

Percobaan 1

```
1 public class percobaan1 {
2
3     static int faktorialRekursif(int n) {
4         if (n == 0) {
5             return 1;
6         } else {
7             return n * faktorialRekursif(n - 1);
8         }
9     }
10
11     static int faktorialIteratif(int n) {
12         int faktor = 1;
13         for (int i = n; i >= 1; i--) {
14             faktor = faktor * i;
15         }
16         return faktor;
17     }
18
19     public static void main(String[] args) {
20         System.out.println(faktorialRekursif(5));
21         System.out.println(faktorialIteratif(5));
22     }
23 }
```

Run | Debug

PROBLEMS 2 OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

120
120

Pertanyaan

1. Apa yang dimaksud dengan fungsi rekursif?
Sebuah fungsi yang memanggil dirinya sendiri secara berulang-ulang untuk menyelesaikan masalah.
2. Bagaimana contoh kasus penggunaan fungsi rekursif ?
Menghitung faktorial, fibonacci, pangkat bilangan, dll.
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!
Hasil dari fungsi rekursif dan iteratif sama, yaitu 120. Fungsi rekursif cenderung lebih mudah dan ringkas, tetapi butuh memori dan waktu lebih banyak. Sedangkan fungsi iteratif lebih cepat dan hemat memori, tapi butuh lebih banyak kode + logika.

Percobaan 2

```
Bilangan yang dihitung : 5
Pangkat : 4
625
```

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!
Proses pemanggilan fungsi tersebut akan berjalan sampai nilai pangkatnya mencapai 0. Karena basis rekursif pada fungsi hitungPangkat adalah jika $y == 0$, returnnya 1 dan berhenti.
2. Tambahkan kode program untuk mencetak deret perhitungan pangkatnya. Contoh : hitungPangkat(2,5) dicetak $2 \times 2 \times 2 \times 2 \times 2 \times 1 = 32$

```
3 public class percobaan2 {
4     static int hitungPangkat(int x, int y, int z) {
5         if (y == 0) {
6             System.out.println(z + "^" + y + " = 1");
7             return(1);
8         } else {
9             System.out.println(z + "^" + y + " = " + z + " x " + z + "^" + (y-1));
10            return (x * hitungPangkat(x, y - 1, z));
11        }
12    }
13    public static void main(String[] args) {
14        Scanner sc = new Scanner(System.in);
15
16        System.out.println(x:"");
17        System.out.print(s:"Bilangan yang dihitung : ");
18        int bilangan = sc.nextInt();
19        System.out.print(s:"Pangkat : ");
20        int pangkat = sc.nextInt();
21        System.out.println(hitungPangkat(bilangan, pangkat, bilangan));
22    }
}
```

Run | Debug

PROBLEMS (2) OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
Bilangan yang dihitung : 2
Pangkat : 4
2^4 = 2 x 2^3
2^3 = 2 x 2^2
2^2 = 2 x 2^1
2^1 = 2 x 2^0
2^0 = 1
16
```

Percobaan 3

Pertanyaan

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

base case :

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

recursion call :

```
} else {  
    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)`

- Fase ekspansi :
 $\text{hitungLaba}(100000, 3) = 1.11 * \text{hitungLaba}(100000, 2) \rightarrow$ memanggil fungsi rekursif dengan tahun = 2
 $\text{hitungLaba}(100000, 2) = 1.11 * \text{hitungLaba}(100000, 1) \rightarrow$ memanggil fungsi rekursif dengan tahun = 1
 $\text{hitungLaba}(100000, 1) = 1.11 * \text{hitungLaba}(100000, 0) \rightarrow$ memanggil fungsi rekursif dengan tahun = 0
 $\text{hitungLaba}(100000, 0) = 100000 \rightarrow$ basis rekursif, mengembalikan nilai saldo awal
- Fase substitusi
 $\text{hitungLaba}(100000, 1) = 1.11 * 100000 = 111000 \rightarrow$ menggabungkan nilai dari $\text{hitungLaba}(100000, 0)$
 $\text{hitungLaba}(100000, 2) = 1.11 * 111000 = 123210 \rightarrow$ menggabungkan nilai dari $\text{hitungLaba}(100000, 1)$
 $\text{hitungLaba}(100000, 3) = 1.11 * 123210 = 136743.1 \rightarrow$ menggabungkan nilai dari $\text{hitungLaba}(100000, 2)$

Tugas

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

```
2
3 public class deretDescendingRekursif {
4
5     public static void rekursif(int n) {
6         if (n >= 0) {
7             System.out.print(n + " ");
8             rekursif(n-1);
9         }
10    }
11
12    public static void iteratif(int n) {
13        for (int i = n; i >= 0; i--) {
14            System.out.print(i + " ");
15        }
16    }
17
18    public static void main(String[] args) {
19        Scanner sc = new Scanner(System.in);
20        System.out.print(s:"Masukkan nilai n : ");
21        int n = sc.nextInt();
22        System.out.println(x:"Hasil fungsi rekursif :");
23        rekursif(n);
24        System.out.println();
25        System.out.println(x:"Hasil fungsi iteratif :");
26        iteratif(n);
27    }
}
```

Run | Debug

PROBLEMS (3) OUTPUT DEBUG CONSOLE TERMINAL PORTS Run

Masukkan nilai n: 2
Hasil fungsi rekursif:
2 1 0
Hasil fungsi iteratif:
2 1 0
D:\DASPRO\rekursif>

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).

```
1 import java.util.Scanner;
2
3 public class penjumlahanRekursif {
4
5     public static int sum(int n) {
6         if (n == 1) {
7             return 1;
8         } else {
9             return n + sum(n-1);
10        }
11    }
12
13    public static void main(String[] args) {
14        Scanner sc = new Scanner(System.in);
15        System.out.println(x:"");
16        System.out.print(s:"Masukkan nilai n : ");
17        int n = sc.nextInt();
18        System.out.println("Hasil penjumlahan dari 1 sampai " + n + " adalah " + sum(n));
19    }
}
```

Run | Debug

PROBLEMS (4) OUTPUT DEBUG CONSOLE TERMINAL PORTS Run: penjumlahanRekursif

Masukkan nilai n : 5
Hasil penjumlahan dari 1 sampai 5 adalah 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).

```
3 public class cekPrimaRekursif {
4     ....
5     public static boolean isPrima(int n, int i) {
6         if (i == 1) {
7             return true;
8         } else {
9             if (n % i == 0) {
10                return false;
11            } else {
12                return isPrima(n, i-1);
13            }
14        }
15    }
16
17    Run | Debug
18    public static void main(String[] args) {
19        Scanner sc = new Scanner(System.in);
20        System.out.print(s:"Masukkan nilai n : ");
21        int n = sc.nextInt();
22        boolean prima = isPrima(n, n-1);
23        if (prima) {
24            System.out.println(n + " adalah bilangan prima");
25        } else {
26            System.out.println(n + " bukan bilangan prima");
27        }
28    }
29 }
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE **TERMINAL** PORTS Run: cekPrimaRekursif + v

Masukkan nilai n: 2
2 adalah bilangan prima

D:\DASPRO\rekursif> d: && cd d:\DASPRO\rekursif && cmd /C "C:\Users\kaylars\AppData\Roaming\Code\User\globalStorage\pleiades.java-extension-pack-jdk\java\17\bin\java.exe -XX:+ShowCodeDetails -cp C:\Users\kaylars\AppData\Roaming\Code\User\workspaceStorage\4d19d6f77dcd4cfd867a78e0e90fd\redhat.java\jdt_ws\rekursif_56f0b9a\bin cekPrimaRekursif "

Masukkan nilai n : 6
6 bukan 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).

```
1 import java.util.Scanner;
2 public class fibonacci {
3     static int[] hitungMarmut (int bulan) {
4         int[] hasil = new int[3];
5         if (bulan <= 2) {
6             hasil[0] = 1;
7             hasil[1] = 0;
8             hasil[2] = 1;
9             return hasil;
10        }
11        int[] marmutSebelum = hitungMarmut (bulan - 1);
12        int totalPasangan = marmutSebelum[0] + marmutSebelum[2];
13        int jumlahPasanganProduktif = marmutSebelum[2];
14        hasil[0] = totalPasangan;
15        hasil[1] = jumlahPasanganProduktif;
16        hasil[2] = totalPasangan - jumlahPasanganProduktif;
17        return hasil;
18    }
19    Run | Debug
20    public static void main(String[] args) {
21        Scanner sc = new Scanner(System.in);
22        System.out.print("Masukkan bulan ke : ");
23        int bulan = sc.nextInt();
24
25        int[] jumlahMarmut = hitungMarmut(bulan);
26
27        System.out.println("Jumlah Pasangan produktif pada bulan ke-" + bulan + " adalah " + jumlahMarmut[1]);
28        System.out.println("Jumlah Pasangan belum produktif pada bulan ke-" + bulan + " adalah " + jumlahMarmut[2]);
29        System.out.println("Total pasangan marmut pada bulan ke-" + bulan + " adalah " + jumlahMarmut[0]);
30    }
```

```
Masukkan bulan ke: 1
Jumlah Pasangan produktif pada bulan ke-1 adalah 0
Jumlah Pasangan belum produktif pada bulan ke-1 adalah 1
Total pasangan marmut pada bulan ke-1 adalah 1
```