

# **PRAKTIKUM PEMROGRAMAN FRAMEWORK**

## **MODUL 6 Laravel Database Tahap Lanjut (Eloquent ORM & Blade Templates)**



Disusun Oleh:  
Purnama Anaking, S.Kom., M.Kom.

**PROGRAM STUDI S1 SISTEM INFORMASI  
FAKULTAS TEKNOLOGI INFORMASI DAN BISNIS  
INSTITUT TEKNOLOGI TELKOM SURABAYA  
2023**

# DAFTAR ISI

DAFTAR ISI	2
1. Membuat File Model	3
2. Implementasi Eloquent pada Fitur Menampilkan Data Pegawai	4
3. Implementasi Eloquent pada Fitur Form Input Pegawai	5
4. Implementasi Eloquent pada Fitur Detail Data Pegawai	8
5. Implementasi Eloquent pada Fitur Form Edit Pegawai	9
6. Implementasi Eloquent pada Fitur Hapus Data Pegawai	12
7. Memisahkan Nav Section	12
8. Mendefinisikan Layout Utama Aplikasi	13
9. Membuat Template untuk Default Content	14
10. Memisahkan Actions Section	14
11. Penyesuaian File View untuk Fitur Menampilkan Data Pegawai	15
12. Penyesuaian File View untuk Fitur Form Input Pegawai	16
13. Penyesuaian File View untuk Fitur Detail Data Pegawai	16
14. Penyesuaian File View untuk Fitur Form Edit Pegawai	17
15. Penyesuaian File View untuk Halaman Home dan Profile	18
16. Tugas	18

## MODUL 6

# LARAVEL DATABASE TAHAP LANJUT

### (Eloquent ORM & Blade Templates)

Pada praktikum kali ini kita akan melanjutkan belajar tentang laravel database. Materi tentang laravel database akan diilustrasikan dalam bentuk studi kasus sederhana secara bertahap. Pada praktikum kali ini kita akan fokus menerapkan Eloquent ORM pada project laravel kita dan juga melengkapi pembahasan dan penerapan blade templates pada project kita. Kegiatan ini agar mahasiswa dapat memahami dan menerapkan manajemen database pada laravel framework dan mengoptimalkan fitur yang dimiliki oleh blade templates.

#### 1. Membuat File Model

- Generate file Model untuk tabel **positions** dan **employees** via Artisan

```
php artisan make:model Position
```

```
php artisan make:model Employee
```

- Tambahkan function **employee()** pada Model **Position** untuk mendefinisikan relationship dengan Model **Employee**.

```
<?php

namespace App\Models;

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

class Position extends Model
{
    use HasFactory;

    public function employees()
    {
        return $this->hasMany(Employee::class);
    }
}
```

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

```
<?php

namespace App\Models;
```

```

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

class Employee extends Model
{
    use HasFactory;

    public function position()
    {
        return $this->belongsTo(Position::class);
    }
}

```

## 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 App\Models\Employee;

```

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

```

<tbody>
    @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">

```



```

        'email' => 'required|email',
        'age' => 'required|numeric',
    ], $messages);

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

    // ELOQUENT
    $employee = New Employee;
    $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');
}

```

- Pastikan Model Position terpanggil pada bagian atas file.

```
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') }}" placeholder="Enter First Name">
                    @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') }}" placeholder="Enter Last Name">
        @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') }}"
placeholder="Enter 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') }}"
placeholder="Enter 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' : '' }}>{{
$position->code.' - '.$position->name }}</option>
            @endforeach
        </select>
        @error('position')
            <div class="text-danger"><small>{{ $message
}}</small></div>
        @enderror
    </div>
</div>
<hr>
<div class="row">
    <div class="col-md-6 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>

```

```

Cancel</a>
    </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>
</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> Back</a>
    </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.

```

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.

```

<form action="{{ route('employees.update', ['employee' => $employee->id]) }}" method="POST">
    @csrf
    @method('put')
    <div class="row justify-content-center">
        <div class="p-5 bg-light rounded-3 col-xl-6">
            <div class="mb-3 text-center">
                <i class="bi-person-circle fs-1"></i>
                <h4>Edit 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="{{ $errors->any() ? old('firstName') :
$employee->firstname }}" placeholder="Enter First Name">
                    @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="{{
$errors->any() ? old('lastName') : $employee->lastname }}"
placeholder="Enter Last Name">
        @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="{{
$errors->any() ? old('email') : $employee->email }}" placeholder="Enter
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="{{
$errors->any() ? old('age') : $employee->age }}" placeholder="Enter 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">
            @php
                $selected = "";
                if ($errors->any())
                    $selected = old('position');
                else
                    $selected =
$employee->position_id;
            @endphp
            @foreach ($positions as $position)

```

```

                                <option value="{{ $position->id }}"
{{ $selected == $position->id ? 'selected' : '' }}>{{ $position->code.' -
'.$position->name }}</option>
                                @endforeach
                                </select>
                                @error('position')
                                <div class="text-danger"><small>{{
$message }}</small></div>
                                @enderror
                                </div>
                                </div>
                                <hr>
                                <div class="row">
                                    <div class="col-md-6 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> Cancel</a>
                                    </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> Edit</button>
                                    </div>
                                </div>
                                </div>
                                </div>
                                </form>

```

## 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.

```

@php
$currentRouteName = Route::currentRouteName();

```

```

@endphp

<nav class="navbar navbar-expand-md navbar-dark bg-primary">
    <div class="container">
        <a href="{{ route('home') }}" class="navbar-brand mb-0 h1"><i
class="bi-hexagon-fill me-2"></i> Data Master</a>

        <button type="button" class="navbar-toggler"
data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent">
            <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse"
id="navbarSupportedContent">
            <hr class="d-md-none text-white-50">

            <ul class="navbar-nav flex-row flex-wrap">
                <li class="nav-item col-2 col-md-auto"><a href="{{
route('home') }}" class="nav-link @if($currentRouteName == 'home') active
@endif">Home</a></li>
                <li class="nav-item col-2 col-md-auto"><a href="{{
route('employees.index') }}" class="nav-link @if($currentRouteName ==
'employees.index') active @endif">Employee</a></li>
            </ul>

            <hr class="d-md-none text-white-50">

            <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>
        </div>
    </div>
</nav>

```

## 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.

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>{{ $pageTitle }}</title>
    @vite('resources/sass/app.scss')
</head>

```

```

<body>
    @include('layouts.nav')
    @yield('content')
    @vite('resources/js/app.js')
</body>
</html>

```

## 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.

```

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

```

## 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.

```

<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>

```

## 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 memanggil file yang berisi kode program untuk actions section dengan Blade Directive `@include()`.

```
@extends('layouts.app')

@section('content')
    <div class="container mt-4">
        <div class="row mb-0">
            <div class="col-lg-9 col-xl-10">
                <h4 class="mb-3">{{ $pageTitle }}</h4>
            </div>
            <div class="col-lg-3 col-xl-2">
                <div class="d-grid gap-2">
                    <a href="{{ route('employees.create') }}" class="btn btn-primary">Create Employee</a>
                </div>
            </div>
        </div>
        <hr>
        <div class="table-responsive border p-3 rounded-3">
            <table class="table table-bordered table-hover table-striped mb-0 bg-white">
                <thead>
                    <tr>
                        <th>First Name</th>
                        <th>Last Name</th>
                        <th>Email</th>
                        <th>Age</th>
                        <th>Position</th>
                        <th></th>
                    </tr>
                </thead>
                <tbody>
                    @foreach ($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>@include('employee.actions')</td>
                        </tr>
                    @endforeach
                </tbody>
            </table>
        </div>
    </div>
```

```

        </div>
    </div>
@endsection

```

## 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.

```

@extends('layouts.app')

@section('content')
    <div class="container-sm my-5">
        <form action="{{ route('employees.store') }}" method="POST">
            @csrf
            <div class="row justify-content-center">
                {{-- Dan Seterusnya --}}
            </div>
        </form>
    </div>
@endsection

```

## 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.

```

@extends('layouts.app')

@section('content')
    <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> Back</a>
    </div>
</div>
</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.

```

@extends('layouts.app')

@section('content')
    <div class="container-sm my-5">
        <form action="{{ route('employees.update', ['employee' =>
$employee->id]) }}" method="POST">
            @csrf
            @method('put')
            <div class="row justify-content-center">
                {{-- Dan Seterusnya --}}
            </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
```

## 16. Tugas

1. **Praktekkan** seluruh poin praktikum yang ada di atas secara **INDIVIDU**.
2. Dokumentasikan seluruhnya (**screenshot kode program, output pada browser, jawaban pertanyaan**) dalam bentuk Laporan Praktikum.
3. **Upload manual** project laravel hasil praktikum ke github (hapus folder **node\_modules & vendor**).
  - a. Buat **repository public** baru pada akun github anda.
  - b. Klik link “**Uploading an existing file**” untuk meng-upload project laravel yang telah anda buat.
  - a. Cantumkan **URL repository** yang telah dibuat pada **Laporan Praktikum**.
4. Kumpulkan Laporan Praktikum (**.pdf**) yang telah dibuat di dalam praktikum via **E-Learning** paling lambat sebelum jadwal praktikum minggu depan.