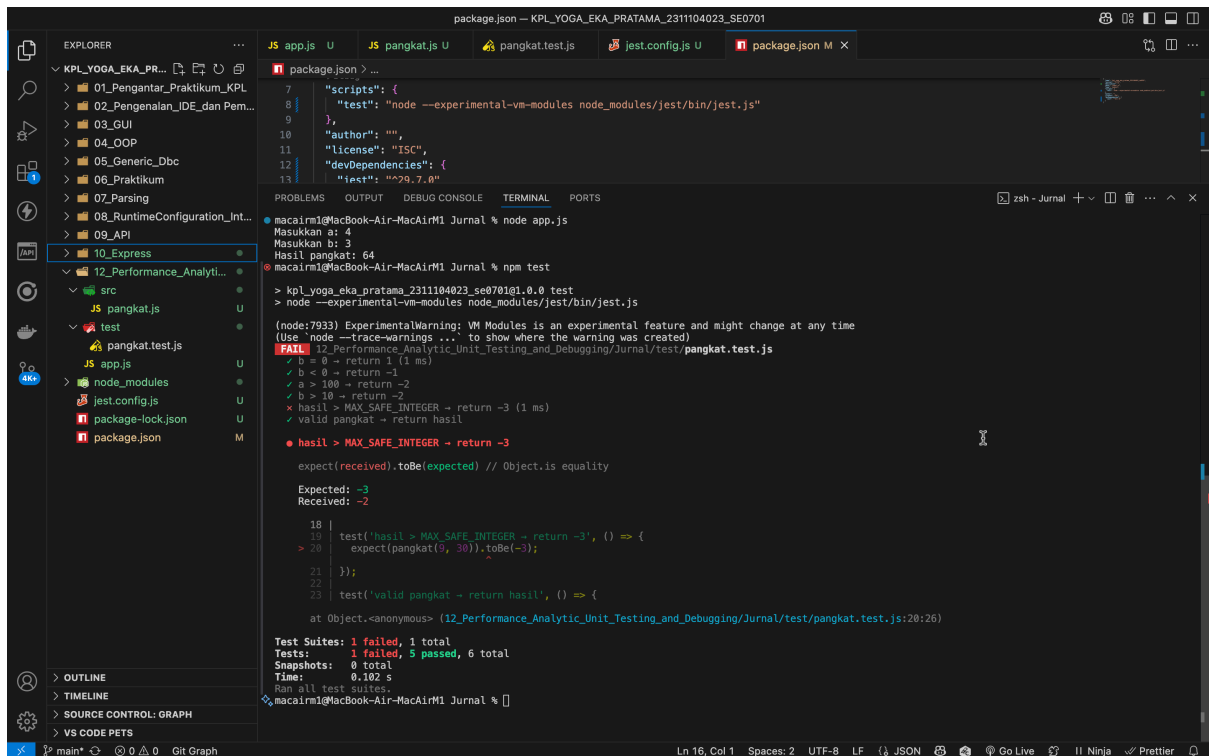


2311104023
Yoga Eka Pratama
SE0701

Jurnal 12



The screenshot shows a VS Code editor with a project named 'KPL_YOGA_EKA_PRATAMA_2311104023_SE0701'. The Explorer panel on the left shows the file structure, including a 'test' directory with 'pangkat.test.js'. The main editor displays the 'package.json' file with the following content:

```
{
  "scripts": {
    "test": "node --experimental-vm-modules node_modules/jest/bin/jest.js"
  },
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "jest": "^29.7.0"
  }
}
```

The terminal window at the bottom shows the execution of the test command. It displays the following output:

```
macairm1@MacBook-Air-MacAirM1 Jurnal % node app.js
Masukkan a: 4
Masukkan b: 64
Hasil pangkat: 64
macairm1@MacBook-Air-MacAirM1 Jurnal % npm test
> kpl_yoga_eka_pratama_2311104023_se0701@1.0.0 test
> node --experimental-vm-modules node_modules/jest/bin/jest.js
(node:7933) ExperimentalWarning: VM Modules is an experimental feature and might change at any time
(Use 'node --trace-warnings ...' to show where the warning was created)
FAIL 12_Performance_Analytic_Unit_Testing_and_Debugging/Jurnal/test/pangkat.test.js
  b = 0 -> return 1 (1 ms)
  ✓ b < 0 -> return -1
  ✓ a > 100 -> return -2
  ✓ b > 10 -> return -2
  ✗ hasil > MAX_SAFE_INTEGER -> return -3 (1 ms)
  ✓ valid pangkat -> return hasil
  ✗ hasil > MAX_SAFE_INTEGER -> return -3

  expect(received).toBe(expected) // Object.is equality
    Expected: -3
    Received: -2

  18 |
  19 |   test('hasil > MAX_SAFE_INTEGER -> return -3', () => {
  20 |     expect(pangkat(9, 30)).toBe(-3);
  21 |   });
  22 |
  23 |   test('valid pangkat -> return hasil', () => {
    at Object.<anonymous> (12_Performance_Analytic_Unit_Testing_and_Debugging/Jurnal/test/pangkat.test.js:20:26)

Test Suites: 1 failed, 1 total
Tests:       1 failed, 5 passed, 6 total
Snapshots:  0 total
Time:        0.102 s
Run all test suites
macairm1@MacBook-Air-MacAirM1 Jurnal %
```

Penjelasan :

Kode ini terdiri dari tiga bagian utama: fungsi pangkat, program input-output dengan readline, dan pengujian menggunakan Jest.

1. Fungsi pangkat(a, b)

Fungsi ini menghitung pangkat dari a pangkat b (abab) dengan ketentuan khusus:

- Jika $b = 0$, hasilnya 1 (aturan matematika dasar).
- Jika $b < 0$, fungsi mengembalikan -1 sebagai indikator input pangkat negatif tidak didukung.
- Jika $a > 100$ atau $b > 10$, fungsi mengembalikan -2 sebagai batasan untuk mencegah perhitungan besar dan lama.
- Selama perhitungan, jika hasil melebihi batas aman angka JavaScript (Number.MAX_SAFE_INTEGER), fungsi mengembalikan -3 menandakan overflow.
- Jika semua kondisi di atas tidak terpenuhi, fungsi mengembalikan hasil pangkat yang valid.

2. Program Input-Output menggunakan readline

Program menggunakan modul readline untuk mengambil input a dan b dari pengguna lewat terminal, kemudian menghitung dan menampilkan hasil pangkat menggunakan fungsi pangkat.

3. Unit Test dengan Jest

Kode juga berisi beberapa pengujian untuk memastikan fungsi pangkat bekerja sesuai harapan, antara lain:

- Mengembalikan 1 saat $b = 0$.
- Mengembalikan -1 jika $b < 0$.
- Mengembalikan -2 jika $a > 100$ atau $b > 10$.
- Mengembalikan -3 jika hasil pangkat melebihi batas aman JavaScript.
- Mengembalikan hasil pangkat yang benar untuk input valid.

Kode ini menunjukkan penerapan fungsi matematikadengan validasi input, interaksi pengguna, dan pengujian unit untuk menjaga kualitas kode.