

Percobaan 1

1. Buat file dengan nama **Percobaan1**
2. Buat class lalu buat fungsi static **faktorialRekursif()**

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

3. Buat fungsi static **faktorialIteratif()**

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

4. Buat fungsi main dan panggil fungsi yang telah dibuat

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

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. Rekursif adalah pemanggilan fungsi di dalam fungsi tersebut

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

2. dalam hal ini, terdapat pemanggilan fungsi dalam fungsi itu sendiri

3. Hasilnya sama, berikut alur pada **faktorialRekursif(5)**:

$$\begin{aligned}
 \text{a. faktorialRekursif}(5) &= 5 * \text{faktorialRekursif}(4) \\
 &= 5 * (4 * \text{faktorialRekursif}(3)) \\
 &= 5 * (4 * (3 * \text{faktorialRekursif}(2))) \\
 &= 5 * (4 * (3 * (2 * \text{faktorialRekursif}(1)))) \\
 &= 5 * (4 * (3 * (2 * (1 * \text{faktorialRekursif}(0)))))
 \end{aligned}$$

$$\begin{aligned}
 \text{b. faktorialIteratif}(5) &= \text{faktor} = \text{faktor} * 5 \\
 &= \text{faktor} = 1 * 5 \\
 &= \text{faktor} = 5 * 4 \\
 &= \text{faktor} = 20 * 3 \\
 &= \text{faktor} = 60 * 2 \\
 &= \text{faktor} = 120 * 1
 \end{aligned}$$

Percobaan 2

1. buat file baru dengan nama **Percobaan2**
2. import Scanner, lalu buat class
3. Buat fungsi **hitungPangkat()** dengan parameter integer

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

4. Buat fungsi main, dan deklarasikan Scanner, int bilangan, dan int pangkat

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

5. Buat program untuk masukkan

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

6. Lakukan pemanggilan fungsi

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

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 $2 \times 2 \times 2 \times 2 \times 1 = 32$

Jawaban

1. Proses pemanggilan akan berhenti jika nilai dari pangkat mencapai 0

```

static int tampilan(int x, int y){
    int temp = x;
    if (y == 0){
        temp = 1;
    }else{
        temp = tampilan(x, y - 1);
        System.out.print(x + "x");
    }
    return temp;
}

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

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

    hasil = hitungPangkat(bilangan, pangkat);
    System.out.println(tampilan(bilangan, pangkat) + "=" + hasil);
}

```

2.

Percobaan 3

1. Buat file baru dengan nama **Percobaan3.java**
2. Import Scanner
3. Buat fungsi hitung laba dengan 2 parameter: double saldo,int tahun

```

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

```

4.

5. Buat fungsi main, lalu deklarasikan Scanner, double saldoAwal, dan int

```

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

```

tahun

6. Tambahkan kode untuk masukkan

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

7. Lakukan pemanggilan fungsi

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

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. Base call:

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

Recursion call:

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

2. Fase ekspansi:

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

Fase Substitusi:

$$\begin{aligned}&= 1.11 * (1.11 * (1.11 * 100000)) \\ &= 1.11 * (1.11 * 111000) \\ &= 1.11 * (123210) \\ &= 136763,1\end{aligned}$$

3. Tugas

Waktu Pengerjaan 120 menit

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

Team Teaching Dasar Pemrograman 2024
Politeknik Negeri Malang

5

Dasar Pemrograman 2024



11	34	55	89
12	55	89	144

```

1  import java.util.Scanner;
2  public class DeretDescendingRekursif {
3      static int deretRekursif(int n){
4          int temp;
5          if(n == 0){
6              temp = 0;
7          }else{
8              System.out.print(n + " ");
9              temp = (deretRekursif(n - 1));
10         }
11         return temp;
12     }
13
14     static void deretIteratif(int n){
15         for (int i = n; i >= 0; i--){
16             System.out.print(i + " ");
17         }
18     }
19
20     Run main | Debug main | Run | Debug
21     public static void main(String[] args) {
22         Scanner sc = new Scanner(System.in);
23         int angka;
24
25         System.out.print(s:"Masukkan angka awal: ");
26         angka = sc.nextInt();
27
28         System.out.print(deretRekursif(angka) + " Ini Rekursif");
29         System.out.println();
30         deretIteratif(angka);
31         System.out.print(s:"Ini Iteratif");
32     }
33

```

1.

```

Masukkan angka awal: 5
5 4 3 2 1 0Ini Rekursif
5 4 3 2 1 0 Ini Iteratif

```

```

1  import java.util.Scanner;
2  public class PenjumlahanRekursif {
3      static int jumlahRekursif(int n){
4          int temp = n;
5          if(n == 0){
6              temp = n;
7          }else{
8              temp += (jumlahRekursif(n - 1));
9          }
10         return temp;
11     }
12
13     static void tampilJumlah(int n){
14         if(n == 0){
15         }else{
16             tampilJumlah(n-1);
17             System.out.print(n + "+");
18         }
19     }
20
21     Run main | Debug main | Run | Debug
22     public static void main(String[] args) {
23         Scanner sc = new Scanner(System.in);
24         int angka;
25
26         System.out.print(s:"Masukkan Angka: ");
27         angka = sc.nextInt();
28         tampilJumlah(angka);
29         System.out.println( "\b=" + jumlahRekursif(angka));
30     }
31 }

```

2.

```

Masukkan Angka: 5
1+2+3+4+5=15

```



```

1
2  import java.util.Scanner;
3
4  public class Fibonacci {
5      static int hitungHamster(int bulan){
6          int hamster = 1;
7          if(bulan <= 1){
8              return bulan;
9          }
10         return hitungHamster(bulan-1) + hitungHamster(bulan-2);
11     }
12
13     Run main | Debug main | Run | Debug
14     public static void main(String[] args) {
15         Scanner sc = new Scanner(System.in);
16         int bulan, hamster;
17
18         System.out.print(s:"Masukkan berapa bulan pembiakan: ");
19         bulan = sc.nextInt();
20         System.out.println(hitungHamster(bulan));
21     }
22

```

3.