

## Acara 17 -18

**Pokok Bahasan : Fungsi Method Procedure**

**Acara Praktikum/Praktek: Minggu 5 / 1-2**

**Tempat : Politeknik Negeri Jember**

**Alokasi Waktu : 100 Menit**

### a. Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu menerapkan fungsi, metode dan procedure

### b. Indikator

Kemampuan mahasiswa dalam memahami Fungsi dalam Java

### c. Dasar Teori

**Pengertian Fungsi**, Dalam pemrograman terdapat istilah **fungsi, prosedur, dan method**, yang ketiganya pada dasarnya adalah sama, di mana kita dapat menyebut ketiganya sebagai fungsi.

- **Prosedur** adalah sebutan untuk fungsi yang tidak mengembalikan nilai. Fungsi ini biasanya ditandai dengan kata kunci `void`.
- **Fungsi** adalah sebutan untuk fungsi yang mengembalikan nilai.
- **Method** adalah fungsi yang berada di dalam sebuah Class.

Ada 2 jenis fungsi di Java, yaitu **fungsi static dan non-static**. Fungsi static adalah fungsi yang dapat dieksekusi langsung tanpa harus melakukan instansiasi objek. Sedangkan fungsi non-static adalah fungsi yang harus dieksekusi dari objek yang sudah diinstansiasi, di mana ini berkaitan dengan OOP. Sehingga pada modul praktikum kali ini kita hanya membahas tentang fungsi static di Java. Cara mendeklarasikan fungsi static di Java adalah dengan menambahkan keyword `static`.

**Sintaks Fungsi di Java**, Fungsi harus dibuat atau ditulis di dalam class. Sintaks dasar penulisan fungsi adalah sebagai berikut:

```
static TypeDataKembalian namaFungsi(){  
    // statement  
}
```

Kata kunci `static`, artinya kita akan membuat fungsi static. `TypeDataKembalian` adalah tipe data dari nilai yang dikembalikan (output) setelah fungsi dieksekusi. Jika fungsi tersebut tidak mengembalikan output, maka `TypeDataKembalian` adalah `void`. Sedangkan, `namaFungsi()` adalah nama fungsi yang kita buat, ditulis dengan cara camel case. Contoh:

```
static void beriSalam(){
```

```
System.out.println("Halo! Selamat Pagi");
}
```

## Cara Menjalankan/Eksekusi Fungsi

Setelah kita membuat fungsi, selanjutnya kita bisa mengeksekusi fungsinya. Fungsi dapat dipanggil dari fungsi main atau dari fungsi yang lainnya. Contoh pemanggilan fungsi dalam fungsi main:

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

Kode selengkapnya, silahkan dicoba pada bagian Praktikum.

## Fungsi dengan Parameter

Parameter adalah variabel yang menampung nilai untuk diproses di dalam fungsi. Parameter berperan sebagai input untuk fungsi. Struktur dasarnya seperti berikut ini:

```
static TipeDataKembalian namaFungsi(TipeData namaParameter, TipeData
namaParameterLain){
    // statement
}
```

Parameter ditulis di antara parenthesis (...) setelah nama fungsi. Bila terdapat lebih dari satu parameter, maka dipisah dengan tanda koma , dan masing-masing parameter harus dideskripsikan tipe datanya.

Misalkan, dibuat parameter bernama ucapan dengan tipe String. Sehingga kita bisa menggunakan variabel ucapan di dalam fungsi berikut ini:

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

Cara eksekusi fungsinya adalah dengan memberikan nilai yang akan diinputkan sebagai parameter, bisa berupa variabel atau langsung nilainya. Contohnya:

```
String halo = "Hallo!";
beriUcapan(halo);
beriUcapan("Selamat datang di pemrograman Java");
```

## Fungsi yang Mengembalikan Nilai

Pada kasus tertentu dibutuhkan sebuah fungsi yang dapat mengembalikan nilai output sehingga bisa diolah pada proses berikutnya. Pengembalian nilai pada fungsi menggunakan keyword return.

```
static TypeDataKembalian namaFungsi(TypeData namaParameter){
    // statement
    return variabelOutput;
}
```

Contoh:

```
static int luasPersegi(int sisi){
    int luas = sisi * sisi;
    return luas;
}
```

Pada contoh tersebut, dibuat sebuah parameter bernama sisi. Kemudian fungsi akan mengembalikan output dengan tipe int (integer) dari variabel luas. Cara pemanggilan fungsi tersebut adalah seperti berikut:

```
System.out.println("Luas Persegi dengan sisi 5 = " + luasPersegi(5));
int luasan = luasPersegi(6);
```

## Praktikum

### Praktikum 1

1. Buat class baru dengan nama **Semangat**.

```
public class Semangat {
}
```

2. Buat fungsi beriSalam() di dalam class tersebut.

```
public class Semangat {
    static void beriSalam() {
        System.out.println("Halo! Selamat Pagi");
    }
}
```

3. Buat fungsi main di dalam class tersebut, dan eksekusi fungsi beriSalam() dari dalam fungsi main().

```
public class Semangat {
    static void beriSalam() {
        System.out.println("Halo! Selamat Pagi");
    }

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

```
}
```

## Praktikum 2

1. Buat fungsi `beriUcapan` dengan sebuah parameter bertipe `String` di dalam class `Semangat`.

```
public class Semangat {  
  
    static void beriSalam() {  
        System.out.println("Halo! Selamat Pagi");  
    }  
  
    static void beriUcapan(String ucapan) {  
        System.out.println(ucapan);  
    }  
  
    public static void main(String[] args) {  
        beriSalam();  
    }  
}
```

2. Eksekusi fungsi `beriUcapan` dari dalam fungsi `main`.

```
public class Semangat {  
  
    static void beriSalam() {  
        System.out.println("Halo! Selamat Pagi");  
    }  
  
    static void beriUcapan(String ucapan) {  
        System.out.println(ucapan);  
    }  
  
    public static void main(String[] args) {  
        beriSalam();  
        String salam = "Selamat datang di pemrograman Java";  
        beriUcapan(salam);  
    }  
}
```

## Praktikum 3

1. Buat class baru dengan nama `Persegi`.

```
public class Persegi {  
  
}
```

2. Buat fungsi `luasPersegi` di dalam class tersebut yang mengembalikan nilai luas (`int`) dan parameter masukan sisi (`int`).

```
public class Persegi {  
  
    static int luasPersegi(int sisi) {  
        int luas = sisi * sisi;  
        return luas;  
    }  
}
```

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

```
public class Persegi {  
  
    static int luasPersegi(int sisi) {  
        int luas = sisi * sisi;  
        return luas;  
    }  
  
    public static void main(String[] args) {  
        int luasan = luasPersegi(5);  
        System.out.println("Luas Persegi dengan sisi 5 = " + luasan);  
    }  
}
```

## Praktikum 4

1. Buat class baru dengan nama Hitung.

```
public class Hitung {  
  
}
```

2. Buatlah fungsi kali di dalam class tersebut yang mengembalikan nilai hasil (int) dan parameter masukan angka1 dan angka2 (int).

```
static int kali(int angka1, int angka2) {  
    int hasil = (angka1 + 10) % (angka2 + 19);  
    return hasil;  
}
```

3. Buatlah fungsi kurang di dalam class tersebut yang mengembalikan nilai hasil (int) dan parameter masukan angka1 dan angka2 (int) dan memanggil fungsi kali.

```
static int kurang(int angka1, int angka2) {  
    angka1 = angka1 + 7;  
    angka2 = angka2 + 4;  
    int hasil = kali(angka1, angka2);  
    return hasil;  
}
```

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

```
public static void main(String[] args) {  
    Scanner input = new Scanner(System.in);  
    System.out.print("Masukkan nilai 1: ");  
    int nilai1 = input.nextInt();  
    System.out.print("Masukkan nilai 2: ");  
    int nilai2 = input.nextInt();  
    int hasil = kurang(nilai1, nilai2);  
    System.out.println("Hasil akhir adalah " + hasil);  
}
```

## Praktikum 5

1. Buat class baru dengan nama Varargs.

```
2. public class Varargs {  
3.  
4. }
```

5. Buatlah fungsi Tampil (bertipe void) di dalam class tersebut dengan menggunakan dua jenis tipe data parameter yaitu string dan int

```
6. static void tampil(String pesan, int... angka) {  
7. System.out.println("String: " + pesan);  
8. System.out.println("Jumlah argumen/parameter: " + angka.length);  
9.  
10. for (int i = 0; i < angka.length; i++) {  
11.     System.out.print(angka[i] + " ");  
12. }  
13. System.out.println();  
14. }
```

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

```
16. public static void main(String[] args) {  
17. tampil("Daspro 2019", 100, 200);  
18. tampil("Teknologi Informasi", 1, 2, 3, 4, 5);  
19. tampil("Polinema");  
20. int[] angka = { 1, 2, 3, 4 };  
21. tampil("Malang", angka);  
22. }
```

## Praktikum 6

1. Buat class baru dengan nama Balok.

```
2. public class Balok {  
3.  
4. }
```

5. Buatlah program untuk menghitung luas permukaan dan volume balok tanpa menggunakan fungsi.

```
6. public static void main(String[] args) {  
7. Scanner input = new Scanner(System.in);  
8. System.out.print("Masukkan panjang: ");  
9. int p = input.nextInt();  
10. System.out.print("Masukkan panjang: ");  
11. int l = input.nextInt();  
12. System.out.print("Masukkan panjang: ");  
13. int t = input.nextInt();  
14.  
15. int luasPermukaan = 2 * ((p * l) + (p * t) + (l * t));  
16. System.out.println("Luas permukaan Balok adalah: " + luasPermukaan);  
17.  
18. int volume = p * l * t;  
19. System.out.println("Volume Balok adalah: " + volume);  
20. }
```

21. Program menghitung luas persegi dan volume balok di atas jika dibuatkan fungsi maka terdapat 3 fungsi yaitu hitungLuasPermukaan, hitungVolume dan fungsi main, seperti dibawah ini:

```

22. static int hitungLuasPermukaan(int p, int l, int t) {
23. int hasil = 2 * ((p * l) + (p * t) + (l * t));
24. return hasil;
25. }
26. static int volume(int p, int l, int t) {
27. int hasil = p * l * t;
28. return hasil;
29. }
30. public static void main(String[] args) {
31. Scanner input = new Scanner(System.in);
32. System.out.print("Masukkan panjang: ");
33. int p = input.nextInt();
34. System.out.print("Masukkan panjang: ");
35. int l = input.nextInt();
36. System.out.print("Masukkan panjang: ");
37. int t = input.nextInt();
38.
39. System.out.println("Luas permukaan Balok adalah: " + hitungLuasPermukaan(p, l, t));
40. System.out.println("Volume Balok adalah: " + hitungVolume(p, l, t));
41. }

```

42. Jelaskan jalannya program menghitung luas persegi dan volume balok diatas yang menggunakan fungsi!

#### d. Alat dan Bahan

1. Komputer
2. Netbeans
3. JDK

#### e. Prosedur Kerja

Kumpulkan hasil percobaan di atas dan tambahkan analisa untuk tiap percobaan, latihan, dan tugas yang telah dibuat. Laporan berupa file word dikumpulkan pada <http://jti.poliije.ac.id/elearning>

#### f. Kesimpulan

Mahasiswa memahami dan mampu mengimplementasikan fungsi

#### g. Rubrik Penilaian

No	Indikator	Skor*			
1	Ketepatan dalam menjalankan serta menjelaskan dari tugas	1	2	3	4
2	Kurang dan ketepatan dalam menjalankan serta menjelaskan dari tugas	1	2	3	4
3	Kurang tepat dalam menjelaskan tugas	1	2	3	4
4	Keterlambatan dan ketidaktepatan dalam menjelaskan tugas	1	2	3	4
Jumlah skor					

## Acara 19-20

**Pokok Bahasan : Fungsi Rekursif**

**Acara Praktikum/Praktek: Minggu 5 / 3-4**

**Tempat : Politeknik Negeri Jember**

**Alokasi Waktu : 100 Menit**

### **a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa mampu menjelaskan mengenai fungsi rekursif
2. Mahasiswa mampu menerapkan fungsi rekursif didalam Bahasa pemrograman

### **b. Indikator**

Kemampuan mahasiswa dalam menjelaskan dan mengimplementasikan fungsi rekursif

### **c. Dasar Teori**

Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri. Hal ini bisa terjadi karena di dalam suatu fungsi rekursi, terdapat statement/baris program yang memanggil fungsi itu sendiri. Ketika suatu fungsi rekursif dipanggil/dijalankan dan kemudian proses eksekusi sudah sampai pada statement pemanggilan fungsi itu sendiri, maka fungsi tersebut akan dipanggil/dijalankan kembali. Dan LAGI, ketika proses eksekusi sudah sampai pada statement pemanggilan fungsi itu sendiri, maka fungsi tersebut akan dipanggil/dijalankan kembali, BEGITU SETERUSNYA hingga didapatkan suatu KONDISI AKHIR dimana proses pemanggilan fungsi tersebut tidak dilakukan lagi. Jika kondisi akhir tersebut tidak ditemukan/tidak ada, maka fungsi tersebut akan dipanggil terus menerus (infinite loop) dan ini tidak diperbolehkan. Pada fungsi rekursif, terdapat dua komponen blok kode, yaitu:

**Base Case:** kode program yang menunjukkan sebuah batas berhenti dari proses rekursif, sehingga apabila nilai batas ini terpenuhi maka proses rekursif diakhiri.

**Recursion Call** atau **Reduction Step:** kode program untuk melakukan pemanggilan terhadap dirinya sendiri.

Pada umumnya, format fungsi rekursif mempunyai bentuk sebagai berikut:

```
if (nilai batas) {  
    // menyelesaikan masalah  
} else {  
    // mendefinisikan kembali masalah menggunakan rekursi  
}
```

Cabang IF merupakan base case, sedangkan ELSE merupakan recursion call. Agar rekursi dapat berhenti, recursion call harus mendekati base case di setiap pemanggilan fungsi rekursif.



Berdasarkan penjelasan tersebut, sekilas terlihat seperti fungsi tersebut dijalankan secara berulang-ulang, dan fungsi rekursif memang biasanya digunakan pada kasus-kasus perulangan. Perhatikan fungsi **tampilDeret()** di bawah ini.

```
public class Deret {  
  
    static void tampilDeret(int x) {  
        if (x > 0) {  
            System.out.print(x + " ");  
            // memanggil fungsi tampilDeret() itu sendiri dengan nilai parameter n -  
1            tampilDeret(x - 1);  
        } else {  
            System.out.println();  
        }  
    }  
  
    public static void main(String[] args) {  
        // memanggil fungsi tampilDeret() dengan nilai parameter 5  
        tampilDeret(5);  
    }  
}
```

Pada contoh fungsi rekursif tersebut, fungsi tampilDeret pertama kali dijalankan (di dalam fungsi main) dengan mengirimkan nilai parameter 5 menggunakan perintah tampilDeret(5). Dari proses pemanggilan fungsi tersebut akan ditampilkan nilai parameternya yaitu 5 dan diikuti dengan pemanggilan fungsi tampilDeret(4). Selanjutnya, dari proses pemanggilan fungsi tersebut akan ditampilkan nilai parameternya yaitu 4 dan diikuti dengan pemanggilan fungsi tampilDeret(3). Selanjutnya, dari proses pemanggilan fungsi tersebut akan ditampilkan nilai parameternya yaitu 3 dan diikuti dengan pemanggilan fungsi tampilDeret(2). Selanjutnya, dari proses pemanggilan fungsi tersebut akan ditampilkan nilai parameternya yaitu 2 dan diikuti dengan pemanggilan fungsi tampilDeret(1). Dari proses pemanggilan fungsi tersebut akan ditampilkan nilai parameternya yaitu 1 dan diikuti dengan pemanggilan fungsi tampilDeret(0). Dari proses pemanggilan fungsi tersebut maka akan ditampilkan ganti baris dan tidak ada pemanggilan fungsi tampilDeret selanjutnya (pemanggilan fungsi rekursif sudah selesai).

```
tampilDeret(5)  
5 -> tampilDeret(4)  
4 -> tampilDeret(3)  
3 -> tampilDeret(2)  
2 -> tampilDeret(1)  
1 -> tampilDeret(0)
```

Tampilan yang muncul dari hasil pemanggilan fungsi tampilDeret() adalah:

```
5 4 3 2 1
```

## Percobaan 1

Pada percobaan ini akan dilakukan pembuatan program untuk menghitung nilai faktorial dari suatu bilangan dengan menggunakan fungsi rekursif. Selain itu, akan dibuat juga fungsi untuk menghitung nilai faktorial dengan menggunakan algoritma iteratif sebagai pembandingnya.

Jalankan NetBeans IDE atau Text Editor

1. Buatlah class dengan nama `Percobaan1`
2. Buat fungsi `static` dengan nama `faktorialRekursif()`, dengan tipe data kembalian fungsi `int` dan memiliki 1 parameter dengan tipe data `int` berupa bilangan yang akan dihitung nilai faktorialnya.

```
static int faktorialRekursif(int n) {  
    if (n == 0) {  
        return 1;  
    } else {  
        return n * faktorialRekursif(n - 1);  
    }  
}
```

3. Buat lagi fungsi `static` dengan nama `faktorialIteratif()`, dengan tipe data kembalian fungsi `int` dan memiliki 1 parameter dengan tipe data `int` berupa bilangan yang akan dihitung nilai faktorialnya.

```
static int faktorialIteratif(int n) {  
    int faktor = 1;  
    for (int i = n; i >= 1; i--) {  
        faktor *= i;  
    }  
    return faktor;  
}
```

4. Buatlah fungsi `main()` dan lakukan pemanggilan terhadap kedua fungsi yang telah dibuat sebelumnya, dan tampilkan hasil yang didapatkan.

```
public static void main(String[] args) {  
    System.out.println(faktorialRekursif(5));  
    System.out.println(faktorialIteratif(5));  
}
```

5. Jalankan program tersebut. Amati apa yang terjadi!

6. Jika ditelusuri, pada saat pemanggilan fungsi faktorialRekursif(5), maka proses yang terjadi dapat diilustrasikan sebagai berikut:

```
faktorialRekursif(5)
  5 * faktorialRekursif(4)
    4 * faktorialRekursif(3)
      3 * faktorialRekursif(2)
        2 * faktorialRekursif(1)
          return 1
        return 2 * 1
      return 3 * 2
    return 4 * 6
  return 5 * 24
120
```

## Percobaan 2

Pada percobaan ini akan dilakukan pembuatan program untuk menghitung pangkat sebuah bilangan dengan menggunakan fungsi rekursif.

1. Jalankan NetBeans IDE atau Text Editor
2. Buatlah class dengan nama Percobaan2
3. Buat fungsi static dengan nama hitungPangkat(), dengan tipe data kembalian fungsi int dan memiliki 2 parameter dengan tipe data int berupa bilangan yang akan dihitung pangkatnya dan bilangan pangkatnya.
- 4.

```
static int hitungPangkat(int x, int y) {
if (y == 0) {
    return 1;
} else {
    return x * hitungPangkat(x, y - 1);
}
}
```

5. Buatlah fungsi main() dan deklarasikan Scanner dengan nama input
6. Buatlah dua buah variabel bertipe int dengan nama bilangan dan pangkat
7. Tambahkan kode berikut ini untuk menerima input dari keyboard

```
System.out.print("Bilangan yang dihitung: ");
bilangan = input.nextInt();
System.out.print("Pangkat: ");
```

```
pangkat = sc.nextInt();
```

8. Lakukan pemanggilan fungsi `hitungPangkat()` yang telah dibuat sebelumnya dengan mengirimkan dua nilai parameter `bilangan` dan `pangkat`.

```
System.out.println(hitungPangkat(bilangan, pangkat));
```

9. Jalankan program tersebut. Amati apa yang terjadi!

### Percobaan 3

Pada percobaan ini akan dilakukan pembuatan program untuk menghitung jumlah uang nasabah yang disimpan di Bank setelah mendapatkan bunga selama beberapa tahun dengan menggunakan fungsi rekursif.

1. Jalankan NetBeans IDE atau Text Editor anda
2. Buatlah class dengan nama `Percobaan3`
3. Buat fungsi `static` dengan nama `hitungBunga()`, dengan tipe data kembalian fungsi `double` dan memiliki 2 parameter dengan tipe data `int` berupa saldo nasabah dan lamanya menabung.

Pada kasus ini dianggap bunga yang ditentukan oleh bank adalah 11% per tahun. Karena perhitungan bunga adalah bunga *saldo*, *sehingga untuk menghitung besarnya uang setelah ditambah bunga adalah saldo + bunga saldo*. Dalam hal ini, besarnya bunga adalah 0.11 *saldo*, dan *saldo dianggap 1 saldo*, sehingga *1 saldo + 0.11 saldo* dapat diringkas menjadi *1.11 \* saldo* untuk perhitungan saldo setelah ditambah bunga (dalam setahun).

```
static double hitungBunga(double saldo, int tahun) {  
    if (tahun == 0) {  
        return saldo;  
    } else {  
        return 1.11 * hitungBunga(saldo, tahun - 1);  
    }  
}
```

4. Buatlah fungsi main dan deklarasikan Scanner dengan nama `input`
5. Buatlah sebuah variabel bertipe `double` dengan nama `saldoAwal` dan sebuah variabel bertipe `int` bernama `tahun`

6. Tambahkan kode berikut ini untuk menerima input dari keyboard

```
System.out.print("Jumlah saldo awal: ");
saldoAwal = input.nextInt();
System.out.print("Lamanya menabung (tahun): ");
tahun = sc.nextInt();
```

7. Lakukan pemanggilan fungsi `hitungBunga()` yang telah dibuat sebelumnya dengan mengirimkan dua nilai parameter.

```
System.out.print("Jumlah uang setelah " + tahun + " tahun: ");
System.out.println(hitungBunga(saldoAwal, tahun));
```

8. Jalankan program tersebut. Amati apa yang terjadi!

**d. Alat dan Bahan**

1. Komputer
2. Netbeans
3. JDK

**e. Prosedur Kerja**

Kumpulkan hasil percobaan di atas dan tambahkan analisa untuk tiap percobaan, latihan, dan tugas yang telah dibuat. Laporan berupa file word dikumpulkan pada <http://jti.polije.ac.id/elearning>

**f. Kesimpulan**

Mahasiswa memahami dan mampu mengimplementasikan fungsi rekursif

**g. Rubrik Penilaian**

No	Indikator	Skor*			
1	Ketepatan dalam menjalankan serta menjelaskan dari tugas	1	2	3	4
2	Kurang dan ketepatan dalam menjalankan serta menjelaskan dari tugas	1	2	3	4
3	Kurang tepat dalam menjelaskan tugas	1	2	3	4
4	Keterlambatan dan ketidaktepatan dalam menjelaskan tugas	1	2	3	4
Jumlah skor					