

# **INTEGRASI WEBSITE GALLERY DENGAN LARAVEL**

**PERTEMUAN 7  
BY FAERUL SALAMUN**

---

# REVIEW 6

- Mempelajari Basic Laravel

# NEXT

- Integrasi Website Gallery Dengan Laravel

# Homepage

# CREATE PROJECT LARAVEL

---

- composer create-project --prefer-dist laravel/laravel talenthub-gallery

```
[faerulsalamun@Faerul-Salamun-MacBook-Pro Downloads % composer create-project --prefer-dist laravel/laravel gallery
Creating a "laravel/laravel" project at "./gallery"
Installing laravel/laravel (v9.1.5)
  - Downloading laravel/laravel (v9.1.5)
  - Installing laravel/laravel (v9.1.5): Extracting archive
Created project in /Users/faerulsalamun/Downloads/gallery
> @php -r "<file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 168 installs, 0 updates, 0 removals
  - Locking brick/math (0.9.3)
  - Locking dflydev/dot-access-data (v3.0.1)
  - Locking doctrine/inflector (2.0.4)
  - Locking doctrine/instantiator (1.4.1)
  - Locking doctrine/lexer (1.2.3)
  - Locking dragonmantank/cron-expression (v3.3.1)
  - Locking egulias/email-validator (3.1.2)
  - Locking facade/ignition-contracts (1.0.2)
  - Locking fakerphp/faker (v1.19.0)
  - Locking filp/whoops (2.14.5)
  - Locking fruitcake/php-cors (v1.2.0)
  - Locking graham-campbell/result-type (v1.0.4)
  - Locking guzzlehttp/guzzle (7.4.2)
  - Locking guzzlehttp/promises (1.5.1)
  - Locking guzzlehttp/psr7 (2.2.1)
  - Locking hamcrest/hamcrest-php (v2.0.1)
  - Locking laravel/framework (v9.8.1)
  - Locking laravel/sail (v1.13.9)
  - Locking laravel/sanctum (v2.15.1)
  - Locking laravel/serializable-closure (v1.1.1)
  - Locking laravel/tinker (v2.7.2)
  - Locking league/commonmark (2.3.0)
  - Locking league/config (v1.1.1)
  - Locking league/flysystem (3.0.17)
  - Locking league/nime-type-detection (1.11.0)
  - Locking mockery/mockery (1.5.0)
  - Locking monolog/monolog (2.5.0)
  - Locking myclabs深深 (v1.1.0)
  - Locking nesbot/carbon (2.57.0)
  - Locking nette/schema (v1.2.2)
  - Locking nette/utils (v3.2.7)
  - Locking nikic/php-parser (v4.13.2)
  - Locking nunomaduro/collision (v6.2.0)
  - Locking phar-io/manifest (2.0.3)
  - Locking phar-io/version (3.2.1)
  - Locking phpdocumentor/reflection-common (2.2.0)
  - Locking phpdocumentor/reflection-docblock (5.3.0)
  - Locking phpdocumentor/type-resolver (1.6.1)
  - Locking phppoption/phppoption (1.8.1)
  - Locking phpspec/prophesy (v1.15.0)
  - Locking phpunit/php-code-coverage (9.2.15)
  - Locking phpunit/php-file-iterator (3.0.6)
  - Locking phpunit/php-invoker (3.1.1)
  - Locking phpunit/php-text-template (2.0.4)
  - Locking phpunit/php-timer (5.0.3)
  - Locking phpunit/phpunit (9.5.20)
  - Locking psr/container (2.0.2)
  - Locking psr/event-dispatcher (1.0.0)
  - Locking psr/http-client (1.0.1)
  - Locking psr/http-factory (1.0.1)
  - Locking psr/http-message (1.0.1)
  - Locking psr/log (3.0.0)
  - Locking psr/simple-cache (3.0.0)
```

# CONFIG DATABASE

---

Untuk melakukan config database dapat mengganti file .env atau file config > database.php

```
⚙ .env      X
⚙ .env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:EEde+zrPbkjvgpC/Nq02Z72K1HgFT/xj3lq/roCGAC0=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=laravel
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DISK=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 MEMCACHED_HOST=127.0.0.1
26
27 REDIS_HOST=127.0.0.1
28 REDIS_PASSWORD=null
29 REDIS_PORT=6379
30
31 MAIL_MAILER=smtp
32 MAIL_HOST=mailhog
33 MAIL_PORT=1025
34 MAIL_USERNAME=null
35 MAIL_PASSWORD=null
36 MAIL_ENCRYPTION=null
37 MAIL_FROM_ADDRESS="hello@example.com"
38 MAIL_FROM_NAME="${APP_NAME}"
39
```

# CREATE TABLE

---

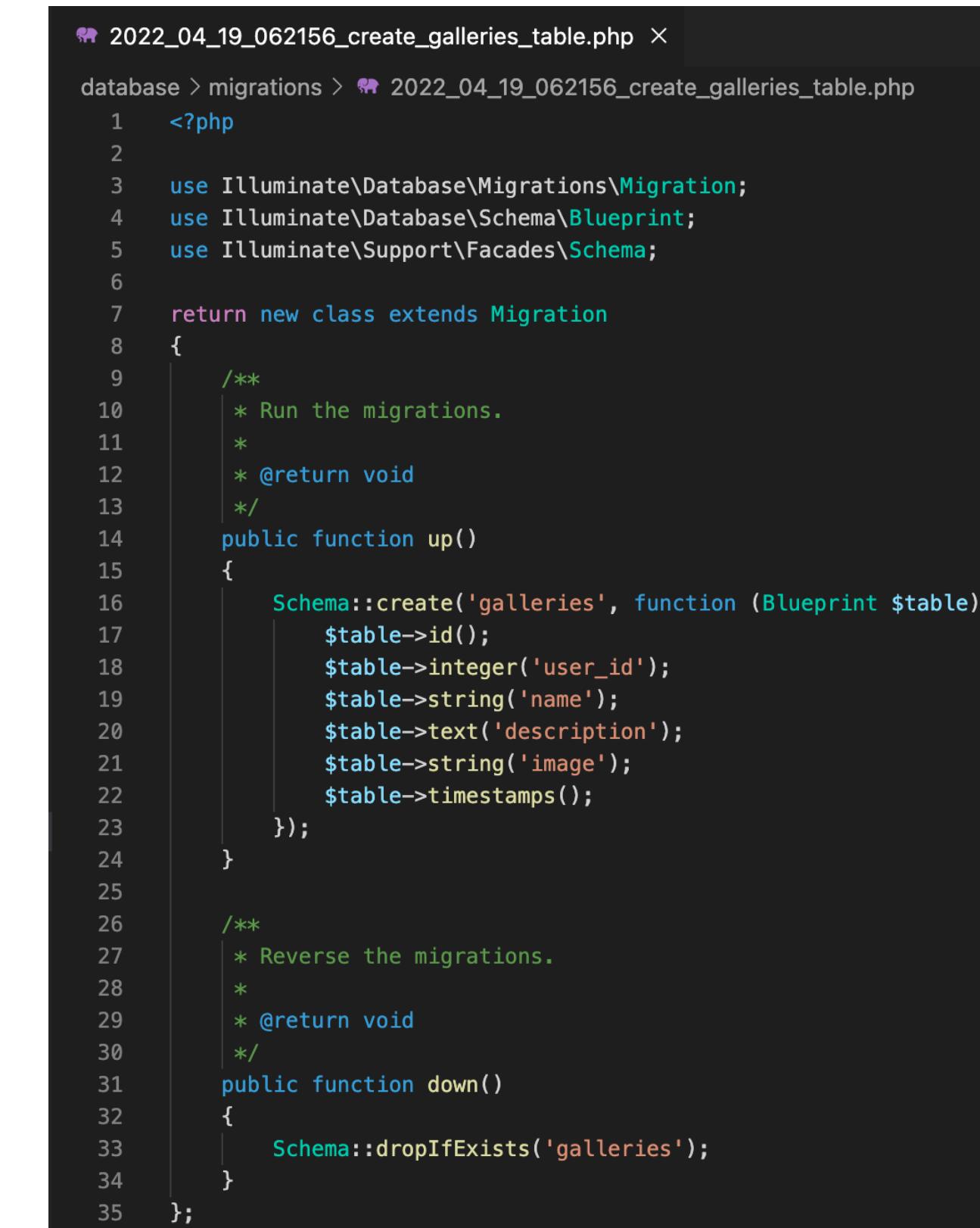
1. Table users tidak perlu dilakukan migration dikarenakan secara default sudah disediakan oleh Laravel
2. Table galleries dengan cara menjalan perintah php artisan make:migration create\_galleries\_table
3. Pada migration table galleries tambahkan code dan perintah php artisan migrate

```
$table->integer('user_id');

$table->string('name');

$table->text('description');

$table->string('image');
```

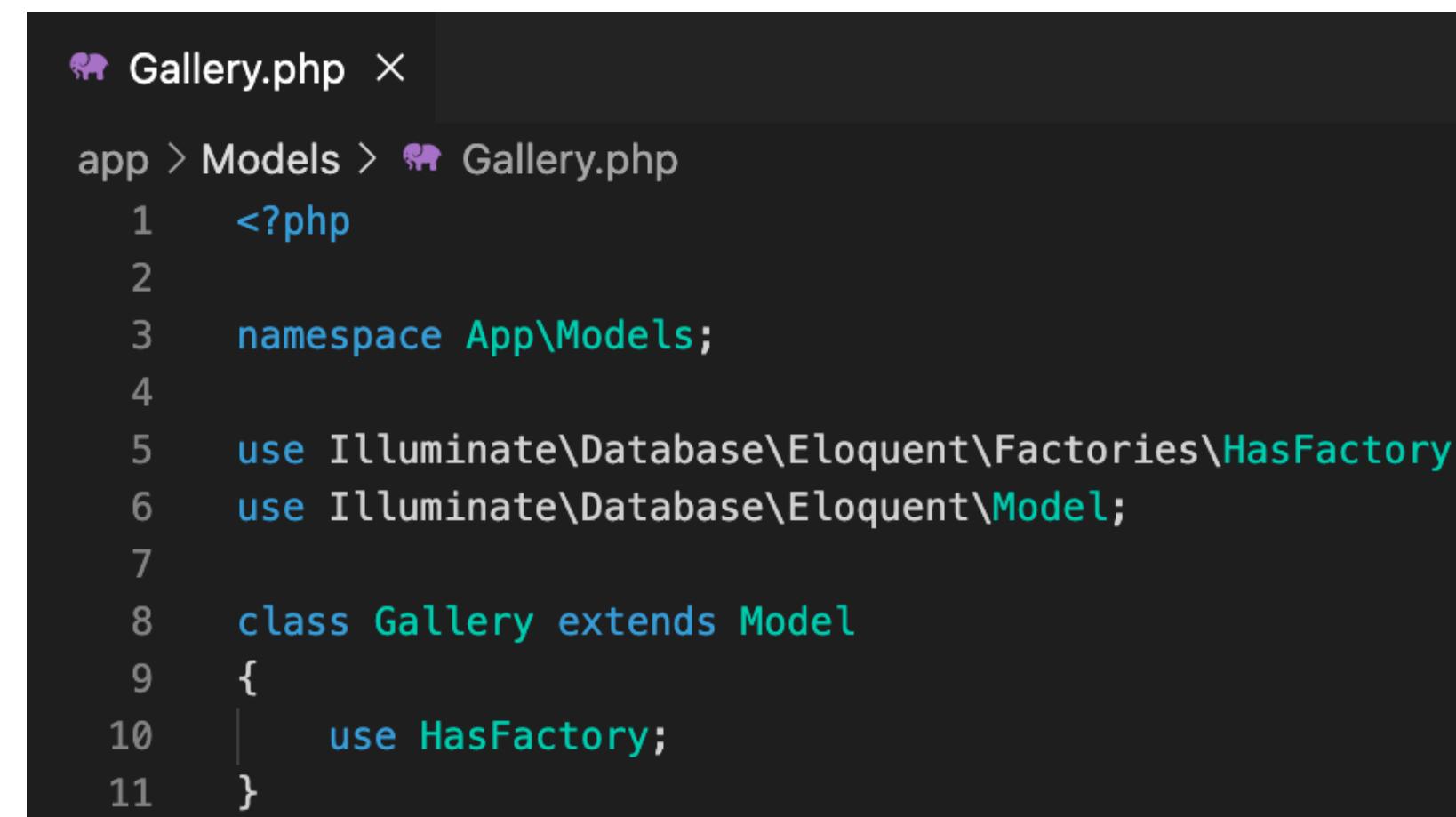


```
2022_04_19_062156_create_galleries_table.php ×
database > migrations > 2022_04_19_062156_create_galleries_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14    public function up()
15    {
16        Schema::create('galleries', function (Blueprint $table) {
17            $table->id();
18            $table->integer('user_id');
19            $table->string('name');
20            $table->text('description');
21            $table->string('image');
22            $table->timestamps();
23        });
24    }
25
26    /**
27     * Reverse the migrations.
28     *
29     * @return void
30     */
31    public function down()
32    {
33        Schema::dropIfExists('galleries');
34    }
35};
```

# CREATE MODEL

---

Buat Model dengan nama Gallery dengan menggunakan perintah php artisan make:model Gallery



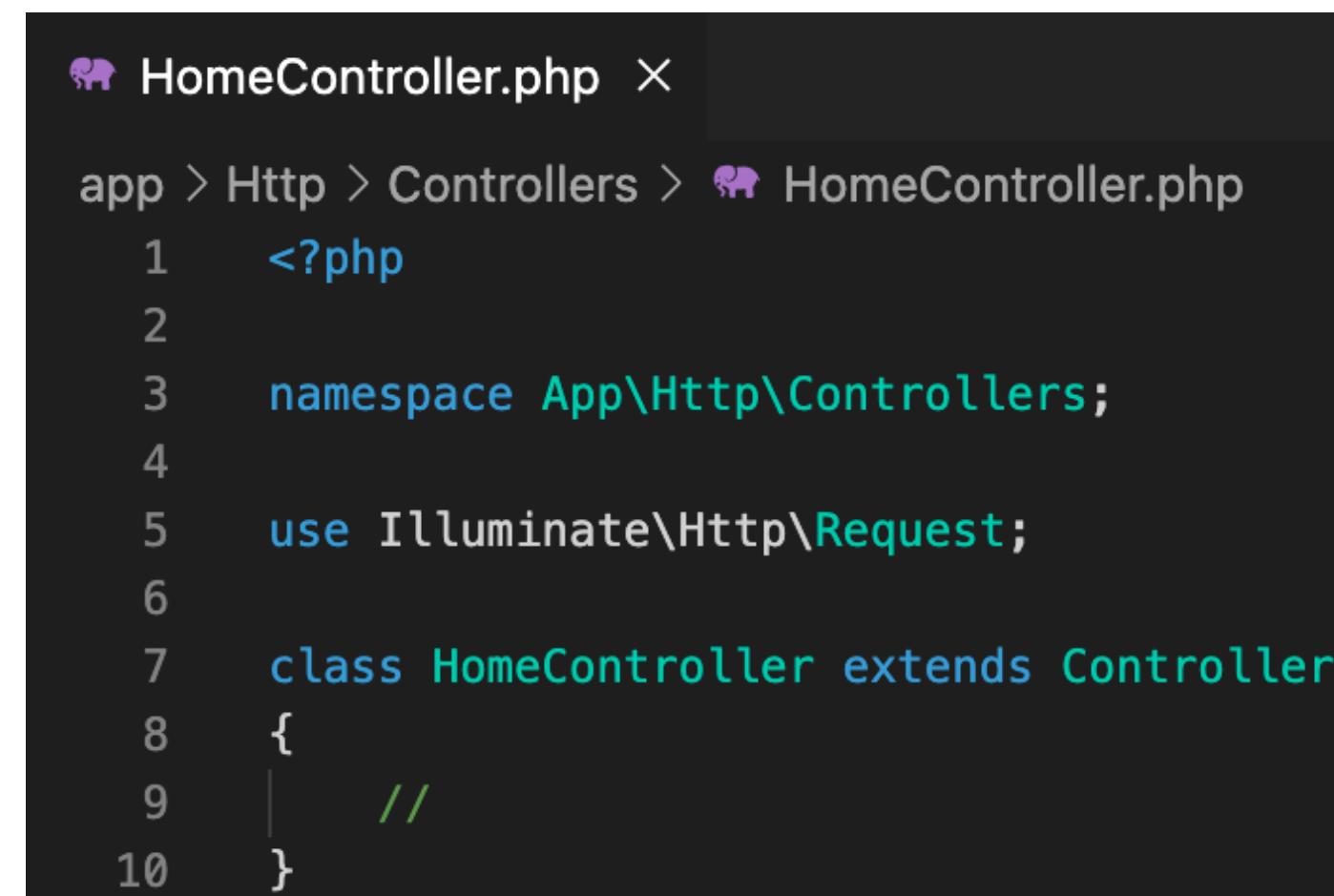
The screenshot shows a code editor window with a dark theme. The file is named 'Gallery.php' and is located in the 'Models' directory under 'app'. The code is as follows:

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Gallery extends Model
9 {
10     use HasFactory;
11 }
```

# CREATE CONTROLLER

---

Buat controller dengan nama HomeController dengan menggunakan perintah php artisan make:controller HomeController



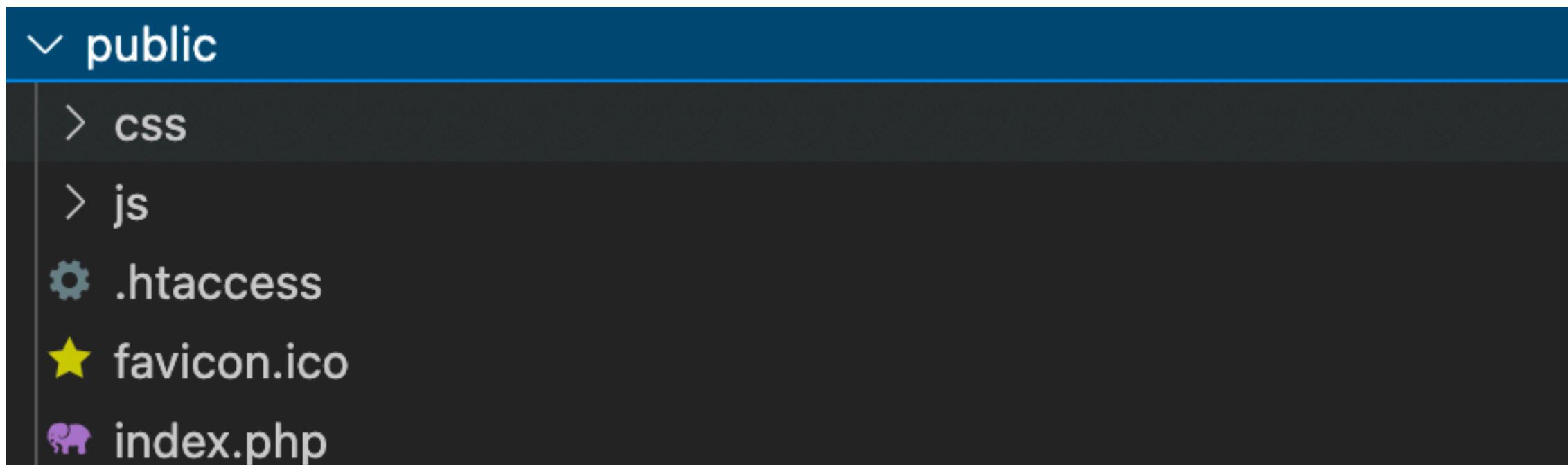
The screenshot shows a code editor window with a dark theme. The title bar says " HomeController.php X". Below it, the file path is shown as "app > Http > Controllers > HomeController.php". The code itself is:

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class HomeController extends Controller
8  {
9      //
10 }
```

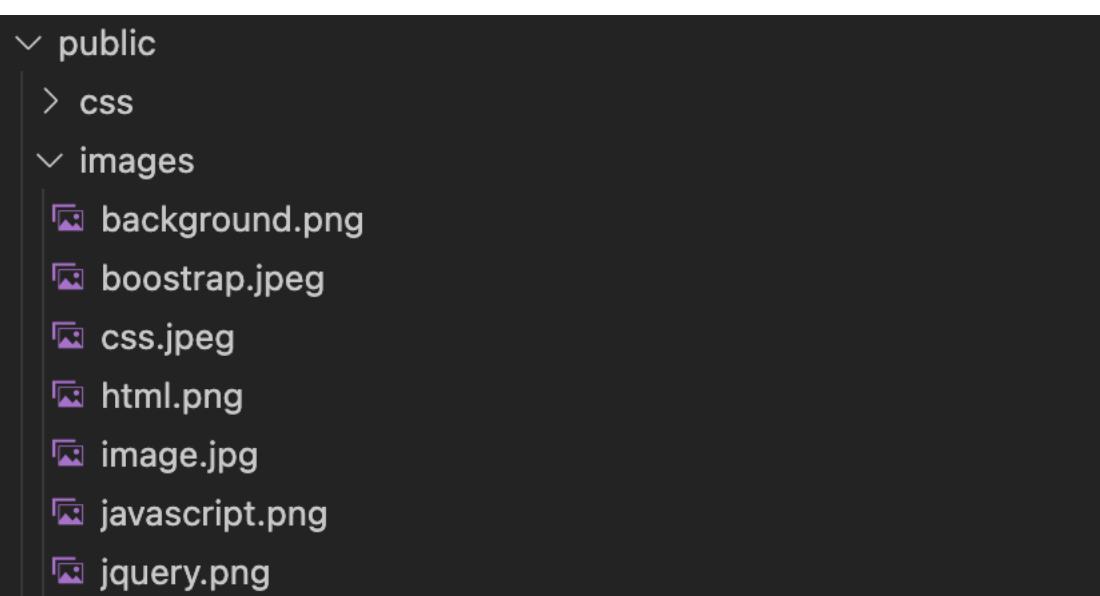
# CREATE VIEW

---

1. Download file bootstrap dan extract.
2. Copy folder css dan js ke dalam folder public



3. Buat folder images pada public, copy semua image dari pertemuan 4



# CREATE VIEW

---

1. Buat folder dengan nama layout dan di dalam folder tersebut buat file dengan nama app.blade.php. Fungsi dari @yield('content') adalah menampilkan data yang akan dikirim oleh @section('nama\_section'). Sedangkan @stack berguna untuk menampilkan data yang akan dikirim oleh @push



The screenshot shows the code editor interface with the file 'app.blade.php' open. The code is a Laravel blade template. It includes a header section with meta tags, a style block for a background image, and a navigation bar with a dropdown menu for authenticated users. The main section contains a centered heading and text. A footer section at the bottom includes a copyright notice. The code uses various Laravel blade directives like @if, @else, @endif, @stack, @yield, and @push.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>@yield('title') | Web Development Talent Hub</title>
        <link href="{{ asset('css/bootstrap.min.css') }}" rel="stylesheet">
        @stack('styles')
        <style>
            .bg-main {
                background-image: url('{{ asset('/images/background.png') }}');
                background-repeat: no-repeat;
                background-size: cover;
            }
        </style>
    </head>
    <body>
        <header>
            <div class="navbar navbar-dark bg-dark">
                <div class="container">
                    <a href="/" class="navbar-brand">
                        <strong>Gallery</strong>
                    </a>
                    @if(Auth::user())
                        <div class="dropdown">
                            <button class="btn dropdown-toggle text-white" data-bs-toggle="dropdown">
                                {{Auth::user()->name}}
                            </button>
                            <ul class="dropdown-menu">
                                <li><a class="dropdown-item" href="{{ url('/gallery') }}>Upload Gallery</a></li>
                                <li><a class="dropdown-item" href="{{ route('processLogout') }}>Logout</a></li>
                            </ul>
                        </div>
                    @else
                        <a class="text-white text-decoration-none" href="{{ url('/login') }}>Login</a>
                    @endif
                </div>
            </div>
        </header>
        <main>
            <section class="py-5 text-center container-fluid bg-main">
                <div class="row py-5">
                    <div class="col-md-12">
                        <h1 class="text-white">Gallery Web Development Talent Hub</h1>
                        <p class="lead text-white">Selamat datang di gallery web development Talenthub 2022 Batch 1</p>
                    </div>
                </div>
            </section>
            <div class="py-5 bg-light">
                <div class="container">
                    @yield('content')
                </div>
            </div>
        </main>
        <footer class="py-5">
            <div class="container">
                <p class="mb-0">Halaman ini dibuat oleh team Web Development</p>
            </div>
        </footer>
    </body>
    <script src="js/bootstrap.min.js"></script>
</html>
```

# CREATE VIEW

---

2. Buat folder dengan nama home dan di dalam folder tersebut buat file dengan nama index.blade.php. Fungsi dari `@extends('layout.app')` menandakan bahwa kita menggunakan layouts.blade.php sebagai layout website. Sedangkan `@section('nama_section')` digunakan untuk mengirim data kepada `@yield`. Fungsi `foreach` yang berguna untuk melakukan pengulangan data yang dikirimkan melalui controller.



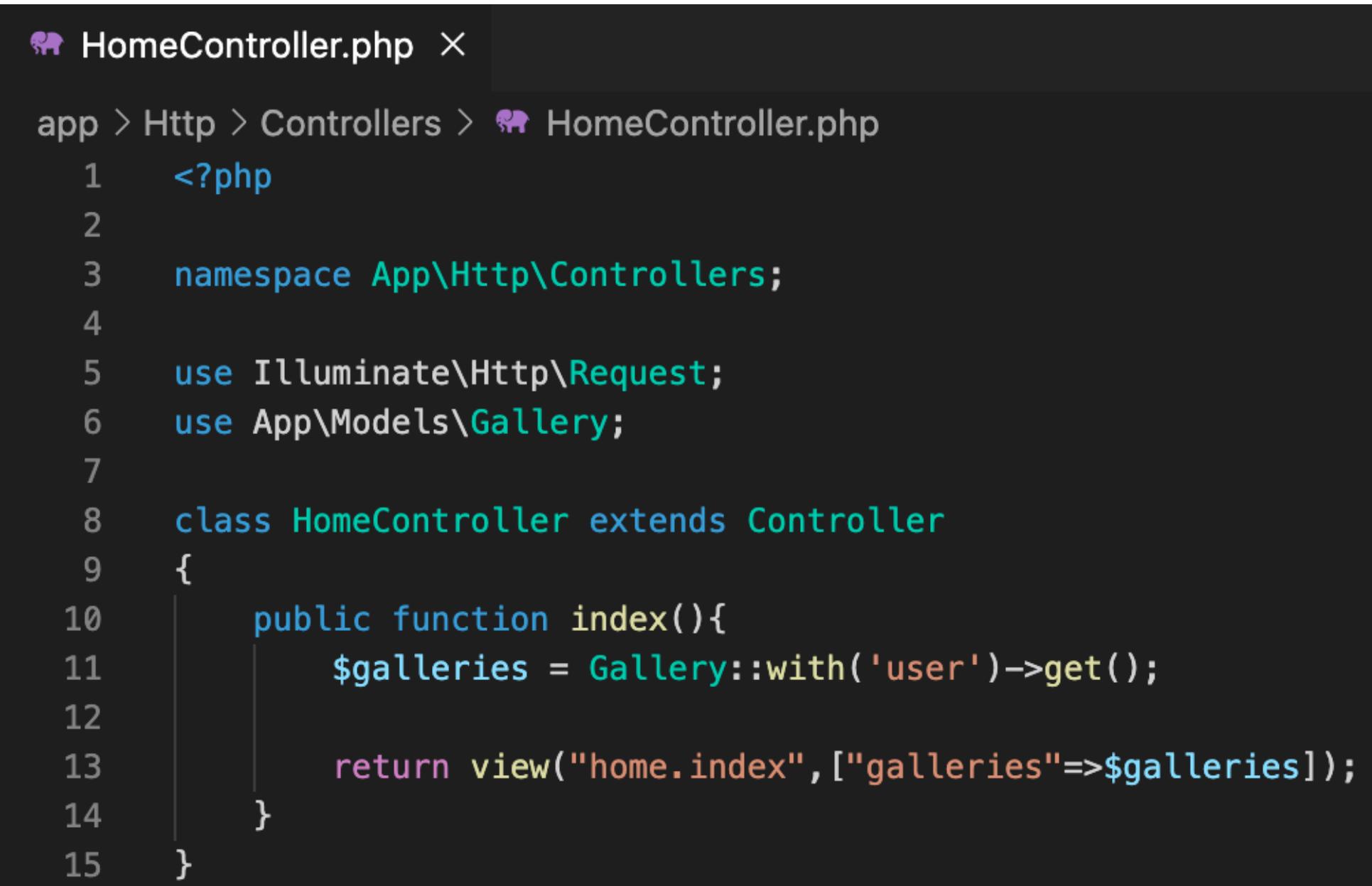
The screenshot shows a code editor window with a dark theme. The file is named "index.blade.php". The code is a Blade template. It starts with `@extends('layout.app')`, followed by `@section('title', 'Home')`. Inside the `@section('content')` block, there is a `<div>` element with a class of "row row-cols-1 row-cols-sm-2 row-cols-md-3 gy-3". Inside this row, a `@foreach` loop iterates over "\$galleries" as "\$gallery". For each gallery, a `<div>` element with a class of "col" is created. Inside this col, a `<div>` element with a class of "card shadow-sm" is created. Inside this card, an `<img>` element with a class of "card-img-top" and attributes width="100%" and height="250" is created, with the source set to `asset($gallery->image)`. Inside the card, a `<div>` element with a class of "card-body" is created, containing two `<p>` elements: one with class "card-text" containing `[$gallery->description]` and another with class "card-text text-end" containing `[$gallery->user->name] - [$gallery->created_at]`. Finally, a closing `</div>` ends the card-body, and another closing `</div>` ends the col. The `@foreach` loop is closed with `@endforeach`, and the entire section is closed with `@endsection`.

```
index.blade.php ×  
resources > views > home > index.blade.php  
1  @extends('layout.app')  
2  
3  @section('title', 'Home')  
4  
5  @section('content')  
6  <div class="row row-cols-1 row-cols-sm-2 row-cols-md-3 gy-3">  
7  |  @foreach ($galleries as $gallery)  
8  |  |  <div class="col">  
9  |  |  |  <div class="card shadow-sm">  
10 |  |  |  |    
11 |  |  |  |  <div class="card-body">  
12 |  |  |  |  <p class="card-text">{{$gallery->description}}</p>  
13 |  |  |  |  <p class="card-text text-end">{{$gallery->user->name}} - {{$gallery->created_at}}</p>  
14 |  |  |  |  </div>  
15 |  |  |  </div>  
16 |  |  </div>  
17 |  &endforeach  
18 |</div>  
19 |@endsection
```

# CREATE FUNCTION CONTROLLER

---

Pada file HomeController buat fungsi dengan nama index yang di dalamnya berguna untuk memanggil view home.index



The screenshot shows a code editor window with a dark theme. The file is named 'HomeController.php' located at 'app > Http > Controllers'. The code defines a class 'HomeController' that extends 'Controller'. It contains a public function 'index()' which retrieves galleries from the 'Gallery' model where the user is associated ('with('user')->get()'), and then returns a view named 'home.index' with the galleries passed as a parameter.

```
 HomeController.php ×  
app > Http > Controllers > HomeController.php  
1  <?php  
2  
3  namespace App\Http\Controllers;  
4  
5  use Illuminate\Http\Request;  
6  use App\Models\Gallery;  
7  
8  class HomeController extends Controller  
9  {  
10     public function index(){  
11         $galleries = Gallery::with('user')->get();  
12  
13         return view("home.index",["galleries"=>$galleries]);  
14     }  
15 }
```

# CREATE SEEDER USER

---

Buat seeder dengan menggunakan perintah php artisan make:seeder UserSeeder, untuk seeder gallery dapat di comment terlebih dahulu dan selanjutnya jalan perintah php artisan db:seed

```
>UserTableSeeder.php ×  
database > seeders > UserTableSeeder.php  
1  <?php  
2  
3  namespace Database\Seeders;  
4  
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;  
6  use Illuminate\Database\Seeder;  
7  use Illuminate\Support\Facades\Hash;  
8  use DB;  
9  
10 class UserTableSeeder extends Seeder  
11 {  
12     /**  
13      * Run the database seeds.  
14      *  
15      * @return void  
16      */  
17     public function run()  
18     {  
19         DB::table('users')->insert([  
20             'name' => 'faerulsalamun',  
21             'email' => 'faerulsalamun@gmail.com',  
22             'password' => Hash::make('123456'),  
23         ]);  
24     }  
25 }
```

```
DatabaseSeeder.php ×  
database > seeders > DatabaseSeeder.php  
1  <?php  
2  
3  namespace Database\Seeders;  
4  
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;  
6  use Illuminate\Database\Seeder;  
7  
8  class DatabaseSeeder extends Seeder  
9  {  
10     /**  
11      * Seed the application's database.  
12      *  
13      * @return void  
14      */  
15     public function run()  
16     {  
17         $this->call([  
18             UserTableSeeder::class  
19         ]);  
20     }  
21 }
```

# CREATE SEEDER GALLERY

---

Buat seeder dengan menggunakan perintah php artisan make:seeder GallerySeeder dan selanjutnya jalankan perintah php artisan db:seed

```
GalleryTableSeeder.php ×  
database > seeders > GalleryTableSeeder.php  
1  <?php  
2  
3  namespace Database\Seeders;  
4  
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;  
6  use Illuminate\Database\Seeder;  
7  use DB;  
8  
9  class GalleryTableSeeder extends Seeder  
10 {  
11     /**  
12      * Run the database seeds.  
13      *  
14      * @return void  
15      */  
16     public function run()  
17     {  
18         DB::table('galleries')->insert([  
19             'name' => 'Gambar CSS',  
20             'description' => 'Ini merupakan Gambar CSS',  
21             'image' => '/images/css.jpeg',  
22             'created_at' => date('Y-m-d H:i:s'),  
23             'user_id' => 1  
24         ]);  
25         DB::table('galleries')->insert([  
26             'name' => 'Gambar Bootstrap',  
27             'description' => 'Ini merupakan Gambar Bootstrap',  
28             'image' => '/images/bootstrap.jpeg',  
29             'created_at' => date('Y-m-d H:i:s'),  
30             'user_id' => 1  
31         ]);  
32         DB::table('galleries')->insert([  
33             'name' => 'Gambar Javascript',  
34             'description' => 'Ini merupakan Gambar Javascript',  
35             'image' => '/images/javascript.png',  
36             'created_at' => date('Y-m-d H:i:s'),  
37             'user_id' => 1  
38         ]);  
39     }  
40 }
```

```
DatabaseSeeder.php ×  
database > seeders > DatabaseSeeder.php  
1  <?php  
2  
3  namespace Database\Seeders;  
4  
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;  
6  use Illuminate\Database\Seeder;  
7  
8  class DatabaseSeeder extends Seeder  
9  {  
10     /**  
11      * Seed the application's database.  
12      *  
13      * @return void  
14      */  
15     public function run()  
16     {  
17         $this->call([  
18             GalleryTableSeeder::class,  
19             // UserTableSeeder::class  
20         ]);  
21     }  
22 }
```

# Homepage

---

Selanjutnya buka localhost:8000 atau localhost/nama-project/public , jika berhasil maka akan tampil layout seperti dibawah ini

The screenshot shows a web application interface. At the top, there is a dark header bar with the word "Gallery" on the left and "Login" on the right. Below the header is a large, colorful background section with abstract purple and red shapes. In the center of this section, the text "Gallery Web Development Talent Hub" is displayed in white, followed by a smaller line of text: "Selamat datang di gallery web development Talenthub 2022 Batch 1". Below this, the main content area features three cards arranged horizontally. The first card, with a blue background, contains the CSS3 logo and the text "Ini merupakan Gambar CSS" and "faerulsalamun - 2022-04-19 17:27:54". The second card, with a purple background, contains the Bootstrap logo and the text "Ini merupakan Gambar Bootstrap" and "faerulsalamun - 2022-04-19 17:27:54". The third card, with a yellow background, contains the JS logo and the text "Ini merupakan Gambar Javascript" and "faerulsalamun - 2022-04-19 17:27:54". At the bottom of the page, there is a footer note: "Halaman ini dibuat oleh team Web Development".

# **LOGIN**

# CREATE VIEW LOGIN

---

Buat folder login pada resources > login dan buat file dengan nama index.blade.php. Fungsi @push berguna untuk mengirim data ke @stack

```
index.blade.php ×
resources > views > login > index.blade.php
1  @extends('layout.app')
2
3  @section('title', 'Home')
4
5  @push('styles')
6  <style>
7      .form-signin {
8          width: 100%;
9          max-width: 400px;
10         padding: 15px;
11         margin: auto;
12     }
13 </style>
14 @endpush
15
16 @section('content')
17 <div class="form-signin text-center">
18     <form action="{{ route('processLogin') }}" method="POST">
19         @csrf
20         <h1 class="mb-3">Silakan Login</h1>
21         @if(session('error'))
22             <div class="alert alert-danger">
23                 <b>Opps!</b> {{session('error')}}
24             </div>
25         @endif
26
27         <div class="form-floating mb-3">
28             <input type="email" class="form-control" placeholder="nama@email.com" id="email" name="email" required>
29             <label for="floatingInput">Alamat Email</label>
30         </div>
31
32         <div class="form-floating mb-3">
33             <input type="password" class="form-control" placeholder="Password" id="password" name="password" required>
34             <label for="floatingPassword">Password</label>
35         </div>
36
37         <p>Belum punya account ? Silakan <a href="{{ url('/register') }}>Register disini</a><p>
38
39         <button class="w-100 btn btn-lg btn-primary" type="submit">Sign In</button>
40     </form>
41 </div>
42 @endsection
```

# CREATE LOGIN CONTROLLER

---

Buat controller dengan nama LoginController dengan menggunakan perintah php artisan make:controller LoginController. Selanjutnya buat function untuk menampilkan view login dan tambahkan route login



```
(LoginController.php)
app > Http > Controllers > LoginController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class LoginController extends Controller
8  {
9      public function index(){
10         return view("login.index");
11     }
12 }
```

Route::get('/login', [LoginController::class, 'index']);

# CREATE FUNCTION PROCESS LOGIN

---

Pada LoginController buat function dengan nama processLogin yang berguna untuk melakukan validasi dan proses login.

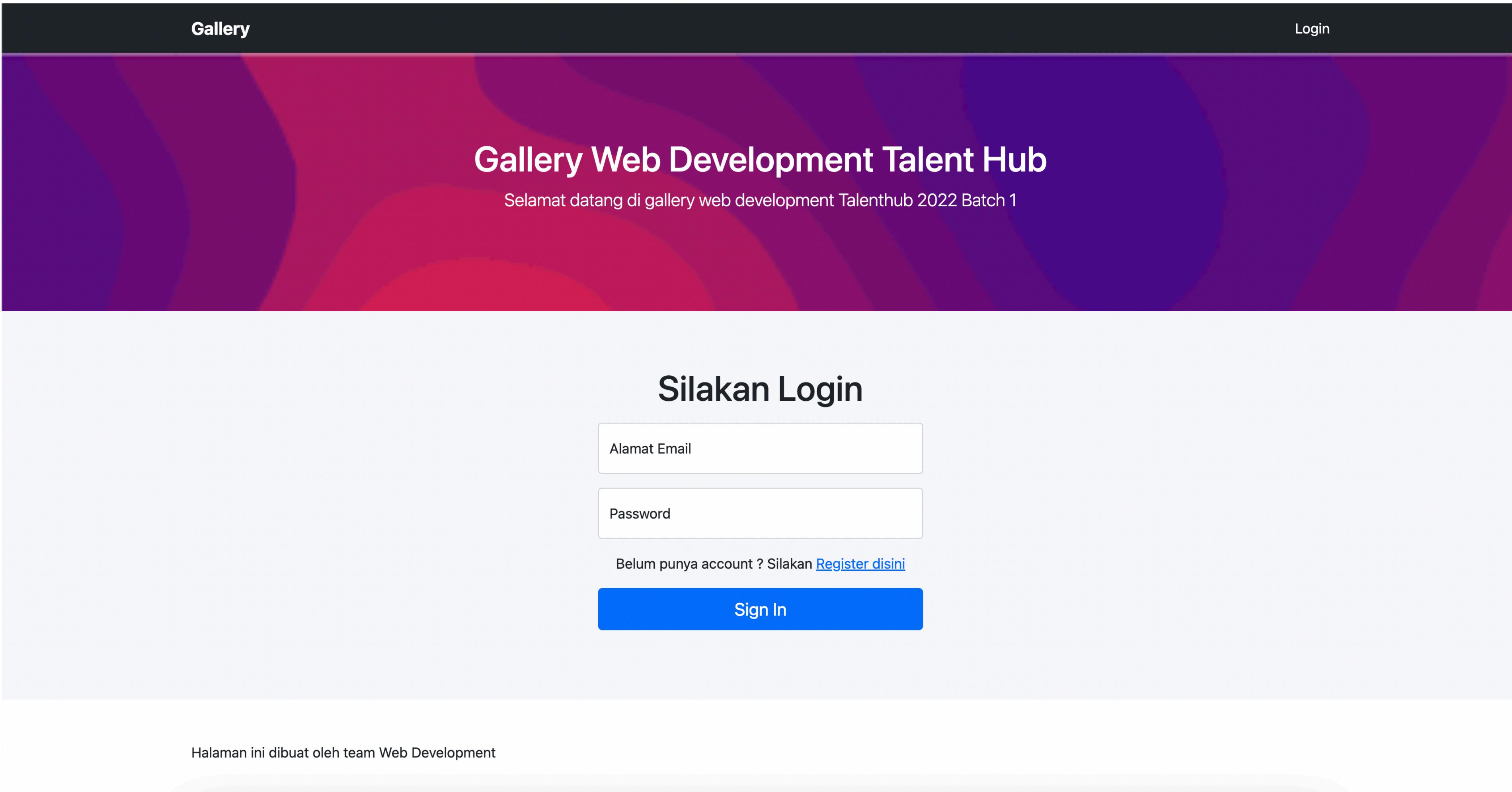
```
public function processLogin(Request $request){
    $data = [
        'email' => $request->input('email'),
        'password' => $request->input('password'),
    ];

    if (Auth::Attempt($data)) {
        return redirect('/');
    }else{
        Session::flash('error', 'Email atau Password Salah');
        return redirect('/login');
    }
}
```

```
Route::post('/processLogin',[LoginController::class,'processLogin'])->name('processLogin');
```

# LOGIN

---



The image shows a login page for a website. At the top, there is a dark header bar with the word "Gallery" on the left and "Login" on the right. Below the header is a large, colorful background image featuring abstract purple and red circular patterns. In the center of this background, the text "Gallery Web Development Talent Hub" is displayed in a large, bold, white font. Below this, a smaller line of text reads "Selamat datang di gallery web development Talenthub 2022 Batch 1". The main content area is a light gray box containing the text "Silakan Login" in bold black font. Below this are two input fields: one for "Alamat Email" and one for "Password", both with placeholder text. At the bottom of the form, there is a link "Belum punya account ? Silakan [Register disini](#)". A blue "Sign In" button is located at the bottom right of the form. At the very bottom of the page, there is a small footer note: "Halaman ini dibuat oleh team Web Development".

Gallery

Login

## Gallery Web Development Talent Hub

Selamat datang di gallery web development Talenthub 2022 Batch 1

### Silakan Login

Alamat Email

Password

Belum punya account ? Silakan [Register disini](#)

Sign In

Halaman ini dibuat oleh team Web Development

# SUCCESS LOGIN

---

Gallery faerulsalamun ▾

## Gallery Web Development Talent Hub

Selamat datang di gallery web development Talenthub 2022 Batch 1



Ini merupakan Gambar CSS  
faerulsalamun - 2022-04-19 17:27:54



Ini merupakan Gambar Bootstrap  
faerulsalamun - 2022-04-19 17:27:54



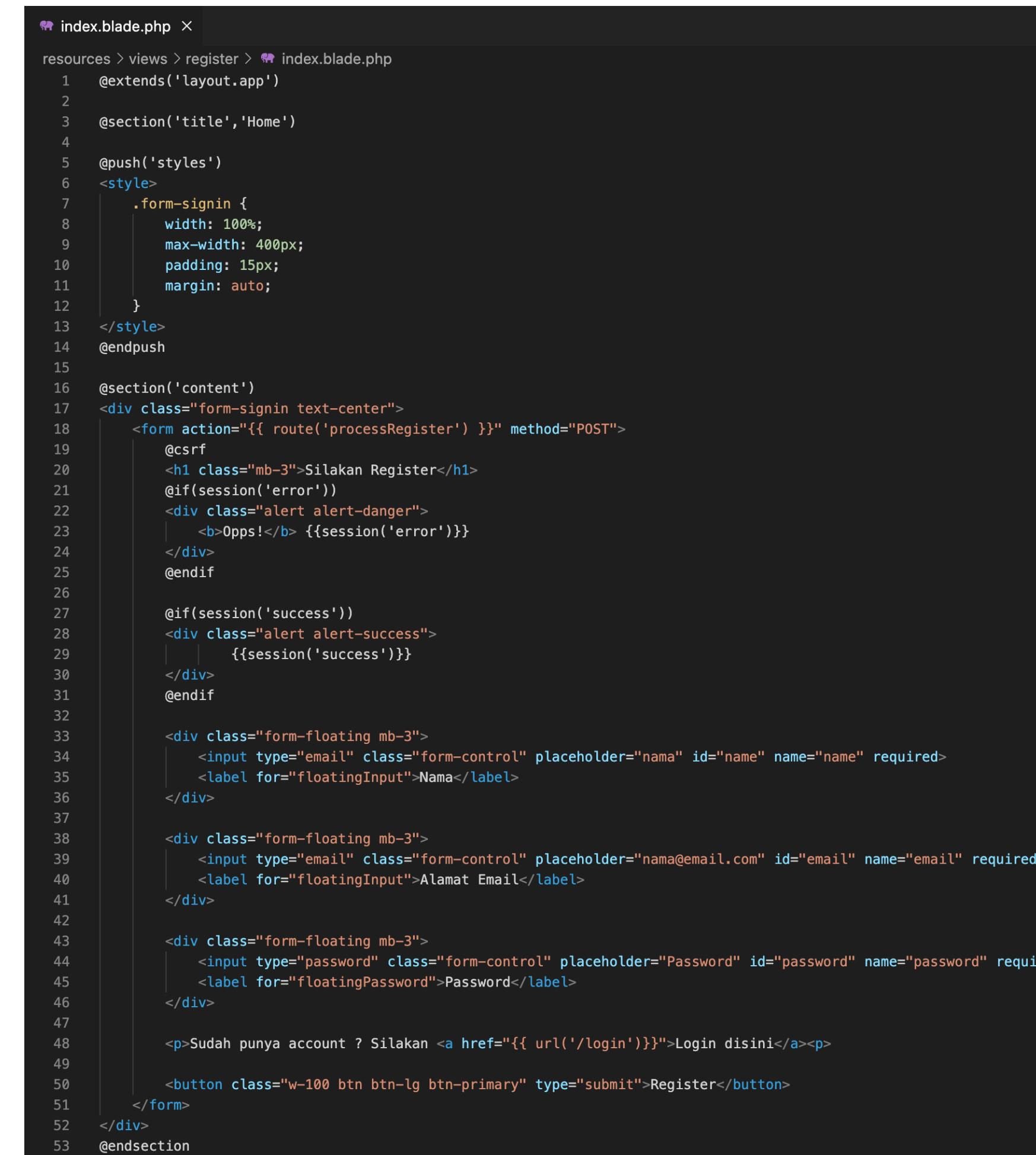
Ini merupakan Gambar Javascript  
faerulsalamun - 2022-04-19 17:27:54

# REGISTER

# CREATE VIEW REGISTER

---

Buat folder register pada resources > register dan buat file dengan nama index.blade.php



```
index.blade.php
resources > views > register > index.blade.php
1 @extends('layout.app')
2
3 @section('title','Home')
4
5 @push('styles')
6 <style>
7   .form-signin {
8     width: 100%;
9     max-width: 400px;
10    padding: 15px;
11    margin: auto;
12  }
13 </style>
14 @endpush
15
16 @section('content')
17 <div class="form-signin text-center">
18   <form action="{{ route('processRegister') }}" method="POST">
19     @csrf
20     <h1 class="mb-3">Silakan Register</h1>
21     @if(session('error'))
22       <div class="alert alert-danger">
23         <b>Opps!</b> {{session('error')}}
24       </div>
25     @endif
26
27     @if(session('success'))
28       <div class="alert alert-success">
29         {{session('success')}}
30       </div>
31     @endif
32
33     <div class="form-floating mb-3">
34       <input type="email" class="form-control" placeholder="nama" id="name" name="name" required>
35       <label for="floatingInput">Nama</label>
36     </div>
37
38     <div class="form-floating mb-3">
39       <input type="email" class="form-control" placeholder="nama@email.com" id="email" name="email" required>
40       <label for="floatingInput">Alamat Email</label>
41     </div>
42
43     <div class="form-floating mb-3">
44       <input type="password" class="form-control" placeholder="Password" id="password" name="password" required>
45       <label for="floatingPassword">Password</label>
46     </div>
47
48     <p>Sudah punya account ? Silakan <a href="{{ url('/login') }}>Login disini</a><p>
49
50     <button class="w-100 btn btn-lg btn-primary" type="submit">Register</button>
51   </form>
52 </div>
53 @endsection
```

# CREATE REGISTER CONTROLLER

---

Buat controller dengan nama RegisterController dengan menggunakan perintah php artisan make:controller RegisterController. Selanjutnya buat function untuk menampilkan view register dan tambahkan route register

```
RegisterController.php ×  
app > Http > Controllers > RegisterController.php  
1  <?php  
2  
3  namespace App\Http\Controllers;  
4  
5  use Illuminate\Http\Request;  
6  use Auth;  
7  use Session;  
8  use Illuminate\Support\Facades\Hash;  
9  use App\Models\User;  
10  
11 class RegisterController extends Controller  
12 {  
13     public function index()  
14     {  
15         return view("register.index");  
16     }  
17 }
```



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

# CREATE FUNCTION PROCESS LOGIN

---

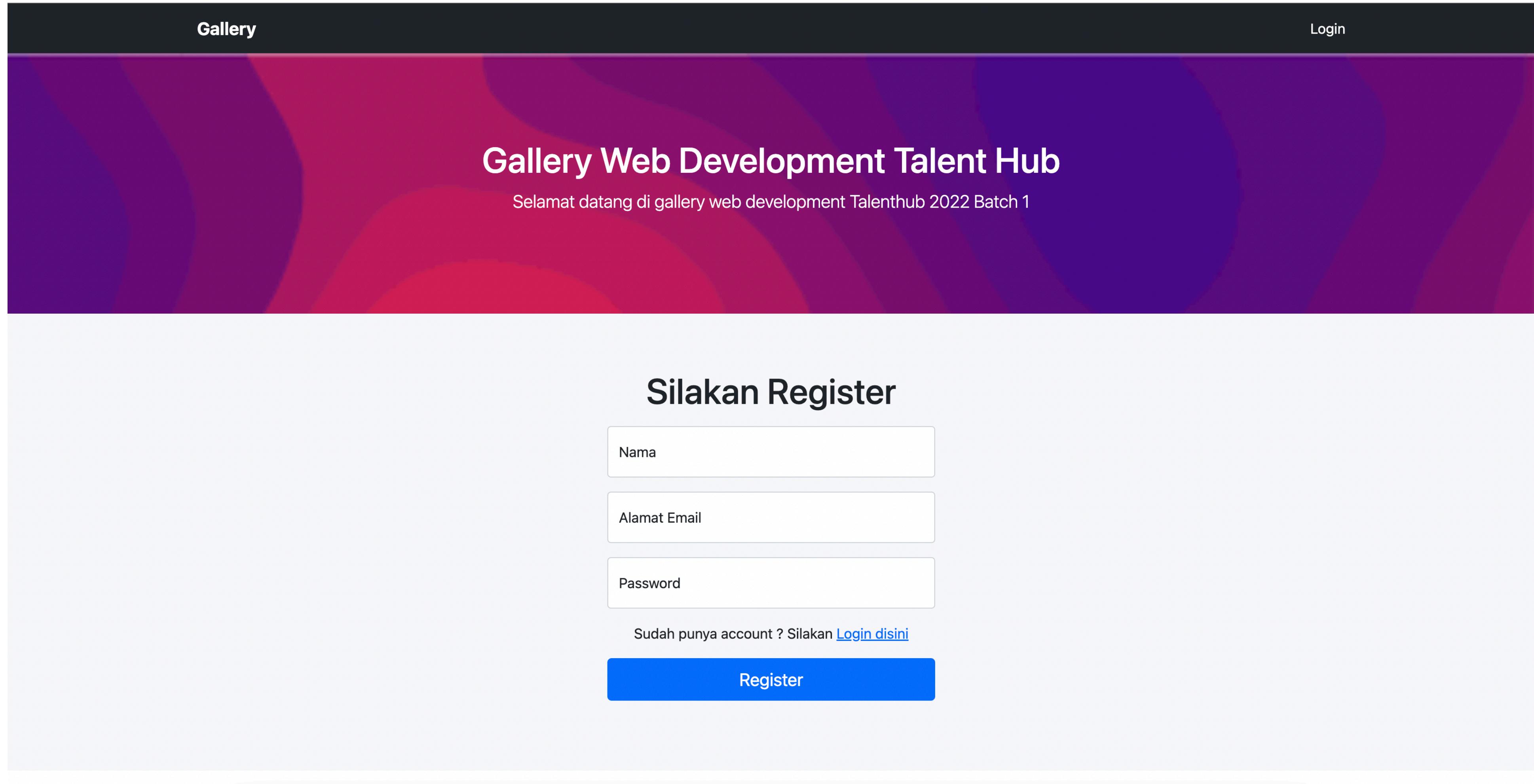
Pada RegisterController buat function dengan nama processRegister yang berguna untuk melakukan validasi dan proses register.

```
public function processRegister(Request $request){  
    // Check data  
    $dataUser = User::where('email',$request->input('email'))->first();  
  
    if ($dataUser) {  
        Session::flash('error', 'Email sudah terdaftar');  
        return redirect('/register');  
    }else{  
        $user = new User();  
        $user->name = $request->input('name');  
        $user->email = $request->input('email');  
        $user->password = Hash::make($request->input('password'));  
        $user->save();  
  
        Session::flash('success', 'Berhasil daftar');  
  
        return redirect('/register');  
    }  
}
```

```
Route::post('/processRegister',[RegisterController::class,'processRegister'])->name('processRegister');
```

# REGISTER

---



The image shows a registration page for a web development talent hub. At the top, there is a dark header bar with the word "Gallery" on the left and "Login" on the right. Below the header is a large, colorful background image featuring abstract shapes in shades of purple, red, and blue. In the center of this image, the text "Gallery Web Development Talent Hub" is displayed in a large, bold, white font. Below this, a smaller line of text reads "Selamat datang di gallery web development Talenthub 2022 Batch 1". The main content area is white and contains the heading "Silakan Register" in a large, bold, black font. Below this are three input fields: "Nama" (Name), "Alamat Email" (Email Address), and "Password". At the bottom of the form, a link says "Sudah punya account ? Silakan [Login disini](#)". A large blue button at the bottom right is labeled "Register".

**Gallery**

**Login**

**Gallery Web Development Talent Hub**

Selamat datang di gallery web development Talenthub 2022 Batch 1

**Silakan Register**

Nama

Alamat Email

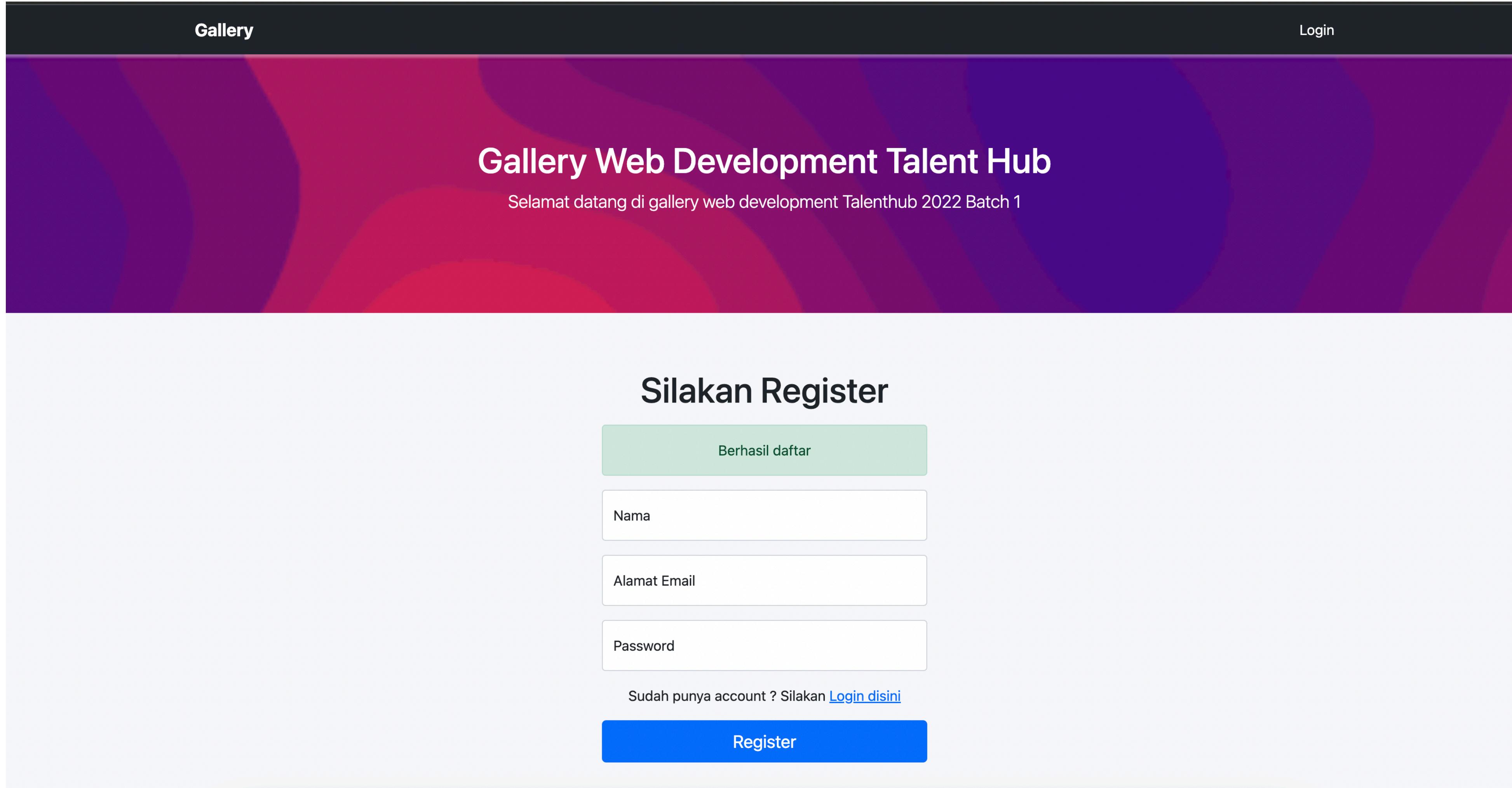
Password

Sudah punya account ? Silakan [Login disini](#)

Register

# SUCCESS REGISTER

---



The image shows a registration page for a web development talent hub. At the top, there's a dark header bar with 'Gallery' on the left and 'Login' on the right. Below the header is a decorative background featuring abstract red and purple circular patterns. In the center, the text 'Gallery Web Development Talent Hub' is displayed in a large, bold, white font. Below it, a smaller white text says 'Selamat datang di gallery web development Talenthub 2022 Batch 1'. The main content area has a light gray background. It features the heading 'Silakan Register' in bold black text. To the right of the heading is a green rectangular button with the text 'Berhasil daftar' in white. Below this are three input fields: 'Nama' (Name), 'Alamat Email' (Email Address), and 'Password'. At the bottom of the form, a link says 'Sudah punya account ? Silakan [Login disini](#)'. A blue rectangular button at the very bottom is labeled 'Register'.

Gallery

Login

Gallery Web Development Talent Hub

Selamat datang di gallery web development Talenthub 2022 Batch 1

Silakan Register

Berhasil daftar

Nama

Alamat Email

Password

Sudah punya account ? Silakan [Login disini](#)

Register

# NEXT ?

---

- Membuat feature CRUD gallery untuk User dan Upload To Web Hosting

The screenshot shows a web application interface for a gallery. At the top, there is a dark header bar with the word "Gallery" on the left and a user profile "faerulsalamun" on the right. Below the header is a large, colorful background image featuring abstract purple and red shapes. In the center of this background, the text "Gallery Web Development Talent Hub" is displayed in a large, bold, white font, followed by a smaller line of text: "Selamat datang di gallery web development Talenthub 2022 Batch 1".

The main content area is a table listing three items:

No	Name	Description	Image	Action
1	Gambar CSS	Ini merupakan Gambar CSS		<button>Edit</button> <button>Delete</button>
2	Gambar Bootstrap	Ini merupakan Gambar Bootstrap		<button>Edit</button> <button>Delete</button>
3	Gambar Javascript	Ini merupakan Gambar Javascript		<button>Edit</button> <button>Delete</button>

A blue "Add" button is located at the top left of the table area.