

JOBSHEET 14
DASAR PEMROGRAMAN



ALVINO VALERIAN D.R
2341720027
D-IV TEKNIK INFORMATIKA

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2023

Pertanyaan

1. Apa yang dimaksud dengan fungsi rekursif?

Fungsi rekursif dalam Java merujuk pada kemampuan suatu fungsi untuk memanggil dirinya sendiri selama proses eksekusi. Dengan kata lain, fungsi tersebut dapat melakukan pemanggilan diri sendiri untuk menyelesaikan tugas yang lebih kompleks.

2. Bagaimana contoh kasus penggunaan fungsi rekursif ?

Perhitungan Faktorial, Pencarian dalam Struktur Data, Pembagian dan Penaklukan, Fibonacci Sequence

3. Pada **Percobaan1**, apakah hasil yang diberikan fungsi **faktorialRekursif()** dan fungsi **faktorialIteratif()** sama? Jelaskan perbedaan alur jalannya program pada penggunaan fungsi rekursif dan fungsi iteratif!

Pada percobaan ini, hasil yang diberikan oleh fungsi **factorialRekursif()** dan **factorialIteratif()** akan sama

Fungsi **factorialRekursif()**:

- Program akan memanggil fungsi rekursif **factorialRekursif(5)**.

- Fungsi ini akan memanggil dirinya sendiri dengan nilai yang berkurang hingga mencapai base case (ketika $n == 0$).

- Setiap pemanggilan rekursif akan mengalikan nilai n dengan hasil dari pemanggilan rekursif sebelumnya.

- Proses ini terus berlanjut hingga mencapai base case, dan hasilnya dikembalikan ke pemanggil sebelumnya.

Fungsi **factorialIteratif()**:

- Program menggunakan pendekatan iteratif dengan menggunakan loop **for**.

- Sebuah variabel faktor diinisialisasi dengan nilai 1.

- Loop akan berjalan dari n ke 1, mengalikan faktor dengan nilai iterasi saat ini.

- Proses ini terus berlanjut hingga loop selesai, dan hasil faktorial dikembalikan.

Perbedaan Utama:

Fungsi rekursif menggunakan pemanggilan diri sendiri dan memerlukan pemanggilan rekursif berulang sampai mencapai base case. Fungsi iteratif menggunakan loop untuk mengulangi operasi secara berurutan hingga mencapai kondisi berhenti.

Pertanyaan

1. Pada **Percobaan2**, terdapat pemanggilan fungsi rekursif

hitungPangkat(bilangan,pangkat) pada fungsi main, kemudian dilakukan pemanggilan fungsi **hitungPangkat()** secara berulang kali. Jelaskan sampai kapan proses pemanggilan fungsi tersebut akan dijalankan!

Dalam contoh ini, proses pemanggilan fungsi **hitungPangkat()** akan terus berlangsung hingga mencapai base case ketika **pangkat == 0**. Pada titik ini, nilai **1** akan dikembalikan, dan proses rekursif akan berhenti. Jadi, proses pemanggilan fungsi tersebut akan dihentikan ketika pangkat mencapai nilai **0**.

2. Tambahkan kode program untuk mencetak deret perhitungan pangkatnya.

Contoh :

hitungPangkat(2,5) dicetak $2 \times 2 \times 2 \times 2 \times 2 \times 1 = 32$

```
public class percobaan2 {
    static int hitungPangkat(int x,int y){
        if(y==0){
            System.out.print(s:"1x");
            return 1;
        }else{
            int hasil = x * hitungPangkat(x, y-1);
            if(y==1){
                System.out.print(x);
            }else{
                System.out.print("x"+x);
            }
            return hasil;
        }
    }

    public static void main(String[] args) {
        int bilangan,pangkat;
        Scanner sc= new Scanner(System.in);
        System.out.print(s:"Bilangan Yang Dihitung: ");
        bilangan = sc.nextInt();
        System.out.print(s:"Bilangan Yang Dipangkat ");
        pangkat = sc.nextInt();

        System.out.print("hitungPangkat("+bilangan+","+pangkat+") Dicetak ");

        int hasilPangkat = hitungPangkat(bilangan, pangkat);

        System.out.println(" = " + hasilPangkat);

        sc.close();
    }
}
```

```
hitungPangkat(2,5) Dicetak 1x2x2x2x2x2 = 32
PS D:\alvino\1.Daspro prak\minggu14>
```

Pertanyaan

1. Pada **Percobaan3**, sebutkan blok kode program manakah yang merupakan “base case”

```
if (tahun ==0 ) {  
    return (saldo);
```

dan “recursion call”!

```
return (1.11 * hitungLaba(saldo, tahun-1));
```

2. Jelaskan trace fase ekspansi dan fase substitusi algoritma perhitungan laba di atas jika diberikan nilai **hitungLaba(100000,3)**

Fase Ekspansi (Expanding Phase):

- Pemanggilan pertama: hitungLaba(100000, 3)
- Pada pemanggilan ini, fungsi akan melakukan pemanggilan dirinya sendiri dengan nilai tahun yang berkurang satu, yaitu hitungLaba(100000, 2).
- Pemanggilan kedua: hitungLaba(100000, 2)
- Pada pemanggilan ini, fungsi kembali memanggil dirinya sendiri dengan nilai tahun yang kembali berkurang satu, yaitu hitungLaba(100000, 1).
- Pemanggilan ketiga: hitungLaba(100000, 1)
- Karena nilai tahun sekarang adalah 0 (basis atau base case), fungsi langsung mengembalikan nilai saldo (yaitu 100000).

Fase Substitusi (Substitution Phase):

- Pemanggilan ketiga mengembalikan nilai 100000.
- Pemanggilan kedua menggunakan nilai kembalian dari pemanggilan ketiga dan mengalikannya dengan 1.11, yaitu $1.11 * 100000$.
- Pemanggilan pertama menggunakan nilai kembalian dari pemanggilan kedua dan mengalikannya lagi dengan 1.11, yaitu $1.11 * (1.11 * 100000)$.

Hasil =

```
Jumlah Saldo Awal :  
100000  
Lamanya Inventaris (tahun) :  
3  
Jumlah Saldo Setelah 3 Tahun : 136763.100000000003
```

TUGAS

```
public class tugas1 {  
    // Fungsi rekursif  
    static void deretDescendingRekursif(int n) {  
        if (n >= 0) {  
            System.out.print(n + " ");  
            deretDescendingRekursif(n - 1);  
        }  
    }  
  
    // Fungsi iteratif  
    static void deretDescendingIteratif(int n) {  
        for (int i = n; i >= 0; i--) {  
            System.out.print(i + " ");  
        }  
    }  
  
    Run | Debug  
    public static void main(String[] args) {  
        int n = 5;  
  
        System.out.println(x:"Fungsi Rekursif:");  
        deretDescendingRekursif(n);  
  
        System.out.println(x:"\n\nFungsi Iteratif:");  
        deretDescendingIteratif(n);  
    }  
}
```

```
Fungsi Rekursif:  
5 4 3 2 1 0
```

```
Fungsi Iteratif:  
5 4 3 2 1 0
```

2.

```
public class tugas2 {  
    static int penjumlahanRekursif(int n) {  
        if (n == 1) {  
            return 1;  
        } else {  
            return n + penjumlahanRekursif(n - 1);  
        }  
    }  
  
    Run | Debug  
    public static void main(String[] args) {  
        int f = 8;  
  
        System.out.println("Penjumlahan untuk f = " + f + " adalah: " + penjumlahanRekursif(f));  
    }  
}
```

```
Penjumlahan untuk f = 8 adalah: 36  
PS D:\alvino\1.Daspro prak\minggu14>
```

3.

```
public class tugas3 {  
    static boolean cekPrimaRekursif(int n, int i) {  
        if (i == 1) {  
            return true;  
        } else {  
            if (n % i == 0) {  
                return false;  
            } else {  
                return cekPrimaRekursif(n, i - 1);  
            }  
        }  
    }  
  
    Run | Debug  
    public static void main(String[] args) {  
        int bilangan = 13;  
  
        if (cekPrimaRekursif(bilangan, bilangan / 2)) {  
            System.out.println(bilangan + " adalah bilangan prima.");  
        } else {  
            System.out.println(bilangan + " bukan bilangan prima.");  
        }  
    }  
}
```

```
13 adalah bilangan prima.
```

4.

```
tugas4.java x tugas4
1 public class tugas4 {
2     static int hitungPasanganMarmut(int bulan) {
3         if (bulan == 0 || bulan == 1) {
4             return 1;
5         } else {
6             return hitungPasanganMarmut(bulan - 1) + hitungPasanganMarmut(bulan - 2);
7         }
8     }
9
10    Run | Debug
11    public static void main(String[] args) {
12        int bulan = 12;
13        int jumlahPasanganMarmut = hitungPasanganMarmut(bulan);
14
15        System.out.println("Pada akhir bulan ke-" + bulan + ", ada " + jumlahPasanganMarmut + " pasangan marmut.");
16    }
17
18 }
```

Pada akhir bulan ke-12, ada 233 pasangan marmut.