

**LAPORAN PRAKTIKUM  
PERANCANGAN DAN PEMROGRAMAN WEB**

**MODUL XI  
LARAVEL**



**Universitas  
Telkom**

Oleh:

(Zaenarif Putra 'Ainurdin) (2311104049)

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK  
DIREKTORAT KAMPUS PURWOKERTO  
UNIVERSITAS TELKOM  
2025**

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Dasar Teori**

Routing pada framework Laravel merupakan mekanisme yang berfungsi sebagai penghubung antara permintaan (request) dari pengguna melalui browser dengan proses yang dijalankan oleh aplikasi. Setiap URL yang diakses akan dipetakan oleh route untuk menentukan apakah permintaan tersebut diarahkan ke controller, view, atau proses tertentu lainnya. Dengan adanya routing, alur kerja aplikasi menjadi lebih terstruktur, terkontrol, dan mudah dikelola karena seluruh lalu lintas request diatur secara terpusat sesuai dengan pola dan aturan yang telah ditentukan.

#### **1.2 Tujuan**

- a. Mahasiswa mampu memahami konsep dan implementasi MVC menggunakan web framework Laravel.

#### **1.3 Manfaat**

- a. Memudahkan pengelolaan alur akses URL dalam aplikasi web.
- b. Memisahkan logika aplikasi dari tampilan sehingga lebih terstruktur.
- c. Mendukung penggunaan parameter, redirect, dan pengelompokan route.
- d. Meningkatkan keamanan dengan pengaturan middleware pada route tertentu.
- e. Mempermudah proses pengembangan dan pemeliharaan aplikasi.

## BAB II

### HASIL PRAKTIKUM

#### 2.1 Guided

##### a. Source Code

Controller.php

```
<?php

namespace App\Http\Controllers;

abstract class Controller
{
    //
}
```

PageController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PageController extends Controller
{
    public function index()
    {
        return view('welcome');
    }

    public function tampil()
    {
        $arrMahasiswa = [
            "Kholifahdina",
            "Rahmat Taufik",
        ];
    }
}
```

```
        "Nita Fitrotunimah",
        "Defrin Anggun Saputri"
    ];

    return view('mahasiswa')->with('mahasiswa',
$arrMahasiswa);
}
}
```

mahasiswa.blade.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Belajar Laravel</title>
</head>
<body>
    <h2>Data Mahasiswa</h2>
    <ol>
        <?php foreach ($mahasiswa as $nama) {
            echo "<li> $nama </li>";
        } ?>
    </ol>
</body>
</html>
```

mahasiswa.blade.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Data Mahasiswa</title>
</head>
<body>
```

```
<h1>Data Mahasiswa</h1>

<ol>
    @forelse ($mahasiswa as $val)
        <li>{{ $val }}</li>
    @empty
        <div>Tidak ada data...</div>
    @endforelse
</ol>

</body>
</html>
```

### master.blade.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>@yield('title')</title>
</head>
<body>
    @yield('content')
</body>
</html>
```

### produk.blade.php

```
@extends('master')
@section('title', 'Data Produk')
@section('content')
Produk 1 : {{ $prod1; }}
<br>
Produk 2 : {{ $prod2; }}
<br>
Produk 3 : {{ $prod3; }}
@endsection
```

### web.php

```
<?php

use Illuminate\Support\Facades\Route;

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

// Membuat route
Route::get('/beranda', function () {
    return "Halaman Beranda";
});

// Route parameter
Route::get('/kendaraan/{jenis}', function ($jenis) {
return "Tampilkan data kendaraan dengan jenis $jenis";
});

// Route dengan optional parameter
Route::get('/kendaraan/{jenis?}/{merek?}', function ($a = 'motor', $b = 'honda') {
return "Cek harga kendaraan $a $b";
});

// Route parameter dengan regular expression
Route::get('/product/{id}', function ($id) {
return "Tampilkan product dengan id = $id";
});

Route::get('/product/{id}', function ($id) {
return "Tampilkan product dengan id = $id";
})->where('id', '[0-9]+');

// Route redirect
Route::get('/hubungi-kami', function () {
return '<h1>Hubungi Kami</h1>';
});
```

```
Route::redirect('/contact-us', '/hubungi-kami');

// Route group
Route::prefix('/admin')->group(function() {
    Route::get('/dashboard', function() {
        return 'Tampilkan dashboard aplikasi';
    });

    Route::get('/datapegawai', function() {
        return 'Tampilkan data pegawai';
    });

    Route::get('/datamahasiswa', function() {
        return 'Tampilkan data mahasiswa';
    });
});

// Route fallback
Route::fallback(function () {
    return "Maaf, alamat tidak ditemukan";
});

// Route priority
Route::get('/baju/1', function () {
    return "Baju ke-1";
});

Route::get('/baju/1', function () {
    return "Baju saya ke-1";
});

Route::get('/baju/1', function () {
    return "Baju kita ke-1";
});

// Route priority menggunakan router parameter
Route::get('/baju/{a}', function ($a) {
    return "Baju ke-$a";
});

Route::get('/baju/{b}', function ($b) {
    return "Baju saya ke-$b";
});
```

```
) ;

Route::get('/baju/{c}', function ($c) {
    return "Baju kita ke-$c";
}) ;

// membuat view
Route::get('/mahasiswa', function () {
$arrMhs = [
"mhs1" => "Abdul hafidz",
"mhs2" => "Aufar Bintang",
"mhs3" => "Muhammad Nur Hamada",
"mhs4" => "Hastin Ajeng"
];
return view('universitas.mahasiswa', $arrMhs);
}) ;

Route::get('/mahasiswa', function () {
$arrMhs = ["Abdul hafidz","Aufar Bintang",
"Muhammad Nur Hamada","Hastin Ajeng"];
return view('universitas.mahasiswa',['mahasiswa' => $arrMhs]);
}) ;

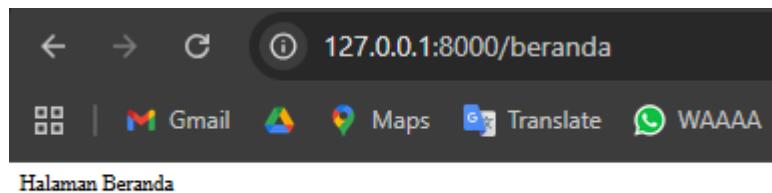
Route::get('/mahasiswa', function () {
$arrMhs = ["Abdul hafidz","Aufar Bintang",
"Muhammad Nur Hamada","Hastin Ajeng"];
return view('universitas.mahasiswa')->with('mahasiswa',
$arrMhs);
}) ;

Route::get('/produk', function () {
$arrProduk = [
"prod1" => "Televisi",
"prod2" => "Kipas Angin",
"prod3" => "Radio"
];
return view('produk', $arrProduk);
}) ;
```

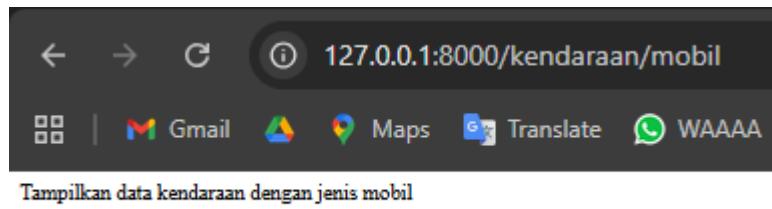
```
});  
  
Route::get('/', [App\Http\Controllers\PageController::class, 'index']);  
Route::get('/mahasiswa', [App\Http\Controllers\PageController::class, 'tampil']);
```

b. Screenshot Output

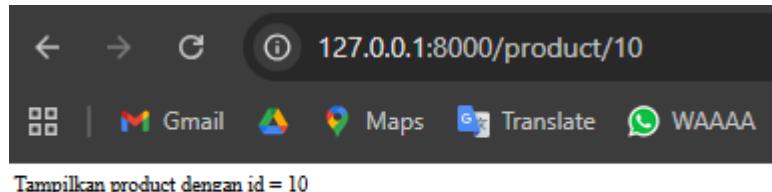
- <http://localhost:8000/beranda> → Menampilkan halaman beranda aplikasi



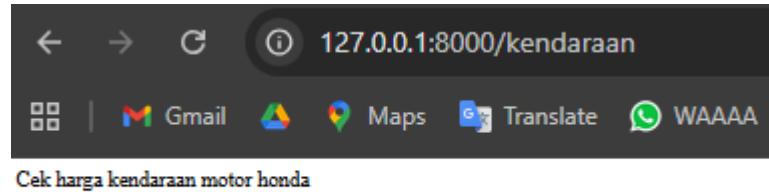
- <http://localhost:8000/kendaraan/mobil> → Menampilkan data kendaraan dengan jenis tertentu



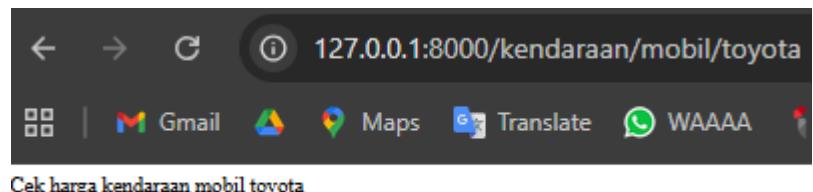
- <http://localhost:8000/product/10> → Menampilkan produk dengan ID berupa angka



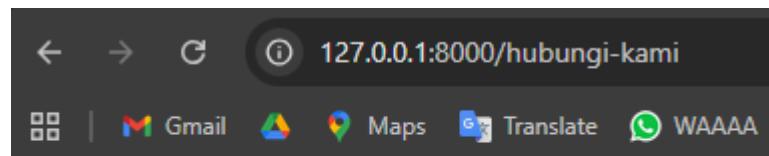
- <http://localhost:8000/kendaraan> → Menampilkan harga kendaraan default (motor honda)



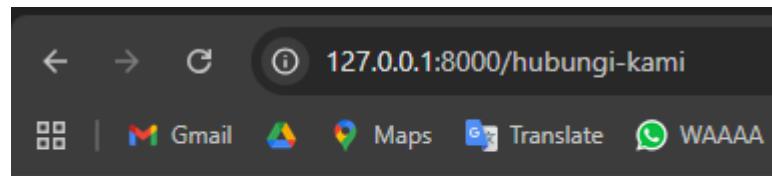
- <http://localhost:8000/kendaraan/mobil/toyota> → Menampilkan harga kendaraan mobil toyota



- <http://localhost:8000/contact-us> → Dialihkan otomatis ke /hubungi-kami

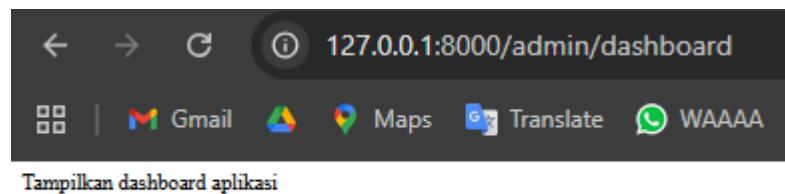


- <http://localhost:8000/hubungi-kami> → Menampilkan halaman Hubungi Kami

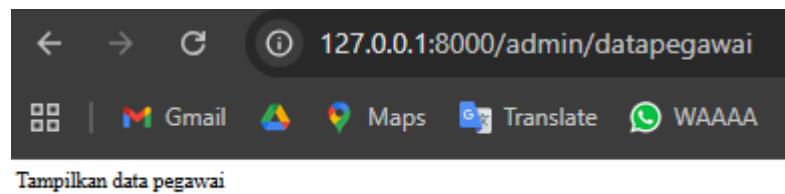


## Hubungi Kami

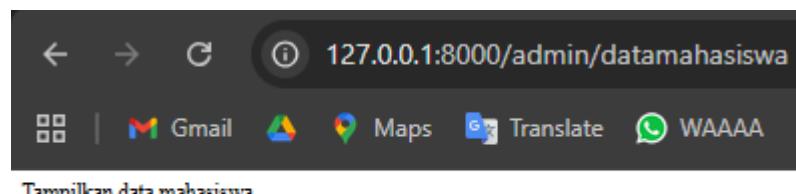
- <http://localhost:8000/admin/dashboard> → Menampilkan dashboard aplikasi



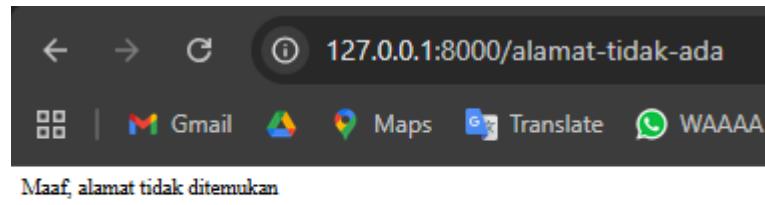
- <http://localhost:8000/admin/datapegawai> → Menampilkan data pegawai



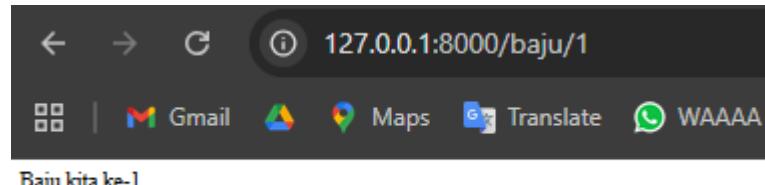
- <http://localhost:8000/admin/datamahasiswa> → Menampilkan data mahasiswa



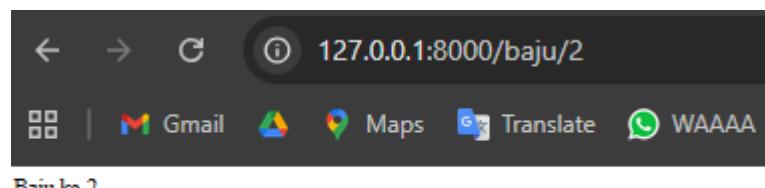
- `http://localhost:8000/alamat-tidak-ada` → Menampilkan pesan: Maaf, alamat tidak ditemukan



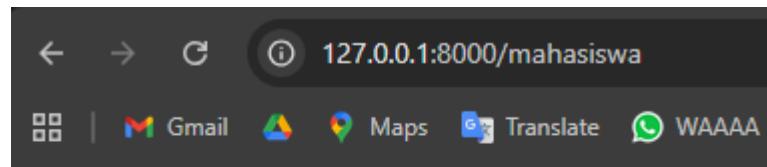
- `http://localhost:8000/baju/1` → Menampilkan "Baju kita ke-1" (karena route terakhir yang dijalankan)



- `http://localhost:8000/baju/2` → Menampilkan "Baju ke-2" (route parameter pertama yang diprioritaskan)



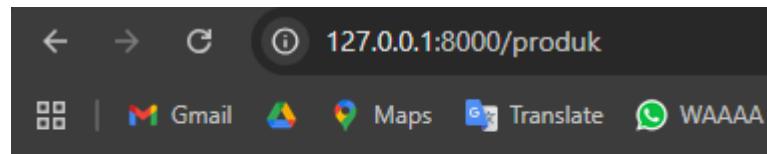
- `http://localhost:8000/mahasiswa` → Menampilkan data mahasiswa menggunakan View + Blade (data dikirim dalam bentuk array dan ditampilkan dengan perulangan)



## Data Mahasiswa

1. Kholifahdina
2. Rahmat Taufik
3. Nita Fitrotunimah
4. Defrin Anggun Saputri

- <http://localhost:8000/produk> → Menampilkan data produk (Televisi, Kipas Angin, Radio)



### c. Deksripsi Program

Program ini merupakan aplikasi web sederhana berbasis framework Laravel yang menerapkan konsep Model–View–Controller (MVC) untuk mengelola alur permintaan pengguna melalui browser hingga menampilkan hasil ke layar. Pada tahap awal, file web.php berperan sebagai router, yaitu komponen yang menerima setiap URL yang diakses pengguna dan menentukan proses selanjutnya, apakah langsung menampilkan teks, memanggil view, atau meneruskan ke controller. Berbagai jenis route diterapkan, seperti route tanpa parameter, route dengan parameter, optional parameter, regular expression, redirect, route group, fallback, dan priority, yang semuanya bertujuan untuk mengatur jalur akses aplikasi secara terstruktur dan terkontrol.

Algoritma kerja program dimulai ketika pengguna mengetikkan alamat URL di browser, kemudian Laravel mencocokkan URL tersebut dengan daftar

route yang ada di web.php. Jika URL sesuai, maka Laravel mengeksekusi fungsi (closure) atau method controller yang dipetakan pada route tersebut. Contohnya, ketika pengguna mengakses halaman /mahasiswa, route akan memanggil method tampil() pada PageController, yang berfungsi untuk menyiapkan data mahasiswa dalam bentuk array. Data ini kemudian dikirim ke view menggunakan method with(), sehingga dapat diproses dan ditampilkan di sisi tampilan.

Controller PageController bertugas sebagai pengolah logika aplikasi, di mana method index() digunakan untuk menampilkan halaman utama (welcome), sedangkan method tampil() digunakan untuk mengelola dan mengirim data mahasiswa ke view. View mahasiswa.blade.php berfungsi sebagai penyaji tampilan, yang menampilkan data mahasiswa dalam bentuk daftar menggunakan perulangan Blade (@forelse) maupun perulangan PHP biasa. Sintaks Blade seperti @extends, @section, dan @yield pada file master.blade.php dan produk.blade.php digunakan untuk menerapkan konsep layout, sehingga struktur tampilan menjadi lebih rapi dan mudah dikelola.

Secara keseluruhan, alur kerja program dimulai dari route → controller → view, di mana setiap sintaks memiliki peran masing-masing: route sebagai pengarah, controller sebagai pengolah logika dan data, serta view sebagai penampil hasil akhir. Output yang dihasilkan berupa halaman web dinamis yang menampilkan teks, data mahasiswa, data produk, serta respon sesuai URL yang diakses, sehingga pengguna dapat memahami bagaimana routing, controller, dan view bekerja secara terpadu dalam Laravel.

## 2.2 Unguided

### 1. Latihan

#### a. Source Code

BarberController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class BarberController extends Controller
{
    public function index()
    {
        $data = [
            'title' => 'Ada Barbershop - Dashboard',
            'kapsters' => [
                ['id' => 1, 'name' => 'Andi', 'position' =>
'Senior Barber'],
                ['id' => 2, 'name' => 'Budi', 'position' =>
'Junior Barber'],
                ['id' => 3, 'name' => 'Citra', 'position' =>
'Stylist'],
            ],
            'total_kapsters' => 15,
            'attendance_today' => 12
        ];

        return view('barbershop.dashboard', $data);
    }
}
```

style.css

```
/* style.css - Simple CSS untuk Barbershop */
body {
```

```
font-family: Arial, sans-serif;
background-color: #f5f5f5;
margin: 0;
padding: 20px;
}

.header {
background-color: #333;
color: white;
padding: 20px;
text-align: center;
}

.container {
max-width: 800px;
margin: 20px auto;
background: white;
padding: 20px;
border-radius: 5px;
}

.button {
background-color: #007bff;
color: white;
padding: 10px 20px;
border: none;
border-radius: 5px;
cursor: pointer;
}

.button:hover {
background-color: #0056b3;
}
```

### script.js

```
// script.js - Simple JavaScript untuk Barbershop
function showWelcome() {
```

```
    alert("Selamat datang di Ada Barbershop!");
}

function changeHeader() {
    document.getElementById("header").style.backgroundColor =
"#555";
}

function showDateTime() {
    const now = new Date();
    document.getElementById("datetime").innerHTML =
now.toLocaleString();
}
```

### dashboard.blade.php

```
<!DOCTYPE html>
<html>
<head>
    <title>{{ $title }}</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            padding: 20px;
        }
        .dashboard {
            max-width: 1000px;
            margin: 0 auto;
        }
        .card {
            background: #f8f9fa;
            padding: 20px;
            margin: 10px 0;
            border-radius: 5px;
            border-left: 4px solid #007bff;
        }
        .kapster-list {
            list-style: none;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="row">
            <div class="col-md-12 dashboard">
                <div class="card">
                    <ul class="list-group kapster-list">
                        <li>Item 1</li>
                        <li>Item 2</li>
                        <li>Item 3</li>
                    </ul>
                </div>
            </div>
        </div>
    </div>
</body>

```

```
        padding: 0;
    }
    .kapster-item {
        padding: 10px;
        border-bottom: 1px solid #ddd;
    }

```

</style>

```
</head>
<body>
    <div class="dashboard">
        <h1>{{ $title }}</h1>

        <div class="card">
            <h2>Statistik</h2>
            <p>Total Kapster: <strong>{{ $total_kapsters }}</strong></p>
            <p>Hadir Hari Ini: <strong>{{ $attendance_today }}</strong></p>
        </div>

        <div class="card">
            <h2>Daftar Kapster</h2>
            <ul class="kapster-list">
                @foreach($kapsters as $kapster)
                <li class="kapster-item">
                    ID: {{ $kapster['id'] }} -
                    Nama: {{ $kapster['name'] }} -
                    Posisi: {{ $kapster['position'] }}
                </li>
                @endforeach
            </ul>
        </div>

        <div class="card">
            <h2>Informasi Sistem</h2>
            <p>Halaman ini ditampilkan menggunakan Controller di Laravel</p>
        </div>
    </div>

```

```
<p>Controller: <code>BarberController</code></p>
<p>Method: <code>index()</code></p>
</div>
</div>
</body>
</html>
```

for.blade.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Perulangan For</title>
</head>
<body>
    <h1>Perulangan For</h1>

    @for ($i = 1; $i <= 10; $i++)
        <p>Ini adalah perulangan ke-{{ $i }}</p>
    @endfor
</body>
</html>
```

mahasiswa.blade.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Data Mahasiswa</title>
</head>
<body>
    <h1>Data Nilai Mahasiswa</h1>
```

```
<ul>
    @foreach ($nilai as $n)
        <li>Nilai: {{ $n }}</li>
    @endforeach
</ul>
</body>
</html>
```

### show\_image.blade.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Tampilkan Gambar</title>
</head>
<body>
    <h1>Gambar Barbershop</h1>
    
    <br>
    <p>Path gambar: {{ asset('images/barbershop.jpg') }}</p>
</body>
</html>
```

### while.blade.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Perulangan While</title>
</head>
<body>
    <h1>Perulangan While</h1>

    @php $i = 1; @endphp
```

```
@while ($i <= 10)
    <p>Ini adalah perulangan ke-{{ $i }}</p>
    @php $i++; @endphp
@endwhile
</body>
</html>
```

with `_assets.blade.php`

```
<!DOCTYPE html>
<html>
<head>
    <title>Dengan CSS & JS</title>
    <!-- Load CSS dari folder css -->
    <link rel="stylesheet" href="{{ asset('css/style.css') }}>
</head>
<body>
    <div class="header" id="header">
        <h1>Ada Barbershop</h1>
        <p>Sistem Manajemen Kapster</p>
    </div>

    <div class="container">
        <h2>Selamat Datang</h2>
        <p>Halaman ini menggunakan CSS dan JavaScript
eksternal</p>

        <button class="button" onclick="showWelcome()">Klik
untuk Welcome</button>
        <button class="button" onclick="changeHeader()">Ubah
Warna Header</button>
        <button class="button"
onclick="showDateTime()">Tampilkan Waktu</button>

        <p id="datetime" style="margin-top: 20px;
font-weight: bold;"></p>
```

```

<h3>Asset yang digunakan:</h3>
<ul>
    <li>CSS: {{ asset('css/style.css') }}</li>
    <li>JavaScript: {{ asset('js/script.js') }}</li>
</ul>
</div>


<script src="{{ asset('js/script.js') }}"></script>
</body>
</html>

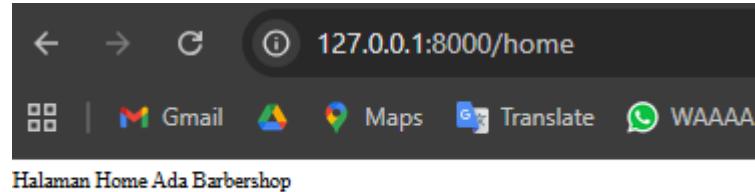
```

b. Screenshot Output

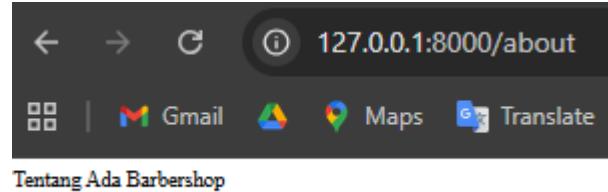
1. TUGAS - Router

Route tanpa parameter:

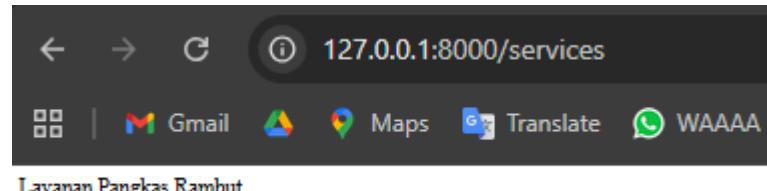
- <http://localhost:8000/home> → Menampilkan halaman Home Barbershop



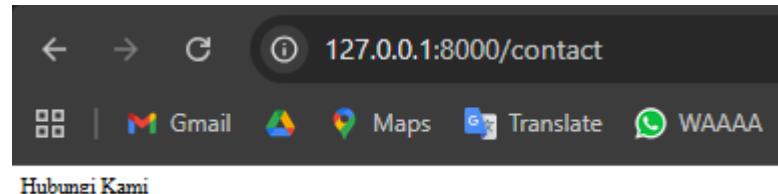
- <http://localhost:8000/about> → Menampilkan halaman Tentang Barbershop



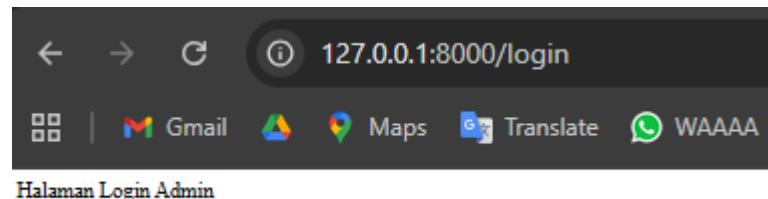
- <http://localhost:8000/services> → Menampilkan layanan pangkas rambut



- http://localhost:8000/contact → Menampilkan halaman kontak

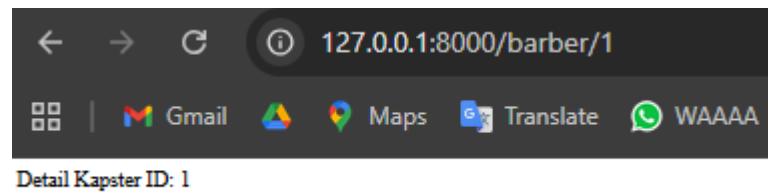


- http://localhost:8000/login → Menampilkan halaman login admin

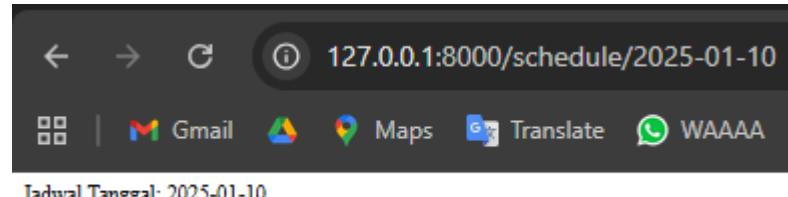


Route dengan parameter:

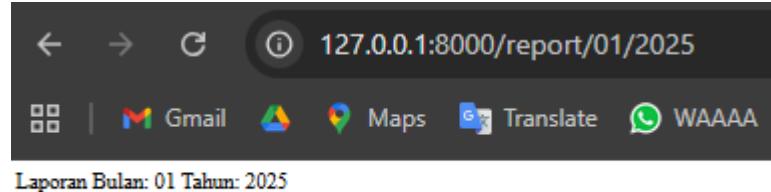
- http://localhost:8000/barber/1 → Menampilkan detail kapster berdasarkan ID



- http://localhost:8000/schedule/2025-01-10 → Menampilkan jadwal berdasarkan tanggal

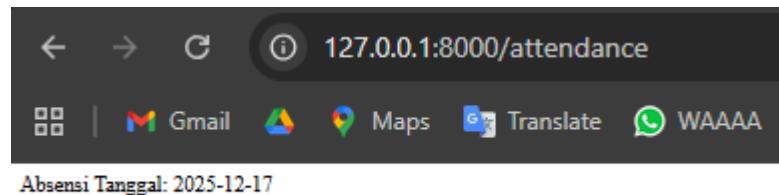


- <http://localhost:8000/report/01/2025> → Menampilkan laporan berdasarkan bulan dan tahun

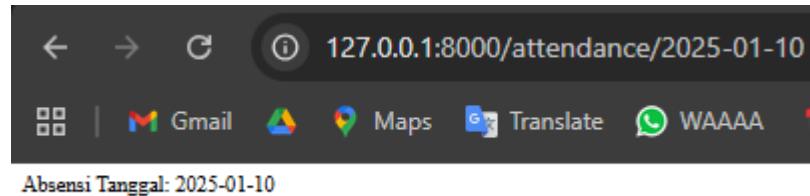


Route dengan optional parameter:

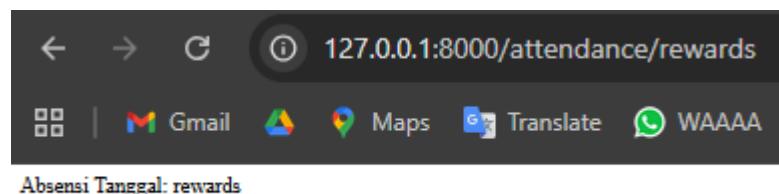
- <http://localhost:8000/attendance> → Menampilkan absensi dengan tanggal hari ini



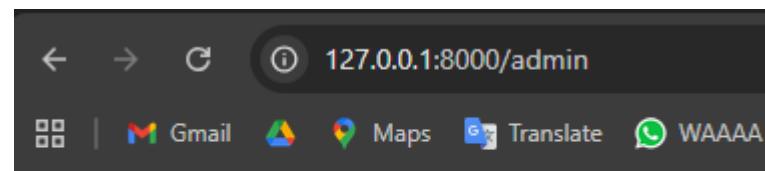
- <http://localhost:8000/attendance/2025-01-10> → Menampilkan absensi berdasarkan tanggal tertentu



- <http://localhost:8000/rewards> → Menampilkan semua reward bulanan



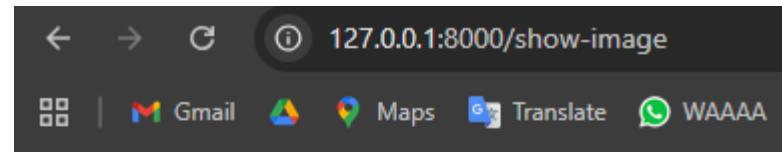
- <http://localhost:8000/admin> → Menampilkan dashboard admin



Admin Section: dashboard, Action: view

## 2. TUGAS - View

- <http://localhost:8000/show-image> → Menampilkan view dengan gambar

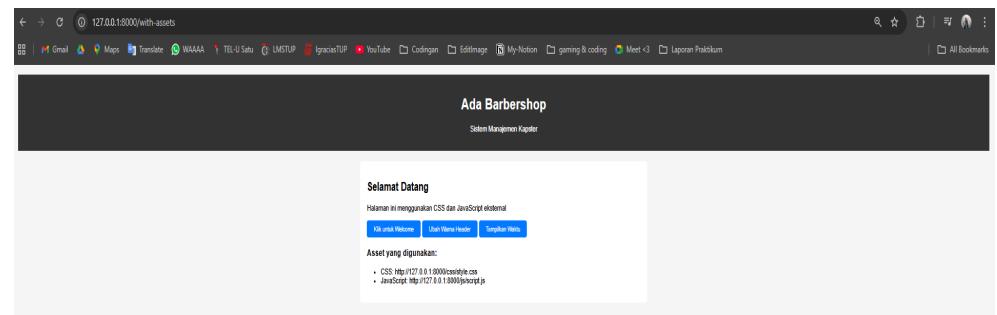


### Gambar Barbershop



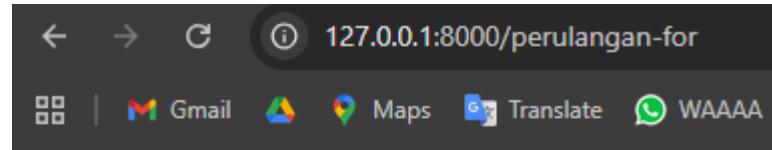
Path gambar: <http://127.0.0.1:8000/images/barbershop.jpg>

- <http://localhost:8000/with-assets> → Menampilkan view dengan CSS dan JavaScript



### 3. TUGAS - BLADE TEMPLATE ENGINE

- <http://localhost:8000/perulangan-for> → Menampilkan perulangan for



### Perulangan For

Ini adalah perulangan ke-1

Ini adalah perulangan ke-2

Ini adalah perulangan ke-3

Ini adalah perulangan ke-4

Ini adalah perulangan ke-5

Ini adalah perulangan ke-6

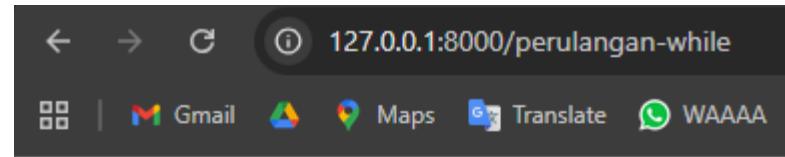
Ini adalah perulangan ke-7

Ini adalah perulangan ke-8

Ini adalah perulangan ke-9

Ini adalah perulangan ke-10

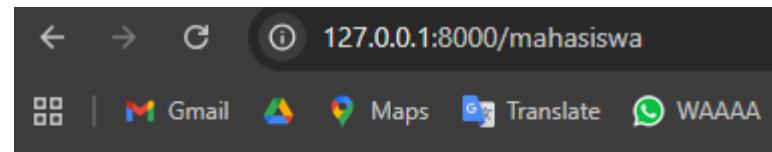
- <http://localhost:8000/perulangan-while> → Menampilkan perulangan while



## Perulangan While

Ini adalah perulangan ke-1  
Ini adalah perulangan ke-2  
Ini adalah perulangan ke-3  
Ini adalah perulangan ke-4  
Ini adalah perulangan ke-5  
Ini adalah perulangan ke-6  
Ini adalah perulangan ke-7  
Ini adalah perulangan ke-8  
Ini adalah perulangan ke-9  
Ini adalah perulangan ke-10

- <http://localhost:8000/mahasiswa> → Menampilkan data nilai mahasiswa\

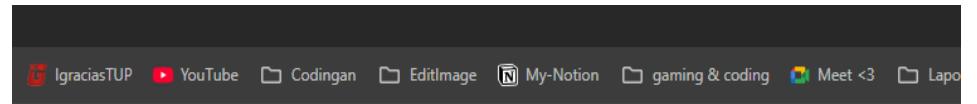


## Data Nilai Mahasiswa

- Nilai: 80
- Nilai: 64
- Nilai: 30
- Nilai: 76
- Nilai: 95

#### 4. TUGAS - Controller

- <http://localhost:8000/barbershop> → Akses controller BarberController@index



### c. Dekripsi Program

Program ini adalah sebuah aplikasi web sederhana yang dibangun menggunakan framework Laravel PHP untuk mengelola sistem informasi mahasiswa. Algoritma yang digunakan mengikuti pola MVC (Model-View-Controller) dimana data mengalir dari route menuju controller, kemudian diproses dan dikirim ke view untuk ditampilkan. Program bekerja dengan membuat beberapa halaman web dasar: halaman home, about, dan data mahasiswa yang menampilkan daftar nilai. Cara kerjanya dimulai ketika pengguna mengakses URL seperti localhost:8000/home, kemudian Laravel akan mencari route yang sesuai di file web.php, menjalankan fungsi controller jika ada, mengambil data dari model jika diperlukan, dan merender view dengan template Blade untuk menghasilkan output HTML yang ditampilkan di browser.

## **BAB III**

### **KESIMPULAN & SARAN**

#### **3.1 Kesimpulan**

Routing merupakan komponen penting dalam Laravel yang berperan sebagai pengatur utama alur permintaan pengguna ke dalam sistem aplikasi. Dengan memanfaatkan routing secara tepat, pengembang dapat membangun aplikasi web yang rapi, efisien, dan mudah dikembangkan sesuai dengan konsep MVC yang diterapkan oleh Laravel.

#### **3.2 Saran**

Dalam pengembangan aplikasi Laravel, disarankan untuk merancang struktur routing sejak awal proyek agar alur aplikasi lebih jelas dan konsisten. Selain itu, penggunaan route group dan penamaan route yang baik akan sangat membantu dalam pengelolaan aplikasi berskala besar.