

Journal Mandiri Modul 14

Nama: Satria Ramadhan

Nim: 2311104026

Kelas: SE-07-01

1. Source Code

a. Index.js

```
14_CleanCode > JM_Modul_14 > JS index.js > ...
1 // index.js
2
3 const { add, subtract, multiply, divide } = require("../mathOperations");
4
5 const numberA = 10;
6 const numberB = 5;
7
8 console.log("=== Operasi Matematika ===");
9 console.log(`Penjumlahan: ${add(numberA, numberB)}`);
10 console.log(`Pengurangan: ${subtract(numberA, numberB)}`);
11 console.log(`Perkalian: ${multiply(numberA, numberB)}`);
12 console.log(`Pembagian: ${divide(numberA, numberB)}`);
13
```

b. mathOperations.js

```
14_CleanCode > JM_Modul_14 > JS mathOperations.js > ...
1 // mathOperations.js
2
3 /**
4  * Menjumlahkan dua angka
5  * @param {number} number1
6  * @param {number} number2
7  * @returns {number}
8  */
9 function add(number1, number2) {
10   return number1 + number2;
11 }
12
13 /**
14  * Mengurangi dua angka
15  * @param {number} number1
16  * @param {number} number2
17  * @returns {number}
18  */
19 function subtract(number1, number2) {
20   return number1 - number2;
21 }
22
23 /**
24  * Mengalikan dua angka
25  * @param {number} number1
26  * @param {number} number2
27  * @returns {number}
28  */
29 function multiply(number1, number2) {
30   return number1 * number2;
31 }
32
33 /**
34  * Membagi dua angka
35  * @param {number} number1
36  * @param {number} number2
37  * @returns {number|string}
38  */
39 function divide(number1, number2) {
40   return number2 === 0 ? "Error: pembagi tidak boleh nol" : number1 / number2;
41 }
42
43 module.exports = {
44   add,
45   subtract,
46   multiply,
47   divide,
48 };
49
```

2. Output

```
> node index.js
=== Operasi Matematika ===
Penjumlahan: 15
Pengurangan: 5
Perkalian: 50
Pembagian: 2
```

3. Penjelasan

Tugas Jurnal Modul 14 ini menginstruksikan untuk melakukan refactoring kode dari tugas sebelumnya agar sesuai dengan standar penulisan kode yang baik. Program ini

memisahkan logika perhitungan ke dalam file `mathOperations.js` dan menamai variabel serta fungsi menggunakan camelCase sebagai konvensi standar. Setiap fungsi diberi komentar deskriptif yang menjelaskan input dan output, sementara kode utama (`index.js`) memanggil fungsi-fungsi tersebut dengan variabel yang jelas dan rapi. Struktur indentasi, whitespace, serta deklarasi variabel juga telah diperbaiki agar mudah dibaca dan sesuai praktik penulisan profesional, menghasilkan program yang bersih, terstruktur, dan mudah dipelihara.