

PRAKTIKUM PEMROGRAMAN FRAMEWORK

MODUL 3 Routing dan Bundling Asset di Laravel



Sistem Informasi
Telkom University
Surabaya

Disusun Oleh:
Fahimah - 1204220061

**PROGRAM STUDI S1 SISTEM INFORMASI
FAKULTAS REKAYASA INDUSTRI
TELKOM UNIVERSITY SURABAYA
2024**

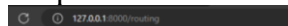
1. Praktik Laravel Routing

Source code

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

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <meta http-equiv="X-UA-Compatible" content="ie=edge">  
    <title>Belajar Laravel Routing</title>  
    @vite('resources/sass/app.scss')  
</head>  
<body>  
    <div class="container m-5">  
        <h1>Belajar Laravel Routing</h1>  
  
        <div class="list-group list-group-numbered mt-4">  
            {{{-- Kode anda selanjutnya letakkan di sini --}}  
        </div>  
        {{{-- Kode program untuk Route Groups di sini --}}  
    </div>  
    @vite('resources/js/app.js')  
</body>  
</html>
```

Output:



Belajar Laravel Routing

Analisa:

Diawali dengan mendefinisi sebuah route menggunakan HTTP dengan method GET, agar route dapat diakses melalui URL '/routing'. '@vite('resources/sass/app.scss')': merupakan directive blade asli dari Laravel dan digunakan untuk mengintegrasikan Vite untuk pengembangan frontend dengan cepat.

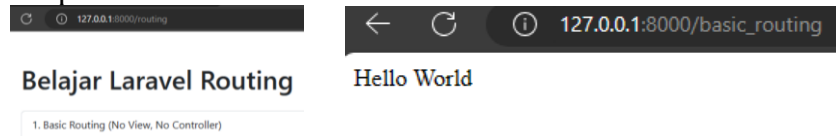
2. Praktik Basic Routing

Source code

```
Route::get('/basic_routing', function() {  
    return 'Hello World';  
});
```

```
<a href="{{ url('/basic_routing') }}" class="list-group-item list-group-item-action"> Basic Routing (No View, No Controller) </a>
```

Output:



Analisa:

Diawali dengan mendefinisikan sebuah route menggunakan HTTP dengan method GET, agar route dapat diakses melalui URL '/basic_routing'. Ketika route diakses dengan dipanggil sebuah fungsi anonim yang mengembalikan dengan string 'Hello World'. Terdapat hyperlink dalam HTML. Dalam URL tersebut dapat diklik yang akan mengarahkan ke route yang sesuai. Kemudian menambahkan styling dengan menampilkan hyperlink sebagai item dari sebuah list grup berasal dari framework CSS seperti Bootstrap.

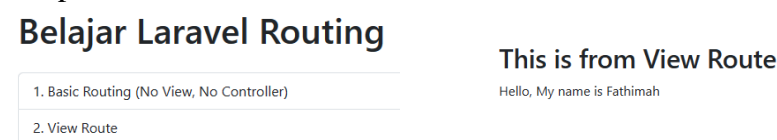
3. Praktik View Route

Source code

```
Route::view('/view_route', 'view_route');
Route::view('/view_route', 'view_route', ['name' => 'Fathimah']);
```

```
<a href="{{ url('/view_route') }}" class="list-group-item list-group-item-action"> View Route </a>
```

Output:



Analisa:

Diawali dengan mendefinisikan sebuah route menggunakan method 'Route::view()' dalam Laravel. route dapat terhubung melalui URL '/view_route' tanpa melalui fungsi method controller. Terdapat hyperlink dalam HTML. Dalam URL tersebut dapat diklik yang akan mengarahkan ke route yang sesuai. Kemudian menambahkan styling dengan menampilkan hyperlink sebagai item dari sebuah list grup berasal dari framework CSS seperti Bootstrap. Teks <a> dapat menampilkan sebagai hyperlink dalam 'View Route'.

4. Praktik Controller Route

Source code

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

```
<a href="{{ url('/controller_route') }}" class="list-group-item list-group-item-action"> Controller Route </a>
```

Output:

Belajar Laravel Routing

This is from Controller

- | |
|---|
| 1. Basic Routing (No View, No Controller) |
| 2. View Route |
| 3. Controller Route |

Analisa:

Diawali dengan mendefinisi sebuah route menggunakan method GET. Route dapat terhubung melalui URL '/controller_route'. Saat URL diakses, dengan memanggil method 'index' dari controller. Terdapat hyperlink dalam HTML. Dalam URL tersebut dapat diklik yang akan mengarahkan ke route yang sesuai. Kemudian menambahkan styling dengan menampilkan hyperlink sebagai item dari sebuah list grup berasal dari framework CSS seperti Bootstrap. Teks <a> dapat menampilkan sebagai hyperlink dalam 'Controller Route'.

5. Praktik Redirect Route

Source code

```
Route::redirect('/', '/routing');
```

```
<a href="{{ url('/') }}" class="list-group-item list-group-item-action">  
Redirect Route </a>
```

Output:

Belajar Laravel Routing

- | |
|---|
| 1. Basic Routing (No View, No Controller) |
| 2. View Route |
| 3. Controller Route |
| 4. Redirect Route |

Analisa:

Diawali dengan mengakses URL root ('/') yang akan diarahkan otomatis ke URL '/routing'. Menambahkan attribute 'href' yang di tujuan URL root akan memicu pengalihan (redirect). Terdapat hyperlink dalam HTML. Dalam URL tersebut dapat diklik yang akan mengarahkan ke route yang sesuai. Kemudian menambahkan styling dengan menampilkan hyperlink sebagai item dari sebuah list grup berasal dari framework CSS seperti Bootstrap. Teks <a> dapat menampilkan sebagai hyperlink dalam 'Redirect Route'.

6. Praktik Route Parameter (Required Parameter)

Source code

```
Route::get('/user/{id}', function($id) {  
    return "User Id: ".$id;  
});  
Route::get('/posts/{post}/comments/{comment}', function($postId, $commentId) {  
    return "Post Id: ".$postId.", Comment Id: ".$commentId;  
});
```

```
<a href="{{ url('/user/12345') }}" class="list-group-item list-group-item-action"> Route Parameter (Required Parameter) - 1 </a>  
<a href="{{ url('/posts/01/comments/20') }}" class="list-group-item list-group-item-action"> Route Parameter (Required Parameter) - 2 </a>
```

Output:

Belajar Laravel Routing

User Id: 12345

- | |
|---|
| 1. Basic Routing (No View, No Controller) |
| 2. View Route |
| 3. Controller Route |
| 4. Redirect Route |
| 5. Route Parameter (Required Parameter) - 1 |

Analisa:

Diawali sebuah route dengan parameter '{id}' di Laravel yang menunjukkan bahwa route akan permintaan untuk URL adalah nilai dinamis yang akan menjadi nilai dari parameter 'id'. URL '/user/12345' akan diakses dengan menghubungkan route akan mengembalikan nilai '12345' sebagai nilai dari parameter 'id'. Route memiliki dua parameter '{post}' dan '{comment}'. Terdapat hyperlink dalam HTML. Dalam URL tersebut dapat diklik yang akan mengarahkan ke route yang sesuai. Kemudian menambahkan styling dengan menampilkan hyperlink sebagai item dari sebuah list grup berasal dari framework CSS seperti Bootstrap. Teks <a> dapat menampilkan sebagai hyperlink.

7. Praktik Route Parameter (Optional Parameter)

Source code

```
Route::get('username/{name?}', function($name = null) {  
    return 'Username: '.$name;  
});
```

```
<a href="{{ url('/username') }}" class="list-group-item list-group-item-  
action"> Route Parameter (Optional Parameter) </a>
```

Output:

Belajar Laravel Routing

Post Id: 01, Comment Id: 20

- | |
|---|
| 1. Basic Routing (No View, No Controller) |
| 2. View Route |
| 3. Controller Route |
| 4. Redirect Route |
| 5. Route Parameter (Required Parameter) - 1 |
| 6. Route Parameter (Required Parameter) - 2 |

Analisa:

Diawali sebuah route dengan parameter '{name}' di Laravel yang menunjukkan bahwa parameter ini merupakan opsional. Ketika URL '/username' dijalankan, maka Laravel akan mengeksekusi fungsi yang sedang terhubung dengan route. Apabila tidak ada nilai yang diberikan, maka menampilkan nilai default 'null'.

'/user/12345' akan diakses dengan menghubungkan route akan mengembalikan nilai '12345' sebagai nilai dari parameter 'id'. Route memiliki dua parameter '{post}' dan '{comment}'. Terdapat hyperlink dalam HTML. Dalam URL tersebut dapat diklik yang akan mengarahkan ke URL '/username' yang sesuai.

8. Praktik Route With Regular Expression Constraints

Source code

```
Route::get('/title/{title}', function($title) {  
    return "Title: ".$title;  
})->where('title', '[A-Za-z]+');
```

```
<a href="{{ url('/title/this-is-my-title') }}" class="list-group-item list-  
group-item-action"> Route With Regular Expression Constraints </a>
```

Output:

Belajar Laravel Routing

This is Fallback Route

1. Basic Routing (No View, No Controller)
2. View Route
3. Controller Route
4. Redirect Route
5. Route Parameter (Required Parameter) - 1
6. Route Parameter (Required Parameter) - 2
7. Route With Regular Expression Constraints

Analisa:

Dalam source code diatas terdapat parameter '{title}' yang akan menangani permintaan ke URL dalam nilai dinamis yang akan menjadi nilai dari parameter 'title' dan terdapat constraint yang diterapkan pada parameter yang mengharuskan nilai tersebut terdiri dari huruf-huruf besar atau tanpa karakter khusus atau angka. Terdapat hyperlink dalam HTML. Dalam URL tersebut dapat diklik yang akan mengarahkan ke URL '/title/this-is-my-title' yang sesuai dengan route yang telah didefinisikan diatas.

9. Praktik Named Route

Source code

```
Route::get('/profile/{profileId}', [RouteController::class, 'profile'])-  
>name('profileRouteName');
```

```
public function profile($profileId) {  
    return "This is Profile from Controller, profile id: ".$profileId; }  

```

```
<a href="{{ route('profileRouteName', ['profileId' => '123']) }}" class="list-  
group-item list-group-item-action"> Named Route </a>
```

Output:

Belajar Laravel Routing

This is Profile from Controller, profile id: 123

1. Basic Routing (No View, No Controller)
2. View Route
3. Controller Route
4. Redirect Route
5. Route Parameter (Required Parameter) - 1
6. Route Parameter (Required Parameter) - 2
7. Route With Regular Expression Constraints
8. Named Route

Analisa:

Diawali dengan mendefinisikan sebuah route menggunakan HTTP method GET. Route dapat terhubung melalui URL '/profile/{profileId}' merupakan parameter dinamis yang akan digunakan sebagai identifikasi profil. Terdapat hyperlink dalam HTML. Dalam URL tersebut dapat diklik yang akan mengarahkan ke route yang sesuai. Kemudian menambahkan styling dengan menampilkan hyperlink sebagai item dari sebuah list grup berasal dari framework CSS seperti Bootstrap.

10. Praktik Route Priority

Source code

```
// Route::get('/route_priority/{rpId}', function($rpId) {  
//     return "This is Route One";  
// });  
Route::get('/route_priority/user', function() {  
    return "This is Route 1";  
});  
Route::get('/route_priority/user', function() {  
    return "This is Route 2";  
});
```

```
<a href="{{ url('/route_priority/user') }}" class="list-group-item list-group-item-action"> Route Priority </a>
```

Output:

Belajar Laravel Routing

- | |
|--|
| 1. Basic Routing (No View, No Controller) |
| 2. View Route |
| 3. Controller Route |
| 4. Redirect Route |
| 5. Route Parameter (Required Parameter) - 1 |
| 6. Route Parameter (Required Parameter) - 2 |
| 7. Route With Regular Expression Constraints |
| 8. Named Route |
| 9. Route Priority |

This is Route 2

Analisa:

Terdapat dua route yang memiliki URL yang sama, yaitu '/route_priority/user'. Keduanya memiliki URL sama dan mereka memiliki prioritas yang sama. Kemudian keduanya akan menimpa route pertama hingga terakhir yang akan diambil oleh Laravel.

11. Praktik Fallback Routes

Source code

```
Route::fallback(function() {  
    return 'This is Fallback Route';  
});
```

```
<a href="{{ url('/asdqwezxc') }}" class="list-group-item list-group-item-action"> Fallback Routes </a>
```

Output:

Belajar Laravel Routing

This is Fallback Route

- | |
|--|
| 1. Basic Routing (No View, No Controller) |
| 2. View Route |
| 3. Controller Route |
| 4. Redirect Route |
| 5. Route Parameter (Required Parameter) - 1 |
| 6. Route Parameter (Required Parameter) - 2 |
| 7. Route With Regular Expression Constraints |
| 8. Named Route |
| 9. Route Priority |
| 10. Fallback Routes |

Analisa:

Dalam source code diatas fallback route akan dipicu apabila tidak ada route yang sesuai, tidak memerlukan URL spesifik. Terdapat hyperlink dalam HTML. Dalam URL tersebut dapat diklik yang akan mengarahkan ke route yang sesuai.

12. Praktik Route Groups (Route Prefixes & Route Name Prefixes)

Source code

```
Route::prefix('admin')->name('admin.')->group(function() {
    Route::get('/dashboard', function() {
        return "This is admin dashboard";
    })->name('dashboard');

    Route::get('/users', function() {
        return "This is user data on admin page";
    })->name('users');

    Route::get('/items', function() {
        return "This is item data on admin page";
    })->name('items');
});
```

```
<h6 class="mt-4">Route Groups (Route Prefixes & Route Name Prefixes)</h6>
    <div class="list-group list-group-numbered mt-4">
        <a href="{{ route('admin.dashboard') }}" class="list-group-item list-group-item-action"> Admin Dashboard </a>
        <a href="{{ route('admin.users') }}" class="list-group-item list-group-item-action"> Admin Users </a>
        <a href="{{ route('admin.items') }}" class="list-group-item list-group-item-action"> Admin Items </a>
    </div>
</div>
```

Output:

Route Groups (Route Prefixes & Route Name Prefixes)

1. Admin Dashboard
2. Admin Users
3. Admin Items

This is admin dashboard

This is user data on admin page

This is item data on admin page

Analisa:

Dalam source code diatas route yang didefinisikan di dalam grup, semua URL route yang ada di dalam grup ini akan dimulai dengan '/admin'. Dengan menetapkan awalan 'admin' berguna untuk memberikan prefix pada nama-nama route agar lebih mudah diidentifikasi dan dipanggil dalam kode. Kemudian mengelompokkan route-route yang didefinisikan di dalam blok untuk memisahkan dan mengorganisir route-route yang terkait. Ketiga route memiliki URL yang dimulai dengan '/admin' karena mereka didefinisikan di dalam grup yang memiliki prefix 'admin'. Di dalam sebuah list-group, terdapat beberapa hyperlink yang mengarah ke route-route yang telah didefinisikan sebelumnya.

13. Praktik View Route List

Output:

```
PS D:\smst 4\FREEMWORK\basic-routing> php artisan route:list

GET|HEAD / ..... Illuminate\Routing > RedirectController
ANY / ..... Illuminate\Routing > RedirectController
POST _ignition/execute-solution ignition.executeSolution > Spatie\Larave...
GET|HEAD _ignition/health-check ignition.healthCheck > Spatie\LaravelIgnitio...
POST _ignition/update-config ignition.updateConfig > Spatie\LaravelIgnitio...
GET|HEAD admin/dashboard ..... admin.dashboard
GET|HEAD admin/items ..... admin.items
GET|HEAD admin/users ..... admin.users
GET|HEAD api/user .....
GET|HEAD basic_routing .....
GET|HEAD controller_route ..... RouteController@index
GET|HEAD posts/{post}/comments/{comment} .....
GET|HEAD profile/{profileId} ..... profileRouteName > RouteController@profile
GET|HEAD route_priority/user .....
GET|HEAD routing .....
GET|HEAD sanctum/csrf-cookie sanctum.csrf-cookie > Laravel\Sanctum > CsrfCoo...
GET|HEAD title/{title} .....
GET|HEAD user/{id} .....
GET|HEAD username/{name?} .....
GET|HEAD view_route .....
GET|HEAD {fallbackPlaceholder} .....

Showing [21] routes

PS D:\smst 4\FREEMWORK\basic-routing> |
```

Analisa:

Merupakan konsep yang berkaitan dengan framework web digunakan untuk melihat daftar seluruh route yang telah didefinisikan dalam aplikasi web. Dengan menggunakan fitur view route list akan memudahkan melihat semua route yang tersedia dalam aplikasi seperti URL, method HTTP.

14. Praktik Route Caching

Output:

```
PS D:\smst 4\FREEMWORK\basic-routing> php artisan route:cache
```

```
INFO Routes cached successfully.
```

```
PS D:\smst 4\FREEMWORK\basic-routing> 
```

```
PS D:\smst 4\FREEMWORK\basic-routing> php artisan route:clear
```

```
INFO Route cache cleared successfully.
```

```
PS D:\smst 4\FREEMWORK\basic-routing> 
```

Analisa:

Dengan menggunakan route caching dapat meningkatkan perform aplikasi laravel, mengurangi beban pada server, dan penyimpanan dalam memori akan lebih cepat. Dapat memperbarui cache route setiap kali ada perubahan dalam definisi route berguna untuk memastikan konsisten dalam aplikasi.

TUGAS

Source code

```
<a href="{{ url('/fath') }}" class="list-group-item list-group-item-action">
Tugas Bootstrap </a>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Day6</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha2/dist/css/bootstrap.min.css">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.6.0/font/bootstrap-icons.css">
  <link rel="stylesheet" href="../styles/style.css">
</head>
<body>
  <nav class="navbar navbar-light bg-light">
    <div class="container">
      <a class="navbar-brand" href="#">
        
      </a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
```

```

        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item">
                    <a class="nav-link active" aria-current="page"
href="#">Home</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="#">About</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="#">Lagu</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="#">Album</a>
                </li>
            </ul>
        </div>
    </div>
</nav>
<div class="bg-light mt-5" id="main">
    <!-- Container -->
    <div class="container py-5 px-4">
        <div class="row">
            <div class="col-md-5 order-md-2">
                
            </div>
            <div class="col-md-7 order-md-1">
                <h1 class="mt-4 display-3"><b> HELLO MY DAY!!!
</b></h1>

                <p class="fs-5 mt-3">
                    My Day nama penggemar fandom dari Day6 merupakan band
pop rock asal korea selatan, dibawah naungan JYP Entertainment.
                    Band ini terdiri dari lima anggota yaitu: Sungji,
Young K, Wonpil, Dowoon, dan jae. Mereka debut pada 7 September 2015.
                    Lagu mereka sangat diterima oleh pendengar yang tidak
menyukai lagu kpop dengan judul lagu Congratulations.
                </p>
                <div class="row">
                    <div class="d-flex flex-column flex-md-row">

                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

<div id="footer" class="bg-light py-5">
  <!-- Container -->
  <div class="container py-5 px-4">
    <div class="row">
      <div class="col-lg-3 mb-5">
        <a href="#" class="logo text-decoration-none">
          <div class="d-flex">
            
            <div class="fs-5 ms-2 text-black">DAY 6</div>
          </div>
        </a>
      <div class="mt-2 text-muted">
        <small>Code licensed <a href="#" class="text-black">MIT</a>, docs <a
href="#" class="text-black">CC BY 3.0</a>.</small>
      </div>
      <div class="mt-2 text-muted">
        <small>Currently v5.3.0-alpha2.</small>
      </div>
    </div>
    <div class="col-6 col-lg-2 offset-lg-1 mb-5">
      <h5>Lagu</h5>
      <ul class="list-unstyled">
        <li class="mb-2"><a href="#">Congratulations/a></li>
        <li class="mb-2"><a href="#">You Were Beautiful/a></li>
        <li class="mb-2"><a href="#">For me/a></li>
        <li class="mb-2"><a href="#">Afraid/a></li>
      </ul>
    </div>
    <div class="col-6 col-lg-2 mb-5">
      <h5>Album</h5>
      <ul class="list-unstyled">
        <li class="mb-2"><a href="#">The Books of Us: Gravity/a></li>
        <li class="mb-2"><a href="#">The Book of Us: The Demon/a></li>
        <li class="mb-2"><a href="#">The Book of Us: Negentropy/a></li>
      </ul>
    </div>
  </div>
</div>
</div>
</div>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha2/dist/js/bootstrap.bundle
.min.js"></script>
</body>
</html>

```

Output:

Belajar Laravel Routing

Tugas Bootstrap

1. Basic Routing (No View, No Controller)

2. View Route

3. Controller Route

4. Redirect Route

5. Route Parameter (Required Parameter) - 1

6. Route Parameter (Required Parameter) - 2

7. Route Parameter (Optional Parameter)

8. Route With Regular Expression Constraints

9. Named Route

10. Route Priority

11. Fallback Routes

