

LAPORAN PRAKTIKUM DASAR PEMROGRAMAN
JOBSHEET 14: FUNGSI (REKURSIF)



Nama: Afifah Khoirunnisa

NIM: 2341720250

Kelas: 1B

Prodi : D-IV Teknik Informatika

2.1 Percobaan 1

Waktu Percobaan : 60

Source Code:

```
public class Percobaan1 {  
    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));  
    }  
}
```

Output:

```
120  
120  
PS D:\COLLEGE\SEMESTER 1\PRAKTIKUM DASAR PEMROGRAMAN\PRAKTIKUM\WEEK 14>
```

Pertanyaan

1. Apa yang dimaksud dengan fungsi rekursif?

Jawaban:

Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri selama eksekusi. Dalam pemrograman, rekursi adalah teknik di mana suatu fungsi memecah tugas menjadi sub-tugas yang lebih kecil dan memanggil dirinya sendiri untuk menyelesaikan sub-tugas tersebut.

2. Bagaimana contoh kasus penggunaan fungsi rekursif ?

Jawaban:

Contoh kasus penggunaan fungsi rekursif yaitu pada operasi factorial.

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:

Fungsi Rekursif:

- Memanggil dirinya sendiri dengan argument yang lebih kecil atau berbeda.

- Menggunakan pemilihan if di mana rekursi berhenti dan nilai kembali langsung dihitung. Fungsi rekursif memecah masalah yang lebih kecil dan dan menyelesaikannya dengan cara serupa.

Fungsi Iteratif:

- Menggunakan pernyataan loop untuk mengulang tugas hingga kondisi keluar dari loop.
- Tidak ada pemanggilan dirinya sendiri dalam pendekatan iteratif.

2.2 Percobaan 2

Waktu Percobaan : 60

Source code:

```
import java.util.Scanner;;

public class Percobaan2 {

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

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

Output:

```
Bilangan yang dihitung: 3
Pangkat: 2
9
PS D:\COLLEGE\SEMESTER 1\PRAKTIKUM DASAR PEMROGRAMAN\PRAKTIKUM\WEEK 14>
```

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!

Jawaban:

Proses pemanggilan tersebut akan dijalankan hingga kondisi yang menjadi pembatasan dari program tersebut. Jika tidak, maka proses ini tidak akan pernah berhenti sampai memori yang digunakan untuk menampung proses tersebut penuh.

2. Tambahkan kode program untuk mencetak deret perhitungannya. Contoh : hitungPangkat(2,5) dicetak 2x2x2x2x2x1 = 32

Jawaban:

Source code:

```
import java.util.Scanner;;

public class Percobaan2 {
    static int bilangan, pangkat;
    static int hitungPangkat(int x, int y) {
        if (y == 0) {
            return (1);
        } else {
            System.out.print(x);
            if (y != 1) {
                System.out.print(s:" x ");
            } else {
                System.out.print(s:" x |1 = ");
            }
            return ( x * hitungPangkat(x, y-1));
        }
    }
}

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

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

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

Output:

```
Bilangan yang dihitung: 2
Pangkat: 5
2 x 2 x 2 x 2 x 2 x 1 = 32
```

2.3 Percobaan 3

Waktu Percobaan : 60

Source code:

```
import java.util.Scanner;

public class Percobaan3 {

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

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

        System.out.print("Jumlah saldo awal : ");
        saldoAwal = sc.nextDouble();
        System.out.print("Lama investasi (tahun): ");
        tahun = sc.nextInt();

        System.out.print("Jumlah saldo setelah " + tahun + " tahun : ");
        System.out.print(hitungLaba(saldoAwal, tahun));
    }
}
```

Output:

```
Jumlah saldo awal : 5000
Lama investasi (tahun): 3
Jumlah saldo setelah 3 tahun : 6838.1550000000025
```

Pertanyaan

1. Pada Percobaan3, sebutkan blok kode program manakah yang merupakan “base case” dan “recursion call”!

Jawaban:

Base case =

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

Recursion call =

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

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

Jawaban:

Trace fase ekspansi:

$$\begin{aligned} & \text{hitungLaba}(100000, 3) \\ &= 1.11 * \text{hitungLaba}(100000, 2) \\ &= 1.11 * 1.11 * \text{hitungLaba}(100000, 1) \\ &= 1.11 * 1.11 * 1.11 * \text{hitungLaba}(100000, 0) \end{aligned}$$

Trace fase substitusi:

$$\begin{aligned} & \text{hitungLaba}(100000, 0) \rightarrow 100000 \text{ (base case)} \\ & \text{hitungLaba}(100000, 1) \rightarrow 1.11 * \text{hitungLaba}(100000, 0) \rightarrow 1.11 * 100000 \\ & \text{hitungLaba}(100000, 2) \rightarrow 1.11 * \text{hitungLaba}(100000, 1) \rightarrow 1.11 * 1.11 * 100000 \\ & \text{hitungLaba}(100000, 3) \rightarrow 1.11 * \text{hitungLaba}(100000, 2) \rightarrow 1.11 * 1.11 * 1.11 * 100000 \end{aligned}$$

3. Tugas

Waktu Pengerjaan 120 menit

1. Buatlah program untuk menampilkan bilangan n sampai 0 dengan menggunakan fungsi rekursif dan fungsi iteratif. (DeretDescendingRekursif).

Jawaban:

Source code:

```
import java.util.Scanner;

public class DeretDescendingRekursif {
    static void DeretDescendingRekursif(int n) {
        if (n > 0) {
            System.out.print(n + " ");
            DeretDescendingRekursif(n-1);
        } else {
            System.out.print(i:0);
        }
    }

    static void DeretDescendingIteratif(int n) {
        System.out.print(n + " ");
        for (int i = n; i >= 1; i--) {
            n -= 1;
            System.out.print(n + " ");
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Masukkan angka yang ingin di urutkan: ");
        int n = sc.nextInt();
        System.out.println(x:"Deret Descending rekursif: ");
        DeretDescendingRekursif(n);
        System.out.println();
        System.out.println(x:"Deret Descending Iteratif: ");
        DeretDescendingIteratif(n);
    }
}
```

Output:

```
Masukkan angka yang ingin di urutkan: 3
Deret Descending rekursif:
3 2 1 0
Deret Descending Iteratif:
3 2 1 0
```

2. Buatlah program yang di dalamnya terdapat fungsi rekursif untuk menghitung penjumlahan bilangan. Misalnya $f = 8$, maka akan dihasilkan $1+2+3+4+5+6+7+8 = 36$ (PenjumlahanRekursif).

Jawaban:

Source code:

```

import java.util.Scanner;

public class PenjumlahanRekursif {
    static int PenjumlahanRekursif(int x) {
        if (x == 1) {
            return 1;
        } else {
            return x + PenjumlahanRekursif(x-1);
        }
    }
}

Run | Debug
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print(s:"Masukkan Angka yang ingin ditambahkan: ");
    int x = sc.nextInt();
    int hasil = PenjumlahanRekursif(x);
    for (int i = 1; i <=x; i++) {
        System.out.print(i);
        if (i < x) {
            System.out.print(s:" + ");
        }
    }
    System.out.print(" = " + hasil);
}
}

```

Output:

```

Masukkan Angka yang ingin ditambahkan: 5
1 + 2 + 3 + 4 + 5 = 15

```

3. Buat program yang di dalamnya terdapat fungsi rekursif untuk mengecek apakah suatu bilangan n merupakan bilangan prima atau bukan. n dikatakan bukan bilangan prima jika ia habis dibagi dengan bilangan kurang dari n. (CekPrimaRekursif).

Jawaban:

Source code:

```

import java.util.Scanner;

public class CekPrimaRekursif {
    static boolean isPrima(int n, int i) {
        if (i == 1) {
            return true;
        } else {
            if (n % i == 0) {
                return false;
            } else {
                return isPrima(n, i - 1);
            }
        }
    }

    static void cekBilanganPrima(int n) {
        if (n > 1 && isPrima(n, n / 2)) {
            System.out.println(x:"Bilangan tersebut adalah bilangan prima.");
        } else {
            System.out.println(x:"Bilangan tersebut bukan bilangan prima.");
        }
    }
}

Run | Debug
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print(s:"Masukkan bilangan: ");
    int n = sc.nextInt();
    cekBilanganPrima(n);
}
}

```

Output:

```

Masukkan bilangan: 41
Bilangan tersebut adalah bilangan prima.

```


4. Sepasang marmut yang baru lahir (jantan dan betina) ditempatkan pada suatu pembiakan. Setelah dua bulan pasangan marmut tersebut melahirkan sepasang marmut kembar (jantan dan betina). Setiap pasangan marmut yang lahir juga akan melahirkan sepasang marmut juga setiap 2 bulan. Berapa pasangan marmut yang ada pada akhir bulan ke-12? Buatlah programnya menggunakan fungsi rekursif! (Fibonacci). Berikut ini adalah ilustrasinya dalam bentuk tabel.

Bulan ke-	Jumlah Pasangan		Total Pasangan
	Produktif	Belum Produktif	
1	0	1	1
2	0	1	1
3	1	1	2
4	1	2	3
5	2	3	5
6	3	5	8
7	5	8	13
8	8	13	21
9	13	21	34
10	21	34	55
11	34	55	89
12	55	89	144

Jawaban:

Source code:

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

Output:

```
Pasangan marmut pada akhir bulan ke-12 adalah 144  
PS D:\COLLEGE\SEMESTER 1\PRAKTIKUM DASAR PEMROGRAMAN\PRAKTIKUM\WEEK 14>
```