

Nama : Adinda Ivanka Maysanda Putri

Kelas : SIB 2B

NIM : 2341760058

JOBSHEET 5

Blade View, Web Templating(AdminLTE), Datatables

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

```
PS C:\laragon\www\Pemrograman-Web-Lanjut\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
  - Downloading almasaeed2010/adminlte (v3.2.0)
  - Downloading jeroennoten/laravel-adminlte (v3.15.0)
  - 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
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].
```

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

php artisan adminlte:install

```
PS C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 5> php artisan adminlte:install
AdminLTE assets files published successfully
Configuration file published successfully
Translation files published successfully
The installation is complete.
PS C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 5> |
```

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 > {} autoload > {} psr-4
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     },
15     "require-dev": {
16         "fakerphp/faker": "^1.9.1",
17         "laravel/pint": "^1.0",
18         "laravel/sail": "^1.18",
19         "mockery/mockery": "^1.4.4",
20         "nunomaduro/collision": "^7.0",
21         "phpunit/phpunit": "^10.1",
22         "spatie/laravel-ignition": "^2.0"

```

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

```

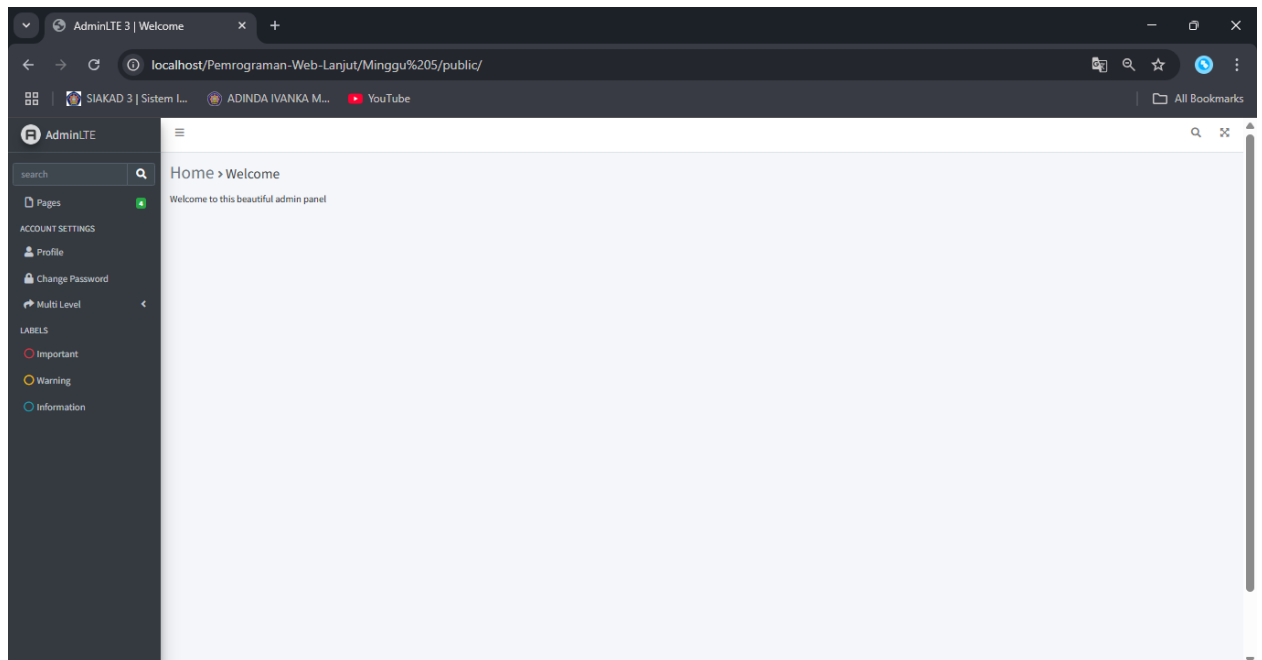
resources > views > layout > app.blade.php > ...
1  @extends('adminlte::page')
2
3  {{-- Extend and customize the browser title --}}
4
5  @section('title')
6      {{ config('adminlte.title') }}
7      @hasSection('subtitle') | @yield('subtitle') @endif
8  @stop
9
10 {{-- Extend and customize the page content header --}}
11
12 @section('content_header')
13     @hasSection('content_header_title')
14         <h1 class="text-muted">
15             @yield('content_header_title')
16
17             @hasSection('content_header_subtitle')
18                 <small class="text-dark">
19                     <i class="fas fa-xs fa-angle-right text-muted"></i>
20                     @yield('content_header_subtitle')
21                 </small>
22             @endif
23         </h1>
24     @endif
25 @stop
26
27 {{-- Rename section content to content_body --}}
28
29 @section('content')
30     @yield('content_body')
31 @stop
32

```

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

```
resources > views > welcome.blade.php > ...
1  @extends('layouts.app')
2
3  {{-- Customize Layout sections --}}
4  @section('subtitle', 'Welcome')
5
6  @section('content_header_title', 'Home')
7  @section('content_header_subtitle', 'Welcome')
8
9  {{-- Content body: main page content --}}
10 @section('content_body')
11 |
12 |     <p>Welcome to this beautiful admin panel.</p>
13 |
14 | @stop
15
16 {{-- Push extra CSS --}}
17 @push('css')
18 |
19 |     {{-- Add here extra stylesheets --}}
20 |     {{-- <link rel="stylesheet" href="/css/admin_custom.css"> --}}
21 |
22 | @endpush
23
24 {{-- Push extra scripts --}}
25 @push('js')
26 |
27 |     <script> console.log("Hi, I'm using the Laravel-AdminLTE package!"); </script>
28 |
29 | @endpush
30
```

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

```
PS C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 5> composer require laravel/ui --dev
laravel/ui is currently present in the require key and you ran the command with the --dev flag, which will move it to the require-dev key.
Do you want to move this requirement? [no]? no
Do you want to re-run the command without --dev? [yes]? yes
./composer.json has been updated
Running composer update laravel/ui
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Writing lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

INFO Discovering packages.

jeroennoten/laravel-adminlte ..... DONE
laravel/sail ..... DONE

PS C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 5> 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 want. 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 (0.20.2)
- 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)
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)
```

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

```
PS C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 5> npm -v
10.9.2
```

Npm digunakan untuk manajemen package javascript.

3. Install Laravel DataTables Vite dan sass

```
PS C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 5> npm i laravel-datatables-vite --save-dev

added 9 packages, and audited 63 packages in 35s

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
```

```

PS C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 5> npm install -D sass

up to date, audited 63 packages in 7s

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

```

4. Edit file resources/js/app.js

```

import './bootstrap';
import '../sass/app.scss';
import 'laravel-datatables-vite';

```

5. Buatlah file 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.css';

```

6. Jalankan dengan npm run dev

```

> dev
> vite

Port 5173 is in use, trying another one...

VITE v5.4.14 ready in 1629 ms

→ Local:   http://localhost:5174/
→ 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

```

PS C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 5> php artisan datatables:make Kategori

INFO  DataTable [C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 5\app\DataTables\KategoriDataTable.php] created successfully.

```

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

```

<?php

namespace App\DataTables;

```

```

use App\Models\Kategori;
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.
     */
    public function html(): HtmlBuilder
    {
        return $this->builder()
            ->setTableId('kategori-table')
            ->columns($this->getColumns())
            ->minifiedAjax()
            ->dom('<"row"<"col-md-12 mb-2"B>>' . // Tombol export di atas, rata kiri
                '<"row align-items-center"<"col-md-6"l><"col-md-6 text-end"f>>' . // Show entries
                & search sejajar
                'rt' . // Tabel
                '<"row align-items-center"<"col-md-6 d-flex align-items-center"i><"col-md-6 d-
                flex justify-content-end p-0"p>>') // Pagination tanpa jarak
    }
}

```

```

->orderBy(1)
->selectStyleSingle()
->buttons([
    Button::make('excel'),
    Button::make('csv'),
    Button::make('pdf'),
    Button::make('print'),
    Button::make('reset'),
    Button::make('reload')
]);
}
/*
 * Get the dataTable columns definition.
 */
public function getColumns(): array
{
    return [
        // Column::computed('action')
        //     ->exportable(false)
        //     ->printable(false)
        //     ->width(60)
        //     ->addClass('text-center'),
        Column::make('kategori_id'),
        Column::make('kategori_kode'),
        Column::make('kategori_nama'),
        Column::make('created_at'),
        Column::make('updated_at'),
    ];
}

/**
 * Get the filename for export.
 */
protected function filename(): string
{
    return 'Kategori_' . date('YmdHis');
}
}

```

9. Ubah kategori model, sesuaikan seperti berikut

```

use Illuminate\Database\Eloquent\Relations\HasMany;

class KategoriModel extends Model{

    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:

```
<?php  
  
namespace App\Http\Controllers;  
  
use App\DataTables\KategoriDataTable;  
use App\Models\KategoriModel;  
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\DB;  
  
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');  
    }  
  
    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,  
        ]);  
    }  
  
    public function delete($id){  
        KategoriModel::where('kategori_id', $id)->delete();  
        return redirect('/kategori');
```

```
}

}
```

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

The screenshot shows the VS Code interface with the Explorer sidebar on the left. The 'resources' folder is expanded, showing subfolders like 'css', 'js', 'sass', 'views', 'auth', and 'kategori'. The 'kategori' folder is selected, and a new file 'index.blade.php' is being created. The main editor area shows the content of the new file, which is a Blade template extending 'layouts.app' and using sections for 'subtitle', 'content_header_title', 'content_header_subtitle', and 'content'. The template includes a card for managing categories with a table and a button to add new categories. The terminal at the bottom shows the command prompt for the project.

```
resources > views > kategori > index.blade.php > ...
1 @extends('layouts.app')
2
3 {{-- Customize layout sections --}}
4 @section('subtitle', 'Kategori')
5 @section('content_header_title', 'Home')
6 @section('content_header_subtitle', 'Kategori')
7
8 @section('content')
9     <div class="container">
10         <div class="card">
11             <div class="card-header">Manage Kategori</div>
12             <div class="card-body">
13                 <a href="{{ url('/kategori/create') }}" class="btn btn-success">Add</a>
14                 <br><br>
15                 {!! $dataTable->table() !!}
16             </div>
17         </div>
18     </div>
19 @endsection
20
21 @push('scripts')
22     {!! $dataTable->scripts() !!}
23 @endpush
24
```

12. Pastikan route untuk kategori sudah tersedia

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

13. Sesuaikan app layout seperti kode berikut

```
@extends('adminlte::page')

{{-- Extend and customize the browser title --}}

@section('title')
{{ config('adminlte.title') }}
@hasSection('subtitle') | @yield('subtitle') @endif
@stop

@vite('resources/js/app.js')

{{-- Extend and customize the page content header --}}

@section('content_header')
@hasSection('content_header_title')
<h1 class="text-muted">
    @yield('content_header_title')
@hasSection('content_header_subtitle')
```

```

        <small class="text-dark">
            <i class="fas fa-xs fa-angle-right text-muted"></i>
            @yield('content_header_subtitle')
        </small>
    @endif
</h1>
@endif
@stop

{{ -- Rename section content to content_body -- }}

@section('content')
@yield('content_body')
@stop

{{ -- Create a common footer -- }}

@section('footer')
<div class="float-right">
    Version: {{ config('app.version', '1.0.0') }}
</div>

<strong>
    <a href="{{ config('app.company_url', '#') }}">
        {{ config('app.company_name', 'My company') }}
    </a>
</strong>
@stop

{{ -- Add common Javascript/Jquery code -- }}

@push('js')
    <script src="https://cdn.datatables.net/2.0.2/js/dataTables.js"></script>

@endpush

@stack('scripts')

{{ -- Add common CSS customizations -- }}

@push('css')
    <link rel="stylesheet" href="https://cdn.datatables.net/2.0.2/css/dataTables.dataTables.css"
/>
    <style type="text/css">

```

```

    {
        {
            -- You can add AdminLTE customizations here --
        }
    }

    /*
    .card-header {      border-bottom: none;
    }
    .card-title {      font-weight: 600;
    }
    */
</style>

```

@endpush

14. Menset ViteJs / script type defaults

```

<?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
    {
        //
    }

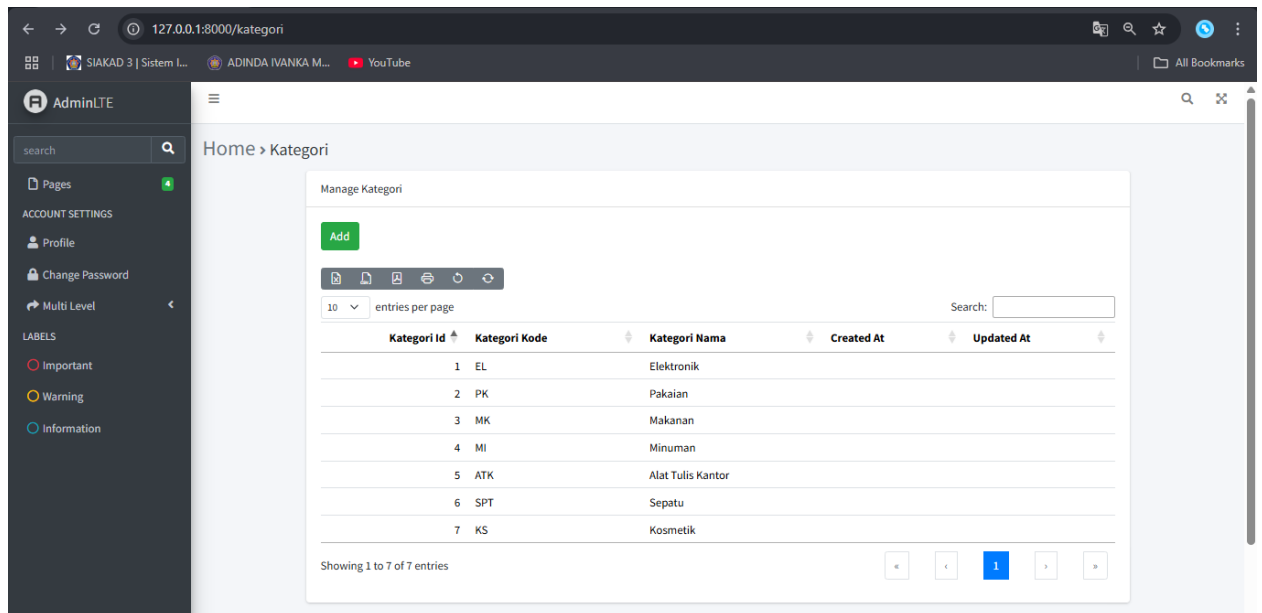
    /**
     * Bootstrap any application services.
     */
    public function boot(): void
    {
        Builder::useVite(); //Menggunakan Vite sebagai bundler aset frontend
    }
}

```

15. Isikan beberapa data ke table kategori

<input type="checkbox"/>	 Edit	 Copy	 Delete	6 SPT	Sepatu	NULL	NULL
<input type="checkbox"/>	 Edit	 Copy	 Delete	7 KS	Kosmetik	NULL	NULL

16. Datatables sudah dapat di load di /kategori



Praktikum 3 –

Membuat form kemudian menyimpan data dalam database Langkah-langkahnya adalah sebagai berikut:

1. Menyesuaikan routing, tambahkan dua routing berikut

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

2. Tambahkan dua function berikut dalam KategoriController

```
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

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

{{-- Customize layout sections --}}
@section('subtitle', 'Kategori')
```

```

@section('content_header_title', 'Kategori')
@section('content_header_subtitle', 'Create')

{{ -- Content body: main page content -- }}
@section('content')

<div class="container">
    <div class="card card-primary">
        <div class="card-header">
            <h3 class="card-title">Buat kategori baru</h3>
        </div>

        <form method="post" action=" ../kategori">
            <div class="card-body">
                <div class="form-group">
                    <label for="kodeKategori">Kode Kategori</label>
                    <input type="text" class="form-control" id="kodeKategori"
name="kodeKategori" placeholder="Masukkan kode kategori">
                </div>

                <div class="form-group">
                    <label for="namaKategori">Nama Kategori</label>
                    <input type="text" class="form-control" id="namaKategori"
name="namaKategori" placeholder="Masukkan nama kategori">
                </div>
            </div>

            <div class="card-footer">
                <button type="submit" class="btn btn-primary">Submit</button>
            </div>
        </form>
    </div>
</div>

@endsection

```

4. Kita lakukan pengecualian proteksi CsrfToken. Karena kita belum melakukan otentikasi .

Kita edit dalam file berikut:

```

<?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.
     */
}

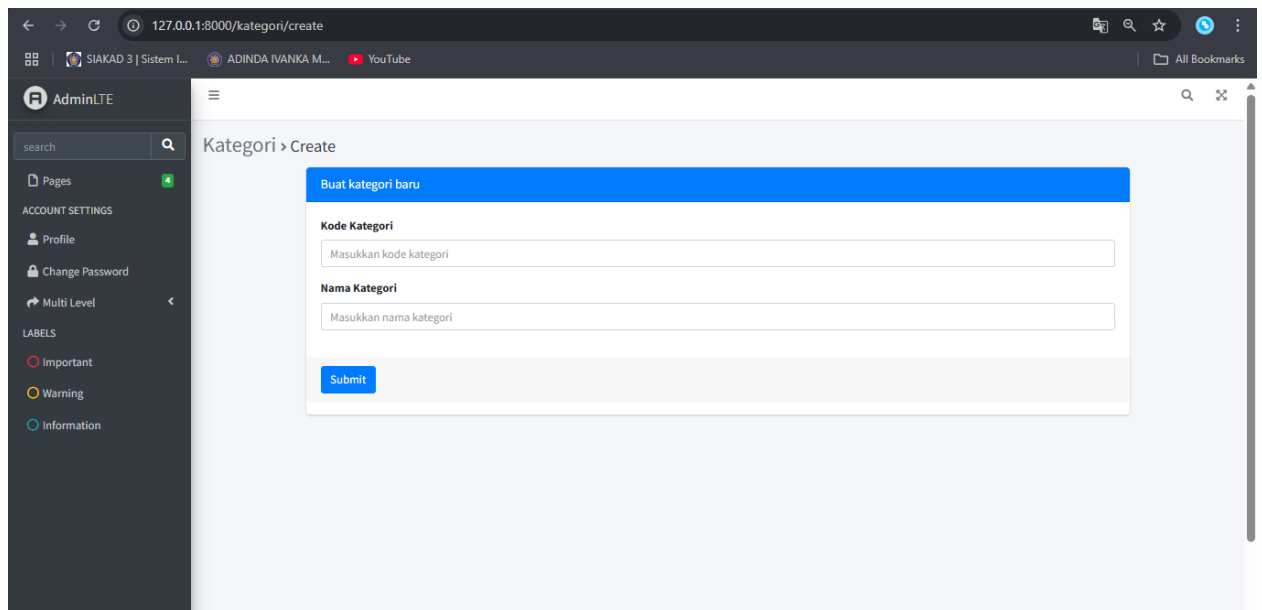
```

```

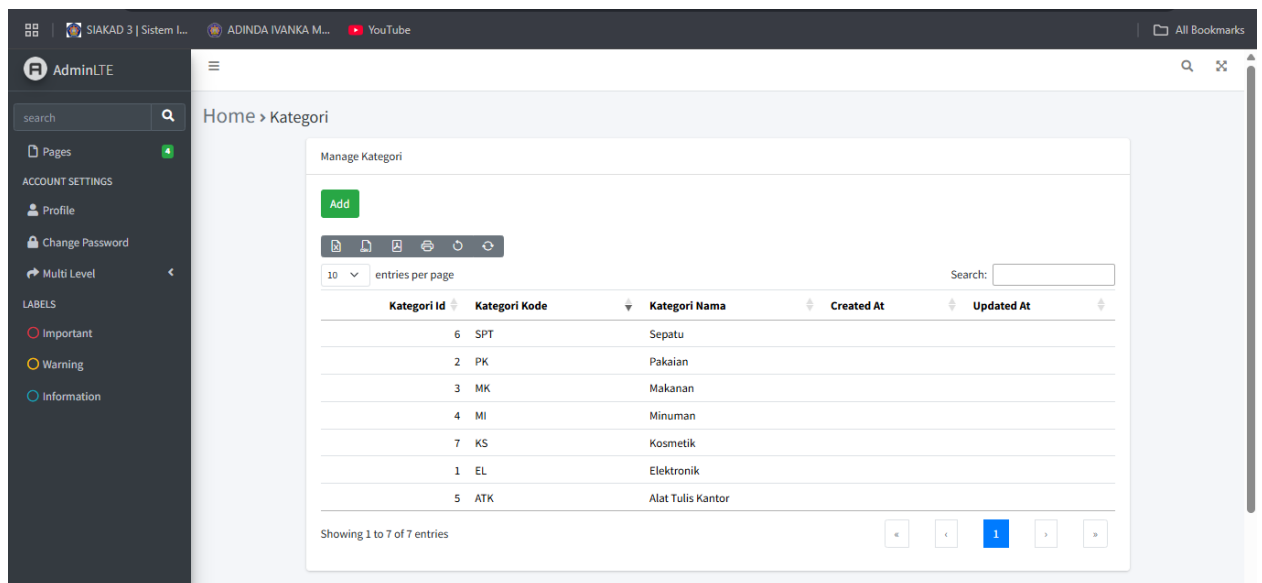
* @var array<int, string>
*/
protected $except = [
    '/kategori'
];
}

```

5. Akses kategori/create



6. Halaman kategori



Tugas Praktikum

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

```

@extends('layouts.app')

```

```

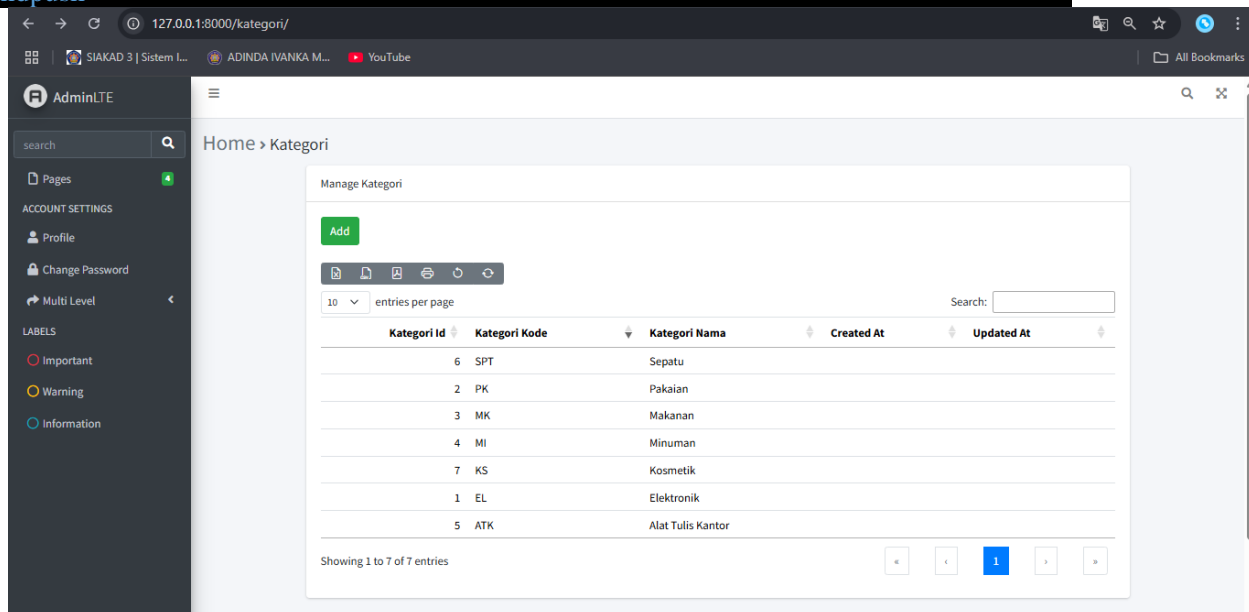
{{-- Customize Layout sections --}}
@section('subtitle', 'Kategori')

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

@section('content')
<div class="container">
  <div class="card">
    <div class="card-header">Manage Kategori</div>
    <div class="card-body">
      <a href="{{ { url('/kategori/create') } }}" class="btn btn-success">Add</a>
      <br><br>
      {!! $dataTable->table() !!}
    </div>
  </div>
</div>
</div>
@endsection

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

```



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

```

'menu' => [
  // Navbar items:
  [
    'type' => 'navbar-search',
    'text' => 'search',
    'topnav_right' => true,
  ],
  [
    'type' => 'fullscreen-widget',

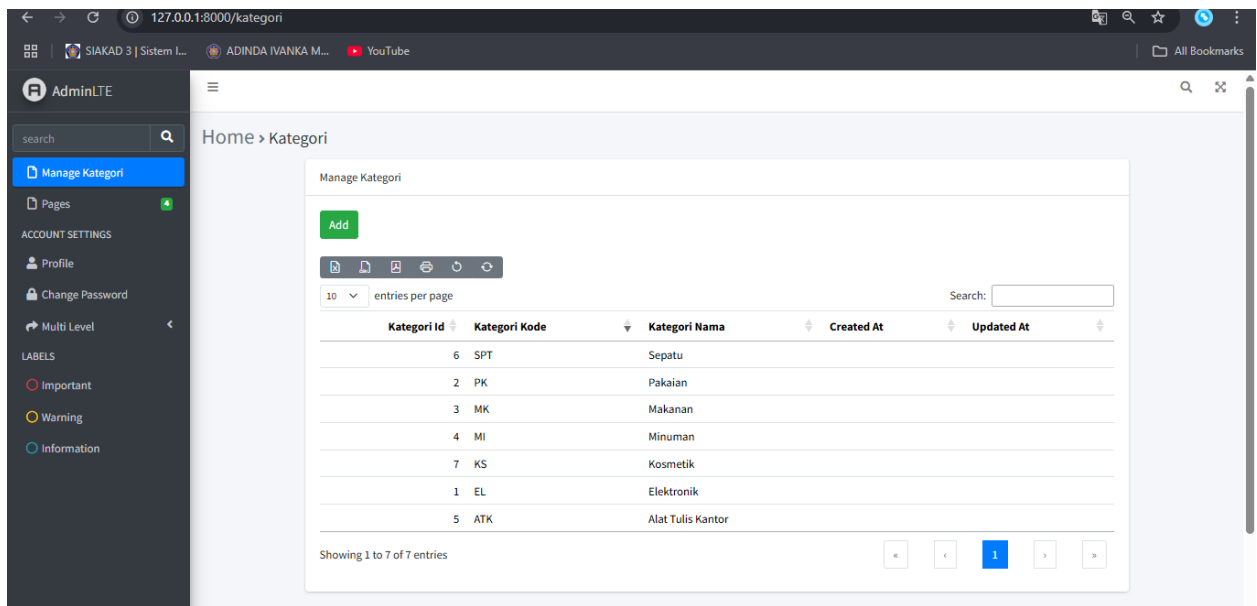
```



```
'topnav_right' => true,
],

// Sidebar items:
[
  'type' => 'sidebar-menu-search',
  'text' => 'search',
],
[
  'text' => 'blog',
  'url' => 'admin/blog',
  'can' => 'manage-blog',
],
[
  'text' => 'Manage Kategori',
  'url' => '/kategori',
  'icon' => 'far fa-fw fa-file',
],
[
  'text' => 'pages',
  'url' => 'admin/pages',
  'icon' => 'far fa-fw fa-file',
  'label' => 4,
  'label_color' => 'success',
],
['header' => 'account_settings'],
[
  'text' => 'profile',
  'url' => 'admin/settings',
  'icon' => 'fas fa-fw fa-user',
],
[
  'text' => 'change_password',
  'url' => 'admin/settings',
  'icon' => 'fas fa-fw fa-lock',
],
[
  'text' => 'multilevel',
  'icon' => 'fas fa-fw fa-share',
  'submenu' => [
    [
      'text' => 'level_one',
      'url' => '#',
    ],
    [
      'text' => 'level_one',
      'url' => '#',
    ],
  ],
  'submenu' => [
```

```
[
  [
    'text' => 'level_two',
    'url' => '#',
  ],
  [
    'text' => 'level_two',
    'url' => '#',
    'submenu' => [
      [
        'text' => 'level_three',
        'url' => '#',
      ],
      [
        'text' => 'level_three',
        'url' => '#',
      ],
    ],
  ],
],
[
  [
    'text' => 'level_one',
    'url' => '#',
  ],
],
],
['header' => 'labels'],
[
  'text' => 'important',
  'icon_color' => 'red',
  'url' => '#',
],
[
  'text' => 'warning',
  'icon_color' => 'yellow',
  'url' => '#',
],
[
  'text' => 'information',
  'icon_color' => 'cyan',
  'url' => '#',
],
],
```



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

```
public function dataTable(QueryBuilder $query): EloquentDataTable
{
    return (new EloquentDataTable($query))
        ->addColumn('action', function ($id) {
            $edit = route('kategori.edit', $id);

            return '<a href="' . $edit . '" class="btn btn-warning btn-sm">Edit</a>
                <a href="' . $delete . '" class="btn btn-danger btn-sm">Delete</a>';
        })
        ->setRowId('id');
}
```

```
public function getColumns(): array
{
    return [
        /* Column::computed('action')
            ->exportable(false)
            ->printable(false)
            ->width(60)
            ->addClass('text-center'), */

        Column::make('kategori_id'),
        Column::make('kategori_kode'),
        Column::make('kategori_nama'),
        Column::make('created_at'),
        Column::make('updated_at'),

        Column::computed('action')
            ->exportable(false)
            ->printable(false)
            ->width(140)
    ];
}
```

```
->addClass('text-center'),  
];  
}
```

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

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

4. Tambahkan action delete di datatables serta controllernya

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

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

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

127.0.0.1:8000/kategori

SIAKAD 3 | Sistem I...ADINDA IVANKA M...YouTube

All Bookmarks

AdminLTE

search

Pages

ACCOUNT SETTINGS

Profile

Change Password

Manage Kategori

Multi Level

LABELS

Important

Warning

Information

Manage Kategori

Home > Kategori

Manage Kategori

Add

10 entries per page

Search:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At	Action
6	SPT	Sepatu			<a>Edit <a>Delete
2	PK	Pakaian			<a>Edit <a>Delete
3	MK	Makanan			<a>Edit <a>Delete
4	MI	Minuman			<a>Edit <a>Delete
7	KS	Kosmetik			<a>Edit <a>Delete
1	EL	Elektronik			<a>Edit <a>Delete
5	ATK	Alat Tulis Kantor			<a>Edit <a>Delete

Showing 1 to 7 of 7 entries

<< < **1** > >>