

## test/utlis.js

```

1  const { CariNilaiPangkat } = require('../utlis');
2
3  test('b = 0 returns 1', () => {
4    expect(CariNilaiPangkat(5, 0)).toBe(1);
5  });
6
7  test('b < 0 returns -1', () => {
8    expect(CariNilaiPangkat(2, -3)).toBe(-1);
9  });
10
11 test('a > 100 or b > 10 returns -2', () => {
12   expect(CariNilaiPangkat(101, 5)).toBe(-2);
13   expect(CariNilaiPangkat(5, 11)).toBe(-2);
14 });
15
16 test('overflow returns -3', () => {
17   expect(CariNilaiPangkat(9, 30)).toBe(-3);
18 });
19
20 test('normal case returns correct power', () => {
21   expect(CariNilaiPangkat(2, 5)).toBe(32);
22 });
23

```

## Penjelasan :

Kode tersebut merupakan rangkaian pengujian unit menggunakan Jest untuk memastikan fungsi CariNilaiPangkat bekerja sesuai yang diharapkan. Pengujian mencakup lima skenario utama: pertama, ketika nilai b adalah 0, fungsi harus mengembalikan 1; kedua, jika b bernilai negatif, hasilnya harus -1; ketiga, jika nilai a melebihi 100 atau b melebihi 10, fungsi harus mengembalikan -2 sebagai tanda input tidak valid; keempat, jika hasil perhitungan pangkat melebihi batas aman JavaScript (Number.MAX\_SAFE\_INTEGER), fungsi harus mengembalikan -3 untuk mencegah overflow; dan terakhir, untuk kasus normal seperti 2 pangkat 5, fungsi harus mengembalikan hasil perhitungan yang benar, yaitu 32. Pengujian ini penting untuk memastikan keandalan fungsi dalam berbagai kondisi.

## index.html

```

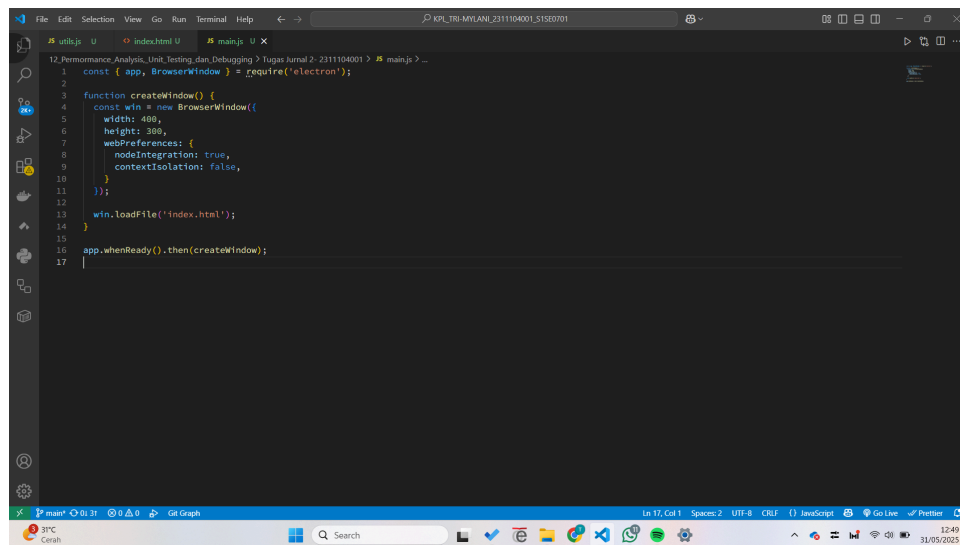
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Modul 12</title>
5  </head>
6  <body>
7    <h2>Pangkat Finder</h2>
8    <input id="a" placeholder="Masukkan a"><br>
9    <input id="b" placeholder="Masukkan b"><br>
10   <button onclick="handleClick()">Hitung Pangkat</button>
11   <p id="output"></p>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

```

### Penjelasan :

Kode HTML tersebut membentuk antarmuka sederhana untuk aplikasi "Pangkat Finder". Terdapat dua input untuk memasukkan nilai a dan b, serta sebuah tombol yang, saat diklik, akan menjalankan fungsi `handleClick()` dari file `renderer.js`. Hasil perhitungan pangkat akan ditampilkan di elemen paragraf dengan id `output`. Antarmuka ini digunakan bersama dengan backend Electron dan fungsi JavaScript untuk menghitung nilai pangkat sesuai input pengguna.

### main.js

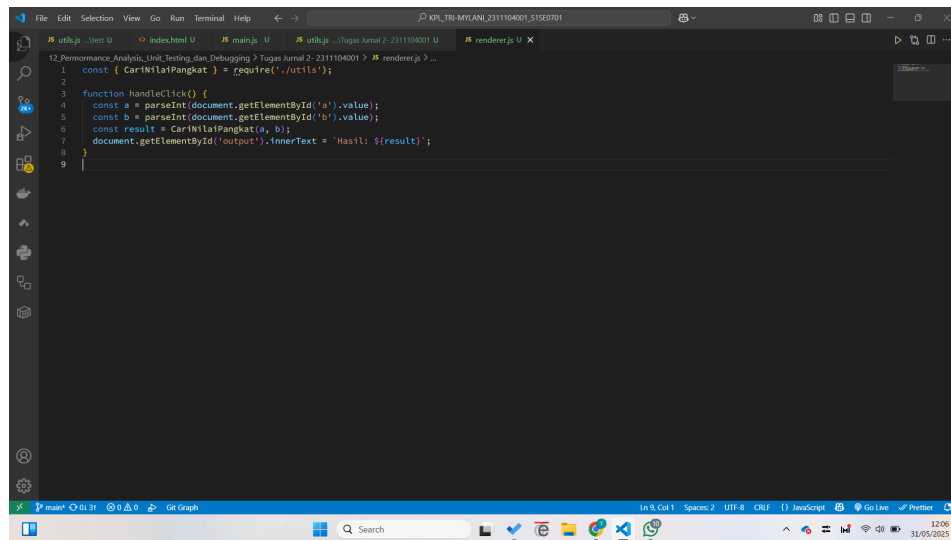


```
12. Performance Analysis, Unit Testing dan Debugging > Tugas Jurnal 2 - 2311104001 > # main.js > ...
1  const { app, BrowserWindow } = require('electron');
2
3  function createWindow() {
4    const win = new BrowserWindow({
5      width: 400,
6      height: 300,
7      webPreferences: {
8       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

### Penjelasan :

Kode tersebut merupakan bagian dari aplikasi Electron yang digunakan untuk membuat jendela aplikasi desktop. Fungsi `createWindow` membuat jendela baru dengan ukuran 400x300 piksel menggunakan `BrowserWindow`, dan mengatur agar Node.js dapat digunakan di dalam file HTML (melalui `nodeIntegration: true` dan `contextIsolation: false`). Setelah aplikasi siap (`app.whenReady()`), fungsi `createWindow` dijalankan untuk menampilkan file `index.html` sebagai antarmuka pengguna.

## renderer.js



```

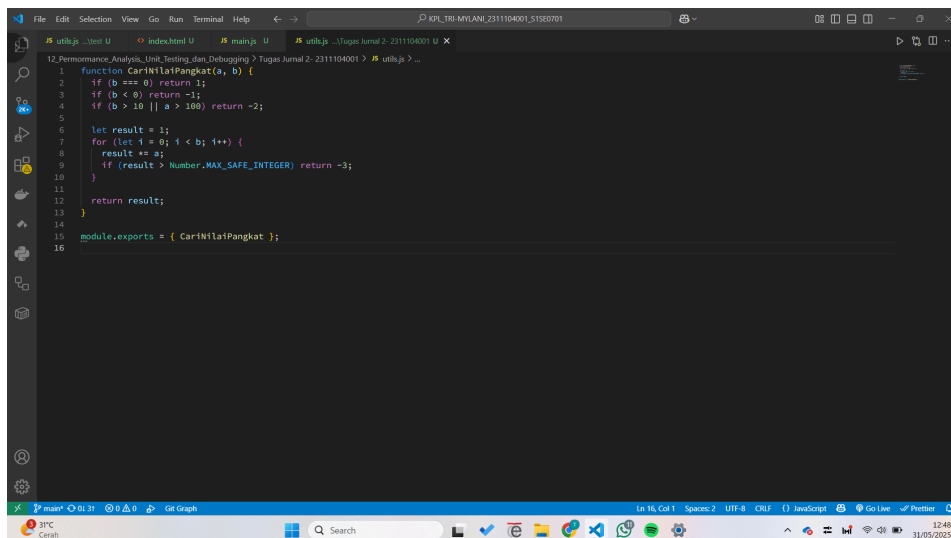
1  const { CarNilaiPangkat } = require('../utils');
2
3  function handleClick() {
4    const a = parseInt(document.getElementById('a').value);
5    const b = parseInt(document.getElementById('b').value);
6    const result = CarNilaiPangkat(a, b);
7    document.getElementById('output').innerText = `Hasil: ${result}`;
8  }
9

```

Penjelasan :

Kode tersebut berfungsi untuk mengambil nilai angka dari dua input dengan id 'a' dan 'b', lalu menggunakan fungsi CariNilaiPangkat dari modul utils untuk menghitung hasil pangkat dari angka tersebut. Setelah hasil dihitung, nilai tersebut ditampilkan pada elemen dengan id 'output' dalam bentuk teks "Hasil: [nilai]".

## utils.js



```

1  function CarNilaiPangkat(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 result = 1;
7    for (let i = 0; i < b; i++) {
8      result *= a;
9      if (result > Number.MAX_SAFE_INTEGER) return -3;
10   }
11
12   return result;
13 }
14
15 module.exports = { CarNilaiPangkat };
16

```

Penjelasan :

Fungsi CariNilaiPangkat digunakan untuk menghitung nilai pangkat dari a dipangkatkan dengan b, dengan beberapa validasi. Jika b bernilai 0, fungsi akan mengembalikan 1. Jika b negatif, akan mengembalikan -1. Jika nilai b lebih dari 10 atau a lebih dari 100, fungsi akan mengembalikan -2. Selama perhitungan, jika hasilnya melebihi batas maksimum angka aman di JavaScript (Number.MAX\_SAFE\_INTEGER), maka fungsi mengembalikan -3. Jika tidak

ada masalah, hasil akhir dari  $a^b$  akan dikembalikan.

### **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**

