Laporan Praktikum Pemrograman Web Lanjut

Jobsheet 5: Blade View, Web Templating(AdminLTE), Datatables.

Oleh :

Danica Nasywa Putriniar (2341760122)

Kelas SIB 2B / 05

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS JURUSAN

TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

Jl. Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur

65141

Praktikum 1 – Integrasi Laravel dengan AdminLte3

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


```
● PS E:\Aplik\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 5> composer require jeroennoten/laravel-adminlte
 ./composer.json has been updated
 Running composer update jeroennoten/laravel-adminlte
 Loading composer repositories with package information
 Updating dependencies
 Lock file operations: 2 installs, 0 updates, 0 removals
   - Locking almasaeed2010/adminlte (v3.2.0)
   - Locking jeroennoten/laravel-adminlte (v3.15.0)
 Writing lock file
 Installing dependencies from lock file (including require-dev)
 Package operations: 2 installs, 0 updates, 0 removals
   - Installing almasaeed2010/adminlte (v3.2.0): Extracting archive
   - Installing jeroennoten/laravel-adminlte (v3.15.0): Extracting archive
 Generating optimized autoload files
 > Illuminate\Foundation\ComposerScripts::postAutoloadDump
 > @php artisan package:discover --ansi

   INFO  Discovering packages.

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

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

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

composer require jeroennoten/laravel-adminlte

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

```
 PS E:\Aplik\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 5> php artisan adminlte:install
● AdminLTE assets files published successfully
 Configuration file published successfully
 Translation files published successfully
 The installation is complete.
 ● DS E:\Aplik\laragon\...\Pemrograman Web Lanjut 2025\Minggu 5>
```

php artisan adminlte:install

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

```json
{
    "name": "laravel/laravel",
    "type": "project",
    "description": "The skeleton application for the Laravel framework.",
    "keywords": ["laravel", "framework"],
    "license": "MIT",
    "require": {
        "php": "^8.1",
        "guzzlehttp/guzzle": "^7.2",
        "jeroennoten/laravel-adminlte": "^3.15",
        "laravel/framework": "^10.10",
        "laravel/sanctum": "^3.3",
        "laravel/tinker": "^2.8"
    },
    "require-dev": {
        "fakerphp/faker": "^1.9.1",
        "laravel/pint": "^1.0",
        "laravel/sail": "^1.18",
        "mockery/mockery": "^1.4.4",
        "nunomaduro/collision": "^7.0",
        "phpunit/phpunit": "^10.1",
        "spatie/laravel-ignition": "^2.0"
    },
    "autoload": {
        "psr-4": {
            "App\\": "app/",
            "Database\\Factories\\": "database/factories/",
            "Database\\Seeders\\": "database/seeders/"
        }
    },
    "autoload-dev": {
        "psr-4": {
            "Tests\\": "tests/"
        }
    },
    "scripts": {
        "post-autoload-dump": [
```

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

```php
@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') }}
```

```blade
{{-- 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

@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 resource/views/welcome.blade.php

```
resources > views > ⚛ welcome.blade.php > ...
  1    @extends('adminlte::page')
  2
  3    @section('title', 'Home')
  4
  5    @section('subtitle', 'Welcome')
  6    @section('content_header')
  7        <h1>Home</h1>
  8        <small>Welcome</small>
  9    @endsection
 10
 11    @section('content')
 12        <p>Welcome to this beautiful admin panel.</p>
 13    @endsection
 14
 15    @push('css')
 16        {{-- Add here extra stylesheets --}}
 17        {{-- <link rel="stylesheet" href="/css/admin_custom.css"> --}}
 18    @endpush
 19
 20    @push('js')
 21        <script> console.log("Hi, I'm using the Laravel-AdminLTE package!"); </script>
 22    @endpush
 23
```

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

Template yields di beberapa bagian terklasifikasi dalam 2 yield:

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

Dokumentasi lebih detail terdapat di link berikut:

https://github.com/jeroennoten/Laravel- AdminLTE/wiki/Usage

5. Kembali ke browser, menuju ke halaman awal
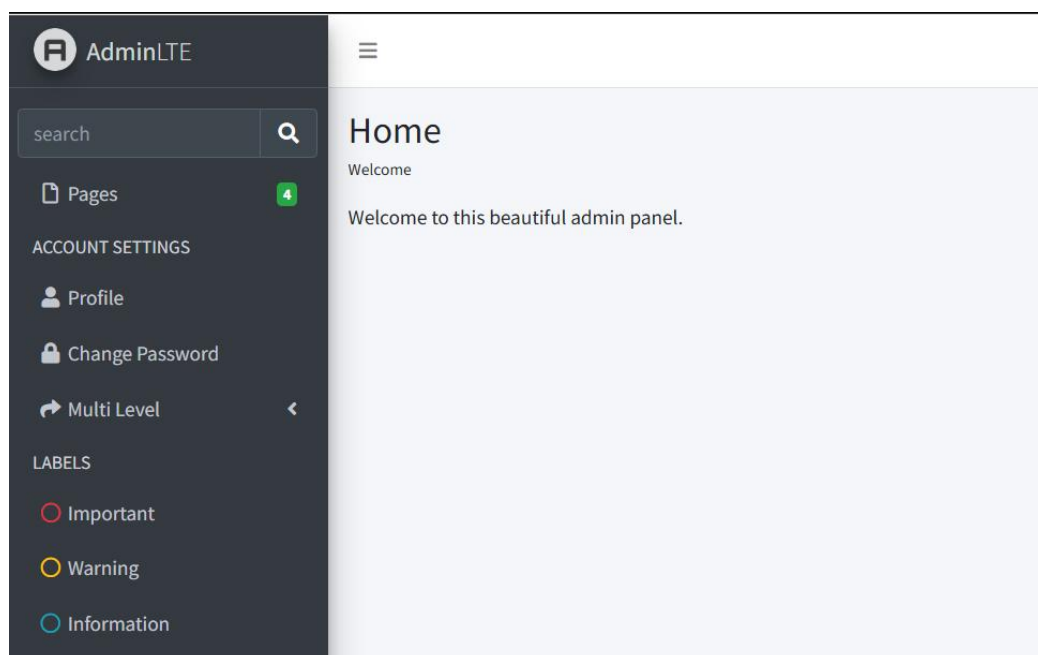
Praktikum 2 – Integrasi dengan DataTables

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

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.

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

npm install -D sass

4. Edit file resources/js/app.js

5. Buatlah file resources/saas/app.scss

6. Jalankan dengan npm run dev

7. Selanjutnya kita akan buat datatables untuk kategori

php artisan datatables:make Kategori

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\Button;
use Yajra\DataTables\Services\DataTable;

class KategoriDataTable extends DataTable
{
    /**
     * Build the DataTable class.
     *
     * @param QueryBuilder $query Results from query() method.
     * @return \Yajra\DataTables\EloquentDataTable
     */
    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(): QueryBuilder
    {
        return KategoriModel::query();
    }

    /**
     * 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'),
            ]);
```

```php
    /**
     * Get the dataTable columns definition.
     */
    public function getColumns(): array
    {
        return [
            Column::computed('action')
                ->exportable(false)
                ->printable(false)
                ->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

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\HasMany;

class KategoriModel extends Model
{
    use HasFactory;

    protected $table = 'm_kategori';
    protected $primaryKey = 'kategori_id';

    protected $fillable = ['kategori_kode', 'kategori_nama'];

    public function barang(): HasMany
    {
        return $this->hasMany(BarangModel::class, 'kategori_id', 'kategori_id');
    }
}
```

Karena masih error maka saya menambahkan BarangModel.php

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;

class BarangModel extends Model
{
    protected $table = 'm_barang'; // Nama tabel di database
    protected $primaryKey = 'barang_id'; // Primary key tabel

    protected $fillable = ['kategori_id', 'barang_kode', 'barang_nama', 'harga'];

    public function kategori(): BelongsTo
    {
        return $this->belongsTo(KategoriModel::class, 'kategori_id', 'kategori_id');
    }
}
```

10. Ubah Kategori Controller sesuaikan dengan skrip berikut:

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\DataTables\KategoriDataTable;

class KategoriController extends Controller
{
    public function index(KategoriDataTable $dataTable)
    {
        return $dataTable->render('kategori.index');
    }
}
```

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

```php
@extends('layouts.app')

{{-- Customize layout sections --}}

@section('subtitle', 'Kategori')
@section('content_header_title', 'Home')
@section('content_header_subtitle', 'Kagegori')

@section('content')
    <div class="container">
        <div class="card">
            <div class="card-header">Manage Kategori</div>
            <div class="card-body">
                {{ $dataTable->table() }}
            </div>
        </div>
    </div>
@endsection

@push('scripts')
    {{ $dataTable->scripts() }}
@endpush
```

12. Pastikan route untung kategori sudah tersedia

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

13. Sesuaikan app layout seperti kode berikut

```php
@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')
        </h1>
    @endif

    @hasSection('content_header_subtitle')
        <small class="text-dark">
            <i class="fa fa-xs fa-angle-right text-muted"></i>
            @yield('content_header_subtitle')
        </small>
    @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('scripts')
    <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/1.10.2/css/dataTables.dataTables.css" />
    <style type="text/css">
        .card-header {
            border-bottom: none;
        }

        .card-title {
            font-weight: 600;
        }
    </style>
@endpush
```

14. Menset ViteJs / script type defaults

```php
<?php

namespace App\Providers;

use Illuminate\Support\ServiceProvider;
use Yajra\DataTables\Html\Builder;

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

    /**
     * Bootstrap any application services.
     */
    public function boot(): void
    {
        Builder::useVite();
    }
}
```

15. Isikan beberapa data ke table kategori

☐	🖉 Edit	🗐 Copy	⊖ Delete	10	CML	Camilan	2025-03-15 23:46:31	*NULL*
☐	🖉 Edit	🗐 Copy	⊖ Delete	11	MNR	Minuman Ringan	2025-03-15 23:46:31	*NULL*

16. Datatables sudah dapat di load di /kategori

Praktikum 3 - Membuat form kemudian menyiman data dalam database

1. Menyesuaikan routing, tambahkan dua routing berikut:

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

2. Tambahkan dua function berikut dalam KategoriController

```php
<?php

namespace App\Http\Controllers;

use App\Models\KategoriModel;
use Illuminate\Http\Request;
use App\DataTables\KategoriDataTable;

class KategoriController extends Controller
{
    public function index(KategoriDataTable $dataTable)
    {
        return $dataTable->render('kategori.index');
    }

    public function create()
    {
        return view('kategori.create');
    }

    public function store(Request $request)
    {
        KategoriModel::create([
            'kategori_kode' => $request->kodeKategori,
            'kategori_nama' => $request->namaKategori,
        ]);
        return redirect('/kategori');
    }
}
```

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

```php
<?php

namespace App\Http\Controllers;

use App\Models\KategoriModel;
use Illuminate\Http\Request;
use App\DataTables\KategoriDataTable;

class KategoriController extends Controller
{
    public function index(KategoriDataTable $dataTable)
    {
        return $dataTable->render('kategori.index');
    }

    public function create()
    {
        return view('kategori.create');
    }

    public function store(Request $request)
    {
        KategoriModel::create([
            'kategori_kode' => $request->kodeKategori,
            'kategori_nama' => $request->namaKategori,
        ]);
        return redirect('/kategori');
    }
}
```

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

```php
<?php

namespace App\Http\Middleware;

use Illuminate\Foundation\Http\Middleware\VerifyCsrfToken as Middleware;

class VerifyCsrfToken extends Middleware
{
    /**
     * The URIs that should be excluded from CSRF verification.
     *
     * @var array<int, string>
     */
    protected $except = [
        '/kategori'
    ];
}
```

5. Akses kategori/create
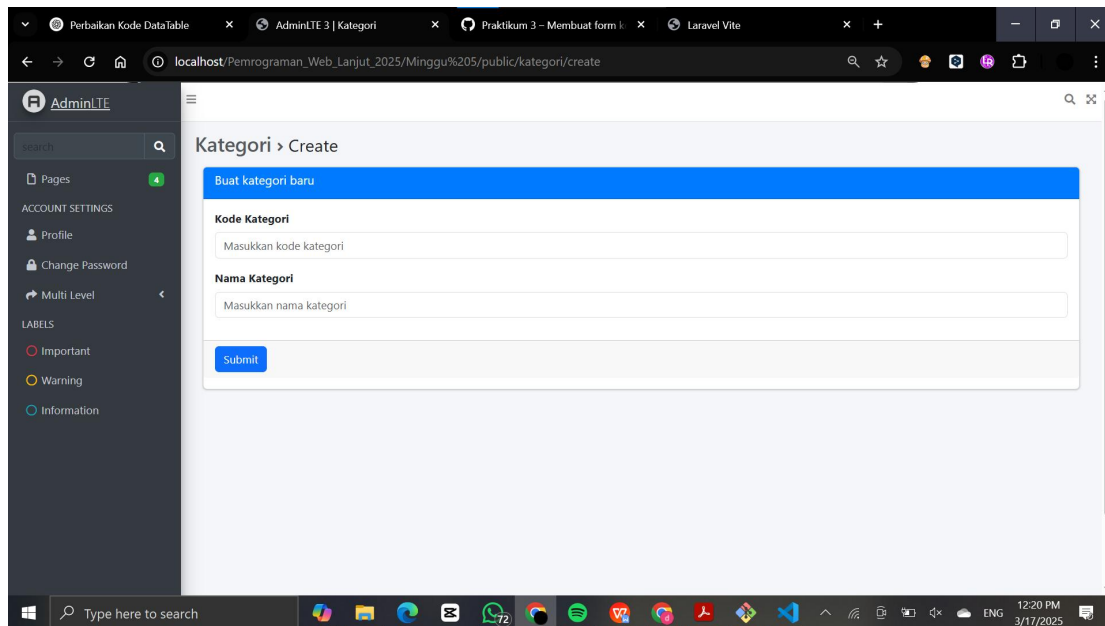
6. Halaman kategori
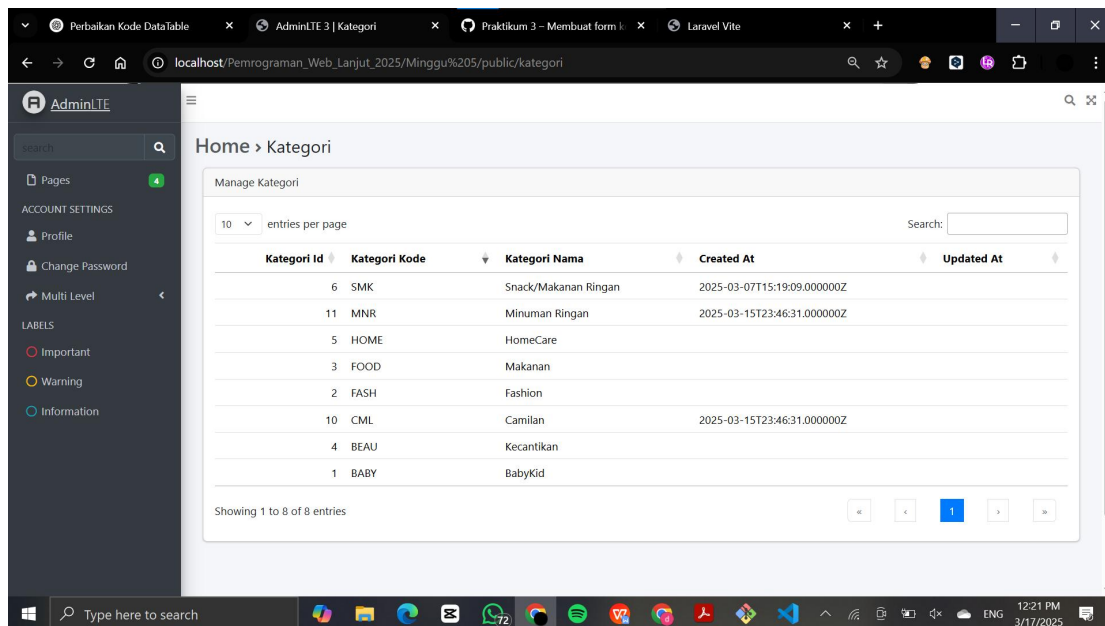
Tugas Praktikum

1. Tambahkan button Add di halaman manage kategori, yang mengarah ke create kategori.
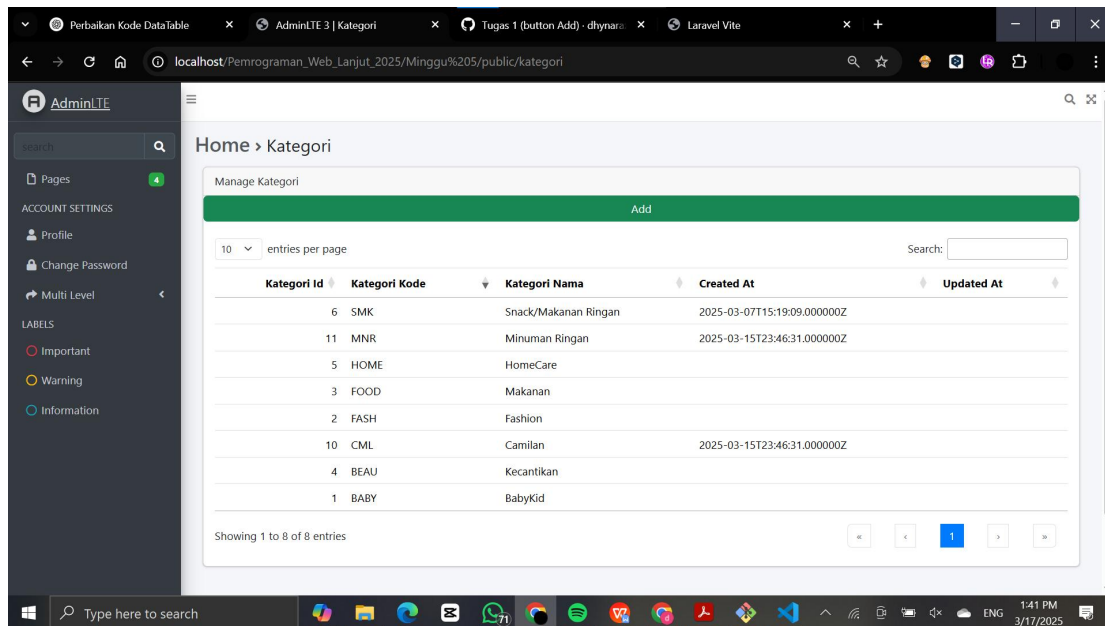
2. Tambahkan menu untuk halaman manage kategori, di daftar menu navbar

Home › Kategori

Manage Kategori

Add

10 ∨ entries per page			Search:	
Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At
6	SMK	Snack/Makanan Ringan	2025-03-07T15:19:09.000000Z	
11	MNR	Minuman Ringan	2025-03-15T23:46:31.000000Z	
5	HOME	HomeCare		
3	FOOD	Makanan		
2	FASH	Fashion		
10	CML	Camilan	2025-03-15T23:46:31.000000Z	
4	BEAU	Kecantikan		
1	BABY	BabyKid		

Showing 1 to 8 of 8 entries

« ‹ 1 › »

```
    ],
    [
        'type' => 'fullscreen-widget',
        'topnav_right' => true,
    ],
    [
        'text' => 'Manage Kategori',
        'url'  => 'kategori',
        'icon' => 'fas fa-list', // Ikon FontAwesome
        'topnav' => true, // Menampilkan di navbar
    ],
```

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

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At	Action
6	SMK	Snack/Makanan Ringan	2025-03-07T15:19:09.000000Z		Edit
11	MNR	Minuman Ringan	2025-03-15T23:46:31.000000Z		Edit
5	HOME	HomeCare			Edit
3	FOOD	Makanan			Edit
2	FASH	Fashion			Edit
10	CML	Camilan	2025-03-15T23:46:31.000000Z		Edit
4	BEAU	Kecantikan			Edit
1	BABY	BabyKid			Edit

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

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

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

resources > views > kategori > edit.blade.php > ...

```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                       <form action="{{ route('kategori.update', $kategori->kategori_id) }}" method="POST">
12                           @csrf
13                           @method('PUT')
14                           <div class="card-body">
15                               <div class="form-group">
16                                   <label for="kodeKategori">Kode Kategori</label>
17                                   <input type="text" class="form-control" name="kodeKategori" id="kodeKategori"
18                                       value="{{ $kategori->kategori_kode }}">
19                               </div>
20                               <div class="form-group">
21                                   <label for="namaKategori">Nama Kategori</label>
22                                   <input type="text" class="form-control" name="namaKategori" id="namaKategori"
23                                       value="{{ $kategori->kategori_nama }}">
24                               </div>
25                           </div>
26                           <div class="card-footer">
27                               <button type="submit" class="btn btn-primary">Simpan</button>
28                           </div>
29                       </form>
30                   </div>
31               </div>
32           </div>
33       </div>
34   @endsection
```

```
public function edit($id)
{
$data = KategoriModel::find($id);
return view('kategori.edit', ['kategori' => $data]);
}

public function update(Request $request, $id)
{
KategoriModel::where('kategori_id', $id)->update([
    'kategori_kode' => $request->kodeKategori,
    'kategori_nama' => $request->namaKategori,
]);

return redirect()->route('kategori.index')->with('success', 'Kategori berhasil diperbarui');
}
}
```

4. Tambahkan action delete di datatables

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At	Action
6	SMK	Snack/Makanan Ringan	2025-03-07T15:19:09.000000Z		Edit Delete
11	MNR	Minuman Ringan	2025-03-15T23:46:31.000000Z		Edit Delete
5	HOME	HomeCare			Edit Delete
3	FOOD	Makanan			Edit Delete
2	FASH	Fashion			Edit Delete
10	CML	Camilan	2025-03-15T23:46:31.000000Z		Edit Delete
4	BEAU	Kecantikan			Edit Delete
1	BABY	BabyKid			Edit Delete

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

```
}

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

```php
$edit = route('kategori.edit', $id);
$delete = route('kategori.delete', $id);

return '<div style="display: flex; gap: 5px;">
    <a href="' . $edit . '" class="btn btn-warning btn-sm">Edit</a>
    <a href="' . $delete . '" class="btn btn-danger btn-sm btn-delete">Delete</a>
</div>';
```