

LAPORAN JOBSHEET 5

Blade View, Web Templating(AdminLTE), Datatables

MATA KULIAH PEMROGRAMAN WEB LANJUT

Dosen Pengampu : Dimas Wahyu Wibowo, S.T., M.T.



Disusun oleh :

Dahniar Davina SIB-2A / 2341760023

JURUSAN TEKNOLOGI INFORMASI

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

POLITEKNIK NEGERI MALANG

2025

A. PRAKTIKUM 1 : Integrasi Laravel dengan AdminLTE3

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

```
Windows@DESKTOP-PE7RQP7 MINGW64 /c:/laragon/www/PWL_2025/coba (
master)
• $ composer require jeroennoten/laravel-adminlte
./composer.json has been updated
Running composer update jeroennoten/laravel-adminlte
```

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

```
master)
• $ php artisan adminlte:install
AdminLTE assets files published successfully
Configuration file published successfully
Translation files published successfully
The installation is complete.
```

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

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

{{-- Extend and customize the browser title --}}
@section('title')
    {{ config('adminlte.title') }}
@endsection

@section('subtitle') | @yield('subtitle') @endif
@stop

{{-- Extend and customize the page content header --}}
@section('content_header')
    @yield('content_header_title')
    <h1 class="text-muted">
        @yield('content_header_title')
    </h1>
    <small class="far fa-fw fa-angle-right text-muted"></small>
    @yield('content_header_subtitle')
</h1>
@elseif
@stop
```

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

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

{{-- Customize layout sections --}}

@section('subtitle', 'Welcome')
@section('content_header_title', 'Home')
@section('content_header_subtitle', 'Welcome')

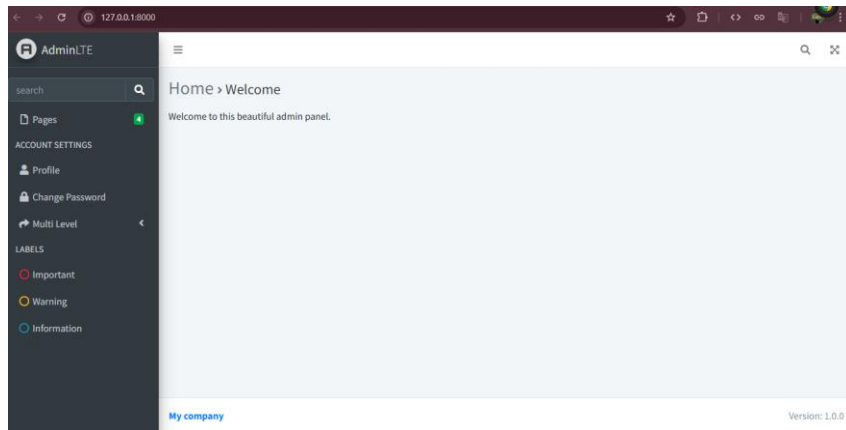
{{-- Content body: main page content --}}

@section('content_body')
    <p>Welcome to this beautiful admin panel.</p>
@stop

{{-- Push extra CSS --}}

@push('css')
    {{-- Add here extra stylesheets --}}
    {{-- <link rel="stylesheet" href="/css/admin_custom.c
@endpush
```

5. Kembali ke browser, menuju ke halaman awal



B. PRAKTIKUM 2 : Integrasi dengan DataTables

1. Install Laravel DataTables

```
Windows@DESKTOP-PE7RQP7 MINGW64 /c:/laragon/www/PWL_2025/coba (master)
$ composer require laravel/ui --dev
./composer.json has been updated
```

```
Windows@DESKTOP-PE7RQP7 MINGW64 /c:/laragon/www/PWL_2025/coba (master)
$ composer require yajra/laravel-datatables:^10.0
./composer.json has been updated
```

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

```
C:\laragon\www\PWL_2025\Minggu5\Jobsheet5>npm -v
10.2.4

C:\laragon\www\PWL_2025\Minggu5\Jobsheet5>node -v
v20.11.1
```

3. Install Laravel DataTables Vite dan sass

```
C:\laragon\www\PWL_2025\Minggu5\Jobsheet5>npm i laravel-datatables-vite --save-dev
up to date, audited 63 packages in 4s
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
Run `npm audit` for details.
```

```
C:\laragon\www\PWL_2025\Minggu5\Jobsheet5>npm install -D sass
up to date, audited 63 packages in 2s
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
Run `npm audit` for details.
```

4. Edit file resources/js/app.js

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

5. Buatlah file resources/sass/app.scss

```
resources > sass > app.scss
1 // Fonts
2 @import url('https://fonts.bunny.net/css?family=Nunito');
3
4 // Bootstrap
5 @import 'bootstrap/scss/bootstrap';
6
7 // DataTables
8 @import 'bootstrap-icons/font/bootstrap-icons.css';
9 @import 'datatables.net-bs5/css/dataTables.bootstrap5.min';
10 @import 'datatables.net-buttons-bs5/css/buttons.bootstrap5.min';
11 @import 'datatables.net-select-bs5/css/select.bootstrap5.min';
```

6. Jalankan dengan npm run dev

```
C:\laragon\www\PWL_2025\Minggu5\Jobsheet5>npm run dev
> dev
> vite

VITE v5.4.14 ready in 3299 ms
  Local: http://localhost:5173/
  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

```
Windows@DESKTOP-PE7RQP7 MINGW64 /c:/laragon/www/PWL_2025/coba
ster)
$ php artisan datatables:make Kategori

INFO Data Table [C:\laragon\www\PWL_2025\coba\app\DataTables\KategoriDataTable.php] created successfully.
```

8. Kita edit KategoriDatable untuk mengatur kolom apasaja yang ingin ditampilkan
Sesuaikan dengan kode program Berikut

```
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\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');
    }
}

```

9. Ubah kategori model, sesuaikan seperti Berikut

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
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 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

```

@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 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') }}
@endsection

@section('subtitle') | @yield('subtitle') @endif
@stop

{{-- Extend and customize the page content header --}}
@section('content_header')
    @yield('content_header_title')
    <h1 class="text-muted">
        @yield('content_header_title')
    </h1>
    <small class="far fa-fw fa-angle-right text-muted"></small>
    @yield('content_header_subtitle')
</h1>
@elseif
@stop

{{-- Rename section content to content_body --}}
@section('content')
    @yield('content_body')

```

14. Menset ViteJs / script type defaults

```

Providers > AppServiceProvider.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

		kategori_id	kategori_kode	kategori_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	KTG001	Elektronik	2025-03-05 09:03:05	2025-03-19 09:03:05
<input type="checkbox"/>	Edit Copy Delete	2	KTG002	Pakaian	2025-03-05 09:03:42	2025-03-19 09:03:42
<input type="checkbox"/>	Edit Copy Delete	3	KTG003	Makanan	2025-03-05 09:04:01	2025-03-19 09:04:01
<input type="checkbox"/>	Edit Copy Delete	4	KTG004	Minuman	2025-03-05 09:04:21	2025-03-19 09:04:21
<input type="checkbox"/>	Edit Copy Delete	5	KTG005	Obat-obatan	2025-03-05 09:04:38	2025-03-19 09:04:38
<input type="checkbox"/>	Edit Copy Delete	6	KTG006	Camilan	2025-03-17 09:02:14	2025-03-19 09:02:14
<input type="checkbox"/>	Edit Copy Delete	7	KTG007	Minuman Kaleng	2025-03-17 09:05:13	2025-03-19 09:05:13

16. Datatables sudah dapat di load di /kategori

127.0.0.1:8000/kategori

Home > Kategori

AdminLTE

search

PAGES

ACCOUNT SETTINGS

Profile

Change Password

Multi Level

LABELS

Important

Warning

Information

Home > Kategori

Manage Kategori

10 entries per page

Search:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At
7	KTG007	Minuman Kaleng	2025-03-17T09:05:13.000000Z	2025-03-19T09:05:13.000000Z
6	KTG006	Camilan	2025-03-17T09:02:14.000000Z	2025-03-19T09:02:14.000000Z
5	KTG005	Obat-obatan		
4	KTG004	Minuman		
3	KTG003	Makanan		
2	KTG002	Pakaian		
1	KTG001	Elektronik		

Showing 1 to 7 of 7 entries

C. PRAKTIKUM 3 : Membuat Form dan disimpan dalam Database

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

```
}  
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:

```
pp > Http > Middleware > VerifyCsrfToken.php > ...  
1  <?php  
2  
3  namespace App\Http\Middleware;  
4  
5  use Illuminate\Foundation\Http\Middleware\VerifyCsrfToken;  
6  
7  class VerifyCsrfToken extends Middleware  
8  {  
9      /**  
10         * The URIs that should be excluded from CSRF verification.  
11         *  
12         * @var array<int, string>  
13         */  
14         protected $except = [  
15             '/kategori'  
16         ];  
17     }  
18
```


5. Akses kategori/create

Kategori > Create

Buat kategori baru

Kode Kategori

Masukkan kode kategori

Nama Kategori

Masukkan nama kategori

Submit

6. Halaman kategori

Home > Kategori

Manage Kategori

10 entries per page

Search:

Kategori Id	Kategori Kode	Kategori Nama	Created At	Updated At
8	KTG008	Sepatu	2025-03-19T02:34:00.000000Z	2025-03-19T02:34:00.000000Z
7	KTG007	Minuman Kaleng	2025-03-17T09:05:13.000000Z	2025-03-19T09:05:13.000000Z
6	KTG006	Camilan	2025-03-17T09:02:14.000000Z	2025-03-19T09:02:14.000000Z
5	KTG005	Obat-obatan		
4	KTG004	Minuman		
3	KTG003	Makanan		
2	KTG002	Pakaian		
1	KTG001	Elektronik		

Showing 1 to 8 of 8 entries

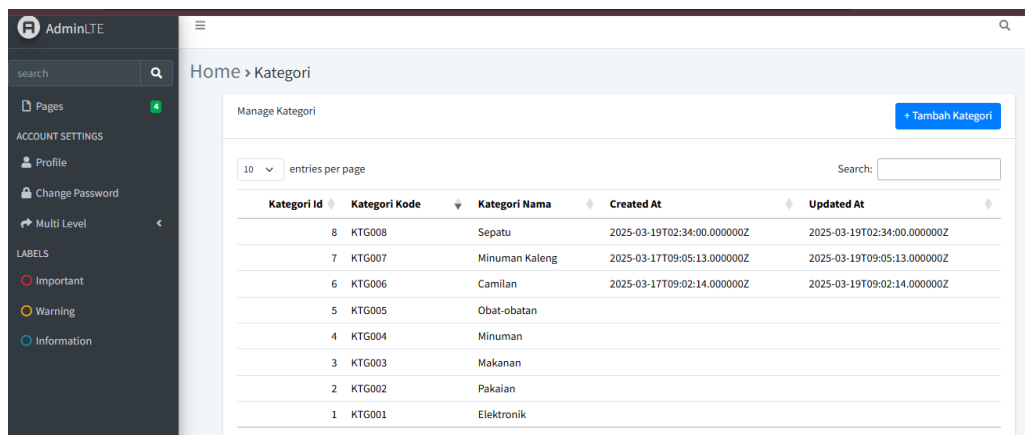
D. TUGAS PRAKTIKUM

1. Tambahkan button Add di halam 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-header">Manage Kategori
      <a href="{{ route('kategori.create') }}" class="btn btn-primary">Add</a>
    </div>
    <div class="card-body">
      {{ $dataTable->table() }}
    </div>
  </div>
</div>
</section>
```

Tambah kode program pada
Kategori/index.blade.php

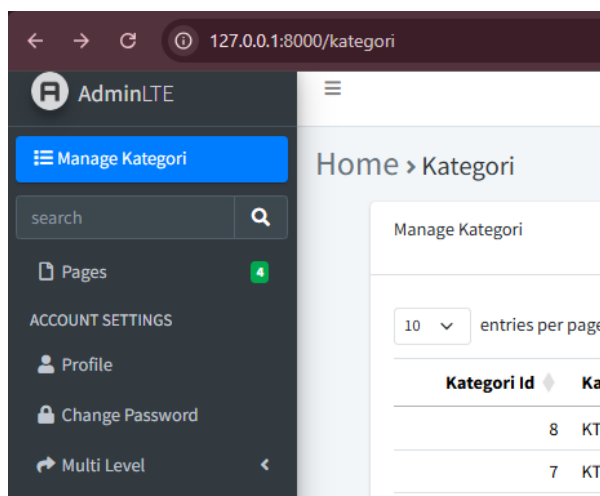
```
Route::get('/kategori/create', [KategoriController::class, 'create'])->name('kategori.create');
```



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

```
'menu' => [
    // Navbar items:
    [
        'text' => 'Manage Kategori',
        'url' => 'kategori',
        'icon' => 'fas fa-list',
    ],
]
```

Tambah kode program pada config/adminlte.php



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

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

```
Column::computed('action')
->exportable(false)
->printable(false)
->width(100)
->addClass('text-center'),
```

Tambah 2 kode program pada app/DataTables/KategoriDataTable.php

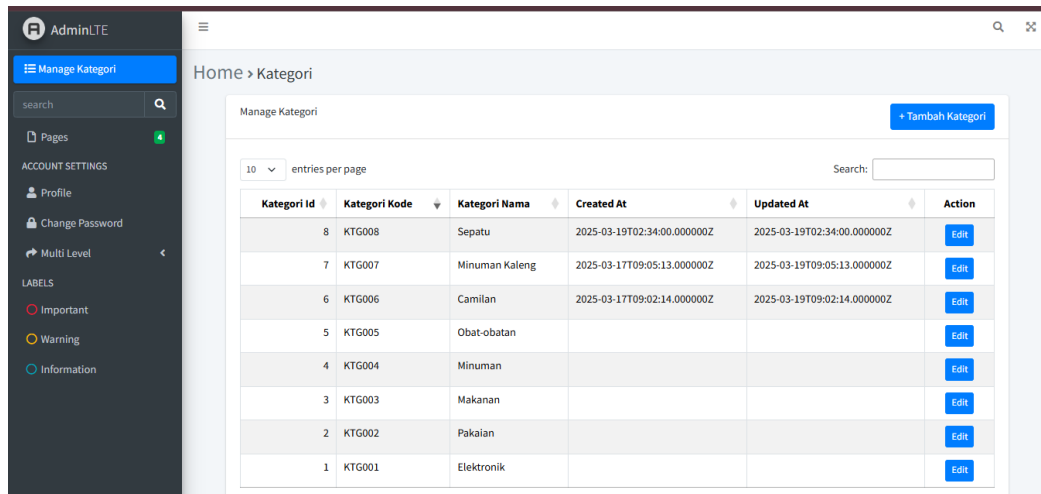
Tambah function edit pada KategoriController.php

Buat file edit pada kategori/edit.blade.php

Tambah 2 kode program
pada index.blade.php

Tambah route edit pada web.php

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

```
return (new EloquentDataTable($query))
    // ->addColumn('action', 'kategori.action')
    ->addColumn('action', function($id){
        $edit = route('kategori.edit', $id);
        $delete = route('kategori.delete', $id);
        return '
            <a href="'.$edit.'" class="btn btn-primary btn-sm">Edit</a>
            <a href="'.$delete.'" class="btn btn-danger btn-sm" onclick="return confirm('\Apakah Anda yakin ingin menghapus kategori ini?')">Delete</a>
        ';
    })
    ->setRowId('id');
```

Tambah kode delete pada kategori/dataTable.php

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

Tambah function delete pada KategoriController.php

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

Tambah route delete pada web.php

