

Nama : Dwita Isma Aprilia  
NIM : H1D022034  
Shift : B Shift Asli : B

## PRAKTIKUM MOBILE PERTEMUAN 1

### **Implementasi 1 : Kelas dan Objek**

Program ini mengilustrasikan cara membuat dan mengelola objek dengan kelas Mahasiswa, menampilkan detail objek, dan mengecek status kelulusan berdasarkan nilai akhir.

1. Definisi Kelas Mahasiswa

Source code :

```
class Mahasiswa {
```

2. Property atau Atribut Kelas Mahasiswa

Source code :

```
String nama;
```

```
int umur;
```

```
double nilaiAkhir;
```

3. Konstruktor

Source code :

```
Mahasiswa(this.nama, this.umur, this.nilaiAkhir);
```

4. Fungsi untuk Menampilkan Detail Mahasiswa

Source code :

```
void tampilkanDetail() {
```

```
    print('Nama: $nama');
```

```
    print('Umur: $umur tahun');
```

```
    print('Nilai Akhir: $nilaiAkhir');
```

```
}
```

5. Fungsi untuk Memeriksa Kelulusan

Source code :

```
void cekKelulusan() {
```

```
    if (nilaiAkhir >= 60.0) {
```

```
        print('$nama lulus');
```

```
    } else {
```

```
        print('$nama tidak lulus');
```

```
    }
```

```
}
```

6. Fungsi Utama (main)

Source code :

```
void main() {
```

```
    Mahasiswa mhs1 = Mahasiswa('Dwita', 20, 75.5);
```

- Mahasiswa mhs2 = Mahasiswa('Isma', 22, 55.0);
7. Menampilkan Detail dan Mengecek Kelulusan

Source code :

```
mhs1.tampilkanDetail();
mhs1.cekKelulusan();

print(""); // Memberikan jarak antara dua output

mhs2.tampilkanDetail();
mhs2.cekKelulusan();
}
```

### **Implementasi 2 : Operasi Matematika**

Program ini mengimplementasikan sebuah fungsi untuk menghitung rata-rata dari daftar nilai.

1. Fungsi calculateAverage

Source code :

```
double calculateAverage(List<int> numbers) {
    int sum = 0;
    for (int number in numbers) {
        sum += number;
    }
    return sum / numbers.length;
}
```

2. Fungsi main

Source code :

```
void main() {
    List<int> scores = [85, 90, 78, 92, 88];

    double average = calculateAverage(scores);
    print("Rata-rata nilai: $average");
}
```

### **Implementasi 3 : Asynchronous**

Program ini mensimulasikan pengambilan data dari server dengan menggunakan mekanisme asynchronous. Dalam hal ini, program tidak berhenti sepenuhnya saat menunggu data. Setelah data diterima, program melanjutkan untuk menampilkan hasilnya.

1. Fungsi fetchData

Source code:

```
Future<String> fetchData() async {
```

```
    await Future.delayed(Duration(seconds: 2));  
    return "Data telah diterima!";  
}
```

## 2. Fungsi main

Source code :

```
void main() async {  
    print("Mengambil data...");
```

```
    String result = await fetchData();
```

```
    print(result);  
}
```