

A. Identitas Job Sheet

IDENTITAS JOB SHEET

Perguruan Tinggi	: Politeknik Negeri Bengkalis	Pertemuan Ke	:
Jurusan/Program Studi	: Teknik Informatika	Job Ke	: 2
Kode Mata Kuliah	:	Halaman	: 73
Nama Mata Kuliah	: MPPL	Mulai Berlaku	:

B. Komponen Job Sheet

2.1. Judul Job

JOB II

Pengenalan *Controller*, *Routing* Laravel dan github

2.2. Tujuan

1. Mahasiswa mampu mengenal dan memahami apa yang dimaksud dengan *controller* dan *routing* dalam laravel.
2. Mahasiswa mampu membuat dan mengelola sebuah *controller* dan *routing* dalam laravel dengan benar.
3. Mahasiswa dapat bernavigasi antar halaman dengan menggunakan *framework* laravel.

2.3. Dasar Teori

Dalam mengembangkan sebuah website, tentunya tidak terlepas dengan yang namanya navigasi antar halaman. Navigasi antar halaman adalah sebuah tindakan dimana pengguna website dapat berpindah dari satu halaman ke halaman yang lain dengan menekan sebuah tombol, meng-klik hyperlink, dan lain-lain. Di dalam pemrograman website yang sudah anda pelajari sebelumnya, bernavigasi ke halaman tertentu sangat mudah sekali, anda hanya perlu menuliskan lokasi halaman tersebut disimpan dalam project, misalnya “/user/login”. Berbeda dengan laravel, anda perlu mendefinisikan *route* terlebih dahulu lalu menuliskan nama *route* yang sudah dibuat tersebut kedalam sebuah link atau sejenisnya. Selain sebuah website dapat bernavigasi ke halaman tertentu, sebuah

website harus dapat melakukan proses seperti menampilkan data dari database, melakukan CRUD, dan lain-lain. Semua hal tersebut dapat anda lakukan dengan mudah pada laravel, dengan menggunakan yang namanya *controller*, dimana di dalam file tersebut akan berisi sebuah proses logika mulai dari menerima sebuah request dan masih banyak lagi. Dalam praktikum kali ini, anda akan mempelajari bagaimana mengimplementasikan kedua cara diatas, untuk itu diharapkan anda menyimak dengan baik langkah-langkah yang akan dilakukan dalam praktikum kali ini.

2.4. Alat & Bahan

Alat dan bahan yang diperlukan dalam praktikum ini adalah:

1. Komputer/laptop dengan minimal versi windows 7.
2. IDE Visual Studio Code.
3. Plugin IDE Visual Studio Code:
 - A. Laravel artisan
 - B. Laravel namespace
 - C. Laravel blade snippets
 - D. Laravel blade spacer
 - E. Laravel goto controller
 - F. Laravel goto view
 - G. Laravel snippets
 - H. Prettier-code formatter
 - I. Laravel blade formatter
4. XAMPP.
5. Browser

2.5. Implementasi

Pada praktikum kali ini kita masih menggunakan project yang sudah dibuat pada **modul 1**. Sebelumnya pada editor VS Code tambahkan extension baru dengan nama **PHP Intelephense**. Fungsi ekstension ini membantu untuk merapikan sintak php. Untuk penggunaanya seleksi semua sintaks (**Ctrl + A**) kemudian tekan tombol (**Ctrl + Shift + F**). maka otomatis sintaks kita akan dirapikan.

Pertama-tama, kita akan mengganti nama file **home.blade.php** yang sudah dibuat sebelumnya menjadi **biodata.blade.php**. Selanjutnya kita buat kembali file

“home.blade.php”, lalu tuliskan kode berikut kedalam file blade yang sudah baru dibuat tadi sebagai berikut:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.cs
s">
    <title>Homepage</title>
</head>
<body>
    <nav class="navbar navbar-dark bg-primary">
        <div class="container">
            <a class="navbar-brand" href="/">
                Politeknik Negeri Bengkalis | D-IV Rekayasa Perangkat Lunak</a>
            </div>
        </nav>
        <div class="container" style="margin-top: 150px">
            <div class="row">
                <div class="col-12 text-center">
                    <h1 class="text-secondary">Selamat Datang!</h1>
                    <h4 class="text-secondary">
                        Di Perpustakaan Politeknik Negeri Bengkalis</h4>
                    <h6 class="mt-3">
                        Silahkan
                        <a href="#" style="text-decoration: none">masuk</a>
                        atau <a href="#" style="text-decoration: none">daftar</a>
                        jika anda belum punya akun
                    </h6>
                </div>
            </div>
        </div>
        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.m
in.js"></script>
    </body>
</html>
```

Setelah itu, pergi ke folder routes lalu buka file web.php, pada file tersebut ubah kode route *default* menjadi seperti berikut

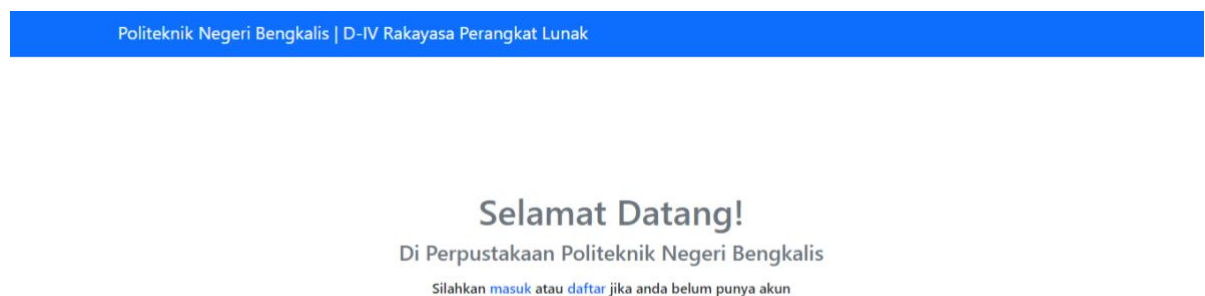
```

use App\Http\Controllers\LoginRegisterController;
/*
|-----

Route::get('/', function () {
    return view('home');
});

```

Setelah itu coba refresh website yang sudah dijalankan, tampilan akan berubah menjadi seperti gambar berikut:



Selanjutnya kita akan membuat halaman login dan register dimana pengguna akan dapat melakukan proses autentikasi nantinya. Buatlah sebuah folder baru dengan nama “**auth**” di dalam folder views, kemudian buatlah 2 file blade di dalam folder auth yang sudah dibuat tadi dengan nama “**login.blade.php**” dan “**register.blade.php**”. Pada file login.blade.php, tuliskan kode sebagai berikut:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.10.5/font/bootstrap-icons.css">
    <title>Login</title>
</head>
<body>
    <div class="container">
        <a href="/">
            <i class="bi-arrow-left h1"></i>
        </a>

```

```

</div>
<div class="container d-flex justify-content-center align-items-
center" style="margin-top: 60px">
  <div class="card" style="width: 35%">
    <div class="card-body p-4">
      <h3 class="card-title text-center">Login</h3>
      <form>
        @csrf
        <div class="form-group mt-4">
          <label class="text-secondary">Email Anda</label>
          <input type="email" class="form-control border border-
secondary form-control-lg" name="email"><br>
        </div>
        <div class="form-group mt-1">
          <label class="text-secondary">Password Anda</label>
          <input type="password"
class="form-control border border-secondary form-control-lg"
name="password">
        </div>
        <button type="submit" class=" form-control btn btn-primary mt-5">
Login</button>
      </form>
      <p class="mt-2 text-center">Belum punya akun?
        <a href="#" style="text-decoration: none">
          Ayo buat akun!
        </a>
      </p>
    </div>
  </div>
</div>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bund
le.min.js"></script>
</body>
</html>

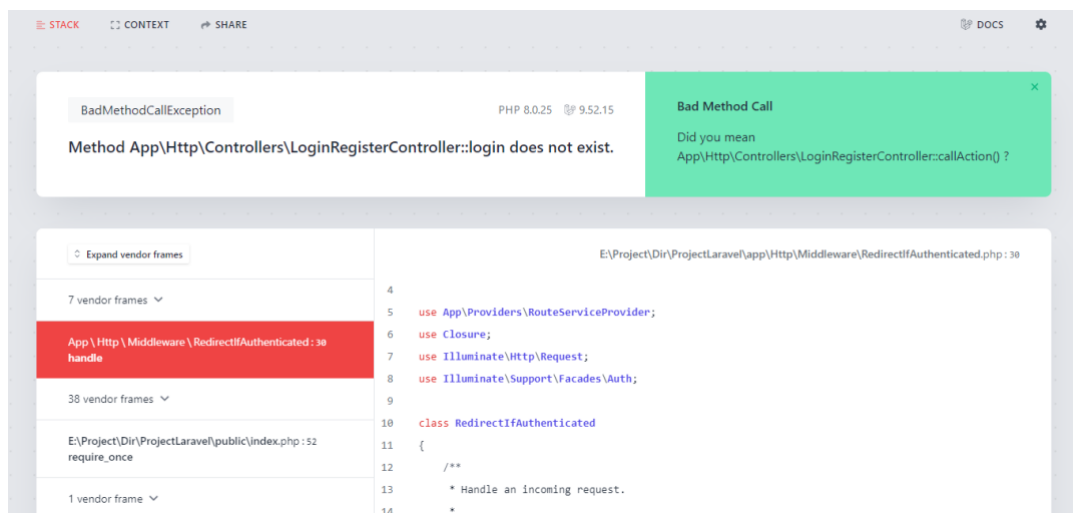
```

Selanjutnya kita akan membuat sebuah *routing* agar pengguna nantinya dapat berpindah ke halaman login dari halaman **HOME** yang sudah kita buat sebelumnya. Kita akan membuat sebuah *controller* dengan nama “LoginRegisterController”. Untuk membuatnya terdapat 2 cara, kita bisa menggunakan terminal atau dengan plugin yang sudah kita install ke dalam vs code. Untuk cara yang pertama kita bisa membuat terminal lalu mengetikkan perintah “**php artisan make:controller LoginRegisterController**” lalu tekan enter, maka akan dibuat file *controller* kedalam project laravel kita di folder **App/Http/Controller**. Untuk cara yang kedua kita bisa menggunakan plugin php artisan, caranya adalah tekan tombol ctrl+shift+p pada keyboard. Pada kolom tuliskan “Artisan:Make Controller” lalu

tuliskan nama controller yang akan kita buat, yakni LoginRegisterController lalu tekan enter, kemudian pilih opsi basic, maka secara otomatis *controller* akan dibuat kedalam project laravel kita. Setelah itu, kita akan membuat sebuah *route* yang nantinya akan mengarahkan kepada halaman login. Caranya adalah tambahkan kode dibawah berikut kedalam file **web.php**.

```
Route::get('/auth/login', [LoginRegisterController::class, 'login'])->>name('auth.login');
```

Selanjutnya, coba jalankan website nya dengan menuliskan pada kolom url seperti ini “**http://127.0.0.1:8000/auth/login**”. Jika url tersebut diakses maka muncul tampilan *error* seperti pada gambar berikut.



hal ini bisa terjadi karena kita belum menambahkan sebuah *method* yang akan mengembalikan nilai berupa *view*/tampilan dari halaman login yang baru saja kita buat.

Namun sebelum memperbaiki, anda terlebih dahulu harus dapat memahami tampilan *error* diatas. Ketika membuat sebuah program, tentunya tidak terlepas dari yang namanya error, baik yang terduga maupun tidak terduga. Berbeda pada saat anda membuat sebuah website menggunakan *php native* dimana ketika terjadi error, pesan yang disampaikan tidak detail sehingga akan membuat anda bingung untuk memperbaiki kesalahan tersebut.

Menggunakan laravel, pada saat anda mengalami *error* pada bagian kode tertentu, anda dapat melihat detail dari pesan *error* tersebut dan melihat pada bagian mana yang menyebabkan hal tersebut dapat terjadi. Dengan demikian, proses untuk *error handling*

dapat menjadi lebih baik. Baik, selanjutnya kita akan memperbaiki kesalahan tersebut dengan bernavigasi kedalam file *controller* yang sudah dibuat tadi lalu tambahkan kode berikut:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class LoginRegisterController extends Controller
{
    public function login() {
        return view('auth.login');
    }
}
```

Silahkan akses lagi “**http://127.0.0.1:8000/auth/login**”, maka halaman login pada website akan tampil. Selanjutnya, kita akan memasukkan route tersebut kedalam kode html yang sudah kita buat. Pergi ke file *home.blade.php* dan tambahkan kode yang diberi warna kuning

```
-----
-----
<h6 class="mt-3">. Silahkan
    <a href="{{ route('auth.login') }}" style="text-
        decoration: none">masuk</a>
    atau
    <a href="#" style="text-decoration: none">
        daftar</a> jika anda belum punya akun
-----
-----
</body>
</html>
```

Setelah menambahkan kode diatas, selanjutnya coba refresh Kembali halaman **home** atau landing page yang sudah kita buat, lalu klik hyperlink “masuk”, jika sudah benar maka akan diarahkan ke halaman login.



Login

Email Anda

Password Anda

Login

Belum punya akun? [Ayo buat akun!](#)

Silahkan tambahkan tombol **Kembali** ke halaman **Home** pada halaman **Login**

Langkah selanjutnya adalah, kita akan membuat halaman register yang cara membuatnya hampir sama dengan kita membuat halaman login. Pertama-tama, tuliskan kode dibawah ini pada file **register.blade.php**.

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

<head>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.10.5/font/bootstrap-icons.css">
    <title>Register</title>
</head>

<body>
    <div class="container">
        <a href="/">
            <i class="bi-arrow-left h1"></i>
        </a>
    </div>
    <div class="container d-flex justify-content-center align-items-
center" style="margin-top: 60px">
        <div class="card" style="width: 35%">
            <div class="card-body p-4">
                <h3 class="card-title text-center">Register</h3>
                <form>
                    @csrf
```



```

        <div class="form-group mt-4">
            <label class="text-secondary">Nama Anda</label>
            <input type="text" class="form-control border
border-secondary form-control-lg"
                name="name"><br>
        </div>
        <div class="form-group mt-1">
            <label class="text-secondary">Email Anda</label>
            <input type="email" class="form-control border
border-secondary form-control-lg"
                name="email"><br>
        <div class="form-group mt-1">
            <label class="text-secondary">Pilih Jenis
Kelamin</label><br>
            <div class="form-check form-check-inline">
                <input class="form-check-input" type="radio"
name="jenisKelamin" value="Laki-laki">
                <label class="form-check-label"
for="inlineRadio1">Laki-laki</label>
            </div>
            <div class="form-check form-check-inline">
                <input class="form-check-input" type="radio"
name="jenisKelamin" value="Perempuan">
                <label class="form-check-label"
for="inlineRadio2">Perempuan</label>
            </div>
        </div><br>
        </div>
        <div class="form-group mt-1">
            <label class="text-secondary">Password
Anda</label>
            <input type="password" class="form-control
border border-secondary form-control-lg"
                name="password">
        </div><br>
        <div class="form-group mt-1">
            <label class="text-secondary">Konfirmasi
Password Anda</label>
            <input type="password" class="form-control
border border-secondary form-control-lg"
                name="password_confirmation" required>
        </div>
        <button type="submit" class="form-control btn btn-
primary mt-5">Register</button>
    </form>
    <p class="mt-2 text-center">Sudah punya akun? <a
href="#" style="text-decoration: none">Ayo

```

```

                                masuk!</a></p>
                                </div>
                                </div>
                                </div>
                                <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></script>
                                </body>
                                </html>

```

Setelah menuliskan kode diatas, kita akan menambahkan sebuah *function* pada *controller* yang akan mengembalikan *view* daripada halaman register. Caranya sama pada halaman login, tuliskan kode dibawah berikut pada file LoginRegisterController.

```

public function register() {
    return view('auth.register');
}

```

Selanjutnya, kita akan menambahkan *route* dari halaman register dengan cara yang sama pada halaman login, dengan menambahkan kode sebagai berikut pada file **web.php**.

```

Route::get('/auth/register', [LoginRegisterController::class,
'register'])->name('auth.register');

```

Setelah itu, kita akan menambahkan *route* yang sudah kita buat kedalam halaman landing page atau home, login, dan register dimana pada register kita akan menambahkan *route* yang akan menuju ke halaman login jika pengguna sudah memiliki akun. Tambahkan kode yang diberi warna kuning.

home.blade.php

```

<h6 class="mt-3">. Silahkan
    <a href="{{ route('auth.login') }}" style="text-decoration: none">masuk</a>
atau
    <a href="{{ route('auth.register') }}" style="text-decoration: none">
    daftar</a> jika anda belum punya akun

```

login.blade.php

```
-----  
-----  
<p class="mt-2 text-center">Belum punya akun?  
  <a href="{{ route('auth.register') }}"  
    style="text-decoration: none">Ayo buat akun!</a></p>  
-----  
-----  
</body>  
</html>
```

register.blade.php

```
-----  
-----  
<button type="submit" class=" form-control btn btn-primary mt-5">  
Register</button>  
  </form>  
  <p class="mt-2 text-center">Sudah punya akun? <a  
href="{{ route('auth.login') }}"  
  style="text-decoration: none">Ayo  
  masuk!</a></p>  
-----  
-----
```

Setelah melengkapi kode pada file *view* yang kita miliki, maka anda bisa bernavigasi ke halaman home atau landing page, register, dan login. Untuk saat ini, belum ada logika yang akan memproses tombol login dan register, sehingga ketika anda meng-klik tombol tersebut tidak akan terjadi apa-apa.

TUGAS

- Silahkan tambah kan 3 VIEW baru (Halaman Web) pada Halaman Home untuk dimana berisikan pengenalan **PRODI RPL** kalian seperti **Berita RPL**, **Profile Lulusan RPL**, dan **Aktivitas Mahasiswa RPL**
- Tambahkan juga navigasi baru pada Halaman Home menuju halaman biodata yang sudah dibuat di Modul 1