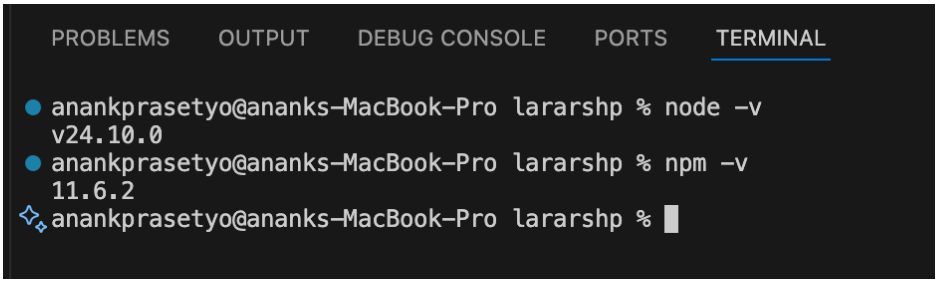
Modul 10

Authentication Laravel

A. Setup Authentication

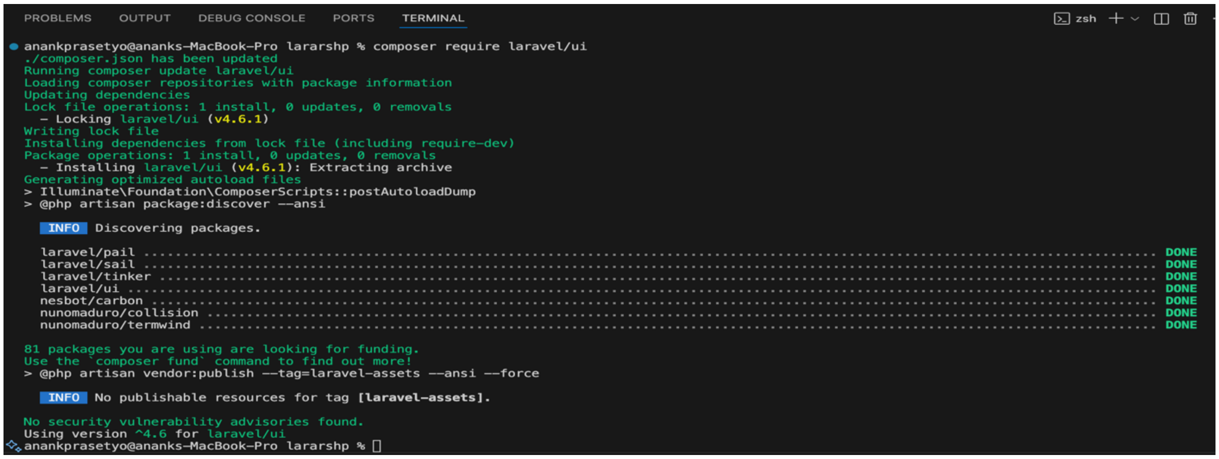
1. Buka file Project sebelumnya
2. Buka terminal atau cmd masing-masing kemudian jalankan perintah
   - node -v
   - npm -v



```
● anankprasetyo@ananks-MacBook-Pro lararshp % node -v
  v24.10.0
● anankprasetyo@ananks-MacBook-Pro lararshp % npm -v
  11.6.2
  anankprasetyo@ananks-MacBook-Pro lararshp % █
```

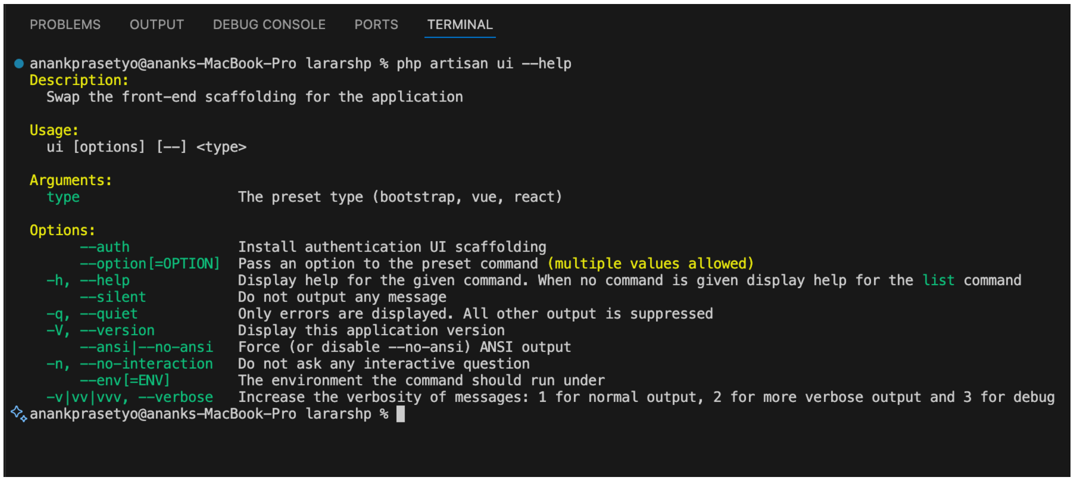
Pastikan sudah keluar semua versinya, jika belum, silahkan install terlebih dahulu npm nya.

3. Install paket composer Laravel UI
   - Jalankan melalui terminal dengan perintah **composer require laravel/ui**



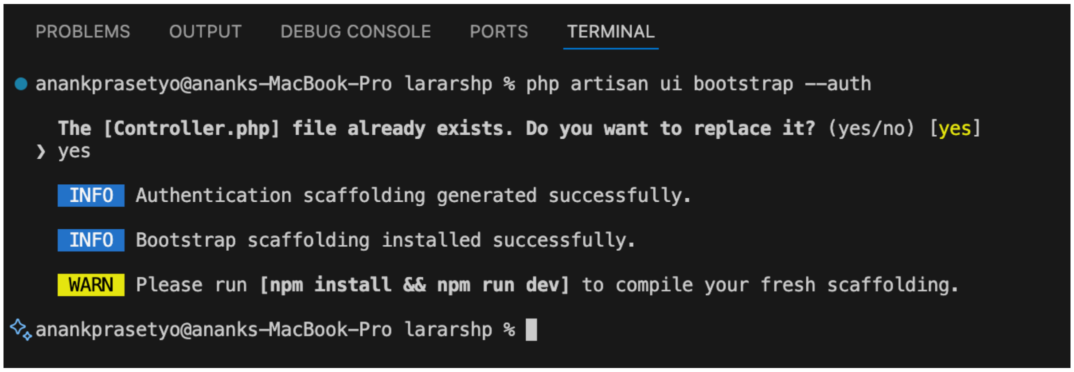
```
● anankprasetyo@ananks-MacBook-Pro lararshp % composer require laravel/ui
  ./composer.json has been updated
  Running composer update laravel/ui
  Loading composer repositories with package information
  Updating dependencies
  Lock file operations: 1 install, 0 updates, 0 removals
    - Locking laravel/ui (v4.6.1)
  Writing lock file
  Installing dependencies from lock file (including require-dev)
  Package operations: 1 install, 0 updates, 0 removals
    - Installing laravel/ui (v4.6.1): Extracting archive
  Generating optimized autoload files
  > Illuminate\Foundation\ComposerScripts::postAutoloadDump
  > @php artisan package:discover --ansi

    INFO  Discovering packages.

    laravel/pail ................................................................................ DONE
    laravel/sail ................................................................................ DONE
    laravel/tinker .............................................................................. DONE
    laravel/ui .................................................................................. DONE
    nesbot/carbon ............................................................................... DONE
    nunomaduro/collision ........................................................................ DONE
    nunomaduro/termwind ......................................................................... DONE

  81 packages you are using are looking for funding.
  Use the `composer fund` command to find out more!
  > @php artisan vendor:publish --tag=laravel-assets --ansi --force

    INFO  No publishable resources for tag [laravel-assets].

  No security vulnerability advisories found.
  Using version ^4.6 for laravel/ui
  anankprasetyo@ananks-MacBook-Pro lararshp % ▯
```

- Setelah selesai, jalankan perintah **php artisan ui –help**



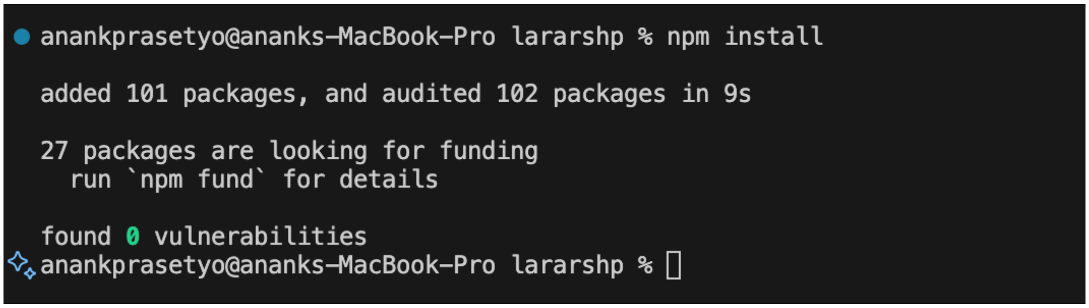
```
● anankprasetyo@ananks-MacBook-Pro lararshp % php artisan ui --help
  Description:
    Swap the front-end scaffolding for the application

  Usage:
    ui [options] [--] <type>

  Arguments:
    type                    The preset type (bootstrap, vue, react)

  Options:
        --auth              Install authentication UI scaffolding
        --option[=OPTION]   Pass an option to the preset command (multiple values allowed)
    -h, --help              Display help for the given command. When no command is given display help for the list command
        --silent            Do not output any message
    -q, --quiet             Only errors are displayed. All other output is suppressed
    -V, --version           Display this application version
        --ansi|--no-ansi    Force (or disable --no-ansi) ANSI output
    -n, --no-interaction    Do not ask any interactive question
        --env[=ENV]         The environment the command should run under
    -v|vv|vvv, --verbose    Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug
  anankprasetyo@ananks-MacBook-Pro lararshp %
```

- Didalam Laravel UI terdapat 3 type auth yang bisa digunakan (bootstrap, vue, react)
- Jalankan perintah **php artisan ui bootstrap –auth** pada terminal masing-masing



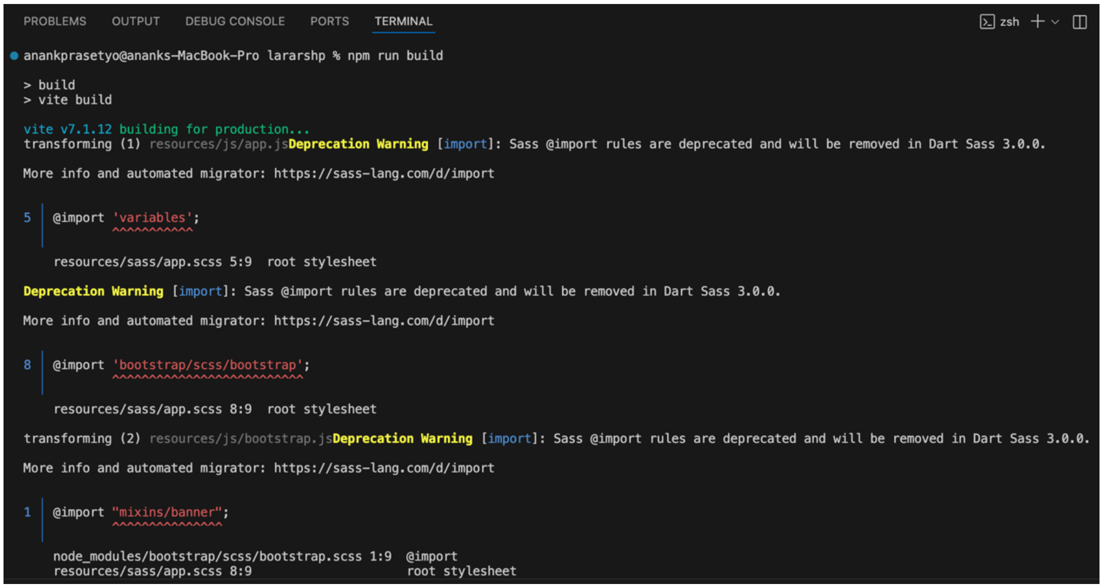
```
● anankprasetyo@ananks-MacBook-Pro lararshp % php artisan ui bootstrap --auth

   The [Controller.php] file already exists. Do you want to replace it? (yes/no) [yes]
  ❯ yes

   INFO  Authentication scaffolding generated successfully.

   INFO  Bootstrap scaffolding installed successfully.

   WARN  Please run [npm install && npm run dev] to compile your fresh scaffolding.

  anankprasetyo@ananks-MacBook-Pro lararshp %
```

- Ketikkan ***yes*** sampai proses selesai
- Jalankan npm install



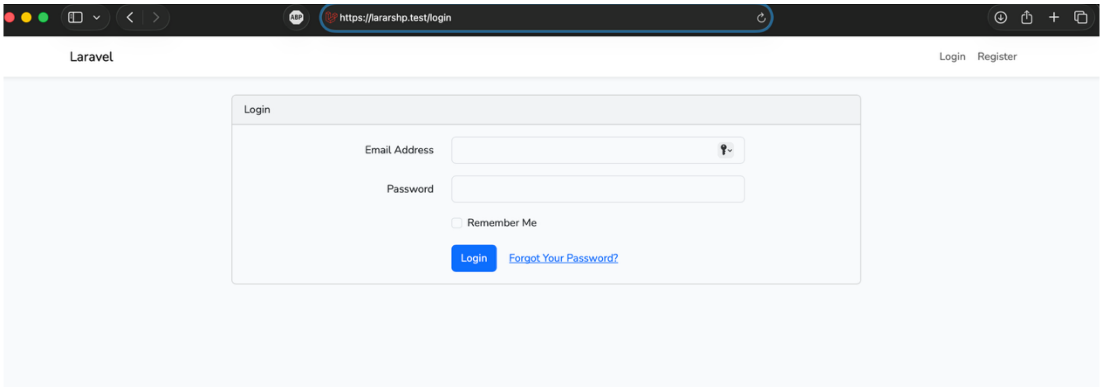
```
● anankprasetyo@ananks-MacBook-Pro lararshp % npm install

  added 101 packages, and audited 102 packages in 9s

  27 packages are looking for funding
    run `npm fund` for details

  found 0 vulnerabilities
  anankprasetyo@ananks-MacBook-Pro lararshp %
```

- Jalankan npm run build



```
● anankprasetyo@ananks-MacBook-Pro lararshp % npm run build

> build
> vite build

vite v7.1.12 building for production...
transforming (1) resources/js/app.jsDeprecation Warning [import]: Sass @import rules are deprecated and will be removed in Dart Sass 3.0.0.

More info and automated migrator: https://sass-lang.com/d/import

  ╷
5 │ @import 'variables';
  │         ^^^^^^^^^^^
  ╵
    resources/sass/app.scss 5:9  root stylesheet

Deprecation Warning [import]: Sass @import rules are deprecated and will be removed in Dart Sass 3.0.0.

More info and automated migrator: https://sass-lang.com/d/import

  ╷
8 │ @import 'bootstrap/scss/bootstrap';
  │         ^^^^^^^^^^^^^^^^^^^^^^^^^^
  ╵
    resources/sass/app.scss 8:9  root stylesheet

transforming (2) resources/js/bootstrap.jsDeprecation Warning [import]: Sass @import rules are deprecated and will be removed in Dart Sass 3.0.0.

More info and automated migrator: https://sass-lang.com/d/import

  ╷
1 │ @import "mixins/banner";
  │         ^^^^^^^^^^^^^^^
  ╵
    node_modules/bootstrap/scss/bootstrap.scss 1:9  @import
    resources/sass/app.scss 8:9                     root stylesheet
```

- Buka url dan coba ketik **nama_domain(localhost)/login**



Login

Email Address

Password

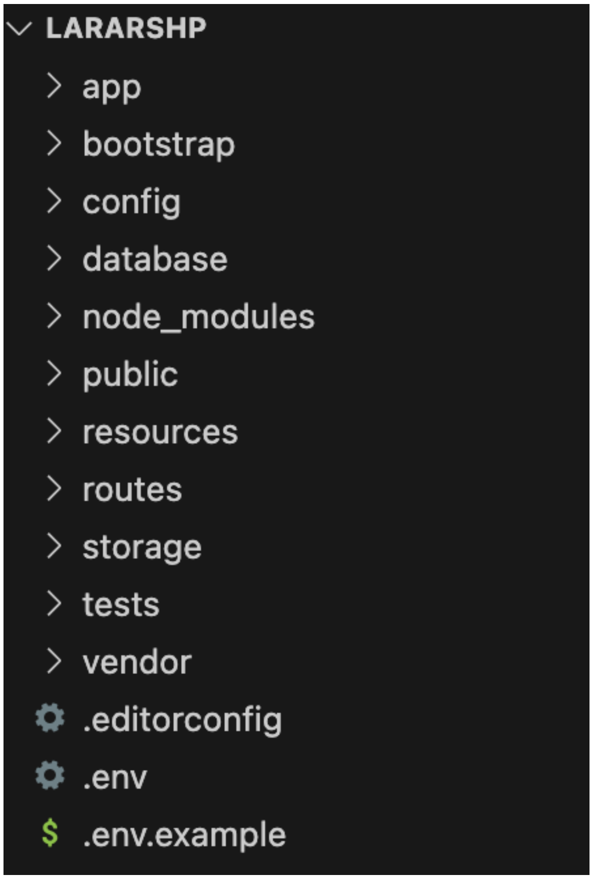
□ Remember Me

Login   Forgot Your Password?

- Pembuatan authentication telah berhasil

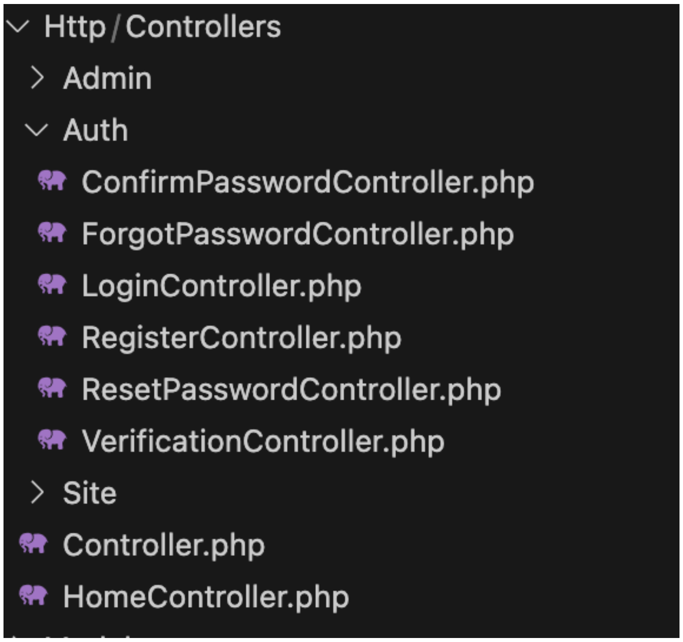
4. Perubahan struktur file
   Setelah membuat generate authentication, struktur file akan berubah karena ada beberapa tambahan file yang baru

- Root file terdapat penambahan node_modules



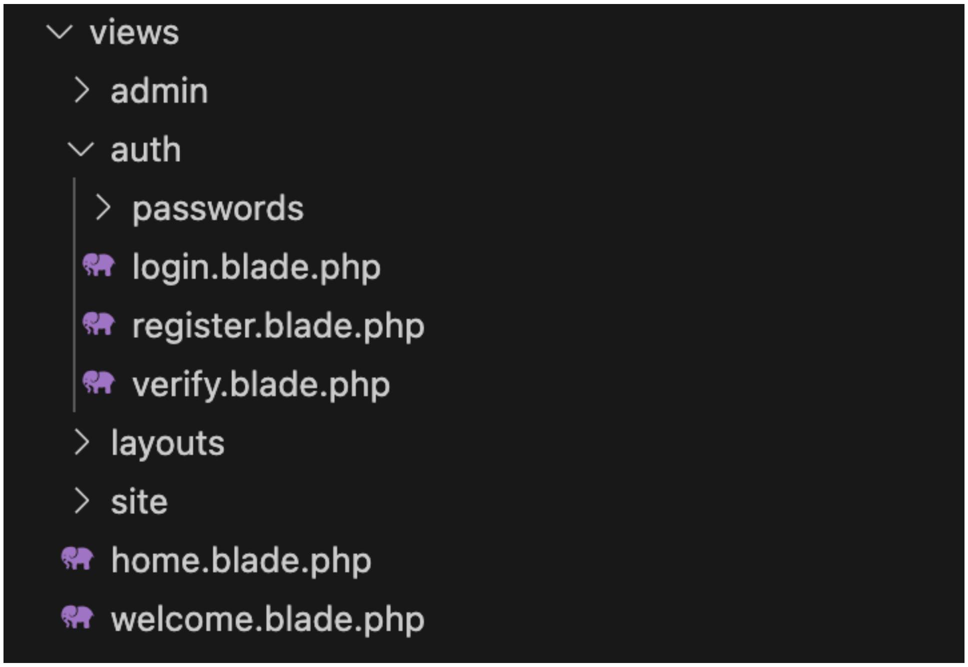
```
∨ LARARSHP
  > app
  > bootstrap
  > config
  > database
  > node_modules
  > public
  > resources
  > routes
  > storage
  > tests
  > vendor
  .editorconfig
  .env
  $ .env.example
```

- Controller terdapat folder auth dan file didalamnya



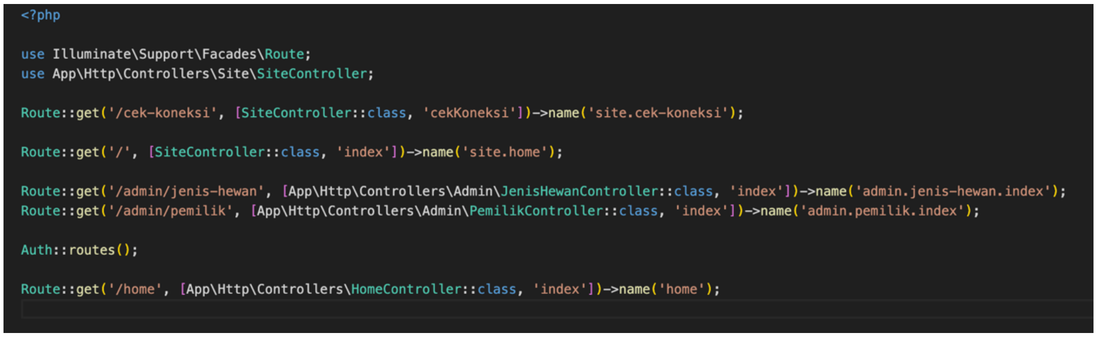
```
∨ Http / Controllers
  > Admin
  ∨ Auth
    ConfirmPasswordController.php
    ForgotPasswordController.php
    LoginController.php
    RegisterController.php
    ResetPasswordController.php
    VerificationController.php
  > Site
  Controller.php
  HomeController.php
```

- Views terdapat folder auth dan file didalamnya



```
views
  admin
  auth
    passwords
    login.blade.php
    register.blade.php
    verify.blade.php
  layouts
  site
  home.blade.php
  welcome.blade.php
```

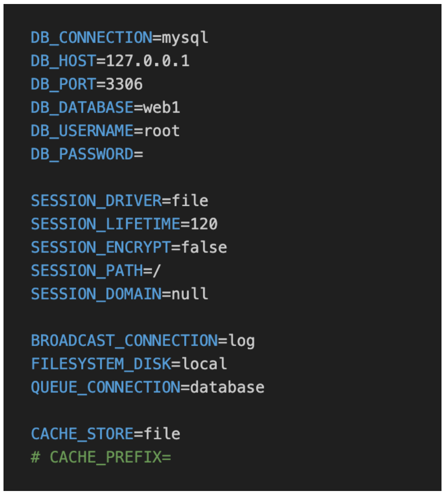
- Routes ada penambahan route baru (auth dan home)



```php
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Site\SiteController;

Route::get('/cek-koneksi', [SiteController::class, 'cekKoneksi'])->name('site.cek-koneksi');

Route::get('/', [SiteController::class, 'index'])->name('site.home');

Route::get('/admin/jenis-hewan', [App\Http\Controllers\Admin\JenisHewanController::class, 'index'])->name('admin.jenis-hewan.index');
Route::get('/admin/pemilik', [App\Http\Controllers\Admin\PemilikController::class, 'index'])->name('admin.pemilik.index');

Auth::routes();

Route::get('/home', [App\Http\Controllers\HomeController::class, 'index'])->name('home');
```

B. Syntac Login

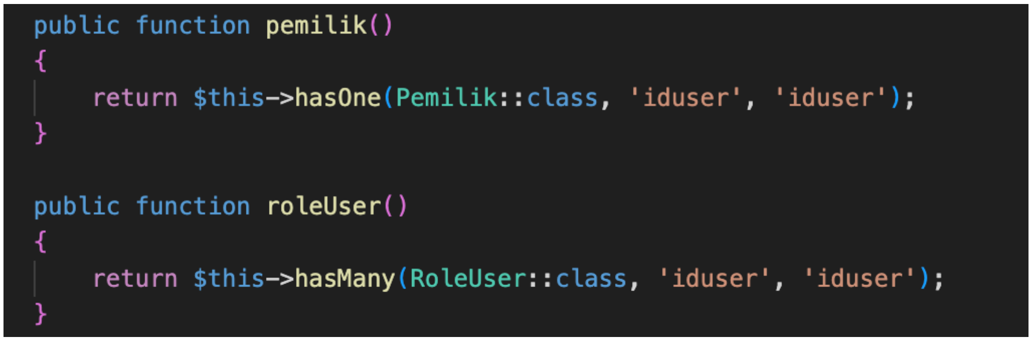
1. Lakukan perubahan pada .env pada cache_store =file



```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=web1
DB_USERNAME=root
DB_PASSWORD=

SESSION_DRIVER=file
SESSION_LIFETIME=120
SESSION_ENCRYPT=false
SESSION_PATH=/
SESSION_DOMAIN=null

BROADCAST_CONNECTION=log
FILESYSTEM_DISK=local
QUEUE_CONNECTION=database

CACHE_STORE=file
# CACHE_PREFIX=
```

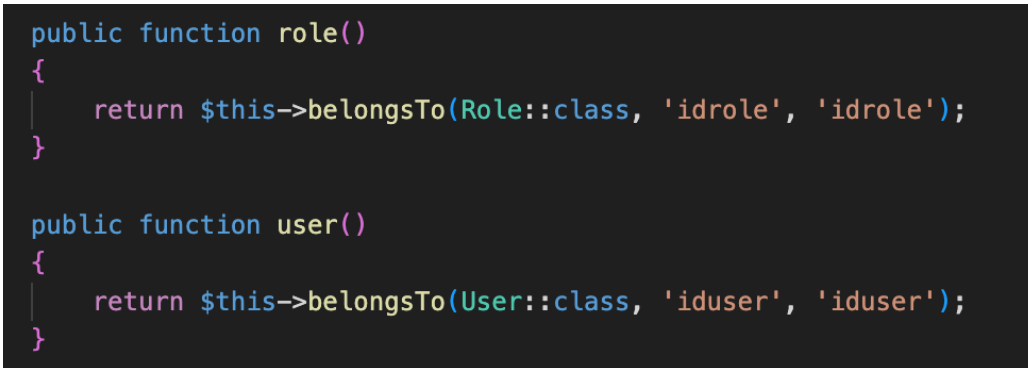
2. Cek dan pastikan pada model user, role dan roleuser sudah terdapat relasi ke masinng-masing model

   - Model user



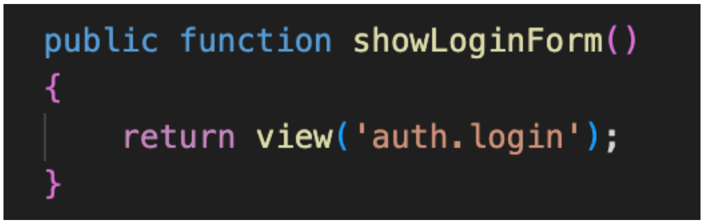
```php
public function pemilik()
{
    return $this->hasOne(Pemilik::class, 'iduser', 'iduser');
}

public function roleUser()
{
    return $this->hasMany(RoleUser::class, 'iduser', 'iduser');
}
```

   - Model roleUser

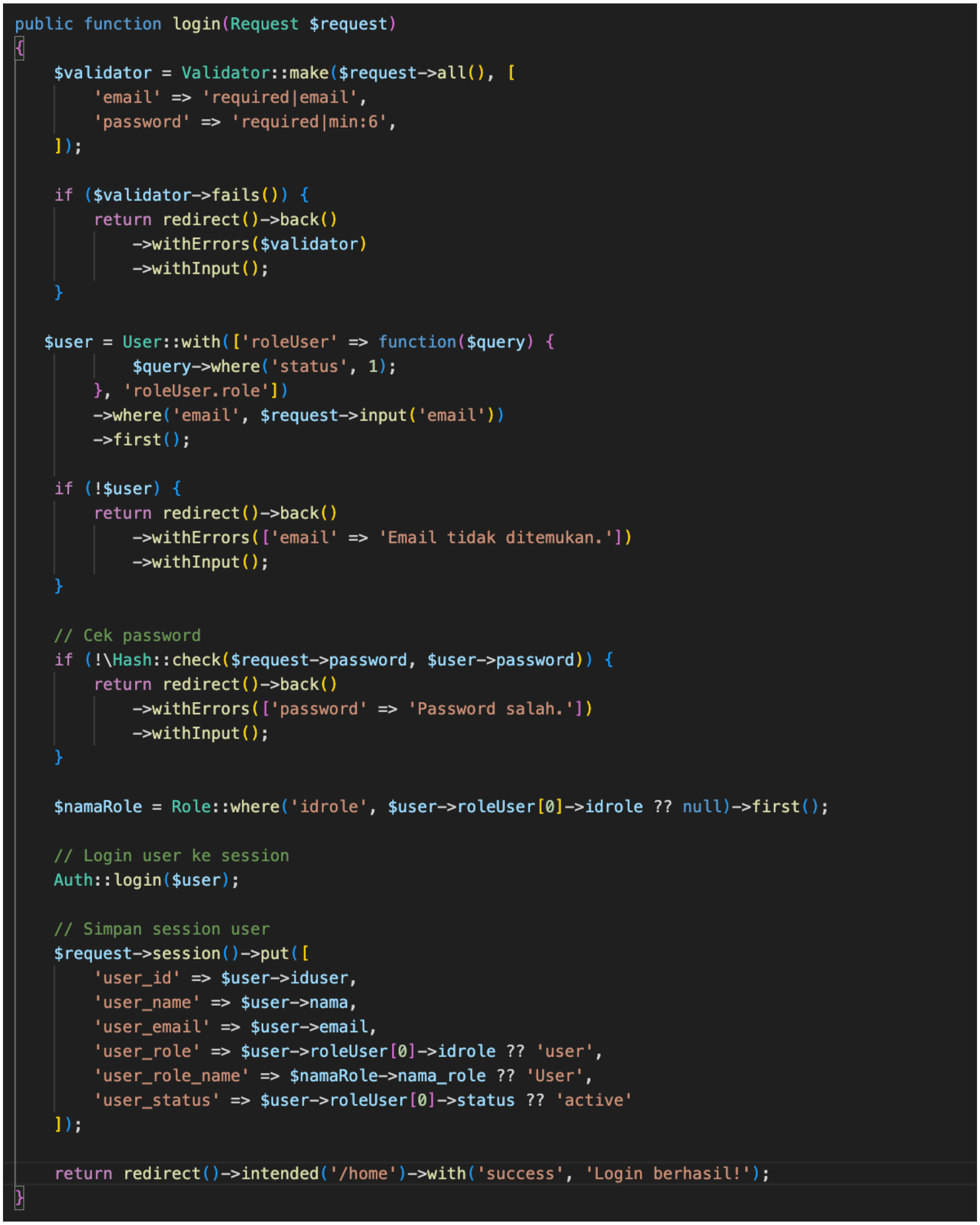


```php
public function role()
{
    return $this->belongsTo(Role::class, 'idrole', 'idrole');
}

public function user()
{
    return $this->belongsTo(User::class, 'iduser', 'iduser');
}
```

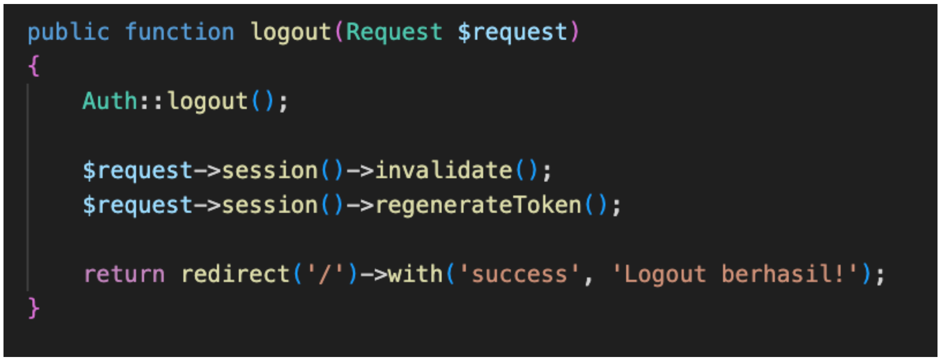
3. Ubah pada Login Controller



```php
public function showLoginForm()
{
    return view('auth.login');
}
```

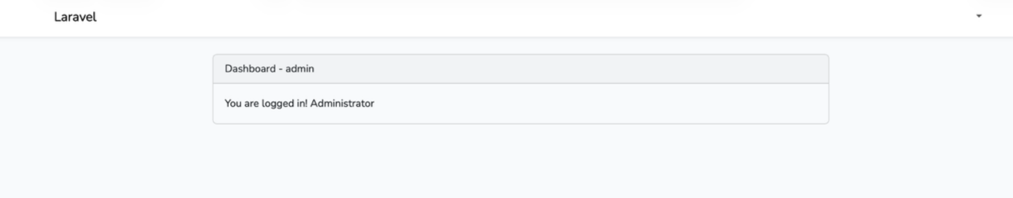


```php
public function login(Request $request)
{
    $validator = Validator::make($request->all(), [
        'email' => 'required|email',
        'password' => 'required|min:6',
    ]);

    if ($validator->fails()) {
        return redirect()->back()
            ->withErrors($validator)
            ->withInput();
    }

   $user = User::with(['roleUser' => function($query) {
            $query->where('status', 1);
        }, 'roleUser.role'])
        ->where('email', $request->input('email'))
        ->first();

    if (!$user) {
        return redirect()->back()
            ->withErrors(['email' => 'Email tidak ditemukan.'])
            ->withInput();
    }

    // Cek password
    if (!\Hash::check($request->password, $user->password)) {
        return redirect()->back()
            ->withErrors(['password' => 'Password salah.'])
            ->withInput();
    }

    $namaRole = Role::where('idrole', $user->roleUser[0]->idrole ?? null)->first();

    // Login user ke session
    Auth::login($user);

    // Simpan session user
    $request->session()->put([
        'user_id' => $user->iduser,
        'user_name' => $user->nama,
        'user_email' => $user->email,
        'user_role' => $user->roleUser[0]->idrole ?? 'user',
        'user_role_name' => $namaRole->nama_role ?? 'User',
        'user_status' => $user->roleUser[0]->status ?? 'active'
    ]);

    return redirect()->intended('/home')->with('success', 'Login berhasil!');
}
```



```php
public function logout(Request $request)
{
    Auth::logout();

    $request->session()->invalidate();
    $request->session()->regenerateToken();

    return redirect('/')->with('success', 'Logout berhasil!');
}
```

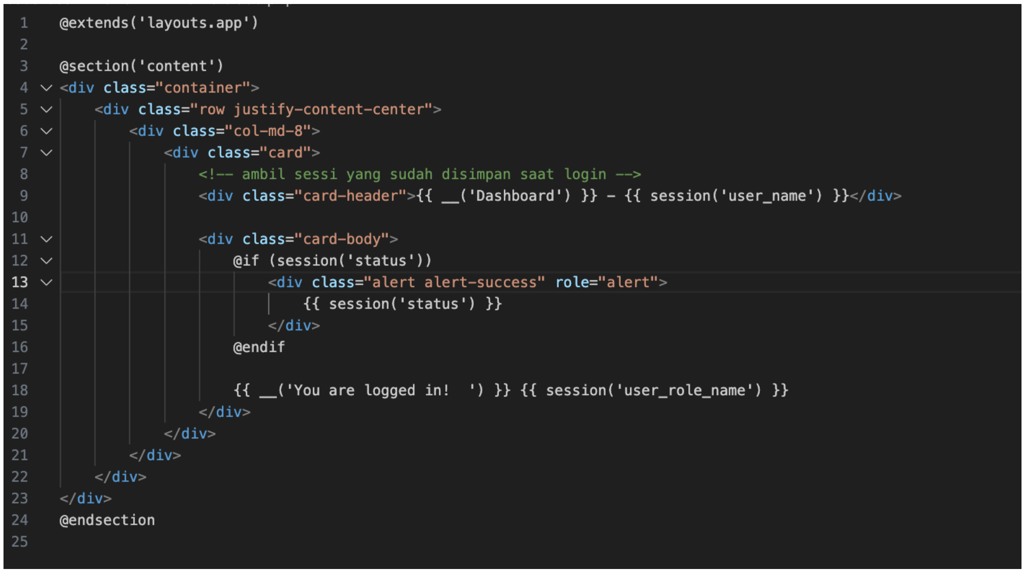
4. Ketika proses dilakukan login maka akan menuju halaman dashboard home



Laravel

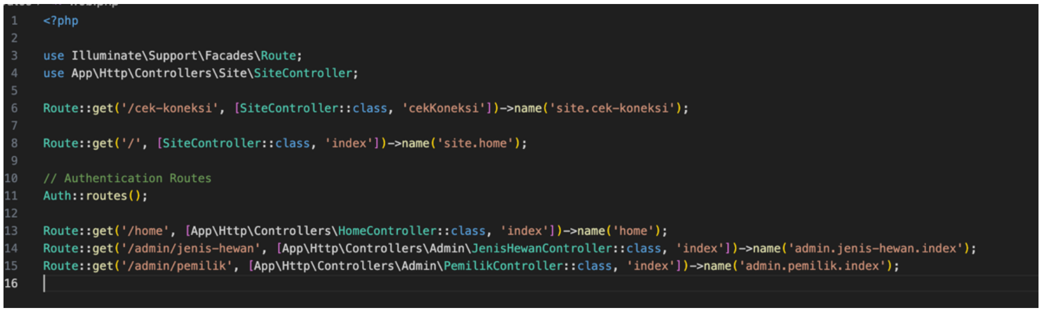
Dashboard - admin

You are logged in! Administrator



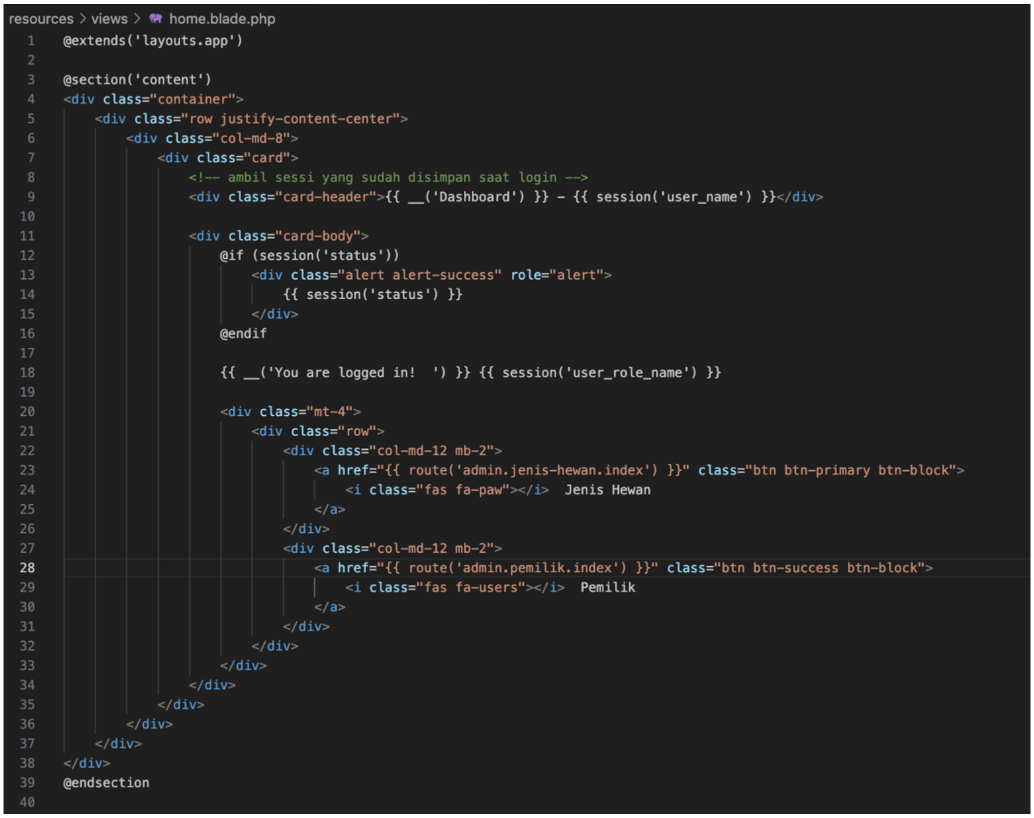
```
@extends('layouts.app')

@section('content')
<div class="container">
    <div class="row justify-content-center">
        <div class="col-md-8">
            <div class="card">
                <!-- ambil sessi yang sudah disimpan saat login -->
                <div class="card-header">{{ __('Dashboard') }} - {{ session('user_name') }}</div>

                <div class="card-body">
                    @if (session('status'))
                        <div class="alert alert-success" role="alert">
                            {{ session('status') }}
                        </div>
                    @endif

                    {{ __('You are logged in!  ') }} {{ session('user_role_name') }}
                </div>
            </div>
        </div>
    </div>
</div>
@endsection

```

5. Ubah route menjadi seperti contoh, pindahkan semua route yang sudah dibuat kebawah dari Auth:routes():



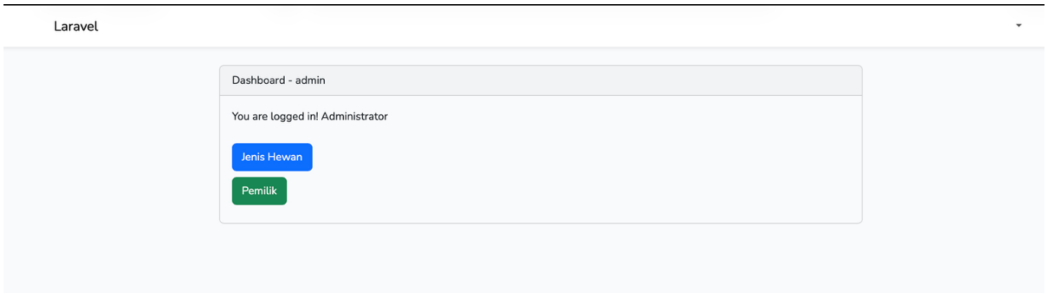
```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Site\SiteController;

Route::get('/cek-koneksi', [SiteController::class, 'cekKoneksi'])->name('site.cek-koneksi');

Route::get('/', [SiteController::class, 'index'])->name('site.home');

// Authentication Routes
Auth::routes();

Route::get('/home', [App\Http\Controllers\HomeController::class, 'index'])->name('home');
Route::get('/admin/jenis-hewan', [App\Http\Controllers\Admin\JenisHewanController::class, 'index'])->name('admin.jenis-hewan.index');
Route::get('/admin/pemilik', [App\Http\Controllers\Admin\PemilikController::class, 'index'])->name('admin.pemilik.index');

```

6. Dan ubah dashboard home, supaya bisa memanggil masing-masing route yang untuk admin



resources > views > home.blade.php

```
@extends('layouts.app')

@section('content')
<div class="container">
    <div class="row justify-content-center">
        <div class="col-md-8">
            <div class="card">
                <!-- ambil sessi yang sudah disimpan saat login -->
                <div class="card-header">{{ __('Dashboard') }} - {{ session('user_name') }}</div>

                <div class="card-body">
                    @if (session('status'))
                        <div class="alert alert-success" role="alert">
                            {{ session('status') }}
                        </div>
                    @endif

                    {{ __('You are logged in!  ') }} {{ session('user_role_name') }}

                    <div class="mt-4">
                        <div class="row">
                            <div class="col-md-12 mb-2">
                                <a href="{{ route('admin.jenis-hewan.index') }}" class="btn btn-primary btn-block">
                                    <i class="fas fa-paw"></i>  Jenis Hewan
                                </a>
                            </div>
                            <div class="col-md-12 mb-2">
                                <a href="{{ route('admin.pemilik.index') }}" class="btn btn-success btn-block">
                                    <i class="fas fa-users"></i>  Pemilik
                                </a>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
@endsection
```



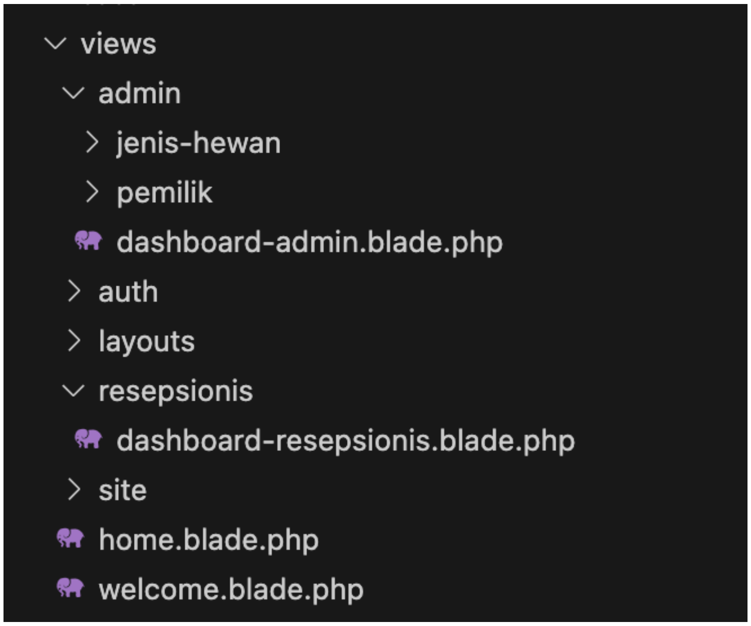
Dashboard - admin

You are logged in! Administrator

Setelah route dipindahkan dibawah auth, maka hanya bisa diakses setelah login.

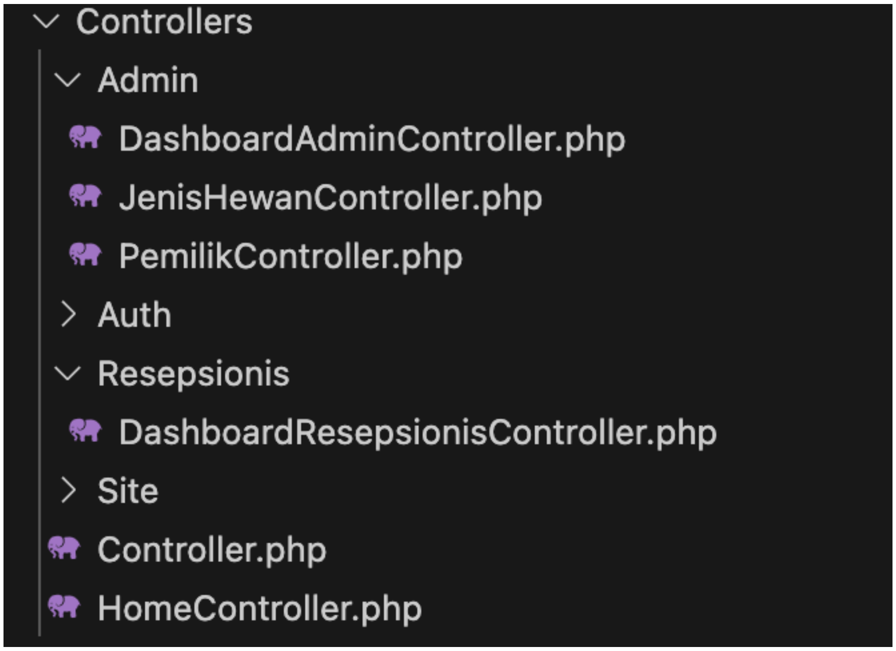
C. Middleware

1. Buat halaman dashboard terlebih dahulu untuk dashboard administrator dan dashboard resepsionis

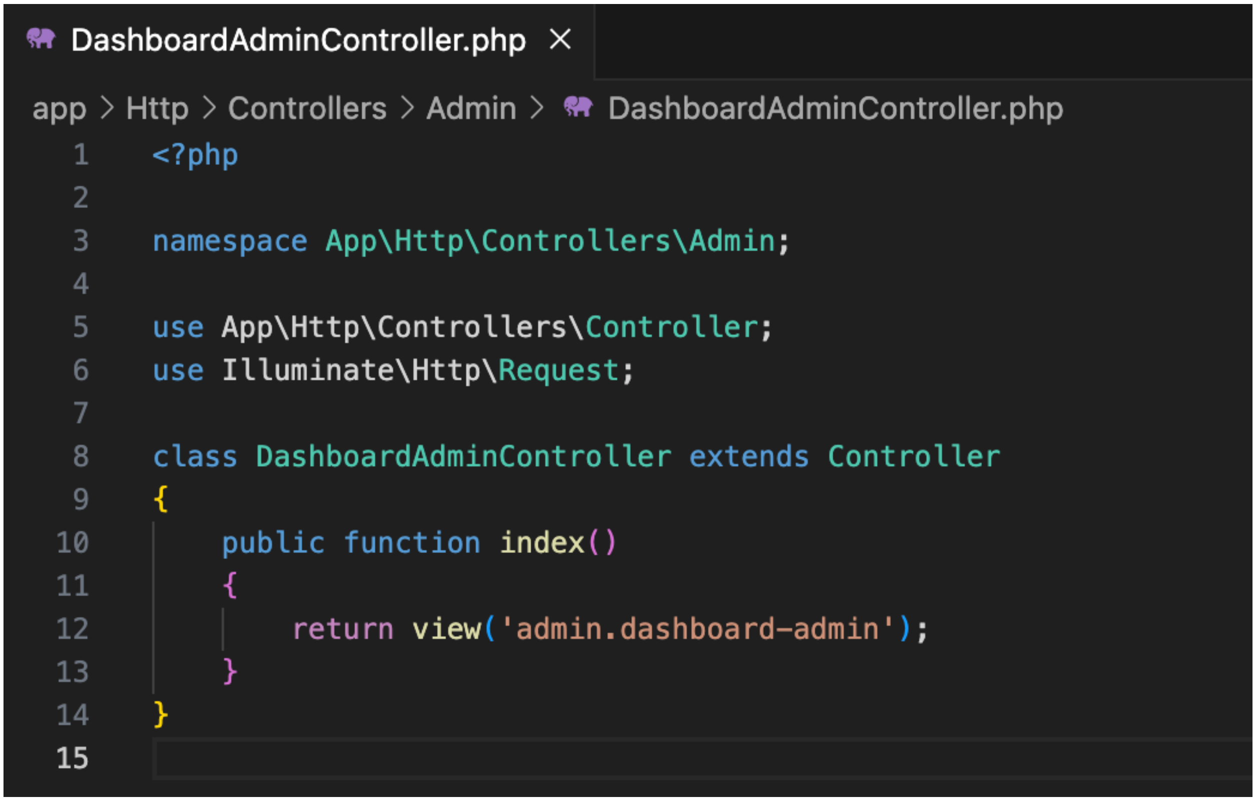


```
v views
  v admin
    > jenis-hewan
    > pemilik
    dashboard-admin.blade.php
  > auth
  > layouts
  v resepsionis
    dashboard-resepsionis.blade.php
  > site
  home.blade.php
  welcome.blade.php
```

2. Buat controller untuk dashboard administrator dan dashboard resepsionis

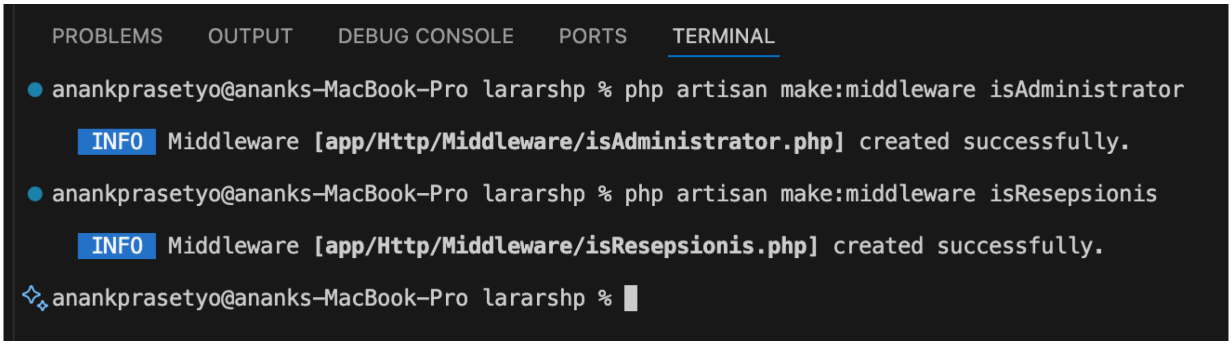


```
v Controllers
  v Admin
    DashboardAdminController.php
    JenisHewanController.php
    PemilikController.php
  > Auth
  v Resepsionis
    DashboardResepsionisController.php
  > Site
  Controller.php
  HomeController.php
```


DashboardAdminController.php
app > Http > Controllers > Admin > DashboardAdminController.php

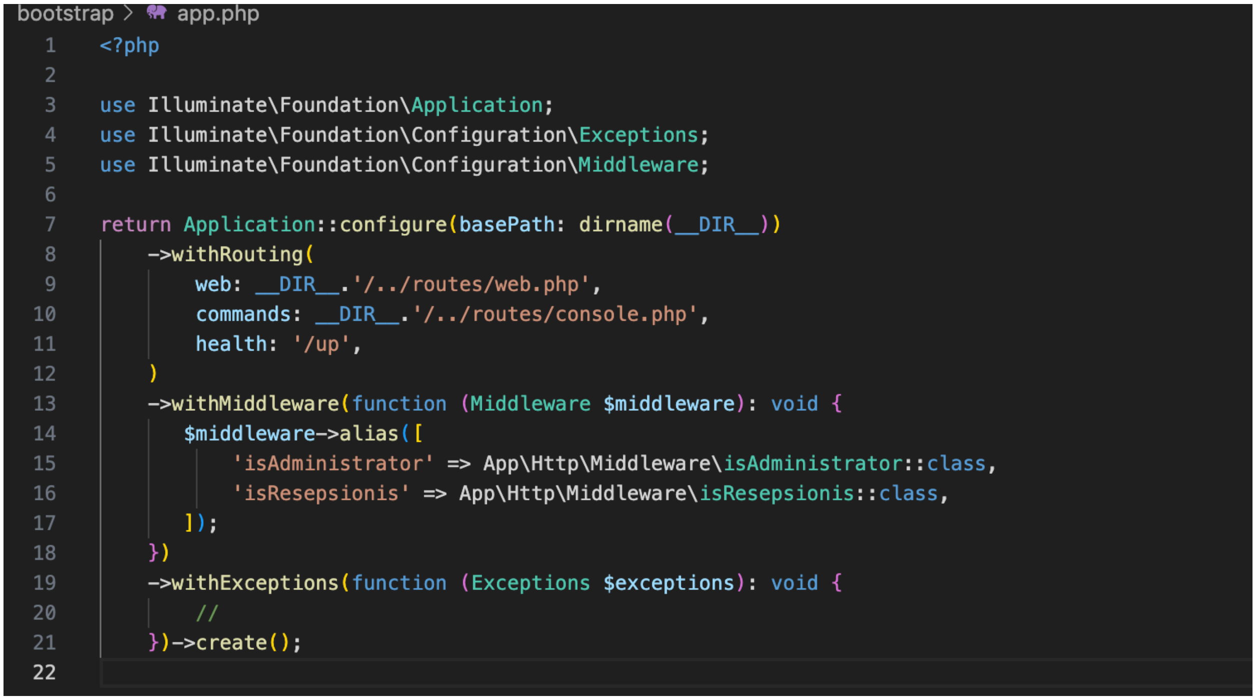
```php
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;

class DashboardAdminController extends Controller
{
    public function index()
    {
        return view('admin.dashboard-admin');
    }
}

```

3. Buat middleware sesuai dengan role base akses contoh role administrator dan resepsionis



```
● anankprasetyo@ananks-MacBook-Pro lararshp % php artisan make:middleware isAdministrator

   INFO  Middleware [app/Http/Middleware/isAdministrator.php] created successfully.

● anankprasetyo@ananks-MacBook-Pro lararshp % php artisan make:middleware isResepsionis

   INFO  Middleware [app/Http/Middleware/isResepsionis.php] created successfully.

 anankprasetyo@ananks-MacBook-Pro lararshp % █
```

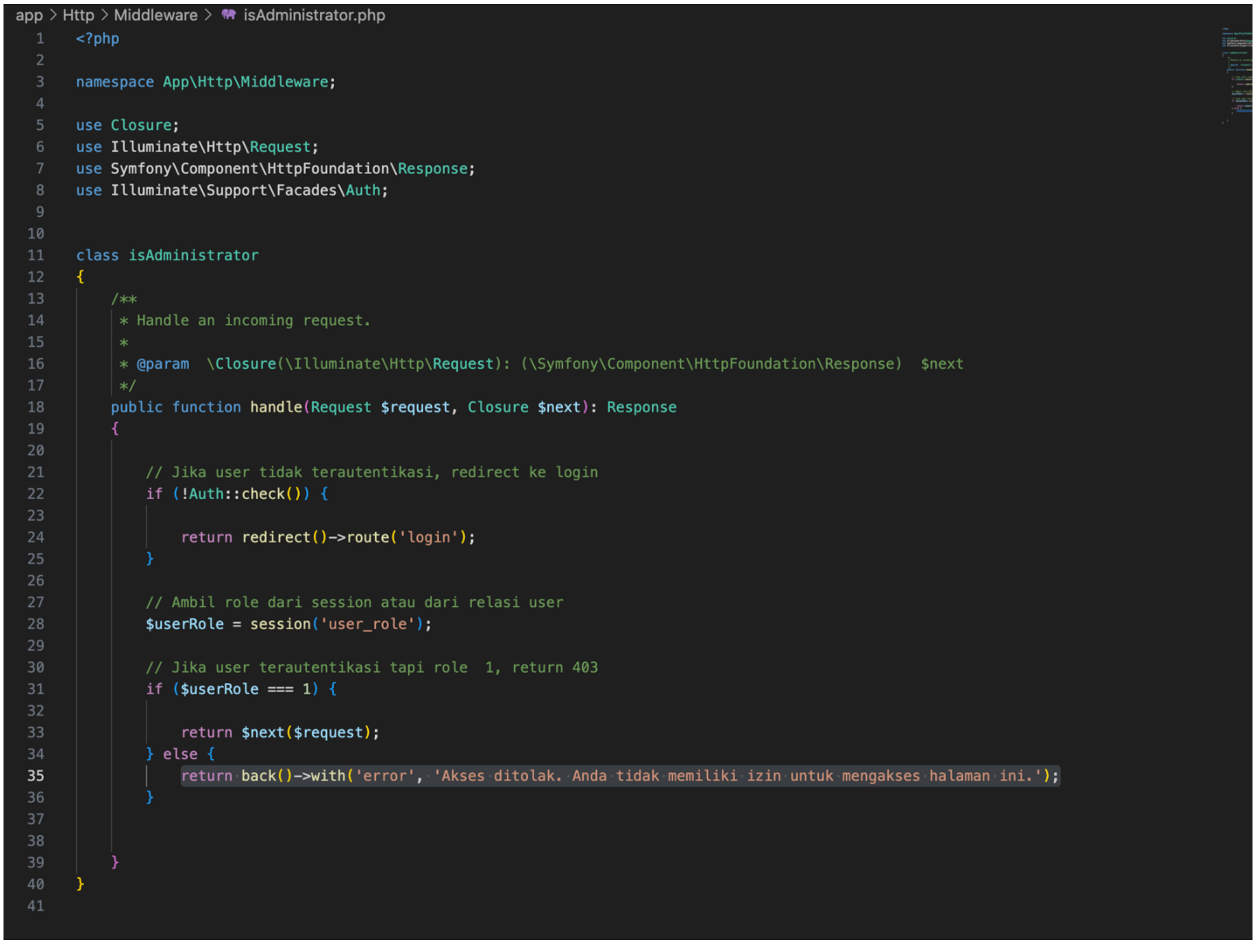
4. Buka file bootstrap/app.php dan tambahkan middleware yang sudah dibuat



bootstrap > app.php

```php
<?php

use Illuminate\Foundation\Application;
use Illuminate\Foundation\Configuration\Exceptions;
use Illuminate\Foundation\Configuration\Middleware;

return Application::configure(basePath: dirname(__DIR__))
    ->withRouting(
        web: __DIR__.'/../routes/web.php',
        commands: __DIR__.'/../routes/console.php',
        health: '/up',
    )
    ->withMiddleware(function (Middleware $middleware): void {
        $middleware->alias([
            'isAdministrator' => App\Http\Middleware\isAdministrator::class,
            'isResepsionis' => App\Http\Middleware\isResepsionis::class,
        ]);
    })
    ->withExceptions(function (Exceptions $exceptions): void {
        //
    })->create();

```

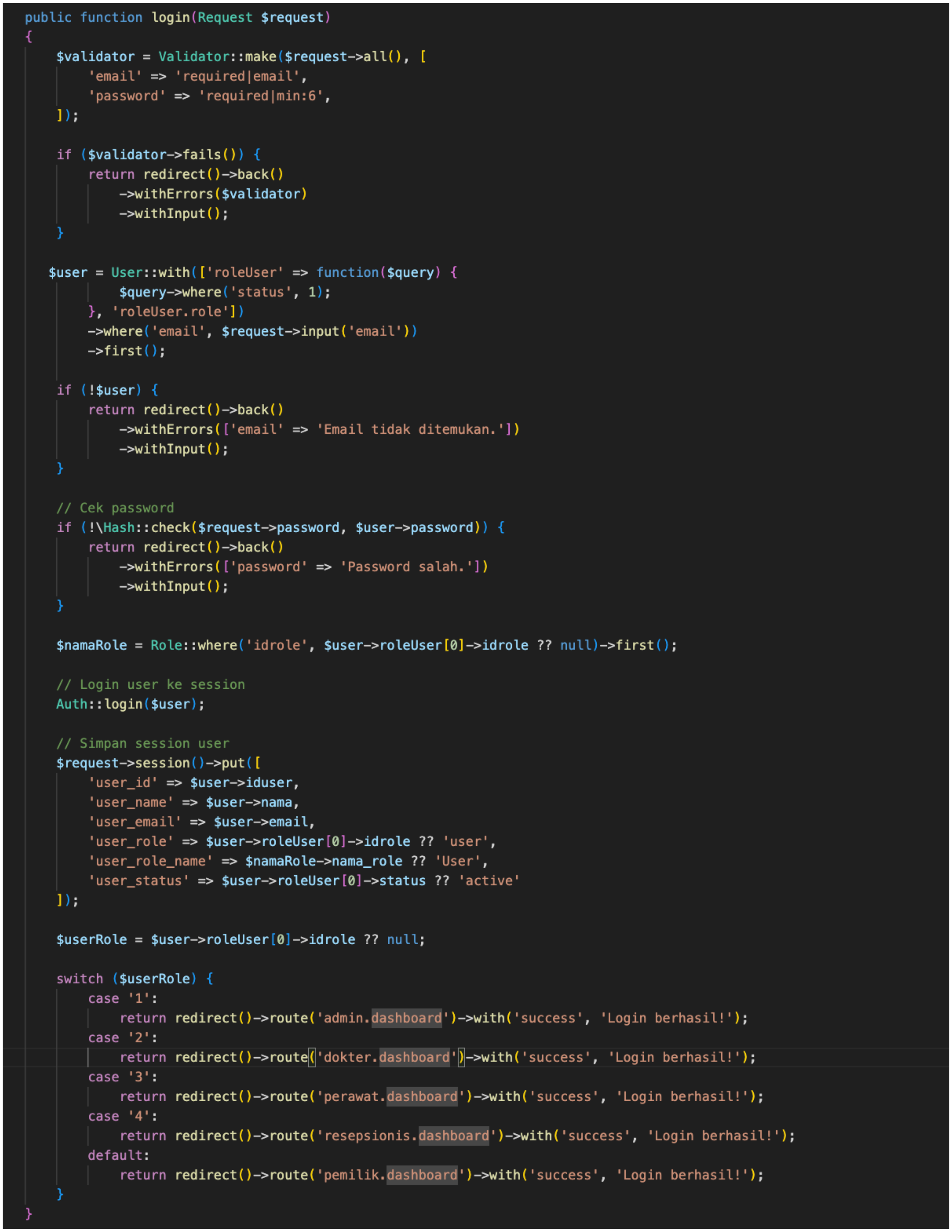
5. Buka file middleware dan lakukan perubahan pada middleware isAdministrator dan isResepsionis



app > Http > Middleware > isAdministrator.php

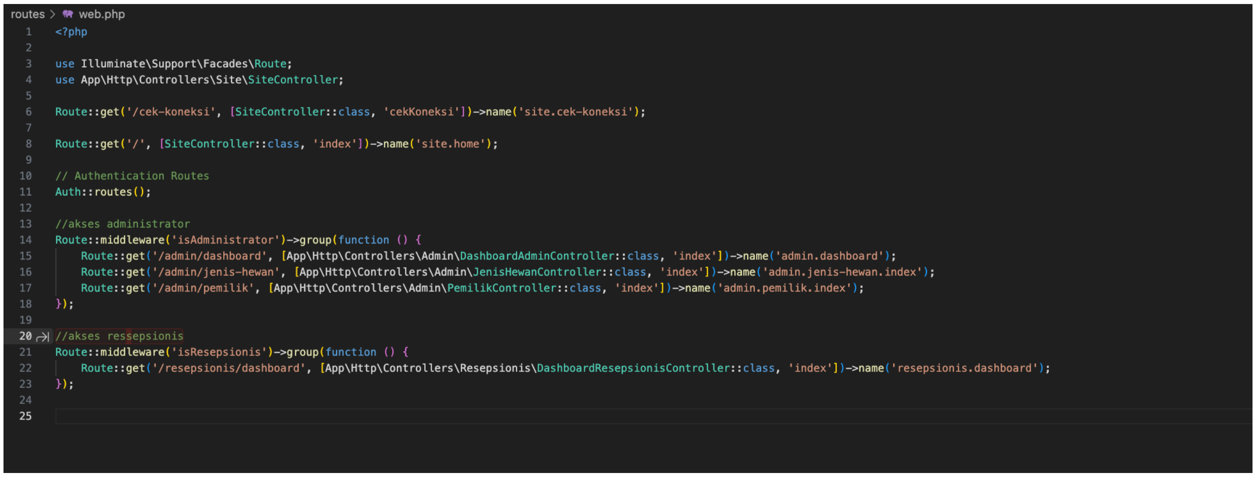
```php
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;
use Symfony\Component\HttpFoundation\Response;
use Illuminate\Support\Facades\Auth;


class isAdministrator
{
    /**
     * Handle an incoming request.
     *
     * @param  \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response)  $next
     */
    public function handle(Request $request, Closure $next): Response
    {

        // Jika user tidak terautentikasi, redirect ke login
        if (!Auth::check()) {

            return redirect()->route('login');
        }

        // Ambil role dari session atau dari relasi user
        $userRole = session('user_role');

        // Jika user terautentikasi tapi role  1, return 403
        if ($userRole === 1) {

            return $next($request);
        } else {
            return back()->with('error', 'Akses ditolak. Anda tidak memiliki izin untuk mengakses halaman ini.');
        }


    }
}

```

6. Ubah function pada logincontroller



```
public function login(Request $request)
{
    $validator = Validator::make($request->all(), [
        'email' => 'required|email',
        'password' => 'required|min:6',
    ]);

    if ($validator->fails()) {
        return redirect()->back()
            ->withErrors($validator)
            ->withInput();
    }

   $user = User::with(['roleUser' => function($query) {
            $query->where('status', 1);
        }, 'roleUser.role'])
        ->where('email', $request->input('email'))
        ->first();

    if (!$user) {
        return redirect()->back()
            ->withErrors(['email' => 'Email tidak ditemukan.'])
            ->withInput();
    }

    // Cek password
    if (!\Hash::check($request->password, $user->password)) {
        return redirect()->back()
            ->withErrors(['password' => 'Password salah.'])
            ->withInput();
    }

    $namaRole = Role::where('idrole', $user->roleUser[0]->idrole ?? null)->first();

    // Login user ke session
    Auth::login($user);

    // Simpan session user
    $request->session()->put([
        'user_id' => $user->iduser,
        'user_name' => $user->nama,
        'user_email' => $user->email,
        'user_role' => $user->roleUser[0]->idrole ?? 'user',
        'user_role_name' => $namaRole->nama_role ?? 'User',
        'user_status' => $user->roleUser[0]->status ?? 'active'
    ]);

    $userRole = $user->roleUser[0]->idrole ?? null;

    switch ($userRole) {
        case '1':
            return redirect()->route('admin.dashboard')->with('success', 'Login berhasil!');
        case '2':
            return redirect()->route('dokter.dashboard')->with('success', 'Login berhasil!');
        case '3':
            return redirect()->route('perawat.dashboard')->with('success', 'Login berhasil!');
        case '4':
            return redirect()->route('resepsionis.dashboard')->with('success', 'Login berhasil!');
        default:
            return redirect()->route('pemilik.dashboard')->with('success', 'Login berhasil!');
    }
}
```

7. Ubah route sesuai dengan middleware masing-masing



routes > web.php

```php
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Site\SiteController;

Route::get('/cek-koneksi', [SiteController::class, 'cekKoneksi'])->name('site.cek-koneksi');

Route::get('/', [SiteController::class, 'index'])->name('site.home');

// Authentication Routes
Auth::routes();

//akses administrator
Route::middleware('isAdministrator')->group(function () {
    Route::get('/admin/dashboard', [App\Http\Controllers\Admin\DashboardAdminController::class, 'index'])->name('admin.dashboard');
    Route::get('/admin/jenis-hewan', [App\Http\Controllers\Admin\JenisHewanController::class, 'index'])->name('admin.jenis-hewan.index');
    Route::get('/admin/pemilik', [App\Http\Controllers\Admin\PemilikController::class, 'index'])->name('admin.pemilik.index');
});

//akses ressepsionis
Route::middleware('isResepsionis')->group(function () {
    Route::get('/resepsionis/dashboard', [App\Http\Controllers\Resepsionis\DashboardResepsionisController::class, 'index'])->name('resepsionis.dashboard');
});

```

Maka ketika sedang login admin, mengkases urlnya respsionis akan tertolak

D. Tugas

1. Buat middleware untuk masing-masing role base (administrator, dokter, perawat, resepsionis dan pemilik)
2. Buat dashboard untuk masing-masing role base
3. Tampilkan data-data (hanya read saja) sesuai dengan masing-masing dari role base yang dimiliki. Contoh admin menampilkan semua data master, resepsionis menampilkan data pendaftaran, pemilik menampilkan data pet

Selamat mencoba dan mengerjakan