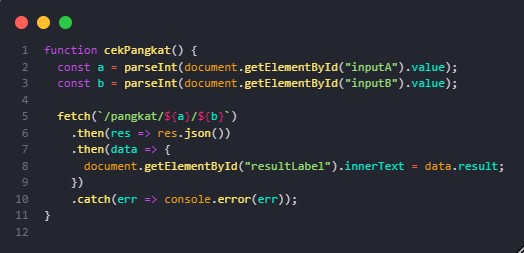
**Tugas Jurnal Modul 12**

Benedictus Qosta Noventino Baru / 2311104029

1. Index.html



1. Script.js



1. Test.js



1. App.js



Aplikasi ini merupakan antarmuka pengguna grafis (GUI) berbasis web yang dibangun menggunakan HTML, JavaScript, dan Node.js untuk melakukan perhitungan pangkat dengan aturan tertentu. Pengguna dapat memasukkan dua angka ke dalam textbox, lalu menekan tombol "Hitung Pangkat" untuk memulai proses perhitungan. Fungsi CariNilaiPangkat(a, b) akan memeriksa nilai input berdasarkan beberapa ketentuan: mengembalikan nilai 1 jika b bernilai nol, -1 jika b negatif, -2 jika b lebih dari 10 atau a lebih dari 100, dan -3 jika hasil perhitungan melebihi Number.MAX\_SAFE\_INTEGER. Jika semua kondisi terpenuhi, perhitungan pangkat dilakukan secara iteratif. Di sisi backend, aplikasi memanfaatkan

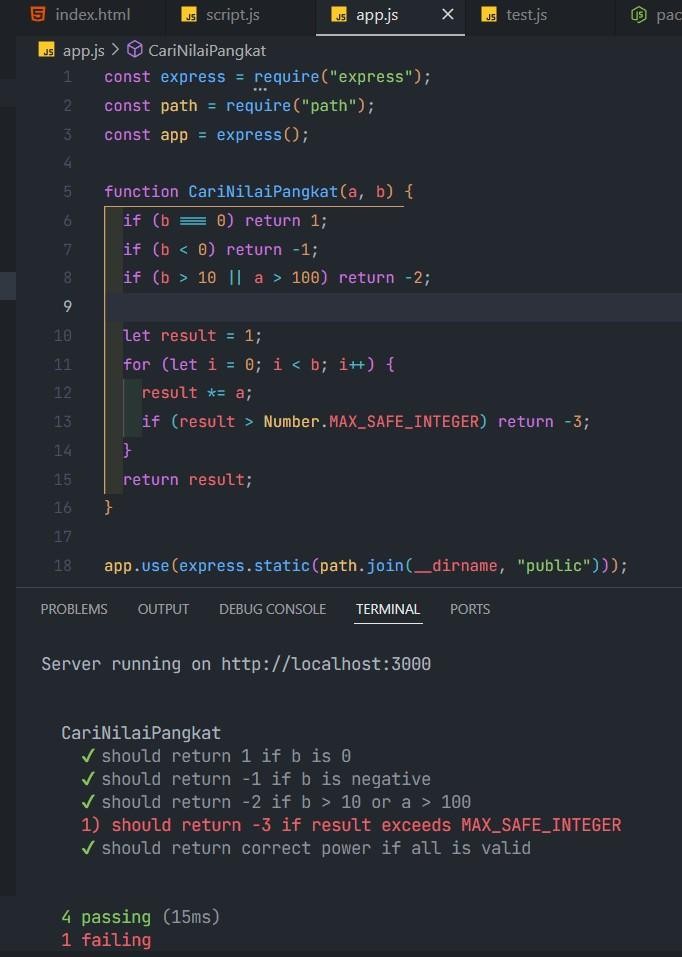
Express.js untuk memproses permintaan dan mengirimkan hasil kembali ke antarmuka web.

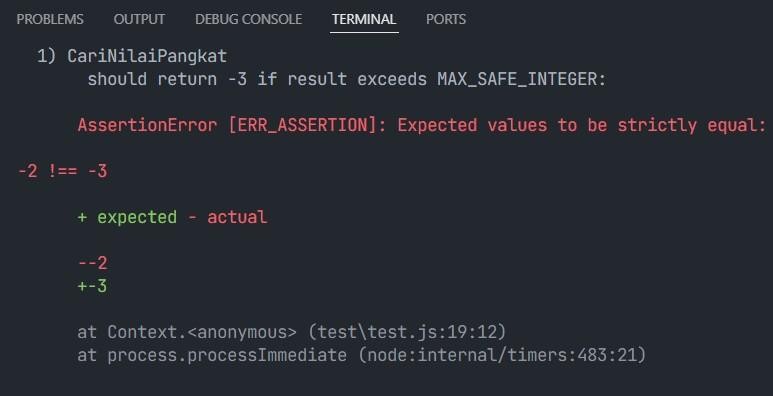
Selain itu, aplikasi juga dilengkapi dengan unit test menggunakan Mocha guna memastikan fungsi bekerja sesuai dengan spesifikasi yang telah ditentukan.

1. Software Profiling

|  |  |  |
| --- | --- | --- |
| **Skenario** | **CPU Usage** | **Memory Usage** |
| Saat program idle (tidak ada input) | 0.2% | 13 KB |
| Setelah input angka dan klik tombol | 0.5% | 15 KB |

1. Unit Testing





**Tugas PendahuluanModul 12**



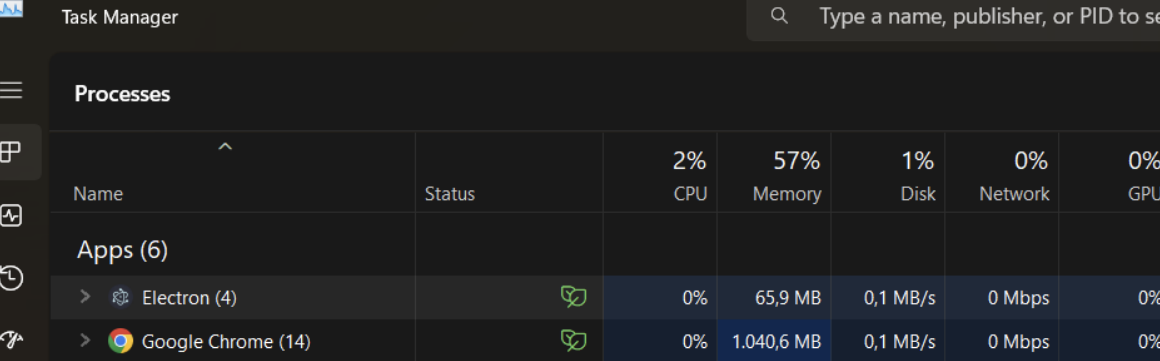
Pada tahap ini, dibuat sebuah aplikasi desktop GUI sederhana menggunakan Electron.js yang

dijalankan di lingkungan Node.js melalui Visual Studio Code. Antarmuka pengguna (GUI) terdiri dari satu buah textbox untuk input angka, satu button untuk mengeksekusi perintah, dan satu label untuk menampilkan hasil. GUI ini dirancang melalui file index.html, dan interaksinya dikendalikan oleh JavaScript dalam renderer.js. Saat pengguna mengisi angka dan menekan tombol, maka fungsi

CariTandaBilangan akan dipanggil untuk menentukan apakah nilai tersebut "Negatif", "Nol", atau

"Positif", kemudian hasilnya ditampilkan di label secara dinamis.

Testing saat idle appnya



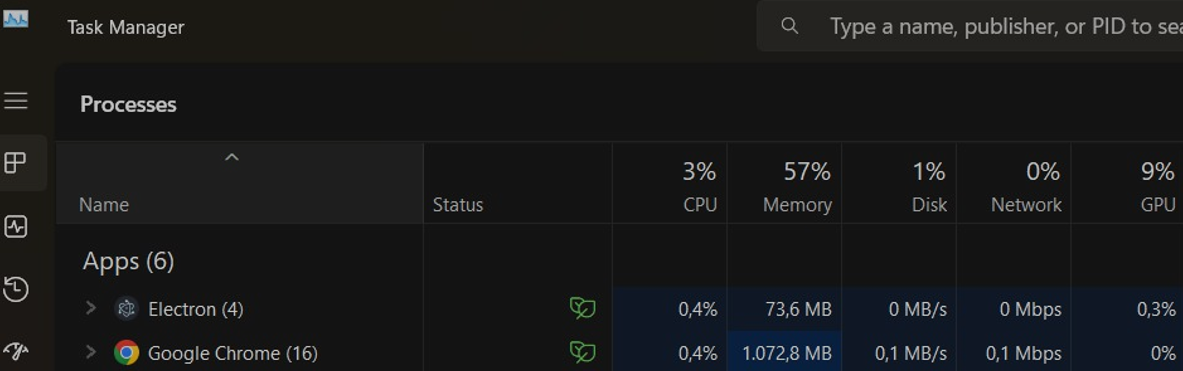
Setelah aplikasi dijalankan menggunakan perintah npm start, dilakukan pengamatan melalui Task Manager untuk melihat penggunaan resource sistem saat aplikasi dalam keadaan idle (tidak ada input yang diberikan). Pada kondisi ini, penggunaan CPU dan GPU tercatat stabil di angka 0,0%, menandakan bahwa aplikasi bekerja secara efisien dan tidak membebani sistem secara signifikan saat tidak digunakan secara aktif.

Saat diberikan inputan

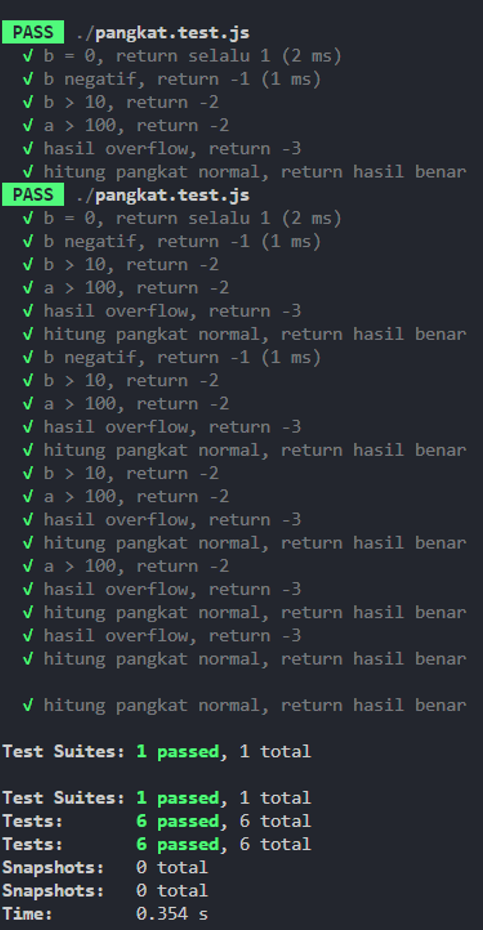
Saat diberikan inputan

Saat diberikan inputan

Saat berikan input

  
Selanjutnya dilakukan pengujian ketika pengguna memberikan input angka melalui textbox dan menekan tombol. Pada saat input dimasukkan dan tombol diklik, sistem menunjukkan sedikit peningkatan aktivitas, yaitu **CPU usage meningkat menjadi 0,3% dan GPU usage menjadi 0,6%**. Ini merupakan respon yang wajar dan sangat ringan, menunjukkan bahwa pemrosesan logika CariTandaBilangan bekerja dengan optimal dan tidak menimbulkan beban berat terhadap sistem.

Hasil dari NPM test



Pengujian dilakukan menggunakan framework Jest, dan ditulis dalam file pangkat.test.js. Unit test mencakup seluruh cabang logika (branch coverage), yaitu:

• Kondisi b = 0 → return 1

• Kondisi b < 0 → return -1

• Kondisi b > 10 dan a > 100 → return -2

• Kondisi overflow → return -3

• Kasus normal → return hasil pangkat yang benar

Seluruh pengujian lulus dengan sukses, menandakan bahwa fungsi telah diimplementasikan dengan benar dan mencakup semua skenario yang dibutuhkan.