

LAPORAN PRAKTIKUM
PERANCANGAN DAN PEMROGRAMAN WEB

MODUL XI
(Laravel : Introduction)



Oleh:

(Faishal Arif Setiawan) (2311104066)

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

BAB I

PENDAHULUAN

A. Dasar Teori

Tuliskan teori dasar yang mendasari materi praktikum, penjelasan lebih mendalam diluar modul akan menjadi nilai tambahan.

B. Tujuan

1. Mahasiswa mampu memahami konsep dan implementasi MVC menggunakan web frameworkLaravel.

BAB II

HASIL

A. GUIDED (Praktikum Terbimbing)

Membuat Halaman View/mahasiwa.blade.php

```
<!DOCTYPE html>
<html>
<head>
<title>Belajar Laravel</title>
```

```
</head>

<body>

<h2>Data Mahasiswa</h2>

<ol>

<li><?php echo $mhs1; ?></li>

<li><?php echo $mhs2; ?></li>

<li><?php echo $mhs3; ?></li>

<li><?php echo $mhs4; ?></li>

</ol>

</body>

</html>
```

Membuat Route/web.php

```
<?php

use Illuminate\Support\Facades\Route;

/*
|-----
| Web Routes
|-----
*/
```

```
*/
```

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

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

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

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

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

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

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

```
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(function () {  
return "Maaf, alamat tidak ditemukan";  
});
```

```
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::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}";
```

```
});
```

```
Route::get('/mahasiswa', function () {
```

```
return view('mahasiswa');
```

```
});
```

```
Route::get('/mahasiswa', function () {
```

```
return view('universitas.mahasiswa',["mhs1"=>"Abdul Hafidz"]);
```

```
});
```

```
Route::get('/mahasiswa', function () {
```

```
return view('universitas.mahasiswa',  
[  
"mhs1" => "Abdul hafidz",  
"mhs2" => "Aufar Bintang",  
"mhs3" => "Muhammad Nur Hamada",  
"mhs4" => "Hastin Ajeng"  
]);  
});
```

Penjelasan:

menjelaskan dasar-dasar Laravel mulai dari konsep MVC (Model-View-Controller). MVC itu cara Laravel memisahkan aplikasi jadi tiga bagian: Model untuk database, View untuk tampilan, dan Controller untuk logika program. Lalu dijelaskan juga tentang Route, yaitu pintu masuk saat user membuka URL—route inilah yang menentukan apakah Laravel akan memanggil tampilan (view) atau controller. Setelah itu, modul membahas View dan Blade, yaitu cara Laravel menampilkan halaman web dengan syntax yang lebih sederhana dibanding PHP biasa, serta bagaimana membuat template agar kode tidak berulang-ulang. Terakhir, modul menjelaskan Controller, yaitu tempat utama logika program ditulis—controller akan menerima request dari route, memproses data (misalnya dari database), lalu mengirim kembali hasilnya ke view.outputnya akan menampilkan data mahasiswa

Output:

Data Mahasiswa

1. Abdul hafidz
2. Aufar Bintang
3. Muhammad Nur Hamada
4. Hastin Ajeng

B. UNGUIDED (Tugas Mandiri)

1. Soal 1:

1.1 TUGAS - Router

1. Buat minimal 5 route tanpa parameter
2. Buat minimal 3 route dengan parameter
3. Buat minimal 3 route dengan optional parameter

Route yang dibuat nantinya akan diimplementasikan pada tugas besar di mata kuliah teori

Jawaban:

Source Code:

```
Route::get(uri: '/', action: [HomeController::class, 'index'])->name(name: 'home');
Route::middleware('guest')->group(callback: function () { void {
Route::get(uri: '/login', action: [AuthController::class, 'loginForm'])->name(name: 'login');
Route::get(uri: '/produk', action: [ProductController::class, 'index'])->name(name: 'products');
Route::get(uri: '/cart', action: [CartController::class, 'index'])
    @return \Illuminate\Routing\Route
Route::get(uri: '/checkout', action: [OrderController::class, 'checkout'])->name(name: 'order');
Route::get(uri: '/produk/{slug}', action: [ProductController::class, 'show'])->name(name: 'product');
Route::get(uri: '/kategori/{category}', action: [HomeController::class, 'category'])->name(name: 'category');
Route::post(uri: '/cart/add/{product}', action: [CartController::class, 'add'])->name(name: 'add');
Route::get('/search/{keyword?}', [ProductController::class, 'search'])->name('search');
Route::get('/admin/laporan/{bulan?}', [AdminDashboardController::class, 'report'])->name('admin');
Route::get('/orders/filter/{status?}', [OrderController::class, 'filter'])->name('orders.filter');
```

2. Soal 2:

Jawaban:

Source Code:

```
<!DOCTYPE html>

<html lang="id">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Tugas 3.1 - Blade Template Engine</title>

<link rel="stylesheet" href="{{ asset('css/style.css') }}">

</head>

<body>

<div class="container">

<h1>Tugas 3.1 - Blade Template Engine</h1>


<h2>1. Perulangan FOR (Bilangan 1 s.d 10)</h2>

<div class="result">

<ul>

@for ($i = 1; $i <= 10; $i++)

<li>Bilangan ke-{{ $i }}</li>

@endfor

</ul>

</div>


<h2>2. Perulangan WHILE (Bilangan 1 s.d 10)</h2>

<div class="result">

<ul>
```

```
@php $j = 1; @endphp
@while ($j <= 10)
<li>Angka: {{ $j }}</li>
@php $j++; @endphp
@endwhile
</ul>
</div>

<h2>3. Perulangan FOREACH (Nilai Mahasiswa)</h2>
<div class="result">
<ul>
@foreach ($nilai as $index => $n)
<li>Nilai ke-{{ $index + 1 }}: {{ $n }}</li>
@endforeach
</ul>
</div>
</div>

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

```
<!DOCTYPE html>
<html lang="id">
<head>
```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Tugas 2.1 - View Asset</title>
<link rel="stylesheet" href="{{ asset('css/style.css') }}">
</head>
<body>
<div class="container">
<h1>Tugas 2.1 - Pengelolaan Asset</h1>

<h2>1. Gambar dari folder img</h2>

<p>Gambar ini berasal dari folder public/img</p>

<h2>2. CSS dari folder css</h2>
<div class="result">
<p>CSS berhasil dimuat dari folder public/css/style.css</p>
<p>Anda dapat melihat styling pada halaman ini.</p>
</div>

<h2>3. JavaScript dari folder js</h2>
<div class="result">
<button onclick="showMessage()">Klik untuk test JavaScript</button>
<p>JavaScript berhasil dimuat dari folder public/js/script.js</p>
</div>
</div>

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

```
</body>
```

```
</html>
```

Output:



Tugas 3.1 - Blade Template Engine

1. Perulangan FOR (Bilangan 1 s.d 10)

Bilangan ke-1

Bilangan ke-2

Bilangan ke-3

Bilangan ke-4

Bilangan ke-5

Bilangan ke-6

Bilangan ke-7

Bilangan ke-8

Bilangan ke-9

Bilangan ke-10

2. Perulangan WHILE (Bilangan 1 s.d 10)

Angka: 1

Angka: 2

Angka: 3

Angka: 4

Angka: 5

Angka: 6

Angka: 7

Angka: 8

Angka: 9

Angka: 10

3. Perulangan FOREACH (Nilai Mahasiswa)

Nilai ke-1: 80

Nilai ke-2: 64

Nilai ke-3: 30

Nilai ke-4: 76

Nilai ke-5: 95

Penjelasan:

Project ini adalah contoh bagaimana kita mengelola asset di Laravel, yaitu file seperti CSS, gambar, dan JavaScript yang disimpan di folder public. Pada bagian `<link rel="stylesheet" ...>` halaman ini mengambil file `style.css` dari folder `public/css`, sehingga tampilan halaman bisa dihias, misalnya mengatur warna, ukuran teks, dan layout. Lalu ada juga tag `` yang menampilkan gambar `sample.jpg` dari folder `public/img`, jadi kita bisa lihat cara Laravel memanggil gambar menggunakan fungsi `asset()`. Di bagian bawah, ada JavaScript yang diambil dari `public/js/script.js`, dan tombol di halaman tersebut digunakan untuk mengetes apakah file JavaScript berhasil dijalankan.

Secara sederhana, project ini menunjukkan bahwa semua file pendukung tampilan website tidak perlu dicampur ke dalam blade, tapi cukup disimpan rapi di folder asset (`css`, `js`, `img`)

dalam folder public, lalu dipanggil menggunakan fungsi asset(). Dengan cara ini, struktur project jadi lebih teratur dan mudah dikelola kalau suatu saat file bertambah banyak.

3. Soal 3:

Kerjakan tugas berikut menggunakan blade

1. Buat perulangan for untuk menampilkan bilangan 1 s.d 10
2. Buat perulangan while untuk menampilkan bilangan 1 s.d 10
3. Buat perulangan foreach untuk menampilkan nilai pada route berikut

```
Route::get('/mahasiswa', function () {  
    $nilai = [80,64,30,76,95];  
    return view('mahasiswa',$nilai);  
});
```

Jawaban:

Source code:

```
<!DOCTYPE html>  
<html lang="id">  
<head>  
<meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-scale=1.0">  
<title>Tugas 3.1 - Blade Template Engine</title>  
<link rel="stylesheet" href="{{ asset('css/style.css') }}">  
</head>  
<body>  
<div class="container">  
<h1>Tugas 3.1 - Blade Template Engine</h1>  
  
<h2>1. Perulangan FOR (Bilangan 1 s.d 10)</h2>  
<div class="result">  
<ul>
```

```
@for ($i = 1; $i <= 10; $i++)
```

```
<li>Bilangan ke-{{ $i }}</li>
```

```
@endfor
```

```
</ul>
```

```
</div>
```

```
<h2>2. Perulangan WHILE (Bilangan 1 s.d 10)</h2>
```

```
<div class="result">
```

```
<ul>
```

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

```
@while ($j <= 10)
```

```
<li>Angka: {{ $j }}</li>
```

```
@php $j++; @endphp
```

```
@endwhile
```

```
</ul>
```

```
</div>
```

```
<h2>3. Perulangan FOREACH (Nilai Mahasiswa)</h2>
```

```
<div class="result">
```

```
<ul>
```

```
@foreach ($nilai as $index => $n)
```

```
<li>Nilai ke-{{ $index + 1 }}: {{ $n }}</li>
```

```
@endforeach
```

```
</ul>
```

```
</div>
```

```
</div>
```

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

</body>

</html>
```

Route:

```
Route::get(uri: '/mahasiswa', action: function (): View {
    $nilai = [80,64,30,76,95];
    return view(view: 'mahasiswa',data: ['nilai' => $nilai]);
});
```

Output:

Tugas 3.1 - Blade Template Engine

1. Perulangan FOR (Bilangan 1 s.d 10)

Bilangan ke-1

Bilangan ke-2

Bilangan ke-3

Bilangan ke-4

Bilangan ke-5

Bilangan ke-6

Bilangan ke-7

Bilangan ke-8

Bilangan ke-9

Bilangan ke-10

2. Perulangan WHILE (Bilangan 1 s.d 10)

Angka: 1

Angka: 2

Angka: 3

Angka: 4

Angka: 5

Angka: 6

Angka: 7

Angka: 8

Angka: 9

Angka: 10

3. Perulangan FOREACH (Nilai Mahasiswa)

Nilai ke-1: 80

Nilai ke-2: 64

Nilai ke-3: 30

Nilai ke-4: 76

Nilai ke-5: 95

Penjelasan:

Project ini adalah latihan untuk memahami bagaimana **perulangan (looping)** bekerja di Laravel menggunakan Blade. Di dalam halaman ini, ada tiga jenis perulangan yang digunakan: **for**, **while**, dan **foreach**. Perulangan **FOR** dipakai untuk menampilkan tulisan “Bilangan ke-1” sampai “Bilangan ke-10”. Lalu, perulangan **WHILE** juga melakukan hal yang sama, tetapi dengan cara kerja yang berbeda—nilai awalnya dibuat dulu, lalu terus bertambah selama masih kurang atau sama dengan 10. Terakhir, perulangan **FOREACH** digunakan untuk menampilkan daftar **nilai mahasiswa** yang dikirim dari route ke view, sehingga setiap nilai akan muncul satu per satu di layar.

Jadi, halaman ini menampilkan tiga daftar:

1. Daftar angka 1 sampai 10 menggunakan loop FOR
2. Daftar angka 1 sampai 10 menggunakan loop WHILE
3. Daftar nilai mahasiswa menggunakan loop FOREACH

4. Soal 4

1. Buat root yang mengarah ke controller
2. Tampilkan template pada view menggunakan controller

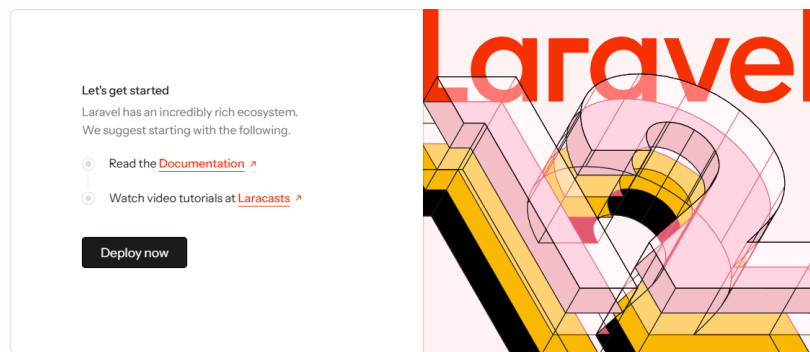
Jawaban:

Source code:

```
Route::get(uri: '/controller-example', action: [MahasiswaController::class, 'index']);
```

Output:

127.0.0.1:8000/controller-example



Penjelasan:

Laravel tidak langsung menampilkan halaman, tetapi akan memanggil sebuah controller bernama MahasiswaController, tepatnya function (method) bernama index() di dalam controller tersebut.

BAB III

PENUTUP

A. Kesimpulan

Pada praktikum dan tugas mandiri ini, saya belajar konsep dasar Laravel mulai dari routing, view, blade template, hingga penggunaan controller. Hal yang paling penting dipahami adalah bahwa Laravel menggunakan konsep **MVC (Model–View–Controller)** untuk memisahkan logika program, tampilan, dan data agar aplikasi menjadi lebih terstruktur. Saya juga belajar bagaimana cara membuat route, menampilkan halaman menggunakan view, mengelola asset seperti CSS, gambar, dan JavaScript, serta menggunakan Blade Template Engine untuk melakukan perulangan dan menampilkan data secara dinamis. Selain itu, saya memahami bagaimana controller bekerja sebagai penghubung antara route dan view.