




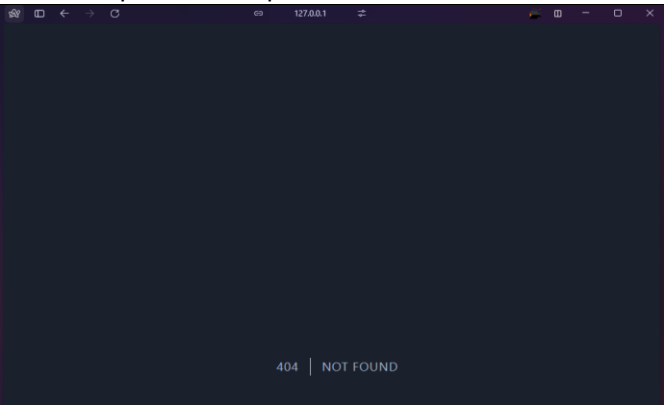
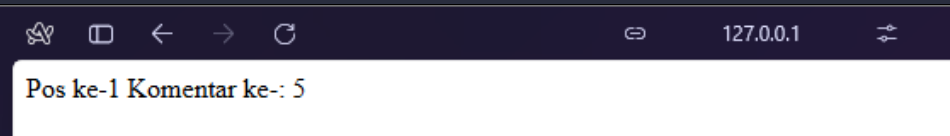
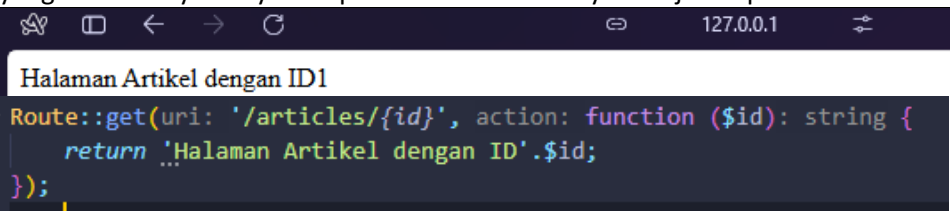
LAPORAN JOBSHEET 2  
PEMROGRAMAN WEB LANJUT



Disusun Oleh :  
RASENDRIYA DAFA SETIADI  
NIM. 2341720125  
TI-2E / 25

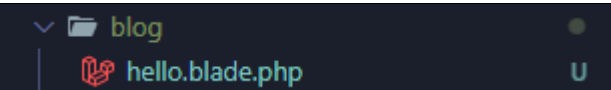


D-IV TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG  
2024/2025

1.	Basic Routing
b.	<pre>Route::get(uri: '/hello', action: function (): string {     return " Hello World!"; });</pre> <p>Fungsi dari kode adalah untuk menampilkan pesan “hello world” Ketika kita memasukkan url /hello</p>
c.	 <p>Ini adalah hasil Ketika kita mengakses /hello</p>
d.	<pre>Route::get(uri: '/world', action: function (): string {     return "World"; });</pre> <p>Fungsi dari kode adalah untuk menampilkan pesan “world” Ketika kita memasukkan url /world</p>
e.	 <p>Ini adalah hasil Ketika kita mengakses /hello</p>
f.	<pre>Route::get(uri: '/', action: function (): string {     // return view('welcome');     return "Selamat Datang!"; });</pre>  <p>Untuk membuat route dengan pesan “Selamat Datang!” kita hanya perlu mengganti return dari yang awalnya return view menjadi return “Selamat Datang!”</p>
g.	<pre>Route::get(uri: '/about', action: function (): string {     return "Rasenddriya Dafa Setiadi - 2341720125"; });</pre>  <p>Ini adalah route /about beserta hasil Ketika kita mengakses url tersebut.</p>

2.	Route Parameters
a.	<pre>Route::get(uri: '/user/{name}', action: function (\$name): string {     return 'Nama saya '.\$name; });</pre>
b.	 <p>Hasil dari kode diatas adalah contoh dari penggunaan parameter pada route Laravel yang pada percobaan ini parameter yang dimasukkan adalah nama yang nanti akan ditangkap dan ditampilkan di output return.</p>
c.	 <p>Hasil yang ditampilkan akan error karena kita tidak memasukkan parameter nama pada url.</p>
d.	<pre>Route::get(uri: '/posts/{post}/comments/{comment}', action: function (\$postId, \$commentId): string {     return 'Pos ke-' . \$postId . " Komentar ke-: " . \$commentId; });</pre>
e.	 <p>Sama seperti percobaan sebelumnya yang membedakan adalah penggunaan parameternya yang sebelumnya hanya 1 di percobaan sebelumnya menjadi 2 parameter.</p>
f.	 <pre>Route::get(uri: '/articles/{id}', action: function (\$id): string {     return 'Halaman Artikel dengan ID' . \$id; });</pre>
3.	Optional Parameters

a.	<pre>Route::get(uri: '/user/{name?}', action: function (\$name=null): string {     return 'Nama saya '.\$name; });</pre>	
b.	 Nama saya	Hasil dari kode diatas adalah contoh penggunaan dari parameter opsional yang sebelumnya error Ketika tidak ada input menjadi tidak error.
c.	 Nama saya daoa	Hasil yang diberikan tetap sama dan tidak ada error Ketika kita memasukkan parameter.
d.	 Nama saya John	Hasil yang ditampilkan adalah Nama Saya John karena pada default parameter di set dengan nama "John".
4.	Route Name	
a.	<pre>Route::get(uri: '/user/profile', action: function(): void {     // });-&gt;name(name: 'profile');</pre>	
b.	 Nama saya profile	
5.	Membuat Controller	
a.		
c.	<pre>&lt;?php namespace App\Http\Controllers; use Illuminate\Http\Request;  2 references   0 implementations class WelcomeController extends Controller {     1 reference   0 overrides   Tabnine   Edit   Test   Explain   Document     public function hello(): string {         return 'Hello World';     } }</pre>	
d.	<pre>Route::get(uri: '/hello', action: [WelcomeController::class, 'hello']);</pre>	
e.	 Hello World	Hasil yang ditampilkan adalah isi dari function hello di WelcomeController.php

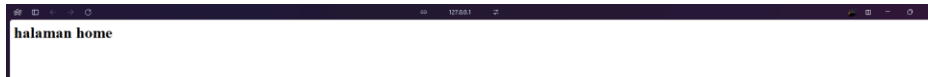
f.	<pre> class PageController extends Controller {     1 reference   0 overrides   Tabnine   Edit   Test   Explain   Document     public function index(): string{         return "Selamat Datang";     }     1 reference   0 overrides   Tabnine   Edit   Test   Explain   Document     public function about(): string{         return "Rasendriya Dafa Setiadi - 2341720125";     }     1 reference   0 overrides   Tabnine   Edit   Test   Explain   Document     public function articles(\$id): string{         return 'Halaman Artikel dengan ID' . \$id;     } }  Route::get(uri: '/', action: [PageController::class,'index']); Route::get(uri: '/about', action: [PageController::class,'about']); Route::get(uri: '/articles/{\$id}', action: [PageController::class,'articles']); </pre>
g.	<pre> Route::get(uri: '/', action: [PageController::class,'index']); Route::get(uri: '/about', action: [aboutController::class,'index']); Route::get(uri: '/articles/{id}', action: [ArticleController::class,'index']); </pre>
6.	Resource Controller
a.	INFO Controller [C:\coolyeah\SEM4\weblanjut\PHL_2024\app\Http\Controllers\PhotoController.php] created successfully.
b.	<pre> Route::resource(name: 'photos', controller: PhotoController::class); </pre>
c.	<pre> GET HEAD / ..... PageController@index POST _ignition/execute-solution ..... Ignition\executeSolution &gt; Spatie\LaravelIgnition &gt; ExecuteSolutionController GET HEAD _ignition/health-check ..... Ignition\healthCheck &gt; Spatie\LaravelIgnition &gt; HealthCheckController POST _ignition/update-config ..... Ignition\$updateConfig &gt; Spatie\LaravelIgnition &gt; UpdateConfigController GET HEAD api/user .....  GET HEAD articles/{id} ..... ArticleController@index GET HEAD here ..... WelcomeController@hello ANY photos ..... Illuminate\Routing &gt; RedirectController GET HEAD photos/create ..... photos.store &gt; PhotoController@store GET HEAD photos/{photo} ..... photos.create &gt; PhotoController@create PUT PATCH photos/{photo} ..... photos.show &gt; PhotoController@show DELETE photos/{photo} ..... photos.update &gt; PhotoController@update GET HEAD photos/{photo}/edit ..... photos.destroy &gt; PhotoController@destroy GET HEAD posts/{post}/comments/{comment} ..... photos.edit &gt; PhotoController@edit GET HEAD sanctum/csrf-cookie ..... sanctum.csrf-cookie &gt; Laravel\Sanctum &gt; CsrfCookieController@show GET HEAD user/profile .....  GET HEAD (account).example.com/user/{id} .....  GET HEAD user/{name} .....  GET HEAD welcome .....  GET HEAD world .....  </pre>
d.	<pre> Route::resource(name: 'photos', controller: PhotoController::class)-&gt;only(methods: ['index', 'show']); Route::resource(name: 'photos', controller: PhotoController::class)-&gt;except(methods: [     'create', 'store', 'update', 'destroy' ]); </pre>
7.	View
a.	<pre> &lt;html&gt; &lt;body&gt;     &lt;h1&gt;Hello, {{ \$name }}&lt;/h1&gt; &lt;/body&gt; &lt;/html&gt; </pre>
b.	<pre> Route::get(uri: '/greeting', action: function (): Factory View {     return view(view: 'hello', data: ['name' =&gt; 'Andi']); }); </pre>
c.	 <p>Web akan menampilkan Hello,Andi karena pada routing /greeting selain kita mengarahkan ke view hello, kita juga memberikan data berupa name yang diisi dengan Andi.</p>

8.	View dalam direktori
a.b.	
c.	<pre>Route::get(uri: '/greeting', action: function (): Factory View {     return view(view: 'blog.hello', data: ['name' =&gt; 'Andi']); });</pre>
d.	 <p>Output yang dihasilkan sama dengan sebelumnya karena kita hanya mengubah direktori dari file dan routingnya saja, tidak ada data atau view yang di edit.</p>
9.	Menampilkan View dari Controller
a.	<pre>class WelcomeController extends Controller {     1 reference   0 overrides   Tabnine   Edit   Test   Explain   Document     public function hello(): string{         return('Hello World');     }     0 references   0 overrides   Tabnine   Edit   Test   Explain   Document     public function greeting(): Factory View{         return view(view: 'blog.hello', data: ['name' =&gt; 'Andi']);     } }</pre>
b.	<pre>Route::get(uri: '/greeting', action: [WelcomeController::class,'greeting']);</pre>
c.	 <p>Hasil yang ditampilkan tetap sama karena kita hanya memindahkan return view dari yang awalnya direct menjadi terpisah dalam controller.</p>
10.	Meneruskan data ke view
a.	<pre>class WelcomeController extends Controller {     1 reference   0 overrides   Tabnine   Edit   Test   Explain   Document     public function hello(): string{         return('Hello World');     }     1 reference   0 overrides   Tabnine   Edit   Test   Explain   Document     public function greeting(): View{         return view(view: 'blog.hello')             -&gt;with(key: 'name',value: 'Andi')             -&gt;with(key: 'occupation',value: 'Astronaut');     } }</pre>
b.	<pre>&lt;html&gt; &lt;body&gt;     &lt;h1&gt;Hello, {{ \$name }}&lt;/h1&gt;     &lt;h1&gt;You are {{ \$occupation }}&lt;/h1&gt; &lt;/body&gt; &lt;/html&gt;</pre>

1.

## Soal Praktikum

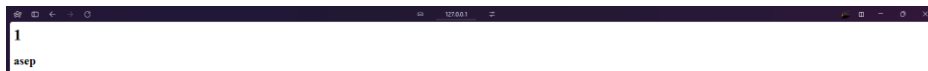
## Halaman Home



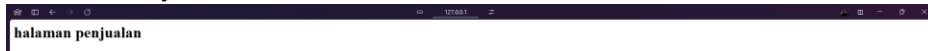
## Halaman Products



## Halaman User



## Halaman Penjualan



## Routes

```
<?php
use App\Http\Controllers\HomeController;
use App\Http\Controllers\PenjualanController;
use App\Http\Controllers\ProductsController;
use App\Http\Controllers\UserController;
use Illuminate\Support\Facades\Route;

/*
|-----
| Web Routes
|-----
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider and all of them will
| be assigned to the "web" middleware group. Make something great!
|
*/

Route::get('/', [HomeController::class,'index']);
Route::get('/penjualan', [PenjualanController::class,'index']);
Route::get('/user/{id}/name/{name}', [UserController::class,'index']);
Route::prefix('category')->group(function () { void {
    Route::get('/food-beverage', [ProductsController::class, 'food']);
    Route::get('/beauty-health', [ProductsController::class, 'beauty']);
    Route::get('/home-care', [ProductsController::class, 'home']);
    Route::get('/baby-kid', [ProductsController::class, 'baby']);
});
});
```

## Controller

```
class HomeController extends Controller
{
    1 reference | 0 overrides | Tabnine | Edit | Test | Explain | Document
    public function index(): Factory\View{
        return view(view: 'home');
    }
}

class ProductsController extends Controller
{
    1 reference | 0 overrides | Tabnine | Edit | Test | Explain | Document
    public function food(): Factory\View{
        return view(view: 'products.fnb');
    }
    1 reference | 0 overrides | Tabnine | Edit | Test | Explain | Document
    public function beauty(): Factory\View{
        return view(view: 'products.bhealth');
    }
    1 reference | 0 overrides | Tabnine | Edit | Test | Explain | Document
    public function home(): Factory\View{
        return view(view: 'products.hcare');
    }
    1 reference | 0 overrides | Tabnine | Edit | Test | Explain | Document
    public function baby(): Factory\View{
        return view(view: 'products.bkid');
    }
}

class PenjualanController extends Controller
{
    1 reference | 0 overrides | Tabnine | Edit | Test | Explain | Document
    public function index(): Factory\View {
        return view(view: 'penjualan');
    }
}

class UserController extends Controller
{
    1 reference | 0 overrides | Tabnine | Edit | Test | Explain | Document
    public function index($id,$name): View{
        return view(view: "user")
        ->with("id", $id)
        ->with("name", $name);
    }
}
```

Link Github : <https://github.com/lostboiii/weblanjut/tree/main/POS>