

LAPORAN PRAKTIKUM DASPRO

Jobsheet 14



Nama: Aqil Rahmat Alifiandi

NIM: 2341760099

Prodi: D-IV Sistem Informasi Bisnis

**JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2023**

Percobaan 1

Kode Program:

```
import java.util.Scanner;
public class Percobaan1_05 {

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

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

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

```
Percobaan1_05
120
120
```

Pertanyaan:

1. Apa yang dimaksud dengan fungsi rekursif?
2. Bagaimana contoh kasus penggunaan fungsi rekursif ?
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!

Jawaban:

1. Fungsi rekursif adalah sebuah fungsi yang memanggil dirinya sendiri selama proses eksekusi. Dengan kata lain, dalam tubuh fungsi tersebut terdapat pemanggilan fungsi itu sendiri.

2. Perhitungan faktorial adalah contoh klasik penggunaan fungsi rekursif.

```
public class RekursifContoh {  
    public static int faktorial(int n) {  
        if (n == 0 || n == 1) {  
            return 1;  
        } else {  
            return n * faktorial(n - 1);  
        }  
    }  
  
    public static void main(String[] args) {  
        int hasil = faktorial(5);  
        System.out.println("Hasil faktorial: " + hasil);  
    }  
}
```

3. Pada Percobaan1, hasil yang diberikan oleh fungsi `faktorialRekursif()` dan `faktorialIteratif()` seharusnya sama, yaitu faktorial dari 5, yang adalah 120. Perbedaan utama terletak pada cara kedua fungsi tersebut menghitung faktorial. Perbedaan utama adalah alur eksekusi: fungsi rekursif memanggil dirinya sendiri secara berulang, sementara fungsi iteratif menggunakan loop untuk mencapai hasil yang sama. Meskipun begitu, keduanya mencapai hasil yang identik pada akhirnya.

Percobaan 2

Kode Program:

```
import java.util.Scanner;  
public class Percobaan2_05 {  
  
    static int hitungPangkat(int x, int y){  
        if (y == 0){  
            return (1);  
        } else {  
            return (x * hitungPangkat (x, y - 1));  
        }  
    }  
  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int bilangan, pangkat;  
  
        System.out.println(x:"Bilangan yang dihitung: ");  
        bilangan = sc.nextInt();  
        System.out.println(x:"Pangkat: ");  
        pangkat =sc.nextInt();  
  
        System.out.println(hitungPangkat(bilangan, pangkat));  
    }  
}
```

```
Percobaan2_05  
Bilangan yang dihitung:  
3  
Pangkat:  
2  
9
```

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!
2. Tambahkan kode program untuk mencetak deret perhitungan pangkatnya. Contoh : hitungPangkat(2,5) dicetak $2 \times 2 \times 2 \times 2 \times 2 \times 1 = 32$

Jawaban:

1. Dalam contoh ini, pemanggilan fungsi rekursif hitungPangkat() akan berlanjut sampai pangkat mencapai nilai 0, dan pada saat itu kondisi dasar akan terpenuhi, dan nilai 1 akan dikembalikan. Jadi, proses pemanggilan fungsi rekursif ini akan berhenti ketika pangkat mencapai 0.
2. Kode Program:

```
import java.util.Scanner;

public class Percobaan2_205 {

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

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int bilangan, pangkat;

        System.out.println(x:"Bilangan yang dihitung: ");
        bilangan = sc.nextInt();
        System.out.println(x:"Pangkat: ");
        pangkat = sc.nextInt();

        System.out.print(hitungPangkat(bilangan, pangkat));
    }
}
```

```
Bilangan yang dihitung:
2
Pangkat:
5
2 * 2 * 2 * 2 * 2 * 1 32%
```

Percobaan 3

Kode Program:

```
import java.util.Scanner;
public class Percobaan3_05 {
    static double hitungLaba (double saldo, int tahun){
        if (tahun == 0){
            return (saldo);
        } else {
            return (1.11 * hitungLaba (saldo, tahun -1));
        }
    }
}

Run | Debug
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    double saldoAwal;
    int tahun;

    System.out.print(s:"Jumlah saldo awal : ");
    saldoAwal = sc.nextInt();
    System.out.print(s:"Lamanya invetasi (tahun): ");
    tahun = sc.nextInt();

    System.out.print("Jumlah Saldo Setelah "+tahun+" Tahun : ")
    System.out.print(hitungLaba(saldoAwal, tahun));
}
```

```
Percobaan3_05
Jumlah saldo awal : 3000
Lamanya invetasi (tahun): 2
Jumlah Saldo Setelah 2 Tahun : 3696.30000000000006%
```

Pertanyaan

1. Pada Percobaan3, sebutkan blok kode program manakah yang merupakan “base case” dan “recursion call”!
2. Jabarkan trace fase ekspansi dan fase substitusi algoritma perhitungan laba di atas jika diberikan nilai `hitungLaba(100000,3)`

Jawaban:

1. java

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

Blok kode di atas merupakan basis kasus ('base case') karena ketika `tahun` sama dengan 0, fungsi `hitungLaba()` mengembalikan nilai `saldo` tanpa melakukan pemanggilan rekursif lebih lanjut.

java

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

Blok kode di atas merupakan pemanggilan rekursif ('recursion call') karena fungsi `hitungLaba()` memanggil dirinya sendiri dengan nilai `tahun` yang berkurang satu setiap kali. Proses ini terus berlanjut hingga mencapai basis kasus.

2. hasil dari pemanggilan hitungLaba(100000, 3) dengan nilai awal 100000 dan lamanya investasi 3 tahun adalah 13310000

Tugas

1. Kode Program:

```
public class DeretDescendingRekursif05 {  
  
    public static void tampilDeretRekursif(int n) {  
        if (n >= 0) {  
            System.out.println(n);  
            tampilDeretRekursif(n - 1);  
        }  
    }  
  
    Run | Debug  
    public static void main(String[] args) {  
        int n = 10;  
        tampilDeretRekursif(n);  
    }  
}
```

```
DeretDescendingRekursif05  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
0
```

```
public class DeretDescendingIteratif05 {  
    public static void tampilDeretIteratif(int n) {  
        for (int i = n; i >= 0; i--) {  
            System.out.println(i);  
        }  
    }  
  
    Run | Debug  
    public static void main(String[] args) {  
        int n = 10;  
        tampilDeretIteratif(n);  
    }  
}
```

```
DeretDescendingIteratif05  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
0
```

2. Kode Program:

```
public class PenjumlahanRekursif05 {  
  
    public static int jumlahRekursif(int n) {  
        if (n == 0) {  
            return 0;  
        } else {  
            return n + jumlahRekursif(n - 1);  
        }  
    }  
  
    Run | Debug  
    public static void main(String[] args) {  
        int n = 8;  
        int hasil = jumlahRekursif(n);  
        System.out.println("Hasil penjumlahan = " + hasil);  
    }  
}
```

PenjumlahanRekursif05
Hasil penjumlahan = 36

3. Kode Program:

```
public class CekPrimaRekursif05 {  
    public static boolean cekPrimaRekursif(int n) {  
        if (n <= 1) {  
            return false;  
        }  
        for (int i = 2; i <= Math.sqrt(n); i++) {  
            if (n % i == 0) {  
                return false;  
            }  
        }  
        return true;  
    }  
  
    Run | Debug  
    public static void main(String[] args) {  
        int n = 7;  
        boolean hasil = cekPrimaRekursif(n);  
        if (hasil) {  
            System.out.println(n + " adalah bilangan prima");  
        } else {  
            System.out.println(n + " bukan bilangan prima");  
        }  
    }  
}
```

CekPrimaRekursif05
7 adalah bilangan prima

4. Kode Program:

```
public class Fibonacci05 {  
  
    public static int hitungMarmut(int bulan) {  
        if (bulan <= 2) {  
            return 1;  
        } else {  
            return hitungMarmut(bulan - 1) + hitungMarmut(bulan - 2);  
        }  
    }  
  
    Run | Debug  
    public static void main(String[] args) {  
        int bulan = 12;  
        int jumlahMarmut = hitungMarmut(bulan);  
        System.out.println("Jumlah pasangan marmut pada akhir bulan ke-" + bulan + " adalah: " + jumlahMarmut);  
    }  
}
```

Jumlah pasangan marmut pada akhir bulan ke-12 adalah: 144