

2311104035

Reza Irawan

## Tugas Jurnal Modul 12

### 1. Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>Modul 12</title>
</head>
<body>
  <h1>Modul 12 - Pangkat Kustom</h1>
  <input type="number" id="inputA" placeholder="Angka A">
  <input type="number" id="inputB" placeholder="Angka B">
  <button onclick="cekPangkat()">Hitung Pangkat</button>
  <p id="resultLabel"></p>

  <script src="script.js"></script>
</body>
</html>
```

### 2. Script.js

```
function cekPangkat() {
  const a = parseInt(document.getElementById("inputA").value);
  const b = parseInt(document.getElementById("inputB").value);

  fetch(`/pangkat/${a}/${b}`)
    .then(res => res.json())
    .then(data => {
      document.getElementById("resultLabel").innerText = data.result;
    })
    .catch(err => console.error(err));
}
```

### 3. Test.js



```
const assert = require("assert");
const { CariNilaiPangkat } = require("../app");

describe("CariNilaiPangkat", () => {
  it("should return 1 if b is 0", () => {
    assert.strictEqual(CariNilaiPangkat(5, 0), 1);
  });

  it("should return -1 if b is negative", () => {
    assert.strictEqual(CariNilaiPangkat(5, -2), -1);
  });

  it("should return -2 if b > 10 or a > 100", () => {
    assert.strictEqual(CariNilaiPangkat(5, 11), -2);
    assert.strictEqual(CariNilaiPangkat(101, 2), -2);
  });

  it("should return -3 if result exceeds MAX_SAFE_INTEGER", () => {
    assert.strictEqual(CariNilaiPangkat(9, 30), -3);
  });

  it("should return correct power if all is valid", () => {
    assert.strictEqual(CariNilaiPangkat(2, 3), 8);
  });
});
```

#### 4. App.js

```
const express = require("express");
const path = require("path");
const app = express();

function CariNilaiPangkat(a, b) {
  if (b === 0) return 1;
  if (b < 0) return -1;
  if (b > 10 || a > 100) return -2;

  let result = 1;
  for (let i = 0; i < b; i++) {
    result *= a;
    if (result > Number.MAX_SAFE_INTEGER) return -3;
  }
  return result;
}

app.use(express.static(path.join(__dirname, "public")));

app.get("/pangkat/:a/:b", (req, res) => {
  const a = parseInt(req.params.a);
  const b = parseInt(req.params.b);
  const result = CariNilaiPangkat(a, b);
  res.json({ result });
});

app.listen(3000, () => {
  console.log("Server running on http://localhost:3000");
});

module.exports = { CariNilaiPangkat };
```

Aplikasi ini adalah implementasi GUI berbasis web menggunakan HTML, JavaScript, dan Node.js untuk menghitung pangkat dengan aturan khusus. Pengguna memasukkan dua angka ke dalam dua textbox, lalu menekan tombol "Hitung Pangkat" untuk memprosesnya. Fungsi `CariNilaiPangkat(a, b)` akan mengevaluasi input berdasarkan beberapa kondisi: mengembalikan 1 jika pangkat b adalah nol, -1 jika b negatif, -2 jika

$b > 10$  atau  $a > 100$ , dan  $-3$  jika hasil pangkat melebihi batas maksimal `Number.MAX_SAFE_INTEGER`. Jika semua valid, fungsi menghitung pangkat dengan iterasi. Backend menggunakan Express.js untuk menangani request dan mengembalikan hasilnya ke halaman web. Aplikasi juga dilengkapi dengan unit testing menggunakan Mocha untuk memastikan fungsi berjalan sesuai aturan.

## 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

```
> mocha
Debugger attached.
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 (18ms)
1 failing

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:19:16)
    at process.processImmediate (node:internal/timers:491:21)
```

