

```

<?php
namespace App\Http\Controllers;

use App\Models\Barang;
use App\Models\Kategori;
use App\Models\Stock;
use Illuminate\Http\Request;
use Yajra\DataTables\Facades\DataTables;

class BarangController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index(Request $request)
    {
        $kategoris = Kategori::all();
        if ($request->ajax()) {
            $query_data = Barang::query();

            if ($request->has('sSearch') &&
                $request->sSearch != "") {
                $search_value = '%' . $request->sSearch . '%';

                $query_data->where(function
                ($query) use ($search_value) {
                    $query->where('kode_buku', 'like',
                    $search_value)
                    ->orWhere('judul', 'like',
                    $search_value)
                    ->orWhere('kelas', 'like',
                    $search_value)

```

```

                    ->orWhere('penerbit', 'like',
                    $search_value)
                    ->orWhere('tahun_terbit', 'like',
                    $search_value);
                });
            }

            $data = $query_data->orderBy('id', 'asc')->get();

            return DataTables::of($data)
                ->addIndexColumn()
                ->addColumn('action', function ($row)
                {
                    $btn = '
                        <form action="" .
                        route('barangs.destroy', $row->id) . ""
                        method="POST" style="display:inline;">
                            <a class="btn btn-info" href="" .
                            route('barangs.show', $row->id) . "">
                                <i class="fas fa-eye"></i>
                            </a>
                            <button type="button" class="btn
                            btn-primary edit-btn" data-id="" . $row->id . ""
                            data-toggle="modal" data-target="#editModal">
                                <i class="fas fa-edit"></i>
                            </button>
                            ' . csrf_field() . '
                            ' . method_field('DELETE') . '
                            <button type="submit"
                            class="btn btn-danger" onclick="return
                            confirm(\'Are you sure you want to delete this
                            item?\')">
                                <i class="fas fa-trash"></i>

```

```

        </button>

    </form>';

    return $btn;

    })

    ->addColumn('jenjang', function
($row) {

        return $row->kategori->jenjang;

    })

    ->addColumn('kelas', function ($row)
{

        return $row->kategori->kelas;

    })

    ->rawColumns(['action'])

    ->make(true);

}

    return view('barangs.index',
compact('kategoris'));

}

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    $kategoris = Kategori::all();

    return view('barangs.create',
compact('kategoris'));

```

```

    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $request->validate([

            'kode_buku' => 'required',

            'judul' => 'required',

            'penerbit' => 'required',

            'tahun_terbit' => 'required',

            'id_kategori' => 'required',

        ]);

        Barang::create($request->all());

        return redirect()->route('barangs.index')-
>with('success', 'Barang created successfully');

    }

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)

```

```

{
    $barang = Barang::findOrFail($id);

    return view('barangs.show',
compact('barang'));
}

public function edit($id){
    $barang = Barang::find($id);
    return response()->json($barang);
}

public function update(Request $request, $id)
{
    $request->validate([
        'kode_buku' => 'required',
        'judul' => 'required',
        'penerbit' => 'required',
        'tahun_terbit' => 'required',
        'id_kategori' => 'required',
    ]);

    $barang = Barang::findOrFail($id);
    $barang->update($request->all());

    return redirect()->route('barangs.index')-
>with('success', 'Barang updated successfully');
}

/**

```

```

* Remove the specified resource from
storage.
*
* @param int $id
* @return \Illuminate\Http\Response
*/

public function destroy($id)
{
    $barang = Barang::findOrFail($id);

    $stock =
Stock::where('id_barang','=',$barang->id)-
>delete();

    $barang->delete();

    return redirect()->route('barangs.index')-
>with('success', 'Gudang deleted successfully');

    // if ($barang->stocks) {
        // // Delete related records in the 'stocks'
table first
        // foreach ($barang->stocks as $stock) {
            // $stock->delete();
        // }

        // $barang->delete();

        // return redirect()-
>route('barangs.index')->with('success', 'Barang
deleted successfully');

    // }
}

}

<?php

```

```

namespace App\Http\Controllers;

use App\Models\Gudang;
use Illuminate\Http\Request;
use Yajra\DataTables\Facades\DataTables;

class GudangController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index(Request $request)
    {
        if ($request->ajax()) {
            $query_data = Gudang::query();

            if ($request->has('sSearch') &&
                $request->sSearch != "") {
                $search_value = '%' . $request->sSearch . '%';

                $query_data->where(function ($query) use ($search_value) {
                    $query->where('kode_perwakilan_regional', 'like', $search_value)
                        ->orWhere('lokasi_gudang', 'like', $search_value)
                        ->orWhere('no_gudang', 'like', $search_value);
                });
            }

            $data = $query_data->orderBy('id', 'asc')->get();

            return DataTables::of($data)
                ->addIndexColumn()
                ->addColumn('action', function ($row) {
                    $btn = '
                        <form action="" .
                        route('gudangs.destroy', $row->id) . ""><i
                        method="POST">

                        <a class="btn btn-info" href="" .
                        route('gudangs.show', $row->id) . ""><i
                        class="fas fa-eye"></i></a>

                        <a class="btn btn-primary"
                        href="" . route('gudangs.edit', $row->id) . ""><i
                        class="fas fa-edit"></i></a>

                        ' . csrf_field() .
                        method_field('DELETE') . '

                        <button type="submit"
                        class="btn btn-danger"><i class="fas fa-trash"></i></button>

                    </form>;

                    return $btn;
                })
                ->rawColumns(['action'])
                ->make(true);
        }

        return view('gudangs.index');
    }
}

```

```

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    return view('gudangs.create');
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $request->validate([
        'kode_perwakilan_regional' => 'required',
        'lokasi_gudang' => 'required',
        'no_gudang' => 'required',
    ]);

    Gudang::create($request->all());

    return redirect()->route('gudangs.index')-
    >with('success', 'Gudang created successfully');
}

```

```

/**
 * Display the specified resource.
 *
 * @param \App\Models\Gudang $gudang
 * @return \Illuminate\Http\Response
 */
public function show(Gudang $gudang)
{
    return view('gudangs.show',
compact('gudang'));
}

/**
 * Show the form for editing the specified
resource.
 *
 * @param \App\Models\Gudang $gudang
 * @return \Illuminate\Http\Response
 */
public function edit(Gudang $gudang)
{
    return view('gudangs.edit',
compact('gudang'));
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\Models\Gudang $gudang

```

```

* @return \Illuminate\Http\Response
*/

public function update(Request $request,
Gudang $gudang)
{
    $request->validate([
        'kode_perwakilan_regional' => 'required',
        'lokasi_gudang' => 'required',
        'no_gudang' => 'required',
    ]);

    $gudang->update($request->all());

    return redirect()->route('gudangs.index')-
>with('success', 'Gudang updated successfully');
}

/**
 * Remove the specified resource from
storage.
 *
 * @param \App\Models\Gudang $gudang
 * @return \Illuminate\Http\Response
 */
public function destroy(Gudang $gudang)
{
    $gudang->delete();

    return redirect()->route('gudangs.index')-
>with('success', 'Gudang deleted successfully');
}
}

```

```

<?php

namespace App\Http\Controllers;

use App\Models\Barang;
use App\Models\Kategori;
use Illuminate\Http\Request;
use Yajra\DataTables\Facades\DataTables;

class KategoriController extends Controller
{
    public function __construct()
    {
        $this->middleware('permission:kategori-
list|kategori-create|kategori-edit|kategori-delete',
['only' => ['index', 'store']]);

        $this->middleware('permission:kategori-
create', ['only' => ['create', 'store']]);

        $this->middleware('permission:kategori-
edit', ['only' => ['edit', 'update']]);

        $this->middleware('permission:kategori-
delete', ['only' => ['destroy']]);
    }

    public function index(Request $request)
    {
        if ($request->ajax()) {

            $query_data = Kategori::select('*');

            if ($request->has('sSearch') &&
$request->sSearch != "") {

                $search_value = '%' . $request-
>sSearch . '%';

```

```

        $query_data->where(function
($query) use ($search_value) {

            $query->where('jenjang', 'like',
$search_value)

            ->orWhere('kelas', 'like',
$search_value);

        });
    }

    $data = $query_data->orderBy('id', 'asc')-
>get();

    return Datatables::of($data)
        ->addIndexColumn()
        ->addColumn('action', function ($row)
{

    $btn = '

        <form action="" .
route('kategoris.destroy', $row->id) . ""
method="POST">

        <a class="btn btn-info" href="" .
route('kategoris.show', $row->id) . "">Show</a>

        <a class="btn btn-primary"
href="" . route('kategoris.edit', $row->id) .
"">Edit</a>

        ' . csrf_field() .
method_field('DELETE') . '

        <button type="submit"
class="btn btn-danger">Delete</button>

    </form>;

    return $btn;

})

->rawColumns(['action'])

->make(true);

```

```

    }

    return view('kategoris.index');
}

/**,
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{

    return view('kategoris.create');
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{

    $request->validate([

        'jenjang' => 'required',
        'kelas' => 'required',

    ]);

    $input = $request->all();

```

```

$input["created_by"] = "1";
$input["updated_by"] = "1";
$input["deleted_by"] = "1";

Kategori::create($input);
return redirect()->route('kategoris.index')-
>with('success', 'Kategori created successfully');
}

/**
 * Display the specified resource.
 *
 * @param \App\Models\Kategori $kategori
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    $kategori = Kategori::findOrFail($id);

    return view('kategoris.show',
compact('kategori'));
}

/**
 * Show the form for editing the specified
resource.
 *
 * @param \App\Models\Kategori $kategori
 * @return \Illuminate\Http\Response
 */
public function edit(Kategori $kategori)
{

```

```

        return view('kategoris.edit',
compact('kategori'));
    }

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\Models\Kategori $kategori
 * @return \Illuminate\Http\Response
 */
public function update(Request $request,
Kategori $kategori)
{
    $request->validate([
        'jenjang' => 'required',
        'kelas' => 'required',
    ]);

    $kategori->update($request->all());

    return redirect()->route('kategoris.index')-
>with('success', 'Kategori updated successfully');
}

/**
 * Remove the specified resource from
storage.
 *
 * @param \App\Models\Kategori $kategori
 * @return \Illuminate\Http\Response

```



```

*/

public function destroy(Kategori $kategori)
{
    $kategori->delete();

    return redirect()->route('kategoris.index')-
    >with('success', 'Kategori deleted successfully');
}
}

<?php

namespace App\Http\Controllers;

use App\Models\Petugas;
use Illuminate\Http\Request;
use Yajra\DataTables\Facades\DataTables;

class PetugasController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index_old()
    {
        $petugas = Petugas::latest()->paginate(5);

        return view('petugas.index',
        compact('petugas'))
        ->with('i', (request()->input('page', 1) - 1)
        * 5);
    }

    public function index(Request $request)

```

```

{
    if ($request->ajax()) {
        $query_data = Petugas::query();

        if ($request->has('sSearch') &&
        $request->sSearch != "") {
            $search_value = '%' . $request-
            >sSearch . '%';

            $query_data->where(function
            ($query) use ($search_value) {
                $query->where('nama', 'like',
                $search_value)
                ->orWhere('no_hp', 'like',
                $search_value)
                ->orWhere('alamat', 'like',
                $search_value)
                ->orWhere('jenis_kelamin', 'like',
                $search_value)
                ->orWhere('foto', 'like',
                $search_value);
            });
        }

        $data = $query_data->orderBy('nama',
        'asc')->get();

        return DataTables::of($data)
        ->addIndexColumn()
        ->addColumn('action', function ($row)
        {
            $btn = '

            <form action="" .
            route('petugas.destroy', $row->id) . ""
            method="POST">

```

```

        <a class="btn btn-info" href="" .
route('petugas.show', $row->id) . "">Show</a>

        <a class="btn btn-primary"
href="" . route('petugas.edit', $row->id) .
"">Edit</a>

        ' . csrf_field() .
method_field('DELETE') . '

        <button type="submit"
class="btn btn-danger">Delete</button>

    </form>;

    return $btn;

})

->rawColumns(['action'])

->make(true);

}

return view('petugas.index');
}

/**
 * Show the form for creating a new resource.
 */
public function create()
{
    return view('petugas.create');
}

/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{

```

```

$request->validate([

    'nama' => 'required',

    'no_hp' => 'required',

    'alamat' => 'required',

    'jenis_kelamin' => 'required',

    'foto' =>

'required|image|mimes:jpeg,png,jpg,gif,svg|max:
2048',

]);

$input = $request->all();

if ($foto = $request->file('foto')) {

    $destinationPath = 'images/';

    $profileImage = date('YmdHis') . "." .
$foto->getClientOriginalExtension();

    $foto->move($destinationPath,
$profileImage);

    $input['image'] = "$profileImage";

} else {

    unset($input['image']);

}

$input["created_by"] = "1";
$input["update_by"] = "1";
$input["deleted_by"] = "1";

Petugas::create($input);

return redirect()->route('petugas.index')-
>with('success', 'Petugas created successfully');

}

/**

```

```

    * Display the specified resource.
    */

    public function show(Petugas $petugas)
    {
        return view('petugas.show',
compact('petugas'));
    }

    /**
     * Show the form for editing the specified
     resource.
     */

    public function edit(Petugas $petugas)
    {
        return view('petugas.edit',
compact('petugas'));
    }

    /**
     * Update the specified resource in storage.
     */

    public function update(Request $request,
Petugas $petugas)
    {
        $request->validate([
            'nama' => 'required',
            'no_hp' => 'required',
            'alamat' => 'required',
            'jenis_kelamin' => 'required',
            // 'foto' =>
'required|image|mimes:jpeg,png,jpg,gif,svg|max:
2048',

```

```

]);

    $input = $request->all();

    if ($foto = $request->file('foto')) {
        $destinationPath = 'images/';

        $profileImage = date('YmdHis') . "." .
$foto->getClientOriginalExtension();

        $foto->move($destinationPath,
$profileImage);

        $input['image'] = "$profileImage";
    } else {
        unset($input['image']);
    }

    $petugas->update($input);

    return redirect()->route('petugas.index')-
>with('success', 'Petugas updated successfully');
}

    /**
     * Remove the specified resource from
     storage.
     */

    public function destroy(Petugas $petugas)
    {
        $petugas->delete();

        return redirect()->route('petugas.index')-
>with('success', 'Petugas deleted successfully');
    }
}

```

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Barang;
```

```
use App\Models\Gudang;
```

```
use App\Models\Kategori;
```

```
use App\Models\Rak;
```

```
use Illuminate\Http\Request;
```

```
use Yajra\DataTables\Facades\DataTables;
```

```
class RakController extends Controller
```

```
{
```

```
    public function index(Request $request)
```

```
    {
```

```
        if ($request->ajax()) {
```

```
            $query_data = Rak::query();
```

```
            if ($request->has('sSearch') &&
                $request->sSearch != "") {
```

```
                $search_value = '%' . $request-
                >sSearch . '%';
```

```
                $query_data->where('no_rak', 'like',
                $search_value);
```

```
            }
```

```
            $data = $query_data->orderBy('id', 'asc')-
            >get();
```

```
        return DataTables::of($data)
```

```
        ->addIndexColumn()
```

```
        ->addColumn('action', function ($row)
```

```
        {
```

```
            $btn = '
```

```
                <form action="" .
```

```
                route('raks.destroy', $row->id) . ""
                method="POST">
```

```
                <a class="btn btn-info" href="" .
                route('raks.show', $row->id) . ""><i class="fas
                fa-eye"></i></a>
```

```
                <button type="button" class="btn
                btn-primary edit-btn" data-id="" . $row->id . ""
                data-toggle="modal" data-target="#editModal">
```

```
                <i class="fas fa-edit"></i>
```

```
            </button>
```

```
            ' . csrf_field() .
```

```
            method_field('DELETE') . '
```

```
                <button type="submit"
                class="btn btn-danger"><i class="fas fa-
                trash"></i></button>
```

```
            </form>;
```

```
            return $btn;
```

```
        })->addColumn('lokasi_gudang',
        function ($row) {
```

```
            return $row->gudang-
            >lokasi_gudang;
```

```
        })
```

```
        ->rawColumns(['action'])
```

```
        ->make(true);
```

```
    }
```

```
    return view('raks.index');
```

```
}
```

```

/**
 * Show the form for creating a new resource.
 */
public function create()
{
    $barangs = Barang::all();
    $kategoris = Kategori::all();
    $gudangs = Gudang::all();

    return view('raks.create', compact('barangs',
'kategoris', 'gudangs'));
}

/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
    $request->validate([
        'no_rak' => 'required',
        // 'id_barang' => 'required',
        //'id_kategori' => 'required',
        'id_gudang' => 'required',
    ]);

    $input = $request->all();
    // $barang =
Barang::find($input['id_barang']);
    // $input['id_kategori'] = $barang-
>id_kategori;
    $input["created_by"] = "1";

```

```

    $input["updated_by"] = "1";
    $input["deleted_by"] = "1";

    Rak::create($input);

    return redirect()->route('raks.index')-
>with('success', 'Rak created successfully');
}

/**
 * Display the specified resource.
 */
public function show(Rak $rak)
{
    return view('raks.show', compact('rak'));
}

/**
 * Show the form for editing the specified
resource.
 */
public function edit($id)
{
    $rak = Rak::find($id);
    $gudangs = Gudang::all();
    return response()->json([
        'rak' => $rak,
        'gudangs' => $gudangs
    ]);
}

```

```

/**
 * Update the specified resource in storage.
 */

public function update(Request $request, Rak $rak)
{
    $request->validate([
        'no_rak' => 'required',
        'id_gudang' => 'required',
    ]);

    $rak->update($request->all());

    return redirect()->route('raks.index')-
>with('success', 'Rak updated successfully');
}

/**
 * Remove the specified resource from
storage.
 */

public function destroy(Rak $rak)
{
    $rak->delete();

    return redirect()->route('raks.index')-
>with('success', 'Rak deleted successfully');
}
}
<?php

```

```

namespace App\Http\Controllers;

use App\Models\Transaksi;
use Illuminate\Http\Request;

class ReportController extends Controller
{
    public function printReport(Request $request)
    {
        $start_date = $request->input('start_date');
        $end_date = $request->input('end_date');
        $id_gudang = $request-
>input('id_gudang');

        $data =
Transaksi::whereBetween('tanggal_masuk',
[$start_date, $end_date])
        ->whereHas('rak.gudang', function
($query) use ($id_gudang) {
            $query->where('id', $id_gudang);
        })
        ->where('status', 'in')
        ->get();

        return
view('Transaksis.report_print_masuk',
compact('start_date', 'end_date', 'data'));
    }

    public function printReportOut(Request
$request)
    {

```

```

$start_date = $request->input('start_date');

$end_date = $request->input('end_date');

$id_gudang = $request-
>input('id_gudang');

```

```

$data =
Transaksi::whereBetween('tanggal_keluar',
[$start_date, $end_date])
->whereHas('rak.gudang', function
($query) use ($id_gudang) {
    $query->where('id', $id_gudang);
})
->where('status', 'out')
->get();

```

```

return
view('Transaksis.report_print_keluar',
compact('start_date', 'end_date', 'data'));
}
}

```

```
<?php
```

```
namespace App\Http\Controllers;
```

```

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Spatie\Permission\Models\Role;
use Spatie\Permission\Models\Permission;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Log;

```

```
class RoleController extends Controller
```

```

{
/**
 * RoleController constructor.
 */
public function __construct()
{
    $this->middleware('permission:role-
list|role-create|role-edit|role-delete', ['only' =>
['index', 'store']]);

    $this->middleware('permission:role-create',
['only' => ['create', 'store']]);

    $this->middleware('permission:role-edit',
['only' => ['edit', 'update']]);

    $this->middleware('permission:role-delete',
['only' => ['destroy']]);
}

/**
 * Display a listing of the resource.
 *
 * @param Request $request
 * @return \Illuminate\Http\Response
 */
public function index(Request $request)
{
    $roles = Role::orderBy('id', 'DESC')-
>paginate(5);

    return view('roles.index', compact('roles'))
->with('i', ($request->input('page', 1) - 1)
* 5);
}

```

```

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    $permissions = Permission::all();
    return view('roles.create',
compact('permissions'));
}

/**
 * Store a newly created resource in storage.
 *
 * @param Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $request->validate([
        'name' => 'required|unique:roles,name',
        // 'permission' => 'required',
        'permission' => 'required|array',
        'permission.*' => 'exists:permissions,id'
    ]);

    $role = Role::create(['name' => $request-
>input('name')]);

```

```

// $role->syncPermissions($request-
>input('permission'));

// return redirect()->route('roles.index')
// ->with('success', 'Role created
successfully');

// Fetch permissions by their IDs
$permissions = Permission::whereIn('id',
$request->input('permission'))->get();

// Log the permissions to debug
Log::info('Fetched permissions: ',
$permissions->toArray());

// Sync permissions with the role
$role->syncPermissions($permissions);

// Redirect back to the roles index with a
success message
return redirect()->route('roles.index')
->with('success', 'Role created
successfully');
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)

```



```

{
    $role = Role::findOrFail($id);

    $rolePermissions = $role->permissions; //
    Assuming you have a relationship defined

    return view('roles.show', compact('role',
    'rolePermissions'));
}

/**
 * Show the form for editing the specified
    resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    $role = Role::findOrFail($id);

    $permission = Permission::all();

    $rolePermissions =
    DB::table("role_has_permissions")
        ->where("role_has_permissions.role_id",
    $id)
        -
    >pluck('role_has_permissions.permission_id',
    'role_has_permissions.permission_id')
        ->all();

    return view('roles.edit', compact('role',
    'permission', 'rolePermissions'));
}

/**

```

```

 * Update the specified resource in storage
 * @param Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $id)
{
    $request->validate([
        'name' => 'required',
        'permission' => 'required',
    ]);

    $role = Role::findOrFail($id);
    $role->name = $request->input('name');
    $role->save();

    $role->syncPermissions($request-
    >input('permission'));

    return redirect()->route('roles.index')
        ->with('success', 'Role updated
    successfully');
}

/**
 * Remove the specified resource from
    storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)

```

```

{
    Role::findOrFail($id)->delete();
    return redirect()->route('roles.index')
        ->with('success', 'Role deleted
successfully');
}
}
@extends('barangs.layout')
@section('content')
<div class="row">
    <div class="col-lg-12 margin-tb">
        <div class="pull-left">
            <h2> jangan lupa isi</h2>
        </div>
        <div class="pull-right">
            <a class="btn btn-success" href="{{ {
route('barangs.create') }}"> Create New
Barang</a>
        </div>
    </div>
</div>

@if ($message = Session::get('success'))
    <div class="alert alert-success">
        <p>{{ $message }}</p>
    </div>
@endif

<table class="table table-bordered">
    <tr>
        <th>No</th>

```

```

        <th>Kode Buku</th>
        <th>Judul</th>
        <th>Kelas</th>
        <th>Penerbit</th>
        <th>Tahun Terbit</th>
        <th>Kategori</th>
        <th>Stock</th>
        <th width="280px">Action</th>
    </tr>

    @foreach ($barangs as $barang)
        <tr>
            <td>{{ ++$i }}</td>
            <td>{{ $barang->kode_buku }}</td>
            <td>{{ $barang->judul }}</td>
            <td>{{ $barang->kelas }}</td>
            <td>{{ $barang->penerbit }}</td>
            <td>{{ $barang->tahun_terbit }}</td>
            <td>
                {{ $barang->kategori->jenjang }}
            </td>
            <td>{{ $barang->stock }}</td>
            <td>
                <form action="{{ {
route('barangs.destroy', $barang->id) }}"
method="POST">

                    <a class="btn btn-info" href="{{ {
route('barangs.show', $barang->id)
}} ">Show</a>

```

```
        <a class="btn btn-primary"
href="{{ route('barangs.edit', $barang->id)
}}">Edit</a>
```

```
@csrf
```

```
@method('DELETE')
```

```
        <button type="submit"
class="btn btn-danger">Delete</button>
```

```
</form>
```

```
</td>
```

```
</tr>
```

```
@endforeach
```

```
</table>
```

```
{!! $barangs->links() !!}
```

```
@endsection
```

```
@extends('layout')
```

```
@section('content')
```

```
<div class="row mt-4">
```

```
<div class="col-lg-12 d-flex justify-
content-between align-items-center">
```

```
<h2 class="text-primary"><i class="fas
fa-map-marker-alt mr-2"></i>Lokasi
Gudang</h2>
```

```
<a class="btn btn-primary" href="{{
route('gudangs.create') }}"><i class="fas fa-plus
mr-1"></i>Create New
```

```
Gudang</a>
```

```
</div>
```

```
</div>
```

```
@if ($message = Session::get('success'))
```

```
<div class="alert alert-success mt-3">
```

```
<p>{{ $message }}</p>
```

```
</div>
```

```
@endif
```

```
<div class="row mt-4">
```

```
<div class="col-lg-12">
```

```
<div class="card shadow-sm">
```

```
<div class="card-header bg-primary
text-white">
```

```
<h4 class="mb-0">Data Lokasi
Gudang</h4>
```

```
</div>
```

```
<div class="card-body">
```

```
<table class="table table-bordered
table-hover data-table">
```

```
<thead class="table-primary">
```

```
<tr>
```

```
<th>No</th>
```

```
<th>Kode Perwakilan
Regional</th>
```

```
<th>Lokasi Gudang</th>
```

```
<th>Nomor Gudang</th>
```

```
<th
width="280px">Action</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
</tbody>
```

```

        </table>
    </div>
</div>
</div>
</div>
    }
    ]
<script type="text/javascript">
    $(function() {
        var table = $('<div>#data-table</div>').DataTable({
            processing: true,
            serverSide: true,
            ajax: "{ route('gudangs.index') }",
            columns: [{
                data: 'id',
                name: 'id'
            },
            {
                data: 'kode_perwakilan_regional',
                name:
'kode_perwakilan_regional'
            },
            {
                data: 'lokasi_gudang',
                name: 'lokasi_gudang'
            },
            {
                data: 'no_gudang',
                name: 'no_gudang'
            },
            {
                data: 'action',
                name: 'action',
                orderable: false,
                searchable: false
            }
        ]
    });
</script>
@endsection

@section('styles')
<style>
    .card {
        border: 1px solid #007bff;
        border-radius: 5px;
    }

    .card-body {
        padding: 20px;
    }

    .thead-dark th {
        background-color: #343a40;
        color: white;
    }

    .btn-success {
        background-color: #28a745;
        border: none;
    }
</style>
</section>

```

```

    }

.alert-success {
    background-color: #d4edda;
    border-color: #c3e6cb;
    color: #155724;
}

</style>
@endsection

@extends('layout')

@section('content')
    <div class="row mt-4">

        <div class="col-lg-12 d-flex justify-
content-between align-items-center">

            <h2><i class="fas fa-th-list mr-2"></i>
Kategori</h2>

            <a class="btn btn-primary" href="{{ {
route('kategoris.create') }}">Buat Kategori
Baru</a>

        </div>

    </div>

    @if ($message = Session::get('success'))

        <div class="alert alert-success mt-3">

            <p>{{ $message }}</p>

        </div>

    @endif

    <div class="row mt-4">

```

```

<div class="col-lg-12">

    <div class="card shadow-sm">

        <div class="card-header bg-primary
text-white">

            <h4 class="mb-0">Data Kategori
Buku</h4>

        </div>

        <div class="card-body">

            <div class="table-responsive">

                <table class="table table-
bordered table-hover data-table">

                    <thead class="table-primary">

                        <tr>

                            <th>No</th>

                            <th>Jenjang</th>

                            <th>Kelas</th>

                            <th
width="280px">Aksi</th>

                        </tr>

                    </thead>

                    <tbody>

                        </tbody>

                </table>

            </div>

        </div>

    </div>

    <div>

        </div>

    </div>

    </div>

    <script type="text/javascript">

```

```

$(function() {
    var table = $('table').DataTable({
        processing: true,
        serverSide: true,
        ajax: "{{ route('kategoris.index') }}",
        columns: [{
            data: 'id',
            name: 'id'
        },
        {
            data: 'jenjang',
            name: 'jenjang'
        },
        {
            data: 'kelas',
            name: 'kelas'
        },
        {
            data: 'action',
            name: 'action',
            orderable: false,
            searchable: false,
            render: function(data, type, row)
            {
                return `
                    <a
href="/kategoris/${row.id}" class="btn btn-info
btn-sm me-1"><i class="fas fa-eye"></i></a>

                    <a
href="/kategoris/${row.id}/edit" class="btn btn-
primary btn-sm me-1"><i class="fas fa-
edit"></i></a>

```

```

<form
action="/kategoris/${row.id}" method="POST"
style="display:inline-block;">

    @csrf

    @method('DELETE')

    <button type="submit"
class="btn btn-danger btn-sm"><i class="fas fa-
trash"></i></button>

</form>;

    }
}
]
});
});
</script>
@endsection

@extends('layout')
@section('content')
    <div class="row">
        <div class="col-lg-12 margin-tb">
            <div class="pull-left">
                <h2>Petugas</h2>
            </div>
            <div class="pull-right">
                <a class="btn btn-success" href="{{
route('petugas.create') }}">Create New
Petugas</a>
            </div>
        </div>
    </div>

```

```

@if ($message = Session::get('success'))
    <div class="alert alert-success">
        <p>{{ $message }}</p>
    </div>
@endif

<table class="table table-bordered data-table">
    <thead>
        <tr>
            <th>No</th>
            <th>Nama</th>
            <th>No Handphone</th>
            <th>Alamat</th>
            <th>Jenis Kelamin</th>
            <th>Foto</th>
            <th width="280px">Action</th>
        </tr>
    </thead>
    <tbody>
    </tbody>
</table>

<script type="text/javascript">
    $(function() {
        var table = $('#data-table').DataTable({
            processing: true,
            serverSide: true,
            ajax: "{{ route('petugas.index') }}",
            columns: [{
                data: 'id',
                name: 'id'
            },
            {
                data: 'nama',
                name: 'nama'
            },
            {
                data: 'no_hp',
                name: 'no_hp'
            },
            {
                data: 'alamat',
                name: 'alamat'
            },
            {
                data: 'jenis_kelamin',
                name: 'jenis_kelamin'
            },
            {
                data: 'foto',
                name: 'foto',
                render: function(data, type, full,
                    meta) {
                    return '{{ $full['foto'] }}</img>';
                }
            },
            {
                data: 'id',
                name: 'id'
            }
        ]
    });

```

```
        data: 'action',
        name: 'action',
        orderable: false,
        searchable: false
    }
]
});
});
</script>
@endsection
```


