector\RedirectResponse {
39     $request->validate([
40         'kodeKategori' => 'required',
41         'namaKategori' => 'required'
42     ]);
43
44     //cari data berdasarkan id
45     $kategori = KategoriModel::findOrFail($id);
46
47     //update data
48     $kategori->kodeKategori = $request->kodeKategori;
49     $kategori->namaKategori = $request->namaKategori;
50     $kategori->save();
51
52     return redirect(to: '/kategori');
53 }
54 }
```

- Modifikasi function **dataTable** dan function **getColumns** pada file **KategoriDataTable.php**

```
20 public function dataTable(QueryBuilder $query): EloquentDataTable {
21     return (new EloquentDataTable($query))
22         ->addColumn('action', 'kategori.action')
23         ->addColumn('action', function($id): string {
24             $edit = route(name: 'kategori.edit', parameters: $id);
25             return "<a href='{$edit}' class='btn btn-warning btn-sm'>Edit</a>";
26         })
27         ->setRowId('id');
28
29 }
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64 public function getColumns(): array {
65     return [
66         // Column::computed('action')
67         // ->exportable(false)
68         // ->printable(false)
69         // ->width(60)
70         // ->addClass('text-center'),
71         Column::make('kategori_id'),
72         Column::make('kategori_kode'),
73         Column::make('kategori_nama'),
74         Column::make('created_at'),
75         Column::make('updated_at'),
76
77         Column::computed('action')
78             ->exportable(false)
79             ->printable(false)
80             ->width(60)
81             ->addClass('text-center'),
82     ];
83 }
84 }
```

- Membuat file pada view/**kategori/edit.blade.php**

```
Minggu5 > Jobsheet5 > 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(name: '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
```



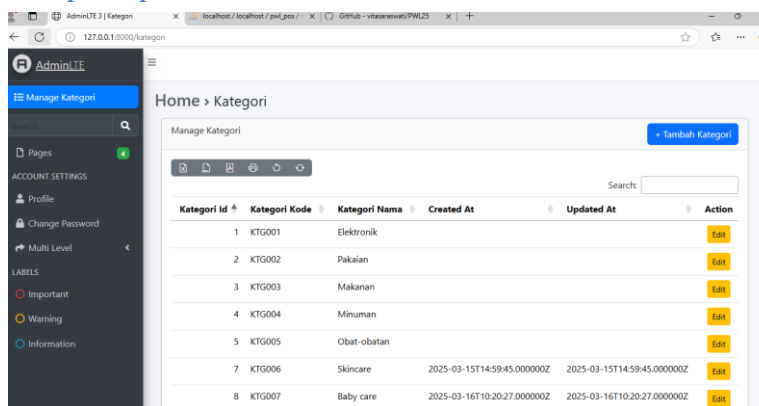
- Menambahkan script action pada file **kategori/index.blade.php**

```
22 @push('scripts')
23 {{ $dataTable->scripts() }}
24
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 </script>
34
35 @endpush
```

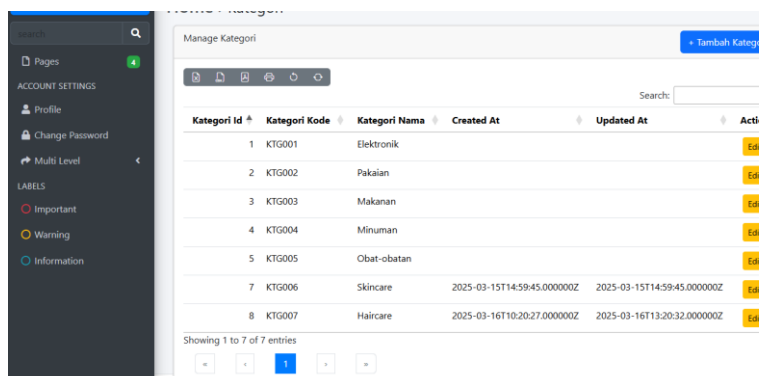
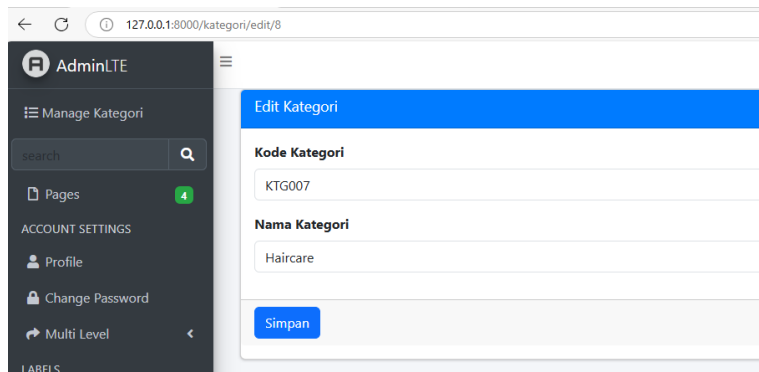
- Menambahkan route pada file web.php

```
47 //TUGAS - NO 3
48 Route::get('/kategori/edit/{id}', [KategoriController::class, 'edit'])->name('kategori.edit');
49 Route::put('/kategori/{id}', [KategoriController::class, 'update'])->name('kategori.update');
```

- Tampilan pada browser



User bisa melakukan edit data, tampilan nya seperti di bawah ini





4. Tambahkan action delete di datatables serta controllernya

- Menambah action delete dan merubah function `getColumns` pada file `KategoriDataTable.php`

```
0 references | 0 overrides
20 public function dataTable(QueryBuilder $query): EloquentDataTable
21 {
22     return (new EloquentDataTable($query))
23         // ->addColumn('action', 'kategori.action')
24         ->addColumn('action', function($id): string {
25             $edit = route(name: 'kategori.edit', parameters: $id);
26             $delete = route(name: 'kategori.delete', parameters: $id);
27             return '
28                 <a href="'.$edit.'" class="btn btn-warning btn-sm">Edit</a>
29                 <a href="'.$delete.'" class="btn btn-danger btn-sm" onclick="return confirm(\''Apakah Anda yakin ingin menghapus kategori ini? Tindakan ini tidak dapat dibatalkan.\');">Delete</a>
30             '
31         });
32     ->setRowId('id');
33 }
```

```
68 public function getColumns(): array
69 {
70     return [
71         // Column::computed('action')
72         // ->exportable(false)
73         // ->printable(false)
74         // ->width(60)
75         // ->addClass('text-center'),
76         Column::make('kategori_id'),
77         Column::make('kategori_kode'),
78         Column::make('kategori_nama'),
79         Column::make('created_at'),
80         Column::make('updated_at'),
81
82         Column::computed('action')
83         ->exportable(false)
84         ->printable(false)
85         ->width(150)
86         ->addClass('text-center'),
87     ];
88 }
```

- Menambahkan method delete pada `KategoriController.php`

```
1 reference | 0 overrides
55 public function delete($id): Redirector|RedirectResponse
56 {
57     KategoriModel::where('kategori_id', $id)->delete();
58     return redirect(to: '/kategori');
59 }
60 }
```

- Menambahkan route delete pada file `web.php`

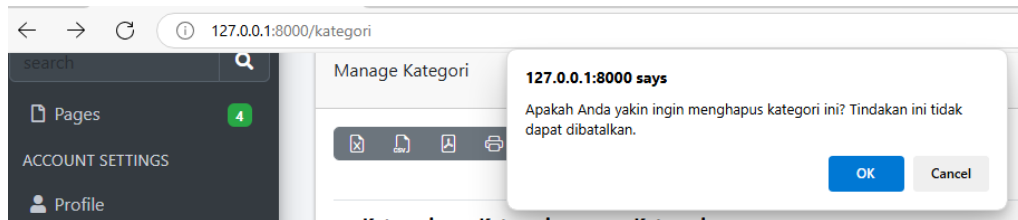
```
50 //TUGAS - NO 4
51 Route::get('/kategori/delete/{id}', [KategoriController::class, 'delete']->name('kategori.delete'));
```

- Tampilan pada web browser

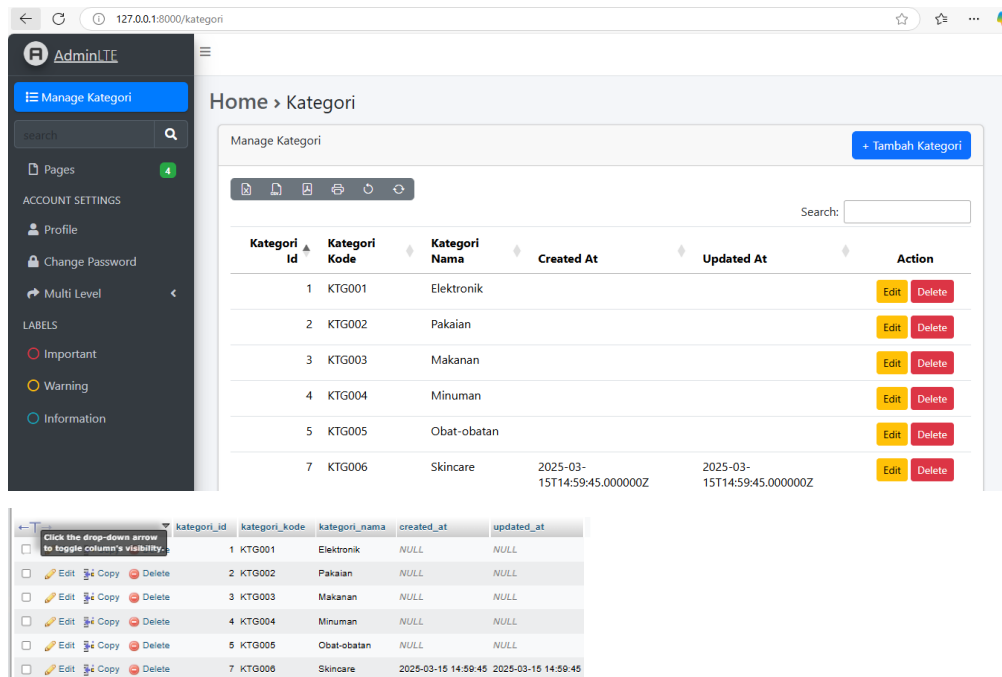
Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At	Action
1	KTG0001	Elektronik			TAMBAH HAPUS
2	KTG0002	Pakaian			TAMBAH HAPUS
3	KTG0003	Makanan			TAMBAH HAPUS
4	KTG0004	Minuman			TAMBAH HAPUS
5	KTG0005	Obat-obatan			TAMBAH HAPUS
7	KTG0006	Skinicare	2025-03-15T14:59:45.000000Z	2025-03-15T14:59:45.000000Z	TAMBAH HAPUS
8	KTG0007	Haircare	2025-03-16T11:20:27.000000Z	2025-03-16T11:20:27.000000Z	TAMBAH HAPUS



Saat user melakukan klik pada button Delete maka akan muncul alert seperti gambar di bawah ini



Jika user memilih OK maka data akan terhapus dari database



References:

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