

## Membuat kalkulator sederhana

### SC code

```
import 'dart:io';

void main() {
  print("=== Aplikasi Kalkulator Sederhana ===");

  // Input bilangan pertama
  stdout.write("Masukkan bilangan pertama: ");
  double num1 = double.parse(stdin.readLineSync()!);

  // Input operator
  stdout.write("Masukkan operator (+, -, *, /): ");
  String operator = stdin.readLineSync()!;

  // Input bilangan kedua
  stdout.write("Masukkan bilangan kedua: ");
  double num2 = double.parse(stdin.readLineSync()!);

  double hasil;

  // Melakukan operasi berdasarkan input operator
  switch (operator) {
    case '+':
      hasil = num1 + num2;
      break;
    case '-':
      hasil = num1 - num2;
      break;
    case '*':
      hasil = num1 * num2;
      break;
    case '/':
      if (num2 != 0) {
        hasil = num1 / num2;
      } else {
        print("Error: Pembagian dengan 0 tidak valid.");
        return;
      }
      break;
    default:
      print("Operator tidak valid.");
      return;
  }

  print("Hasil: $num1 $operator $num2 = $hasil");
}
```

## Penjelasan Kode

### 1. Import library

```
import 'dart:io';
```

Kode ini digunakan untuk mengimpor library `dart:io`, yang memungkinkan aplikasi melakukan operasi input dan output (I/O) melalui konsol

### 2. `void main() {`

Fungsi `main()` adalah titik awal eksekusi program di Dart. Semua kode yang ada di dalam fungsi `main()` akan dieksekusi ketika aplikasi dijalankan

### 3. `print("=== Aplikasi Kalkulator Sederhana ===");`

menampilkan judul dari pemrograman tersebut

### 4. Mengambil input dari pengguna

- `stdout.write()`: menampilkan pesan ke konsol tanpa berpindah ke baris baru (berbeda dengan `print()`).
- `stdin.readLineSync()`: membaca input yang dimasukkan oleh pengguna dari konsol sebagai `String`.
- `double.parse()`: digunakan untuk mengonversi input `String` menjadi tipe data `double` (bilangan desimal). Tanda seru (!) di akhir `readLineSync()` memastikan bahwa input tidak null (kosong).

### 5. `stdout.write("Masukkan operator (+, -, *, /): ");`

```
String operator = stdin.readLineSync()!;
```

Untuk memasukkan operator matematika (+, -, \*, atau /). Operator ini disimpan dalam variabel bertipe `String`.

### 6. `stdout.write("Masukkan bilangan kedua: ");`

```
double num2 = double.parse(stdin.readLineSync()!);
```

Memasukkan bilangan kedua yang akan digunakan dalam operasi matematika

### 7. `double hasil;`

Variabel `hasil` dideklarasikan sebagai tipe `double` untuk menyimpan hasil perhitungan operasi matematika.

### 8. Melakukan operasi matematika

```
switch (operator) {  
  case '+':  
    hasil = num1 + num2;  
    break;  
  case '-':  
    hasil = num1 - num2;  
    break;  
}
```

```

case '*':
    hasil = num1 * num2;
    break;
case '/':
    if (num2 != 0) {
        hasil = num1 / num2;
    } else {
        print("Error: Pembagian dengan 0 tidak valid.");
        return;
    }
    break;
default:
    print("Operator tidak valid.");
    return;
}

```

- Switch-case digunakan untuk mengevaluasi nilai dari variabel operator. Berdasarkan operator yang dipilih (+, -, \*, atau /), program akan melakukan operasi matematika yang sesuai.
  - Untuk setiap case:
    - Case '+': Dilakukan operasi penjumlahan  $\text{num1} + \text{num2}$ .
    - Case '-': Dilakukan operasi pengurangan  $\text{num1} - \text{num2}$ .
    - Case '\*': Dilakukan operasi perkalian  $\text{num1} * \text{num2}$ .
    - Case '/': Dilakukan operasi pembagian  $\text{num1} / \text{num2}$ . Sebelum operasi pembagian, ada pengecekan apakah  $\text{num2}$  bernilai 0. Jika  $\text{num2} == 0$ , aplikasi mengeluarkan pesan error karena pembagian dengan nol tidak valid, lalu keluar dari program dengan return.
  - default: Jika pengguna memasukkan operator yang tidak valid (misalnya selain +, -, \*, atau /), program akan menampilkan pesan kesalahan dan keluar dari fungsi main().
9. `print("Hasil: $num1 $operator $num2 = $hasil");`
- Setelah operasi matematika selesai, hasil perhitungan ditampilkan dengan menggunakan `print()`. String interpolation (`$num1`, `$operator`, dan `$hasil`) digunakan untuk menyisipkan nilai variabel ke dalam string

Contoh hasil dari aplikasinya

```

=== Aplikasi Kalkulator Sederhana ===
Masukkan bilangan pertama: 3
Masukkan operator (+, -, *, /): *
Masukkan bilangan kedua: 20
Hasil: 3.0 * 20.0 = 60.0

```