

## JOBSHEET 05

### Blade View, Web Templating(AdminLTE), Datatables

Nama : Raffy Jamil Octavialdy

Kelas : TI - 2H

NIM : 2241720082

#### Praktikum 1 – Integrasi Laravel dengan AdminLte3

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

```
PS D:\Semester 4\Pemrograman Web Lanjut\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
Lock file operations: 2 installs, 0 updates, 0 removals
  - Locking almasaeed2010/adminlte (v3.2.0)
  - Locking jeroennoten/laravel-adminlte (v3.10.0)
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 (v3.10.0)
  - 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
```

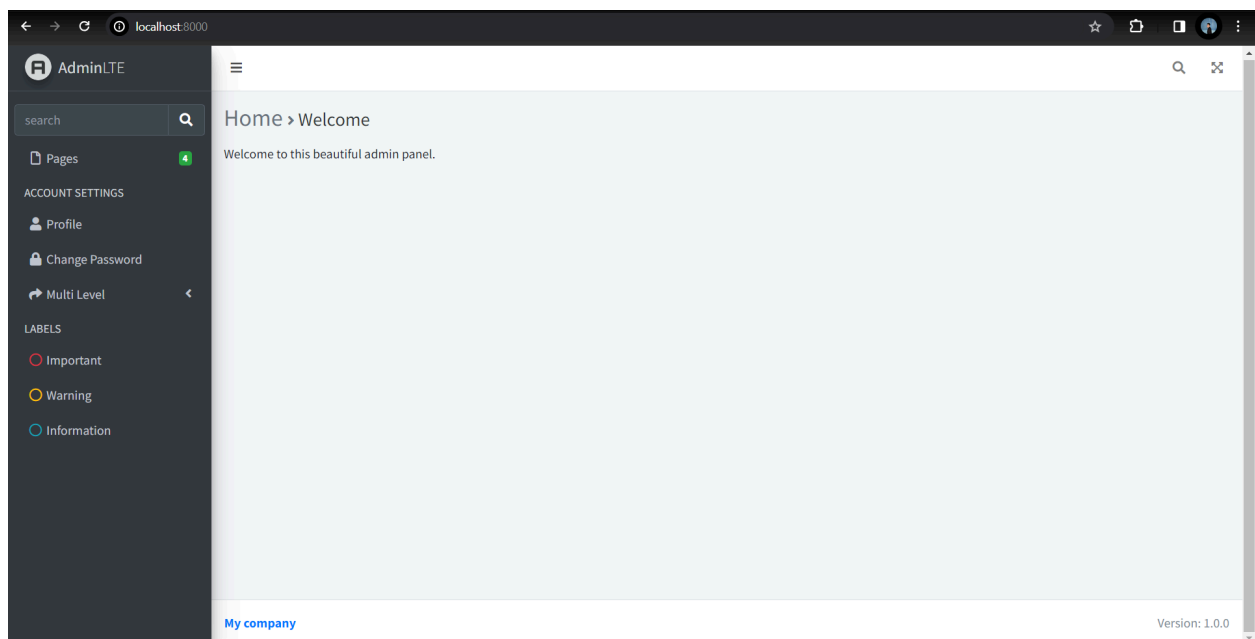
2. Melakukan instalasi requirement project di atas dengan command berikut:

```
PS D:\Semester 4\Pemrograman Web Lanjut\PWL_POS> php artisan adminlte:install
Basic assets installed successfully.
Configuration file installed successfully.
Translation files installed successfully.
The installation is complete.
```

```

"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": "^3.10",
    "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"
},

```



## Praktikum 2 – Integrasi dengan DataTables

### 1. Menginstall Laravel DataTables

```
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

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
yajra/laravel-datatables-buttons ..... DONE
yajra/laravel-datatables-editor ..... DONE
yajra/laravel-datatables-fractal ..... DONE
yajra/laravel-datatables-html ..... DONE
yajra/laravel-datatables-oracle ..... DONE
```

### 2. Menginstall nodejs

```
C:\Users\Asus>npm -v
9.6.7
```

### 3. Menginstall Laravel DataTables Vite dan sass

```
PS D:\Semester 4\Pemrograman Web Lanjut\PWL_POS> npm i laravel-datatables-vite --save-dev

added 33 packages, and audited 34 packages in 10s

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

found 0 vulnerabilities
```

```
PS D:\Semester 4\Pemrograman Web Lanjut\PWL_POS> npm install -D sass

added 15 packages, and audited 49 packages in 2s

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

found 0 vulnerabilities
```

4. Edit file resources/js/app.js

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

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

```
PS D:\Semester 4\Pemrograman Web Lanjut\PWL_POS> npm run dev

> dev
> vite

VITE v5.1.6 ready in 320 ms

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

LARAVEL v10.46.0 plugin v1.0.2

→ APP_URL: http://localhost
```

7. Selanjutnya buat datatables untuk kategori

```
PS D:\Semester 4\Pemrograman Web Lanjut\PWL_POS> php artisan datatables:make Kategori
```

```
INFO DataTable [D:\Semester 4\Pemrograman Web Lanjut\PWL_POS\app\DataTables\KategoriDataTable.php] created successfully.
```

8. Edit KategoriDataTable untuk mengatur kolom apa saja yang ingin ditampilkan

```
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.
     */
}
```

9. Ubah kategori model, sesuaikan seperti berikut

```
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:

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

11. Buat view blade index untuk kategori resources/views/kategori/index.blade.php

```
@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">
                {{ $dataTable->table() }}
            </div>
        </div>
    </div>
@endsection

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

12. Pastikan route untung kategori sudah tersedia

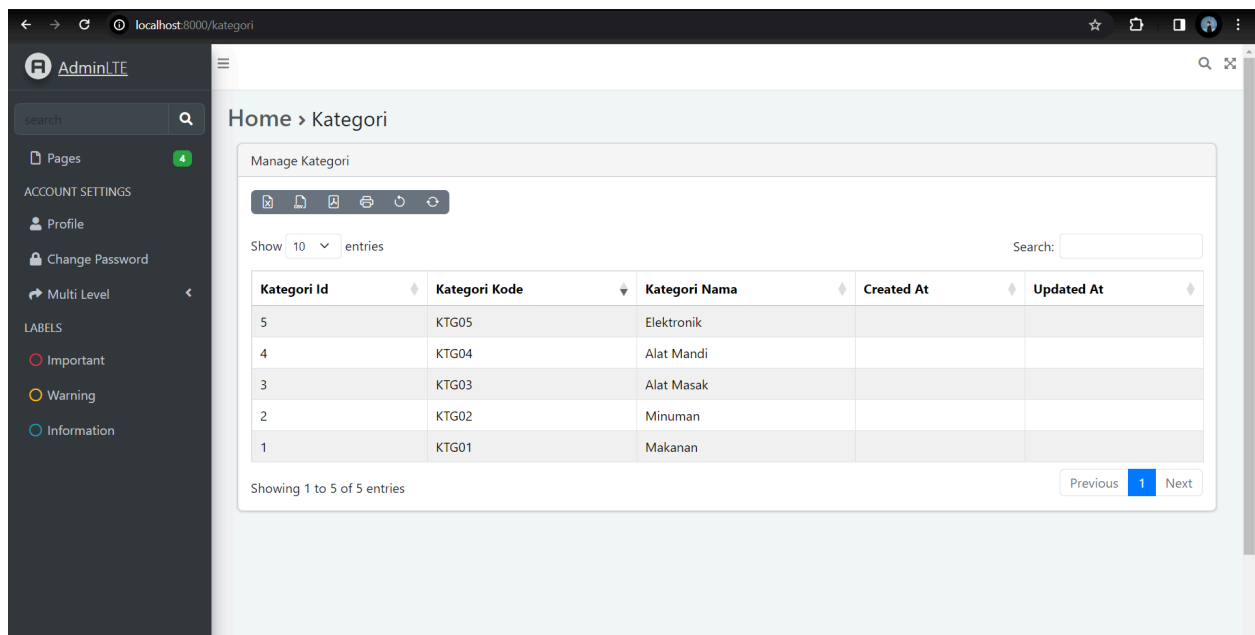
```
Route::get('/kategori', [KategoriController::class, 'index']);
```

### 13. Set ViteJs / script type defaults

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

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

### 14. Datatables sudah dapat di load di /kategori



The screenshot shows a web application interface with a sidebar on the left and a main content area. The sidebar contains a search bar, a 'Pages' link with a green badge showing '4', and sections for 'ACCOUNT SETTINGS' (Profile, Change Password, Multi Level) and 'LABELS' (Important, Warning, Information). The main content area has a breadcrumb 'Home > Kategori' and a 'Manage Kategori' section. This section includes a table with 5 entries, a search bar, and pagination controls.

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At
5	KTG05	Elektronik		
4	KTG04	Alat Mandi		
3	KTG03	Alat Masak		
2	KTG02	Minuman		
1	KTG01	Makanan		

Showing 1 to 5 of 5 entries

Previous 1 Next

### Praktikum 3 – Membuat form kemudian menyimpan data dalam database

1. Menyesuaikan routing, tambahkan dua routing berikut

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

2. Tambahkan dua function berikut dalam KategoriController

```
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

```
@section('content')
<div class="container">
<div class="card card-primary">
<div class="card-header">
<h3 class="card-title">Buat Kategori Baru</h3>
</div>

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

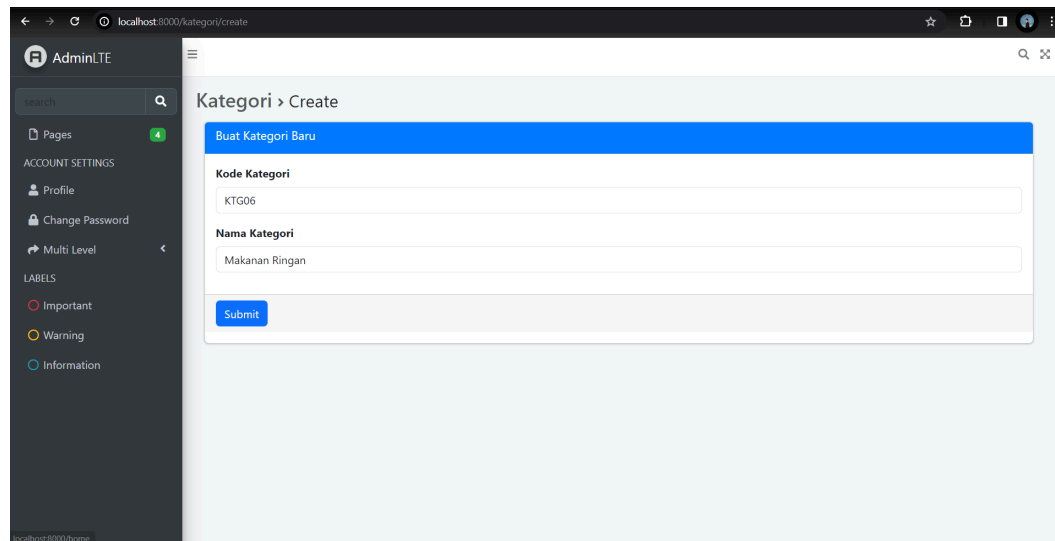
<div class="card-footer">
<button type="submit" class="btn btn-primary">Submit</button>
</div>
</form>
</div>
</div>
@endsection
```



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

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

## 5. Akses kategori/create



AdminLTE

search

Pages

ACCOUNT SETTINGS

Profile

Change Password

Multi Level

LABELS

Important

Warning

Information

localhost:8000/home

Kategori > Create

Buat Kategori Baru

Kode Kategori

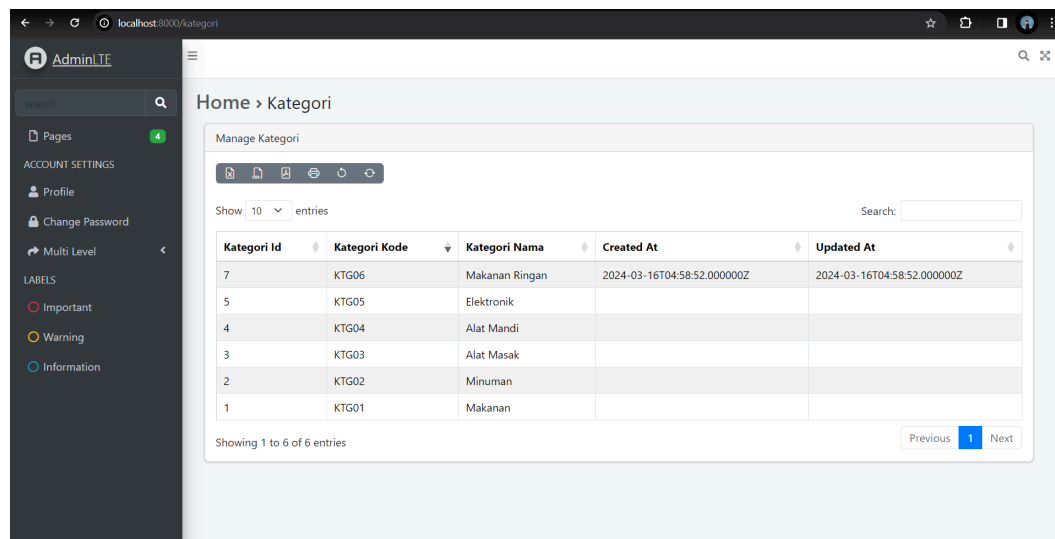
KTG06

Nama Kategori

Makanan Ringan

Submit

## 6. Halaman kategori



AdminLTE

search

Pages

ACCOUNT SETTINGS

Profile

Change Password

Multi Level

LABELS

Important

Warning

Information

localhost:8000/kategori

Home > Kategori

Manage Kategori

Show 10 entries

Search:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At
7	KTG06	Makanan Ringan	2024-03-16T04:58:52.000000Z	2024-03-16T04:58:52.000000Z
5	KTG05	Elektronik		
4	KTG04	Alat Mandi		
3	KTG03	Alat Masak		
2	KTG02	Minuman		
1	KTG01	Makanan		

Showing 1 to 6 of 6 entries

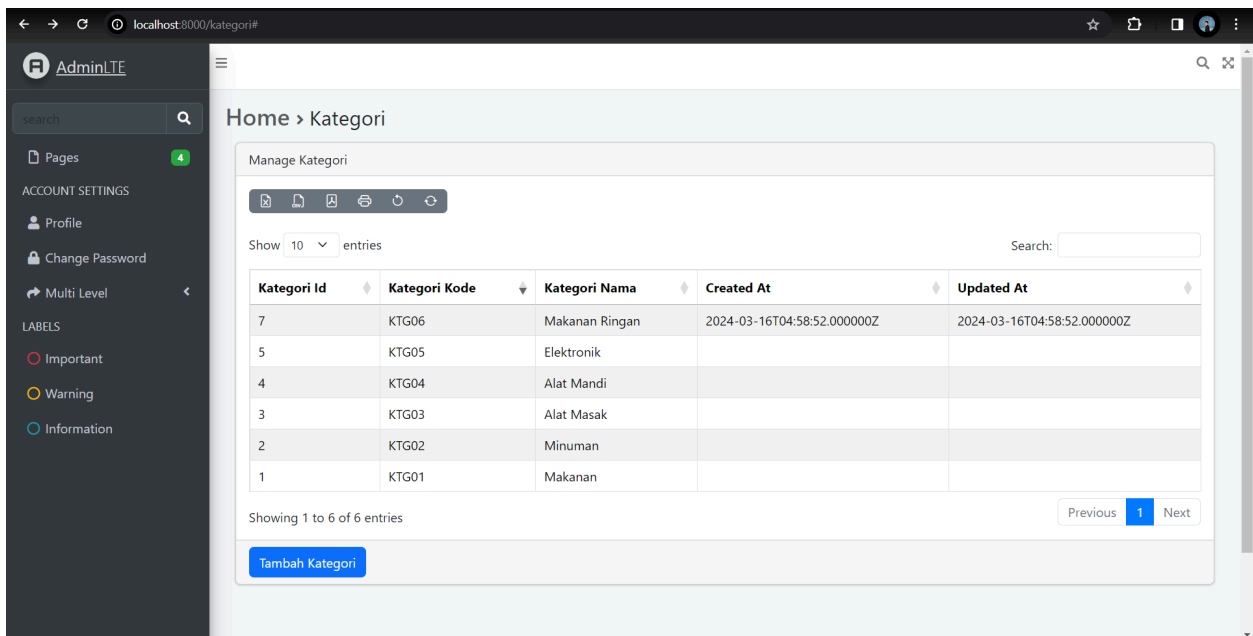
Previous 1 Next

## Tugas Praktikum

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

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

    <div class="card-footer">
      <a href="/kategori/create" class="btn btn-primary">Tambah Kategori</a>
    </div>
  </div>
</div>
@endsection
```



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

```
@section('content_top_nav_right')
{{-- Menu for manage kategori --}}
<li class="nav-item">
  <a href="/kategori" class="nav-link">
    <h6>Manage Kategori</h6>
  </a>
</li>
@stop
```



### 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 ($kategori) {
            return '<a href="' . route('kategori.edit', $kategori->kategori_id) . '" class="btn btn-primary">Edit</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::make('action')
            ->exportable(false)
            ->printable(false)
            ->width(60)
            ->addClass('text-center')
            ->title('Actions'),
    ];
}
```

```
public function edit(KategoriModel $kategori)
{
    return view('kategori.edit', compact('kategori'));
}

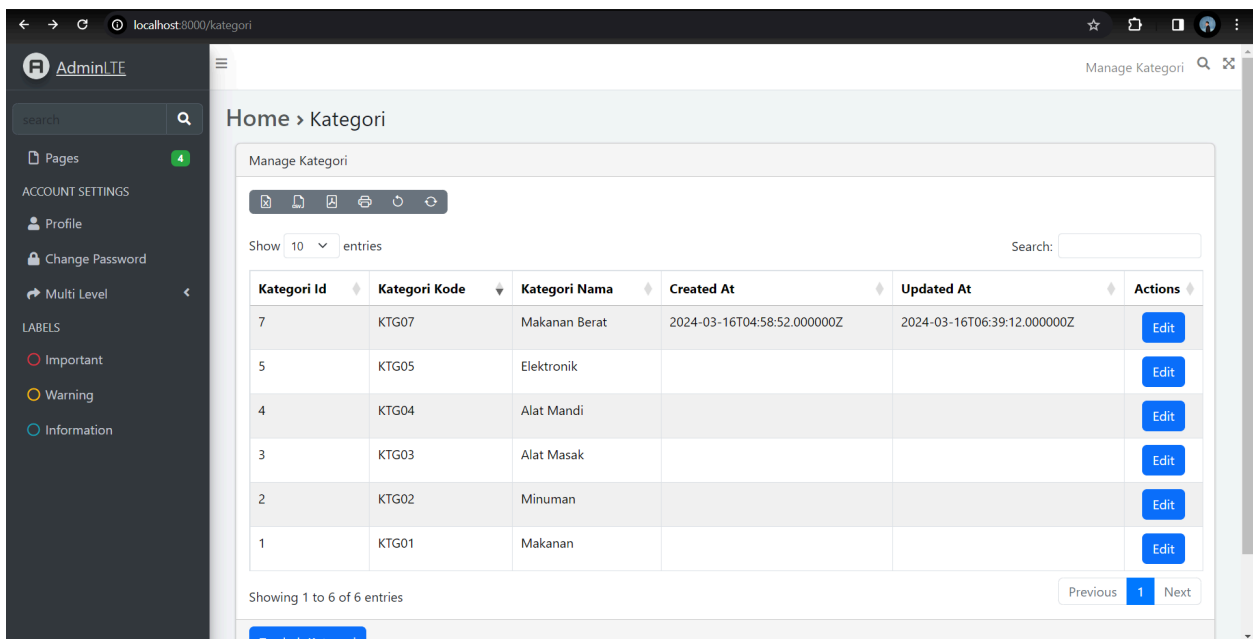
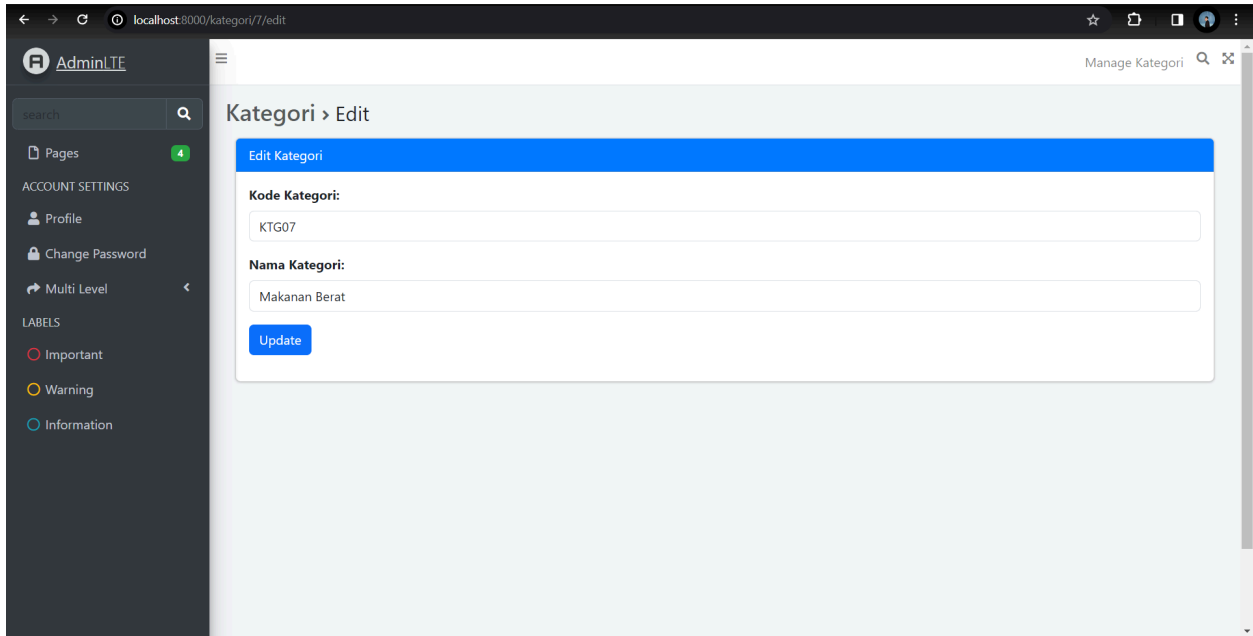
public function update($id, Request $request)
{
    $kategori = KategoriModel::find($id);

    $kategori->kategori_kode = $request->kategori_kode;
    $kategori->kategori_nama = $request->kategori_nama;

    $kategori->save();

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

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



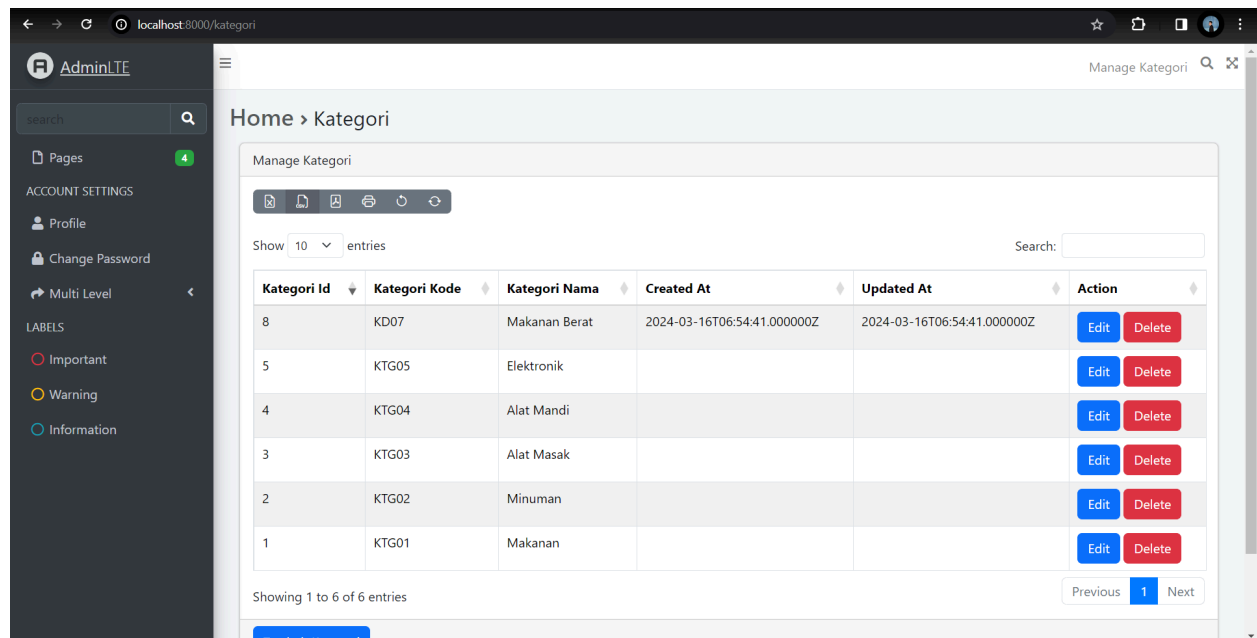
#### 4. Tambahkan action delete di datatables serta controllernya

```
public function dataTable(QueryBuilder $query): EloquentDataTable
{
    return (new EloquentDataTable($query))
        ->addColumn('action', function ($kategori) {
            return '<a href="' . route('kategori.edit', $kategori->kategori_id) . '" class="btn btn-primary">Edit</a>'
                . '<form action="' . route('kategori.destroy', $kategori->kategori_id) . '" method="POST" style="display: inline;">'
                . '<input type="hidden" name="csrf_field()" value="' . csrf_field() . '">'
                . '<input type="hidden" name="method" value="DELETE">'
                . '<button type="submit" class="btn btn-danger">Delete</button>'
                . '</form>';
        })
        ->setRowId('id');
}
```

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

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

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



The screenshot shows a web application interface for managing categories. The main content area displays a table titled 'Manage Kategori' with the following data:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At	Action
8	KD07	Makanan Berat	2024-03-16T06:54:41.000000Z	2024-03-16T06:54:41.000000Z	<a href="#">Edit</a> <a href="#">Delete</a>
5	KTG05	Elektronik			<a href="#">Edit</a> <a href="#">Delete</a>
4	KTG04	Alat Mandi			<a href="#">Edit</a> <a href="#">Delete</a>
3	KTG03	Alat Masak			<a href="#">Edit</a> <a href="#">Delete</a>
2	KTG02	Minuman			<a href="#">Edit</a> <a href="#">Delete</a>
1	KTG01	Makanan			<a href="#">Edit</a> <a href="#">Delete</a>

The interface includes a sidebar with navigation links (AdminLTE, Pages, ACCOUNT SETTINGS, Profile, Change Password, Multi Level, LABELS) and a search bar. The table shows 6 entries, and the pagination indicates 'Showing 1 to 6 of 6 entries'.

AdminLTE

search

Pages 4

ACCOUNT SETTINGS

Profile

Change Password

Multi Level

LABELS

Important

Warning

Information

Manage Kategori

Home > Kategori

Manage Kategori

Show 10 entries

Search:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At	Action
5	KTG05	Elektronik			<div>EditDelete</div>
4	KTG04	Alat Mandi			<div>EditDelete</div>
3	KTG03	Alat Masak			<div>EditDelete</div>
2	KTG02	Minuman			<div>EditDelete</div>
1	KTG01	Makanan			<div>EditDelete</div>

Showing 1 to 5 of 5 entries

Previous1Next

Tambah Kategori