

# LAPORAN RESPONSI PEMROGRAMAN BERORIENTASI OBJEK

## Pertemuan 1

### 1. Alur Program

1. Program dimulai dari UjiPengguna.main()
2. Membuat objek Pengguna dengan data: umur=25, berat=60, tinggi=170
3. Memanggil method tampilInfo() pada objek Pengguna
4. tampilInfo() memanggil berbagai method lainnya secara berantai:
  - hitungBMI() menghitung nilai BMI
  - tampilKategori() menentukan kategori berdasarkan BMI
  - hitungKalori() menghitung kebutuhan kalori
5. Menampilkan demonstrasi penggunaan konstanta final

### 2. Fungsi yang digunakan

#### 2.1 Kelas Pengguna

Atribut/State:

java

```
int umur;           // Menyimpan umur pengguna dalam tahun
double berat;       // Menyimpan berat badan dalam kg
double tinggi;      // Menyimpan tinggi badan dalam cm
final double faktorAktivitas = 1.2; // Konstanta untuk perhitungan kalori
```

Method/Fungsi:

#### 1.hitungBMI() - Fungsi

java

```
public double hitungBMI() {
    double tinggiMeter = tinggi / 100.0; // Konversi cm ke meter
    return berat / (tinggiMeter * tinggiMeter); // Rumus BMI
}
```

- Tipe: Fungsi (mengembalikan nilai)
- Tujuan: Menghitung Body Mass Index (BMI)
- Return: Nilai BMI (double)

#### 2.tampilKategori() - Prosedur

java

```
public void tampilKategori() {
    double bmi = hitungBMI(); // Memanggil fungsi hitungBMI()

    if (bmi < 18.5) {
        System.out.println("Kategori: Kurus");
    }
}
```

```

} else if (bmi < 25) {
    System.out.println("Kategori: Normal");
} else {
    System.out.println("Kategori: Obesitas");
}
}



- Tipe: Prosedur (tidak mengembalikan nilai)
- Tujuan: Menentukan kategori BMI berdasarkan nilai
- Output: Menampilkan kategori ke konsol

```

### 3.hitungKalori() - Fungsi

java

```

public double hitungKalori() {
    return berat * faktorAktivitas * 10; // Rumus kalori
}



- Tipe: Fungsi (mengembalikan nilai)
- Tujuan: Menghitung kebutuhan kalori harian
- Return: Jumlah kalori (double)

```

### 4.tampilInfo() - Prosedur

java

```

public void tampilInfo() {
    System.out.println("== Informasi Kesehatan Pengguna ==");
    System.out.println("Umur: " + umur + " tahun");
    System.out.println("Berat: " + berat + " kg");
    System.out.println("Tinggi: " + tinggi + " cm");
    System.out.printf("BMI: %.2f\n", hitungBMI()); // Format 2 desimal
    tampilKategori();
    System.out.println("Kebutuhan Kalori: " + (int)hitungKalori() + " kal/hari");
    System.out.println("=====\\n");
}

```

- Tipe: Prosedur (tidak mengembalikan nilai)
- Tujuan: Menampilkan semua informasi kesehatan pengguna
- Fitur: Menggunakan printf() untuk format angka

## 2.2 Kelas UjiPengguna (Main Class)

java

```

public class UjiPengguna {
    public static void main(String[] args) {
        // Membuat objek dan mengisi data
        Pengguna p = new Pengguna();
        p.umur = 25;
        p.berat = 60;
        p.tinggi = 170;
    }
}

```

```

// Menampilkan informasi kesehatan
p.tampilInfo();

// Demonstrasi konstanta final
System.out.println("Nilai faktor aktivitas: " + p.faktorAktivitas);
System.out.println("Faktor aktivitas bersifat final (konstan)");
System.out.println("Artinya nilai ini tidak bisa diubah setelah diinisialisasi\n");

// Contoh kode yang akan menyebabkan error
System.out.println("Jika kita mencoba menulis:");
System.out.println("p.faktorAktivitas = 1.5;\n");

System.out.println("Maka compiler akan menolak dengan error:");
System.out.println("cannot assign a value to final variable faktorAktivitas");
}
}

```

### 3. Output

```

==== Informasi Kesehatan Pengguna ====
Umur: 25 tahun
Berat: 60.0 kg
Tinggi: 170.0 cm
BMI: 20.76
Kategori: Normal
Kebutuhan Kalori: 720 kal/hari
=====

Nilai faktor aktivitas: 1.2
Faktor aktivitas bersifat **final** (konstan).
Mencoba mengubahnya (misalnya: p.faktorAktivitas = 1.5;) akan menyebabkan Error Kompilasi.

```