

LAPORAN PRAKTIKUM PEMROGRAMAN WEB

PRAKTIKUM 11 – MVC ROUTE



Disusun oleh:

Nama : Alfath Roziq Widhayaka

Nim : L0122012

Kelas : A

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA

UNIVERSITAS SEBELAS MARET

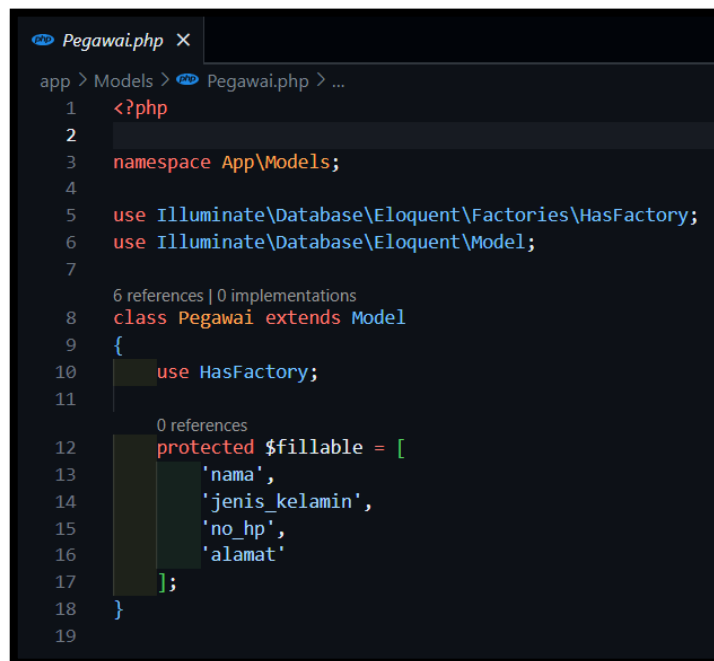
2024

BAB I

A. Penjelasan Source Code

Tugas praktikum Pemrograman Web week ke 11 ini saya membuat project Laravel dengan menerapkan materi MVC Route. MVC Route disini yang saya buat adalah data pegawai sederhana yang juga menerapkan sistem CRUD. Fungsi MVC Route disini sebagai untuk alur kerja aplikasi menjadi lebih terstruktur dan mudah untuk dikelola. Route dalam Laravel berfungsi untuk menentukan URL yang akan diakses oleh pengguna dan menghubungkannya dengan Controller yang sesuai. Cara untuk membuat MVC Route dengan mudah yaitu dengan mengetik pada terminal **php artisan make pegawai -mrc**. Berikut penjelasan untuk masing-masing pembuatan MVC Route nya.

1. Model



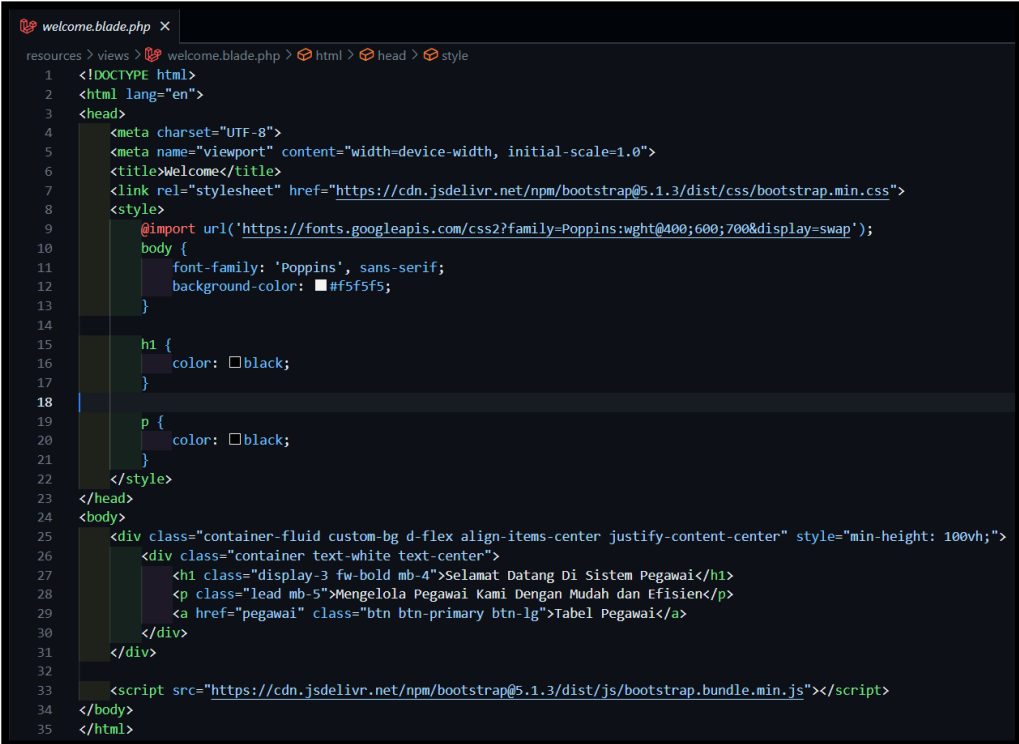
```
Pegawai.php x
app > Models > Pegawai.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  6 references | 0 implementations
9  class Pegawai extends Model
10 {
11     use HasFactory;
12
13     0 references
14     protected $fillable = [
15         'nama',
16         'jenis_kelamin',
17         'no_hp',
18         'alamat'
19     ];
20 }
```

Gambar 1 Pegawai.php

Model Pegawai ini merupakan sebuah kelas yang digunakan untuk berinteraksi dengan tabel "**pegawai**" di database. Kelas ini mewarisi fitur-fitur dari kelas induk Model yang disediakan oleh **Eloquent**, sehingga dapat memanfaatkan kemampuan ORM yang kuat. Kelas ini menggunakan **trait HasFactory** untuk memungkinkan pembuatan data palsu selama pengujian. Properti **\$fillable** mendefinisikan atribut-atribut yang dapat diisi secara massal saat membuat atau memperbarui data pegawai, seperti **nama**, **jenis_kelamin**, **no_hp**, dan **alamat**.

2. View

- Welcome



```
resources > views > welcome.blade.php > html > head > style
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Welcome</title>
7   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css">
8   <style>
9     @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;600;700&display=swap');
10    body {
11      font-family: 'Poppins', sans-serif;
12      background-color: #f5f5f5;
13    }
14
15    h1 {
16      color: black;
17    }
18
19    p {
20      color: black;
21    }
22  </style>
23 </head>
24 <body>
25   <div class="container-fluid custom-bg d-flex align-items-center justify-content-center" style="min-height: 100vh;">
26     <div class="container text-white text-center">
27       <h1 class="display-3 fw-bold mb-4">Selamat Datang Di Sistem Pegawai</h1>
28       <p class="lead mb-5">Mengelola Pegawai Kami Dengan Mudah dan Efisien</p>
29       <a href="pegawai" class="btn btn-primary btn-lg">Tabel Pegawai</a>
30     </div>
31   </div>
32
33   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
34 </body>
35 </html>
```

Gambar 2 welcome.blade.php

Kode View milik **welcome** diatas merupakan kode HTML yang isinya nanti dapat menampilkan halaman awal yang berisi judul dan tombol yang bilamana ditekan akan berpindah ke halaman **tabel data** pegawai milik **datapegawai.blade.php**.

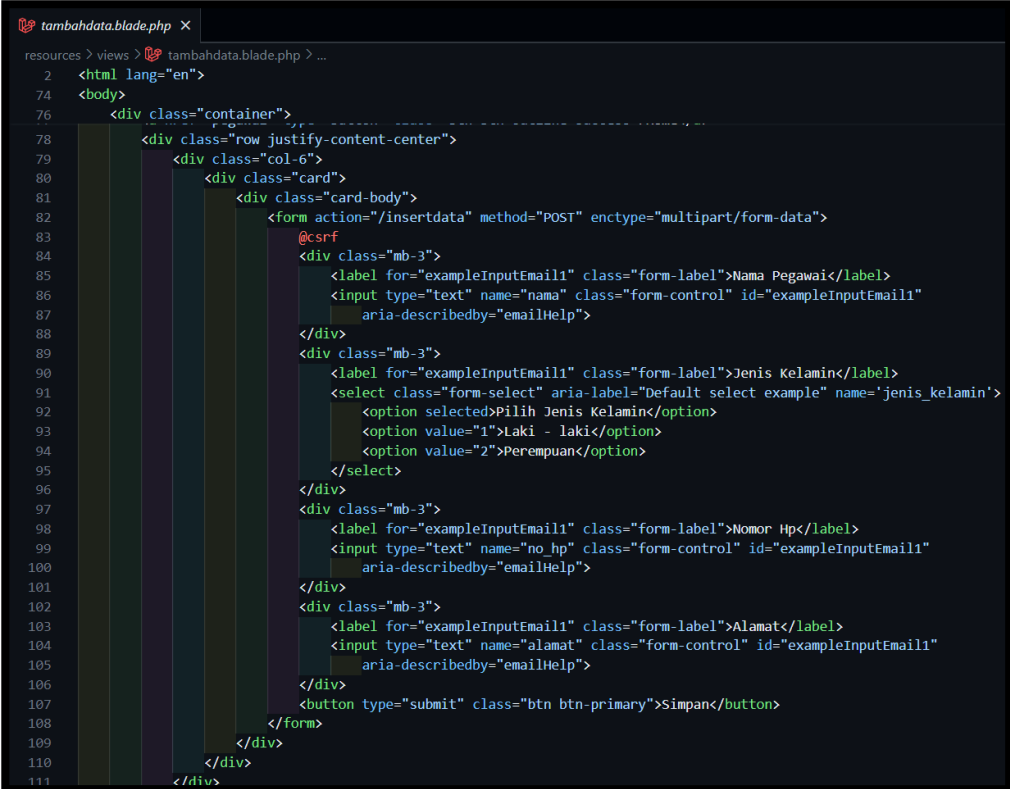
▪ Tabel

```
datapegawai.blade.php
resources > views > datapegawai.blade.php > html > head > style > body
2  <html lang="en">
63 <body>
65   <div class="container">
69     <div class="row">
70       <table class="table table-hover">
72         <tr>
73           <th scope="col">#</th>
74           <th scope="col">Nama</th>
75           <th scope="col">Jenis Kelamin</th>
76           <th scope="col">No Hp</th>
77           <th scope="col">Alamat</th>
78           <th class="text-center" scope="col">Ops</th>
79         </tr>
80       </thead>
81       <tbody class="table-group-divider">
82         @foreach ($data as $row)
83           <tr>
84             <th scope="row">{{ $row->id }}</th>
85             <td>{{ $row->nama }}</td>
86             <td>{{ $row->jenis_kelamin }}</td>
87             <td>{{ $row->no_hp }}</td>
88             <td>{{ $row->alamat }}</td>
89             <td>
90               <a href="editdata/{{ $row->id }}" type="button"
91                 class="btn btn-outline-warning">Edit</a>
92               <a href="delete/{{ $row->id }}"
93                 onclick="return confirm('Yakin hapus pegawai atas nama {{ $row->nama }} ?')
94                 type="button" class="btn btn-outline-danger">Hapus</a>
95             </td>
96           </tr>
97         @endforeach
98       </tbody>
99     </table>
100   </div>
101 </div>
102 </body>
```

Gambar 3 datapegawai.blade.php

Kode HTML milik **datapegawai.blade.php** diatas merupakan kode yang nantinya akan menampilkan halaman yang berisi tabel data pegawai. Halaman ini pengguna dapat melihat daftar pegawai melalui tabel dan juga dapat melakukan menambah, mengedit, dan menghapus pegawai.

- **Tambah Data**

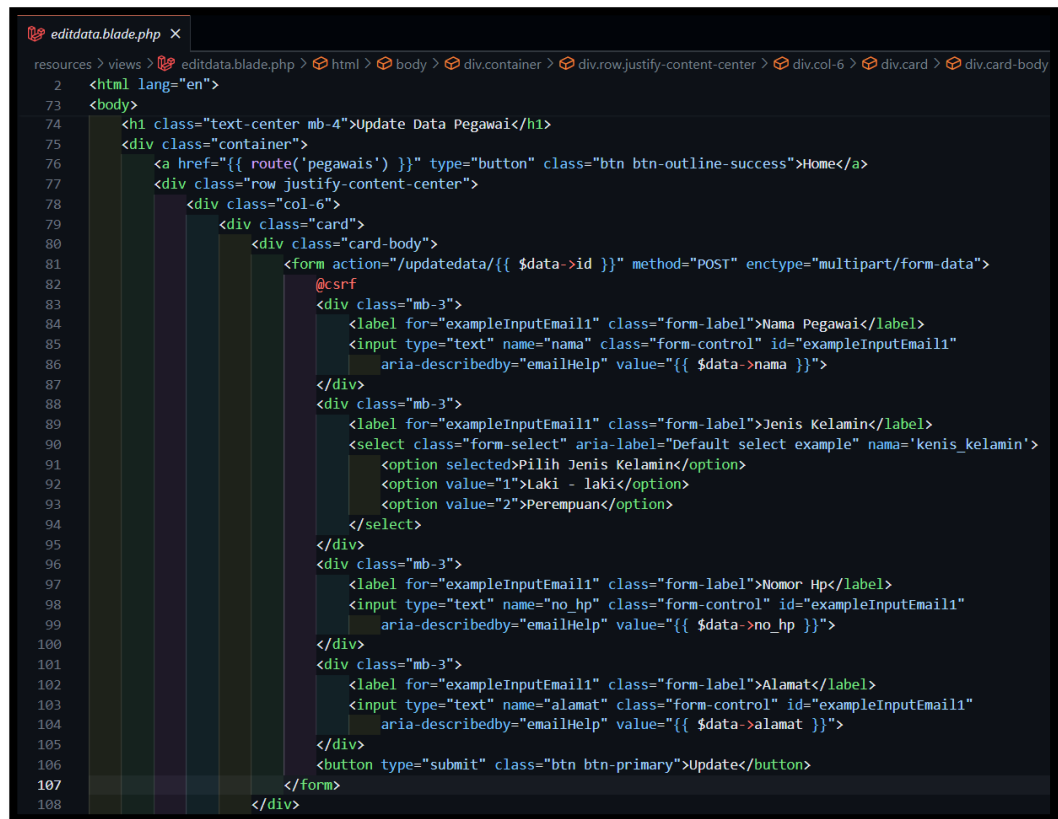


```
resources > views > tambahdata.blade.php > ...
2  <html lang="en">
74 <body>
76   <div class="container">
78     <div class="row justify-content-center">
79       <div class="col-6">
80         <div class="card">
81           <div class="card-body">
82             <form action="/insertdata" method="POST" enctype="multipart/form-data">
83               @csrf
84               <div class="mb-3">
85                 <label for="exampleInputEmail1" class="form-label">Nama Pegawai</label>
86                 <input type="text" name="nama" class="form-control" id="exampleInputEmail1"
87                   aria-describedby="emailHelp">
88               </div>
89               <div class="mb-3">
90                 <label for="exampleInputEmail1" class="form-label">Jenis Kelamin</label>
91                 <select class="form-select" aria-label="Default select example" name='jenis_kelamin'>
92                   <option selected>Pilih Jenis Kelamin</option>
93                   <option value="1">Laki - laki</option>
94                   <option value="2">Perempuan</option>
95                 </select>
96               </div>
97               <div class="mb-3">
98                 <label for="exampleInputEmail1" class="form-label">Nomor Hp</label>
99                 <input type="text" name="no_hp" class="form-control" id="exampleInputEmail1"
100                   aria-describedby="emailHelp">
101               </div>
102               <div class="mb-3">
103                 <label for="exampleInputEmail1" class="form-label">Alamat</label>
104                 <input type="text" name="alamat" class="form-control" id="exampleInputEmail1"
105                   aria-describedby="emailHelp">
106               </div>
107               <button type="submit" class="btn btn-primary">Simpan</button>
108             </form>
109           </div>
110         </div>
111       </div>
112     </div>
113   </div>
```

Gambar 4 tambahdata.blade.php

Kode HTML milik **tambahdata.blade.php** diatas merupakan kode yang nantinya akan menampilkan halaman form untuk pengguna dalam menambahkan pegawai.

▪ Edit Data

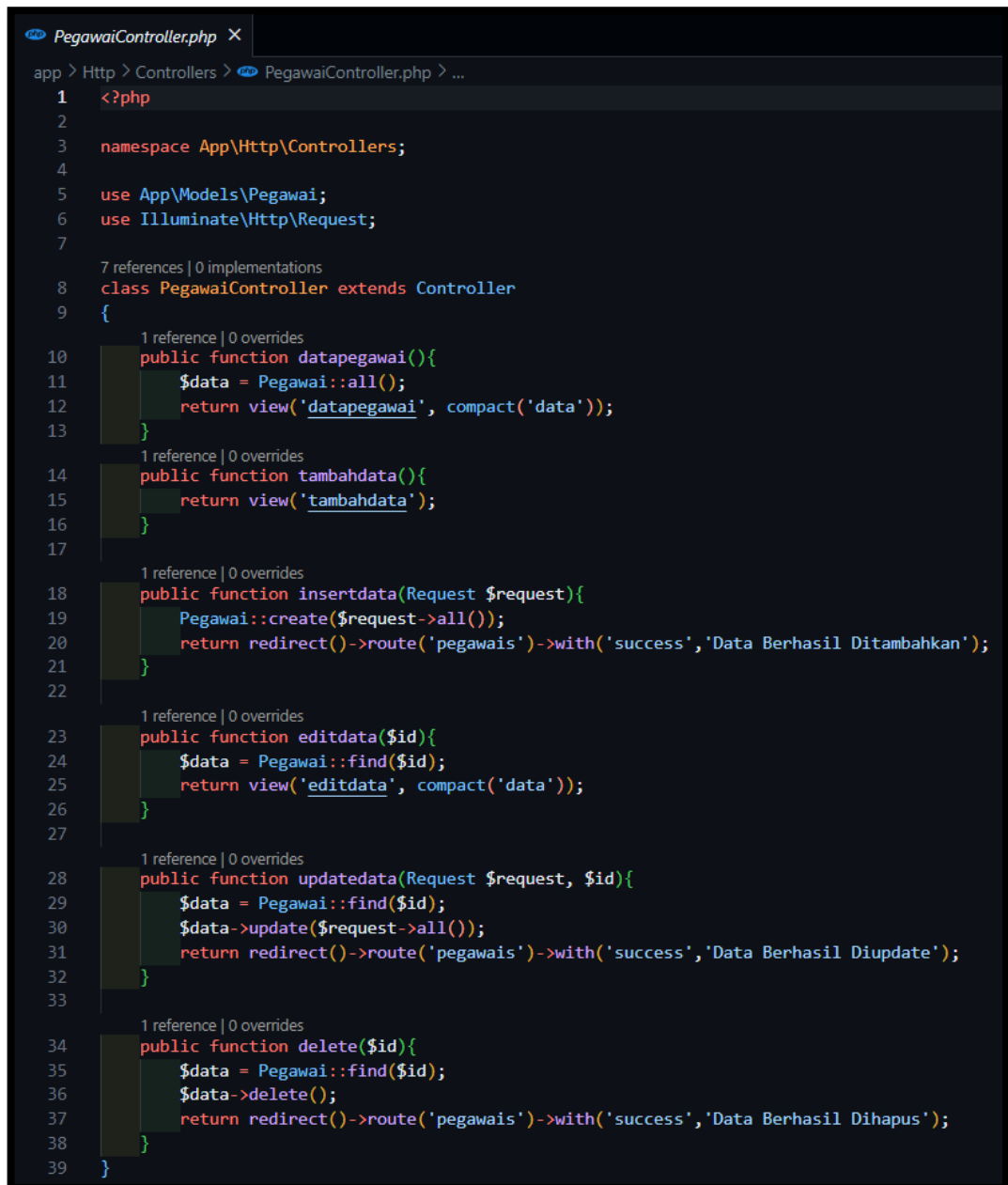


```
resources > views > editdata.blade.php > html > body > div.container > div.row.justify-content-center > div.col-6 > div.card > div.card-body
2  <html lang="en">
73 <body>
74   <h1 class="text-center mb-4">Update Data Pegawai</h1>
75   <div class="container">
76     <a href="{{ route('pegawais') }}" type="button" class="btn btn-outline-success">Home</a>
77     <div class="row justify-content-center">
78       <div class="col-6">
79         <div class="card">
80           <div class="card-body">
81             <form action="/updatedata/{{ $data->id }}" method="POST" enctype="multipart/form-data">
82               @csrf
83               <div class="mb-3">
84                 <label for="exampleInputEmail1" class="form-label">Nama Pegawai</label>
85                 <input type="text" name="nama" class="form-control" id="exampleInputEmail1"
86                   aria-describedby="emailHelp" value="{{ $data->nama }}">
87               </div>
88               <div class="mb-3">
89                 <label for="exampleInputEmail1" class="form-label">Jenis Kelamin</label>
90                 <select class="form-select" aria-label="Default select example" name="jenis_kelamin">
91                   <option selected>Pilih Jenis Kelamin</option>
92                   <option value="1">Laki - laki</option>
93                   <option value="2">Perempuan</option>
94                 </select>
95               </div>
96               <div class="mb-3">
97                 <label for="exampleInputEmail1" class="form-label">Nomor Hp</label>
98                 <input type="text" name="no_hp" class="form-control" id="exampleInputEmail1"
99                   aria-describedby="emailHelp" value="{{ $data->no_hp }}">
100             </div>
101             <div class="mb-3">
102                 <label for="exampleInputEmail1" class="form-label">Alamat</label>
103                 <input type="text" name="alamat" class="form-control" id="exampleInputEmail1"
104                   aria-describedby="emailHelp" value="{{ $data->alamat }}">
105             </div>
106             <button type="submit" class="btn btn-primary">Update</button>
107           </form>
108         </div>
74   </div>
```

Gambar 5 editdata.blade.php

Kode HTML milik **editdata.blade.php** diatas merupakan kode yang nantinya akan menampilkan halaman form untuk pengguna dalam mengedit pegawai.

3. Controller




```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Pegawai;
6 use Illuminate\Http\Request;
7
8 class PegawaiController extends Controller
9 {
10     public function datapegawai(){
11         $data = Pegawai::all();
12         return view('datapegawai', compact('data'));
13     }
14     public function tambahdata(){
15         return view('tambahdata');
16     }
17
18     public function insertdata(Request $request){
19         Pegawai::create($request->all());
20         return redirect()->route('pegawais')->with('success','Data Berhasil Ditambahkan');
21     }
22
23     public function editdata($id){
24         $data = Pegawai::find($id);
25         return view('editdata', compact('data'));
26     }
27
28     public function updatedata(Request $request, $id){
29         $data = Pegawai::find($id);
30         $data->update($request->all());
31         return redirect()->route('pegawais')->with('success','Data Berhasil Diupdate');
32     }
33
34     public function delete($id){
35         $data = Pegawai::find($id);
36         $data->delete();
37         return redirect()->route('pegawais')->with('success','Data Berhasil Dihapus');
38     }
39 }
```

Gambar 6 PegawaiController.php

Kode ini merupakan sebuah Controller Laravel bernama **PegawaiController** yang bertanggung jawab untuk menangani logika terkait **pegawai**. Controller ini memiliki beberapa metode yang memungkinkan pengguna untuk melihat daftar pegawai (**datapegawai**), menambahkan pegawai baru (**tambahdata** dan **insertdata**), mengedit data pegawai (**editdata** dan **updatedata**), serta menghapus data pegawai (**delete**). Metode-metode ini menggunakan model Pegawai untuk berinteraksi dengan data di database, dan mengembalikan tampilan yang sesuai atau mengarahkan pengguna ke rute yang tepat. Secara keseluruhan, PegawaiController menyediakan

fungsionalitas CRUD (Create, Read, Update, Delete) yang dibutuhkan untuk mengelola data pegawai dalam aplikasi Laravel.

4. Route



```
web.php x
routes > web.php > ...
1  <?php
2
3  use App\Http\Controllers\PegawaiController;
4  use Illuminate\Support\Facades\Route;
5
6  // Halaman Welcome
7  Route::get('/', function () {
8      return view('welcome');
9  });
10
11 // Halaman Tabel
12 Route::get('/pegawai', [PegawaiController::class, 'datapegawai'])->name('pegawais');
13
14 // Halaman Tambah Data
15 Route::get('/tambahdata', [PegawaiController::class, 'tambahdata'])->name('tambahdata');
16 Route::post('/insertdata', [PegawaiController::class, 'insertdata'])->name('insertdata');
17
18 // Halaman Edit Data
19 Route::get('editdata/{id}', [PegawaiController::class, 'editdata'])->name('editdata');
20 Route::post('/updatedata/{id}', [PegawaiController::class, 'updatedata'])->name('updatedata');
21
22 // Hapus Data
23 Route::get('/delete/{id}', [PegawaiController::class, 'delete'])->name('delete');
```

Gambar 7 web.php

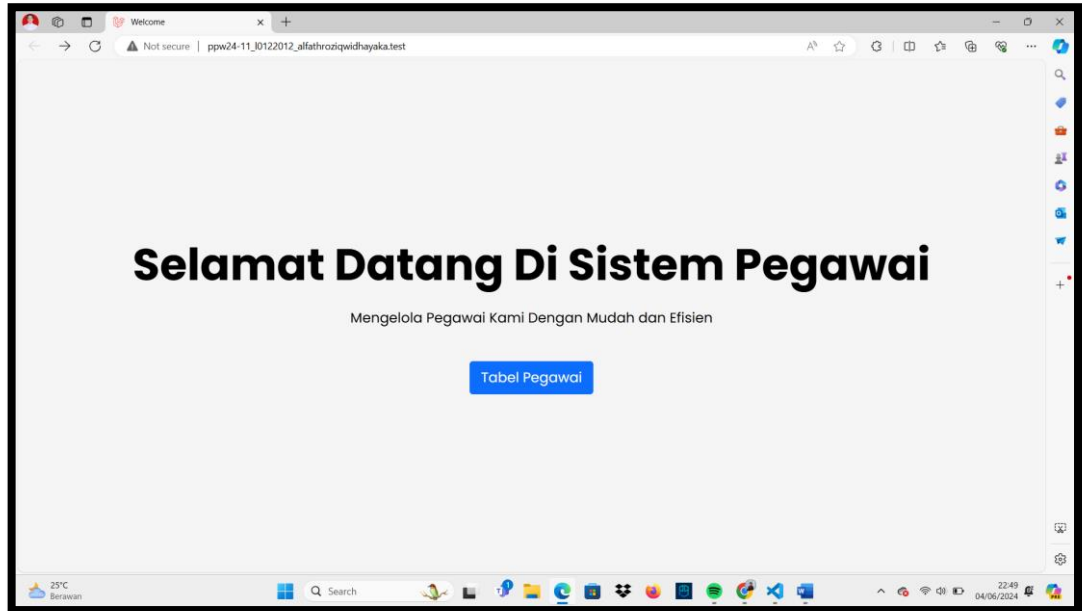
Kode ini merupakan definisi rute (**route**) dalam aplikasi Laravel. Rute-rute ini mengatur bagaimana aplikasi akan merespons permintaan HTTP dari pengguna. Pertama, terdapat rute untuk halaman welcome yang menampilkan tampilan awal aplikasi. Kemudian, ada rute untuk menampilkan daftar pegawai yang ditangani oleh metode **datapegawai** pada **PegawaiController**. Selanjutnya, ada rute untuk menampilkan halaman tambah data pegawai, yang terdiri dari dua rute yaitu satu untuk menampilkan formulirnya dan satu lagi untuk memproses data yang dikirimkan. Selain itu, terdapat rute untuk menampilkan halaman edit data pegawai, yang juga terdiri dari dua rute yaitu satu untuk menampilkan formulirnya dan satu lagi untuk memproses pembaruan data. Terakhir, ada rute untuk menghapus data pegawai. Keseluruhan rute-rute ini menggunakan nama (**name**) untuk memudahkan pengaksesan dan pengiriman URL.

BAB II

A. Hasil Output

Berikut merupakan hasil output dari project laravel yang saya buat menerapkan materi MVC Route.

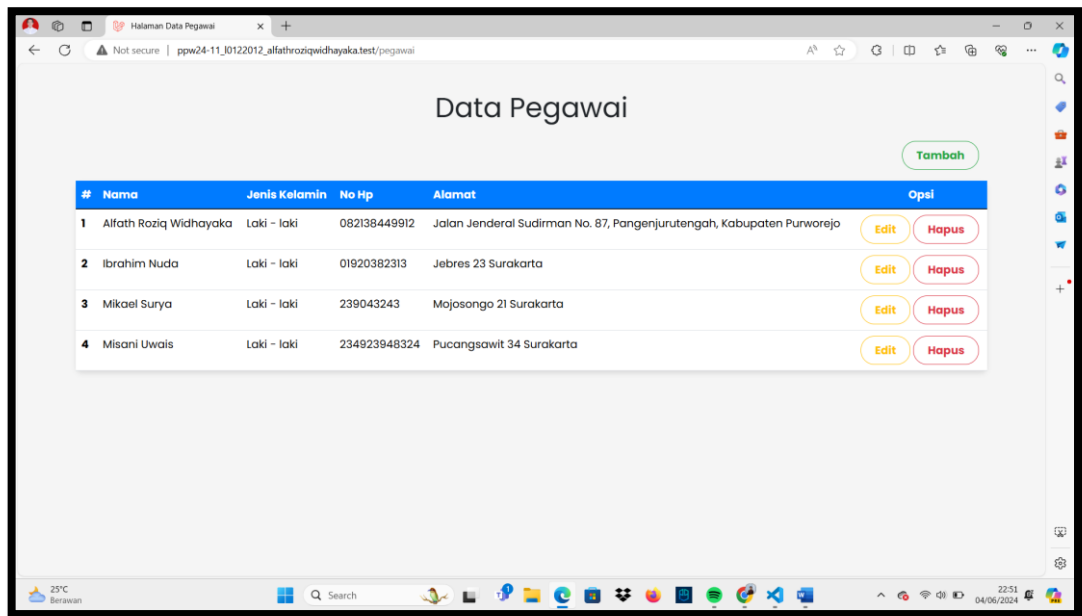
1. Welcome



Gambar 8 welcome

Gambar diatas merupakan hasil output ketika project laravel dijalankan. Terlihat ini merupakan halaman welcome yang menampilkan judul dan satu tombol yang nantinya akan mengarah ke tabel data pegawai.

2. Tabel Data



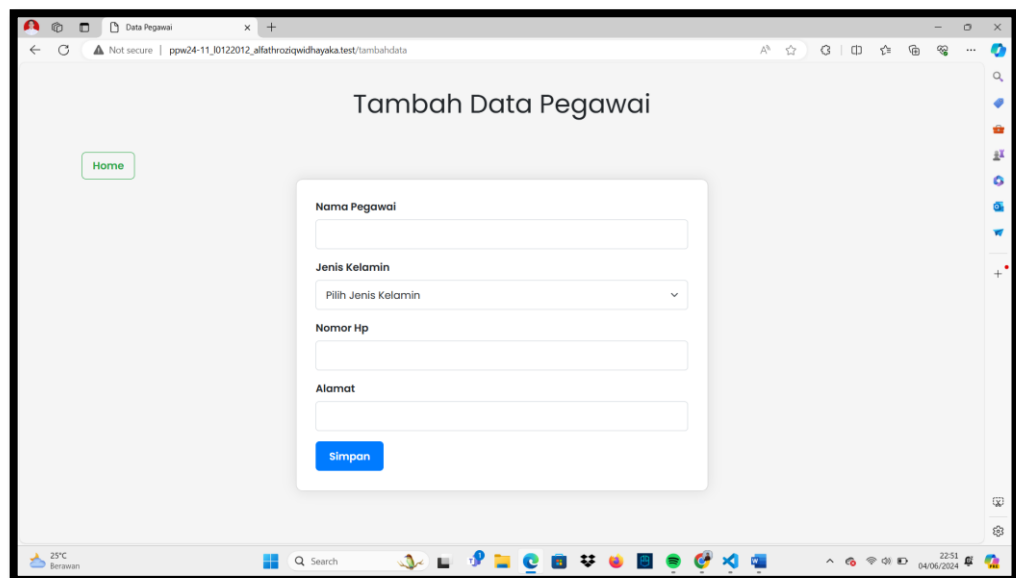
The screenshot shows a web browser window with the URL `ppw24-11_j0122012_alfathroziqwidhayaka.test/pegawai`. The page title is "Data Pegawai". In the top right corner, there is a green button labeled "Tambah". Below it is a table with the following data:

#	Nama	Jenis Kelamin	No Hp	Alamat	Opsl
1	Alfath Roziq Widhayaka	Laki - laki	082138449912	Jalan Jenderal Sudirman No. 87, Pangenjurutengah, Kabupaten Purworejo	<button>Edit</button> <button>Hapus</button>
2	Ibrahim Nuda	Laki - laki	01920382313	Jebres 23 Surakarta	<button>Edit</button> <button>Hapus</button>
3	Mikael Surya	Laki - laki	239043243	Mojosongo 21 Surakarta	<button>Edit</button> <button>Hapus</button>
4	Misani Uwais	Laki - laki	234923948324	Pucangsawit 34 Surakarta	<button>Edit</button> <button>Hapus</button>

Gambar 8 tabel data

Gambar diatas merupakan hasil output ketika pengguna telah menekan tombol **Tabel Pegawai** pada halaman welcome. Disini terlihat halaman menampilkan tabel dengan isinya yaitu daftar pegawai yang dapat ditambah, diedit, dan dihapus.

3. Tambah Data



The screenshot shows a web browser window with the URL `ppw24-11_j0122012_alfathroziqwidhayaka.test/tambahdata`. The page title is "Tambah Data Pegawai". In the top left corner, there is a green button labeled "Home". The main content is a form with the following fields:

- Nama Pegawai:
- Jenis Kelamin:
- Nomor Hp:
- Alamat:
- Simpan:

Gambar 9 tambah data

Jika pengguna menekan tombol **Tambah** pada halaman tabel data, maka gambar diatas merupakan halaman yang diberikan kepada pengguna untuk menambahkan data pegawai yang ingin ditambah

4. Edit Data

The screenshot displays a web browser window with the title 'Update Data Pegawai'. The address bar shows a local test URL. The main content area features a form with the following fields: 'Nama Pegawai' containing 'Alfath Roziq Widhayaka', 'Jenis Kelamin' with a dropdown menu showing 'Pilih Jenis Kelamin', 'Nomor Hp' containing '082138449912', and 'Alamat' containing 'Jalan Jenderal Sudirman No. 87, Pangenjurutengah, Kabupaten Purv'. A blue 'Update' button is positioned below the address field. To the left of the form is a green 'Home' button. The browser's taskbar at the bottom shows the system clock as 22:52 on 04/06/2024.

Gambar 10 edit data

Ketika pengguna memilih tombol **Edit**, maka gambar diatas merupakan tampilan yang diberikan untuk pengguna mengubah atau memodifikasi isi data pada tabel.

BAB III

B. Kesimpulan

Untuk praktikum minggu ke 11 ini, saya telah membuat project Laravel yang mengimplementasikan **MVC Route**, yang mencakup sistem data pegawai sederhana dengan fungsionalitas **CRUD**. Proyek ini terdiri dari kelas **Model** untuk berinteraksi dengan database, **View** untuk menampilkan data, **Controller** untuk menangani logika website, dan **Route** untuk menentukan kemana arah jika salah satu tombol ditekan. Output dari project ini meliputi halaman welcome, tabel yang menampilkan data pegawai, dan formulir untuk menambah dan mengedit data pegawai.