

pangkat.test.js

```

1  const { CariNilaiPangkat } = require('../utils/pangkat');
2
3  test('b = 0 harus return 1', () => {
4    expect(CariNilaiPangkat(5, 0)).toBe(1);
5  });
6
7  test('b negatif return -1', () => {
8    expect(CariNilaiPangkat(5, -2)).toBe(-1);
9  });
10
11 test('a > 100 atau b > 10 return -2', () => {
12   expect(CariNilaiPangkat(101, 2)).toBe(-2);
13   expect(CariNilaiPangkat(5, 11)).toBe(-2);
14 });
15
16 test('hasil pangkat besar return -3', () => {
17   expect(CariNilaiPangkat(9, 30)).toBe(-3);
18 });
19
20 test('normal case 2^3 = 8', () => {
21   expect(CariNilaiPangkat(2, 3)).toBe(8);
22 });
23
24

```

Penjelasan :

Kode tersebut adalah kumpulan unit test menggunakan framework Jest untuk menguji fungsi CariNilaiPangkat dari modul pangkat. Setiap test memverifikasi perilaku fungsi terhadap kondisi tertentu. Jika $b = 0$, hasil harus 1. Jika b bernilai negatif, fungsi harus mengembalikan -1. Bila nilai a lebih dari 100 atau b lebih dari 10, maka hasilnya harus -2 sebagai bentuk validasi input. Untuk kasus perhitungan pangkat yang menghasilkan angka sangat besar dan melebihi batas aman JavaScript, fungsi harus mengembalikan -3. Terakhir, test normal seperti 2^3 memastikan fungsi memberikan hasil pangkat yang benar, yaitu 8. Pengujian ini penting untuk menjamin keakuratan dan ketahanan fungsi terhadap berbagai input.

pangkat.js

```

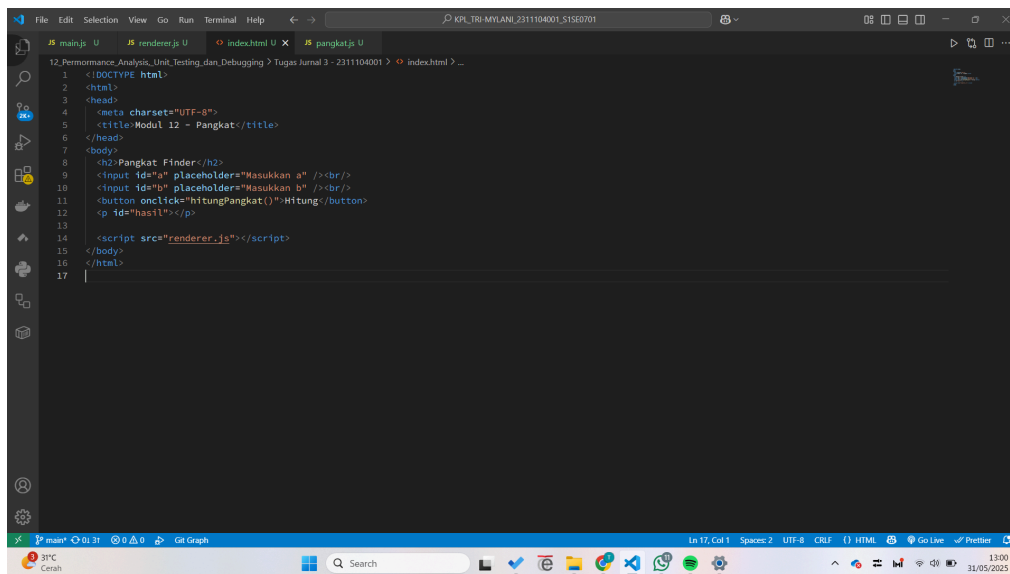
1  function CariNilaiPangkat(a, b) {
2    if (b === 0) return 1;
3    if (b < 0) return -1;
4    if (b > 10 || a > 100) return -2;
5
6    let hasil = 1;
7    for (let i = 0; i < b; i++) {
8      hasil *= a;
9      if (hasil > Number.MAX_SAFE_INTEGER) return -3;
10   }
11   return hasil;
12 }
13
14 module.exports = { CariNilaiPangkat };
15
16

```

Penjelasan :

Fungsi CariNilaiPangkat digunakan untuk menghitung nilai pangkat dari a pangkat b dengan sejumlah validasi input. Jika b sama dengan 0, fungsi langsung mengembalikan 1 karena aturan matematika. Jika b bernilai negatif, fungsi mengembalikan -1 sebagai tanda tidak mendukung pangkat negatif. Jika a lebih dari 100 atau b lebih dari 10, fungsi mengembalikan -2 sebagai pembatasan input. Perhitungan dilakukan melalui perulangan yang mengalikan a sebanyak b kali, dan jika hasilnya melebihi batas aman angka JavaScript (Number.MAX_SAFE_INTEGER), maka fungsi akan mengembalikan -3 untuk mencegah kesalahan numerik. Jika semua validasi terpenuhi dan tidak ada masalah, hasil akhir perhitungan akan dikembalikan.

index.html



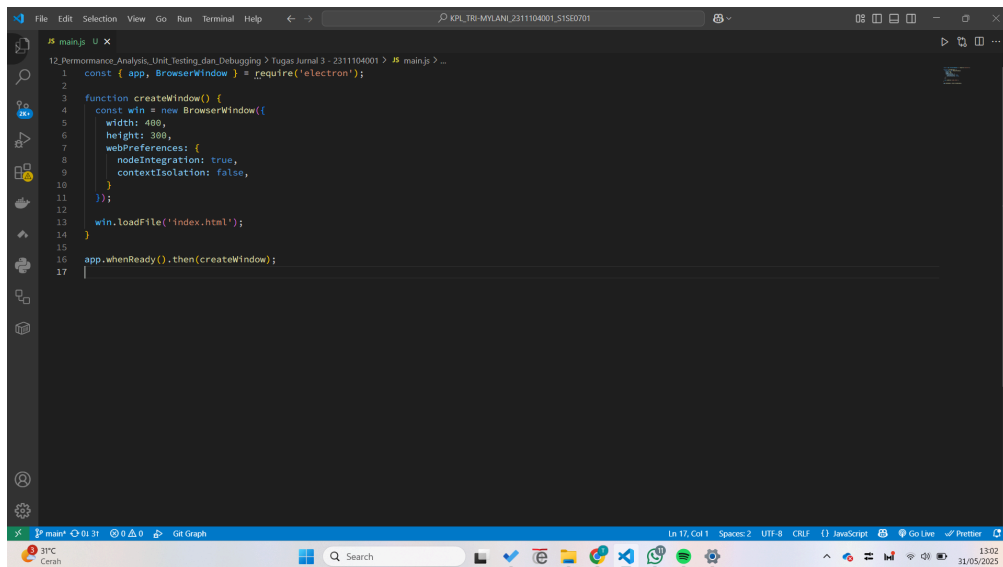
```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Modul 12 - Pangkat</title>
6 </head>
7 <body>
8   <h2>Pangkat Finder</h2>
9   <input id="a" placeholder="Masukkan a" /><br/>
10  <input id="b" placeholder="Masukkan b" /><br/>
11  <button onclick="hitungPangkat()">Hitung</button>
12  <p id="hasil"></p>
13
14  <script src="renderer.js"></script>
15 </body>
16 </html>
17

```

Penjelasan :

HTML tersebut adalah antarmuka pengguna sederhana untuk aplikasi "Pangkat Finder" yang memungkinkan pengguna menghitung pangkat dari dua angka. Terdapat dua input untuk memasukkan nilai a (basis) dan b (pangkat), serta sebuah tombol yang akan memicu fungsi hitungPangkat() saat diklik. Hasil perhitungan akan ditampilkan pada elemen paragraf dengan id hasil. Logika perhitungan disediakan dalam file eksternal renderer.js yang disisipkan melalui elemen <script>. Struktur ini cocok untuk dijalankan dalam konteks aplikasi desktop menggunakan Electron.

main.js


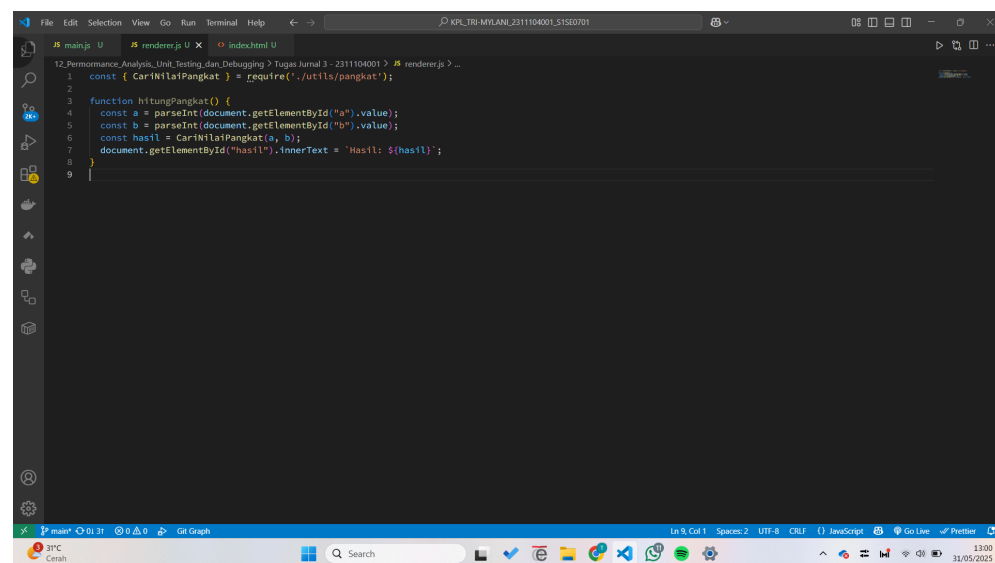
```

12_Performance_Analysik_Unit_Testing_dan_Debugging > Tugas Jurnal 3 - 2311104001 > main.js > ...
1  const { app, BrowserWindow } = require('electron');
2
3  function createWindow() {
4    const win = new BrowserWindow({
5      width: 400,
6      height: 300,
7      webPreferences: {
8        nodeIntegration: true,
9        contextIsolation: false,
10      }
11    });
12    win.loadFile('index.html');
13
14    app.whenReady().then(createWindow);
15  }
16
17

```

Penjelasan :

Kode JavaScript tersebut adalah bagian utama dari aplikasi Electron yang berfungsi untuk membuat GUI. Fungsi `createWindow()` untuk membuat sebuah jendela baru berukuran 400x300 piksel menggunakan objek `BrowserWindow`. Properti `webPreferences` diatur agar penggunaan modul Node.js di dalam file HTML dengan `nodeIntegration: true` dan menonaktifkan isolasi konteks (`contextIsolation: false`) agar skrip di halaman HTML bisa langsung mengakses Node.js. Setelah aplikasi Electron siap dijalankan (`app.whenReady()`), fungsi `createWindow` dipanggil untuk memuat file `index.html` ke dalam jendela aplikasi.

renderer.js


```

12_Performance_Analysik_Unit_Testing_dan_Debugging > Tugas Jurnal 3 - 2311104001 > renderer.js > ...
1  const { CarHitlaPangkat } = require('../utils/pangkat');
2
3  function hitungPangkat() {
4    const a = parseInt(document.getElementById("a").value);
5    const b = parseInt(document.getElementById("b").value);
6    const hasil = CarHitlaPangkat(a, b);
7    document.getElementById("hasil").innerText = `Hasil: ${hasil}`;
8  }
9

```

Penjelasan :

Kode JavaScript tersebut berfungsi untuk mengambil input dari pengguna berupa dua angka, a dan b, lalu menghitung hasil a pangkat b menggunakan fungsi CariNilaiPangkat yang diimpor dari file utils/pangkat. Fungsi hitungPangkat() akan dijalankan saat tombol diklik, mengambil nilai dari elemen input dengan id "a" dan "b", mengubahnya menjadi bilangan bulat dengan parseInt, lalu menghitung hasilnya. Hasil perhitungan tersebut kemudian ditampilkan dalam elemen paragraf dengan id "hasil". Kode ini merupakan bagian dari antarmuka aplikasi berbasis HTML dan JavaScript, kemungkinan digunakan dalam proyek Electron.

npm init -y

Membuat file konfigurasi package.json secara otomatis untuk proyek Node.js

npm install electron --save-dev

Perintah npm install electron --save-dev digunakan untuk menginstal Electron sebagai development dependency dalam proyek Node.js. Setelah perintah ini dijalankan, akan muncul folder baru bernama node_modules/ yang berisi semua pustaka yang dibutuhkan proyek. Selain itu, file package-lock.json akan dibuat atau diperbarui untuk mencatat versi dan struktur dependensi secara rinci, memastikan konsistensi instalasi. File package.json juga akan diperbarui, khususnya pada bagian devDependencies, yang akan mencantumkan Electron beserta versinya

npx electron main.js

