

**PENGEMBANGAN APLIKASI WEBSITE**

**MODUL 14**

**Laravel Database Tahap Lanjut  
(Eloquent ORM & Blade Templates)**



**Sistem Informasi  
Telkom University  
Surabaya**

Disusun Oleh:

Dhini Julia Ningrum (1204230058)

**PROGRAM STUDI S1 SISTEM INFORMASI**

**FAKULTAS REKAYASA INDUSTRI**

**TELKOM UNIVERSITY SURABAYA**

**2024**

- 1) Tambahkankan function **employee()** pada Model **Position** untuk mendefinisikan relationship dengan Model **Employee**

```
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Position extends Model
9  {
10     use HasFactory;
11
12     public function employees()
13     {
14         return $this->hasMany(Employee::class);
15     }
16
17 }
18
```

- Tambahkankan function **position()** pada Model **Employee** untuk mendefinisikan relationship dengan Model **Position**

```

app > Models > Employee.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Employee extends Model
9  {
10     use HasFactory;
11
12     public function position()
13     {
14         return $this->belongsTo(Position::class);
15     }
16 }
17

```

## 2) Implementasi Eloquent pada Fitur Menampilkan Data Pegawai

- Buka file **EmployeeController** dan ubah query dengan pendekatan **Eloquent Model** pada function **index()** seperti di bawah ini

```

public function index()
{
    $pageTitle = 'Employee List';

    // ELOQUENT
    $employees = Employee::all();

    return view('employee.index', [
        'pageTitle' => $pageTitle,
        'employees' => $employees
    ]);
}

```

- Pastikan Model Employee terpanggil pada bagian atas file

```

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use App\Models\Employee;

```

- Buka file View **/views/employee/index.blade.php** dan sesuaikan metode penangkapan data pegawai seperti di bawah ini

```
@foreach($employees as $employee)
    <tr>
        <td>{{ $employee->firstname }}</td>
        <td>{{ $employee->lastname }}</td>
        <td>{{ $employee->email }}</td>
        <td>{{ $employee->age }}</td>
        <td>{{ $employee->position->name }}</td>
        <td>
            {{-- ACTIONS SECTION --}}
            <div class="d-flex">
                <a href="{{ route('employees.show', ['employee' => $employee->id]) }}"
                    class="btn btn-outline-dark btn-sm me-2"><i
                        class="bi-person-lines-fill"></i></a>
                <a href="{{ route('employees.edit', ['employee' => $employee->id]) }}"
                    class="btn btn-outline-dark btn-sm me-2"><i class="bi-pencil-square"></i></a>
                <div>
                    <form action="{{ route('employees.destroy', ['employee' => $employee->id]) }}"
                        method="POST">
                        @csrf
                        @method('delete')
                        <button type="submit" class="btn btn-outline-dark btn-sm me-2"><i
                            class="bi-trash"></i></button>
                    </form>
                </div>
            </div>
        </td>
    </tr>
</foreach>
```

### 3) Implementasi Eloquent pada Fitur Form Input Pegawai

- Buka file **EmployeeController** dan ubah query dengan pendekatan **Eloquent Model** pada function **create()** dan **store()** seperti di bawah ini

```
19 {
20     /**
21      *
22      */
23     public function create()
24     {
25         $pageTitle = 'Create Employee';
26         // RAW SQL Query
27         $positions = DB::select('select * from positions');
28
29         return view('employee.create', compact('pageTitle', 'positions'));
30     }
31
32     /**
33      * Store a newly created resource in storage.
34      */
35     public function store(Request $request)
36     {
37         $messages = [
38             'required' => ':Attribute harus diisi.',
39             'email' => 'Isi :attribute dengan format yang benar',
40             'numeric' => 'Isi :attribute dengan angka'
41         ];
42
43         $validator = Validator::make($request->all(), [
44             'firstName' => 'required',
45             'lastName' => 'required',
46             'email' => 'required|email',
47             'age' => 'required|numeric',
48         ], $messages);
49
50         if ($validator->fails()) {
51             return redirect()->back()->withErrors($validator)->withInput();
52         }
53
54         // INSERT QUERY
55         DB::table('employees')->insert([
56             'firstName' => $request->firstName,
57             'lastName' => $request->lastName,
58             'email' => $request->email,
59             'age' => $request->age,
60             'position_id' => $request->position_id
61         ]);
62     }
63 }
```

- Pastikan Model Position terpanggil pada bagian atas file

```

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use App\Models\Employee;
use App\Models\Position;

```

- Buka file View **/views/employee/create.blade.php** dan sesuaikan metode penangkapan data pegawai seperti di bawah ini

```

<form action="{{ route('employees.store') }}" method="POST">
    @csrf
    <div class="row justify-content-center">
        <div class="p-5 bg-light rounded-3 border col-xl-6">
            <div class="mb-3 text-center">
                <i class="bi-person-circle fs-1"></i>
                <h4>Create Employee</h4>
            </div>
            <hr>
            <div class="row">
                <div class="col-md-6 mb-3">
                    <label for="firstName" class="form-label">First Name</label>
                    <input class="form-control @error('firstName') is-invalid @enderror" type="text" name="firstName" id="firstName" value="{{ old('firstName') }}">
                    @error('firstName')
                        <div class="text-danger"><small>{{ $message }}</small></div>
                    @enderror
                </div>
                <div class="col-md-6 mb-3">
                    <label for="lastName" class="form-label">Last Name</label>
                    <input class="form-control @error('lastName') is-invalid @enderror" type="text" name="lastName" id="lastName" value="{{ old('lastName') }}">
                    @error('lastName')
                        <div class="text-danger"><small>{{ $message }}</small></div>
                    @enderror
                </div>
                <div class="col-md-6 mb-3">
                    <label for="email" class="form-label">Email</label>
                    <input class="form-control @error('email') is-invalid @enderror" type="text" name="email" id="email" value="{{ old('email') }}">
                    @error('email')
                        <div class="text-danger"><small>{{ $message }}</small></div>
                    @enderror
                </div>
                <div class="col-md-6 mb-3">
                    <label for="age" class="form-label">Age</label>
                    <input class="form-control @error('age') is-invalid @enderror" type="text" name="age" id="age" value="{{ old('age') }}">
                    @error('age')

```

```

                        <div class="text-danger"><small>{{ $message }}</small></div>
                    @enderror
                </div>
            </div>
            <div class="col-md-6 mb-3">
                <label for="age" class="form-label">Age</label>
                <input class="form-control @error('age') is-invalid @enderror" type="text" name="age" id="age" value="{{ old('age') }}">
                @error('age')
                    <div class="text-danger"><small>{{ $message }}</small></div>
                @enderror
            </div>
            <div class="col-md-12 mb-3">
                <label for="position" class="form-label">Position</label>
                <select name="position" id="position" class="form-select">
                    @foreach ($positions as $position)
                        <option value="{{ $position->id }}" {{ old('position') == $position->id ? 'selected' : '' }}>{{ $position->code.
                    @endforeach
                </select>
                @error('position')
                    <div class="text-danger"><small>{{ $message }}</small></div>
                @enderror
            </div>
            <hr>
            <div>
                <div>
                    <div>
                        <div>
                            <div>
                                The div element has no special meaning at all. It represents its children. It can be used with the class, lang, and title attributes to mark up
                                semantics common to a group of consecutive elements.
                            </div>
                            <div class="col-md-6 d-grid">
                                <button type="submit" class="btn btn-dark btn-lg mt-3"><i class="bi-check-circle me-2"></i> Save</button>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</form>

```

#### 4) Implementasi Eloquent pada Fitur Detail Data Pegawai

- Buka file **EmployeeController** dan ubah query dengan pendekatan **Eloquent Model** pada function **show()** seperti di bawah ini

```
public function show(string $id)
{
    $pageTitle = 'Employee Detail';

    // ELOQUENT
    $employee = Employee::find($id);

    return view('employee.show', compact('pageTitle', 'employee'));
}
```

- Buka file View **/views/employee/show.blade.php** dan sesuaikan metode penangkapan data pegawai seperti di bawah ini

```
<div class="container-sm my-5">
    <div class="row justify-content-center">
        <div class="p-5 bg-light rounded-3 col-xl-4 border">
            <div class="mb-3 text-center">
                <i class="bi-person-circle fs-1"></i>
                <h4>Detail Employee</h4>
            </div>
            <hr>
            <div class="row">
                <div class="col-md-12 mb-3">
                    <label for="firstName" class="form-label">First Name</label>
                    <h5>{{ $employee->firstname }}</h5>
                </div>
                <div class="col-md-12 mb-3">
                    <label for="lastName" class="form-label">Last Name</label>
                    <h5>{{ $employee->lastname }}</h5>
                </div>
                <div class="col-md-12 mb-3">
                    <label for="email" class="form-label">Email</label>
                    <h5>{{ $employee->email }}</h5>
                </div>
                <div class="col-md-12 mb-3">
                    <label for="age" class="form-label">Age</label>
                    <h5>{{ $employee->age }}</h5>
                </div>
                <div class="col-md-12 mb-3">
                    <label for="age" class="form-label">Position</label>
                    <h5>{{ $employee->position->name }}</h5>
                </div>
            </div>
            <hr>
            <div class="row">
                <div class="col-md-12 d-grid">
                    <a href="{{ route('employees.index') }}" class="btn btn-outline-dark btn-lg mt-3"><i class="bi-arrow-left-circle me-2"></i>
                </div>
            </div>
        </div>
    </div>
</div>
```

#### 5) Implementasi Eloquent pada Fitur Form Edit Pegawai

- Buka file **EmployeeController** dan ubah query dengan pendekatan **Eloquent Model** pada function **edit()** dan **update()** seperti di bawah ini



```

/**
 * Show the form for editing the specified resource.
 */
public function edit(string $id)
{
    $pageTitle = 'Edit Employee';

    // ELOQUENT
    $positions = Position::all();
    $employee = Employee::find($id);

    return view('employee.edit', compact('pageTitle', 'positions', 'employee'));
}

```

```

public function update(Request $request, string $id)
{
    $messages = [
        'required' => ':Attribute harus diisi.',
        'email' => 'Isi :attribute dengan format yang benar',
        'numeric' => 'Isi :attribute dengan angka'
    ];

    $validator = Validator::make($request->all(), [
        'firstName' => 'required',
        'lastName' => 'required',
        'email' => 'required|email',
        'age' => 'required|numeric',
    ], $messages);

    if ($validator->fails()) {
        return redirect()->back()->withErrors($validator)->withInput();
    }

    // ELOQUENT
    $employee = Employee::find($id);
    $employee->firstname = $request->firstName;
    $employee->lastname = $request->lastName;
    $employee->email = $request->email;
    $employee->age = $request->age;
    $employee->position_id = $request->position;
    $employee->save();

    return redirect()->route('employees.index');
}

```

- Buka file View **/views/employee/edit.blade.php** dan sesuaikan metode penangkapan data pegawai seperti di bawah ini

```

3 @section('content')
4 {div class="container-sm my-5">
5     <form action="{{ route('employees.update', ['employee' => $employee->id]) }}" method="POST">
6         @csrf
7         @method('put')
8         <div class="row justify-content-center">
9             <div class="p-5 bg-light rounded-3 col-xl-6">
10                 <div class="mb-3 text-center">
11                     <i class="bi person-circle fs-1"></i>
12                     <h4>Edit Employee</h4>
13                 </div>
14                 <hr>
15                 <div class="row">
16                     <div class="col-md-6 mb-3">
17                         <label for="firstName" class="form-label">First Name</label>
18                         <input class="form-control @error('firstName') is-invalid @enderror" type="text" name="firstName" id="firstName" value="{{ $
19                             @error('firstName')
20                             <div class="text-danger"><small>{{ $message }}</small></div>
21                             @enderror
22                         </div>
23                         <div class="col-md-6 mb-3">
24                             <label for="lastName" class="form-label">Last Name</label>
25                             <input class="form-control @error('lastName') is-invalid @enderror" type="text" name="lastName" id="lastName" value="{{ $err
26                                 @error('lastName')
27                                 <div class="text-danger"><small>{{ $message }}</small></div>
28                                 @enderror
29                             </div>
30                             <div class="col-md-6 mb-3">
31                                 <label for="email" class="form-label">Email</label>
32                                 <input class="form-control @error('email') is-invalid @enderror" type="text" name="email" id="email" value="{{ $errors->any(
33                                     @error('email')
34                                     <div class="text-danger"><small>{{ $message }}</small></div>
35                                     @enderror
36                                 </div>
37                                 <div class="col-md-6 mb-3">
38                                     <label for="age" class="form-label">Age</label>
39                                     <input class="form-control @error('age') is-invalid @enderror" type="text" name="age" id="age" value="{{ $errors->any() ? ol
40                                         @error('age')
41                                         <div class="text-danger"><small>{{ $message }}</small></div>

```

## 6) Implementasi Eloquent pada Fitur Hapus Data Pegawai

- Buka file **EmployeeController** dan ubah query dengan pendekatan **Eloquent Model** pada function **destroy()** seperti di bawah ini

```

public function destroy(string $id)
{
    // ELOQUENT
    Employee::find($id)->delete();

    return redirect()->route('employees.index');
}

```

## 7) Memisahkan Nav Section

- Buat file **View baru** pada **/views/layouts/nav.blade.php** kemudian pindahkan kode program nav section (yang berulang dipakai di file-file View sebelumnya) pada file baru tersebut. Kemudian sesuaikan seperti kode program di bawah ini



```

1  @php
2  | $currentRouteName = Route::currentRouteName();
3  @endphp
4
5  <nav class="navbar navbar-expand-md navbar-dark bg-primary">
6  |   <div class="container">
7  |     <a href="{{ route('home') }}" class="navbar-brand mb-0 h1"><i class="bi-hexagon-fill me-2"></i> Data Master</a>
8  |
9  |     <button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent">
10 |       <span class="navbar-toggler-icon"></span>
11 |     </button>
12 |
13 |     <div class="collapse navbar-collapse" id="navbarSupportedContent">
14 |       <hr class="d-md-none text-white-50">
15 |
16 |       <ul class="navbar-nav flex-row flex-wrap">
17 |         <li class="nav-item col-2 col-md-auto"><a href="{{ route('home') }}" class="nav-link @if($currentRouteName == 'home') active @endif"
18 |           <li class="nav-item col-2 col-md-auto"><a href="{{ route('employees.index') }}" class="nav-link @if($currentRouteName == 'employees
19 |
20 |       </ul>
21 |
22 |       <hr class="d-md-none text-white-50">
23 |
24 |       <a href="{{ route('profile') }}" class="btn btn-outline-light my-2 ms-md-auto"><i class="bi-person-circle me-1"></i> My Profile</a>
25 |     </div>
26 |   </div>
27 </nav>
28

```

## 8) Mendefinisikan Layout Utama Aplikasi

- Buat file **View baru** pada **/views/layouts/app.blade.php** yang akan kita jadikan template utama pada aplikasi web kita. Sesuaikan kode program nya seperti di bawah ini

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6  |   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  |   <title>{{ $pageTitle }}</title>
8  |   @vite('resources/sass/app.scss')
9  </head>
10 <body>
11 |   @include('layouts.nav')
12 |   @yield('content')
13 |   @vite('resources/js/app.js')
14 </body>
15 </html>
16

```

## 9) Membuat Template untuk Default Content

- Buat file **View baru** pada **/views/default.blade.php** yang akan kita jadikan template default untuk konten di aplikasi web kita. Sesuaikan kode program nya seperti di bawah ini

```

1 <div class="container mt-4">
2   <h4>{{ $pageTitle }}</h4>
3   <hr>
4   <div class="d-flex align-items-center py-2 px-4 bg-light rounded-3 border">
5     <div class="bi-house-fill me-3 fs-1"></div>
6     <h4 class="mb-0">Well done! this is {{ $pageTitle }}.</h4>
7   </div>
8 </div>
9
10

```

#### 10) Memisahkan Actions Section

- Buat file **View baru** pada **/views/employee/actions.blade.php** kemudian pindahkan kode program **actions section** pada file **/views/employee/index.blade.php** pada file **actions.blade.php** tersebut. Kemudian sesuaikan kode program seperti di bawah ini

```

1 <div class="d-flex">
2   <a href="{{ route('employees.show', ['employee' => $employee->id]) }}" class="btn btn-outline-dark btn-sm me-2"><i class="bi-person-lines-fill">
3   <a href="{{ route('employees.edit', ['employee' => $employee->id]) }}" class="btn btn-outline-dark btn-sm me-2"><i class="bi-pencil-square"></i>
4
5   <div>
6     <form action="{{ route('employees.destroy', ['employee' => $employee->id]) }}" method="POST">
7       @csrf
8       @method('delete')
9       <button type="submit" class="btn btn-outline-dark btn-sm me-2"><i class="bi-trash"></i></button>
10    </form>
11  </div>
12 </div>
13

```

#### 11) Penyesuaian File View untuk Fitur Menampilkan Data Pegawai

- Buka file **View** pada **/views/employee/index.blade.php** kemudian implementasikan konsep inheritance dengan Blade Directive **@extends()** file layout utama (**/layouts/app.blade.php**), implementasi dynamic content dengan mendefinisikan Blade Directive **@section()** dan memanggilkan file yang berisi kode program untuk actions section dengan Blade Directive **@include()**

```

1 @extends('layouts.app')
2
3 @section('content')
4   <div class="container mt-4">
5     <div class="row mb-0">

```

```

61         </div>
62     </div>
63 </div>
64 @endsection
65     @vite('resources/js/app.js')
66 </body>
67
68 </html>

```

## 12) Penyesuaian File View untuk Fitur Form Input Pegawai

- Buka file **View** pada **/views/employee/create.blade.php** kemudian sesuaikan kode program nya seperti di bawah ini

```

1  @extends('layouts.app')
2
3  @section('content')
4      <div class="container-sm mt-5">

```

```

        </form>
    </div>
@endsection
    @vite('resources/js/app.js')
</body>
</html>

```

## 13) Penyesuaian File View untuk Fitur Detail Data Pegawai

- Buka file **View** pada **/views/employee/show.blade.php** kemudian sesuaikan kode program nya seperti di bawah ini

```

1  @extends('layouts.app')
2
3  @section('content')
4      <div class="container-sm my-5">
5          <div class="row justify-content-center">

```

```

        </div>
    </div>
</div>
</div>
@endsection

```

#### 14) Penyesuaian File View untuk Fitur Form Edit Pegawai

- Buka file **View** pada **/views/employee/edit.blade.php** kemudian sesuaikan kode program nya seperti di bawah ini

```

1  @extends('layouts.app')
2
3  @section('content')
4  <div class="container-sm my-5">

```

```

        </div>
    </div>
</form>
</div>
@endsection

```

#### 15) Penyesuaian File View untuk Halaman Home dan Profile

- Buka file **View** pada **/views/home.blade.php** dan **/views/profile.blade.php** kemudian sesuaikan kode program nya seperti di bawah ini

```

@extends('layouts.app')

@section('content')
    @include('default')
@endsection

```