

## Tugas Jurnal Modul 12

### 1. 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>Modul 12 - Pangkat Kustom</h1>
9   <input type="number" id="inputA" placeholder="Angka A">
10  <input type="number" id="inputB" placeholder="Angka B">
11  <button onclick="cekPangkat()">Hitung Pangkat</button>
12  <p id="resultLabel"></p>
13
14  <script src="script.js"></script>
15 </body>
16 </html>
17
```

### 2. Script.js

```
1 function cekPangkat() {
2   const a = parseInt(document.getElementById("inputA").value);
3   const b = parseInt(document.getElementById("inputB").value);
4
5   fetch(`/pangkat/${a}/${b}`)
6     .then(res => res.json())
7     .then(data => {
8       document.getElementById("resultLabel").innerText = data.result;
9     })
10    .catch(err => console.error(err));
11 }
12
```

### 3. Test.js

```
1  const assert = require("assert");
2  const { CariNilaiPangkat } = require("../app");
3
4  describe("CariNilaiPangkat", () => {
5    it("should return 1 if b is 0", () => {
6      assert.strictEqual(CariNilaiPangkat(5, 0), 1);
7    });
8
9    it("should return -1 if b is negative", () => {
10     assert.strictEqual(CariNilaiPangkat(5, -2), -1);
11   });
12
13   it("should return -2 if b > 10 or a > 100", () => {
14     assert.strictEqual(CariNilaiPangkat(5, 11), -2);
15     assert.strictEqual(CariNilaiPangkat(101, 2), -2);
16   });
17
18   it("should return -3 if result exceeds MAX_SAFE_INTEGER", () => {
19     assert.strictEqual(CariNilaiPangkat(9, 30), -3);
20   });
21
22   it("should return correct power if all is valid", () => {
23     assert.strictEqual(CariNilaiPangkat(2, 3), 8);
24   });
25 });
26
```

#### 4. App.js

```
1  const express = require("express");
2  const path = require("path");
3  const app = express();
4
5  function CariNilaiPangkat(a, b) {
6    if (b === 0) return 1;
7    if (b < 0) return -1;
8    if (b > 10 || a > 100) return -2;
9
10   let result = 1;
11   for (let i = 0; i < b; i++) {
12     result *= a;
13     if (result > Number.MAX_SAFE_INTEGER) return -3;
14   }
15   return result;
16 }
17
18 app.use(express.static(path.join(__dirname, "public")));
19
20 app.get("/pangkat/:a/:b", (req, res) => {
21   const a = parseInt(req.params.a);
22   const b = parseInt(req.params.b);
23   const result = CariNilaiPangkat(a, b);
24   res.json({ result });
25 });
26
27 app.listen(3000, () => {
28   console.log("Server running on http://localhost:3000");
29 });
30
31 module.exports = { CariNilaiPangkat };
32
```

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.

## 5. 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

## 6. Unit Testing



```
index.html script.js app.js test.js pac
app.js > CariNilaiPangkat
1  const express = require("express");
2  const path = require("path");
3  const app = express();
4
5  function CariNilaiPangkat(a, b) {
6    if (b === 0) return 1;
7    if (b < 0) return -1;
8    if (b > 10 || a > 100) return -2;
9
10   let result = 1;
11   for (let i = 0; i < b; i++) {
12     result *= a;
13     if (result > Number.MAX_SAFE_INTEGER) return -3;
14   }
15   return result;
16 }
17
18 app.use(express.static(path.join(__dirname, "public")));

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Server running on http://localhost:3000

CariNilaiPangkat
✓ should return 1 if b is 0
✓ should return -1 if b is negative
✓ should return -2 if b > 10 or a > 100
1) should return -3 if result exceeds MAX_SAFE_INTEGER
✓ should return correct power if all is valid

4 passing (15ms)
1 failing
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

1) CariNilaiPangkat  
should return -3 if result exceeds MAX\_SAFE\_INTEGER:

AssertionError [ERR\_ASSERTION]: Expected values to be strictly equal:

-2 !== -3

+ expected - actual

--2

+-3

at Context.<anonymous> (test\test.js:19:12)

at process.processImmediate (node:internal/timers:483:21)