2311104025
Eduardo Bagus Prima Julian
SE0701

Main.js:

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

  let result = 1;
  try {
    for (let i = 0; i < b; i++) {
      result = checkedMultiply(result, a);
    }
    return result;
  } catch (e) {
    return -3;
  }
}

function checkedMultiply(x, y) {
  let res = x * y;
  if (!Number.isSafeInteger(res)) {
    throw new Error("Overflow");
  }
  return res;
}

function hitung() {
  const a = parseInt(document.getElementById("inputA").value);
  const b = parseInt(document.getElementById("inputB").value);
  const hasil = CariNilaiPangkat(a, b);
  document.getElementById("output").innerText = `Hasil: ${hasil}`;
}

// Export untuk unit test
if (typeof module !== 'undefined') {
  module.exports = { CariNilaiPangkat };
}
```

Penjelasan:

Fungsi CariNilaiPangkat menghitung pangkat a^b dengan pengecekan khusus: jika pangkat b nol, mengembalikan 1; jika b negatif, mengembalikan -1 sebagai kode error; jika a lebih dari 100 atau b lebih dari 10, mengembalikan -2 sebagai batasan input; dan saat perhitungan dilakukan secara iteratif, fungsi checkedMultiply memastikan hasil perkalian masih dalam batas integer aman JavaScript, jika terjadi overflow (nilai melebihi batas aman), fungsi akan melempar error dan CariNilaiPangkat menangkapnya untuk mengembalikan kode error -3. Pendekatan ini menjamin hasil pangkat valid dan terhindar dari overflow tanpa menggunakan Math.pow, serta memberikan kode error yang jelas untuk berbagai kondisi tidak valid.

Index.html:

index.html > html

```html
<!DOCTYPE html>
<html>
<head>
  <title>Hitung Pangkat</title>
</head>
<body>
  <h1>Form Pangkat</h1>
  <input type="number" id="inputA" placeholder="Masukkan a"><br><br>
  <input type="number" id="inputB" placeholder="Masukkan b"><br><br>
  <button onclick="hitung()">Hitung Pangkat</button><br><br>
  <label id="output"></label>

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

Penjelasan:

halaman HTML sederhana yang menyediakan form untuk menghitung pangkat dari dua angka a dan b. Pengguna memasukkan nilai a dan b di dua kotak input angka, lalu menekan tombol Hitung Pangkat. Ketika tombol diklik, fungsi JavaScript hitung() dipanggil, yang mengambil nilai input, memanggil fungsi CariNilaiPangkat dari file main.js untuk menghitung a^b, lalu menampilkan hasilnya pada elemen label dengan id output. File main.js yang berisi logika perhitungan di-load melalui tag <script>, sehingga interaksi halaman dan kalkulasi pangkat dapat berjalan secara dinamis di browser.

Main.test.js:

```
JS main.test.js > ...
1   const { CariNilaiPangkat } = require('./main');
2
3   test('Pangkat 2^3 = 8', () => {
4     expect(CariNilaiPangkat(2, 3)).toBe(8);
5   });
6
7   test('Pangkat dengan b = 0', () => {
8     expect(CariNilaiPangkat(0, 0)).toBe(1);
9   });
10
11  test('Pangkat dengan b < 0', () => {
12    expect(CariNilaiPangkat(2, -1)).toBe(-1);
13  });
14
15  test('Pangkat dengan a > 100', () => {
16    expect(CariNilaiPangkat(101, 2)).toBe(-2);
17  });
18
19  test('Pangkat dengan b > 10', () => {
20    expect(CariNilaiPangkat(2, 11)).toBe(-2);
21  });
22
23  test('Pangkat overflow (9^30)', () => {
24    expect(CariNilaiPangkat(9, 30)).toBe(-3);
25  });
```

Penjelasan:
Fungsi ini diuji pada beberapa skenario untuk memastikan hasilnya sesuai ekspektasi:

- Test 1: Memastikan 2^3 menghasilkan 8.
- Test 2: Jika pangkat b = 0, hasilnya harus 1 (aturan matematika pangkat).
- Test 3: Jika pangkat negatif (b < 0), fungsi harus mengembalikan kode error -1.
- Test 4: Jika basis a lebih dari 100, fungsi harus mengembalikan kode error -2.
- Test 5: Jika pangkat b lebih dari 10, juga harus mengembalikan -2.
- Test 6: Jika terjadi overflow (misalnya 9^30 yang sangat besar), fungsi harus mendeteksinya dan mengembalikan kode error -3.

Test ini berguna untuk memastikan fungsi CariNilaiPangkat bekerja dengan benar pada kasus normal maupun batasan input dan error, sehingga program lebih andal dan dapat menangani kasus edge dengan tepat.

Output: