

LAPORAN PRAKTIKUM
PERANCANGAN DAN PEMROGRAMAN WEB

MODUL 12
(LARAVEL 1)



Oleh:

(Farhan Kurniawan)

(2311104073)

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

BAB I

PENDAHULUAN

A. Dasar Teori

Laravel adalah sebuah *web application framework* berbasis bahasa pemrograman PHP yang bersifat *open-source* dan menggunakan pola arsitektur Model–View–Controller (MVC). Framework ini dirancang untuk mempermudah dan mempercepat proses pengembangan aplikasi web dengan menyediakan berbagai fitur bawaan seperti routing, manajemen database melalui ORM Eloquent, sistem templating Blade, validasi form, serta pengelolaan autentikasi dan keamanan. Dengan konsep MVC, Laravel mampu memisahkan logika bisnis, pengolahan data, dan tampilan antarmuka sehingga kode program menjadi lebih terstruktur, mudah dipahami, dan mudah dipelihara.

Laravel pertama kali dikembangkan oleh Taylor Otwell pada tahun 2011 dan hingga kini terus berkembang menjadi salah satu framework PHP paling populer di dunia. Popularitas Laravel didukung oleh dokumentasi yang lengkap, komunitas yang besar, serta ekosistem yang kaya seperti Composer, Artisan CLI, dan berbagai package pendukung. Selain itu, Laravel juga dikenal karena menghasilkan aplikasi yang aman, bersih, dan efisien, sehingga sangat cocok digunakan baik untuk pembelajaran pengembangan web maupun untuk membangun aplikasi web skala kecil hingga besar.

B. Tujuan

Tujuan dari penggunaan framework Laravel adalah untuk membantu pengembang dalam membangun aplikasi web secara lebih cepat, terstruktur, dan efisien dengan menerapkan konsep Model–View–Controller (MVC). Melalui Laravel, pengembang dapat memanfaatkan berbagai fitur bawaan seperti routing, controller, view, dan Blade Template Engine sehingga proses pengembangan tidak perlu dilakukan dari awal. Selain itu, Laravel bertujuan untuk memudahkan pengelolaan kode program agar lebih rapi, mudah dipelihara, serta mampu meningkatkan keamanan dan kualitas aplikasi web yang dikembangkan.

BAB II

HASIL

A. UNGUIDED

1. Input:

routes/web.php

Membuat 5 Route Tanpa Parameter

```
Route::get('/home', function () {  
    return "Halaman Home";  
});  
  
Route::get('/about', function () {  
    return "Halaman About";  
});  
  
Route::get('/contact', function () {  
    return "Halaman Contact";  
});  
  
Route::get('/profile', function () {  
    return "Halaman Profile";  
});  
  
Route::get('/help', function () {  
    return "Halaman Help";  
});
```

Membuat 3 Route Dengan Parameter

```
Route::get('/home', function () {  
    return "Halaman Home";  
});  
  
Route::get('/about', function () {  
    return "Halaman About";  
});  
  
Route::get('/contact', function () {  
    return "Halaman Contact";  
});  
  
Route::get('/profile', function () {  
    return "Halaman Profile";  
});  
  
Route::get('/help', function () {  
    return "Halaman Help";  
});
```

Membuat 3 Route Dengan Optional Parameter

```
Route::get('/kendaraan/{jenis?}', function ($jenis = "Motor") {  
    return "Jenis Kendaraan: $jenis";  
});  
  
Route::get('/buku/{judul?}/{penulis?}', function ($judul = "Laravel", $penulis = "Anonim") {  
    return "Judul: $judul, Penulis: $penulis";  
});  
  
Route::get('/nilai/{angka?}', function ($angka = 0) {  
    return "Nilai: $angka";  
});
```

2. TUGAS VIEW & ASSET

public/assets/css/style.css

```
body {  
    background-color: #f4f4f4;  
    font-family: Arial;  
    text-align: center;  
}
```

public/assets/js/script.js

```
alert("JavaScript berhasil dijalankan");
```

Membuat View

resources/views/asset.blade.php

```
<!DOCTYPE html>  
<html>  
<head>  
    <title>Asset Laravel</title>  
    <link rel="stylesheet" href="{{ asset('assets/css/style.css') }}">  
</head>  
<body>  
  
    <h1>Halaman Asset</h1>  
  
      
  
    <script src="{{ asset('assets/js/script.js') }}"></script>  
</body>  
</html>
```

Route ke View

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

3. TUGAS BLADE TEMPLATE ENGINE

resources/views/mahasiswa.blade.php

3.1 Perulangan FOR (1–10)

```
<h3>Perulangan For</h3>  
@for ($i = 1; $i <= 10; $i++)  
    {{ $i }} <br>  
@endfor
```

3.2 Perulangan WHILE (1–10)

```
<h3>Perulangan While</h3>  
@php $i = 1; @endphp  
@while ($i <= 10)  
    {{ $i }} <br>  
    @php $i++; @endphp  
@endwhile
```

3.3 Perulangan FOREACH dari Route

routes/web.php

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

mahasiswa.blade.php

```
<h3>Perulangan Foreach</h3>  
@foreach ($nilai as $n)  
    Nilai: {{ $n }} <br>  
@endforeach
```

4. TUGAS CONTROLLER

4.1. Membuat Controller

php artisan make:controller PageController

app/Http/Controllers/PageController.php

```
<?php  
  
namespace App\Http\Controllers;  
  
class PageController extends Controller
```

```

{
    public function index() {
        return view('welcome');
    }

    public function mahasiswa() {
        $data = ["Andi", "Budi", "Citra", "Dina"];
        return view('mahasiswa_controller', ['mahasiswa' => $data]);
    }
}

```

4.2. Route ke Controller

routes/web.php

```
use App\Http\Controllers\PageController;
```

```
Route::get('/', [PageController::class, 'index']);
```

```
Route::get('/mhs', [PageController::class, 'mahasiswa']);
```

4.3. View dari Controller

resources/views/mahasiswa_controller.blade.php

```

<!DOCTYPE html>
<html>
<head>
    <title>Data Mahasiswa</title>
</head>
<body>

<h2>Daftar Mahasiswa</h2>
<ul>
@foreach ($mahasiswa as $mhs)
    <li>{{ $mhs }}</li>
@endforeach
</ul>

</body>
</html>

```

BAB III

PENUTUP

A. Kesimpulan

Berdasarkan praktikum Modul 9 yang telah dilakukan, dapat disimpulkan bahwa PHP merupakan bahasa pemrograman server-side yang mampu mengolah data dan menghasilkan halaman web yang dinamis. Melalui pembuatan program konversi suhu dan kalkulator diskon, mahasiswa dapat memahami penerapan variabel, operator, serta struktur kondisi seperti *if-else* dalam menyelesaikan permasalahan perhitungan. Selain itu, penggunaan fungsi bawaan PHP seperti *number_format()* membantu menampilkan hasil perhitungan secara lebih rapi dan sesuai kebutuhan pengguna.

Selain itu, pada manipulasi array nilai mahasiswa, PHP terbukti efektif dalam pengolahan data menggunakan array dan perulangan. Mahasiswa dapat menampilkan nilai tertinggi, nilai terendah, rata-rata, jumlah kelulusan, serta mengurutkan data nilai dengan memanfaatkan fungsi array dan struktur perulangan. Dengan praktikum ini, mahasiswa diharapkan mampu memahami konsep dasar PHP secara menyeluruh serta mengaplikasikannya dalam pengembangan aplikasi web sederhana berbasis server-side scripting.