

Nama : Rayhan Sandika Ardaffa
NIM : 20240040187
Kelas : TI24A
Mata Kuliah : Pemrograman Berbasis Platform / Sesi 5

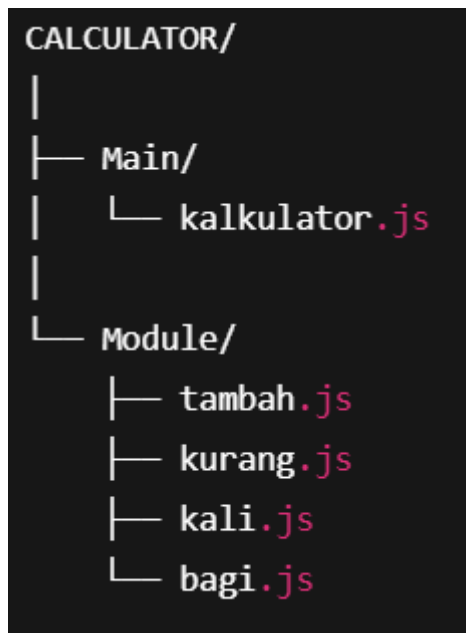
i. TUJUAN PROYEK

Tujuan dari proyek ini adalah membuat sebuah program kalkulator sederhana berbasis *Node.js* yang dapat melakukan operasi aritmatika dasar seperti penjumlahan, pengurangan, perkalian, dan pembagian. Selain itu, proyek ini juga bertujuan untuk menerapkan konsep modularisasi, yaitu memisahkan kode ke dalam beberapa file (module) agar lebih rapi, mudah dikelola, dan dapat digunakan kembali.

ii. LANGKAH – LANGKAH Pengerjaan

a) Membuat Struktur Folder

Struktur proyek dibuat seperti berikut:



Folder Main digunakan untuk menyimpan file utama (*kalkulator.js*) yang menjalankan program.

Folder Module berisi file-file fungsi untuk setiap operasi matematika.

b) Membuat File Module

Setiap operasi matematika dibuat di file terpisah menggunakan *module.exports*

Contohnya pada file *tambah.js*:

```
Module > JS tambahjs > ...
1  const pertambahan = (a,b) => {
2      const hasil = a + b; // operasi penjumlahan
3      return `Hasil pertambahan ${a} dan ${b} adalah ${hasil}` ;
4  };
5
6  module.exports = {pertambahan}; // ekspor fungsi pertambahan agar bisa digunakan di file lain
```

Modul ini berisi satu fungsi *pertambahan()* yang menerima dua parameter (*a* dan *b*), lalu mengembalikan hasilnya.

Cara ini membuat kode lebih terorganisir dan setiap operasi bisa dikelola secara terpisah.

c) Membuat Program Utama

Pada file *kalkulator.js*, dibuat logika utama menggunakan *readline* untuk menerima input dari pengguna.

Kode utamanya menampilkan menu, meminta pilihan operasi, lalu memanggil modul sesuai pilihan.

Contoh cuplikan kodenya:

```
else if (opsi == '1') { // jika pilihan 1
    rl.question('Masukkan angka pertama: ', (jawaban1) => { // meminta input angka pertama
        a = parseFloat(jawaban1); //ubah input string menjadi angka desimal
        rl.question('Masukkan angka kedua: ', (jawaban2) => { // meminta input angka kedua
            b = parseFloat(jawaban2);
            const { pertambahan } = require('../Module/tambah'); // import fungsi pertambahan
            console.log(pertambahan(a, b)); // panggil fungsi pertambahan dan tampilkan hasilnya
            rl.close(); // tutup readline interface
        });
    });
}
```

Fungsi *require()* digunakan untuk mengimpor modul dari folder lain.

Dengan cara ini, program utama tidak perlu menulis ulang rumus pertambahan, cukup memanggil fungsi dari modul.

d) Menangani Input Tidak Valid

Program menggunakan kondisi:

```
rl.question('Masukkan pilihan (1/2/3/4): ', (jawaban) => {
    opsi = jawaban;
    if (!['1', '2', '3', '4'].includes(opsi)) { // bila input bukan 1,2,3,4
        console.log("Pilihan tidak valid. Silakan pilih antara 1, 2, 3, atau 4.");
        menu(); // tampilkan menu lagi
        pilihMenu(); // panggil fungsi pilihMenu lagi agar bisa memilih ulang
    }
}
```

Bagian ini memastikan bahwa jika pengguna memasukkan pilihan selain 1–4, maka program akan menampilkan pesan error dan meminta input ulang tanpa harus dijalankan kembali.

iii. IMPLEMENTASI MODULARISASI

Modularisasi diimplementasikan dengan cara:

- Memisahkan fungsi perhitungan ke dalam file terpisah di folder Module.
- Mengimpor modul ke file utama menggunakan *require()*.
- Memanggil fungsi dari modul saat pengguna memilih operasi tertentu.

Keuntungan modularisasi:

- Kode lebih terstruktur dan mudah dibaca.
- Memudahkan debugging karena setiap fungsi berada di file sendiri.
- Fungsi bisa digunakan ulang di proyek lain tanpa perlu menulis ulang kode.

iv. HASIL PENGUJIAN

Contoh hasil saat menjalankan program:

```
PS C:\Tugas\Semester 3\Pemrograman Berbasis Platform\Sesi 5\Tugas_Sesi_5\calculator\Main> node kalkulator
Pilih operasi matematika:
1. Pertambahan (+)
2. Pengurangan (-)
3. Perkalian (*)
4. Pembagian (/)
Masukkan pilihan (1/2/3/4): 5
Pilihan tidak valid. Silakan pilih antara 1, 2, 3, atau 4.
Pilih operasi matematika:
1. Pertambahan (+)
2. Pengurangan (-)
3. Perkalian (*)
4. Pembagian (/)
Masukkan pilihan (1/2/3/4):
```

```
Pilih operasi matematika:
1. Pertambahan (+)
2. Pengurangan (-)
3. Perkalian (*)
4. Pembagian (/)
Masukkan pilihan (1/2/3/4): 1
Masukkan angka pertama: 3
Masukkan angka kedua: 2
Hasil pertambahan 3 dan 2 adalah 5
PS C:\Tugas\Semester 3\Pemrograman Berbasis Platform\Sesi 5\Tugas_Sesi_5\calculator\Main>
```

v. KESIMPULAN

Dari proyek ini dapat disimpulkan bahwa modularisasi sangat membantu dalam pengembangan aplikasi karena kode menjadi lebih mudah dikelola dan dipelihara.

Konsep pemisahan antara logika utama dan fungsi perhitungan membuat program kalkulator lebih rapi, efisien, dan fleksibel untuk dikembangkan lebih lanjut.