

	MODUL PRAKTIKUM	JTI
	PEMROGRAMAN WEB LANJUT	
	Authentication RESTFUL API	Hal 1 of 5

TUJUAN PRAKTIKUM

1. Mahasiswa mampu memahami konsep Authentication RESTful API
2. Mahasiswa mampu menerapkan Authentication RESTful API di framework
3. Mahasiswa mampu menguji Authentication RESTful API di framework

TEORI SINGKAT

Authentication RESTful API pada framework Laravel dapat menggunakan package Sanctum. Laravel sanctum menyediakan featherweight authentication system untuk Single Page Application (SPA), mobile application dan API berbasis token yang sederhana. Sanctum memungkinkan setiap pengguna aplikasi menghasilkan beberapa token API untuk akun mereka. Token ini dapat diberikan kemampuan atau cakupan yang menentukan tindakan mana yang diizinkan untuk dilakukan oleh token.

Keamanan RESTful API dapat menggunakan middleware Laravel. middleware tergolong sebuah software layer. Sesuai namanya yaitu middle yang artinya tengah, maka letaknya berada di antara controller dan router. Ada juga yang mendefinisikan middleware sebagai software penengah antara satu aplikasi dengan aplikasi lain, sehingga proses integrasinya berjalan lebih mudah

PRAKTIKUM

Pada praktikum kali ini, menggunakan project sebelumnya yaitu RESTful API dengan beberapa modifikasi sebagai berikut:

INSTALL SANCTUM

```
composer require laravel/sanctum
```

KONFIGURASI SANCTUM

```
php artisan vendor:publish --provider="Laravel\Sanctum\SanctumServiceProvider"
```

MIGRATION DATABASE

```
php artisan migrate
```

MEMBUAT CONTROLLER

```
php artisan make:controller API/AuthController
```



```
<?php
```

```
namespace App\Http\Controllers\API;
```

```
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Validator;
use App\Models\User;
```

```
class AuthController extends Controller
{
```

```
public function register(Request $request)
{
    $validator = Validator::make($request->all(), [
        'name' => 'required|string|max:255',
        'email' => 'required|string|email|max:255|unique:users',
        'password' => 'required|string|min:8'
    ]);

    if($validator->fails()){
        return response()->json($validator->errors());
    }

    $user = User::create([
        'name' => $request->name,
        'email' => $request->email,
        'password' => Hash::make($request->password)
    ]);

    $token = $user->createToken('auth_token')->plainTextToken;

    return response()
        ->json(['data' => $user, 'access_token' => $token, 'token_type' => 'Bearer', ]);
}
```

```
public function login(Request $request)
{
    if (!Auth::attempt($request->only('email', 'password')))
    {
        return response()
            ->json(['message' => 'User tidak terdaftar'], 401);
    }

    $user = User::where('email', $request['email'])->firstOrFail();

    $token = $user->createToken('auth_token')->plainTextToken;

    return response()
        ->json(['message' => 'Berhasil, Selamat datang '.$user->name.' ', 'access_token' => $token,
    }
}
```

	MODUL PRAKTIKUM	JTI Hal 3 of 5
	PEMROGRAMAN WEB LANJUT	
	Authentication RESTFUL API	

```

public function logout()
{
    auth()->user()->tokens->delete();

    return [
        'message' => 'Terima kasih'
    ];
}
}

```

MENDEFINISIKAN ROUTE

```

<?php

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\MahasiswaController;
use App\Http\Controllers\API\AuthController;
Route::post('/register', [AuthController::class, 'register']);
Route::post('/login', [AuthController::class, 'login']);
Route::group(['middleware' => ['auth:sanctum']], function () {

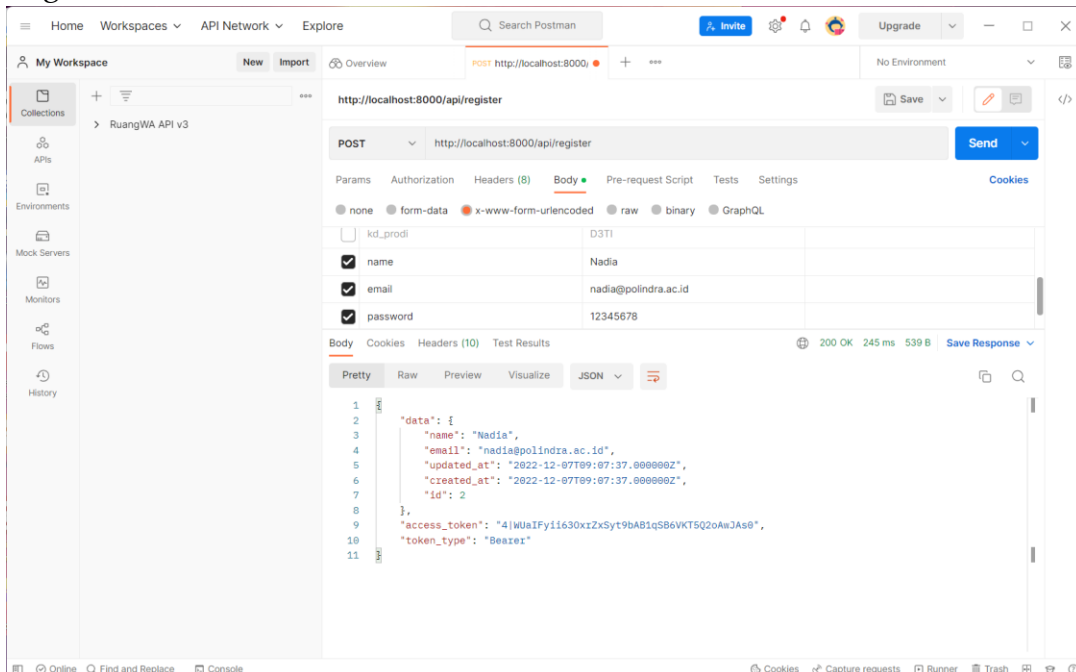
    Route::get('/mahasiswa', [MahasiswaController::class, 'index']);
    Route::get('/mahasiswa/show/{id}', [MahasiswaController::class, 'show']);
    Route::delete('/mahasiswa/destroy/{id}', [MahasiswaController::class, 'destroy']);
    Route::post('/mahasiswa', [MahasiswaController::class, 'store']);
    Route::put('/mahasiswa/update/{id}', [MahasiswaController::class, 'update']);
    Route::post('/logout', [AuthController::class, 'logout']);

});

```

TESTING API

a. Register



The screenshot shows a Postman interface with a POST request to `http://localhost:8000/api/register`. The request body is in the `x-www-form-urlencoded` format with the following data:

Key	Value
<code>kd_prodi</code>	<code>D3TI</code>
<code>name</code>	<code>Nadia</code>
<code>email</code>	<code>nadia@polindra.ac.id</code>
<code>password</code>	<code>12345678</code>

The response is a JSON object with the following structure:

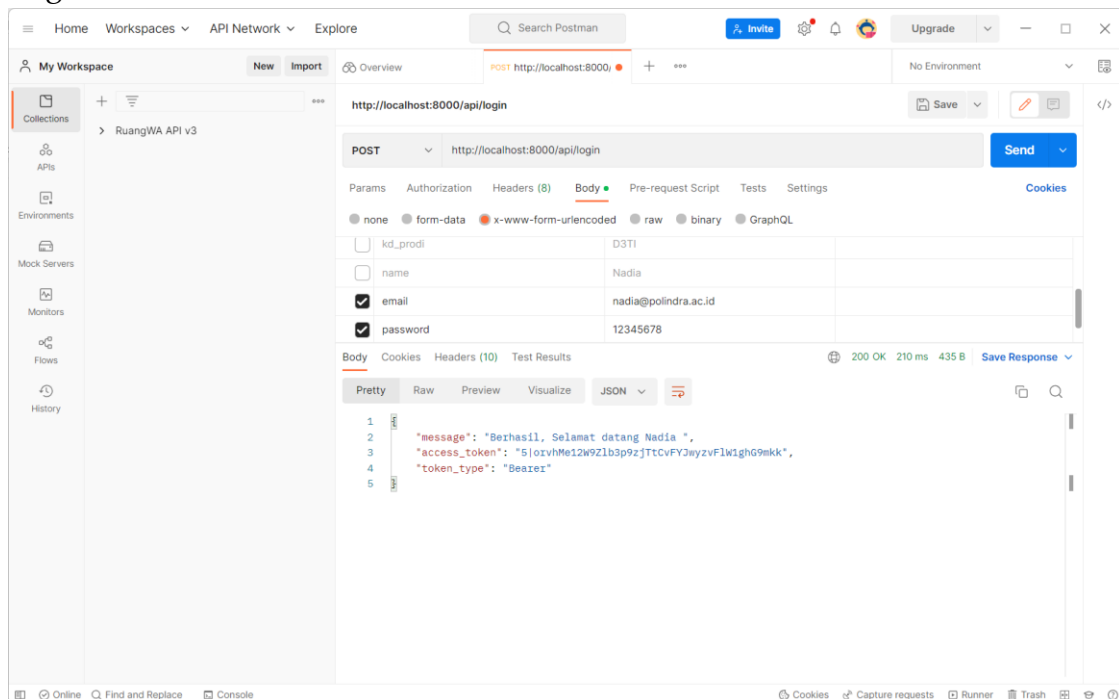
```

{
  "data": {
    "name": "Nadia",
    "email": "nadia@polindra.ac.id",
    "updated_at": "2022-12-07T09:07:37.000000Z",
    "created_at": "2022-12-07T09:07:37.000000Z",
    "id": 2
  },
  "access_token": "41WUaIfyi630xzSyt9bAB1q5B6VKT5Q2oAw3As0",
  "token_type": "Bearer"
}

```

	MODUL PRAKTIKUM	JTI Hal 4 of 5
	PEMROGRAMAN WEB LANJUT	
	Authentication RESTFUL API	

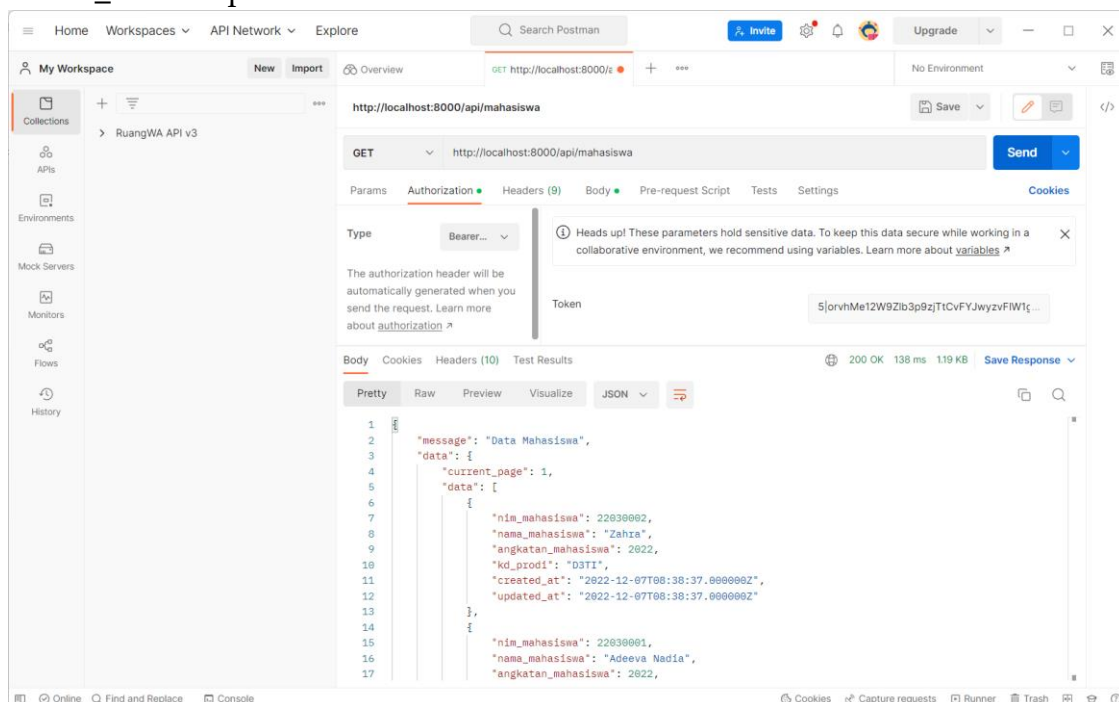
b. Login



access_token hasil login dapat digunakan untuk proses *authentication* ke data mahasiswa. Sehingga silahkan catat untuk dilakukan proses selanjutnya.

c. Akses Mahasiswa

Pilih tab Authorization, kemudian pilih type Bearer Token dan Masukkan **access_token** seperti berikut:



	MODUL PRAKTIKUM	JTI
	PEMROGRAMAN WEB LANJUT	
	Authentication RESTFUL API	Hal 5 of 5

TUGAS

1. Buatlah sebuah middleware RESTFul tabel prodi untuk melakukan proses tampil, tambah data, perubahan data, dan hapus.
2. Lakukan pengujiannya menggunakan postman

Catatan:

- Langkah-langkah pembuatannya dapat mengikuti praktikum
- Laporan praktikum tugas dan sourcecode dikumpulkan ke elearning