

LAPORAN
PRAKTIKUM
WEBSITE LANJUT



RIZQI REZA DANUARTA
2241720057
D4-TEKNIK INFORMATIKA
KELAS 2C



Praktikum 1 – Integrasi Laravel dengan AdminLTE3

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

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

`composer require jeroennoten/laravel-adminlte`

```
C:\Windows\system32\cmd.exe x + v
Package operations: 2 installs, 0 updates, 0 removals
- Downloading almasaeed2010/adminlte (v3.2.0)
- Downloading jeroennoten/laravel-adminlte (v3.10.0)
- Installing almasaeed2010/adminlte (v3.2.0): Extracting archive
- Installing jeroennoten/laravel-adminlte (v3.10.0): Extracting archive
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

 INFO  Discovering packages.

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

83 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].

No security vulnerability advisories found.
Using version ^3.10 for jeroennoten/laravel-adminlte
c:\laragon\www\PWL_POS>
```

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

`php artisan adminlte:install`

```
c:\laragon\www\PWL_POS>php artisan adminlte:install
Basic assets installed successfully.
Configuration file installed successfully.
Translation files installed successfully.
The installation is complete.

c:\laragon\www\PWL_POS>
```

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 > ...
  Install | Update | Check
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" (7.8.1),
10         "jeroennoten/laravel-adminlte": "^3.10" (v3.10.0),
11         "laravel/framework": "^10.10" (v10.45.1),
12         "laravel/sanctum": "^3.3" (v3.3.3),
13         "laravel/tinker": "^2.8" (v2.9.0)
14     },
15     "require-dev": {
16         "fakerphp/faker": "^1.9.1" (v1.23.1),
17         "laravel/pint": "^1.0" (v1.14.0),
18         "laravel/sail": "^1.18" (v1.28.0),
19         "mockery/mockery": "^1.4.4" (1.6.7),
20         "nunomaduro/collision": "^7.0" (v7.10.0),
21         "phpunit/phpunit": "^10.1" (10.5.11),
22         "spatie/laravel-ignition": "^2.0" (2.4.2)
23     },
24     "autoload": {
25         "psr-4": {
26             "App\\": "app/",
27             "Database\\Factories\\": "database/factories/",
28             "Database\\Seeders\\": "database/seeders/"
29         }
30     },
31     "autoload-dev": {
```

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



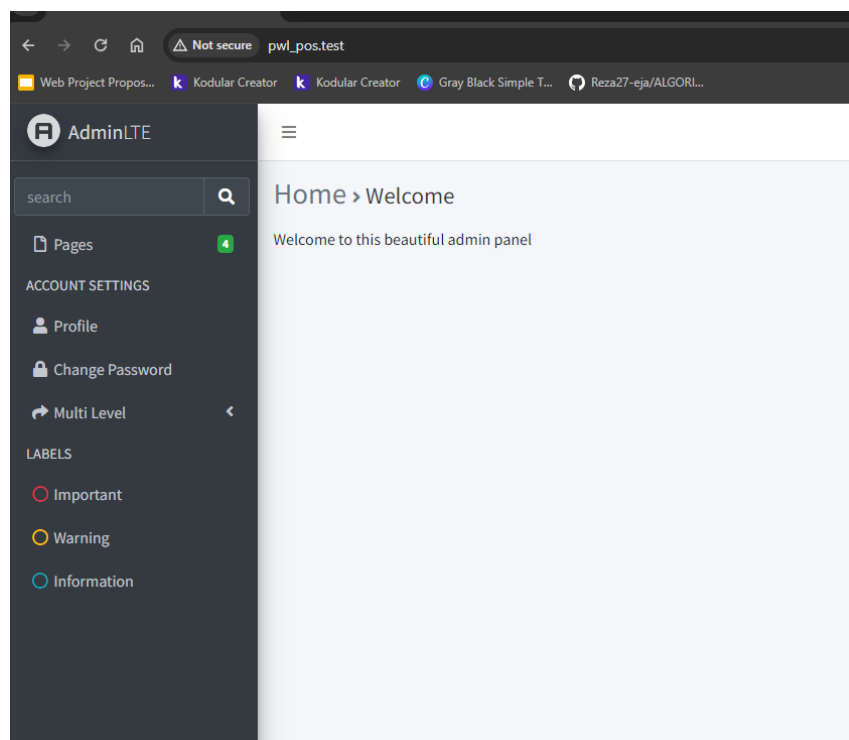
```
1  @extends('adminlte::page')
2
3  {{-- Extend and customize the browser title --}}
4
5  @section('title')
6      {{ config('adminlte.title') }}
7      @hasSection('subtitle') | @yield('subtitle') @endif
8  @stop
9
10 @vite('resources/js/app.js')
11
12 {{-- Extend and customize the page content header --}}
13
14 @section('content_header')
15     @hasSection('content_header_title')
16         <h1 class="text-muted">
17             @yield('content_header_title')
18
19             @hasSection('content_header_subtitle')
20
21                 <small class="text-dark">
22                     <i class="fas fa-angle-right text-muted"></i>
23                     @yield('content_header_subtitle')
24                 </small>
25             @endif
26         </h1>
27     @endif
28 @stop
29
30 {{-- Include Navbar --}}
31 @section('navbar')
32     <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
33         <a class="navbar-brand" href="{{ url('/') }}">Your App Name</a>
34         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
35             <span class="navbar-toggler-icon"></span>
36         </button>
37         <div class="collapse navbar-collapse" id="navbarNav">
38             <ul class="navbar-nav">
39                 <li class="nav-item">
40                     <a class="nav-link" href="{{ url('/kategori') }}">Manage Kategori</a>
41                 </li>
42                 <!-- Add other menu items here -->
43             </ul>
44         </div>
45     </nav>
46 @stop
47
48 {{-- Rename section content to content_body --}}
49
50 @section('content')
51     @yield('content_body')
52 @stop
53
54 {{-- Create a common footer --}}
55
56 @section('footer')
57     <div class="float-right">
58         Version: {{ config('app.version', '1.0.0') }}
59     </div>
60
61     <strong>
62         <a href="{{ config('app.company_url', '#') }}">
63             {{ config('app.company_name', 'My company') }}
64         </a>
65     </strong>
66 @stop
67
68
69 {{-- Add common Javascript/Jquery code --}}
70
71 @push('js')
72     <script src="https://cdn.datatables.net/2.0.2/js/dataTables.js"></script>
73 @endpush
74
75 @stack('scripts')
76
77
78 {{-- Add common CSS customizations --}}
79
80 @push('css')
81     <link rel="stylesheet" href="https://cdn.datatables.net/2.0.2/css/dataTables.dataTables.css" />
82     <style type="text/css">
83         {{-- You can add AdminLTE customizations here -->
84     </style>
85 @endpush
86
```

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



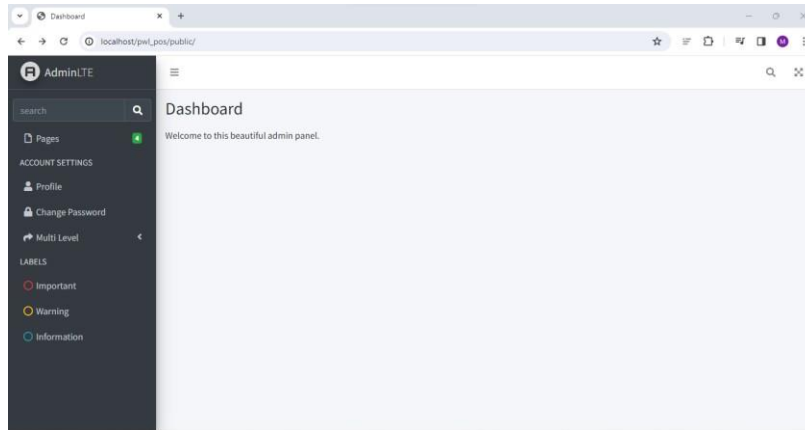
```
1 @extends( 'layout.app' )
2
3 @section('subtitle', 'Welcome')
4 @section('content_header_title', 'Home')
5 @section( 'content_header_subtitle', 'Welcome')
6
7 @section('content_body')
8     <p>Welcome to this beautiful admin panel</p>
9 @stop
10
11 @push('css')
12
13 @endpush
14
15 @push( 'js' )
16     <script> console.log("Hi, I'm using the Laravel-AdminLTE package");</script>
17 @endpush
```

Maka hasil output dari modifikasi view/layout/app dan view/welcome akan menjadi seperti tampilan berikut ini





5. Kembali ke browser, menuju ke halaman awal.



Praktikum 2 – Integrasi dengan DataTables

1. Install Laravel DataTables `composer require laravel/ui --dev composer require yajra/laravel-datatables:^10.0`
2. Pastikan nodejs sudah terinstall, dengan perintah `npm -v`.

```
PS C:\Users\ASUS> npm -v
10.2.4
PS C:\Users\ASUS> |
```

3. Install Laravel DataTables Vite dan sass `npm i laravel-datatables-vite --save-dev npm install -D sass`

```
c:\laragon\www\PWL_POS>npm i laravel-datatables-vite --save-dev npm install -D sass
added 50 packages, and audited 270 packages in 18s

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

found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 10.2.4 -> 10.5.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.5.0
npm notice Run `npm install -g npm@10.5.0` to update!
npm notice
c:\laragon\www\PWL_POS>|
```

4. Edit file `resources/js/app.js`



```
1 import './bootstrap';  
2 import "../sass/app.scss";  
3 import 'laravel-datatables-vite'
```

5. Buatlah file resources/sass/app.scss

```
1 // Import font Nunito  
2 @import url('https://fonts.googleapis.com/css2?family=Nunito:wght@400;600;700&display=swap');  
3  
4 // Import Bootstrap  
5 @import 'bootstrap/scss/bootstrap';  
6  
7 // Import Bootstrap Icons  
8 @import 'bootstrap-icons/font/bootstrap-icons.css';  
9  
10 // Import DataTables styles  
11 @import 'datatables.net-bs5/css/dataTables.bootstrap5.min.css';  
12 @import 'datatables.net-buttons-bs5/css/buttons.bootstrap5.min.css';  
13 @import 'datatables.net-select-bs5/css/select.bootstrap5.css';  
14
```

6. Jalankan dengan `npm run dev`

```
C:\Windows\system32\cmd.e: X + v  
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.5.0  
npm notice Run npm install -g npm@10.5.0 to update!  
npm notice  
c:\laragon\www\pwl_pos>npm run dev  
  
> dev  
> vite  
  
VITE v5.1.6 ready in 1175 ms  
→ Local: http://localhost:5173/  
→ Network: use --host to expose  
→ press h + enter to show help  
  
LARAVEL v10.45.1 plugin v1.0.2  
→ APP_URL: http://pwl\_pos.test
```



7. Selanjutnya kita akan buat datatables untuk kategori `php artisan datatables:make Kategori`

```
Using version ^10.0 for yajra/laravel-datatables-buttons
PS C:\laragon\www\PWL_POS> php artisan datatables:make Kategori

INFO DataTable [C:\laragon\www\PWL_POS\app\DataTables\KategoriDataTable.php] created successfully.
PS C:\laragon\www\PWL_POS> 
```

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

```
1 <?php
2
3 namespace App\DataTables;
4
5 use App\Models\KategoriModel;
6 use Illuminate\Database\Eloquent\Builder as QueryBuilder;
7 use Yajra\DataTables\EloquentDataTable;
8 use Yajra\DataTables\Html\Builder as HtmlBuilder;
9 use Yajra\DataTables\Html/Button;
10 use Yajra\DataTables\Html\Column;
11 use Yajra\DataTables\Services\DataTable;
12
13 class KategoriDataTable extends DataTable
14 {
15     /**
16      * Build the DataTable class.
17      *
18      * @param QueryBuilder $query Results from query() method.
19      */
20     public function dataTable(QueryBuilder $query): EloquentDataTable
21     {
22         return (new EloquentDataTable($query))
23             ->addColumn('action', function ($data) {
24                 return '<div class="d-flex gap-1">
25                     <a href="' . route('kategori.edit', $data->kategori_id) . '" class="btn btn-warning" style="width: 40px; height: 40px;">
26                         <i class="fas fa-edit"></i>
27                     </a>
28                     <form action="' . route('kategori.destroy', $data->kategori_id) . '" method="POST" style="display: inline;">
29                         <input type="hidden" name="_token" value="' . csrf_field() . '">
30                         <input type="hidden" name="_method" value="DELETE">
31                         <button type="submit" class="btn btn-danger" style="width: 40px; height: 40px;"><i class="fas fa-trash"></i></button>
32                     </form>
33                 </div>;
34             })
35             ->setRowId('id');
36     }
37
38     /**
39      * Get the query source of dataTable.
40      */
41     public function query(KategoriModel $model): QueryBuilder
42     {
43         return $model->newQuery();
44     }
45
46     /**
47      * Optional method if you want to use the html builder.
48      */
49     public function html(): HtmlBuilder
50     {
51         return $this->builder()
52             ->setTableId('kategori-table')
53             ->columns($this->getColumns())
54             ->minifiedAjax()
55             ->orderBy(1)
56             ->selectStyleSingle()
57             ->buttons([
58                 Button::make('excel'),
59                 Button::make('csv'),
60                 Button::make('pdf'),
61                 Button::make('print'),
62                 Button::make('reset'),
63                 Button::make('reload')
64             ]);
65     }
66
67     /**
68      * Get the dataTable columns definition.
69      */
70     public function getColumns(): array
71     {
72         return [
73             Column::make('kategori_id'),
74             Column::make('kategori_kode'),
75             Column::make('kategori_nama'),
76             Column::make('created_at'),
77             Column::make('updated_at'),
78             Column::computed('action')
79                 ->exportable(false)
80                 ->printable(false)
81                 ->width(60)
82                 ->addClass('text-center'),
83         ];
84     }
85
86     /**
87      * Get the filename for export.
88      */
89     protected function filename(): string
90     {
91         return 'Kategori_' . date('YmdHis');
92     }
93 }
```




9. Ubah kategori model, sesuaikan seperti berikut

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

10. Ubah Kategori Controller sesuaikan dengan skrip berikut:

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\DataTables\KategoriDataTable;
6  use App\Models\KategoriModel;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\DB;
9
10 class KategoriController extends Controller
11 {
12     public function index(KategoriDataTable $dataTable)
13     {
14         return $dataTable->render('kategori.index');
15         // $data = [
16         //     'kategori_kode' => 'SNK',
17         //     'kategori_nama' => 'Snack/Makanan Ringan',
18         //     'created_at' => now()
19         // ];
20         // DB::table('m_kategori')->insert($data);
21         // return 'Insert data baru berhasil';
22
23         // $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->update(['kategori_nama' => 'Cemilan']);
24         // return 'Update data berhasil. Jumlah data yang di update : ' . $row . ' baris';
25
26         // $row = DB::table('m_kategori')->where('kategori_kode', 'SNK')->delete();
27         // return 'Delete data berhasil. Jumlah data yang di hapus : ' . $row . ' baris';
28
29         // $data = DB::select('select * from m_kategori');
30         // return view('kategori', ['data' => $data]);
31     }
32 }
```



11. Buat folder kategori di dalam resources/view, kemudian buat view blade index untuk kategori [resources/views/kategori/index.blade.php](#)

```
1  @extends('layout.app')
2
3  @section('subtitle', 'Kategori')
4  @section('content_header_title', 'Home')
5  @section('content_header_subtitle', 'Kategori')
6
7  @section('content')
8      <div class="container">
9          <div class="card">
10             <div class="card-header">Manage Kategori</div>
11             <div class="d-flex justify-content-end">
12                 <a href="/kategori/create" class="btn btn-primary">
13                     <i class="bi bi-plus-circle-fill mr-1"></i>
14                     <span>Tambah Kategori</span>
15                 </a>
16             </div>
17             {!! $dataTable->table() !!}
18         </div>
19     </div>
20 @endsection
21
22 @push('scripts')
23     {!! $dataTable->scripts() !!}
24 @endpush
25
```

12. Pastikan route untuk kategori sudah tersedia

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

13. Sesuaikan app layout seperti kode berikut



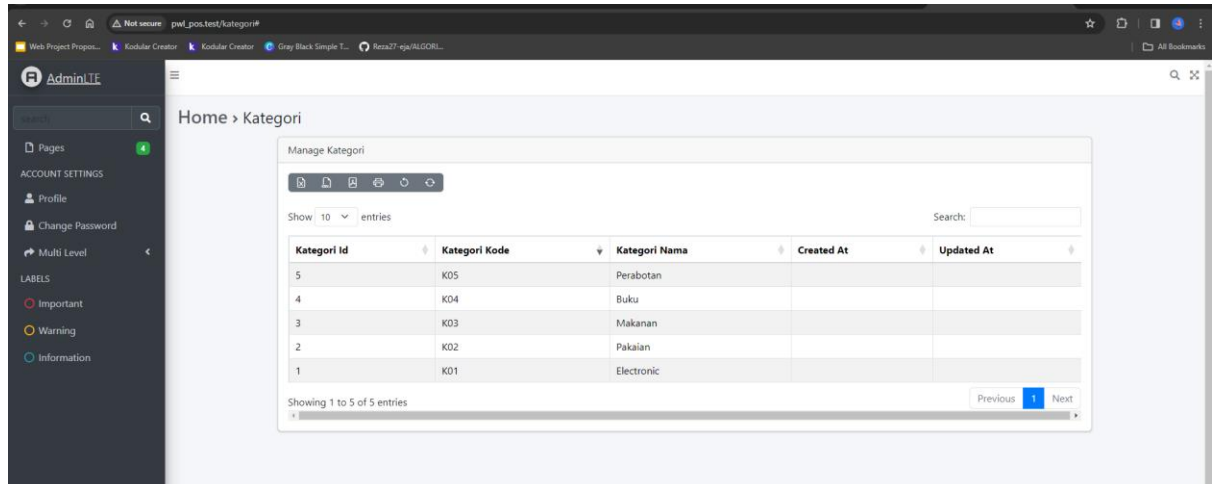
```
1  @extends('adminlte::page')
2
3  {{-- Extend and customize the browser title --}}
4
5  @section('title')
6      {{ config('adminlte.title') }}
7      @hasSection('subtitle') | @yield('subtitle') @endif
8  @stop
9
10 @vite('resources/js/app.js')
11
12 {{-- Extend and customize the page content header --}}
13
14 @section('content_header')
15     @hasSection('content_header_title')
16         <h1 class="text-muted">
17             @yield('content_header_title')
18
19             @hasSection('content_header_subtitle')
20
21                 <small class="text-dark">
22                     <i class="fas fa-xs fa-angle-right text-muted"></i>
23                     @yield('content_header_subtitle')
24                 </small>
25             @endif
26         </h1>
27     @endif
28 @stop
29
30 {{-- Include Navbar --}}
31 @section('navbar')
32     <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
33         <a class="navbar-brand" href="{{ url('/') }}">Your App Name</a>
34         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
35             <span class="navbar-toggler-icon"></span>
36         </button>
37         <div class="collapse navbar-collapse" id="navbarNav">
38             <ul class="navbar-nav">
39                 <li class="nav-item">
40                     <a class="nav-link" href="{{ url('/kategori') }}">Manage Kategori</a>
41                 </li>
42                 <!-- Add other menu items here -->
43             </ul>
44         </div>
45     </nav>
46 @stop
47
48 {{-- Rename section content to content_body --}}
49
50 @section('content')
51     @yield('content_body')
52 @stop
53
54 {{-- Create a common footer --}}
55
56 @section('footer')
57     <div class="float-right">
58         Version: {{ config('app.version', '1.0.0') }}
59     </div>
60
61     <strong>
62         <a href="{{ config('app.company_url', '#') }}">
63             {{ config('app.company_name', 'My company') }}
64         </a>
65     </strong>
66 @stop
67
68
69
70 {{-- Add common Javascript/Jquery code --}}
71
72 @push('js')
73     <script src="https://cdn.datatables.net/2.0.2/js/dataTables.js"></script>
74 @endpush
75
76 @stack('scripts')
77
78 {{-- Add common CSS customizations --}}
79
80 @push('css')
81     <link rel="stylesheet" href="https://cdn.datatables.net/2.0.2/css/dataTables.dataTables.css" />
82     <style type="text/css">
83         {{-- You can add AdminLTE customizations here --}}
84     </style>
85 @endpush
86
```

14. Menset ViteJs / script type defaults



```
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 }
26
```

15. Isikan beberapa data ke table kategori



Praktikum 3 – Membuat form kemudian menyimpan data dalam database

1. Menyesuaikan routing, tambahkan dua routing berikut

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

2. Tambahkan dua function berikut dalam KategoriController

```
1 public function create()
2 {
3     return view('kategori.create');
4 }
5
6
7 public function store(Request $request)
8 {
9     KategoriModel::create([
10         'kategori_kode' => $request->kodeKategori,
11         'kategori_nama' => $request->namaKategori,
12     ]);
13     return redirect('/kategori');
14 }
```



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

```
1  @extends('layout.app')
2
3  @section('subtitle', 'kategori')
4  @section('content_header_title', 'Kategori')
5  @section('content_header_subtitle', 'Create')
6
7  @section('content')
8      <div class="container">
9          <div class="card card-primary">
10             <div class="card-header">
11                 <h3 class="card-title">Buat Kategori Baru</h3>
12             </div>
13
14             <form method="POST" action="../../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="Kode Kategori">
19                     </div>
20                     <div class="form-group">
21                         <label for="kodeKategori">Nama Kategori</label>
22                         <input type="text" class="form-control" id="kodeKategori" name="namaKategori" placeholder="Nama Kategori">
23                     </div>
24                     <div class="card-footer">
25                         <button type="submit" class="btn btn-primary">Submit</button>
26                         <a href="{{ url('/kategori') }}" class="btn btn-secondary">Back</a>
27                     </div>
28                 </div>
29             </form>
30         </div>
31     </div>
32 @endsection
33
```

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

```
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   }
18
```

5. Akses kategori/create



Kategori > Create

Buat Kategori Baru

Kode Kategori
M_DGN

Nama Kategori
Minuman Dingin

Submit Back

6. Halaman kategori

Home > Kategori

Manage Kategori

Tambah Kategori

Show 10 entries

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At	Action
9	MKN	Makanan	2024-03-17T08:45:27.000000Z	2024-03-17T08:45:27.000000Z	Edit Delete
11	M_DGN	Minuman Dingin	2024-03-17T14:42:10.000000Z	2024-03-17T14:42:10.000000Z	Edit Delete
5	K05	Perabotan			Edit Delete
4	K04	Buku			Edit Delete
3	K03	Makanan			Edit Delete
2	K02	Pakaian			Edit Delete
1	K01	Electronic			Edit Delete

Showing 1 to 7 of 7 entries

Previous Next

Tugas Praktikum

1. Tambahkan button Add di dalam manage kategori, yang mengarah ke create kategori baru
 - Saya memodifikasi pada view/kategori/index.blade.php



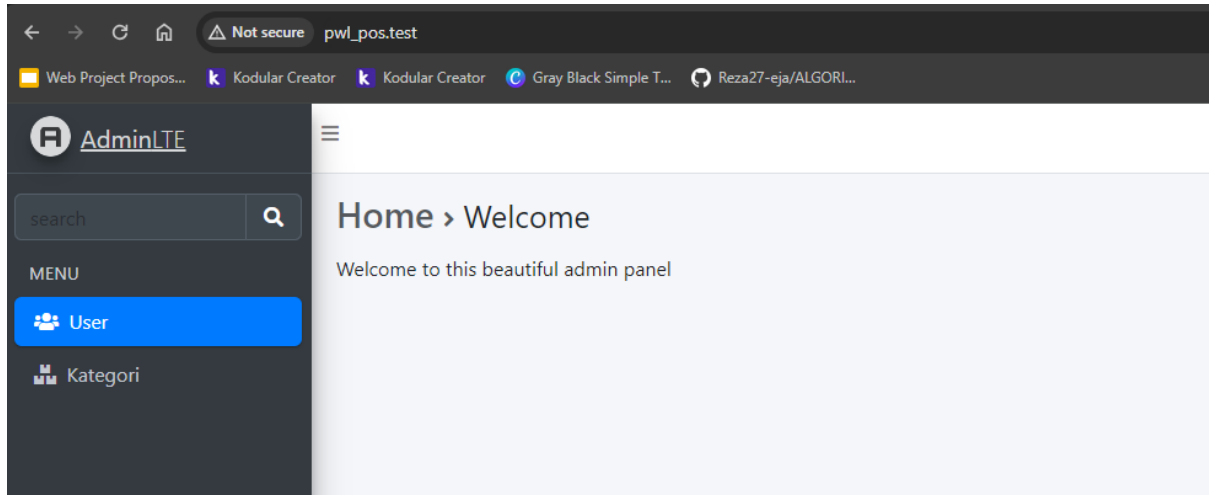
```
1 @extends('layout.app')
2
3 @section('subtitle', 'Kategori')
4 @section('content_header_title', 'Home')
5 @section('content_header_subtitle', 'Kategori')
6
7 @section('content')
8     <div class="container">
9         <div class="card">
10             <div class="card-header">Manage Kategori</div>
11             <div class="d-flex justify-content-end">
12                 <a href="/kategori/create" class="btn btn-primary">
13                     <i class="bi bi-plus-circle-fill mr-1"></i>
14                     <span>Tambah Kategori</span>
15                 </a>
16             </div>
17             {{ $dataTable->table() }}
18         </div>
19     </div>
20 @endsection
21
22 @push('scripts')
23     {{ $dataTable->scripts() }}
24 @endpush
25
```

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

- Saya memodifikasi pada folder config/adminlte.php

```
1 'menu' => [
2     // Navbar items:
3     [
4         'type' => 'navbar-search',
5         'text' => 'search',
6         'topnav_right' => true,
7     ],
8     [
9         'type' => 'fullscreen-widget',
10        'topnav_right' => true,
11    ],
12    // Sidebar items:
13    [
14        'type' => 'sidebar-menu-search',
15        'text' => 'search',
16    ],
17    [
18        'text' => 'blog',
19        'url' => 'admin/blog',
20        'can' => 'manage-blog',
21    ],
22    ['header' => 'MENU'],
23    [
24        'text' => 'User',
25        'url' => '/',
26        'icon' => 'fas fa-users mr-1',
27    ],
28    [
29        'text' => 'Kategori',
30        'url' => 'kategori',
31        'icon' => 'fa fa-boxes mr-1',
32    ],
33    ],
34 ]
```

Hasil outputnya akan menjadi seperti ini



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

- Pertama saya menambahkan routes seperti berikut ini

```
5 route::get('{id}', function() { return view('kategori.edit'); });
7 route::post('{id}', function() { return view('kategori.edit'); });
```

- Lalu saya menambahkan method pada KategoriController

```
1 public function edit($id)
2 {
3     $kategori = KategoriModel::find($id);
4     return view('kategori.edit', compact('kategori'));
5 }
6
7 public function update($id, Request $request)
8 {
9     $kategori = KategoriModel::find($id);
10    if (!$kategori) {
11        return redirect()->back()->with('error', 'Kategori tidak ditemukan.');
```

- Lalu saya memodifikasi pada KategoriDataTable menambahkan syntax perintah untuk button edit Dimana dalam syntax ini mengarahkan ke route dengan mengvalidasi id dari kategori_id yang digunakan untuk id mengedit data yang diinginkan



```
1 <?php
2
3 namespace App\DataTables;
4
5 use App\Models\KategoriModel;
6 use Illuminate\Database\Eloquent\Builder as QueryBuilder;
7 use Vajra\DataTables\EloquentDataTable;
8 use Vajra\DataTables\Html\Builder as HtmlBuilder;
9 use Vajra\DataTables\Html/Button;
10 use Vajra\DataTables\Html\Column;
11 use Vajra\DataTables\Services\DataTable;
12
13 class KategoriDataTable extends DataTable
14 {
15     /**
16      * Build the DataTable class.
17      *
18      * @param QueryBuilder $query Results from query() method.
19      */
20     public function dataTable(QueryBuilder $query): EloquentDataTable
21     {
22         return (new EloquentDataTable($query))
23             ->addColumn('action', function ($data) {
24                 return 'div class="d-flex gap-1">
25                     <a href="" . route('kategori.edit', $data->kategori_id) . "" class="btn btn-warning" style="width: 40px; height: 40px;">
26                         <i class="fas fa-edit"></i>
27                     </a>
28                     <form action="" . route('kategori.destroy', $data->kategori_id) . "" method="POST" style="display: inline;">
29                         . csrf_field() .
30                         . method_field('DELETE') .
31                         <button type="submit" class="btn btn-danger" style="width: 40px; height: 40px;"><i class="fas fa-trash"></i></button>
32                     </form>
33                 </div>;
34             })
35             ->setRowId('id');
36     }
37
38     /**
39      * Get the query source of dataTable.
40      */
41     public function query(KategoriModel $model): QueryBuilder
42     {
43         return $model->newQuery();
44     }
45
46     /**
47      * Optional method if you want to use the html builder.
48      */
49     public function html(): HtmlBuilder
50     {
51         return $this->builder()
52             ->setTableId('kategori-table')
53             ->columns($this->getColumns())
54             ->minifiedAjax()
55             ->orderBy(1)
56             ->selectStyleSingle()
57             ->buttons([
58                 Button::make('excel'),
59                 Button::make('csv'),
60                 Button::make('pdf'),
61                 Button::make('print'),
62                 Button::make('reset'),
63                 Button::make('reload')
64             ]);
65     }
66
67     /**
68      * Get the dataTable columns definition.
69      */
70     public function getColumns(): array
71     {
72         return [
73             Column::make('kategori_id'),
74             Column::make('kategori_kode'),
75             Column::make('kategori_nama'),
76             Column::make('created_at'),
77             Column::make('updated_at'),
78             Column::computed('action')
79                 ->exportable(false)
80                 ->printable(false)
81                 ->width(60)
82                 ->addClass('text-center'),
83         ];
84     }
85
86     /**
87      * Get the filename for export.
88      */
89     protected function filename(): string
90     {
91         return 'Kategori_' . date('YmdHis');
92     }
93 }
94
95
```



- Setelah itu saya membuat view yang digunakan untuk mengelola halaman edit

```
1 @extends('layout.app')
2
3 @section('subtitle', 'Kategori')
4 @section('content_header_title', 'Kategori')
5 @section('content_header_subtitle', 'Edit')
6
7 @section('content')
8 <div class="container">
9   <div class="card card-primary">
10     <div class="card-header">
11       <h3 class="card-title">Ubah Kategori</h3>
12     </div>
13     <form action="/kategori/{{ $kategori->kategori_id }}" method="POST">
14       {{ csrf_field() }}
15       {{ method_field('PUT') }}
16
17       <div class="card-body">
18         <div class="form-group">
19           <label for="kodekategori">Kode Kategori</label>
20           <input type="text" class="form-control" id="kodekategori" name="kodekategori" placeholder="Masukkan Kode Kategori" value="{{ $kategori->kategori_kode }}">
21         </div>
22         <div class="form-group">
23           <label for="namakategori">Nama Kategori</label>
24           <input type="text" class="form-control" id="namakategori" name="namakategori" placeholder="Masukkan Nama Kategori" value="{{ $kategori->kategori_nama }}">
25         </div>
26       </div>
27       <div class="card-footer">
28         <button type="submit" class="btn btn-primary">Ubah</button>
29         <a href="{{ url('/kategori') }}" class="btn btn-secondary">Back</a>
30       </div>
31     </form>
32   </div>
33 </div>
34 @endsection
35
```

- Lalu hasil outputnya seperti berikut ini
me > Kategori

Manage Kategori

Tambah Kategori

Show 10 entries

Search:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At	Action
9	MKN	Makanan	2024-03-17T08:45:27.000000Z	2024-03-17T08:45:27.000000Z	<div>✎ 🗑</div>
11	M_DGN	Minuman Dingin	2024-03-17T14:42:10.000000Z	2024-03-17T14:42:10.000000Z	<div>✎ 🗑</div>
5	K05	Perabotan			<div>✎ 🗑</div>

- Klik button edit dan akan diarahkan ke halaman edit

Kategori > Edit

Ubah Kategori

Kode Kategori

MKN

Nama Kategori

Makanan

Ubah

Back

- Selanjutnya kita bisa mengedit data yang kita ingin disini saya ingin mengubah data kategori diatas menjadi M_PNS seperti berikut ini



Kategori > Edit

Ubah Kategori

Kode Kategori
M_PNS

Nama Kategori
Minuman Panas

Ubah Back

- Setelah sudah maka klik ubah untuk memproses hasilnya

Home > Kategori

Manage Kategori

Tambah Kategori

Show 10 entries

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At	Action
9	M_PNS	Minuman Panas	2024-03-17T08:45:27.000000Z	2024-03-17T15:05:59.000000Z	Edit Delete
11	M_DGN	Minuman Dingin	2024-03-17T14:42:10.000000Z	2024-03-17T14:42:10.000000Z	Edit Delete

- Maka data berhasil di edit atau di update
4. Tambahkan action delete di datatables serta controllernya

- Saya menambahkan routes yang digunakan untuk mengarahkan ke rute delete

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

- Lalu saya menambahkan method delete yang digunakan untuk delete data yang diinginkan

```
1 public function destroy($id)
2 {
3     KategoriModel::destroy($id);
4     return redirect('/kategori');
5 }
```

- Lalu saya menambahkan syntax pada KategoriDataTable



```
1 public function dataTable(QueryBuilder $query): EloquentDataTable
2 {
3     return (new EloquentDataTable($query))
4         ->addColumn('action', function ($data) {
5             return '<div class="d-flex gap-1">
6                 <a href="" . route('kategori.edit', $data->kategori_id) . "" class="btn btn-warning" style="width: 40px; height: 40px;">
7                     <i class="fas fa-edit"></i>
8                 </a>
9                 <form action="" . route('kategori.destroy', $data->kategori_id) . "" method="POST" style="display: inline;">
10                     ' . csrf_field() . '
11                     ' . method_field('DELETE') . '
12                     <button type="submit" class="btn btn-danger" style="width: 40px; height: 40px;"><i class="fas fa-trash"></i></button>
13                 </form>
14             </div>;
15         })
16         ->setRowId('id');
17 }
```

- Selanjutnya saya membuat view yang digunakan untuk membuat halaman delete

```
1 @extends('layout.app')
2
3 @section('subtitle', 'kategori')
4 @section('content_header_title', 'Kategori')
5 @section('content_header_subtitle', 'Delete')
6
7 @section('content')
8     <div class="container">
9         <div class="card card-danger">
10             <div class="card-header">
11                 <h3 class="card-title">Delete Kategori</h3>
12             </div>
13
14             <div class="card-body">
15                 <p>Apakah Anda yakin ingin menghapus kategori ini?</p>
16             </div>
17
18             <div class="card-footer">
19                 <form method="POST" action="{{ route('kategori.destroy', $kategori->id) }}">
20                     @csrf
21                     @method('DELETE')
22                     <button type="submit" class="btn btn-danger">Delete</button>
23                     <a href="{{ url('/kategori') }}" class="btn btn-secondary">Cancel</a>
24                 </form>
25             </div>
26         </div>
27     </div>
28 @endsection
29
```

- Maka saya sudah bisa menghapus data yang saya inginkan seperti berikut ini adalah data sebelum saya hapus



Home > Kategori

Manage Kategori

Tambah Kategori

Show 10 entries

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At	Action
9	M_PNS	Minuman Panas	2024-03-17T08:45:27.000000Z	2024-03-17T15:05:59.000000Z	
11	M_DGN	Minuman Dingin	2024-03-17T14:42:10.000000Z	2024-03-17T14:42:10.000000Z	
5	K05	Perabotan			
4	K04	Buku			
3	K03	Makanan			
2	K02	Pakaian			
1	K01	Electronic			

Showing 1 to 7 of 7 entries

Previous 1 Next

- Setelah di hapus maka akan seperti berikut ini

Home > Kategori

Manage Kategori

Tambah Kategori

Show 10 entries

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At	Action
11	M_DGN	Minuman Dingin	2024-03-17T14:42:10.000000Z	2024-03-17T14:42:10.000000Z	
5	K05	Perabotan			
4	K04	Buku			

Link github pengerjaan jobsheet 5 adalah berikut ini :

https://github.com/reziag/pwl_pos