

Nama : Gilang Ashar Aldiansyah

NIM : H1D021060

Migration :

Surat

```
1. <?php
2.
3. use Illuminate\Database\Migrations\Migration;
4. use Illuminate\Database\Schema\Blueprint;
5. use Illuminate\Support\Facades\Schema;
6.
7. return new class extends Migration
8. {
9.     /**
10.      * Run the migrations.
11.      */
12.     public function up(): void
13.     {
14.         Schema::create('surats', function (Blueprint $table) {
15.             $table->id();
16.             $table->bigInteger('nomor_surat');
17.             $table->foreign('pengirim')->references('id')->
18. >on('pengirim')->onDelete('cascade');
19.             $table->foreign('penerima')->references('id')->
20. >on('penerima')->onDelete('cascade');
21.             $table->char('isi_surat');
22.             $table->foreign('kategori')->references('id')->
23. >on('kategori')->onDelete('cascade');
24.             $table->timestamps();
25.         });
26.     }
27.
28.     /**
29.      * Reverse the migrations.
30.      */
31.     public function down(): void
32.     {
33.         Schema::dropIfExists('surats');
34.     }
35. }
```

Pengirim

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('pengirims', function (Blueprint $table) {
            $table->id();
            $table->char('pengirim');
            $table->bigInteger('no_hp');
            $table->char('alamat');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('pengirims');
    }
};
```

Penerima

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('penerimas', function (Blueprint $table) {
            $table->id();
            $table->char('penerima');
            $table->bigInteger('no_hp');
            $table->char('alamat');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('penerimas');
    }
};
```

Kategori

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('kategoris', function (Blueprint $table) {
            $table->id();
            $table->char('nama_kategori');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('kategoris');
    }
};
```

CRUD

Surat

```
<?php

namespace App\Http\Controllers;

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

class SuratController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        $surat = Surat::orderBy('created_at', 'DESC')->get();

        return view('surat.index', compact('surat'));
    }

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

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        $request->validate([
            'nomor_surat' => 'required|numeric|min:0',
            'pengirim' => 'required|min:0',
            'penerima' => 'required|min:0',
            'isi_surat' => 'required|min:0',
            'kategori' => 'required|min:0'
        ]);

        Surat::create([
            'nomor_surat' => $request->nomor_surat,
            'pengirim' => $request->pengirim,
            'penerima' => $request->penerima,
            'isi_surat' => $request->isi_surat,
```

```

        'kategori' => $request->kategori,
    ]);

    return redirect()->route('surat.index')->with('success', 'Produk
Berhasil Ditambahkan');
}

/**
 * Display the specified resource.
 */
public function show(string $id)
{
    $surat = Surat::findOrFail($id);

    return view('surat.show', compact('surat'));
}

/**
 * Show the form for editing the specified resource.
 */
public function edit(string $id)
{
    $surat = Surat::findOrFail($id);

    return view('surat.edit', compact('surat'));
}

public function update(Request $request, string $id)
{
    $surat = Surat::findOrFail($id);
    $request->validate([
        'nomor_surat' => 'required|numeric|min:0',
        'pengirim' => 'required|min:0',
        'penerima' => 'required|min:0',
        'isi_surat' => 'required|min:0',
        'kategori' => 'required|min:0'
    ]);
    $surat->update($request->all());

    return redirect()->route('surat.index')->with('success', 'Produk
berhasil diperbaharui');
}

/**
 * Remove the specified resource from storage.
 */
public function destroy(string $id)
{

```

```
        $surat = Surat::findOrFail($id);

        $surat->delete();

        return redirect()->route('surat.index')->with('success', 'Surat
berhasil dihapus');
    }
}
```

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Surat extends Model
{
    use HasFactory;

    protected $fillable = [
        'nomor_surat',
        'pengirim',
        'penerima',
        'isi_surat',
        'kategori',
    ];
}
```

Pengirim

```
<?php

namespace App\Http\Controllers;

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

class PengirimController extends Controller
{
    public function index()
    {
        $pengirim = Pengirim::orderBy('created_at', 'DESC')->get();

        return view('pengirim.index', compact('pengirim'));
    }

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

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        $request->validate([
            'pengirim' => 'required|min:0',
            'no_hp' => 'required|numeric|min:0',
            'alamat' => 'required|min:0',
        ]);

        pengirim::create([
            'pengirim' => $request->pengirim,
            'no_hp' => $request->no_hp,
            'alamat' => $request->alamat,
        ]);

        return redirect()->route('pengirim.index')->with('success', 'Data
        Berhasil Ditambahkan');
    }

    /**
     * Display the specified resource.
     */
}
```



```

    */
    public function show(string $id)
    {
        $pengirim = Pengirim::findOrFail($id);

        return view('pengirim.show', compact('pengirim'));
    }

    /**
     * Show the form for editing the specified resource.
     */
    public function edit(string $id)
    {
        $pengirim = Pengirim::findOrFail($id);

        return view('pengirim.edit', compact('pengirim'));
    }

    public function update(Request $request, string $id)
    {
        $pengirim = Pengirim::findOrFail($id);
        $request->validate([
            'pengirim' => 'required|min:0',
            'no_hp' => 'required|numeric|min:0',
            'alamat' => 'required|min:0',
        ]);
        $pengirim->update($request->all());

        return redirect()->route('pengirim.index')->with('success', 'Data
berhasil diperbaharui');
    }

    /**
     * Remove the specified resource from storage.
     */
    public function destroy(string $id)
    {
        $pengirim = Pengirim::findOrFail($id);

        $pengirim->delete();

        return redirect()->route('pengirim.index')->with('success', 'Data
berhasil dihapus');
    }
}

```

Penerima

```
<?php

namespace App\Http\Controllers;

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

class PenerimaController extends Controller
{
    public function index()
    {
        $penerima = Penerima::orderBy('created_at', 'DESC')->get();

        return view('penerima.index', compact('penerima'));
    }

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

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        $request->validate([
            'penerima' => 'required|min:0',
            'no_hp' => 'required|numeric|min:0',
            'alamat' => 'required|min:0',
        ]);

        penerima::create([
            'penerima' => $request->penerima,
            'no_hp' => $request->no_hp,
            'alamat' => $request->alamat,
        ]);

        return redirect()->route('penerima.index')->with('success', 'Data
        Berhasil Ditambahkan');
    }

    /**
     * Display the specified resource.
     */
}
```

```

    */
    public function show(string $id)
    {
        $penerima = Penerima::findOrFail($id);

        return view('penerima.show', compact('penerima'));
    }

    /**
     * Show the form for editing the specified resource.
     */
    public function edit(string $id)
    {
        $penerima = penerima::findOrFail($id);

        return view('penerima.edit', compact('penerima'));
    }

    public function update(Request $request, string $id)
    {
        $penerima = penerima::findOrFail($id);
        $request->validate([
            'penerima' => 'required|min:0',
            'no_hp' => 'required|numeric|min:0',
            'alamat' => 'required|min:0',
        ]);
        $penerima->update($request->all());

        return redirect()->route('penerima.index')->with('success', 'Data
berhasil diperbaharui');
    }

    /**
     * Remove the specified resource from storage.
     */
    public function destroy(string $id)
    {
        $pengirim = Penerima::findOrFail($id);

        $pengirim->delete();

        return redirect()->route('pengirim.index')->with('success', 'Data
berhasil dihapus');
    }
}

```

Login

```
<?php

namespace App\Http\Controllers;
use App\Models\User;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;

class LoginController extends Controller
{
    public function index()
    {
        return view('login.index', [
            'title' => 'login'
        ]);
    }

    public function store(Request $request){
        $validatedData = $request->validate([
            'name' => 'required|max:255',
            'email' => 'required|email:dns|unique:users',
            'password' => 'required|min:8|max:255'
        ]);

        $validatedData['password'] = Hash::make($validatedData['password']);

        User::create($validatedData);

        return redirect()->route('surat.index')->with('success', 'Berhasil
Login');
    }

    public function authenticate(Request $request){
        $credentials = $request->validate([
            'email' => 'required|email:dns',
            'password' => 'required',
        ]);

        if(Auth::attempt($credentials)) {
            $request->session()->regenerate();
            return redirect()->intended('/surat');
        }

        return back()->withErrors([
```

```
        'email' => 'The provided credentials do not match our records.',  
    ])->onlyInput('email');  
}  
}
```

Login

Email

Password

Submit

Daftar Surat

Tambahkan Surat

No Surat	Pengirim	Penerima	Isi Surat	Kategori	Aksi
1	Azhar	gilang	ggwp	Resmi	<div>DetailEditDelete</div>

Detail surat

Nomor Surat

12

pengirim

Azhar

Penerima

gilang

Isi surat

ggwp

Kategori

Resmi

Created At

2023-07-03 08:08:08

Updated At

2023-07-03 08:08:08



Tambahkan Surat