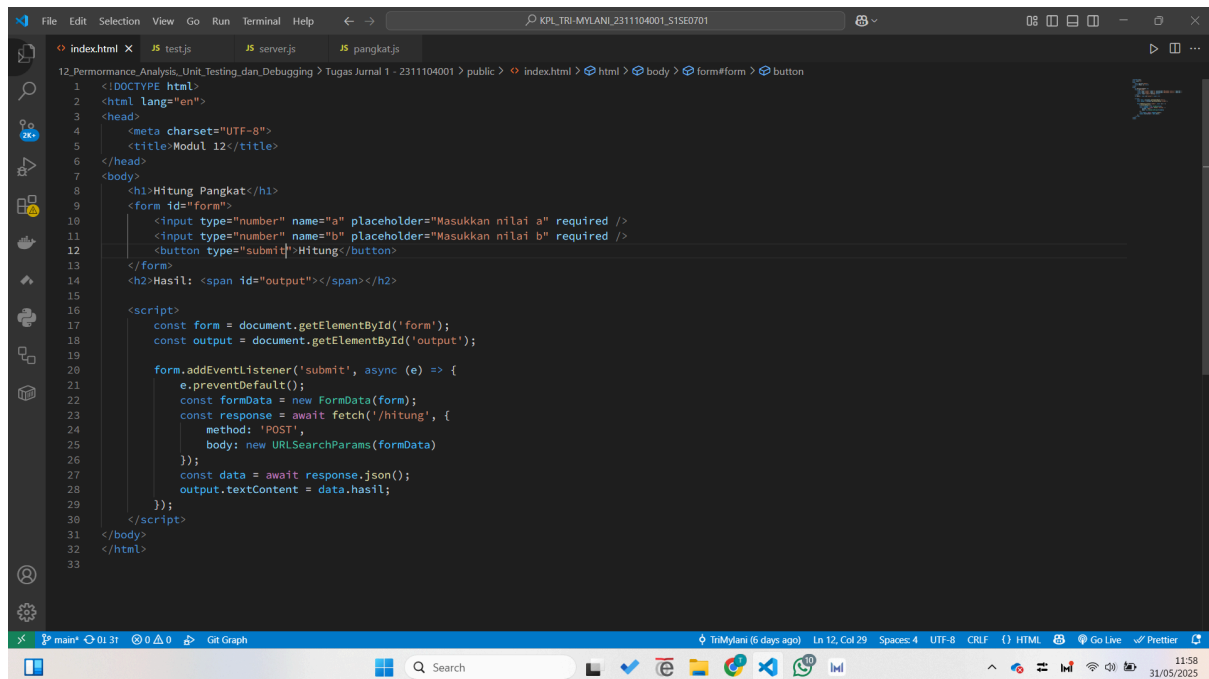


index.html



```

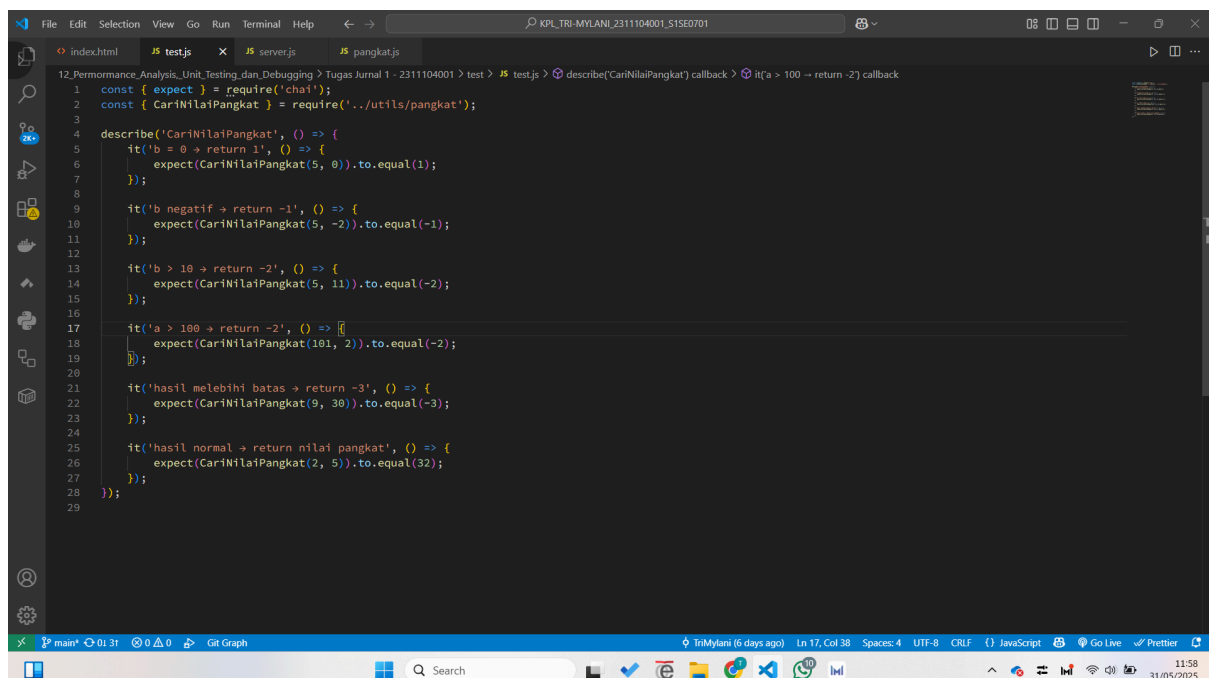
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Modul 12</title>
6 </head>
7 <body>
8   <h1>Hitung Pangkat</h1>
9   <form id="form">
10     <input type="number" name="a" placeholder="Masukkan nilai a" required />
11     <input type="number" name="b" placeholder="Masukkan nilai b" required />
12     <button type="submit">Hitung</button>
13   </form>
14   <h2>Hasil: <span id="output"></span></h2>
15
16   <script>
17     const form = document.getElementById('form');
18     const output = document.getElementById('output');
19
20     form.addEventListener('submit', async (e) => {
21       e.preventDefault();
22       const formData = new FormData(form);
23       const response = await fetch('/hitung', {
24         method: 'POST',
25         body: new URLSearchParams(formData)
26       });
27       const data = await response.json();
28       output.textContent = data.hasil;
29     });
30   </script>
31 </body>
32 </html>
33

```

Penjelasan :

Kode HTML dan JavaScript tersebut merupakan antarmuka web sederhana untuk menghitung pangkat dari dua bilangan (a dan b) yang dimasukkan oleh pengguna. Ketika tombol "Hitung" ditekan, data dikirim ke server melalui metode POST ke endpoint /hitung, lalu server mengembalikan hasil perhitungan dalam format JSON yang kemudian ditampilkan di halaman.

test.js



```

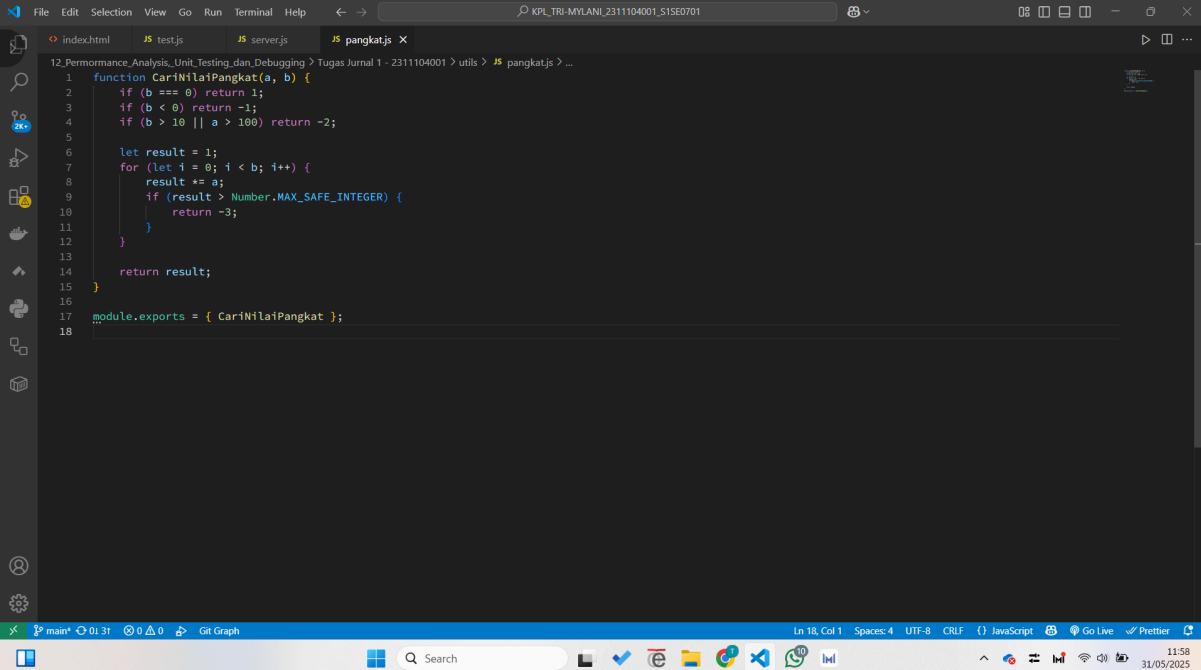
1 const { expect } = require('chai');
2 const { CarNilaiPangkat } = require('../utils/pangkat');
3
4 describe('CarNilaiPangkat', () => {
5   it('b = 0 → return 1', () => {
6     expect(CarNilaiPangkat(5, 0)).to.equal(1);
7   });
8
9   it('b negatif → return -1', () => {
10     expect(CarNilaiPangkat(5, -2)).to.equal(-1);
11   });
12
13   it('b > 10 → return -2', () => {
14     expect(CarNilaiPangkat(5, 11)).to.equal(-2);
15   });
16
17   it('a > 100 → return -2', () => {
18     expect(CarNilaiPangkat(101, 2)).to.equal(-2);
19   });
20
21   it('hasil melebihi batas → return -3', () => {
22     expect(CarNilaiPangkat(9, 30)).to.equal(-3);
23   });
24
25   it('hasil normal → return nilai pangkat', () => {
26     expect(CarNilaiPangkat(2, 5)).to.equal(32);
27   });
28 });
29

```

Penjelasan :

Fungsi CariNilaiPangkat menggunakan framework Mocha dan assertion library Chai, hasil pangkat terlalu besar melebihi batas tipe data (mengembalikan -3), serta kasus normal untuk memastikan fungsi menghitung pangkat dengan benar; setiap pengujian menggunakan `expect().to.equal()` untuk mencocokkan hasil aktual dengan yang diharapkan.

pangkat.js



```

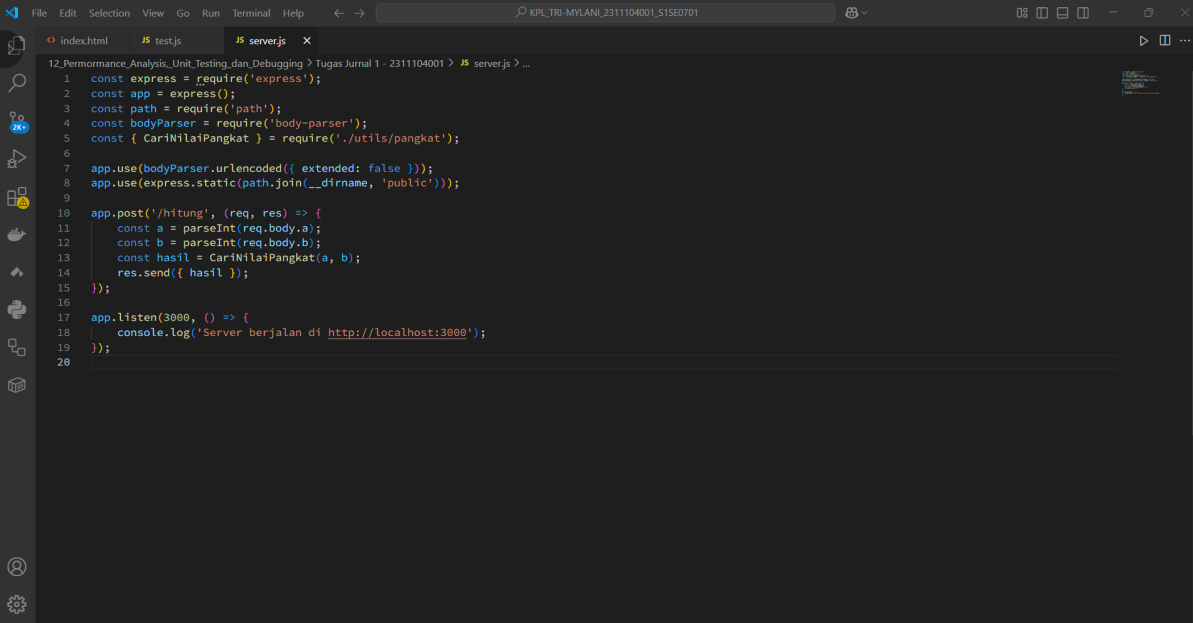
1 function CariNilaiPangkat(a, b) {
2   if (b === 0) return 1;
3   if (b < 0) return -1;
4   if (b > 10 || a > 100) return -2;
5
6   let result = 1;
7   for (let i = 0; i < b; i++) {
8     result *= a;
9     if (result > Number.MAX_SAFE_INTEGER) {
10      return -3;
11    }
12  }
13
14  return result;
15 }
16
17 module.exports = { CariNilaiPangkat };
18

```

Penjelasan :

Fungsi CariNilaiPangkat di atas digunakan untuk menghitung nilai pangkat dari dua bilangan (a^b) dengan beberapa aturan khusus. Jika b sama dengan 0, fungsi akan mengembalikan 1; jika b negatif, akan mengembalikan -1; jika b lebih dari 10 atau a lebih dari 100, akan mengembalikan -2; dan jika hasil perkalian melebihi batas aman integer JavaScript (`Number.MAX_SAFE_INTEGER`), maka akan mengembalikan -3. Jika semua kondisi lolos, fungsi akan menghitung pangkat a terhadap b secara iteratif tanpa menggunakan fungsi bawaan, lalu mengembalikan hasilnya. Fungsi ini diekspor agar bisa digunakan di file lain seperti dalam pengujian unit test.

server.js



```

1 const express = require('express');
2 const app = express();
3 const path = require('path');
4 const bodyParser = require('body-parser');
5 const { CariNilaiPangkat } = require('./utils/pangkat');
6
7 app.use(bodyParser.urlencoded({ extended: false }));
8 app.use(express.static(path.join(__dirname, 'public')));
9
10 app.post('/hitung', (req, res) => {
11   const a = parseInt(req.body.a);
12   const b = parseInt(req.body.b);
13   const hasil = CariNilaiPangkat(a, b);
14   res.send({ hasil });
15 });
16
17 app.listen(3000, () => {
18   console.log('Server berjalan di http://localhost:3000');
19 });
20

```

Penjelasan :

Kode di atas adalah implementasi backend menggunakan Express.js yang berfungsi sebagai server untuk menghitung pangkat berdasarkan input dari client. Server ini menerima data a dan b melalui metode POST ke endpoint /hitung, lalu memanggil fungsi CariNilaiPangkat untuk menghitung hasilnya, dan mengembalikan hasil tersebut dalam format JSON. Selain itu, server juga mengatur middleware body-parser untuk membaca data form, dan menyajikan file statis dari folder public (misalnya file HTML). Server berjalan pada port 3000 dan bisa diakses melalui `http://localhost:3000`.

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 server.js

