Jobsheet 05: Blade View, Web Templating(AdminLTE), Datatables

Dosen Pengampu Mata Kuliah :

M. Hasyim Ratsanjani, S.Kom., M.Kom.

Dibuat Oleh :

Daffa Yudisa Akbar : TI-2H / 10 : 2241720008

Politeknik Negeri Malang

Jurusan Teknologi Informasi

D-IV Teknik Informatika

2024

A. Praktikum 1 – Integrasi Laravel dengan AdminLte3

No.	Langkah Praktikum
1.	Dalam root folder project lakukan command berikut, untuk mendefinisikan requirement project. composer require jeroennoten/laravel-adminlte

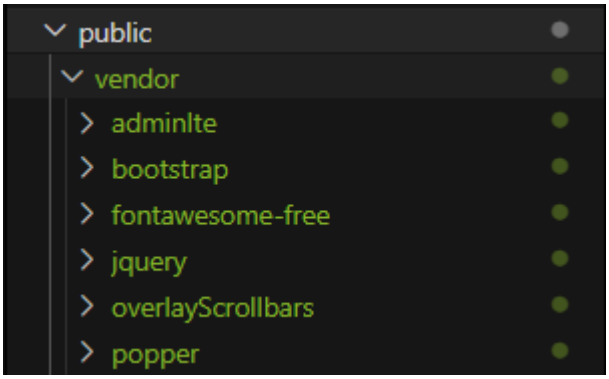
```
PS C:\laragon\www\PWL_POS> composer require jeroennoten/laravel-adminlte
./composer.json has been updated
Running composer update jeroennoten/laravel-adminlte
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Installing dependencies from lock file (including require-dev)
Package operations: 2 installs, 0 updates, 0 removals
  - Downloading almasaeed2010/adminlte (v3.2.0)
  - Installing almasaeed2010/adminlte (v3.2.0): Extracting archive
  - Installing jeroennoten/laravel-adminlte (v3.11.0): 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
  pestphp/pest-plugin-laravel ..... DONE
  spatie/laravel-ignition ..... DONE

90 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 ^3.11 for jeroennoten/laravel-adminlte
PS C:\laragon\www\PWL_POS>
```

2.	Melakukan instalasi requirement project di atas dengan command berikut: php artisan adminlte:install

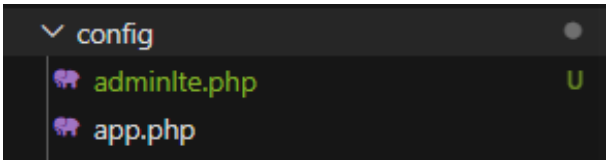
```
PS C:\laragon\www\PWL_POS> php artisan adminlte:install
Basic assets installed successfully.
Configuration file installed successfully.
Translation files installed successfully.
The installation is complete.
PS C:\laragon\www\PWL_POS>
```

Perintah di atas akan meng-install:

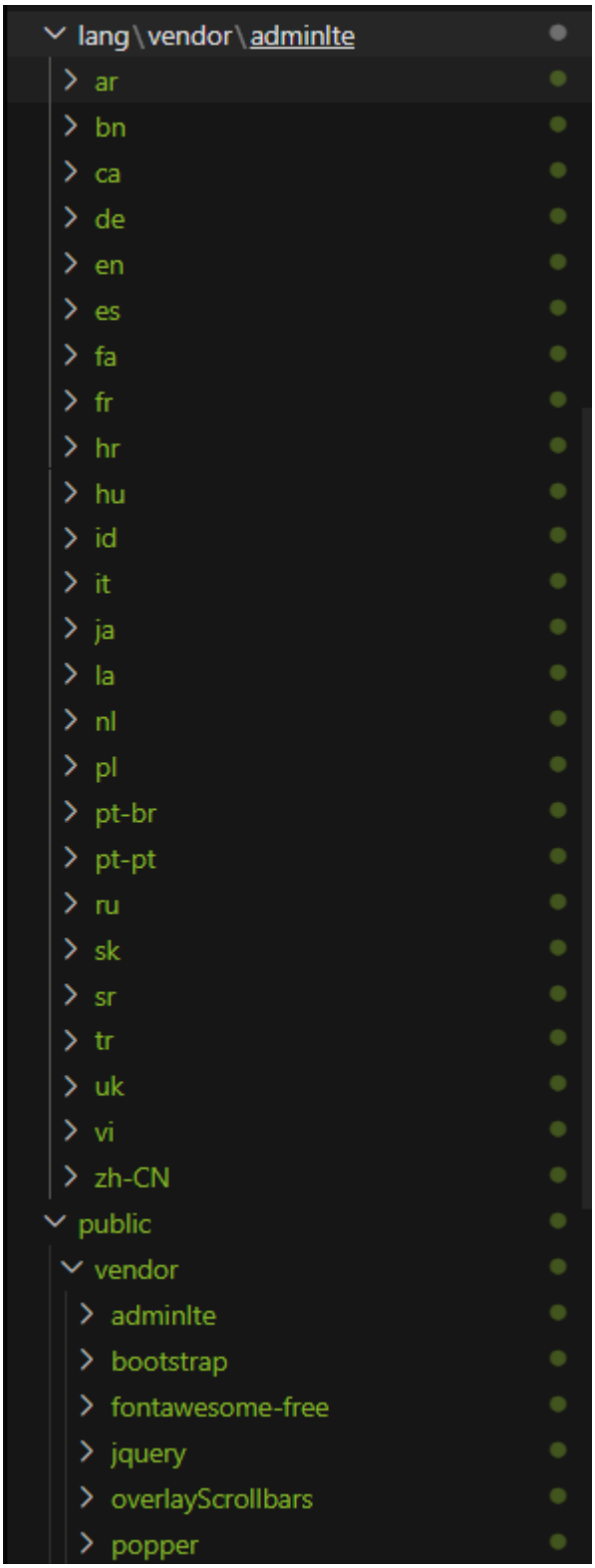
• AdminLTE distribution files dan dependensinya (Bootstrap, jQuery, etc.) dalam folder public/vendor .

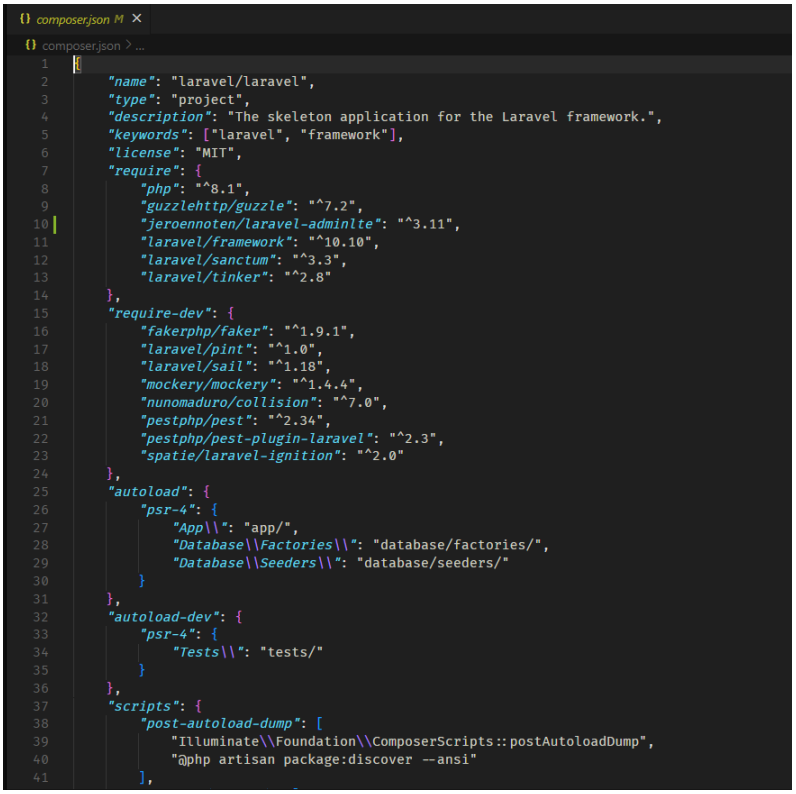
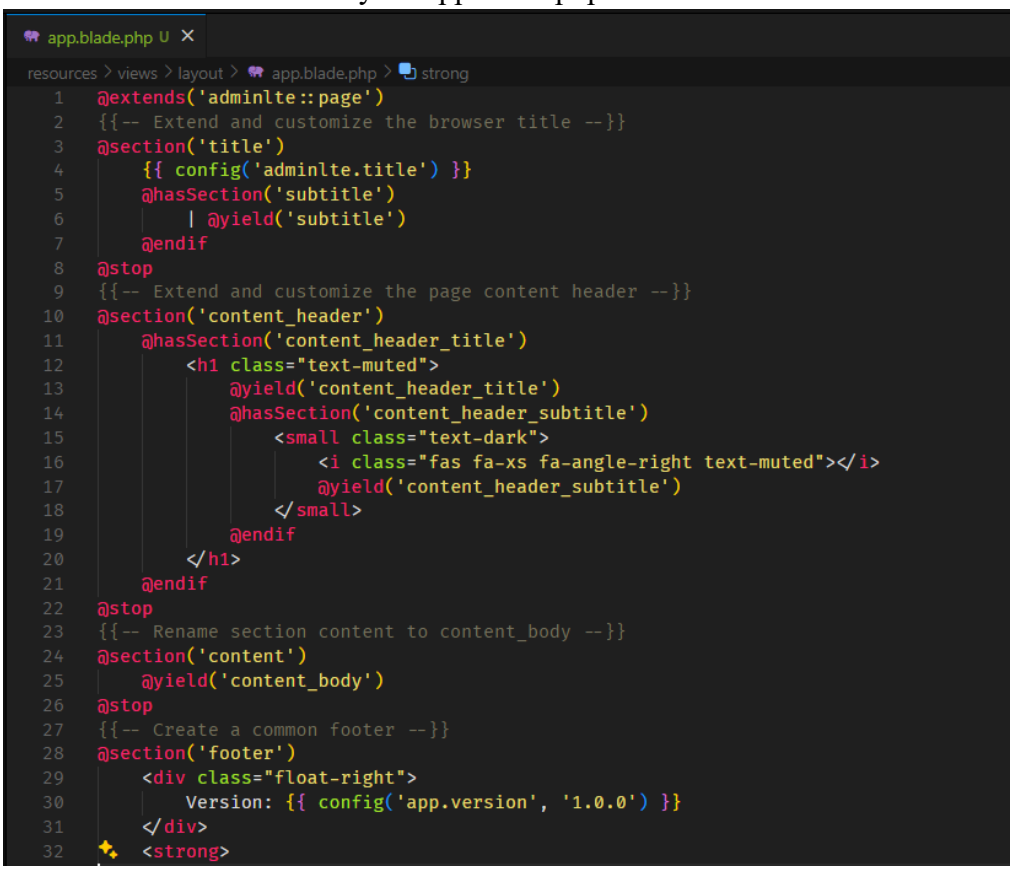


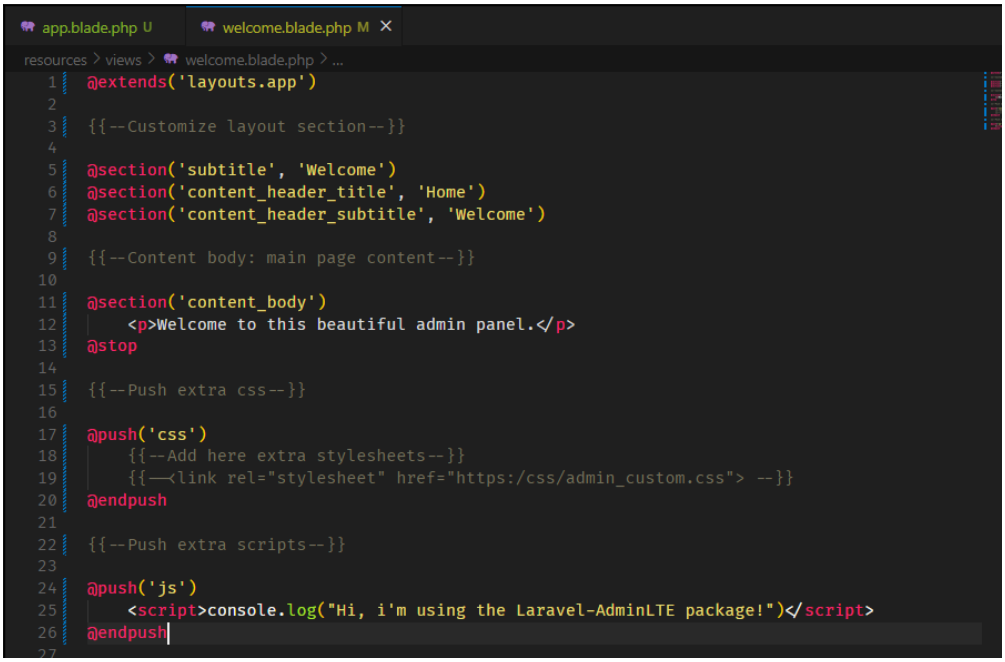
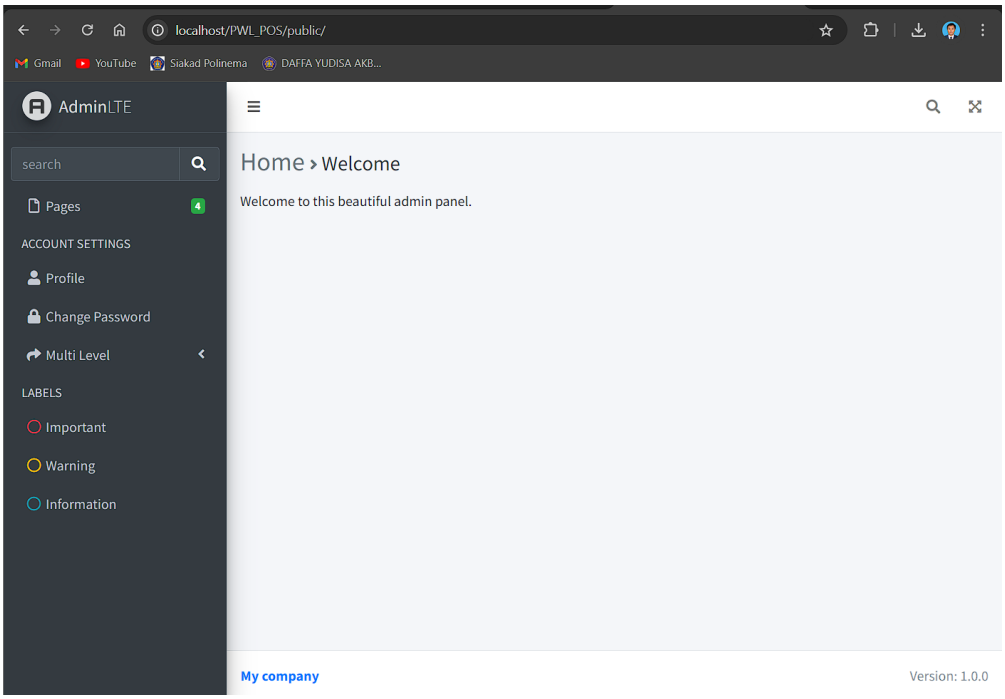
• Konfigurasi package di file config/adminlte.php



• Paket translasi di folder lang/vendor/adminlte/



	• Dalam composer.json akan otomatis ditambahkan require untuk laravel-adminlte 
3.	Buat file resources/views/layout/app.blade.php. 

4.	Edit resources/views/welcome.blade.php, kemudian replace seluruh kodenya dengan kode berikut  app.blade.php U welcome.blade.php M resources > views > welcome.blade.php > ... 1 @extends('layouts.app') 2 3 {{--Customize layout section--}} 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 Welcome to this beautiful admin panel. 13 @stop 14 15 {{--Push extra css--}} 16 17 @push('css') 18 {{--Add here extra stylesheets--}} 19 {{—<link rel="stylesheet" href="https:/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
5.	Kembali ke browser, menuju ke halaman awal.  localhost/PWL_POS/public/ Gmail YouTube Siakad Polinema DAFFA YUDISA AKB... AdminLTE search Pages 4 ACCOUNT SETTINGS Profile Change Password Multi Level LABELS Important Warning Information Home > Welcome Welcome to this beautiful admin panel. My company Version: 1.0.0

B. Praktikum 2 – Integrasi dengan DataTables

No.	Langkah Praktikum
1.	Install Laravel DataTables composer require laravel/ui –dev

```
PS C:\laragon\www\PWL_POS> 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 (v4.5.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
  - Downloading laravel/ui (v4.5.0)
  - Installing laravel/ui (v4.5.0): 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
  pestphp/pest-plugin-laravel ..... DONE
  spatie/laravel-ignition ..... DONE

90 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 ^4.5 for laravel/ui
PS C:\laragon\www\PWL_POS>
```

composer require yajra/laravel-datatables:^10.0

```
PS C:\laragon\www\PWL_POS> composer require yajra/laravel-datatables:^10.0
The "10.0" constraint for "yajra/laravel-datatables" appears too strict and will likely not match what you w
ant. See https://getcomposer.org/constraints
./composer.json has been updated
Running composer update yajra/laravel-datatables
Loading composer repositories with package information
Updating dependencies
Lock file operations: 7 installs, 0 updates, 0 removals
  - Locking league/fractal (0.20.1)
  - Locking yajra/laravel-datatables (v10.0.0)
  - Locking yajra/laravel-datatables-buttons (v10.0.0)
  - Locking yajra/laravel-datatables-editor (v1.25.4)
  - Locking yajra/laravel-datatables-fractal (v10.0.0)
  - Locking yajra/laravel-datatables-html (v10.12.0)
  - Locking yajra/laravel-datatables-oracle (v10.11.4)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 7 installs, 0 updates, 0 removals
  - Downloading yajra/laravel-datatables-oracle (v10.11.4)
  - Downloading yajra/laravel-datatables-html (v10.12.0)
  - Downloading league/fractal (0.20.1)
  - Downloading yajra/laravel-datatables-fractal (v10.0.0)
  - Downloading yajra/laravel-datatables-editor (v1.25.4)
  - Downloading yajra/laravel-datatables-buttons (v10.0.0)
  - Downloading yajra/laravel-datatables (v10.0.0)
  - Installing yajra/laravel-datatables-oracle (v10.11.4): Extracting archive
  - Installing yajra/laravel-datatables-html (v10.12.0): Extracting archive
  - Installing league/fractal (0.20.1): Extracting archive
  - Installing yajra/laravel-datatables-fractal (v10.0.0): Extracting archive
  - Installing yajra/laravel-datatables-editor (v1.25.4): Extracting archive
  - Installing yajra/laravel-datatables-buttons (v10.0.0): Extracting archive
  - Installing yajra/laravel-datatables (v10.0.0): Extracting archive
8 package suggestions were added by new dependencies, use `composer suggest` to see details.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
```

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

```
PS C:\laragon\www\PWL_POS> npm -v
10.2.4
PS C:\laragon\www\PWL_POS>
```

3. Install Laravel DataTables Vite dan sass

	npm i laravel-datatables-vite --save-dev
	`PS C:\laragon\www\PWL_POS> npm i laravel-datatables-vite --save-dev`
	
	`added 33 packages, and audited 34 packages in 27s`
	
	`8 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`
	`PS C:\laragon\www\PWL_POS>`
	
	npm install -D sass
	`PS C:\laragon\www\PWL_POS> npm install -D sass`
	
	`added 15 packages, and audited 49 packages in 9s`
	
	`10 packages are looking for funding`
	`  run 'npm fund' for details`
	
	`found 0 vulnerabilities`
	`PS C:\laragon\www\PWL_POS>`

4.	Edit file resources/js/app.js
	app.js
	resources > js > app.js
	`import './bootstrap';`
	`import '../sass/app.scss';`
	`import 'laravel-datatables-vite';`

5.	Buatlah file resources/saas/app.scss
	app.scss
	resources > sass > app.scss
	`// Fonts`
	`@import url('https://fonts.bunny.net/css?family=Nunito');`
	
	`// Bootstrap`
	`@import 'bootstrap/scss/bootstrap';`
	
	`// DataTables`
	`@import 'bootstrap-icons/font/bootstrap-icons.css';`
	`@import 'datatables.net-bs5/css/dataTables.bootstrap5.min.css';`
	`@import 'datatables.net-buttons-bs5/css/buttons.bootstrap5.min.css';`
	`@import 'datatables.net-select-bs5/css/select.bootstrap5.min.css';`

6.	Jalankan dengan npm run dev

```
PS C:\laragon\www\PWL_POS> npm run dev

> dev
> vite


  VITE v5.1.6  ready in 297 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help

  LARAVEL v10.47.0  plugin v1.0.2

  ➜  APP_URL: http://localhost
```

7. Selanjutnya kita akan buat datatables untuk kategori, dengan perintah php artisan datatables:make Kategori

```
PS C:\laragon\www\PWL_POS> php artisan datatables:make Kategori

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

8. Kita edit KategoriDatable untuk mengatur kolom apasaja yang ingin ditampilkan Sesuaikan dengan kodeprogram berikut

app > DataTables > KategoriDataTable.php > ...

```php
<?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\Button;
use Yajra\DataTables\Html\Column;
use Yajra\DataTables\Html\Editor\Editor;
use Yajra\DataTables\Html\Editor\Fields;
use Yajra\DataTables\Services\DataTable;

class KategoriDataTable extends DataTable
{
    /**
     * Build the DataTable class.
     *
     * @param QueryBuilder $query Results from query() method.
     */
    public function dataTable(QueryBuilder $query): EloquentDataTable
    {
        return (new EloquentDataTable($query))
            /* ->addColumn('action', 'kategori.action') */
            ->setRowId('id');
    }
    /**
     * Get the query source of dataTable.
     */
    public function query(KategoriModel $model): QueryBuilder
    {
        return $model->newQuery();
    }
    /**
     * Optional method if you want to use the html builder.
     */
```

app > DataTables > KategoriDataTable.php > ...

```php
class KategoriDataTable extends DataTable
    public function html(): HtmlBuilder
                    Button::make('csv'),
                    Button::make('pdf'),
                    Button::make('print'),
                    Button::make('reset'),
                    Button::make('reload')
                ]);
    }
    /**
     * Get the dataTable columns definition.
     */
    public function getColumns(): array
    {
        return [
            /* Column::computed('action')
    ->exportable(false)
    ->printable(false)
    ->width(60)
    ->addClass('text-center'), */
                Column::make('kategori_id'),
                Column::make('kategori_kode'),
                Column::make('kategori_nama'),
                Column::make('created_at'),
                Column::make('updated_at'),
        ];
    }
    /**
     * Get the filename for export.
     */
    protected function filename(): string
    {
        return 'Kategori_' . date('YmdHis');
    }
}
```

9. Ubah kategori model, sesuaikan seperti berikut

KategoriModel.php M

app > Models > KategoriModel.php > PHP Intelephense > KategoriModel > barang

```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, 'barang_id', 'barang_id');
    }
}
```

10. Ubah Kategori Controller sesuaikan dengan skrip berikut:

	app > Http > Controllers > KategoriController.php > PHP Intelephense > KategoriController > index
	`<?php`
	
	`namespace App\Http\Controllers;`
	
	`use App\DataTables\KategoriDataTable;`
	`use Illuminate\Http\Request;`
	`use Illuminate\Support\Facades\DB;`
	
	`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
	resources > views > kategori > index.blade.php
	`@extends('layouts.app')`
	
	`{{--Customixe Layout Sections--}}`
	
	`@section('subtitle', 'Kategori')`
	`@section('content_header_title', 'Home')`
	`@section('content_header_subtitle', 'Kategori')`
	
	`@section('content')`
	`<div class="container">`
	`<div class="card">`
	`<div class="card-header">Manager Kategori</div>`
	`<div class="card-body"> {{ $dataTable->table() }}</div>`
	`</div>`
	`</div>`
	`@endsection`
	
	`@push('scripts')`
	`{{ $dataTable->scripts() }}`
	`@endpush`

12.	Pastikan route untung kategori sudah tersedia `Route::get('/kategori', [KategoriController::class, 'index']);`
13.	Sesuaikan app layout seperti kode berikut

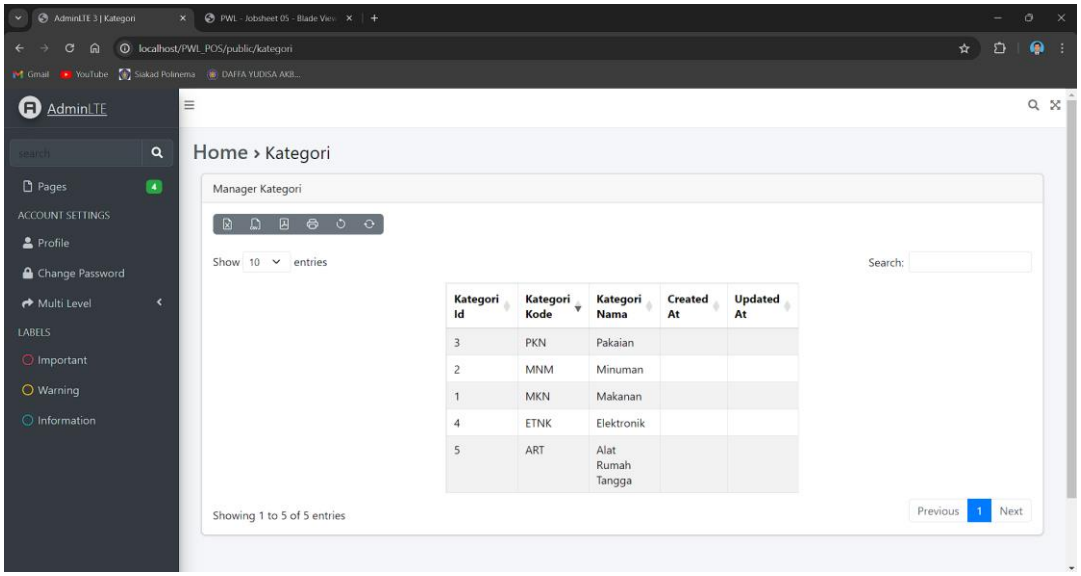
resources > views > layouts > app.blade.php > style

```blade
@extends('adminlte::page')
{{-- Extend and customize the browser title --}}
@section('title')
    {{ config('adminlte.title') }}
    @hasSection('subtitle')
        | @yield('subtitle')
    @endif
@stop
@vite('resources/js/app.js')
{{-- Extend and customize the page content header --}}
@section('content_header')
    @hasSection('content_header_title')
        <h1 class="text-muted">
            @yield('content_header_title')
            @hasSection('content_header_subtitle')
                <small class="text-dark">
                    <i class="fas fa-xs fa-angle-right text-muted"></i>
                    @yield('content_header_subtitle')
                </small>
            @endif
        </h1>
    @endif
@stop
{{-- Rename section content to content_body --}}
@section('content')
    @yield('content_body')
@stop
{{-- Create a common footer --}}
@section('footer')
    <div class="float-right">
        Version: {{ config('app.version', '1.0.0') }}
    </div>
    <strong>
        <a href="{{ config('app.company_url', '#') }}">
            {{ config('app.company_name', 'My company') }}
        </a>
    </strong>
@stop
{{-- Add common Javascript/Jquery code --}}
@push('js')
    <script src="https://cdn.datatables.net/2.0.2/js/dataTables.js"></script>
@endpush
@stack('scripts')
{{-- Add common CSS customizations --}}
@push('css')
    <link rel="stylesheet" href="https://cdn.datatables.net/2.0.2/css/dataTables.dataTables.css" />
    <style type="text/css">
        {{-- You can add AdminLTE customizations here --}}
        /*
       .card-header {
       border-bottom: none;
       }
       .card-title {
       font-weight: 600;
       }
       */
    </style>
@endpush

```

14. Menset ViteJs / script type defaults

app > Providers > AppServiceProvider.php > ...

```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.	Datatables sudah dapat di load di /kategori 

C. Praktikum 3 – Membuat form kemudian menyimpan data dalam database

No.	Langkah Praktikum

1.	Menyesuaikan routing, tambahkan dua routing berikut
	`Route::get('/kategori/create', [KategoriController::class, 'create']);`
	`Route::post('/kategori', [KategoriController::class, 'store']);`

2.	Tambahkan dua function berikut dalam KategoriController

KategoriModel.php | KategoriController.php M | KategoriDataTable.php | create.blade.php M | index.blade.php M

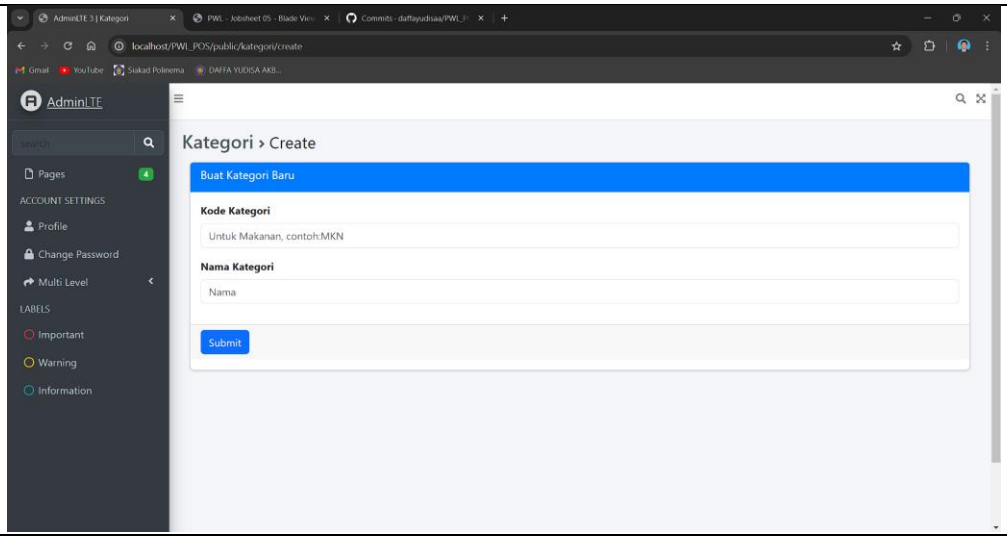
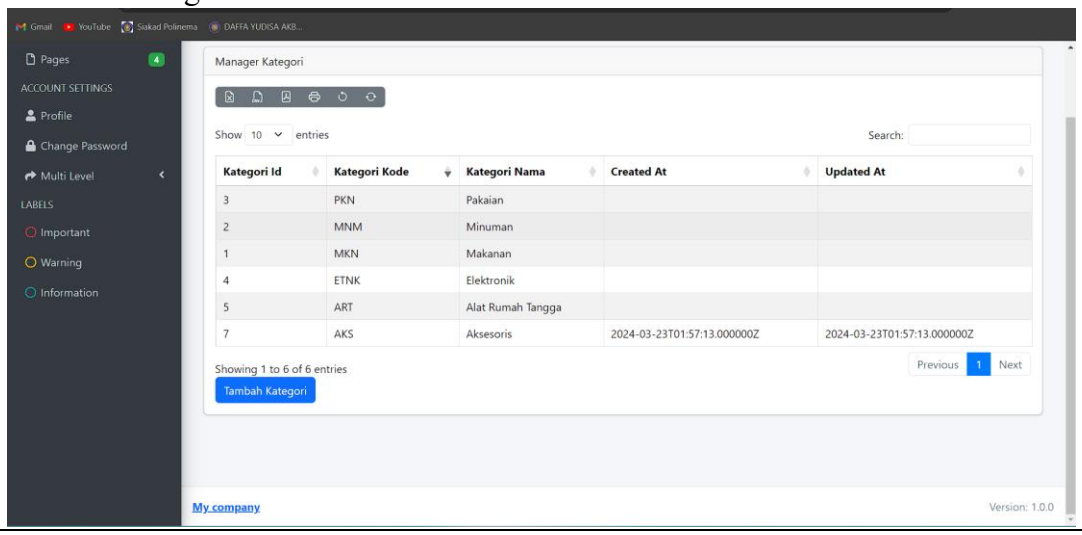
app > Http > Controllers > KategoriController.php > PHP Intelephense > KategoriController > store

```php
<?php

namespace App\Http\Controllers;

use App\DataTables\KategoriDataTable;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use App\Models\KategoriModel;

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 $request)
    {
        //dd($request->all());
        KategoriModel::create([
            'kategori_kode' => $request->kategori_kode,
            'kategori_nama' => $request->kategori_nama,
        ]);
        return redirect('/kategori');
    }

```

3.	Dalam folder views/kategori, buatlah file dengan nama create.blade.php resources > views > kategori > create.blade.php > div.container > div.card.card-primary > form > div.card-body > div.form-group > input#kategori_nama.form-control {{-- Customize layout sections --}} @section('subtitle', 'Kategori') @section('content_header_title', 'Kategori') @section('content_header_subtitle', 'Create') {{-- Content body: main page content --}} @section('content') <h3 class="card-title">Buat Kategori Baru</h3> <form method="POST" action="../kategori"> <label for="kodeKategori">Kode Kategori</label> <input type="text" class="form-control" id="kategori_kode" name="kategori_kode" placeholder="Untuk Makanan, contoh:MKN"> <label for="namaKategori">Nama Kategori</label> <input type="text" class="form-control" id="kategori_nama" name="kategori_nama" placeholder="Nama"> <button type="submit" class="btn btn-primary">Submit</button> </form> @endsection

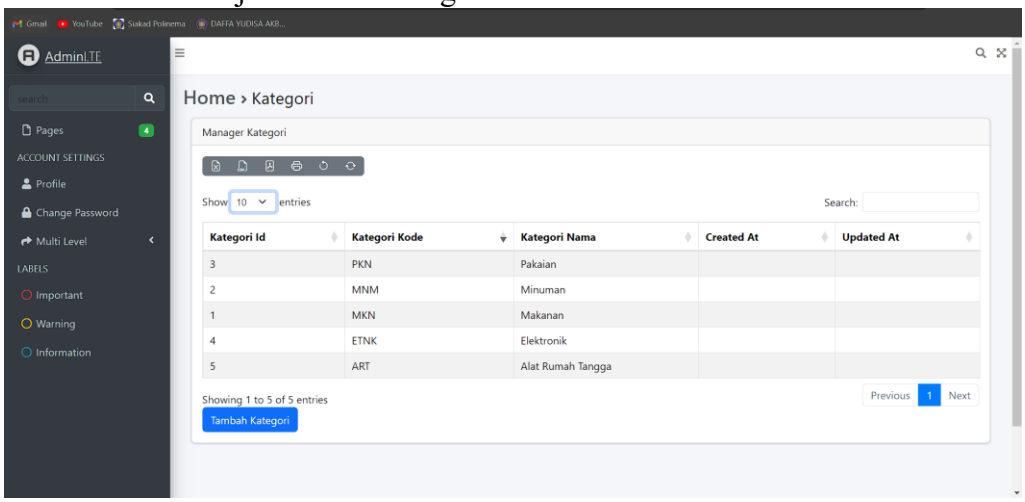
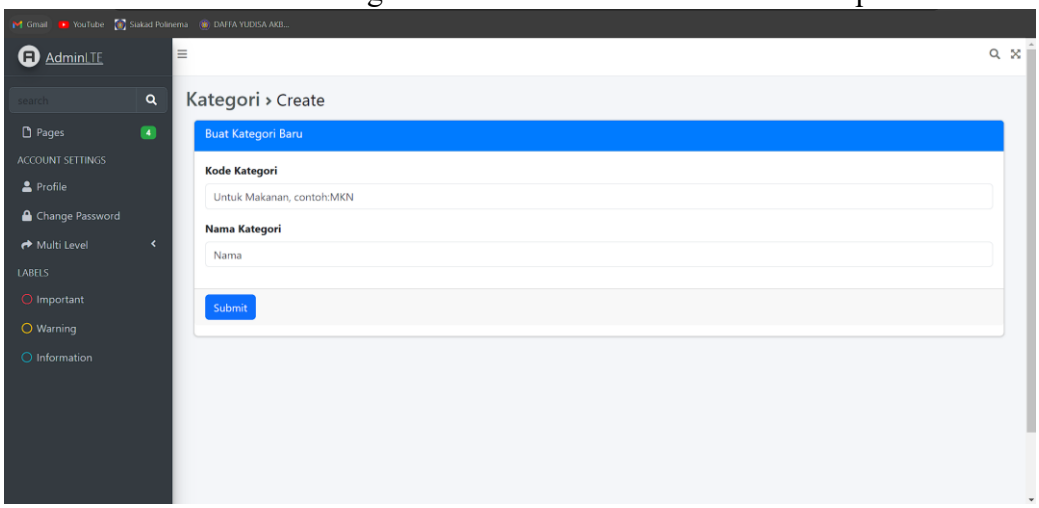
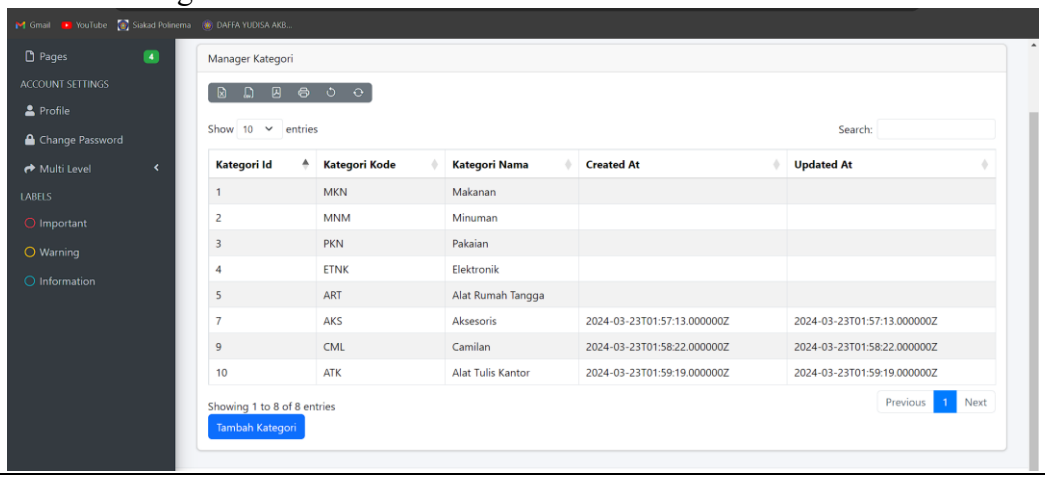
4.	Kita lakukan pengecualian proteksi CsrfToken. Karena kita belum melakukan otentikasi
	. Kita edit dalam file berikut:
	app > Http > Middleware > VerifyCsrfToken.php > ...
	<?php
	
	namespace App\Http\Middleware;
	
	use Illuminate\Foundation\Http\Middleware\VerifyCsrfToken as Middleware;
	
	class VerifyCsrfToken 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 

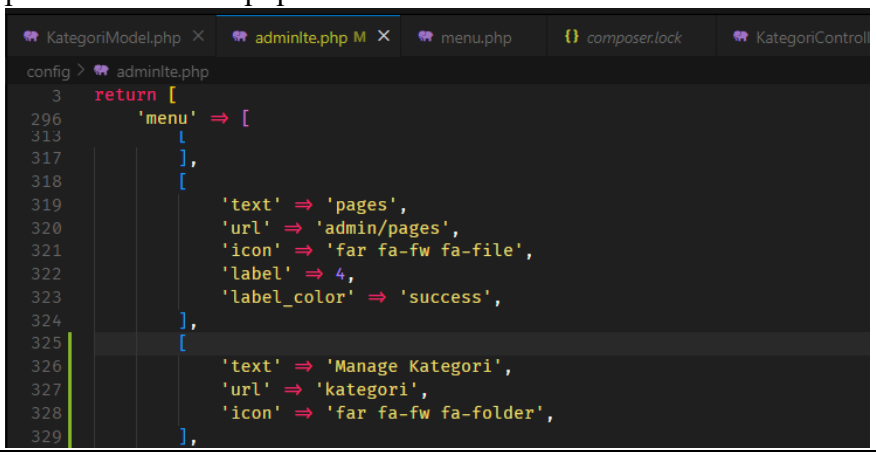
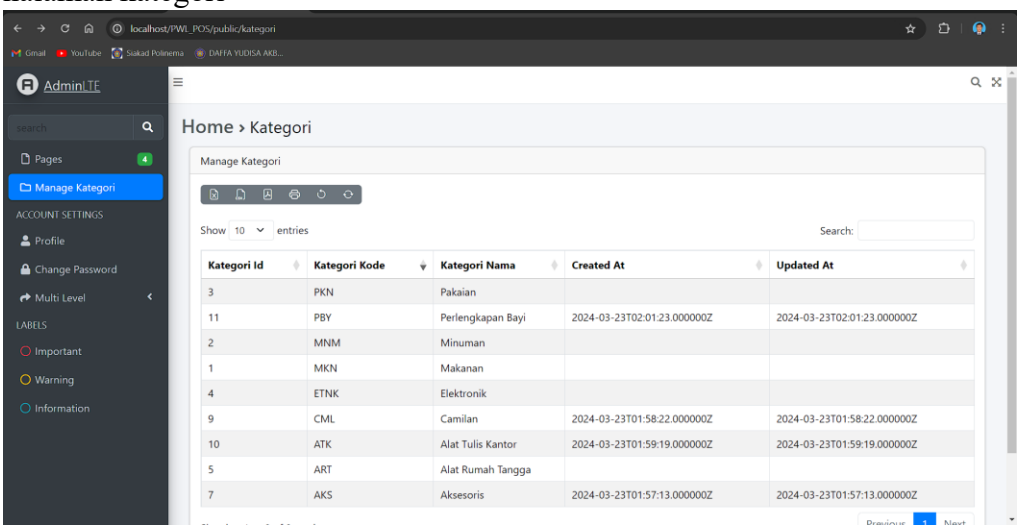
- **Tugas 1 : Tambahkan button Add di halam manage kategori, yang mengarah ke create kategori baru**

No.	Hasil Screenshot
1.	Penambahan button Tambah Kategori yang akan menuju pada create kategori pada file index.blade.php resources > views > kategori > index.blade.php > ... 1 @extends('layouts.app') 2 3 {{--Customixe Layout Sections--}} 4 5 @section('subtitle', 'Kategori') 6 @section('content_header_title', 'Home') 7 @section('content_header_subtitle', 'Kategori') 8 9 @section('content') 10 11 12 Manager Kategori 13 {{ $dataTable->table() }} 14 <a class="btn btn-primary" href={{url("/kategori/create")}}>Tambah Kategori</a> 15 16 17 18 @endsection 19 20 @push('scripts') 21 {{ $dataTable->scripts() }} 22 @endpush

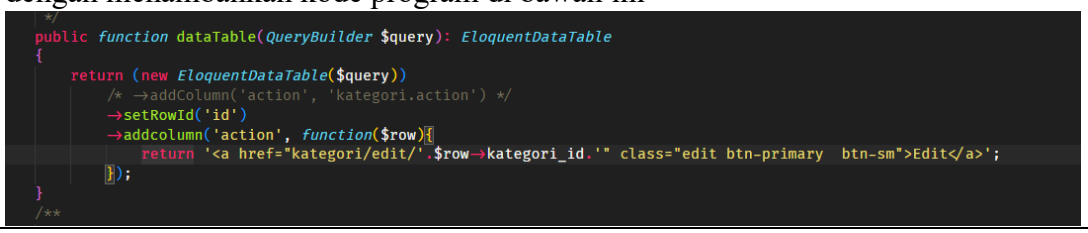
2.	Hasil ketika menuju halaman kategori  Ketika tombol Tambah Kategori diklik maka akan muncul form seperti berikut  Setelah mengisi form dan menyimpan kategori, maka akan diarahkan menuju halaman kategori 

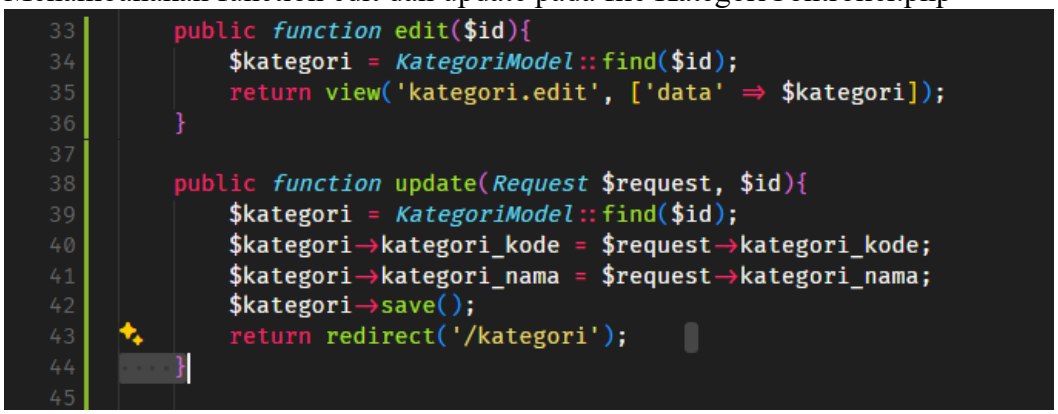
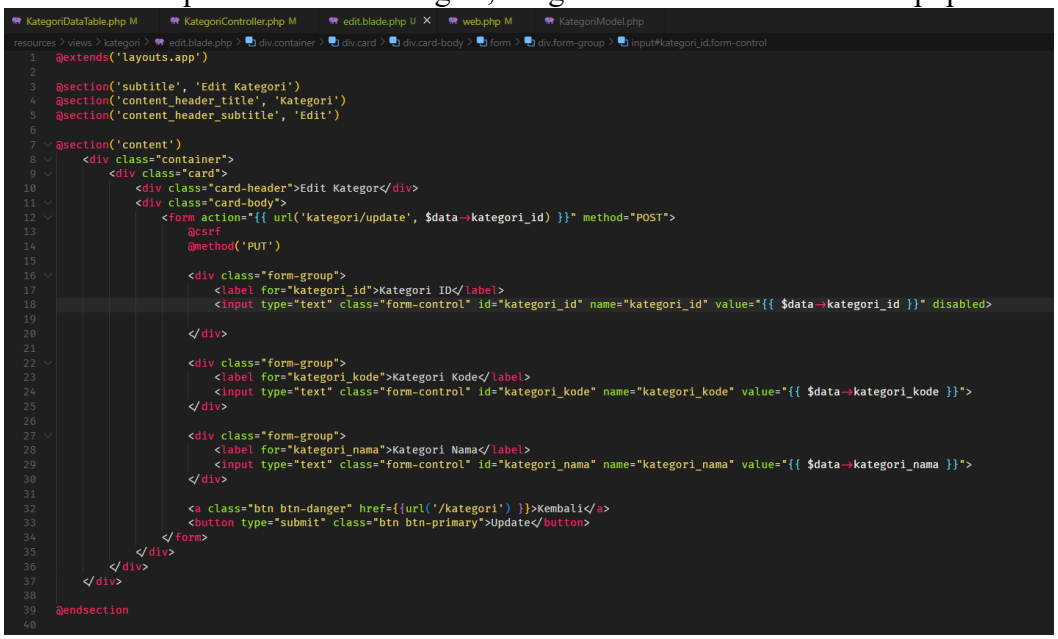
- **Tugas 2: Tambahkan menu untuk halaman manage kategori, di daftar menu navbar**

No.	Hasil Screenshot

1.	Menambahkan Manage Kategori pada sidebar dengan menambahkan kode berikut pada file adminlte.php 
2.	Manage Kategori berhasil ditambahkan pada sidebar dan ketika diklik akan menuju halaman kategori 

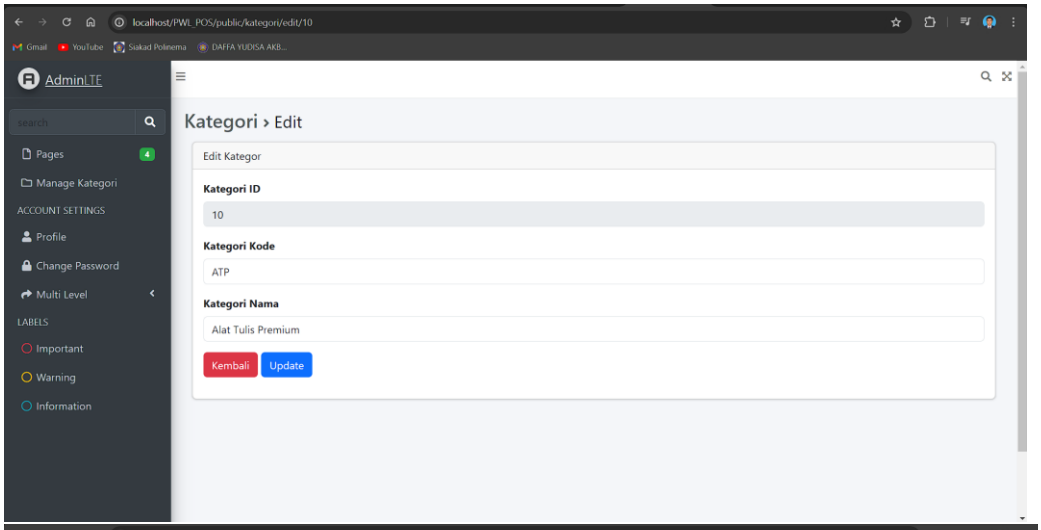
- **Tugas 3: Tambahkan action edit di datatables dan buat halaman edit serta controllernya**

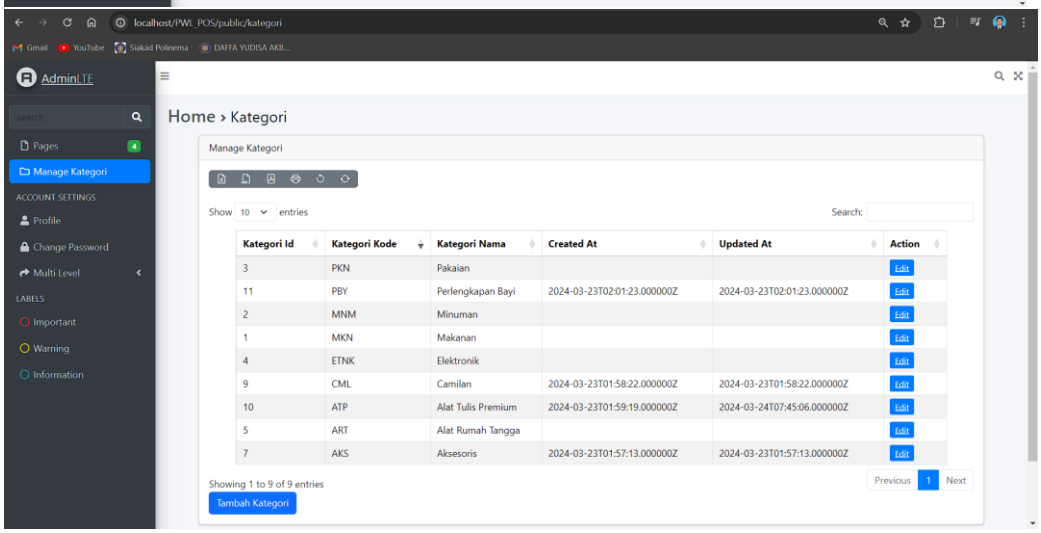
No.	Hasil Screenshot
1.	Membuat kolom action yang berisi tombol edit pada file KategoriDataTable.php dengan menambahkan kode program di bawah ini 

2.	Menambahakan function edit dan update pada file KategoriController.php 
3.	Membuat tampilan form edit kategori, dengan membuat file edit.blade.php 
4.	Menyiapkan route pada file web.php

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

5. Hasil:





- **Tugas 4: Tambahkan action delete di datatables serta controllernya**

No.	Hasil Screenshot
1.	Menambahkan tombol delete pada kolom action pada file KategoriDataTable.php
2.	Menambahkan fungsi destroy pada file KategoriController.php

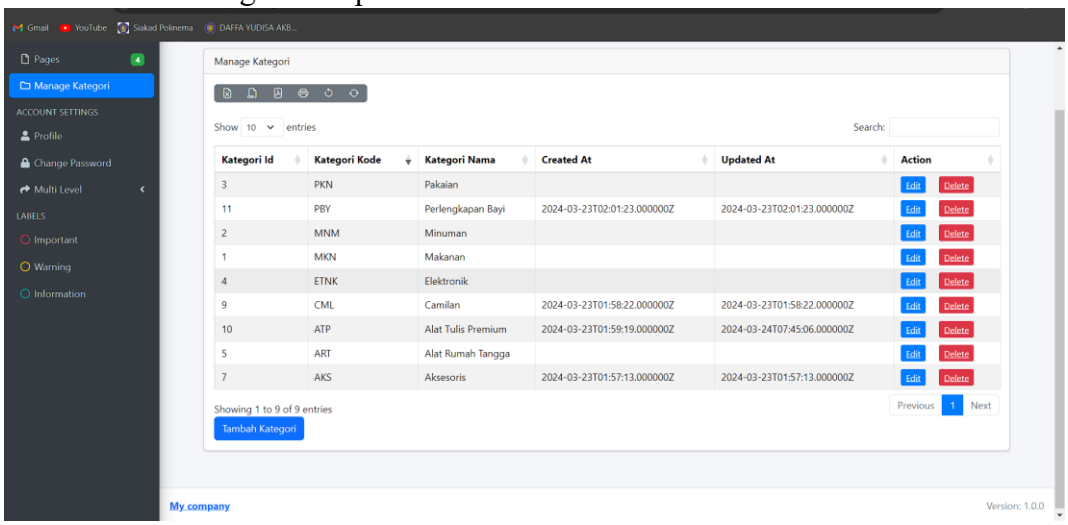
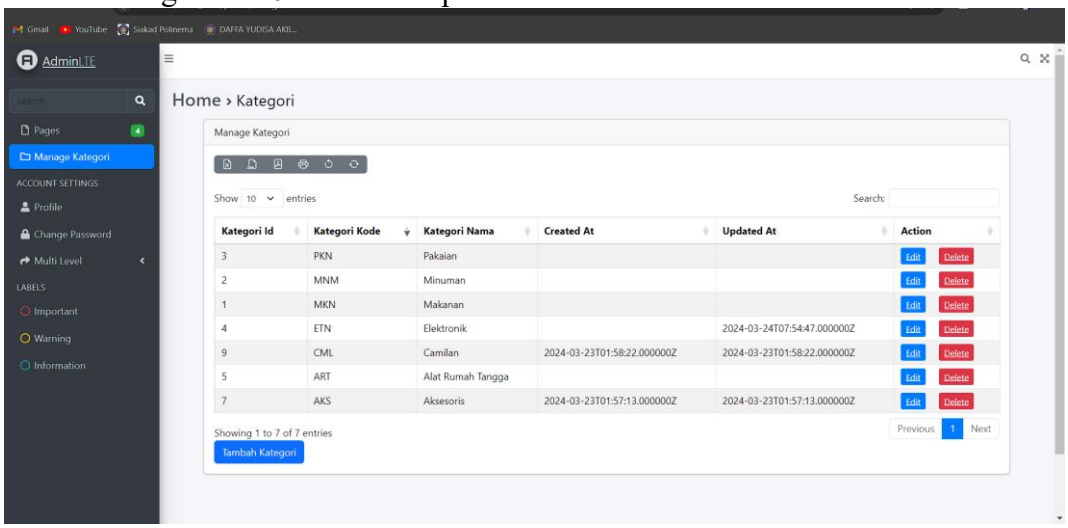
```
public function dataTable(QueryBuilder $query): EloquentDataTable
{
    return (new EloquentDataTable($query))
        /* ->addColumn('action', 'kategori.action') */
        ->setRowId('id')
        ->addcolumn('action', function($row){
            $action = '<a href="kategori/edit/'.$row->kategori_id.'" class="edit btn-primary  btn-sm">Edit</a>';
            $action .= '     ';
            $action .= '<a href="kategori/delete/'.$row->kategori_id.'" class="delete btn-danger  btn-sm">Delete</a>';
            return $action;
        });
}
```

```php
public function destroy($id) {
    KategoriModel::find($id)->delete();

    return redirect('/kategori');
}
```

3. Menyiapkan route seperti gambar di bawah ini

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

4. Sebelum data kategori dihapus



Setelah kategori id 10 dan 11 dihapus