

**Laporan Praktikum**  
**Mata Kuliah Pemrograman Web**



**"JWT (JSON WEB TOKEN)"**

Dosen Pengampu :  
Willdan Aprizal Arifin S.pd M.kom

Disusun Oleh :  
RIYADI TRI WALUYA BUDI  
2308065

**PROGRAM STUDI SISTEM INFORMASI KELAUTAN**  
**UNIVERSITAS PENDIDIKAN INDONESIA**  
**2024**

## **I.PENDAHULUAN**

Standar terbuka (RFC 7519) JSON Web Token (JWT) digunakan untuk mentransfer informasi secara aman sebagai objek JSON yang ditandatangani antara dua entitas. JWT sering digunakan untuk otorisasi dan autentikasi di aplikasi web kontemporer. Teknologi ini menawarkan cara mudah dan efektif untuk menjaga keamanan dan integritas data selama proses komunikasi.

Tujuan Penggunaan JWT dalam Pemrograman Web:

1. Autentikasi: memastikan bahwa pengguna yang mengakses sistem adalah pengguna yang sah.
2. Otorisasi: memberikan akses terbatas sesuai dengan peran atau hak pengguna yang telah diautentikasi.
3. Keamanan Data: Informasi dalam JWT ditandatangani menggunakan algoritma hashing seperti HMAC atau RSA, sehingga tidak dapat dimodifikasi tanpa deteksi.
4. Efisiensi: JWT berbentuk self-contained, sehingga dapat memuat informasi penting tanpa mengganggu.

## **II.ALAT DAN BAHAN**

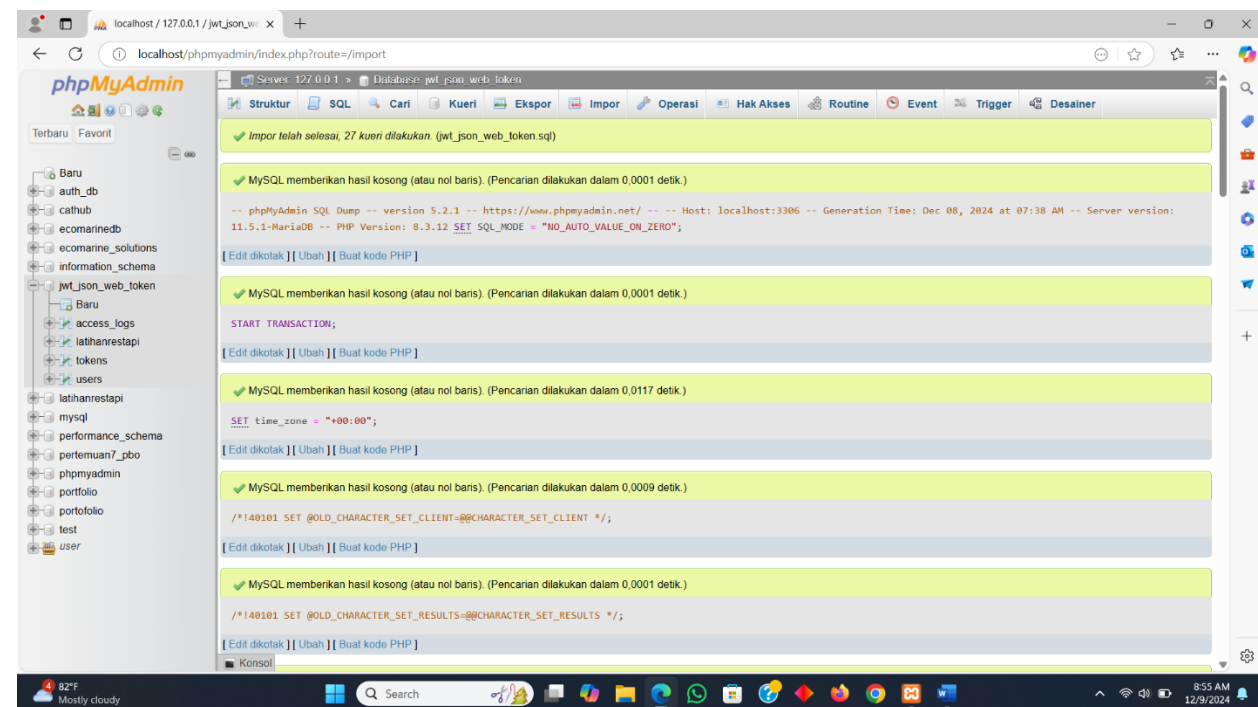
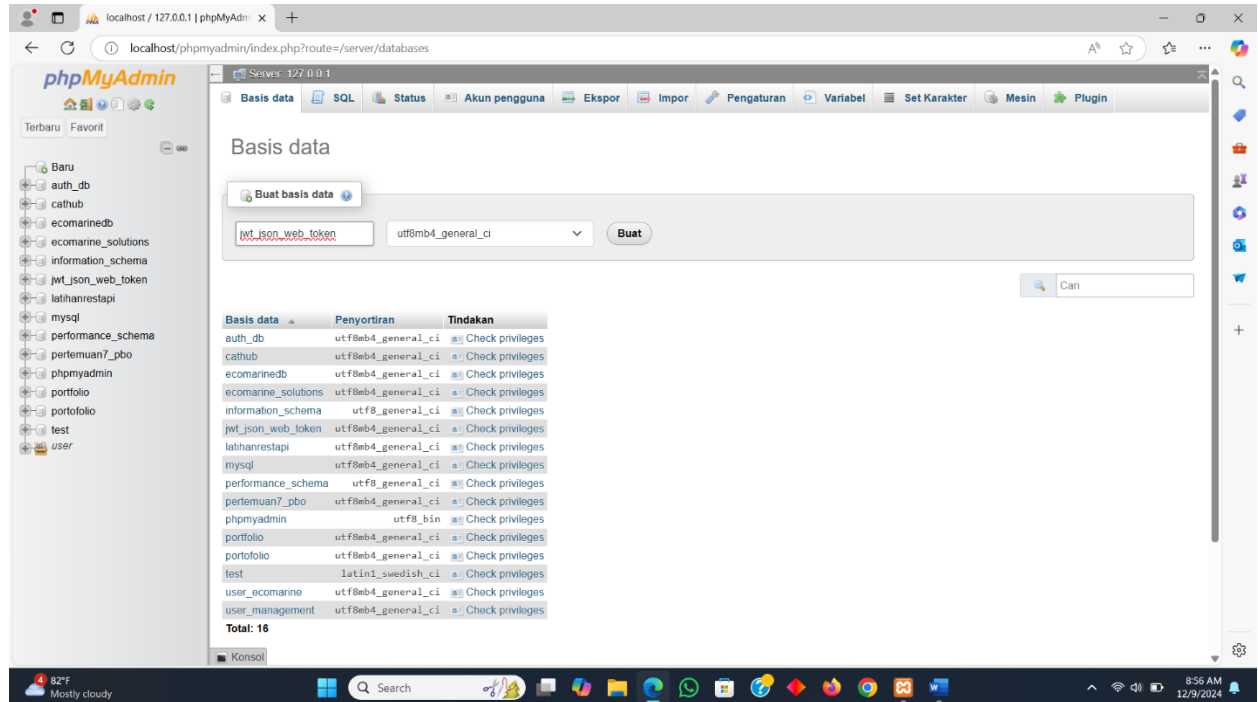
-Xampp

-VsCode

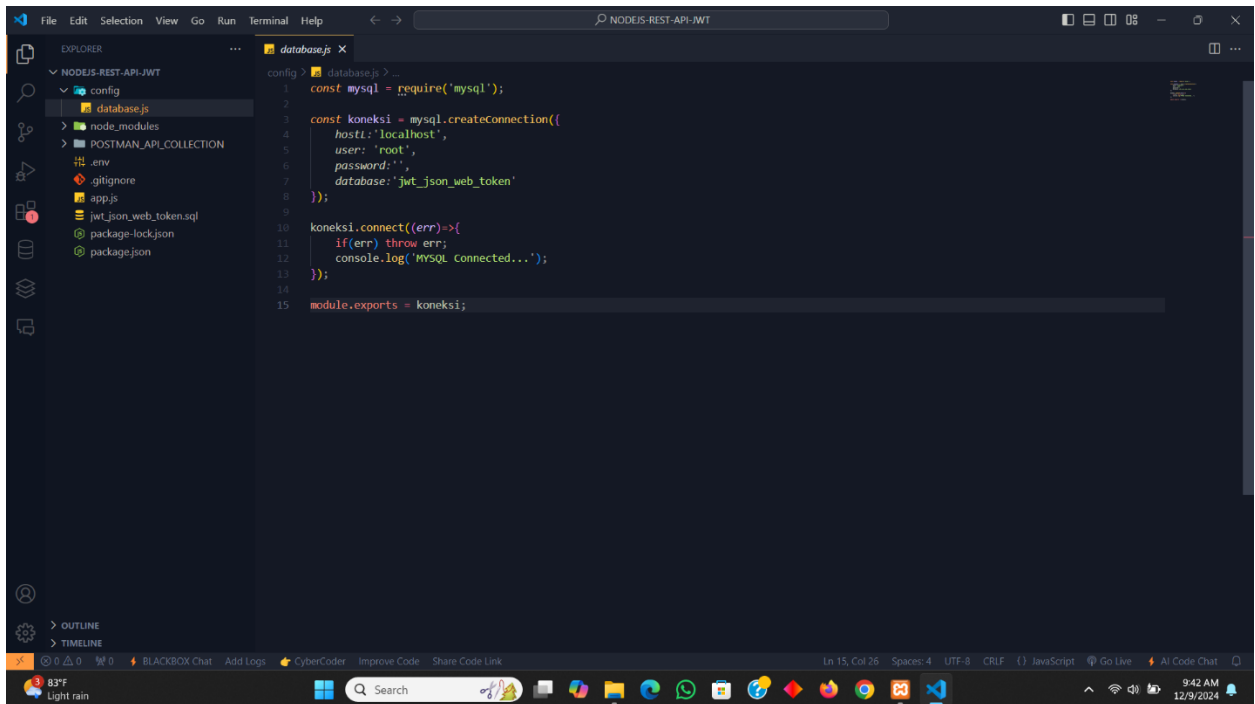
-Database nya

### III.LANGKAH KERJA

1. Pertama Buat Databasenya di PhpMyadmin dengan nama “jwt\_json\_web\_token” dan setelah itu klik menu import dan pilih file sql yang telah dibuat kemudian klik create untuk memasukkan database nya.

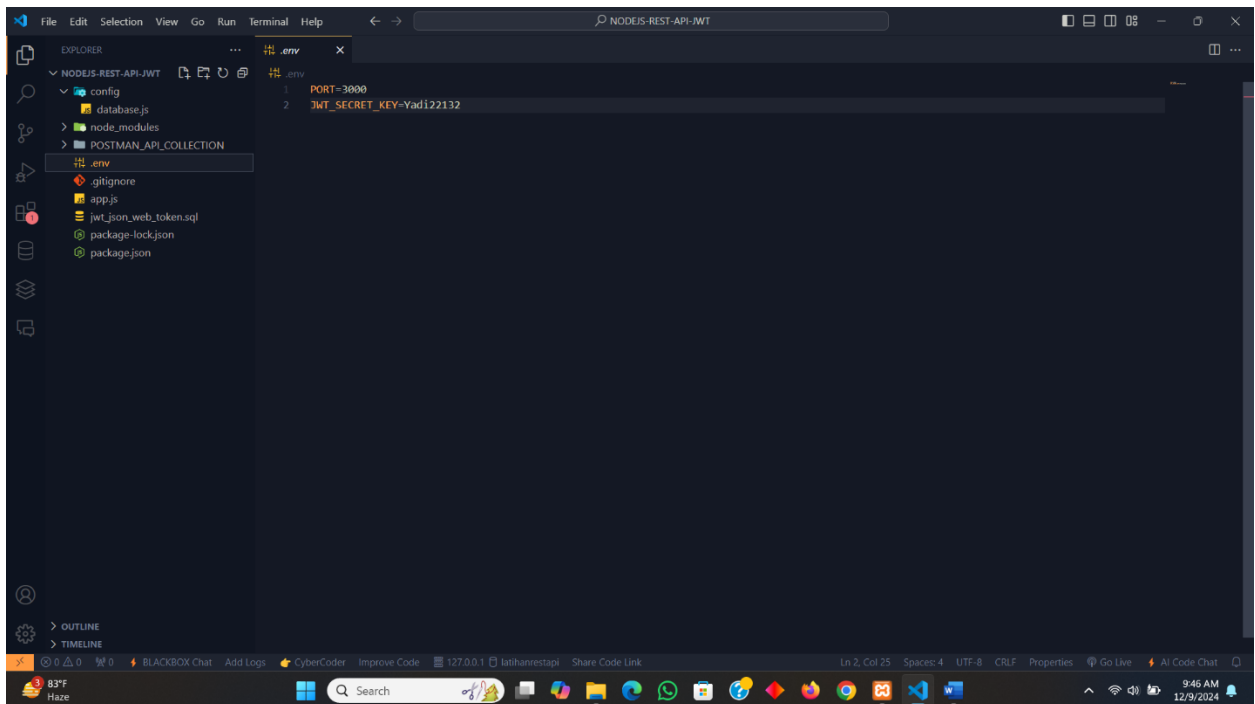


2. Di file Config isi database.js maka tampilan codingannya seperti di bawah ini



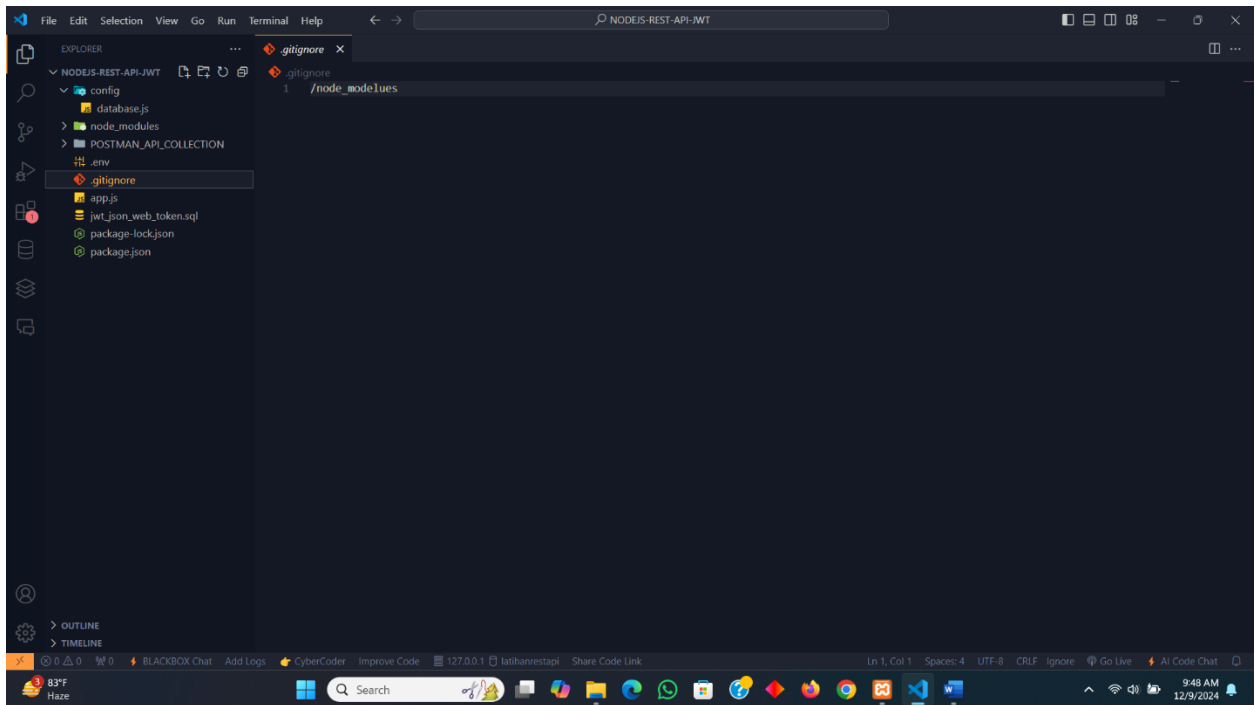
```
1 const mysql = require('mysql');
2
3 const koneksi = mysql.createConnection({
4   host: 'localhost',
5   user: 'root',
6   password: '',
7   database: 'jwt_json_web_token'
8 });
9
10 koneksi.connect((err) => {
11   if (err) throw err;
12   console.log('MySQL Connected...');
13 });
14
15 module.exports = koneksi;
```

3. isi dari env

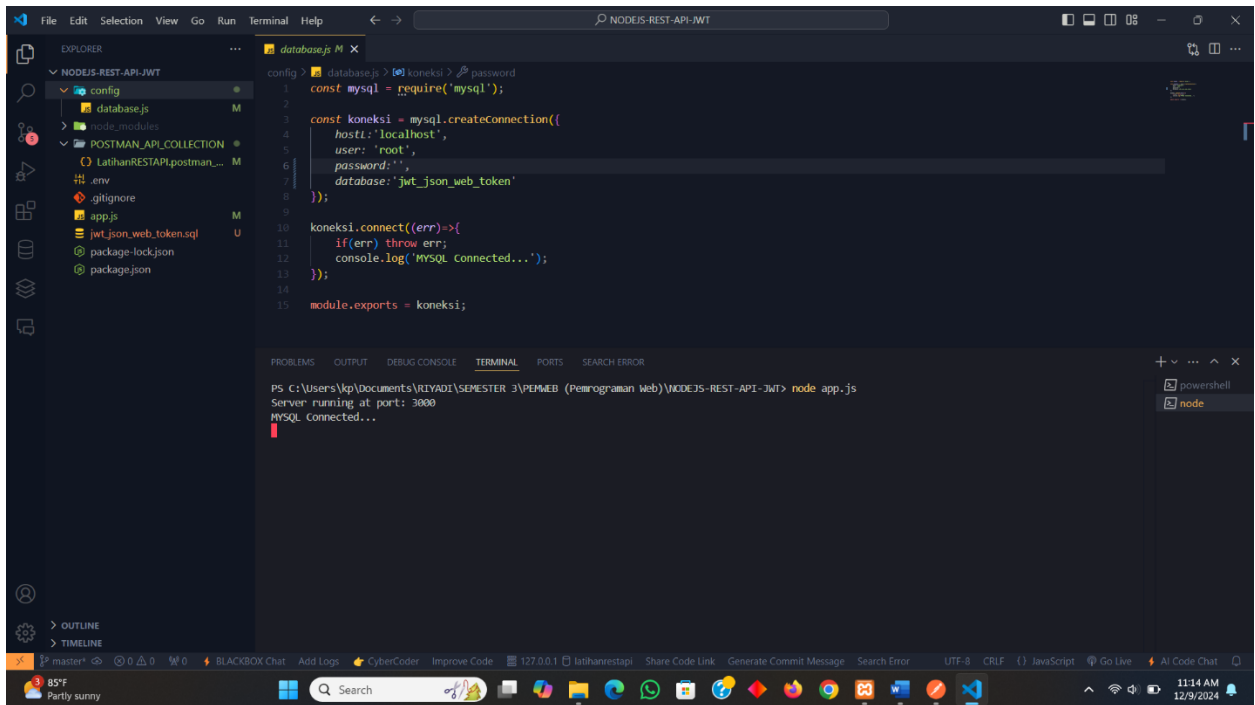


```
1 PORT=3000
2 JWT_SECRET_KEY=Yad122132
```

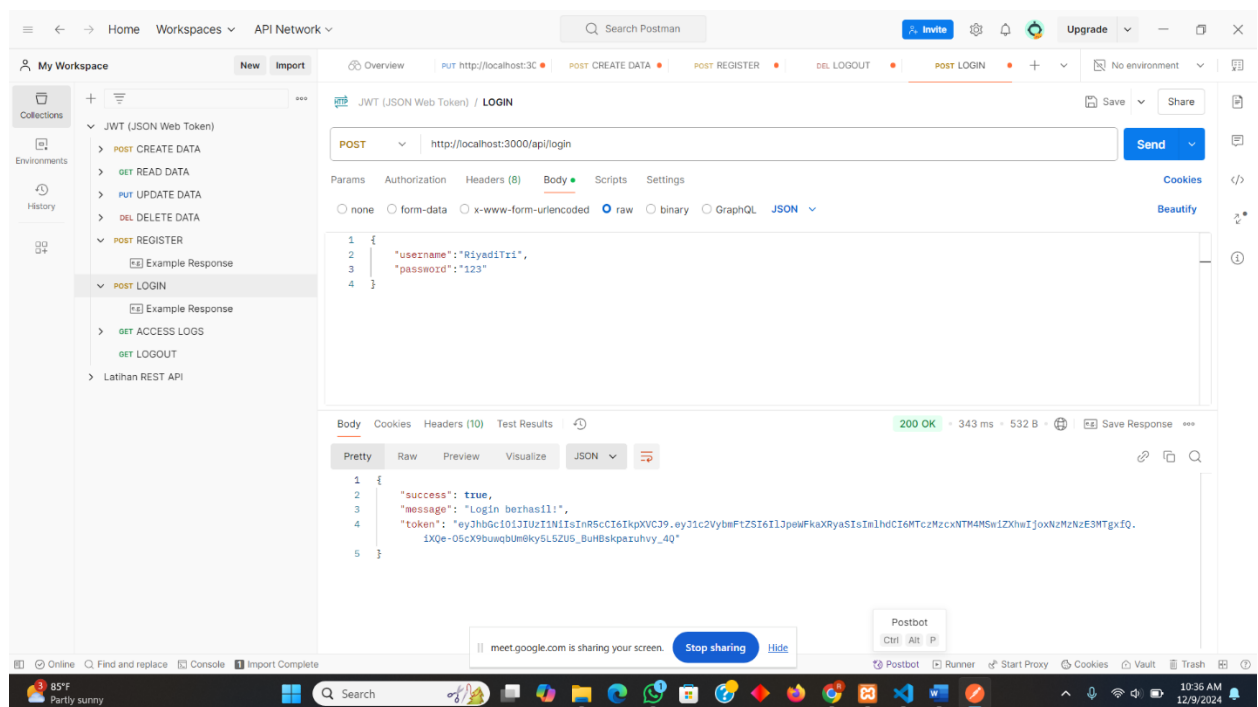
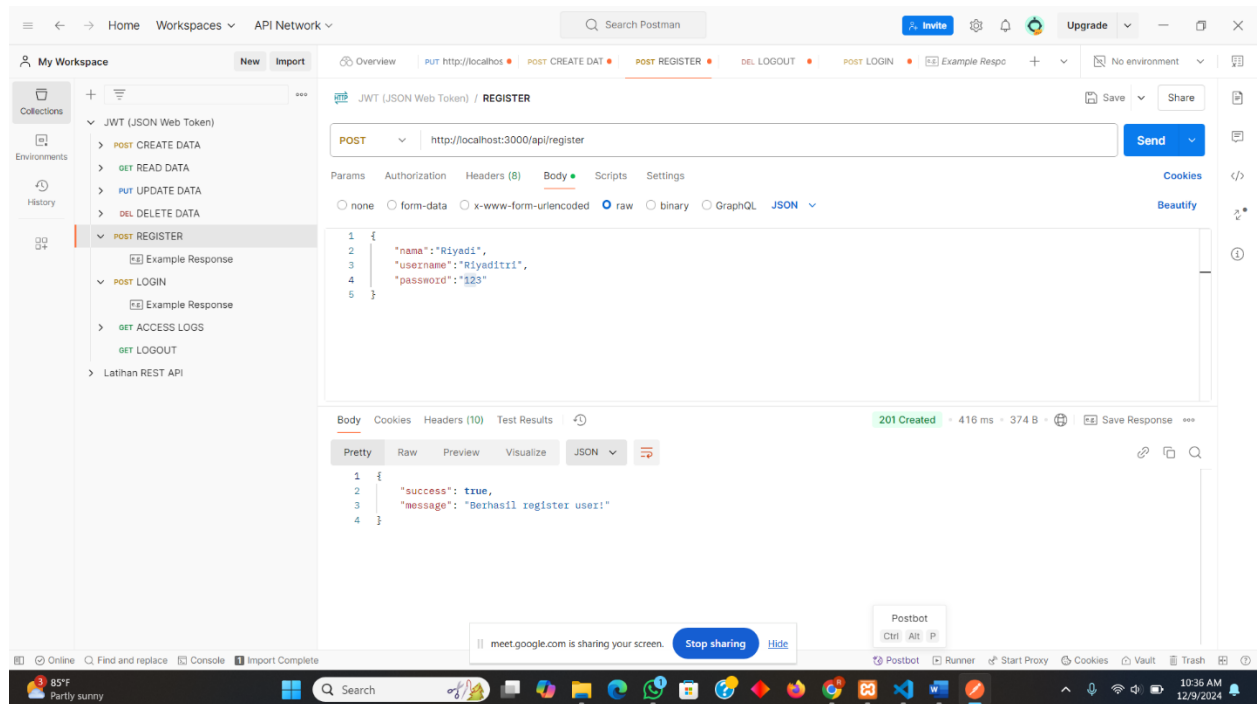
#### 4.isi dari gitignore



#### 5.Pertama,ketik di terminal “node app.js” untuk menjalankan programnya di postman

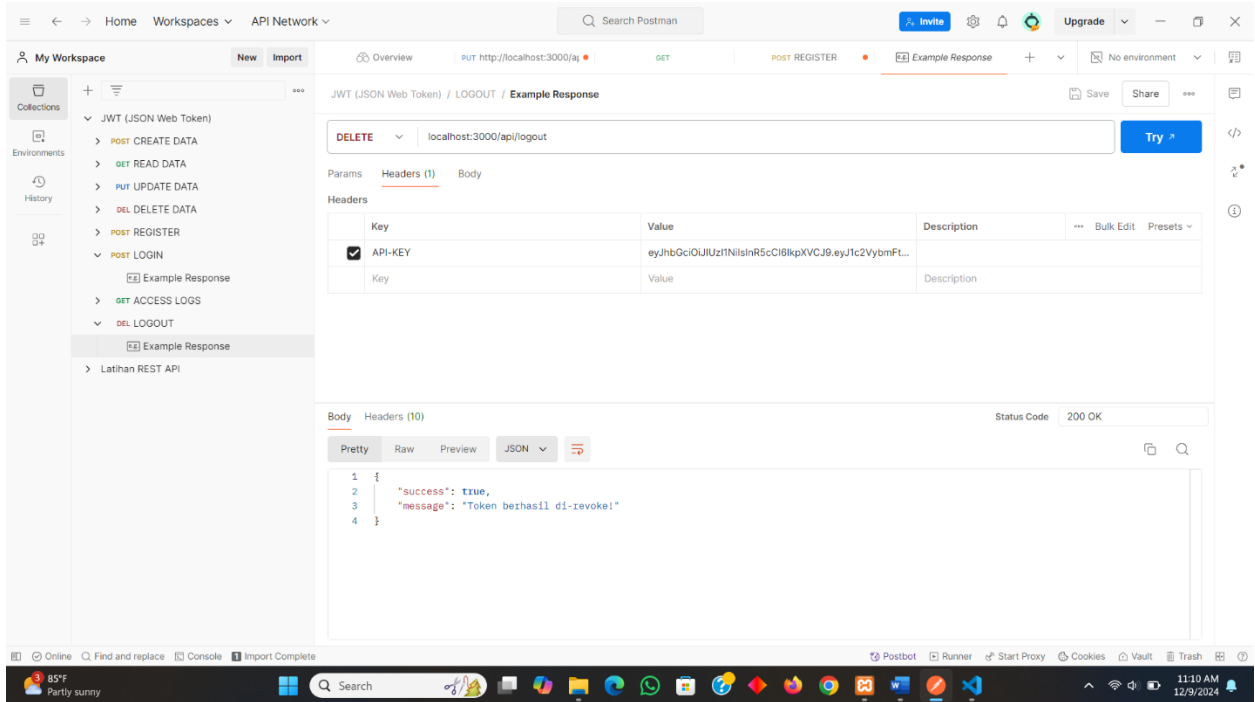


### 1.Membuat Sistem token untuk mengakses Data



2. Siapapun harus login dan dapatkan token (secret key).

### 3. Simpan seluruh token yang berlaku dan tidak berlaku (Revoke Token dan Rate Limiting)

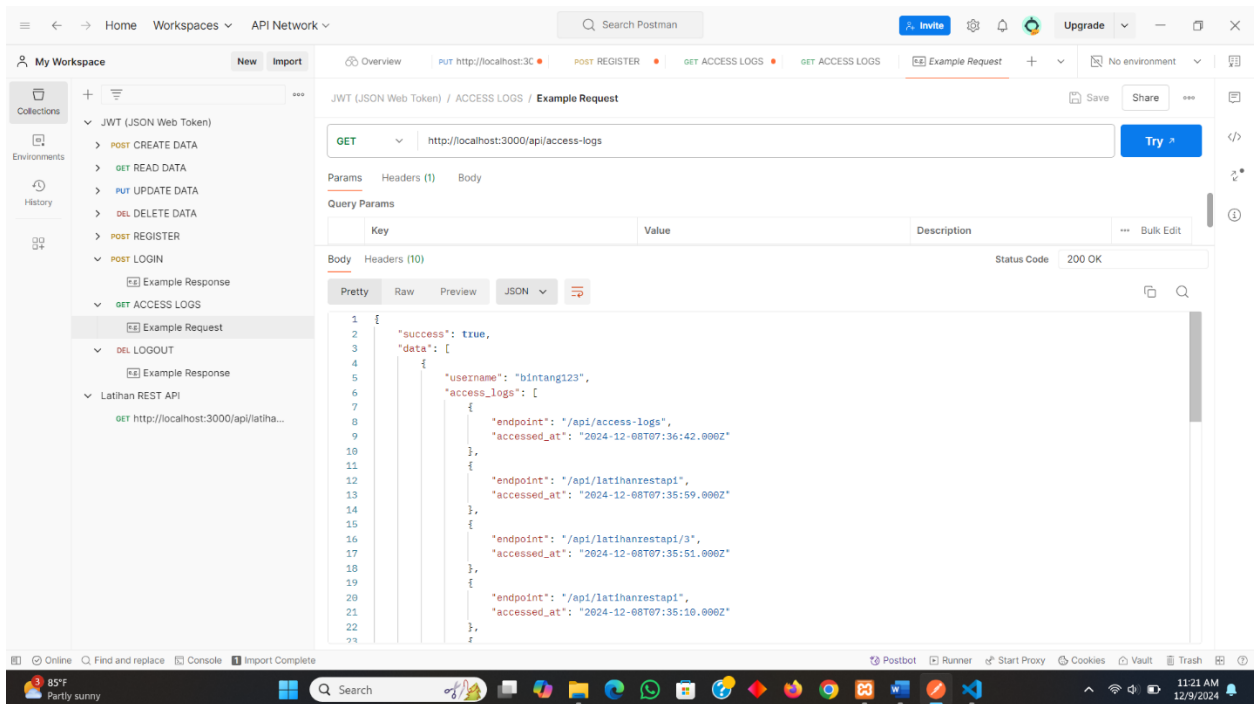


The screenshot shows the Postman interface with a workspace named 'API Network'. The left sidebar lists collections, including 'JWT (JSON Web Token)' and 'Lathan REST API'. The main panel displays a 'DELETE' request to 'localhost:3000/api/logout'. The 'Headers' tab is active, showing an 'API-KEY' header. The 'Body' tab shows a JSON response:

```
1 {
2   "success": true,
3   "message": "Token berhasil di-revoke!"
4 }
```

The status code is '200 OK'.

### 4. Mencatat Siapa Saja yang mengakses Data



The screenshot shows the Postman interface with a workspace named 'API Network'. The left sidebar lists collections, including 'JWT (JSON Web Token)' and 'Lathan REST API'. The main panel displays a 'GET' request to 'http://localhost:3000/api/access-logs'. The 'Body' tab shows a JSON response:

```
1 {
2   "success": true,
3   "data": [
4     {
5       "username": "bintang123",
6       "access_logs": [
7         {
8           "endpoint": "/api/access-logs",
9           "accessed_at": "2024-12-08T07:36:42.000Z"
10        },
11        {
12          "endpoint": "/api/lathanrestapi",
13          "accessed_at": "2024-12-08T07:35:59.000Z"
14        },
15        {
16          "endpoint": "/api/lathanrestapi/3",
17          "accessed_at": "2024-12-08T07:35:51.000Z"
18        },
19        {
20          "endpoint": "/api/lathanrestapi",
21          "accessed_at": "2024-12-08T07:35:10.000Z"
22        }
23      ]
24    }
25  ]
26 }
```

The status code is '200 OK'.

## IV.KESIMPULAN

Kesimpulan dari praktikum pemrograman web yang berfokus pada pembuatan sistem JWT (JSON Web Token) dapat dijelaskan sebagai berikut:

### 1. Pengenalan JWT

- JWT adalah sebuah standar token yang berbasis JSON, dirancang untuk memfasilitasi pertukaran informasi secara aman antara klien dan server.
- Struktur JWT terdiri dari tiga komponen utama: Header, Payload, dan Signature, yang terpisah oleh tanda titik ( ` . ` ).

### 2. Manfaat JWT

- JWT banyak digunakan dalam autentikasi dan otorisasi pengguna dalam aplikasi web.
- Dengan JWT, sistem dapat bersifat stateless, yang berarti server tidak perlu menyimpan informasi sesi karena semua data yang relevan sudah terenkapsulasi dalam token.

### 3. Tahapan Implementasi

- Membuat JWT: Token dihasilkan setelah pengguna berhasil masuk, biasanya dengan menggunakan pustaka seperti `jsonwebtoken` pada Node.js atau pustaka sejenis dalam bahasa pemrograman lainnya.
- Memvalidasi JWT: Server memeriksa keaslian JWT yang diterima dari klien dengan memverifikasi tanda tangan digital serta mengecek masa berlakunya (expiration).
- Menggunakan JWT di Klien: Klien mengirimkan JWT dalam header permintaan, contohnya di dalam `Authorization: Bearer`.

### 4. Keamanan JWT

- JWT harus dienkripsi atau dikirimkan melalui protokol aman seperti HTTPS untuk mencegah penyadapan.
- Penting untuk menetapkan waktu kedaluwarsa (`exp`) yang singkat pada token guna mengurangi risiko pemanfaatan token yang dicuri.
- Disarankan untuk menggunakan algoritma enkripsi yang kuat, seperti HS256 atau RS256.



## 5. Kelebihan

- JWT mudah diintegrasikan dengan berbagai framework dan bahasa pemrograman.
- Menurunkan beban server karena sifatnya yang stateless.
- Dapat digunakan untuk mengakses sumber daya API lintas domain.

## 6. Kekurangan

- Ukuran JWT yang relatif besar dapat memperlambat waktu respons.
- Jika token tidak dienkripsi, data dalam payload dapat diakses oleh pihak yang tidak berwenang.

## 7. Kesimpulan Praktis

- Sistem berbasis JWT sangat sesuai untuk aplikasi yang memerlukan autentikasi modern dan komunikasi API lintas platform.
- Namun, penerapan yang kurang hati-hati dapat menciptakan celah keamanan. Oleh karena itu, sangat penting untuk memahami dan mengikuti praktik terbaik dalam penggunaannya.

Dengan pemahaman dasar-dasar ini, Anda dapat mengimplementasikan JWT secara aman dan efisien dalam aplikasi web Anda.