

KUSUMA WEB ACADEMY

# DASAR-DASAR JALUR JAVA

Modul 09: Fungsi Modular - Method & Parameter

Hak Cipta © 2026 Kusuma Web. All Rights Reserved.

## Apa itu Method?

- ▶ Blok kode terorganisir yang diberi nama dan hanya akan berjalan ketika dipanggil secara eksplisit.
- ▶ **Tujuan utama:** Mencegah duplikasi kode (\*Don't Repeat Yourself\* / DRY) dan merapikan kompleksitas alur logika aplikasi.

## Struktur Anatomi Deklarasi

### Struktur Umum Sintaks

```
modifier tipeReturn  
nama(parameter) {}
```

- ▶ **Modifier:** Menentukan hak akses (contoh: `public`, `private`, atau `static`).
- ▶ **Tipe Return:** Tipe data hasil keluaran method. Gunakan `void` jika tidak ada hasil kembalian.
- ▶ **Parameter:** Input variabel opsional yang diproses di dalam method.

# Metode Tanpa Balikan (Void) vs Metode Dengan Balikan

## 1. Method Void (Prosedur)

- ▶ Melakukan aksi tertentu secara langsung tanpa mengembalikan nilai apa pun ke pemanggilnya.

```
// Tidak mengembalikan apa pun
public void sapaUser(String nama) {
    System.out.println("Halo, " + nama)
    ;
}
```

## 2. Method Non-Void (Return Value)

- ▶ Mengolah data dan wajib mengembalikan hasil pengolahan menggunakan kata kunci return.

```
// Mengembalikan data bertipe int
public int hitungLuasPersegi(int sisi)
{
    int luas = sisi * sisi;
    return luas; // Wajib bertipe int
}
```

## Cara Pemanggilan

Method void langsung dipanggil: `sapaUser("Budi");`

Method non-void ditampung variabel: `int hasil = hitungLuasPersegi(10);`

# Cara Kerja Parameter: Pass-by-Value di Java

## Konsep Pass-by-Value

- ▶ Java **\*\*selalu\*\*** menyalin nilai variabel (salinan value mentah atau salinan alamat referensi objek) ke variabel parameter lokal baru di dalam method.
- ▶ Manipulasi variabel primitif di dalam method **tidak akan pernah** mengubah variabel aslinya di luar method.

## Visualisasi Perilaku Kode

```
public class DemoValue {  
    static void ubahAngka(int x) {  
        x = 99; // Hanya merubah salinan  
               x lokal  
    }  
  
    public static void main(String[] args  
        ) {  
        int angka = 10;  
        ubahAngka(angka);  
        System.out.println(angka); //  
            Tetap 10!  
    }  
}
```

## Apa itu Method Overloading?

- ▶ Kondisi di mana suatu class memiliki beberapa method dengan **nama yang sama**, namun memiliki **parameter berbeda**.
- ▶ Perbedaan parameter diidentifikasi dari:
  1. Jumlah total parameter.
  2. Tipe data dari parameter.
  3. Urutan tipe data parameter.
- ▶ Tipe return tidak bisa dijadikan pembeda tunggal dalam overloading.

## Implementasi Overloading

```
public class Kalkulator {  
    // Overload 1: Dua parameter int  
    int tambah(int a, int b) {  
        return a + b;  
    }  
  
    // Overload 2: Tiga parameter int  
    int tambah(int a, int b, int c) {  
        return a + b + c;  
    }  
  
    // Overload 3: Dua parameter double  
    double tambah(double a, double b) {  
        return a + b;  
    }  
}
```