**Tugas 7 Mandiri**

Pengembangan Perangkat Lunak





**Pemograman Backend**

Nama : Faiz Haidar

NIM : 22255601008

**PeTIK II Jombang YBM PLN Tahun Akademik 2022/2023**

**Plosogeneng, Kec. Jombang, Kabupaten Jombang, Jawa Timur**

**Laporan praktikum dari materi Form Processing dan Validasi.**

1. Membuat model Pegawai dengan perintah





1. Menambahkan script berikut ke dalam file *Pegawai.php* yg berfungsi sebagai mapping ke tabel pegawai

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

*class* Pegawai *extends* *Model*

{

    use HasFactory;

*protected* $table = 'pegawai';

}

1. Dengan perintah di bawah ini, maka di dalam folder app/Http/controller akan ada file baru PegawaiController.php. Di dalam file PegawaiController.php yang berada di folder folder app/Http/controller, secara otomatis akan terbentuk fungsi-fungsi CRUD (Create, Read, Update, Delete) tetapi masih kosong. Tugas kitalah yang akan mengisi body fungsi-fungsi yang berada di dalam controller tersebut.



1. Menambahkan Route baru di **routes/web.php**

Route::resource('pegawai', PegawaiController::class);

1. Membuat fungsi index pada *PegawaiController* untuk menampilkan data pada tabel pegawai

*public* *function* index()

    {

*//dapatkan seluruh data pegawai dengan query builder*

        $ar\_pegawai = DB::table('pegawai')->get();

*//arahkan ke halaman baru dengan menyertakan data pegawai(compact)*

*//di resources/views/pegawai/index.blade.php*

        return view('pegawai.index', compact('ar\_pegawai'));

    }

1. Membuat fungsi create pada *PegawaiController* untuk menampilkan form

*public* *function* create()

    {

*//mengarahkan ke hal form input*

        return view('pegawai.form');

    }

1. Membuat tampilan form pada views/pegawai/form.blade.php

@extends('adminlte::page')

@section('title', 'Data Pegawai')

@section('content\_header')

    @if ($errors->any())

        <div *class*="alert alert-danger">

            <ul>

                @foreach ($errors->all() as $error)

                    <li>{{ $error }}</li>

                @endforeach

            </ul>

        </div>

    @endif

    <h1>Form Pegawai</h1>

@stop

@section('content')

<form *action*="{{ route('pegawai.store') }}" *method*="POST">

    @csrf

    <div *class*="form-group">

        <label *for*="">NIP</label>

        <input *type*="text" *name*="nip" *class*="form-control">

    </div>

    <div *class*="form-group">

        <label *for*="">Nama</label>

        <input *type*="text" *name*="nama" *class*="form-control">

    </div>

    <div *class*="form-group">

        <label *for*="">Alamat</label>

        <textarea  *name*="alamat" *class*="form-control"></textarea>

    </div>

    <div *class*="form-group">

        <label *for*="">Email</label>

        <input *type*="email" *name*="email" *class*="form-control">

    </div>

    <button *type*="submit" *class*="btn btn-primary">Simpan</button>

</form>

@stop

@section('css')

    <link *rel*="stylesheet" *href*="/css/admin\_custom.css">

@stop

@section('js')

    <script>

        consol.log('Hi')

    </script>

@stop

1. Membuat fungsi store pada PegawaiController berfungsi untuk validasi data pada from yang telah dibuat

*public* *function* store(Request $request)

    {

*//1. proses validasi data*

        $validasi = $request->validate(

            [

                'nip' => 'required|unique:pegawai|numeric',

                'nama' => 'required|max:50',

                'alamat' => 'required',

                'email' => 'required|max:50|regex:/(.+)@(.+)\.(.+)/i',

            ],

*//2. menampilkan pesan kesalahan*

*//(di slide selanjutnya)*

*//pesan kesalahan saat invalid data (kelanjutan slide sebelumnya)*

            [

                'nip.required' => 'NIP Wajib di Isi',

                'nip.unique' => 'NIP Tidak Boleh Sama',

                'nip.numeric' => 'Harus Berupa Angka',

                'nama.required' => 'Nama Wajib di Isi',

                'alamat.required' => 'Alamat Wajib di Isi',

                'email.required' => 'Email Wajib di Isi',

                'email.regex' => 'Harus berformat email',

            ],

        );

*//lanjutan slide sebelumnya*

*//2. proses input data tangkap request dari form input*

        DB::table('pegawai')->insert(

            [

                'nip' => $request->nip,

                'nama' => $request->nama,

                'alamat' => $request->alamat,

                'email' => $request->email,

            ]

        );

*//2.landing page*

        return redirect('/pegawai');

    }

1. Tampilan pada form yang telah dibuat



