

PRAKTIKUM
Pengembangan Aplikasi Website
LEMBAR KERJA MAHASISWA
Modul 12



oleh:

Johanes Oryza Pradana Agdhiwinata (IS-06-01 - 1204230105)

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS REKAYASA INDUSTRI
TELKOM UNIVERSITY SURABAYA
2024

1. Generate Laravel Project dan Laravel UI

- Buat project laravel baru via composer

```
composer create-project laravel/laravel your-project-name
```

- Masuk ke dalam folder project laravel yang baru saja dibuat, kemudian install package Laravel UI.

```
composer require laravel/ui
```

- Generate scaffolding untuk project Laravel berbasis CSS framework Bootstrap.

```
php artisan ui bootstrap
```

- Jalankan script di bawah ini untuk compile scaffolding Bootstrap yang barusan di-setup.

```
npm install
```

- Jalankan script di bawah ini untuk mengaktifkan local development server Laravel sehingga aplikasi web dapat diakses melalui browser.

```
php artisan serve
```

2. Bundling Asset dengan Vite

- Periksa hasil instalasi Anda. Ketik script ini di prompt perintah Anda:

```
node -v
```

```
npm -v
```

- Install semua dependencies yang dibutuhkan untuk Bundling Asset dengan Vite, dengan mengetikkan perintah:

```
npm install
```

- Terapkan fitur “**Refreshing on Save**” dengan memeriksa file konfigurasi file vite pada **vite.config.js** dan sesuaikan seperti gambar di bawah ini.

```
import { defineConfig } from 'vite';
import laravel from 'laravel-vite-plugin';

export default defineConfig({
  plugins: [
    laravel({
      input: [
        'resources/sass/app.scss',
        'resources/js/app.js',
      ],
      refresh: true,
    }),
  ],
});
```

- Terapkan fitur “**Processing Static Assets With Vite**” dengan buka file **/resources/js/app.js** lalu sesuaikan kode program seperti di bawah ini. Vite akan merujuk pada path direktori yang kita definisikan untuk mengambil aset gambar/image yang kita butuhkan nantinya.

```
import './bootstrap';
import.meta.glob(['../images/**']);
```

- Buat folder pada **/resources** dengan nama **images**. Letakkan aset gambar/image

yang akan kita gunakan pada website kita pada folder tersebut.

- Buka file View bernama **welcome.blade.php**. Hapus seluruh kode program yang ada. Kemudian, letakkan dan arahkan file **css** dan **javascript** yang telah didefinisikan di atas, pada file view (blade) menggunakan Blade Directive **@vite()**. Panggil asset gambar/image dengan pendekatan Vite 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>Belajar Controller & View pada Laravel</title>
  @vite('resources/sass/app.scss')
</head>
<body>
  <div class="container text-center my-5">
    <h1 class="mb-4">Belajar Controller & View pada Laravel</h1>
    { -- Contoh cara mereferensikan gambar di dalam file blade dengan
menggunakan pendekatan Vite -- }
    

    <div class="col-md-2 offset-md-5 mt-4">
      <div class="d-grid gap-2">
        <a class="btn btn-dark" href="">Home</a>
      </div>
    </div>
  </div>
  @vite('resources/js/app.js')
</body>
</html>
```

- Jalankan Vite.
 - Untuk mode **development** (jalankan ini saja jika sedang development):

```
npm run dev
```

- Atau untuk mode **production** (ketika akan deploy ke server):

```
npm run build
```

3. Install Bootstrap dan Bootstrap Icons Terbaru pada Project Laravel

- Jalankan perintah berikut pada project laravel anda untuk menginstall bootstrap, popper dan bootstrap icons terbaru.

```
npm install bootstrap@5.3.0-alpha3 bootstrap-icons @popperjs/core
```

- Di dalam project laravel anda, buka file **resources\sass\app.scss** dan tambahkan:

```
@import "bootstrap-icons/font/bootstrap-icons.css";
```

- Hapus atau Comment kode import Font & import Variables pada file **resources\sass\app.scss**. Kode program akan terlihat seperti di bawah ini:

```
// Fonts
// @import url("https://fonts.bunny.net/css?family=Nunito");
```

```
// Variables
// @import "variables";

// Bootstrap
@import "bootstrap/scss/bootstrap";
@import "bootstrap-icons/font/bootstrap-icons.css";
```

4. Praktik Halaman Home

- Generate sebuah **Basic Controller** dengan nama **HomeController** menggunakan perintah Artisan berikut ini. File Controller akan secara otomatis dibuatkan dan diletakkan pada folder **app/Http/Controllers/**.

```
php artisan make:controller HomeController
```

- Buka file **/routes/web.php** kemudian buat Route dengan kode seperti di bawah ini:

```
Route::get('home', [HomeController::class, 'index'])->name('home');
```

- Pastikan kode di bawah ini ada di bagian atas file **/routes/web.php**.

```
use App\Http\Controllers\HomeController;
```

- Sesuaikan kode program pada file **HomeController** tersebut seperti di bawah ini:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HomeController extends Controller
{
    function index()
    {
        $pageTitle = 'Home';

        return view('home', ['pageTitle' => $pageTitle]);
    }
}
```

- Buat file View dengan nama **home.blade.php** pada folder **resources/views/**. Sesuaikan kode program pada file View tersebut 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>
    <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-lg-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 active">Home</a></li>
    <li class="nav-item col-2 col-md-auto"><a href="{{
route('employees.index') }}" class="nav-link">Employee List</a></li>
    </ul>

    <hr class="d-lg-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>

<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>

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

```

- Pertanyaan:
 - Jelaskan dan dokumentasikan setiap kode program yang anda kerjakan di atas!
 - Buka file View **welcome.blade.php**, arahkan button **Home** ke **Route** dengan nama **"home"** yang telah dibuat pada Bab ini.

5. Praktik Halaman My Profile

- Generate sebuah **Single Action Controller** dengan nama **ProfileController** menggunakan perintah Artisan berikut ini. File Controller akan secara otomatis dibuatkan dan diletakkan pada folder **app/Http/Controllers/**.

```
php artisan make:controller ProfileController --invokable
```

- Buka file **/routes/web.php** kemudian buat Route dengan kode seperti di bawah ini:

```
Route::get('profile', ProfileController::class)->name('profile');
```

- Pastikan kode di bawah ini ada di bagian atas file **/routes/web.php**.

```
use App\Http\Controllers\ProfileController;
```

- Sesuaikan kode program pada file **ProfileController** tersebut seperti di bawah ini:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class ProfileController extends Controller
{
    /**
     * Handle the incoming request.
     */
    public function __invoke(Request $request)
    {
        $pageTitle = 'Profile';

        return view('profile', ['pageTitle' => $pageTitle]);
    }
}
```

- Buat file View dengan nama **profile.blade.php** pada folder **resources/views/**. Sesuaikan kode program pada file View tersebut 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>
    <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-lg-none text-white-50">

                <ul class="navbar-nav flex-row flex-wrap">
                    <li class="nav-item col-6 col-md-auto"><a href="{{ route('home') }}" class="nav-link">Home</a></li>
```

```

        <li class="nav-item col-6 col-md-auto"><a href="{{ route('employees.index') }}" class="nav-link">Employee List</a></li>
    </ul>

    <hr class="d-lg-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>

<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-person-circle me-3 fs-1"></div>
        <h4 class="mb-0">Well done! this is {{ $pageTitle }}.</h4>
    </div>
</div>

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

```

- Pertanyaan:
 - Jelaskan dan dokumentasikan setiap kode program yang anda kerjakan di atas!

6. Praktik Halaman Employee List

- Generate sebuah **Resource Controller** dengan nama **EmployeeController** menggunakan perintah Artisan berikut ini. File Controller akan secara otomatis dibuatkan dan diletakkan pada folder **app/Http/Controllers/**.

```
php artisan make:controller EmployeeController --resource
```

- Buka file **/routes/web.php** kemudian buat Route dengan kode seperti di bawah ini:

```
Route::resource('employees', EmployeeController::class);
```

- Pastikan kode di bawah ini ada di bagian atas file **/routes/web.php**.

```
use App\Http\Controllers\EmployeeController;
```

- Sesuaikan kode program **function index()** pada file **EmployeeController** tersebut seperti di bawah ini:

```

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

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

```

- Buat folder baru dengan nama **/employee** pada **resources/views/**. Kemudian buat file View dengan nama **index.blade.php** di dalam folder tersebut. Sesuaikan kode program pada file View tersebut 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>
  <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-lg-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">Home</a></li>
          <li class="nav-item col-2 col-md-auto"><a href="{{ route('employees.index') }}" class="nav-link active">Employee List</a></li>
        </ul>

        <hr class="d-lg-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>

  <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></th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Purnama</td>
      <td>Anaking</td>
      <td>purnama.anaking@gmail.com</td>
      <td>20</td>
      <td>
        <div class="d-flex">
          <a href="{{ route('employees.show', ['employee' => 1]) }}"
class="btn btn-outline-dark btn-sm me-2"><i class="bi-person-lines-fill"></i></a>
          <a href="{{ route('employees.edit', ['employee' => 1]) }}"
class="btn btn-outline-dark btn-sm me-2"><i class="bi-pencil-square"></i></a>

          <div>
            <form action="{{ route('employees.destroy', ['employee' => 1])
}}" 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>
    <tr>
      <td>Adzanil</td>
      <td>Rachmadhi</td>
      <td>adzanil.rachmadhi@gmail.com</td>
      <td>25</td>
      <td>
        <div class="d-flex">
          <a href="{{ route('employees.show', ['employee' => 2]) }}"
class="btn btn-outline-dark btn-sm me-2"><i class="bi-person-lines-fill"></i></a>
          <a href="{{ route('employees.edit', ['employee' => 2]) }}"
class="btn btn-outline-dark btn-sm me-2"><i class="bi-pencil-square"></i></a>

          <div>
            <form action="{{ route('employees.destroy', ['employee' => 2])
}}" 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>
<tr>
<td>Berlian</td>
<td>Rahmy</td>
<td>berlian.rahmy@gmail.com</td>
<td>23</td>
<td>
    <div class="d-flex">
        <a href="{ { route('employees.show', ['employee' => 3]) } }"
class="btn btn-outline-dark btn-sm me-2"><i class="bi-person-lines-fill"></i></a>
        <a href="{ { route('employees.edit', ['employee' => 3]) } }"
class="btn btn-outline-dark btn-sm me-2"><i class="bi-pencil-square"></i></a>
    </div>
    <form action="{ { route('employees.destroy', ['employee' => 3])
} }" 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>
</tbody>
</table>
</div>
</div>

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

```

- Pertanyaan:
 - Jelaskan dan dokumentasikan setiap kode program yang anda kerjakan di atas!
 - Tunjukkan dan buktikan ada Route apa saja yang dihasilkan pada Bab ini!

7. Praktik Halaman Create Employee

- Sesuaikan kode program **function create()** pada file **EmployeeController** tersebut seperti di bawah ini:

```

public function create()
{
    $pageTitle = 'Create Employee';

```

```

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

```

- Buat file View dengan nama **create.blade.php** pada **/resources/views/employee/**. Sesuaikan kode program pada file View tersebut 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>
  <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-lg-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">Home</a></li>
          <li class="nav-item col-2 col-md-auto"><a href="{{ route('employees.index') }}" class="nav-link">Employee List</a></li>
        </ul>

        <hr class="d-lg-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>

  <div class="container-sm mt-5">
    <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">

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

```

```

        <div class="alert alert-danger alert-dismissible fade show">
            {{ $error }}
            <button type="button" class="btn-close" data-bs-dismiss="alert"
aria-label="Close"></button>
        </div>
    @endforeach
@endif

<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" type="text" name="firstName"
id="firstName" value="" placeholder="Enter First Name">
    </div>
    <div class="col-md-6 mb-3">
        <label for="lastName" class="form-label">Last Name</label>
        <input class="form-control" type="text" name="lastName"
id="lastName" value="" placeholder="Enter Last Name">
    </div>
    <div class="col-md-6 mb-3">
        <label for="email" class="form-label">Email</label>
        <input class="form-control" type="text" name="email" id="email"
value="" placeholder="Enter Email">
    </div>
    <div class="col-md-6 mb-3">
        <label for="age" class="form-label">Age</label>
        <input class="form-control" type="text" name="age" id="age"
value="" placeholder="Enter Age">
    </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>
</form>
</div>

@vite('resources/js/app.js')

```

```
</body>
</html>
```

- Sesuaikan kode program **function store()** pada file **EmployeeController** tersebut seperti di bawah ini:

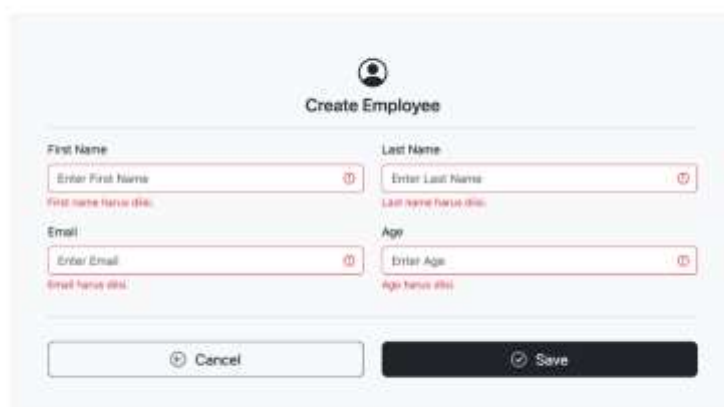
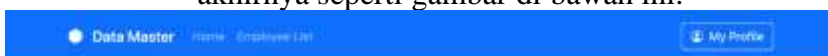
```
public function store(Request $request)
{
    $messages = [
        'required' => ':Attribute harus diisi.',
        'email' => 'Isi :attribute dengan format yang benar',
        'numeric' => 'Isi :attribute dengan angka'
    ];

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

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

    return $request->all();
}
```

- Pertanyaan:
 - Jelaskan dan dokumentasikan setiap kode program yang anda kerjakan di atas!
 - Tunjukkan penerapan **“Passing Data to Views”** pada Bab ini.
 - Tunjukkan penerapan **“CSRF Protection”** pada Bab ini.
 - Terapkan **“Displaying The Validation Errors”** pada Bab ini yang hasil akhirnya seperti gambar di bawah ini.



Terapkan **“Repopulating Forms”** pada Bab ini sehingga ketika muncul pesan error pada saat submit form, nilai yang sudah diisi pada form tidak ter-reset tetapi tetap muncul pada form tersebut.

Pertanyaan 1 :

1. Bagian Header (Head)

```
<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>
```

Penjelasan:

- **<meta charset="UTF-8">**: Mengatur encoding karakter halaman menjadi UTF-8 untuk mendukung berbagai karakter.
- **<meta http-equiv="X-UA-Compatible" content="IE=edge">**: Menginstruksikan browser untuk menggunakan versi terbaru dari rendering engine.
- **<meta name="viewport" content="width=device-width, initial-scale=1.0">**: Memastikan halaman responsif dengan menyesuaikan lebar viewport dengan perangkat.
- **<title>{{ \$pageTitle }}</title>**: Menampilkan judul halaman dinamis yang diambil dari variabel \$pageTitle.
- **@vite('resources/sass/app.scss')**: Mengimpor file SCSS yang dikelola oleh Vite untuk mendukung styling halaman.

2. Bagian Navigasi (Navbar)

```
<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-lg-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 active">Home</a>
        </li>
        <li class="nav-item col-2 col-md-auto">
          <a href="{{ route('employees.index') }}" class="nav-link">Employee List</a>
        </li>
      </ul>

      <hr class="d-lg-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>
```

Penjelasan:

- **<nav>**: Elemen navigasi utama yang menggunakan kelas **Bootstrap Navbar** untuk

membuat menu navigasi responsif.

- **Kelas navbar-expand-md navbar-dark bg-primary:**
 - navbar-expand-md: Navbar akan diperluas pada ukuran layar medium (md) ke atas.
 - navbar-dark bg-primary: Mengatur warna tema navbar menjadi gelap dengan latar belakang biru.
- **Logo dan Brand:**
 - `<a href="{{ route('home') }}"`: Menampilkan logo aplikasi dan mengarah ke route home.
 - `<i class="bi-hexagon-fill me-2"></i>`: Menggunakan ikon Bootstrap Icons untuk menampilkan ikon hexagon.
- **Hamburger Menu:**
 - `<button type="button">`: Tombol untuk menampilkan/menyembunyikan menu saat layar kecil.
- **Menu Navigasi:**
 - Menggunakan **Bootstrap Grid System** (col-2 col-md-auto) untuk membuat navigasi responsif.
 - Terdapat dua tautan:
 - **Home**: Menuju halaman utama.
 - **Employee List**: Menuju halaman daftar karyawan.
- **Tombol Profil:**
 - `<a href="{{ route('profile') }}"`: Tautan ke halaman profil.
 - `btn btn-outline-light`: Tombol dengan style outline warna putih.

3. Bagian Konten Utama

```
<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>
```

Penjelasan:

- `<div class="container mt-4">`: Membuat wadah utama dengan margin atas sebesar 4.
- `<h4>{{ $pageTitle }}</h4>`: Menampilkan judul halaman berdasarkan variabel \$pageTitle.
- `<hr>`: Garis horizontal sebagai pembatas.
- **Alert Box (Konten Informasi):**
 - `<div class="d-flex align-items-center ...">`: Menggunakan **Flexbox** untuk menyusun ikon dan teks secara horizontal.
 - **bg-light rounded-3 border**: Memberikan latar belakang terang, sudut membulat, dan border.
 - `<div class="bi-house-fill me-3 fs-1"></div>`: Menampilkan ikon rumah (Bootstrap Icons) dengan margin kanan dan ukuran besar.
 - `<h4 class="mb-0">`: Menampilkan teks deskriptif dengan margin bawah 0.

4. Bagian Footer (Scripts)

```
@vite('resources/js/app.js')
```

Penjelasan:

- Mengimpor file JavaScript dari Vite untuk mendukung fitur interaktivitas, seperti navbar collapse, efek animasi, dan lain-lain.

Fitur Utama dalam Kode

1. **Responsivitas**: Menggunakan Bootstrap Grid dan Flexbox.
2. **Dynamic Content**: Memanfaatkan variabel Laravel Blade ({{ \$pageTitle }}).
3. **Integrasi Asset**: SCSS dan JS dikelola menggunakan Vite untuk optimasi.

4. Navigasi Sederhana: Tautan halaman dengan route Laravel.

Pertanyaan 2 :

1. Struktur Dokumen HTML

```
<!DOCTYPE html>
<html lang="en">
```

Penjelasan:

- **<!DOCTYPE html>**: Mendefinisikan dokumen sebagai HTML5.
- **<html lang="en">**: Menentukan bahasa utama dokumen adalah bahasa Inggris (en), berguna untuk SEO dan alat bantu aksesibilitas.

2. Bagian <head>

```
<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>
```

Penjelasan:

- **<meta charset="UTF-8">**: Mengatur encoding teks menjadi UTF-8 untuk mendukung berbagai karakter.
- **<meta http-equiv="X-UA-Compatible" content="IE=edge">**: Memastikan kompatibilitas dengan browser modern.
- **<meta name="viewport" content="width=device-width, initial-scale=1.0">**: Mengatur halaman agar responsif terhadap berbagai ukuran perangkat.
- **<title>{{ \$pageTitle }}</title>**: Menampilkan judul halaman dinamis yang diambil dari variabel Laravel Blade \$pageTitle.
- **@vite('resources/sass/app.scss')**: Mengimpor file SCSS untuk styling dengan pengelolaan Vite.

3. Bagian Navbar (Navigasi Utama)

```
<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-lg-none text-white-50">
      <ul class="navbar-nav flex-row flex-wrap">
        <li class="nav-item col-6 col-md-auto"><a href="{{ route('home') }}" class="nav-link">Home</a></li>
        <li class="nav-item col-6 col-md-auto"><a href="{{ route('employees.index') }}" class="nav-link">Employee List</a></li>
      </ul>
      <hr class="d-lg-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>
```


Penjelasan:

1. **<nav>**: Komponen navigasi utama.
2. **Kelas navbar navbar-expand-md navbar-dark bg-primary**:
 - navbar-expand-md: Navigasi akan diperluas pada ukuran layar medium ke atas.
 - navbar-dark bg-primary: Warna teks navbar dibuat terang dengan latar biru.
3. **Logo dan Brand**:
 - ****: Link menuju halaman utama (home).
 - **<i class="bi-hexagon-fill me-2">**: Ikon hexagon dari **Bootstrap Icons** sebagai logo.
4. **Hamburger Menu**:
 - **<button type="button">**: Tombol untuk membuka menu navigasi pada perangkat kecil.
 - **data-bs-toggle="collapse"**: Memastikan menu akan menggunakan animasi collapse Bootstrap.
5. **Tautan Navigasi**:
 - Terdiri dari dua tautan:
 - **Home**: Menuju halaman utama.
 - **Employee List**: Menuju halaman daftar karyawan.
 - Kelas col-6 col-md-auto: Mengatur ukuran kolom agar responsif.
6. **Tombol Profil**:
 - ****: Tautan ke halaman profil.
 - **btn btn-outline-light**: Tombol bergaya outline dengan warna teks putih.

4. Bagian Konten Utama

```
<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-person-circle me-3 fs-1"></div>
    <h4 class="mb-0">Well done! this is {{ $pageTitle }}.</h4>
  </div>
</div>
```

Penjelasan:

1. **<div class="container mt-4">**:
 - Membuat container untuk konten utama dengan margin atas (margin-top) sebesar 4.
2. **<h4>{{ \$pageTitle }}</h4>**:
 - Menampilkan judul halaman berdasarkan variabel \$pageTitle.
3. **<hr>**:
 - Garis horizontal untuk pemisah.
4. **Card Box (Pemberitahuan/Informasi)**:
 - **<div class="d-flex align-items-center ..." >**:
 - Menggunakan **Flexbox** untuk menyusun ikon dan teks secara horizontal.
 - **py-2 px-4**: Padding vertikal 2 dan horizontal 4.
 - **bg-light rounded-3 border**: Latar terang, sudut membulat, dan border untuk desain bersih.
 - **<div class="bi-person-circle me-3 fs-1">**:
 - Ikon lingkaran orang (Bootstrap Icons).
 - me-3: Margin kanan 3, fs-1: Ukuran font besar.
 - **<h4 class="mb-0">**:
 - Menampilkan teks deskripsi dengan margin bawah nol (mb-0).

5. Footer (Scripts)

```
@vite('resources/js/app.js')
```

Penjelasan:

- **@vite**: Mengimpor file JavaScript yang dikelola oleh Vite.

- **resources/js/app.js:** File ini mendukung fitur interaktivitas, termasuk navigasi dan animasi yang ditangani oleh Bootstrap.

Fitur Utama Kode

1. **Responsivitas:**
 - Navbar dan konten dibuat responsif menggunakan Bootstrap Grid dan Flexbox.
2. **Dukungan Dinamis:**
 - Variabel Blade (`{{ $pageTitle }}`) memungkinkan judul halaman berubah sesuai kebutuhan.
3. **Bootstrap Integration:**
 - Menggunakan komponen Bootstrap untuk navigasi, desain, dan interaktivitas.
4. **Vite Asset Management:**
 - SCSS dan JS dikelola oleh Vite untuk efisiensi dan pengembangan modern.
5. **Ikun:**
 - Bootstrap Icons dipakai untuk memberikan sentuhan visual pada navigasi dan konten.

Pertanyaan 3

1. Penjelasan Function create()

```
public function create()
{
    $pageTitle = 'Create Employee';

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

Penjelasan:

- **\$pageTitle:** Variabel yang menyimpan judul halaman (Create Employee) untuk diteruskan ke view.
- **compact('pageTitle'):** Fungsi ini digunakan untuk "Passing Data to Views," yaitu mengirimkan variabel \$pageTitle ke file view yang dituju (employee.create).
- **view('employee.create', ...):** Memanggil file view bernama create.blade.php yang berada di dalam folder resources/views/employee.

Penerapan "Passing Data to Views":

Pada function ini, data \$pageTitle dikirimkan ke view create.blade.php menggunakan fungsi compact. Pada view, data ini diakses menggunakan sintaks Blade `{{ $pageTitle }}` untuk menampilkan judul halaman.

2. File View create.blade.php

Bagian Penting dan Implementasi:

CSRF Protection

```
<form action="{{ route('employees.store') }}" method="POST">
    @csrf
```

- **@csrf:** Melindungi form dari serangan **Cross-Site Request Forgery (CSRF)**. Token ini secara otomatis diverifikasi oleh Laravel setiap kali form dikirim.

Displaying Validation Errors

```
@if ($errors->any())
    @foreach ($errors->all() as $error)
        <div class="alert alert-danger alert-dismissible fade show">
            {{ $error }}
            <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
        </div>
    @endforeach
@endif
```

- **\$errors->any()**: Mengecek apakah ada error validasi yang terjadi.
- **\$errors->all()**: Mendapatkan daftar semua pesan error.
- **Alert**: Menampilkan pesan error di dalam elemen **Bootstrap Alert** dengan opsi untuk menutupnya menggunakan tombol.

Repopulating Forms

```
<input class="form-control" type="text" name="firstName" id="firstName" value="{{ old('firstName') }}" placeholder="Enter First Name">
```

- **old('firstName')**:
 - Mengambil nilai lama yang sebelumnya diisi oleh pengguna.
 - Ketika form gagal divalidasi, data yang sudah diisi tetap akan muncul di input form sehingga pengguna tidak perlu mengulang pengisian.

3. Function store()

```
public function store(Request $request)
{
    $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();
    }

    return $request->all();
}
```

Penjelasan:

1. **Custom Validation Messages:**
 - **\$messages**: Pesan khusus untuk validasi.
 - required: Pesan error jika field wajib tidak diisi.
 - email: Pesan error untuk format email yang salah.
 - numeric: Pesan error untuk field yang harus berupa angka.
2. **Validator:**
 - **Validator::make(\$request->all(), [...rules...], \$messages)**:
 - Memeriksa input dari form dengan aturan validasi dan pesan yang telah didefinisikan.
 - **Rules**:
 - firstName, lastName, email, dan age harus diisi (required).
 - email: Harus dalam format email.
 - age: Harus berupa angka.
3. **Handling Errors:**
 - **if (\$validator->fails())**: Mengecek apakah validasi gagal.
 - **redirect()->back()->withErrors(\$validator)->withInput()**:
 - Mengembalikan pengguna ke halaman form sebelumnya.
 - **withErrors**: Mengirimkan error validasi ke view.
 - **withInput()**: Mengirimkan kembali data input pengguna ke view.
4. **Repopulating Forms:**
 - Menggunakan fungsi **old('field_name')** untuk menampilkan data input yang telah dikirimkan.

4. Contoh Implementasi Dalam View

```
<div class="col-md-6 mb-3">
  <label for="firstName" class="form-label">First Name</label>
  <input class="form-control" type="text" name="firstName" id="firstName" value="{{
old('firstName') }}" placeholder="Enter First Name">
</div>
<div class="col-md-6 mb-3">
  <label for="email" class="form-label">Email</label>
  <input class="form-control" type="text" name="email" id="email" value="{{ old('email') }}"
placeholder="Enter Email">
</div>
```

- Pada contoh ini, ketika pengguna mengisi form dan validasi gagal, nilai firstName dan email tetap akan muncul di input form karena menggunakan fungsi **old()**.

5. Proses Akhir

- Ketika validasi gagal, error ditampilkan melalui komponen alert Bootstrap.
- Form tetap terisi dengan data pengguna melalui fungsi **old()**.
- Form dilindungi dari serangan CSRF menggunakan **@csrf**.
- Semua data dikembalikan ke pengguna melalui mekanisme Laravel, sehingga pengalaman pengguna lebih baik.

Git Hub Modul 12 :

<https://github.com/JohanesOryza/Modul-11-update-main>