**app\Models\Protap.php**

<?php

namespace App\Models;

use Backpack\CRUD\app\Models\Traits\CrudTrait;

use Illuminate\Database\Eloquent\Model;

class Protap extends Model

{

    use CrudTrait;

    /\*

    |--------------------------------------------------------------------------

    | GLOBAL VARIABLES

    |--------------------------------------------------------------------------

    \*/

    protected $table = 'protaps';

    // protected $primaryKey = 'id';

    // public $timestamps = false;

    protected $guarded = ['id'];

    // protected $fillable = [];

    // protected $hidden = [];

    // protected $dates = [];

    /\*

    |--------------------------------------------------------------------------

    | FUNCTIONS

    |--------------------------------------------------------------------------

    \*/

    public function setFileAttribute($value)

    {

        $attribute\_name = "file";

        $disk = "public\_beneran";

        $destination\_path = 'file/' . date("Ymd");

        $this->uploadFileToDisk($value, $attribute\_name, $disk, $destination\_path);

    }

}

**app\Http\Controllers\Admin\ProtapCrudController.php**

<?php

namespace App\Http\Controllers\Admin;

use App\Http\Requests\ProtapRequest;

use App\Models\Protap;

use Backpack\CRUD\app\Http\Controllers\CrudController;

use Backpack\CRUD\app\Library\CrudPanel\CrudPanelFacade as CRUD;

/\*\*

 \* Class ProtapCrudController

 \* @package App\Http\Controllers\Admin

 \* @property-read \Backpack\CRUD\app\Library\CrudPanel\CrudPanel $crud

 \*/

class ProtapCrudController extends CrudController

{

    use \Backpack\CRUD\app\Http\Controllers\Operations\ListOperation;

    use \Backpack\CRUD\app\Http\Controllers\Operations\CreateOperation;

    use \Backpack\CRUD\app\Http\Controllers\Operations\UpdateOperation;

    use \Backpack\CRUD\app\Http\Controllers\Operations\DeleteOperation;

    use \Backpack\CRUD\app\Http\Controllers\Operations\ShowOperation;

    /\*\*

     \* Configure the CrudPanel object. Apply settings to all operations.

     \*

     \* @return void

     \*/

    public function setup()

    {

        CRUD::setModel(\App\Models\Protap::class);

        CRUD::setRoute(config('backpack.base.route\_prefix') . '/protap');

        CRUD::setEntityNameStrings('protap', 'protaps');

    }

    /\*\*

     \* Define what happens when the List operation is loaded.

     \*

     \* @see  https://backpackforlaravel.com/docs/crud-operation-list-entries

     \* @return void

     \*/

    protected function setupListOperation()

    {

        $this->crud->addButtonFromView('line', 'preview', 'preview', 'end');

        $this->crud->addButtonFromView('line', 'download', 'download', 'end');

        $this->crud->denyAccess('show');

        CRUD::column('id');

        CRUD::column('no\_petunjuk');

        CRUD::column('name');

        CRUD::column('tgl');

        // CRUD::column('file');

        CRUD::column('pembina');

        // CRUD::column('created\_at');

        // CRUD::column('updated\_at');

        /\*\*

         \* Columns can be defined using the fluent syntax or array syntax:

         \* - CRUD::column('price')->type('number');

         \* - CRUD::addColumn(['name' => 'price', 'type' => 'number']);

         \*/

    }

    /\*\*

     \* Define what happens when the Create operation is loaded.

     \*

     \* @see https://backpackforlaravel.com/docs/crud-operation-create

     \* @return void

     \*/

    protected function setupCreateOperation()

    {

        CRUD::setValidation(ProtapRequest::class);

        // CRUD::field('id');

        CRUD::field('no\_petunjuk');

        CRUD::field('name');

        CRUD::field('tgl')->type('date');

        CRUD::field('file')->type('upload')->upload(true)->disk('public\_beneran');

        CRUD::field('pembina');

        // CRUD::field('created\_at');

        // CRUD::field('updated\_at');

        /\*\*

         \* Fields can be defined using the fluent syntax or array syntax:

         \* - CRUD::field('price')->type('number');

         \* - CRUD::addField(['name' => 'price', 'type' => 'number']));

         \*/

    }

    /\*\*

     \* Define what happens when the Update operation is loaded.

     \*

     \* @see https://backpackforlaravel.com/docs/crud-operation-update

     \* @return void

     \*/

    protected function setupUpdateOperation()

    {

        $this->setupCreateOperation();

    }

    public function download($id)

    {

        $doktrin = Protap::find($id);

        $file\_path = public\_path($doktrin->file);

        return response()->download($file\_path);

    }

    public function previewpdf($id)

    {

        $doktrin = Protap::find($id);

        $file\_path = public\_path($doktrin->file);

        return response()->file($file\_path);

    }

}

**database\migrations\2022\_04\_06\_172127\_protap.php**

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class Protap extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('protaps', function (Blueprint $table) {

            $table->id();

            $table->string('no\_petunjuk');

            $table->string('name');

            $table->date('tgl');

            $table->string('file');

            $table->string('pembina')->nullable();

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('protaps');

    }

}