ROUTING & BLADE

Modul 2

Framework-based Programming

ROUTING

Route atau **Routing** berperan sebagai penghubung antara user dengan keseluruhan framework. Dalam Laravel, setiap alamat web yang kita ketik di web browser akan melewati *route* terlebih dahulu. *Route*-lah yang menentukan kemana proses akan dibawa, apakah ke Controller atau ke View.

Untuk mengakses routing pada Laravel kita dapat membuka file web.php yang terletak di :

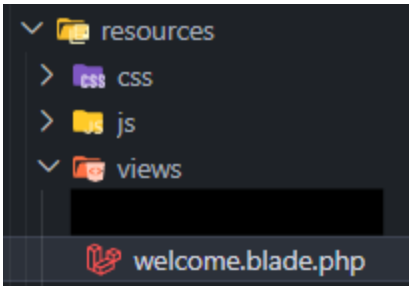
```
routes > web.php
```

Berikut adalah tampilannya :

routes > web.php

```php
<?php

use Illuminate\Support\Facades\Route;

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

Diatas merupakan syntax standar route pada laravel. *'view('welcome');* merupakan nama file blade yang digunakan oleh route, dalam contoh kasus ini nama file nya adalah 'welcome.blade.php'.



Passing Data

Pada Laravel kita dapat menambahkan data static dengan sebuah array asosiatif di fungsi dari view pada *route/web.php* untuk mengirim data ke suatu halaman. Selain itu kita dapat passing data dari *return* secara langsung seperti contoh dibawah.

routes > web.php

```php
Route::get('/', function () { // Passing data
    $data = [
        'id' => 1,
        'nama' => 'Alianur',
        'nim' => '22091106045'
    ];
    return view('welcome', [
        'data' => $data,
        'title' => 'welcome'
    ]);
});
```

resources > views > welcome.blade.php

```
    </head>
        ........
        <title>{{ $title }}</title>
        .......
    </head>
    <body class="flex flex-col min-h-screen items-center gap-2 ">
        <p>id : {{$data['id']}}</p>
        <p>nama : {{$data['nama']}}</p>
        <p>nim : {{$data['nim']}}</p>
    </body>
```

Dan hasilnya akan menampilkan $data dan $title seperti dibawah :

Named Route

Pada Laravel kita dapat memanggil route untuk mengakses halaman dengan menggunakan path atau URL, namun cara penggunaan terbaiknya adalah menggunakan **Named Route,** dimana route diberi nama agar lebih mudah diingat dan dipanggil.

routes > web.php

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

Daripada memanggilnya seperti ini:

```
<a href="/welcome"></a> 😣
```

Kita dapat memanggilnya seperti ini

```
<a href="{{ route('welcome') }}"></a> 😀👍
```

Route Parameter

Pada Laravel, route juga memiliki yang namanya ‘parameter’. Parameter ini dapat kita manfaatkan salah satunya untuk menampilkan halaman dengan data yang berbeda dengan cara menangkap ‘id’ data misalnya , seperti :

routes > web.php

```
Route::get('/orang/{id}', function ($id) {
    return view('orang', [
        'id' => $id
    ]);
})->name('orang.detail');
```

Yang dapat diakses dengan :

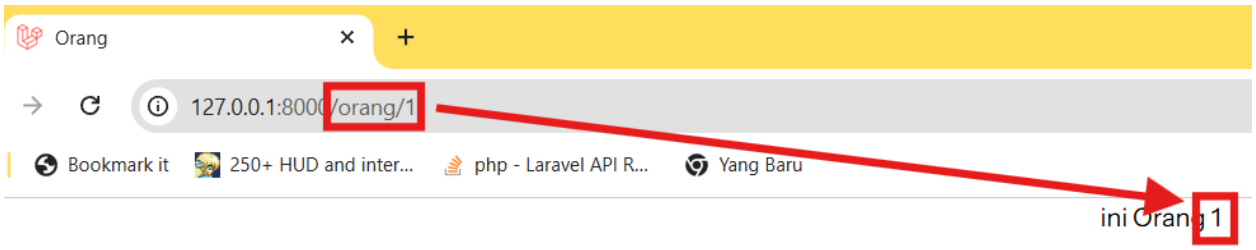
resources > views > welcome.blade.php

```
                                {{-- v ini Parameternya --}}
<a href="{{ route('orang.detail', 1) }}"></a>
{{-- Atau --}}                          {{-- v ini Parameternya --}}
<a href="{{ route('orang.detail', ['id' => 1]) }}"></a>
```

resources > views > orang.blade.php

```
    <body class="flex flex-col min-h-screen items-center gap-2 ">
        ini Orang {{ $id ?? 'tanpa id' }}
    </body>
```

Yang akan menampilkan :



Selain itu kita juga dapat membuat parameter pada route menjadi opsional :

routes > web.php

```
Route::get('/orang/{id?}', function (?int $id = null) {
    return view('orang', [
        'id' => $id
    ]);
})->name('orang.detail.optional');
```

resources > views > welcome.blade.php

```
<a href="{{ route('orang.detail.optional', ['id' => 3]) }}"></a>
<a href="{{ route('orang.detail.optional') }}"></a>
```

BLADE TEMPLATING ENGINE

Blade merupakan template engine bawaan Laravel. Berbeda dengan template engine populer lainnya, Blade tidak membatasi kalian untuk tetap menggunakan kode PHP yang asli. Fun Fact, Semua template/file Blade yang dibuat akan dikompilasi ke dalam kode asli PHP. Untuk menggunakan Balde, cukup buat sebuah file dengan nama **namaFIle.blade.php** di dalam folder **resources/views**. Atau gunakan :

```
php artisan make:view namaFile
```

Syntax Dasar

Gunakan kurung kurawal ganda **{{}}** untuk memanggil variabel.

```
<p>{{ $variable }}</p>
```

Tag PHP

Blade juga menyediakan tag PHP yang lebih ringkas, bahkan sudah ada intellisensenya.

```
@php
    $variable = 0;
@endphp
```

Escaped Character

Secara default {{ }} sudah di parser ke htmlspecialcharacter() untuk menghindari XSS atau injeksi kode. Gunakan syntax seperti dibawah jika kalian tidak menginginkan hal itu.

```
@php
    $content = '<script>alert("XSS!")</script><b>Bold Text</b>';
@endphp
<p>Escaped: {{ $content }}</p>
<p>Unescaped: {!! $content !!}</p> <----------------
<p>Blade asli: @{{ content }}</p>
```

If Else dan Switch Case

Blade juga menyediakan cara mudah untuk melakukan percabangan pada sebuah program.

```
@php
    $text = '';
@endphp

@if ($text == '1')
    <p class="">if1</p>
@elseif ($text == '2')
    <p class="">elseif 2</p>
@else
    <p class="">else {{ $text }}</p>
@endif

{{-- Switch Case --}}
@switch($text)
    @case('1')
        <p class="">case 1</p>
        @break
    @case('2')
        <p class="">case 2</p>
        @break
    @default
        <p class="">default {{ $text }}</p>
@endswitch
```

Loop

Blade juga menyediakan cara mudah untuk melakukan perulangan pada sebuah program.

```
{{-- For Loop --}}
@php
    $array = ['kucing', 'anjing', 'jerapah'];
@endphp
<h1 class="">for</h1>
@for ($i = 0; $i < count($array); $i++)
    <div class="">
        {{ $array[$i] }}
    </div>
@endfor

{{-- While Loop --}}
<h1 class="">while</h1>
@php
    $i = 0;
@endphp
@while ($i < count($array))
    <div class="">
        {{ $array[$i] }}
    </div>
    @php
        $i++;
    @endphp
@endwhile

{{-- Foreach Loop --}}
<h1 class="">foreach</h1>
@foreach ($array as $item)
    <div class="">
        {{ $item }}
    </div>
@endforeach
```

```
        {{-- Forelse Loop --}}
        <h1 class="">forelse</h1>
        @forelse ($array as $item)
            <div class="">
                {{ $item }}
            </div>
        @empty
            <div class="">
                Array kosong
            </div>
        @endforelse

        {{-- $Loop (Loop Variable --}}
        <h1 class="">foreach + $loop</h1>
        @foreach ($array as $item)
            <div class="">
                {{-- $loop->iteration = nomor urut loop (mulai dari
1) --}}
                {{ $loop->iteration }}. {{ $item }}

                {{-- $loop->first = true hanya pada iterasi pertama
--}}
                @if ($loop->first) (pertama) @endif

                {{-- $loop->last = true hanya pada iterasi terakhir
--}}
                @if ($loop->last) (terakhir) @endif
            </div>
        @endforeach
```

Untuk lebih lengkapnya silahkan cek dokumentasi resmi dari laravel mengenai Blade di link berikut

Dokumentasi Blade

Layout & Component

Layout digunakan untuk membuat kerangka utama (misalnya berisi navbar, footer, dan struktur dasar) yang bisa dipakai ulang di banyak halaman. **Component** bagian kecil dan terpisah dari tampilan (seperti footer atau navbar) yang bisa dipanggil di berbagai halaman agar kode lebih rapi dan mudah dipelihara.

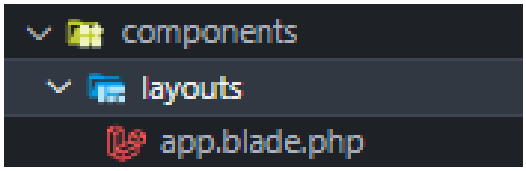
Dalam Blade, kita bisa membuat layout utama yang berisi elemen tetap seperti navbar dan footer. Dengan begitu, setiap halaman cukup memanggil layout tersebut tanpa harus menulis ulang navbar dan footer di tiap file.

Agar lebih mudah dipahami kita akan langsung praktek pembuatannya secara langsung. Jalankan command berikut

```
php artisan make:component layouts/app ⟵ namaFolder/namaFile
```

Command tersebut akan membuat folder 'components' (jika belum ada), folder '*layouts*' yang didalamnya terdapat component atau file yang bernama '*app*'

```
resources > views > components > layouts > app.blade.php
```



Pertama kita buat halaman dasar terlebih dahulu, copy code berikut dalam tag 'body' pada file 'welcome.blade.php'

Class untuk body

```
<body class="flex flex-col min-h-screen items-center gap-2">
```

resources > views > welcome.blade.php

```
        <!-- NAV -->
        <nav class="flex flex-col items-center bg-[#FFF4E8] lg:w-[720px]
w-[540px] px-4 py-2 shadow-md text-[#2A2622] border border-[#bdb5ad]
rounded-b-xl font-medium">
            <div class="flex justify-between w-full">
                <div class="w-20 font-bold">
                    <h1>FrameWork</h1>
                </div>
                <h1><a href="">Login</a></h1>
            </div>
            <ul class="flex gap-2">
                <li><a href="">Beranda</a></li>
                <li><a href="">Route</a></li>
                <li><a href="">Loop</a></li>
                <li><a href="">Percabangan</a></li>
            </ul>
        </nav>

        <!-- MAIN SECTION -->
        <main class="flex-grow flex flex-col items-center w-full">
            <section class="lg:w-[720px] w-[540px] p-2 bg-[#FFF4E8] rounded-xl
border border-[#bdb5ad] shadow-lg">
                <h1 class="font-bold text-xl text-[#2A2622] mb-2">Member</h1>
                <div class="grid lg:grid-cols-5 md:grid-cols-4 grid-cols-3
font-medium space-y-2 text-[#2A2622]">
                    {{-- @foreach ($data as $data)
                        <div class="w-16 text-center ml-auto mr-auto">
                            <img class="object-contain bg-gradient-to-b
from-amber-600 to-amber-100 rounded-lg px-0.5 pt-0.5" src="{{ $data['img'] }}"
alt="">
                            <p>{{ $data['id']. '. ' .$data['name'] }}</p>
                        </div>
                    @endforeach --}}
                    {{-- <div class="w-16 text-center ml-auto mr-auto">
                        <img class="object-contain bg-gradient-to-b
from-amber-600 to-amber-100 rounded-lg px-0.5 pt-0.5"
src="https://sunderarmor.com/GENSHIN/Characters/1/Albedo.png" alt="">
                        <p>{{ $data[0]['id']. '. ' .$data[0]['name'] }}</p>
                    </div> --}}
                </div>
            </section>
```

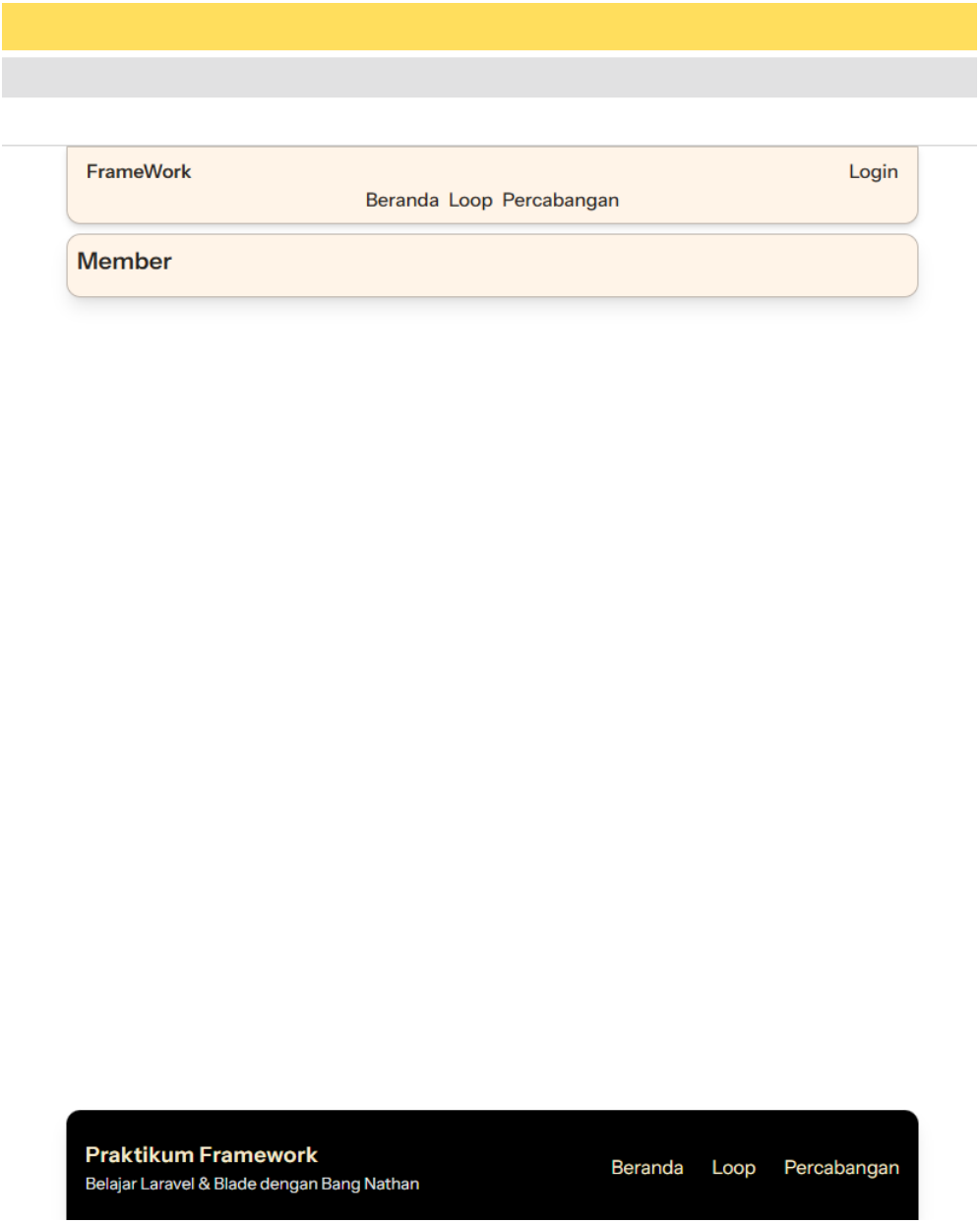
```
        </main>

        <!-- FOOTER -->
        <footer class="bg-[#000000] text-[#F8EDCE] py-6 rounded-t-xl
lg:w-[720px] w-[540px] shadow-lg">
            <div class="px-4 flex flex-col md:flex-row justify-between
items-center space-y-4 md:space-y-0">
                <div class="text-center md:text-left">
                    <h1 class="text-lg font-bold tracking-wide">Praktikum
Framework</h1>
                    <p class="text-sm text-[#E8EBEA]">Belajar Laravel & Blade
dengan Bang Nathan</p>
                </div>

                <div class="flex space-x-6">
                    <a href="#" class="hover:text-[#CAECEC] transition
ease-in-out duration-300">Beranda</a>
                    <a href="#" class="hover:text-[#CAECEC] transition
ease-in-out duration-300">Route</a>
                    <a href="#" class="hover:text-[#CAECEC] transition
ease-in-out duration-300">Loop</a>
                    <a href="#" class="hover:text-[#E2B395] transition
ease-in-out duration-300">Percabangan</a>
                </div>

            </div>
        </footer>
```

Route :

routes > web.php

```
Route::get('/', function () {
    return view('welcome', [
        "title" => "Welcome",
    ]);
})->name('welcome');
```

Maka akan muncul tampilan seperti berikut :



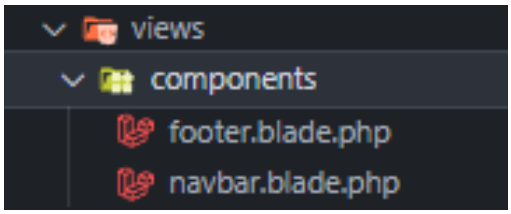
Pembuatan Component Navbar dan Footer

Pertama jalankan command ini untuk membuat component

```
php artisan make:component navbar ⟵ Nama File
```

```
php artisan make:component footer ⟵ Nama File
```

Maka akan terbuat folder 'component' yang dimana semua component berada



Kemudian kita ambil barisan code dari file '*welcome.blade.php*' tadi ke file component navbar dan footer yang sudah kita buat tadi

resources > views > components > navbar.blade.php

```
<nav class="flex flex-col items-center bg-[#FFF4E8] lg:w-[720px]
w-[540px] px-4 py-2 shadow-md text-[#2A2622] border border-[#bdb5ad]
rounded-b-xl font-medium">
    <div class="flex justify-between w-full">
        <div class="w-20 font-bold">
            <h1>FrameWork</h1>
        </div>
        <h1><a href="">Login</a></h1>
    </div>
    <ul class="flex gap-2">
        <li><a href="">Beranda</a></li>
        <li><a href="">Loop</a></li>
        <li><a href="">Percabangan</a></li>
    </ul>
</nav>
```

resources > views > components > footer.blade.php

```
<footer class="bg-[#000000] text-[#F8EDCE] py-6 rounded-t-xl
lg:w-[720px] w-[540px] shadow-lg">
    <div class="px-4 flex flex-col md:flex-row justify-between
items-center space-y-4 md:space-y-0">
        <div class="text-center md:text-left">
            <h1 class="text-lg font-bold tracking-wide">Praktikum
Framework</h1>
            <p class="text-sm text-[#E8EBEA]">Belajar Laravel & Blade
dengan Bang Nathan</p>
        </div>

        <div class="flex space-x-6">
            <a href="#" class="hover:text-[#CAECEC] transition
ease-in-out duration-300">Beranda</a>
            <a href="#" class="hover:text-[#CAECEC] transition
ease-in-out duration-300">Loop</a>
            <a href="#" class="hover:text-[#E2B395] transition
ease-in-out duration-300">Percabangan</a>
        </div>
    </div>
</footer>
```

Pembuatan Layout Global

Buka kembali file '*app.blade.php*' yang sudah kita buat tadi, lalu tambahkan :

resources > views > components > layouts > app.blade.php

```
    <!-- NAVBAR -->
    <x-navbar />

    <!-- MAIN -->
    <main class="flex-grow flex flex-col items-center w-full">
        {{ $slot }}
    </main>

    <!-- FOOTER -->
    <x-footer />
```

Penjelasan

1. *'x-namaFile'* : ini merupakan cara pemanggilan component-component yang sudah dibuat yang terletak pada folder *'resource/views/component'.*
2. *'{{ $slot }}'* : tempat menaruh konten dinamis dari halaman yang memanggil component tersebut.

Pengaplikasian Layout

Untuk menggunakan layout dan component yang sudah kita buat tadi kita buka kembali file '*welcome.blade.php*' dan tambahkan :

resources > views > welcome.blade.php

```
        <x-layouts.app :title="'Halaman Member'">
            <section class="lg:w-[720px] w-[540px] p-2 bg-[#FFF4E8]
rounded-xl border border-[#bdb5ad] shadow-lg">
                <h1 class="font-bold text-xl text-[#2A2622]
mb-2">Member</h1>
                <div class="grid lg:grid-cols-5 md:grid-cols-4
grid-cols-3 font-medium space-y-2 text-[#2A2622]">
                </div>
            </section>
        </x-layouts.app>
```

Penjelasan

1. *x-namaFolder.namaFile'* :. ini merupakan cara pemanggilan component jika terletak pada subfolder seperti '*resource/views/component/layouts/app.blade.php*'

Maka akan muncul tampilan seperti sebelumnya

Menampilkan Data

Kita dapat menampilkan data statis pada halaman sama seperti pada penjelasan '*Route*' sebelumnya

Kita akan membuat component untuk tampilan data:

```
php artisan make:component avatar
```

Ganti Route menjadi seperti berikut

routes> web.php

```
Route::get('/', function () {
    $data = [
        [
            'id' => 1,
            'name' => 'albedo',
            'img' =>
'https://sunderarmor.com/GENSHIN/Characters/1/Albedo.png'
        ],
        [
            'id' => 2,
            'name' => 'alex',
            'img' =>
'https://sunderarmor.com/GENSHIN/Characters/1/Nahida.png'
        ],
        [
            'id' => 3,
            'name' => 'aldi',
            'img' =>
'https://sunderarmor.com/GENSHIN/Characters/1/Alhaitham.png'
        ],
        [
            'id' => 4,
            'name' => 'andi',
            'img' =>
'https://sunderarmor.com/GENSHIN/Characters/1/Ayato.png'
        ]
    ];

    return view('welcome', [
        "title" => "Welcome",
        "data" => $data
    ]);
})->name('welcome');
```

Kemudian tambahkan code berikut ke file component 'avatar.blade.php'

resources > views > components > avatar.blade.php

```
@props(['src'])

<div class="w-16 text-center ml-auto mr-auto">
    <img class="object-contain bg-gradient-to-b from-amber-600
to-amber-100 rounded-lg px-0.5 pt-0.5" src="{{ $src }}" alt="">
    {{ $slot }}
</div>
```

Dan tambahkan code yang di highlight berikut ke file 'welcome.blade.php'

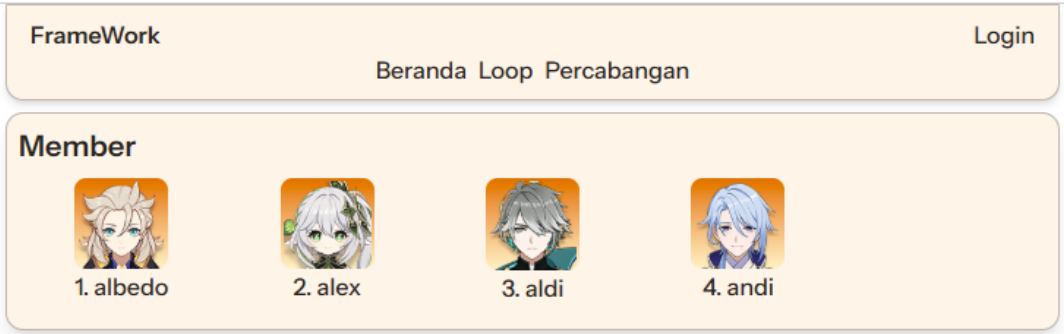
resources > views > welcome.blade.php

```
        <x-layouts.app :title="'Halaman Member'">
            <section class="lg:w-[720px] w-[540px] p-2 bg-[#FFF4E8]
                <h1 class="font-bold text-xl text-[#2A2622]...
                <div class="grid lg:grid-cols-5 md:grid-cols-4.
                    @foreach ($data as $data)
                        <x-avatar :src="$data['img']">
                            {{ $data['id'] . '. ' . $data['name'] }}
                        </x-avatar>
                    @endforeach
                </div>
            </section>
        </x-layouts.app>
```

Penjelasan

1. *@props(['variable'])* : Merupakan cara component dapat menerima data, dalam kasus ini *@props(['src'])* yang mengoper data ke *src="{{ $src }}"*
2. *:props=data* : Merupakan props dari component yang sudah kita buat pada file avatar tadi, dalam kasus ini *:src="$data['img']*

Jika sudah melakukan tahap tahap diatas maka akan muncul tampilan seperti berikut :



Cara ini merupakan cara terbaru dari laravel, jika kalian ingin menggunakan cara lama maka kalian bisa melihatnya pada dokumentasi laravel yang bisa diakses dari link berikut

Dokumentasi Laravel

REFERENSI

Routing - Laravel 12.x - The PHP Framework For Web Artisans

Blade Templates - Laravel 12.x - The PHP Framework For Web Artisans