

LAPORAN PRAKTIKUM
PEMROGRAMAN WEB LANJUT



DISUSUN OLEH:
THORIQ FATHURROZI

2H/28

2241720052

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2024



Jurusan Teknologi Informasi Politeknik Negeri Malang
Pemrograman Web Lanjut
Mata Kuliah Pemrograman Web Lanjut
Pengampu: Tim Ajar Pemrograman Web Lanjut
April 2024

Nama	:	Thoriq Fathurrozi
Nim	:	2241720052
Kelas	:	2H

1.	<p>Sebelum melangkah lebih lanjut setting file pada config/auth.php ubah pada bagian model menjadi model yang telah kita buat sebelumnya untuk menyimpan username dan password</p> <pre>thoriq fathurrozi, 2 months ago • initial commit</pre> <pre>'providers' => ['users' => ['driver' => 'eloquent', 'model' => App\Models\UserModel::class,],],</pre> <p>kode tersebut digunakan acuan model untuk pengecekan auth</p>
2.	<p>Membuat middleware dengan perintah pada cmd atau terminal</p> <pre>frrzy@Thoriqs-MacBook-Pro PWL_POS % php artisan make:middleware Cek_login</pre> <pre>INFO Middleware [app/Http/Middleware/Cek_login.php] created successfully.</pre> <p>Setelah itu cek didalam folder app/Http/Middleware apakah sudah ada file bernama Cek_login.php</p> <p>perintah tersebut digunakan untuk membuat middleware</p>
3.	<p>Kemudian tambahkan kode dibawah ini, letakkan di function handle()</p>



Jurusan Teknologi Informasi Politeknik Negeri Malang
Pemrograman Web Lanjut
Mata Kuliah Pemrograman Web Lanjut
 Pengampu: Tim Ajar Pemrograman Web Lanjut
 April 2024

	<pre> <?php namespace App\Http\Middleware; use Closure; use Illuminate\Http\Request; use Symfony\Component\HttpFoundation\Response; use Illuminate\Support\Facades\Auth; 1 reference 0 implementations class Cek_login { /** * Handle an incoming request. * * @param \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response) \$next */ 0 references 0 overrides public function handle(Request \$request, Closure \$next, \$roles): Response { // return \$next(\$request); // cek sudah login atau belum . jika belum kembali ke halaman login if (!Auth::check()) { return redirect('login'); } // simpan data user pada variabel \$user \$user = Auth::user(); // jika user memiliki level sesuai pada kolom lanjutkan request if (\$user->level_id == \$roles) { return \$next(\$request); } // jika tidak memiliki akses maka kembalikan ke halaman login return redirect('login')->with('error', 'Maaf anda tidak memiliki akses'); // return \$next(\$request); } } </pre> <p>class Cek_login digunakan untuk mengecek apakah sudah terautentikasi jika sudah maka akan dilanjutkan ke request setelahnya jika tidak maka akan dikembalikan ke halaman login</p>
4.	<p>Ketika sudah membuat middleware, langkah berikutnya harus registrasikan pada kernel.php. Agar middleware dapat dibaca oleh sistem. Buka folder App/Http/Kernel.php dan tambahkan code pada variabel \$middlewareAliases seperti dibawah</p>



Jurusan Teknologi Informasi Politeknik Negeri Malang
Pemrograman Web Lanjut
Mata Kuliah Pemrograman Web Lanjut
Pengampu: Tim Ajar Pemrograman Web Lanjut
April 2024

	<pre>0 references protected \$middlewareAliases = ['auth' => \App\Http\Middleware\Authenticate::class, 'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class, 'auth.session' => \Illuminate\Session\Middleware\AuthenticateSession::class, 'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class, 'can' => \Illuminate\Auth\Middleware\Authorize::class, 'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class, 'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class, 'precognitive' => \Illuminate\Foundation\Http\Middleware\HandlePrecognitiveRequests::class, 'signed' => \App\Http\Middleware\ValidateSignature::class, 'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class, 'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class, // tambahkan middleware alias dengan nama cek_login 'cek_login' => \App\Http\Middleware\Cek_Login::class,];</pre> <p>kode tersebut digunakan untuk menambahkan middleware kedalam route agar dapat digunakan</p>
5.	<p>Membuat Controller Auth. Pada langkah ini kita akan membuat controller auth, yang dimana di dalamnya terdapat proses autentikasi dari request login. Jalankan perintah make:controller</p> <pre>● frrzy@Thoriqs-MacBook-Pro PWL_POS % php artisan make:controller AuthController INFO Controller [app/Http/Controllers/AuthController.php] created successfully.</pre> <p>Buka file App/Http/Controllers/AuthController.php yang sudah kita buat. Lalu kita akan isi perintah code untuk membuat proses autentikasi , verifikasi, dan logout perintah tersebut digunakan untuk membuat controller yang dinamai AuthController</p>
6.	<p>Kemudian buat kode AuthController.php seperti dibawah ini</p>



Jurusan Teknologi Informasi Politeknik Negeri Malang
Pemrograman Web Lanjut
Mata Kuliah Pemrograman Web Lanjut
Pengampu: Tim Ajar Pemrograman Web Lanjut
April 2024

```
<?php

namespace App\Http\Controllers;

use App\Models\UserModel;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;

6 references | 0 implementations
class AuthController extends Controller
{
    1 reference | 0 overrides
    public function index()
    {
        // kita ambil data user lalu disimpan pada variabel user
        $user = Auth::user();

        // kondisi jika usernya ada
        if ($user) {
            // jika user nya memiliki level admin
            if ($user->level_id == '1') {
                return redirect()->intended('admin');
            }
            // jika usernya memiliki level manager
            else if ($user->level_id == '2') {
                return redirect()->intended('manager');
            }
        }
        return view('login');
    }
}
```

```
1 reference | 0 overrides
public function proses_login(Request $request)
{
    // kita buat validasi pada saat tombol login di klik
    // validasi nya username & password wajib di isi

    $request->validate([
        'username' => 'required',
        'password' => 'required'
    ]);

    // ambil data request username & password saja
    $credential = $request->only('username', 'password');
    // cek jika data username dan password valid (sesuai) dengan data
    if (Auth::attempt($credential)) {
        // kalau berhasil disimpan data user ya di variabel $user
        $user = Auth::user();

        // cek lagi jika level user admin maka arahkan kehalaman admin
        if ($user->level_id == '1') {
            // dd($user->level_id)
            return redirect()->intended('admin');
        }
        // tapi jika level usernya user biasa maka arahkan ke halaman user
        else {
            return redirect()->intended('manager');
        }
        // jika belum ada role maka ke halaman /
        return redirect()->intended('/');
    }
    // jika ga ada data user yang valid maka kembalikan lagi ke halaman login
    // pastikan kirim pesan error juga kalau login gagal ya
    return redirect('login')
        ->withInput()
        ->withErrors(['login_gagal' => 'Pastikan kembali username dan password yang dimasukan sudah benar']);
}
```



```
1 reference | 0 overrides
public function register()
{
    // tampilkan view register
    return view('register');
}

// aksi form register
1 reference | 0 overrides
public function proses_register(Request $request)
{
    // kita buat validasi nih buat proses register
    // validasinya yaitu semua field wajib di isi
    // validasi username itu harus unique atau tidak boleh duplicate username ya
    $validator = Validator::make($request->all(), [
        'nama' => 'required',
        'username' => 'required|unique:m_user',
        'password' => 'required',
    ]);

    // kalau gagal kembali ke halaman register dengan memunculkan pesan error
    if ($validator->fails()) {
        return redirect('/register')
            ->withErrors($validator)
            ->withInput();
    }

    // kalau berhasil isi level dan hash passwordnya ya biar secure
    $request['level_id'] = '2';
    $request['password'] = Hash::make($request->password);

    // masukan semua data pada request ke table user
    UserModel::create($request->all());

    // kalo berhasil arahkan ke halaman login
    return redirect()->route('login');
}
```

```
0 references | 0 overrides
public function logout(Request $request)
{
    // logout itu harus menghapus sessionnya
    $request->session()->flush();

    // jalankan juga fungsi logout pada auth
    Auth::logout();
    // kembali ke halaman login
    return redirect('login');
}
```

kode tersebut digunakan untuk menghandle proses dari autentikasi seperti login dan register



Jurusan Teknologi Informasi Politeknik Negeri Malang
Pemrograman Web Lanjut
Mata Kuliah Pemrograman Web Lanjut
Pengampu: Tim Ajar Pemrograman Web Lanjut
April 2024

7.	<p>Kemudian tambahkan kode pada route pada file routes/web.php seperti kode program dibawah</p> <pre>Route::get('login', [AuthController::class, 'index'])->name('login'); Route::get('register', [AuthController::class, 'register'])->name('register'); Route::post('proses_login', [AuthController::class, 'proses_login'])->name('proses_login'); Route::get('logout', [AuthController::class, 'logout'])->name('logout'); Route::post('proses_register', [AuthController::class, 'proses_register'])->name('proses_register'); // kita atur juga untuk middleware menggunakan group pada routing // didalamnya terdapat group untuk mengecek kondisi login // jika user yang login merupakan admin maka akan diarahkan ke AdminController // jika user yang login merupakan manager maka akan diarahkan ke UserController Route::group(['middleware' => ['auth']], function () { Route::group(['middleware' => ['cek_login:1']], function () { Route::resource('admin', AdminController::class); }); Route::group(['middleware' => ['cek_login:2']], function () { Route::resource('manager', ManagerController::class); }); });</pre> <p>kode tersebut untuk menghandle route dari proses autentikasi</p>
8.	<p>Kemudian buat file controller dengan nama AdminController sebagai halaman yang boleh diakses oleh admin saja. Jalankan perintah dibawah</p> <pre>frrzy@Thoriqs-MacBook-Pro PWL_POS % php artisan make:controller AdminController INFO Controller [app/Http/Controllers/AdminController.php] created successfully.</pre> <p>perintah tersebut digunakan untuk membuat controller yang dinamai AdminController</p>
9.	<p>Buka folder App/Http/Controllers dan isi file AdminController seperti dibawah</p>



	<pre>2 references 0 implementations class AdminController extends Controller { 0 references 0 overrides public function index() { return view('admin'); } }</pre> <p>kode tersebut digunakan untuk menghandle tampilan dari admin</p>
10.	<p>Kemudian buat file controller dengan nama ManagerController sebagai halaman yang boleh diakses oleh admin saja. Jalankan perintah dibawah</p> <pre>frrzy@Thoriqs-MacBook-Pro PWL_POS % php artisan make:controller ManagerController INFO Controller [app/Http/Controllers/ManagerController.php] created successfully.</pre> <p>perintah tersebut digunakan untuk membuat controller yang dinamai ManagerController</p>
11.	<p>Buka folder App/Http/Controllers dan isi file ManagerController seperti dibawah</p>



	<div><div>2 references 0 implementations</div><pre>class ManagerController extends Controller { 0 references 0 overrides public function index() { return view('manager'); } }</pre></div> <p>kode tersebut digunakan untuk menghandle tampilan dari manager</p>
12.	Pada langkah terakhir yaitu membuat layout sederhana
13.	<p>Buatlah halaman login dengan membuka folder resources/views dengan nama file login.blade.php. Lalu kita isi filenya seperti contoh dibawah</p> <pre>resources > views > login.blade.php > div.alert.alert-warning.alert-dismissible.fade.show > button.close @extends('adminlte::auth.auth-page', ['auth_type' => 'login']) {{-- Menambahkan CSS tambahan --}} @section('adminlte_css_pre') <link rel="stylesheet" href="{{ asset('vendor/icheck-bootstrap/icheck-bootstrap.min.css') }}"> @stop {{-- Mengatur URL untuk login, register, dan reset password --}} @php(\$login_url = View::getSection('login_url') ?? config('adminlte.login_url', 'login')) @php(\$register_url = View::getSection('register_url') ?? config('adminlte.register_url', 'register')) @php(\$password_reset_url = View::getSection('password_reset_url') ?? config('adminlte.password_reset_url', 'password/reset')) @if (config('adminlte.use_route_url', false)) @php(\$login_url = \$login_url ? route(\$login_url) : '') @php(\$register_url = \$register_url ? route(\$register_url) : '') @php(\$password_reset_url = \$password_reset_url ? route(\$password_reset_url) : '') @else @php(\$login_url = \$login_url ? url(\$login_url) : '') @php(\$register_url = \$register_url ? url(\$register_url) : '') @php(\$password_reset_url = \$password_reset_url ? url(\$password_reset_url) : '') @endif {{-- Mengatur judul halaman login --}} @section('auth_header', __('adminlte::adminlte.login_message')) {{-- Formulir login --}} @section('auth_body') {{-- Menampilkan pesan error jika login gagal --}} @error('login_gagal') <div class="alert alert-warning alert-dismissible fade show" role="alert"> Warning! {{ \$message }} <button type="button" class="close" data-dismiss="alert" aria-label="Close"> &times; </button> </div> @enderror <div class="card"> <div class="card-body"> <h1>Login</h1> <div class="form-group"> <input type="text" class="form-control" placeholder="Email" /> </div> <div class="form-group"> <input type="password" class="form-control" placeholder="Password" /> </div> <div class="form-group"> <input type="checkbox" class="checkbox" /> Remember me </div> <div class="form-group"> <button type="button" class="btn btn-primary" value="Login">Login</button> </div> </div> </div></pre>



Jurusan Teknologi Informasi Politeknik Negeri Malang
Pemrograman Web Lanjut
Mata Kuliah Pemrograman Web Lanjut
Pengampu: Tim Ajar Pemrograman Web Lanjut
April 2024

```
{{-- Formulir login --}}
<form action="{{ url('proses_login') }}" method="post">
    @csrf

    {{-- Field username --}}
    <div class="input-group mb-3">
        <input type="text" name="username" class="form-control @error('username') is-invalid @enderror" value="{{ old('username') }}"
            placeholder="Username" autofocus>

        <div class="input-group-append">
            <div class="input-group-text">
                <span class="fas fa-envelope {{ config('adminlte.classes_auth_icon', '') }}"></span>
            </div>
        </div>

        @error('username')
        <span class="invalid-feedback" role="alert">
            <strong>{{ $message }}</strong>
        </span>
        @enderror
    </div>

    {{-- Field password --}}
    <div class="input-group mb-3">
        <input type="password" name="password" class="form-control @error('password') is-invalid @enderror" placeholder="{{ __
            ('adminlte::adminlte.password') }}">

        <div class="input-group-append">
            <div class="input-group-text">
                <span class="fas fa-lock {{ config('adminlte.classes_auth_icon', '') }}"></span>
            </div>
        </div>

        @error('password')
        <span class="invalid-feedback" role="alert">
            <strong>{{ $message }}</strong>
        </span>
        @enderror
    </div>

    {{-- Field remember me --}}
    <div class="row">
        <div class="col-7">
            <div class="icheck-primary" title="{{ __('adminlte::adminlte.remember_me_hint') }}">
                <input type="checkbox" name="remember" id="remember" {{ old('remember') ? 'checked' : '' }}>

                <label for="remember">
                    {{ __('adminlte::adminlte.remember_me') }}
                </label>
            </div>
        </div>

        {{-- Tombol login --}}
        <div class="col-5">
            <button type="submit" class="btn btn-block {{ config('adminlte.classes_auth_btn', 'btn-flat btn-primary') }}">
                <span class="fas fa-sign-in-alt"></span>
                {{ __('adminlte::adminlte.sign_in') }}
            </button>
        </div>
    </div>
</form>
@stop

{{-- Footer untuk halaman login --}}
@section('auth_footer')
    {{-- Menampilkan link untuk registrasi jika diaktifkan --}}
    @if ($register_url)
    <p class="my-0">
        <a href="{{ route('register') }}">
            {{ __('adminlte::adminlte.register_a_new_membership') }}
        </a>
    </p>
    @endif
@stop
```

kode tersebut adalah tampilan dari login yang berupa html code



Jurusan Teknologi Informasi Politeknik Negeri Malang
Pemrograman Web Lanjut
Mata Kuliah Pemrograman Web Lanjut
Pengampu: Tim Ajar Pemrograman Web Lanjut
April 2024

14. Buatlah halaman register dengan membuka folder resources/views dengan nama file **register.blade.php**. Lalu kita isi filenya seperti contoh dibawah

```
@extends('adminlte::auth.auth-page', ['auth_type' => 'register'])

@php( $login_url = View::getSection('login_url') ?? config('adminlte.login_url', 'login'))
@php( $register_url = View::getSection('register_url') ?? config('adminlte.register_url', 'register'))

@if (config('adminlte.use_route_url', false))
@php( $login_url = $login_url ? route($login_url) : '' )
@php( $register_url = $register_url ? route($register_url) : '' )
@else
@php( $login_url = $login_url ? url($login_url) : '' )
@php( $url = $register_url ? url($register_url) : '' )
@endif
@section('auth_header', __('adminlte::adminlte.register_message'))
@section('auth_body')
<form action="{{url($register_url)}}" method="post">
    @csrf
    {{-- Nama Field --}}
    <div class="input-group mb-3">
        <input type="text" name="nama" class="form-control @error('nama') is-invalid @enderror" value="{{ old('nama') }}"
        placeholder="Nama" autofocus>
        <div class="input-group-append">
            <div class="input-group-text">
                <span class="fas fa-user {{ config('adminlte.classes_auth_icon', '') }}"></span>
            </div>
        </div>
        @error('nama')
        <span class="invalid-feedback" role="alert">
            <strong>{{ $message }}</strong>
        </span>
        @enderror
    </div>
    {{-- Username Field --}}
    <div class="input-group mb-3">
        <input type="text" name="username" class="form-control @error('username') is-invalid @enderror" value="{{ old('username') }}"
        placeholder="Username" autofocus>
        <div class="input-group-append">
            <div class="input-group-text">
                <span class="fas fa-user {{ config('adminlte.classes_auth_icon', '') }}"></span>
            </div>
        </div>
        @error('username')
        <span class="invalid-feedback" role="alert">
            <strong>{{ $message }}</strong>
        </span>
        @enderror
    </div>
</form>
```



Jurusan Teknologi Informasi Politeknik Negeri Malang
Pemrograman Web Lanjut
Mata Kuliah Pemrograman Web Lanjut
Pengampu: Tim Ajar Pemrograman Web Lanjut
April 2024

	<pre> {{-- Password Field --}} <div class="input-group mb-3"> <input type="password" name="password" class="form-control @error('password') is-invalid @enderror" placeholder="{{__ ('adminlte::adminlte.password')}}"> <div class="input-group-append"> <div class="input-group-text"> </div> </div> @error('password') {{ \$message }} @enderror </div> {{-- Register Button --}} <button type="submit" class="btn btn-block {{config('adminlte.classes_auth_btn', 'btn-flat btn-primary')}}"> {{__('adminlte::adminlte.register')}} </button> </form> @stop @section('auth_footer') <p class="my-0"> {{__('adminlte::adminlte.i_already_have_a_membership')}} </p> </div> </pre> <p>kode tersebut adalah tampilan dari halaman register yang berbentuk html code</p>
15.	Buatlah halaman register dengan membuka folder resources/views dengan nama file admin.blade.php . Lalu kita isi filenya seperti contoh dibawah



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

{{-- Customize layout sections --}}

@section('subtitle','Admin')
@section('content_header_title','Home')
@section('content_header_subtitle','Admin')

@section('content')
<div class="container">
    <div class="card">
        <div class="card-header">Tampilan
            @php
                echo (Auth::user()->level_id == 1) ? 'Admin' : 'Manager';
            @endphp
        <div class="card-body">
            <h1>Login Sebagai:
                @php
                    echo (Auth::user()->level_id == 1) ? 'Admin' : 'Manager';
                @endphp
            </h1>
            <a href="{{ route('logout') }}">Logout</a>
        </div>
    </div>
</div>
</div>
@endsection

@push('js')
@endpush
```

kode tersebut adalah tampilan dari halaman admin

16. Buatlah halaman register dengan membuka folder resources/views dengan nama file **manager.blade.php**. Lalu kita isi filenya seperti contoh dibawah



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

{{-- Customize layout sections --}}

@section('subtitle', 'Manager')
@section('content_header_title', 'Home')
@section('content_header_subtitle', 'Manager')

@section('content')
<div class="container">
    <div class="card">
        <div class="card-header">Tampilan
            @php
                echo (Auth::user()→level_id == 1) ? 'Admin' : 'Manager';
            @endphp
        <div class="card-body">
            <h1>Login Sebagai:
                @php
                    echo (Auth::user()→level_id == 1) ? 'Admin' : 'Manager';
                @endphp
            </h1>
            <a href="{{ route('logout') }}">Logout</a>
        </div>
    </div>
</div>
</div>
@endsection

@push('js')
@endpush
```

kode tersebut adalah tampilan dari halaman manager

17. Jalankan kode diatas dan buat laporan dengan isi screenshoot hasil dan beri penjelasan pada setiap screenshoot hasil dan *commit* perubahan pada *git*.



Jurusan Teknologi Informasi Politeknik Negeri Malang
Pemrograman Web Lanjut
Mata Kuliah Pemrograman Web Lanjut
Pengampu: Tim Ajar Pemrograman Web Lanjut
April 2024

tampilan login

disebelah kiri jika salah memasukkan username dan password, dan jika benar maka akan ke masuk ketampilan login

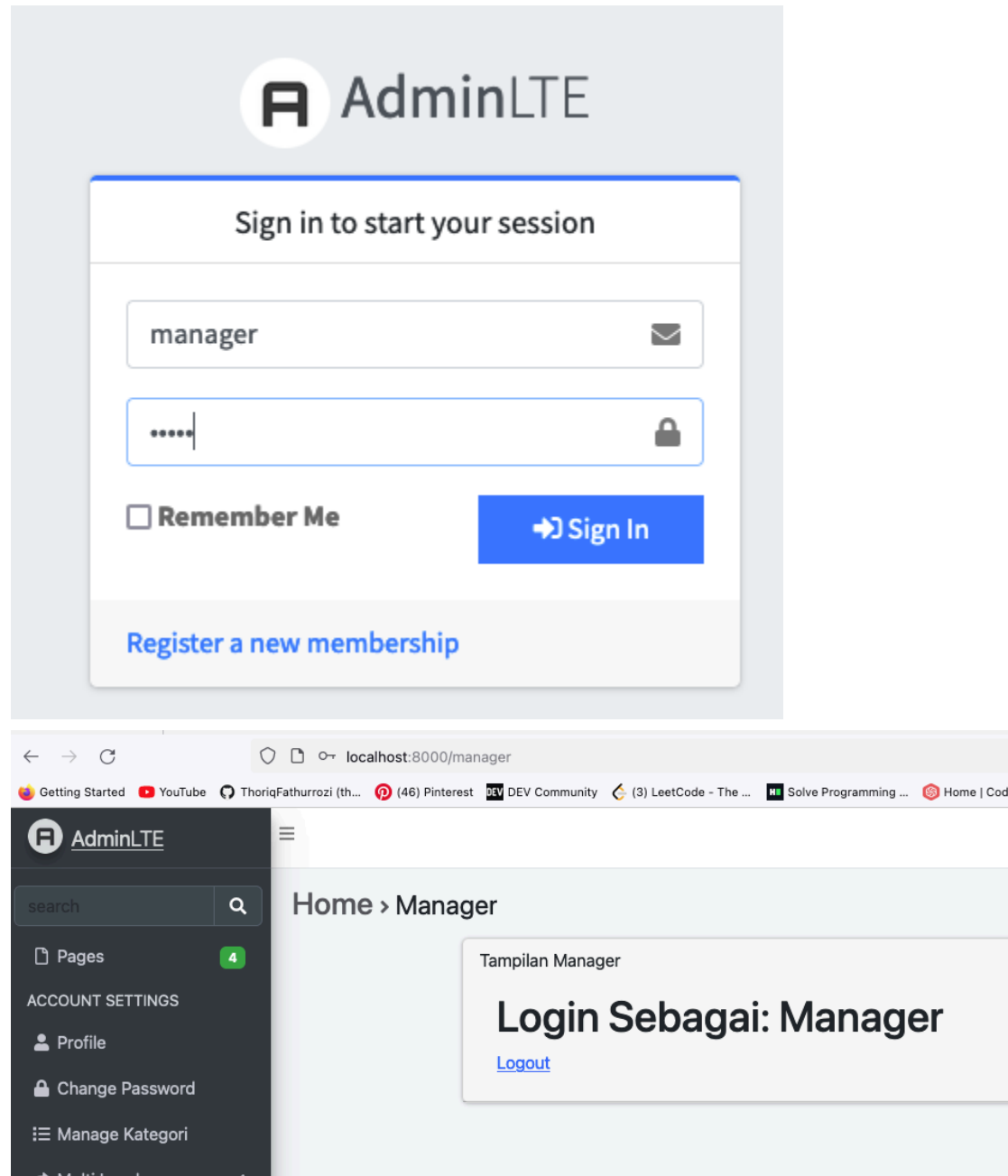
ini saya login menggunakan credential admin

tampilan admin

login sebagai manager



Jurusan Teknologi Informasi Politeknik Negeri Malang
Pemrograman Web Lanjut
Mata Kuliah Pemrograman Web Lanjut
Pengampu: Tim Ajar Pemrograman Web Lanjut
April 2024

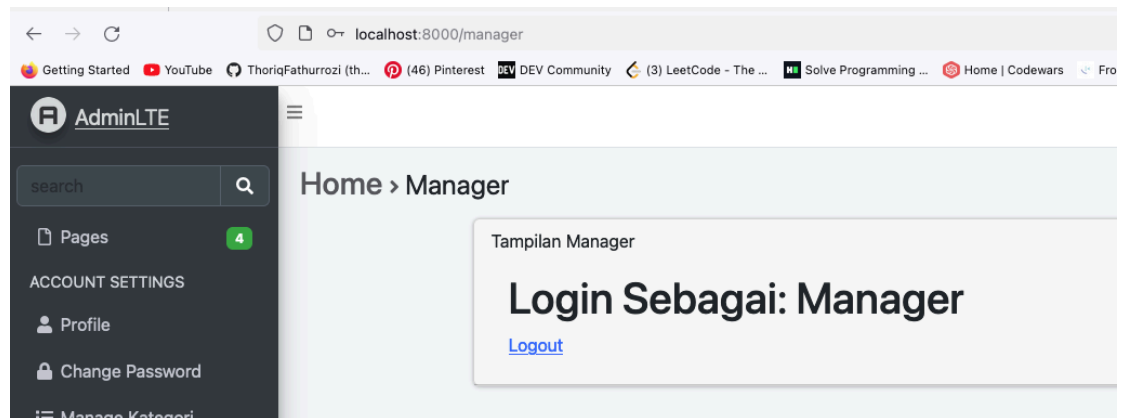


ini halaman ketika login menggunakan credential manager



Jurusan Teknologi Informasi Politeknik Negeri Malang
Pemrograman Web Lanjut
Mata Kuliah Pemrograman Web Lanjut
Pengampu: Tim Ajar Pemrograman Web Lanjut
April 2024

register user thoriq



ketika berhasil register dan login maka akan di redirect ke halaman manager