

Nama : Tiara anggie maharani
: Nicolas pranandi dermawan
Nim : 20220040174
: 20220040175
Kelas : TI24M
Mata Kuliah : Pemrograman Berbasis Platform

Tugas 10

1. Endpoint mana yang paling sulit dikerjakan? Mengapa?
2. Apa error yang kamu temui hari ini dan bagaimana cara memperbaikinya?
3. Mengapa API modern menggunakan JSON?

Jawab

1. Endpoint yang paling tersulit adalah PUT untuk update data `/items/{nim}`, karena yang digunakan untuk memperbarui data berdasarkan identifikasi unik. Kesulitan ini muncul karena memerlukan integrasi logika yang kompleks, termasuk pencarian entri yang ada di basis data, validasi input baru untuk memastikan integritas data, serta penanganan kesalahan seperti kode respons HTTP 404 jika identifikasi tidak ditemukan.
2. Kesalahan yang kami hadapi hari ini adalah kesalahan SQL seperti 'ER_DUP_ENTRY' (duplicate entry) pada saat mencoba menambahkan data melalui endpoint POST ketika nilai `nidn` atau `nim` sudah ada di basis data, yang menyebabkan aplikasi mengembalikan respons Internal Server Error (status 500) tanpa penanganan khusus. Hal ini terjadi karena basis data MySQL (seperti dikonfigurasi di `db.js`) memiliki kolom `nidn` dan `nim` sebagai kunci unik, tetapi kode tidak memeriksa duplikasi sebelum insert. Solusi yang saya terapkan adalah dengan menambahkan query pengecekan duplikasi sebelum insert, serta penanganan error spesifik dalam callback query, seperti berikut:

```
connection.query('SELECT * FROM dosen WHERE nidn = ?', [nidn], (checkError, checkResults) => {  
  if (checkError) return res.status(500).json({ message: 'Internal Server Error' });  
  if (checkResults.length > 0) return res.status(409).json({ message: 'NIDN sudah ada' });  
  connection.query('INSERT INTO dosen ...', [nidn, ...], (error, results) => {  
    // ...  
  });  
});
```
3. JSON (JavaScript Object Notation) menjadi pilihan utama dalam API modern karena sifatnya yang ringan, mudah dibaca oleh manusia, serta kemampuannya untuk diproses secara efisien oleh berbagai bahasa pemrograman seperti Python dan JavaScript. Selain itu, ukuran data yang relatif kecil mengurangi latensi dalam layanan web, sehingga mendukung pertukaran data yang cepat dan skalabel.