



JOBSHEET 11

Fungsi Rekursif

Nama: Rizqi Bagus Andrean

Kelas: TI-1D

Absen: 25

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));  
    }  
}
```

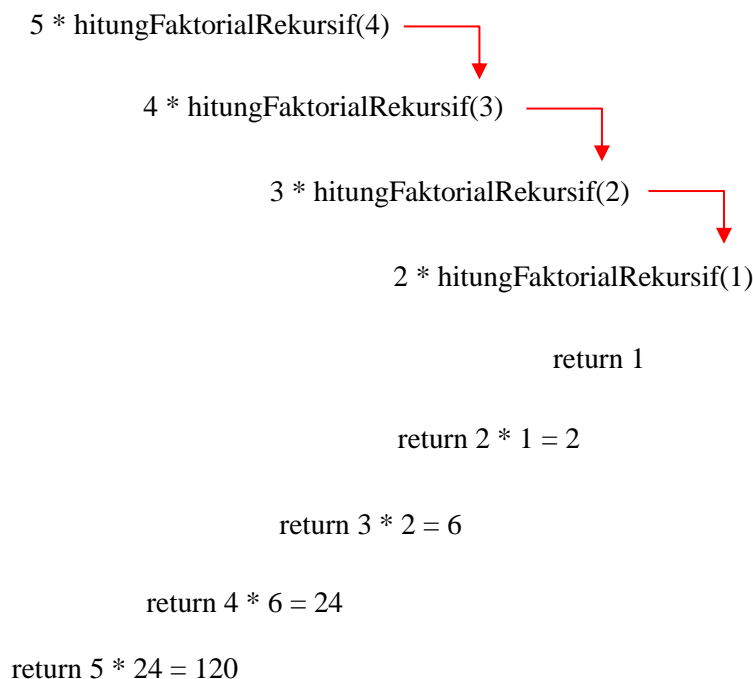
3. Buat lagi fungsi static dengan nama **faktorialIteratif()**, dengan tipe data kembalian fungsi **int** dan memiliki 1 parameter dengan tipe data **int** berupa bilangan yang akan dihitung nilai faktorialnya.

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

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

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

5. Jalankan program tersebut. Amati apa yang terjadi!
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?
Fungsi yang memanggil dirinya sendiri
2. Bagaimana contoh kasus penggunaan fungsi rekursif ?
Bisa digunakan untuk menghitung 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!

Sama,

Fungsi rekursif menggunakan proses pemanggilan diri sendiri secara berulang. Sedangkan, Fungsi iteratif menggunakan proses perulangan.

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

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!

Pada **Percobaan2**, proses pemanggilan fungsi `hitungPangkat()` akan dijalankan sampai kondisi dasar terpenuhi. Kondisi dasar tersebut adalah ketika nilai `pangkat` sama dengan 0. Pada saat itu, fungsi `hitungPangkat()` akan mengembalikan nilai 1, yang merupakan hasil dari bilangan pokok yang dipangkatkan dengan 0.

Jika nilai `pangkat` tidak sama dengan 0, maka fungsi `hitungPangkat()` akan melakukan pemanggilan dirinya sendiri dengan nilai `pangkat` yang berkurang 1. Hal ini akan terus dilakukan sampai nilai `pangkat` menjadi 0.

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

```
1 import java.util.Scanner;
2
3 public class Percobaan2 {
4
5
6     static String ctk = "";
7
8     public static void main(String[] args) {
9         int bilangan, pangkat;
10        Scanner sc = new Scanner(System.in);
11        System.out.print("Bilangan yang dihitung: ");
12        bilangan = sc.nextInt();
13        System.out.print("Pangkat: ");
14        pangkat = sc.nextInt();
15
16        System.out.println(hitungPangkat(bilangan, pangkat));
17        System.out.println(ctk + "1 = " + hitungPangkat(bilangan, pangkat));
18
19        sc.close();
20    }
21
22    static int hitungPangkat(int x, int y) {
23
24        if (y == 0) {
25            return 1;
26        } else {
27            ctk += x + " x ";
28            return x * hitungPangkat(x, y-1);
29        }
30    }
31
32
33
34
35
36 }
37
```

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** * **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**
4. Buatlah sebuah variabel bertipe double dengan nama **saldoAwal** dan sebuah variabel bertipe int bernama **tahun**
5. Tambahkan kode berikut ini untuk menerima input dari keyboard

```
System.out.print(s:"Jumlah saldo awal : ");
saldoAwal = sc.nextInt();
System.out.print(s:"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(d:hitungLaba(saldo:saldoAwal, tahun));
```

7. Jalankan program tersebut. Amati apa yang terjadi!

Pertanyaan

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

```
2. System.out.print(hitungLaba(saldo, tahun))
```

Ini adalah pemanggilan fungsi rekursif pertama kali.

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

Pada fase ekspansi, fungsi **hitungLaba()** akan terus melakukan pemanggilan diri sendiri sampai kondisi dasar terpenuhi. Pada kasus **hitungLaba(100000,3)**, proses ekspansi akan berlangsung sebagai berikut:

```
Tahun | Saldo
-----|-----
```



```
1 | 100000
2 | 111000
3 | 122110
```

Pada tahun pertama, fungsi `hitungLaba()` akan dipanggil dengan nilai saldo awal sebesar 100.000 dan jumlah tahun sebesar 2. Fungsi ini akan menghitung bunga untuk tahun pertama, yaitu $100.000 * 0,111 = 11.100$. Kemudian, fungsi akan mengembalikan nilai saldo akhir, yaitu $100.000 + 11.100 = 111.000$.

Pada tahun kedua, fungsi `hitungLaba()` akan dipanggil dengan nilai saldo awal sebesar 111.000 dan jumlah tahun sebesar 1. Fungsi ini akan menghitung bunga untuk tahun kedua, yaitu $111.000 * 0,111 = 12.211$. Kemudian, fungsi akan mengembalikan nilai saldo akhir, yaitu $111.000 + 12.211 = 123.211$.

Pada tahun ketiga, fungsi `hitungLaba()` akan dipanggil dengan nilai saldo awal sebesar 123.211 dan jumlah tahun sebesar 0. Fungsi ini akan mencapai kondisi dasar, karena jumlah tahun telah menjadi 0. Oleh karena itu, fungsi akan mengembalikan nilai saldo, yaitu 123.211.

3. Tugas

Waktu Pengerjaan 120 menit

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

```
1  /**
2   * TugasIndividu1
3   */
4  public class TugasIndividu1 {
5
6      public static void main(String[] args) {
7          System.out.println(deretDescRekursif(5));
8          System.out.print(deretDescIteratif(5));
9      }
10
11     static int deretDescRekursif(int n) {
12         if (n == 0) {
13             return 0;
14         } else {
15             System.out.print(n + " ");
16             return deretDescRekursif(n-1);
17         }
18     }
19
20     static int deretDescIteratif(int n) {
21         for (int i = n; i >= 1; i--) {
22             System.out.print(i + " ");
23         }
24
25         return 0;
26
27     }
28 }
29
30
31 }
```

```
Acer\AppData\Roam
5 4 3 2 1 0
5 4 3 2 1 0
PS C:\Users\Acer\T
```

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  public class TugasIndividu2 {
2
3      private static String ctk = "";
4      private static int dummy = 0;
5
6      public static void main(String[] args) {
7          System.out.print(hitungPenjumlahan(8));
8      }
9
10     static int hitungPenjumlahan(int n) {
11         dummy++;
12         if (n == 0) {
13             System.out.print(ctk);
14             return 0;
15         } else {
16
17             if (n == 1) {
18                 ctk += dummy + " = ";
19             } else {
20                 ctk += dummy + " + ";
21             }
22             return n + hitungPenjumlahan(n-1);
23         }
24     }
25 }
26

```

```

PS C:\Users\Acer> cd C:\Users\Acer\Tugas Kuliah\Semester 1\Praktek Daspro\jo
Acer\AppData\Roaming\Code\User\workspaceStorage\93d2735d9c
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36
PS C:\Users\Acer\Tugas Kuliah\Semester 1\Praktek Daspro\jo

```

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**).

```

1  public class TugasIndividu3 {
2      public static void main(String[] args) {
3          System.out.println(cekPrimaRekursive(13));
4      }
5
6      static boolean cekPrimaRekursive(int n) {
7          if (n <= 1) {
8              return false;
9          } else if (n == 2) {
10             return true;
11          } else {
12              for (int i = 2; i * i <= n; i++) {
13                  if (n % i == 0) {
14                      return false;
15                  }
16              }
17              return true;
18          }
19      }
20  }
21  }
22

```

```

s\program_63bfd992\bin\ 'TugasIndividu3'
true
PS C:\Users\Acer\Tugas_Kuliah\Semester 1\Praktek Dasar

```

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

```

1  public class TugasIndividu4 {
2      public static void main(String[] args) {
3
4          for (int bulan = 1; bulan <=12; bulan ++) {
5              if (bulan == 1 || bulan == 2 ) {
6                  System.out.println("Bulan ke-" + bulan + " : " + 1);
7                  continue;
8              }
9              int total = fibonanci(bulan-1);
10             System.out.println("Bulan ke-" + bulan + " : " + total);
11         }
12     }
13
14     static int fibonanci(int n) {
15         if (n == 0 || n == 1) {
16             return 1;
17         } else {
18             return fibonanci(n - 1) + fibonanci(n - 2);
19         }
20     }
21 }
22
23

```

```

Acer\AppData\Roaming\Code\User\Workspaces\
Bulan ke-1 : 1
Bulan ke-2 : 1
Bulan ke-3 : 2
Bulan ke-4 : 3
Bulan ke-5 : 5
Bulan ke-6 : 8
Bulan ke-7 : 13
Bulan ke-8 : 21
Bulan ke-9 : 34
Bulan ke-10 : 55
Bulan ke-11 : 89
Bulan ke-12 : 144
PS C:\Users\Acer\Tugas Kuliah\Semester 1\Pr

```