

JOBSHEET 12

Fungsi Rekursif

1. Tujuan

- Mahasiswa memahami konsep fungsi rekursif
- Mahasiswa mampu mengimplementasikan fungsi rekursif dalam kode program

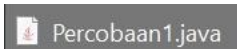
2. Praktikum

2.1 Percobaan 1

Waktu Percobaan : 60

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.

1. Buat project baru bernama Rekursif, dan buat file Java 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 = faktor * i;  
    }  
}
```

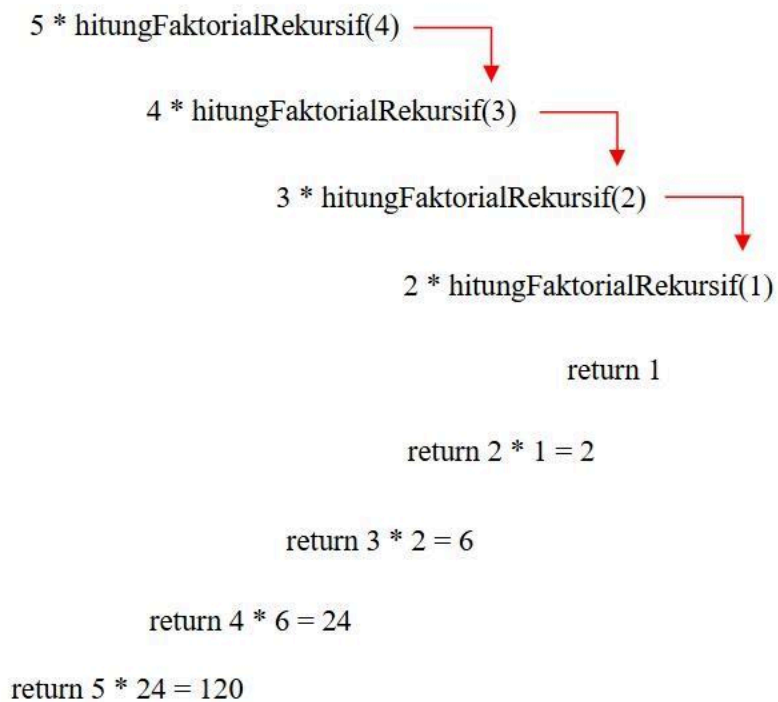
```
    return faktor;  
}
```

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

5. Jalankan program tersebut. Amati apa yang terjadi!

```
desws_b3765\jdt_ws\jdt.1s  
120  
120  
PS C:\Users\WINDOWS 11>
```

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



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 Pertanyaan

1. Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri secara langsung atau tidak langsung untuk menyelesaikan suatu masalah.

2. Menghitung faktorial, Mencari Angka Fibonacci

3. iya sama,

Perbedaan alur

1. Rekursif

- Fungsi memanggil dirinya sendiri dengan argumen yang lebih kecil hingga mencapai *base case* (kondisi $n == 0$ atau $n == 1$).
- `faktorialRekursif(5)` → memanggil `faktorialRekursif(4)`
- `faktorialRekursif(4)` → memanggil `faktorialRekursif(3)`
- `faktorialRekursif(3)` → memanggil `faktorialRekursif(2)`
- `faktorialRekursif(2)` → memanggil `faktorialRekursif(1)`
- `faktorialRekursif(1)` → memanggil `faktorialRekursif(0)` (mengembalikan 1, karena mencapai base case).

2. Iteratif (`faktorialIteratif`)

- Fungsi menggunakan perulangan untuk menghitung faktorial secara langsung.
- perulangan menggunakan `for`
- deklarasi `int faktor = 1` untuk base case
- deklarasi inialisasi `int i = n`, perulangan akan terus berjalan selama `i` lebih dari atau sama dengan 1 lalu akan di `decrement(i--)`
- `faktor = faktor dikali i` (selama perulangan)
- setelah perulangan selesai akan di `return`

2.2 Percobaan 2

Waktu Percobaan : 60

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

1. Pada project Rekursif, dan buat file Java dengan nama Percobaan2



2. 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.

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

3. Buatlah fungsi main dan deklarasikan Scanner dengan nama sc

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

4. Buatlah dua buah variabel bertipe int dengan nama bilangan dan pangkat

```
int bilangan, pangkat;
```

5. Tambahkan kode berikut ini untuk menerima input dari keyboard

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

6. Lakukan pemanggilan fungsi hitungPangkat yang telah dibuat sebelumnya dengan mengirimkan dua nilai parameter.

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

7. Jalankan program tersebut. Amati apa yang terjadi!

```
Bilangan yang dihitung: 4  
Pangkat: 2  
16
```

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 2x2x2x2x2x1 = 32

Jawaban Pertanyaan

1. Fungsi ini menggunakan konsep rekursif, yaitu memanggil dirinya sendiri dengan nilai parameter yang berubah setiap kali.

Kondisi dasar (*base case*): Ketika parameter **pangkat** bernilai **0**, fungsi akan berhenti memanggil dirinya sendiri dan mengembalikan nilai **1**.

2.

```
import java.util.Scanner;

public class PangkatRekursif {

    static int hitungPangkat(int bilangan, int pangkat, StringBuilder deret) {
        if (pangkat == 0) {
            deret.append("1");
            return 1;
        } else {
            deret.append(bilangan);
            if (pangkat > 1) {
                deret.append("x");
            }
            return bilangan * hitungPangkat(bilangan, pangkat - 1, deret);
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Masukkan bilangan: ");
```

```

    int bilangan = sc.nextInt();

    System.out.print("Masukkan pangkat: ");

    int pangkat = sc.nextInt();

    StringBuilder deretPerhitungan = new StringBuilder();

    int hasil = hitungPangkat(bilangan, pangkat, deretPerhitungan);

    System.out.println("Perhitungan: " + deretPerhitungan + " = " + hasil);

}
}

```

2.3 Percobaan 3

Waktu Percobaan : 60

Pada percobaan ini akan dilakukan pembuatan program untuk menghitung jumlah uang investor yang digunakan sebagai investasi setelah mendapatkan laba selama beberapa tahun dengan menggunakan fungsi rekursif.

1. Pada project Rekursif, dan buat file Java dengan nama Percobaan3



2. Buat fungsi static dengan nama hitungLaba(), dengan tipe data kembalian fungsi double dan memiliki 2 parameter dengan tipe data int berupa saldo investor dan lamanya investasi.

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

```

static double hitungLaba(double saldo, int tahun) {
    if (tahun == 0) {

```

```

        return (saldo);
    } else {
        return (1.11 * hitungLaba(saldo, tahun - 1));
    }
}

```

3. Buatlah fungsi main dan deklarasikan Scanner dengan nama sc

```

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

```

4. Buatlah sebuah variabel bertipe double dengan nama saldoAwal dan sebuah variabel bertipe int bernama tahun

```

double saldoAwal;
int tahun;

```

5. Tambahkan kode berikut ini untuk menerima input dari keyboard

```

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

```

6. Lakukan pemanggilan fungsi hitungLaba yang telah dibuat sebelumnya dengan mengirimkan dua nilai parameter.

```

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

```

7. Jalankan program tersebut. Amati apa yang terjadi!

```

Jumlah saldo awal : 100000
Lamanya investasi (tahun) : 2
Jumlah saldo setelah 2 tahun : 123210.000000000003

```

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 Pertanyaan

1. base casenya adalah tahun = , refusionalnya adalah `return (1.11 * hitungLaba(saldo, tahun - 1));`

2.

Trace Fase Ekspansi dan Substitusi

Input:

- `saldoAwal = 100000`
- `tahun = 3`

Fase Ekspansi (Proses pemanggilan rekursif):

1. **Pemanggilan pertama:** `hitungLaba(100000, 3)`
 - $\text{saldo} = 1.11 \times \text{hitungLaba}(100000, 2)$
 $\text{saldo} = 1.11 \times \text{hitungLaba}(100000, 2)$
2. **Pemanggilan kedua:** `hitungLaba(100000, 2)`
 - $\text{saldo} = 1.11 \times \text{hitungLaba}(100000, 1)$
 $\text{saldo} = 1.11 \times \text{hitungLaba}(100000, 1)$
3. **Pemanggilan ketiga:** `hitungLaba(100000, 1)`
 - $\text{saldo} = 1.11 \times \text{hitungLaba}(100000, 0)$
 $\text{saldo} = 1.11 \times \text{hitungLaba}(100000, 0)$
4. **Pemanggilan keempat:** `hitungLaba(100000, 0)`
 - $\text{saldo} = 100000$
 $\text{saldo} = 100000$ (nilai dasar, tidak ada lagi pemanggilan rekursif)

Fase Substitusi (Proses pengembalian hasil rekursif):

- Setelah mencapai `hitungLaba(100000, 0)`, kita mulai menggantikan setiap pemanggilan rekursif dengan hasil yang sesuai.
1. **Pemanggilan keempat:** `hitungLaba(100000, 0)` mengembalikan **100000**.
 2. **Pemanggilan ketiga:** `hitungLaba(100000, 1)` mengembalikan:
 $1.11 \times 100000 = 111000$
 $1.11 \times 100000 = 111000$
 3. **Pemanggilan kedua:** `hitungLaba(100000, 2)` mengembalikan:
 $1.11 \times 111000 = 123210$
 $1.11 \times 111000 = 123210$
 4. **Pemanggilan pertama:** `hitungLaba(100000, 3)` mengembalikan:
 $1.11 \times 123210 = 136369.1$
 $1.11 \times 123210 = 136369.1$

Hasil Akhir:

Saldo setelah 3 tahun adalah **136369.1**.

3. Tugas

Waktu Pengerjaan 120 menit

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

1.

```
import java.util.Scanner;
```

```
public class TugasNo1 {
```

```

static int Iteratif(int n) {
    for(int i = n; i >= 0; i--) {
        System.out.println(i);
    }
    return(n);
}

static int Rekursif(int n) {
    if (n < 0) {
        return(n-n);
    } else {
        return(n * Rekursif(n- 1));
    }
}

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

    System.out.print("Berapa banyak n? : ");

    int bnykN = sc.nextInt();

    Rekursif(bnykN);

    Iteratif(bnykN);
}
}

```

2.

```
import java.util.Scanner;
```

```

public class TugasNo2 {

    static int Rekursif(int n) {

        if (n == 1) {

            return 1;

        } else {

            return n + Rekursif(n - 1);

        }

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("nilai x: ");

        int x = sc.nextInt();

        if (x < 1) {

            System.out.println("x harus lebih besar dari 0 : ");

        } else {

            int hassilll = Rekursif(x);

            System.out.println("Penjumlahan bilangan dari 1 hingga " + f + "
adalah: " + hasil);

        }

    }

}

```

3.

```

import java.util.Scanner;

public class TugasNo3 {

    static int jumlahPasangan(int bulan) {

```

```
        if (bulan == 1 || bulan == 2) {  
            return 1;  
        } else {  
            return jumlahPasangan(bulan - 1) + jumlahPasangan(bulan - 2);  
        }  
    }  
}  
  
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
  
    System.out.print("Masukkan bulan ke berapa: ");  
  
    int bulanKe = scanner.nextInt();  
  
    if (bulanKe < 1) {  
        System.out.println("Bulan harus lebih besar dari 0.");  
    } else {  
        int totalPasangan = jumlahPasangan(bulanKe);  
  
        System.out.println("Jumlah pasangan marmut pada bulan ke-" + bulanKe  
+ " adalah: " + totalPasangan);  
    }  
}  
}
```