

LAMPIRAN

Listing Program Register

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

    if ($validator->fails()) {
        return response()->error(status: 422,
message: $validator->errors());
    }

    $user = User::create([
        'name' => $request->input('name'),
        'username' => $request-
>input('username'),
        'email' => $request->input('email'),
        'password' => bcrypt($request-
>input('password')),
    ]);

    if (!$user) {
        return response()->error(status: 500,
message: 'Failed to register user. ');
    }
    $token = Auth::login($user);
    $token = JWTAuth::fromUser($user);
    $token = Crypt::encrypt($token);

    return response()->success(['user' => $user,
'token' => $token], message: 'User registered
successfully. ');
}
```

Listing Program Login

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

    if ($validator->fails()) {
        return response()->error(status: 422,
message: $validator->errors());
    }

    $credentials = [
        'password' => $request-
>input('password'),
    ];

    $usernameOrEmail = $request-
>input('username_or_email');

    if (filter_var($usernameOrEmail,
FILTER_VALIDATE_EMAIL)) {
        $credentials['email'] =
$usernameOrEmail;
    } else {
        $credentials['username'] =
$usernameOrEmail;
    }

    if (!Auth::attempt($credentials)) {
        return response()->error(status: 401,
message: 'Kata sandi atau nama pengguna salah.');
```

Listing Program Middleware dan Route API

```

// JwtMiddleware.php
public function handle(Request $request, Closure
$next): Response
{
    try {
        // Mengambil token dari header Authorization
        $token = str_replace('Bearer ', '', $request-
>header('Authorization'));
        // Dekripsi token menggunakan Crypt
        $token = Crypt::decrypt($token);

        // Menetapkan token untuk digunakan oleh
JWTAuth
        JWTAuth::setToken($token);

        // Mendapatkan pengguna terkait dengan token
        JWTAuth::toUser();
    } catch (Exception $e) {
        // Menangani berbagai jenis exception yang
mungkin terjadi
        if ($e instanceof TokenInvalidException) {
            // Respons jika token tidak valid
            return response()->error(null, 401, 'Token
is Invalid');
        } else if ($e instanceof TokenExpiredException)
{
            // Mengambil token yang baru jika token
sudah kedaluwarsa
            $newToken =
JWTAuth::refresh(JWTAuth::getToken());

            // Melanjutkan request dengan menyertakan
header Authorization yang baru
            return $next($request)-
>withHeaders(['Authorization' => 'Bearer ' .
$newToken]);
        } else {
            // Respons jika terjadi kesalahan lainnya
            return response()->error(null, 401, 'Token
is Invalid');
        }
    }
    // Melanjutkan request jika tidak ada masalah
dengan token
    return $next($request);
}

```

Uji Register

The screenshot shows a REST client interface for a POST request to `http://opaapi.test/api/v1/register`. The request body is form-urlencoded with the following data:

Key	Value
name	Muhammad Ilham
username	ilhaoeba
email	ilhao@gmail.com
password	Password1234!

The response is a JSON object indicating a successful registration:

```

{
  "error": false,
  "errors": [],
  "message": "User registered successfully.",
  "data": {
    "user": {
      "name": "Muhammad Ilham",
      "username": "ilhaoeba",
      "email": "ilhao@gmail.com",
      "updated_at": "2023-12-14T07:01:11.000000Z",
      "created_at": "2023-12-14T07:01:11.000000Z",
      "id": 4
    }
  },
  "token": "eyJodDI6IjE4OTVubjB3OHJjVURlM0N4NWkzT2c2P5SisInZhbHVlIjo1VWl0bnVkdjU4K1N6S3VtD91TdVRCc25sS2ZGYzNiMlIdE01AzM0SGdko4Ow1S3ZyOFlhZVnRolL1NoNWhwVChRckxvS8n2lVWjps2hadUZKTU9NMzYzK2ZlMlJlNHNkOFB1OWNlc2E5M1hpUkxmkMkYnJ2Z3ZGU1kvcko2VWdkWd1T3qjM3Fsd2I0k29wY8s3RfPeUqkgrYVg2M2cHskd1MjUvR1h2eUVRL9MvVdTWg42UGvW61HcnR4KzVUSHR28VpVn3Uc1J3TnpWUtK0m1PeUqkgrYVg2M2c3Qz2oTVRJUH3bFc3e1ZybnFod8gvdK4xcHdS43BMSwtubVez2TJ3T21DT26RbGR1N6N6Z216Y1RMTFJVFajTn90NjRU8mk0N8NHwC4R0tzc3JkdF01bmR2aHc4Rw1ZU3ZiZ890StxaG1mZ3F3Zf8ubnhKYXovbVh20E10TGWkHhMEJ3cKsoQwrd1oGZVU1Q641NEVsc0V4Sn0LbwZkQ9KdFoyVkg5V8VhZ2U8N2t1UHbndZF03h10wVodjVaUwJ1NzVOR69jM1hqNgpITGv3c2szUk9YMXh4STVSCiIsIm1hy161jVikMwQ1NDJ0TFHNGQ1NjYxMTJlMmU5MTM5MGM0NTBkN2QxNtczZjgMzU4MzcxMwFkYmNlMjg5VTI1Y2M1MTEiLCB0YXciOiIiIiF0=="
}

```

Uji Login

The screenshot shows a REST client interface for a POST request to `http://opaapi.test/api/v1/login`. The request body is form-data with the following data:

Key	Type	Value
username_or_email	Text	ilhaoeba
password	Text	Password1234!

The response is a JSON object indicating a successful login:

```

{
  "message": "sukses",
  "data": {
    "user": {
      "id": 4,
      "name": "Muhammad Ilham",
      "email": "ilhao@gmail.com",
      "email_verified_at": null,
      "remember_token": null,
      "created_at": "2023-12-14T07:01:11.000000Z",
      "updated_at": "2023-12-14T07:01:11.000000Z",
      "username": "ilhaoeba"
    }
  },
  "token": "eyJodDI6IjE4OTVubjB3OHJjVURlM0N4NWkzT2c2P5SisInZhbHVlIjo1M3gwZWJiRUZiZ0E0bWlvaVFCd2R2OVJhZEpMaU5Mc0IyeTRyY0V3REdHNGwUdRlSNvOmw8QzycEteYlN4jdrZVNUdGhZ2R0QW4vNVZjRjFvb6NlWj1jw6110GZkQ9S9JmZwE12V1ZV53Z0Z6V8NlVwG1sT1h3ZThxNjYyZGtFNM1JVF3FUDJXenZUckdQVnY2M1VzeGyxR1Vcehka3pQDStac1NjRGl2Xkx0VGVNTRFRpakZetWlodUoze1BlV2nNtXESbzFHTeJjduRHhW4zWLRDhNnMUJJaQTRKRWQW01ZD2pcFoRtZHz2BxHFlV1Jje82d21Za21u1NLUk1NOVY4R6kv20Mxc8NYUNj3a22POXVU1FyNS91cHrWDERX5s1hYTRocXVDMhEB3pabVdzYkF4e16VHRQsDFORjd9Yy83WfZtYy4YkgvSTRtanBy2lVsV3wodDvbk1wTm11SHYvRzNYTXUvRvHSTRzK2RvN2xJQwVCNEtxdzHY2w1b1FFG1ERVNCN1Abw9ERkx1Mn1b11bEsxcXRzTWO0UVImIi02piS9dX0wR4b1V4MSIstM1hYy161jVikMwQ1NDJ0TFHNGQ1NjYxMTJlMmU5MTM5MGM0NTBkN2QxNtczZjgMzU4MzcxMwFkYmNlMjg5VTI1Y2M1MTEiLCB0YXciOiIiIiF0=="
}

```

Uji Akses Endpoint dengan Token JWT

http://opaapi.test/api/v1/users/4

GET http://opaapi.test/api/v1/users/4

Authorization

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization

Token

```
eyJpdll6lUkKEM5ibXlzTGFKZmFNWXZJTEQyMFE9PSlslZhbHVlIjoIM3gwZWJlRUZ1cEpBbWlvaFcd2RZOVJhZEpMeU5McW1yeTRYeDV3REdHNGwwUTRLSnVWw0QzZycEtEYlN4NjdrZVNUdGNHZA0RQWw4vNVZjRjFVb0NiWjJlWGI1OGZkQs9JMEZwEi2VIZVXJOZ0V0NWwVWGlslThJZThxNjVYZ0tPNW1JVFJFUDJXemZUckdQVmY2MlVreGYxR1Vlcmhka3pQOStac1NIRG1oZWxpVGwVWFRRPakZ1TWlodUoZelBlV2pNTXE5bzFHTeJjdURHNW4zWIRDWnNmUUJaQTRKRWVQWDU1ZDZpcFdrTzZHZzBXaHFYlJjem02d21ZazLU1NLUkNOVY4RGkvZ0Mxc0NyTUNja2ZPOXVUT1FyWS9icHVDWERXSmhYTRocXVdWnBeb3pabVdzYkF4eEi6VHRqSDFORjd0Yy83WFZlYys4YkxSTRtanByZlV3shwdDVxbk1wTmlSHYvRzNYTXUvRmVhSTRzK2RVN2xJQmVCNEtdzdXy2w1b1FFNGIERVNCNIawBw9ERkx1Mn1b1J1bEsxcXRrTWoOUVplMitOZ1pIS0dXOWR4b1V4MSlslm1hYi6ljk1MTRlMDZlMDQ4MDE4NzA3MTVhMTMwMWFmYzZmZjZlVjZlU1ZmVlNmMzY2EwOTQwYWFjMGQ2NDQ4Mzk0Y2MlLj0YWciOllifQ==
```

Body

```
1 {
2   "code": 200,
3   "error": false,
4   "errors": [],
5   "message": "sukses",
6   "data": {
7     "id": 4,
8     "name": "Muhammad Ilham",
9     "email": "ilham@gmail.com",
10    "email_verified_at": null,
11    "remember_token": null,
12    "created_at": "2023-12-14T07:01:11.000000Z",
13    "updated_at": "2023-12-14T07:01:11.000000Z",
14    "username": "ilhamoeba"
15  }
16 }
```

200 OK

http://opaapi.test/api/v1/profile/bio

POST http://opaapi.test/api/v1/profile/bio

Body

```
1 {
2   "id_bio": 4,
3   "gender": "m",
4   "birth_place": "sleman",
5   "birth_date": "2001-01-06",
6   "bio_country_id": 100,
7   "bio_province_id": 34,
8   "bio_city_id": 3404,
9   "creation_id": 0,
10  "modified_id": 0,
11  "full_name": "Muhammad Ilham Pratama",
12  "phone": "+6285157769433"
13 }
```

Body

```
1 {
2   "code": 201,
3   "error": false,
4   "errors": [],
5   "message": "sukses",
6   "data": {
7     "id_bio": 4,
8     "gender": "m",
9     "birth_place": "sleman",
10    "birth_date": "2001-01-06",
11    "bio_province_id": 34,
12    "bio_city_id": 3404,
13    "bio_country_id": 100,
14    "creation_id": 0,
15    "modified_id": 0,
16    "full_name": "Muhammad Ilham Pratama",
17    "phone": "+6285157769433",
18    "modified_date": "2023-12-14T08:12:03.000000Z",
19    "creation_date": "2023-12-14T08:12:03.000000Z"
20  }
21 }
```

201 Created

Ketentuan Kelulusan

PEMBERITAHUAN SEBELUM UJIAN:

Pengumpulan akhir dokumen Tugas Akhir/Skripsi melewati Jum'at, 19 Januari 2024, mahasiswa harus menyelesaikan registrasi dan KRS semester berikutnya."

KRITERIA KELULUSAN UJIAN SIDANG / PENDADARAN

1. Lulus ujian tanpa syarat, disebut kriteria 1.
2. Lulus bersyarat, disebut kriteria 2, yaitu dengan sedikit perbaikan atau penyempurnaan text dan atau program dalam waktu maks sampai tanggal 18 Maret 2024 dan tidak ada ujian lagi. Jika dalam waktu yang ditentukan mahasiswa tersebut tidak dapat menyelesaikan, maka, mahasiswa yang bersangkutan dianggap tidak lulus ujian.
3. Tidak lulus ujian sidang/pendadaran, disebut kriteria 3, dijelaskan, disarankan Ketua Tim Penguji untuk mempelajari ulang materi, merombak program/teks, atau mengganti judul.

Ketentuan bagi peserta yang tidak lulus ujian sidang / pendadaran:

1. Mahasiswa wajib menempuh ujian sidang/pendadaran ulang"
2. Kesempatan ujian sidang/pendadaran ulang hanya diberikan dalam rentang waktu maksimum 6 bulan, setelah ujian sidang/pendadaran
3. Jika sampai batas waktu maksimum 6 bulan tersebut belum dapat diajukan/diselesaikan, maka calon peserta ujian dinyatakan sebagai mahasiswa peserta Skripsi/TGA baru, dengan segala ketentuan yang berlaku bagi peserta baru.
4. Mahasiswa yang akan menempuh ujian sidang/pendadaran ulang ini diwajibkan membayar biaya ujian setara 2 SKS praktik, sesuai tahun Angkatan.

Yogyakarta, 19 Januari 2024

Memahami dan bersedia mematuhi peraturan di atas,



Muhammad Ilham Pratama

KEPUTUSAN HASIL UJIAN PENDADARAN

Sesuai dengan hasil sidang pendadaran pada tanggal 19 Januari 2024 maka:

Nama Mahasiswa : Muhammad Ilham Pratama

NIM / Program Studi : 205410182 / Informatika

Jenjang : S1

dinyatakan LULUS dengan kriteria 2

Ketua Penguji : Badiyanto, S.Kom., M.Kom.

CATATAN

Hari, tanggal	Jumat, 19 Januari 2024	
Waktu	13.00	
Nama	Muhammad Ilham Pratama	
No. Mahasiswa / Jurusan	205410182 / Informatika	
Hal yang harus diperbaiki		Pemberi Catatan
1. setiap pengujian, URL ditunjukkan di laravelnya seperti apa (sisi client dan server) kemudian dijelaskan 2. cara memanggil di server untuk memanggil end point, dijelaskan (routing server)		Pak badi
1. tabel tinjauan pustaka direvisi, bukan kolom judul, studi kasus --> ganti kolom objek, metode, hasil 2. catatan perbaikan ada di naskah		Pak badi

SURAT KETERANGAN
PERSETUJUAN PUBLIKASI

Bahwa yang bertanda tangan di bawah ini:

Nama : Muhammad Ilham Pratama
No. Mahasiswa : 205410182
Jurusan : Informatika
Jenjang : S1
Judul : IMPLEMENTASI JSON WEB TOKEN (JWT) DALAM
PENGEMBANGAN PERSONAL ASSESSMENT
WEBSITE DENGAN FRAMEWORK LARAVEL
(MAGANG BERSERTIFIKAT KAMPUS MERDEKA)
STUDI KASUS: PT. ENGINEERING CAREER CENTER
(ECC)

menyerahkan karya ilmiah kepada pihak perpustakaan UNIVERSITAS
TEKNOLOGI DIGITAL INDONESIA dan menyetujui untuk diunggah ke
DigitalLibrary UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA sesuai
dengan ketentuan yang berlaku untuk kepentingan riset dan pendidikan.

Yogyakarta, 1 Februari 2024

Penulis, ..



Muhammad Ilham Pratama

NIM: 205410182