

# Laporan Pratikum Dasar Pemrograman

## Pertemuan – 13 : Jobsheet 13 Fungsi 1



Nama : Queenadhynar Azarine D.A

NIM : 2341760109

Prodi : D-IV Sistem Informasi Bisnis

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2023/2024**

## Percobaan 1: Membuat Fungsi Tanpa Parameter

1. Buat class baru, dan simpan file tersebut dengan nama TerimaKasih\_NoAbsen.java.
2. Buat fungsi UcapanTerimaKasih di dalam class tersebut.

```
import java.util.Scanner;

public class TerimaKasih_24 {
    public static void UcapanTerimaKasih () {
        System.out.println("Thank you for being the best teacher in the world.\n"+
            "You inspired in me a love for learning and made me feel like I could ask you anything.");
    }
}
```

3. Buat fungsi main di dalam class tersebut, dan eksekusi fungsi UcapanTerimaKasih dari dalam fungsi main.

```
public static void main(String[] args) {
    UcapanTerimaKasih();
}
}
```

Hasil Run :

```
Thank you for being the best teacher in the world.
You inspired in me a love for learning and made me
feel like I could ask you anything.
```

## Pertanyaan!

1. Apakah fungsi tanpa parameter selalu digunakan untuk fungsi “void” saja?
  - Tidak, fungsi tanpa parameter tidak selalu digunakan hanya untuk fungsi void (fungsi yang tidak mengembalikan nilai). Meskipun seringkali fungsi tanpa parameter digunakan untuk fungsi void, tetapi Anda juga dapat memiliki fungsi tanpa parameter yang mengembalikan nilai.
2. Apakah bisa kalimat “Thank you for.....dst” dituliskan tanpa fungsi UcapanTerimaKasih? modifikasi kode program tersebut untuk dapat menampilkan suatu kalimat tanpa dan dengan menggunakan fungsi!
  - Dengan menggunakan fungsi :

```

1  import java.util.Scanner;
2
3  public class TerimaKasih_24 {
4      public static void UcapanTerimaKasih () {
5          System.out.println("Thank you for being the best teacher in the world.\n"+
6              "You inspired in me a love for learning and made me feel like I could ask you anything.");
7      }
8
9      public static void main(String[] args) {
10         UcapanTerimaKasih();
11     }
12 }
13

```

➤ Hasil run :

```

Thank you for being the best teacher in the world.
You inspired in me a love for learning and made me feel
like I could ask you anything.

```

- Penjelasan jika menggunakan Fungsi UcapanTerimaKasih digunakan untuk mengorganisir dan memisahkan tugas spesifik, dan itu dapat digunakan kembali di tempat lain dalam program jika diperlukan.
- Tanpa menggunakan fungsi :

```

1  import java.util.Scanner;
2
3  public class TerimaKasih_24_1 {
4      public static void main(String[] args) {
5          System.out.println("Thank you for being the best teacher in the world.\n" +
6              "You inspired in me a love for learning and made me feel like I could ask you anything.");
7      }
8  }
9

```

➤ Hasil run :

```

Thank you for being the best teacher in the world.
You inspired in me a love for learning and made me feel
like I could ask you anything.

```

3. Apakah keuntungan menggunakan fungsi di dalam program?

- **Modularitas:** Memecah program menjadi modul-modul kecil untuk mempermudah pengembangan dan pemahaman kode.
- **Reusabilitas:** Fungsi dapat digunakan kembali di berbagai bagian program, mengurangi pengkodean ulang.
- **Pemeliharaan yang Mudah:** Memodifikasi atau memperbaiki program lebih sederhana karena fokus pada perubahan fungsi tertentu.
- **Penyederhanaan Kode:** Menyederhanakan kode dengan mengelompokkan pernyataan ke dalam unit fungsional.

- **Abstraksi:** Memungkinkan penggunaan fungsionalitas tanpa perlu memahami detail implementasinya.
- **Testing yang Lebih Mudah:** Pengujian dapat dilakukan secara terpisah untuk setiap fungsi.
- **Efisiensi Kode:** Menulis kode yang lebih efisien dengan kemampuan menggunakan kembali logika tertentu.
- Penggunaan fungsi membantu meningkatkan kualitas dan keterbacaan kode, serta mempermudah pengembangan dan pemeliharaan program.

## Percobaan 2: Membuat Fungsi Dengan Parameter

1. Buat fungsi UcapanTambahan dengan sebuah parameter bertipe String di dalam class TerimaKasih\_NoAbsen!

```
}  
public static void UcapanTambahan (String ucapan){  
    System.out.println(ucapan);  
}
```

2. Eksekusi fungsi UcapanTambahan dari dalam fungsi main.

```
public static void main(String[] args) {  
    UcapanTerimaKasih();  
    String ucapan = "Terima Kasih Pak.. Bu.. Semoga Sehat Selalu";  
    UcapanTambahan(ucapan);  
}
```

## Pertanyaan!

1. Apakah kegunaan suatu parameter di dalam fungsi?
  - Pengiriman Data: Parameter memungkinkan Anda mengirim data ke fungsi saat memanggilnya.
  - Memasukkan Nilai: Parameter memungkinkan fungsi menerima nilai yang dapat digunakan di dalamnya.
  - Fleksibilitas: Parameter memberikan fleksibilitas, memungkinkan fungsi digunakan dengan nilai yang berbeda setiap kali dipanggil.
  - Reusabilitas: Fungsi dengan parameter dapat digunakan kembali dengan berbagai data input.
2. Apakah parameter sama dengan variabel? jelaskan!
  - Ya, tapi Tidak Selalu Sama: Parameter adalah variabel yang digunakan dalam definisi fungsi untuk menerima nilai yang diberikan saat pemanggilan fungsi. Jadi, dalam hal ini, parameter adalah variabel. Namun, secara umum, variabel dapat merujuk pada data yang dipegang di berbagai konteks dalam program, sedangkan parameter adalah variabel yang terkait dengan suatu fungsi.
3. Apakah parameter di dalam Bahasa pemrograman Java hanya digunakan untuk melewati data input saja? Bagaimana dengan data output?
  - Data Input dan Output: Parameter di Java dapat digunakan untuk melewati data input ke fungsi, tetapi juga dapat digunakan untuk mengembalikan data output dari fungsi. Dalam Java, Anda dapat menggunakan tipe data return untuk mengembalikan nilai dari suatu fungsi. Jadi, parameter dapat digunakan untuk membawa data ke dalam fungsi dan mengembalikan data dari fungsi.

### Percobaan 3: Membuat Fungsi dengan Nilai Kembalian

1. Buat class baru, dan menyimpan file tersebut dengan nama Ucapan\_NoAbsen.java.
2. Buat fungsi PenerimaUcapan di dalam class tersebut yang mengembalikan nilai namaOrang (String) dan parameter masukan nama (String).

```
import java.util.Scanner;
public class Ucapan_24 {
    public static String PenerimaUcapan(){
        Scanner sc = new Scanner(System.in);
        System.out.print("Tuliskan NAMA orang yang ingin Anda beri ucapan: ");
        String namaOrang = sc.nextLine();
        sc.close();
        return namaOrang;
    }
}
```

3. Buat fungsi main di dalam class tersebut, dan eksekusi fungsi PenerimaUcapan dari dalam fungsi main.

```
public static void main(String[] args) {
    String nama = PenerimaUcapan();
    System.out.println("Thank you " +nama+"\nMay the force be with you.");
}
}
```

Hasil run :

```
Tuliskan NAMA orang yang ingin Anda beri ucapan: Dhynar
Thank you Dhynar
May the force be with you.
```

### Pertanyaan!

1. Jelaskan mengenai, kapan suatu fungsi membutuhkan nilai kembalian (return)!
  - **Mengembalikan Hasil Perhitungan:**  
Fungsi seringkali digunakan untuk melakukan perhitungan atau operasi tertentu. Nilai hasil perhitungan tersebut kemudian dikembalikan menggunakan pernyataan return.
  - **Mengembalikan Status atau Kode Error:**  
Dalam beberapa kasus, fungsi dapat mengembalikan nilai yang menunjukkan status operasi atau kode kesalahan. Ini memungkinkan pemanggil fungsi untuk mengetahui apakah operasi tersebut berhasil atau gagal.
  - **Mengembalikan Objek atau Struktur Data:**  
Fungsi juga dapat mengembalikan objek atau struktur data yang kompleks. Contoh, sebuah fungsi dapat membentuk dan mengembalikan objek baru.

- **Ketika Anda Membutuhkan Nilai untuk Digunakan Lebih Lanjut:**  
Jika Anda ingin menggunakan hasil dari suatu fungsi di tempat pemanggilan, Anda membutuhkan nilai kembalian untuk menyimpan hasil tersebut dalam suatu variabel.
  - Jadi, fungsi memerlukan nilai kembalian ketika Anda ingin mengirimkan data hasil atau informasi khusus dari fungsi tersebut ke bagian lain dari program Anda. Jika fungsi hanya digunakan untuk melakukan tindakan (tanpa hasil yang perlu dikembalikan), Anda dapat menggunakan tipe data void pada definisi fungsi.
2. Apakah bisa ditambahkan statement `System.out.println` di dalam fungsi dengan nilai kembalian? Apa pengaruh nya?
- Contoh jika di beri `System.out.println` di dalam fungsi dengan nilai kembalian  
Kode Program ;

```
1  import java.util.Scanner;
2  public class Ucapan_24 {
3      public static String PenerimaUcapan() {
4          Scanner sc = new Scanner(System.in);
5          System.out.print("Tuliskan NAMA orang yang ingin Anda beri ucapan: ");
6          String namaOrang = sc.nextLine();
7          System.out.print("Ucapan diterima untuk " + namaOrang);
8          sc.close();
9          return namaOrang;
10     }
11     public static void main(String[] args) {
12         String nama = PenerimaUcapan();
13         System.out.println(" Thank you "+nama+"\nMay the force be with you ");
14     }
15 }
16 }
17 }
```

Hasil Run :

```
Tuliskan NAMA orang yang ingin Anda beri ucapan: Dhynar
Ucapan diterima untuk Dhynar Thank you Dhynar
May the force be with you.
```

3. Apakah bisa fungsi tanpa nilai kembalian di dalam fungsi main dipanggil tanpa dilewatkan ke variabel? seperti pada percobaan 1? Jelaskan!
- Ya, fungsi tanpa nilai kembalian (void) dapat dipanggil di dalam fungsi main tanpa harus menyimpan nilainya dalam variabel. Dalam contoh yang diberikan memanggil fungsi `PenerimaUcapan` tanpa menyimpan nilai kembaliannya dalam variabel. Pemanggilan ini berguna jika Anda hanya ingin menjalankan fungsi tersebut tanpa memerlukan nilai kembalian untuk digunakan di bagian lain dari program.

- Dalam kasus ini, fungsi `PenerimaUcapan` digunakan untuk menerima input dari pengguna dan mencetak pesan terima kasih. Jadi, tidak ada kebutuhan untuk menyimpan nilai kembalian dalam variabel, dan pemanggilan langsung tanpa penyimpanan hasilnya adalah suatu pendekatan yang umum.
- Contoh Fungsi tanpa nilai kembalian di dalam fungsi `main` dipanggil tanpa dilewatkan ke variabel

Kode Program :

```
1  import java.util.Scanner;
2  public class Ucapan_24 {
3      public static String PenerimaUcapan(){
4          Scanner sc = new Scanner(System.in);
5          System.out.print("Tuliskan NAMA orang yang ingin Anda beri ucapan: ");
6          String namaOrang = sc.nextLine();
7          sc.close();
8          return namaOrang;
9      }
10
11     public static void main(String[] args) {
12         // Memanggil fungsi tanpa menyimpan nilai kembalian dalam variabel
13         System.out.println("Thank you " + PenerimaUcapan() + "\nMay the force be with you.");
14     }
15 }
16
```

Hasil Run :

```
Tuliskan NAMA orang yang ingin Anda beri ucapan: Dhynar
Thank you Dhynar
May the force be with you.
```



#### Percobaan 4: Fungsi yang dapat meng-Calling Fungsi Lain

1. Buat class baru, simpan file tersebut dengan nama nama UcapanTerimaKasih\_NoAbsen.java. Pada class ini, kita akan mencoba menggabungkan fungsi-fungsi yang sudah kita buat sebelumnya pada class TerimaKasih dan Ucapan.
2. Buatlah fungsi PenerimaUcapan, dimana pada class tersebut akan mengembalikan nilai namaOrang (String) dan parameter masukan nama (String).

```
public static String PenerimaUcapan() {  
    Scanner sc = new Scanner(System.in);  
    System.out.print("Tuliskan NAMA orang yang ingin Anda beri ucapan: ");  
    String namaOrang = sc.nextLine();  
    sc.close();  
    return namaOrang;  
}
```

3. Buatlah fungsi UcapanTerimaKasih di dalam class tersebut yang memanggil fungsi PenerimaUcapan

```
public static void UcapanTerimaKasih() {  
    String nama = PenerimaUcapan();  
    System.out.println("Thank you " +nama+ " for being the best teacher in the world.\n" +  
        "You inspired in me a love for learning and made me feel like I could ask you anything.");  
}
```

4. Buat fungsi main di dalam class tersebut, dan eksekusi fungsi UcapanTerimaKasih dari dalam fungsi main.

```
public static void main(String[] args) {  
    UcapanTerimaKasih();  
}
```

Hasil Run :

```
Tuliskan NAMA orang yang ingin Anda beri ucapan: Dhynar  
Thank you Dhynar for being the best teacher in the world.  
You inspired in me a love for learning and made me feel like I could ask you anything  
.
```

## Pertanyaan!

1. Berdasarkan uji coba pada percobaan 4, manakah fungsi yang akan eksekusi pertama kali? Jelaskan!
  - Fungsi yang akan dieksekusi pertama kali adalah main. Dalam program Java, eksekusi program dimulai dari fungsi main. Ketika Anda menjalankan program, Java akan mencari fungsi main dan menjalankannya terlebih dahulu.
2. Manakah tata cara penulisan fungsi benar di dalam class? Di atas fungsi main ataukah dibawah fungsi main? Jelaskan!
  - Dalam Java, tidak ada persyaratan khusus untuk penempatan fungsi di dalam class selama sintaksisnya benar. Namun, konvensi umumnya adalah menempatkan fungsi main di atas atau di awal class. Ini membantu dalam membaca kode dengan mudah karena fungsi main adalah titik awal eksekusi program. Fungsi-fungsi lainnya, seperti `PenerimaUcapan` dan `UcapanTerimaKasih`, dapat ditempatkan di atas atau di bawah fungsi main tergantung pada preferensi dan struktur logis program.
3. Modifikasi program di atas dengan menambahkan fungsi `UcapanTambahan()` dengan input parameter `String`. Fungsi `UcapanTambahan()` berisi ucapan tambahan yang ingin disampaikan kepada penerima ucapan.
  - Contoh program yang diberi Fungsi `UcapanTambahan()` berisi ucapan tambahan yang ingin disampaikan kepada penerima ucapan

Kode Program :

```
1  import java.util.Scanner;
2
3  public class UcapanTerimaKasih_24_1 {
4      public static String PenerimaUcapan() {
5          Scanner sc = new Scanner(System.in);
6          System.out.print("Tuliskan NAMA orang yang ingin Anda beri ucapan: ");
7          String namaOrang = sc.nextLine();
8          sc.close();
9          return namaOrang;
10     }
11
12     public static void UcapanTerimaKasih() {
13         String nama = PenerimaUcapan();
14         System.out.println("Thank you " + nama + " for being the best teacher in the world.\n" +
15             "You inspired in me a love for learning and made me feel like I could ask you anything.");
16     }
17
18     public static void UcapanTambahan(String tambahan) {
19         System.out.println(tambahan);
20     }
21
22     public static void main(String[] args) {
23         UcapanTerimaKasih();
24
25         // Menambahkan ucapan tambahan
26         String tambahanUcapan = "Your guidance has been invaluable. I am truly grateful.";
27         UcapanTambahan(tambahanUcapan);
28     }
29 }
30
```

Hasil run :

```
Tuliskan NAMA orang yang ingin Anda beri ucapan: Bu Triana
Thank you Bu Triana for being the best teacher in the world.
You inspired in me a love for learning and made me feel like I could ask you a
nything.
Your guidance has been invaluable. I am truly grateful.
```

### Percobaan 5: Fungsi Varargs

1. Buat class baru, simpan file tersebut dengan nama Percobaan5NoAbsen.java.
2. Buatlah fungsi Tampil (bertipe void) di dalam class tersebut dengan menggunakan dua jenis tipe data parameter yaitu string dan int

```
static void Tampil(String str, int...a){
    System.out.println("String : "+ str);
    System.out.println("Jumlah argumen/parameter: " + a.length);

    for (int i : a){
        System.out.print(i + " ");
    }
    System.out.println();
}
```

3. Buat fungsi main di dalam class tersebut, dan eksekusi fungsi Tampil dari dalam fungsi main.

```
public static void main(String[] args) {

    Tampil("Daspro 2019", 100, 200);
    Tampil("Teknologi Informasi", 1,2,3,4,5);
    Tampil("Polinema");
}
```

Hasil Run:

```
String : Daspro 2019
Jumlah argumen/parameter: 2
100 200
String : Teknologi Informasi
Jumlah argumen/parameter: 5
1 2 3 4 5
String : Polinema
Jumlah argumen/parameter: 0
```

### **Pertanyaan!**

1. Jelaskan mengapa penulisan parameter di praktikum 5 di tulis dengan int... a!
  - Penulisan int... a dalam parameter fungsi menunjukkan penggunaan varargs dalam Java. Varargs (variable-length arguments) memungkinkan sebuah fungsi menerima jumlah argumen yang tidak terbatas dari tipe data yang sama. Dengan menggunakan varargs, Anda dapat memanggil fungsi dengan jumlah argumen yang bervariasi.
2. Sebutkan kegunaan varargs dalam implementasi kode program untuk menyelesaikan permasalahan dalam dunia nyata! (minimal 3)
  - Contoh 1 (Menghitung Rata-rata):
  - Contoh 2 (Menggabungkan String):
  - Contoh 3 (Menghitung Total Belanja):
  - Contoh (Menghitung Diskon Belanja):
3. Bisakah kita menggunakan dua tipe data varargs dalam satu fungsi? Berikan contohnya!
  - Tidak, Anda tidak dapat menggunakan dua tipe data varargs dalam satu fungsi. Varargs hanya dapat memiliki satu jenis tipe data. Jika Anda memerlukan dua jenis tipe data, Anda dapat membuat dua fungsi terpisah atau menggunakan tipe data yang lebih umum seperti Object dan melakukan penanganan tipe data secara manual. Contoh:

## Percobaan 6: Pembuatan Kode Program, Dengan Fungsi versus Tanpa Fungsi

1. Buat class baru, simpan file tersebut dengan nama Percobaan6NoAbsen.java.
2. Buatlah program untuk menghitung luas persegi panjang dan volume balok tanpa menggunakan fungsi

```
public static void main(String[] args) {
    Scanner sc01 = new Scanner(System.in);
    int p,l,t,L,vol;

    System.out.print("Masukkan Panjang: ");
    p = sc01.nextInt();
    System.out.print("Masukkan Lebar: ");
    l = sc01.nextInt();
    System.out.print("Masukkan Tinggi: ");
    t = sc01.nextInt();

    L = p*l;
    System.out.println("Luas Persegi Panjang adalah : "+ L);
    vol = p*l*t;
    System.out.println("Volume Balok Adalah : "+ vol);
}
```

3. Program menghitung luas persegi dan volume balok diatas jika dibuatkan fungsi maka terdapat 3 fungsi yaitu hitungLuas, hitungVolume dan fungsi main, seperti dibawah ini:

Fungsi hitungLuas

```
static int hitungLuas (int pjg, int lb){
    int Luas = pjg*lb;
    return Luas;
}
```

Fungsi hitungVolume

```
static int hitungVolume(int tinggi, int a, int b){
    int volume=hitungLuas(a, b)*tinggi;
    return volume;
}
```

Fungsi main

```

public static void main(String[] args) {
    Scanner sc01 = new Scanner(System.in);
    int p,l,t,L,vol;

    System.out.print("Masukkan Panjang: ");
    p = sc01.nextInt();
    System.out.print("Masukkan Lebar: ");
    l = sc01.nextInt();
    System.out.print("Masukkan Tinggi: ");
    t = sc01.nextInt();

    L = p*l;
    System.out.println("Luas Persegi Panjang adalah : "+ L);
    vol = p*l*t;
    System.out.println("Volume Balok Adalah : "+ vol);
}

```

4. Lakukan validasi hasil dengan memberikan contoh inputan pada program tersebut!

Tampilkan hasilnya!

```

Masukkan Panjang: 3
Masukkan Lebar: 4
Masukkan Tinggi: 2
Luas Persegi Panjang adalah : 12
Volume Balok Adalah : 24

```

### Pertanyaan!

- Sebutkan tahapan dan urutan eksekusi pada percobaan 6 di atas!
  - Deklarasi Variabel:  
Variabel p, l, t, L, dan vol dideklarasikan untuk menyimpan panjang, lebar, tinggi, luas, dan volume.
  - Input dari Pengguna:
  - Menggunakan objek Scanner, program meminta pengguna memasukkan nilai panjang (p), lebar (l), dan tinggi (t).
  - Perhitungan Luas dan Volume:
  - Menggunakan rumus  $L = p * l$ , program menghitung luas (L) persegi panjang dan mencetaknya.
  - Menggunakan fungsi hitungLuas, program juga dapat menghitung luas persegi panjang.
  - Menggunakan rumus  $vol = p * l * t$ , program menghitung volume (vol) balok dan mencetaknya.
  - Menggunakan fungsi hitungVolume, program juga dapat menghitung volume balok.
  - Panggilan Fungsi:  
Fungsi hitungLuas dipanggil dengan parameter a dan b, yang nilainya adalah panjang dan lebar dari input pengguna.

- Fungsi hitungVolume dipanggil dengan parameter tinggi, a, dan b, yang nilainya adalah tinggi, panjang, dan lebar dari input pengguna.
- Cetak Hasil:  
Hasil luas persegi panjang dan volume balok dicetak ke layar.
- Penutup Scanner:  
Objek Scanner ditutup untuk menghindari kebocoran sumber daya.
- Selesai:  
Program selesai dijalankan setelah mencetak hasil luas dan volume.

2. Apakah output dari program dibawah ini kemudian jelaskan alur jalannya program tersebut!

- **\*\*Alur Jalannya Program:\*\***
  1. **\*\*Deklarasi Fungsi `TampilHinggaKei`:\*\***  
- Fungsi ini menerima satu parameter `i` dan mencetak angka dari 1 hingga `i` tanpa spasi.
  2. **\*\*Deklarasi Fungsi `Jumlah`:\*\***  
- Fungsi ini menerima dua parameter `bil1` dan `bil2`, dan mengembalikan jumlah dari keduanya.
  3. **\*\*Deklarasi Fungsi `TampilJumlah`:\*\***  
- Fungsi ini menerima dua parameter `bil1` dan `bil2`, dan memanggil `TampilHinggaKei` dengan argumen hasil penjumlahan dari `bil1` dan `bil2`.
  4. **\*\*Fungsi `main`:\*\***  
- Membuat variabel `temp` yang menyimpan hasil penjumlahan dari 1 dan 1, yaitu 2.  
- Memanggil fungsi `TampilJumlah` dengan parameter `temp` dan 5.
- **\*\*Alur Eksekusi:\*\***
  1. **\*\*Panggilan Fungsi `main`:\*\***  
- Membuat variabel `temp` dengan nilai 2.  
- Memanggil fungsi `TampilJumlah` dengan parameter `temp` (2) dan 5.
  2. **\*\*Panggilan Fungsi `TampilJumlah`:\*\***  
- Memanggil fungsi `TampilHinggaKei` dengan argumen hasil penjumlahan dari `temp` (2) dan 5.  
- Panggilan ini menghasilkan pemanggilan `TampilHinggaKei(7)`.
  3. **\*\*Panggilan Fungsi `TampilHinggaKei`:\*\***  
- Memasuki loop untuk `j` dari 1 hingga 7.  
- Mencetak angka `j` ke layar tanpa spasi.
  4. **\*\*Output Program:\*\***  
- Angka 1 sampai 7 dicetak ke layar tanpa spasi.

3. Pada saat apakah fungsi yang kita buat harus menggunakan parameter atau tidak? Pada saat apakah fungsi yang kita buat harus memiliki nilai kembalian atau tidak?Jelaskan!

- **\*\*Kapan Fungsi Harus Menggunakan Parameter:\*\***
  1. **\*\*Ketika Fungsi Memerlukan Input Eksternal:\*\***

- Jika fungsi membutuhkan nilai atau data dari luar fungsi (dari tempat pemanggilan), Anda perlu menggunakan parameter. Parameter memberikan cara untuk mengirimkan informasi ke dalam fungsi.

2. **\*\*Ketika Fungsi Dapat Digunakan untuk Berbagai Kasus:\*\***

- Jika Anda ingin membuat fungsi yang dapat digunakan dengan berbagai input, menggunakan parameter memungkinkan fleksibilitas dan reusabilitas. Sebagai contoh, fungsi menghitung luas persegi panjang dapat menerima panjang dan lebar sebagai parameter.

➤ **\*\*Kapan Fungsi Harus Memiliki Nilai Kembalian:\*\***

1. **\*\*Ketika Fungsi Menghasilkan Hasil Perhitungan atau Pemrosesan:\*\***

- Jika fungsi melakukan perhitungan atau pemrosesan tertentu dan hasilnya perlu disampaikan ke pemanggil, Anda perlu menggunakan nilai kembalian. Ini memungkinkan pemanggil untuk menggunakan hasil tersebut.

2. **\*\*Ketika Fungsi Melakukan Operasi yang Merubah Keadaan (State):\*\***

- Jika fungsi melakukan operasi yang mengubah keadaan atau data internal, nilai kembalian dapat digunakan untuk memberi tahu pemanggil apakah operasi berhasil atau memberikan informasi tambahan tentang hasil operasi.

3. **\*\*Ketika Fungsi Harus Mengembalikan Nilai yang Digunakan Pada Tempat Lain:\*\***

- Jika nilai yang dihasilkan oleh fungsi diperlukan di bagian lain dari program, fungsi harus memiliki nilai kembalian. Ini memungkinkan nilai tersebut digunakan atau disimpan untuk keperluan selanjutnya.

➤ Jadi, penggunaan parameter dan nilai kembalian tergantung pada kebutuhan fungsi dan bagaimana fungsi tersebut akan digunakan dalam program. Fungsi dapat memiliki parameter tanpa nilai kembalian, nilai kembalian tanpa parameter, atau keduanya tergantung pada kebutuhan spesifiknya.



## Tugas

1. Buatlah sebuah class KubusNoAbsen yang di dalamnya terdapat fungsi untuk menghitung volume kubus dan luar permukaan kubus!

➤ Kode Program :

```
1  import java.util.Scanner;
2  public class Kubus24 {
3      static int hitungVolume (int sisi) {
4          int Volume = sisi*sisi*sisi;
5          return Volume;
6      }
7      static int hitungLuasPermukaan (int a) {
8          int Luas = 6*a;
9          return Luas;
10     }
11     public static void main(String[] args) {
12         Scanner sc = new Scanner (System.in);
13         int s, L, vol;
14         System.out.print("Masukkan sisi ");
15         s = sc.nextInt();
16
17         vol = hitungVolume(s);
18         System.out.println("Volume Kubus adalah " + vol);
19         L = hitungLuasPermukaan(s);
20         System.out.println("Luas Permukaan Kubus adalah " + L);
21
22     }
23 }
```

➤ Hasil Run:

```
Masukkan sisi 2
Volume Kubus adalah 8
Luas Permukaan Kubus adalah 12
PS E:\Daspro\Pratikum\Tugas\Jobsheet 13> █
```