

Nama : Sheilandra Zarawiba

Kelas : TI-1D

NIM : 2441007020110

Absen : 28

## JOBSHEET 12

### Fungsi Rekursif

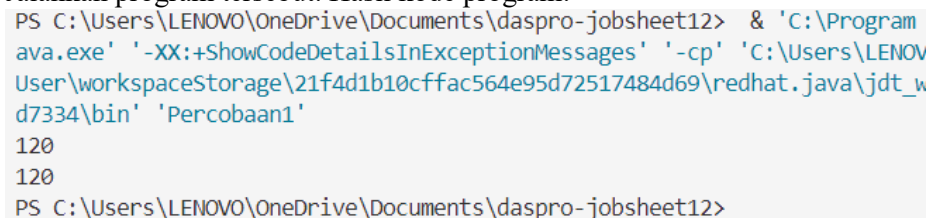
#### Percobaan 1

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



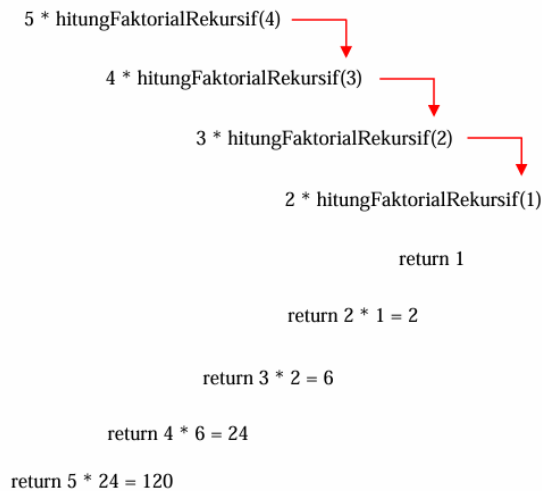
```
J Percobaan1.java U X
J Percobaan1.java > Language Support for Java(TM) by Red Hat > Percobaan1
1 public class Percobaan1 {
2     static int faktorialRekursif(int n) {
3         if (n == 0) {
4             return (1);
5         } else {
6             return (n * faktorialRekursif(n-1));
7         }
8     }
9
10    static int faktorialIteratif(int n) {
11        int faktor = 1;
12        for (int i = n; i >= 1; i-- ) {
13            faktor = faktor * i;
14        }
15        return faktor;
16    }
17
18    public static void main(String[] args) {
19        System.out.println(faktorialRekursif(n:5));
20        System.out.println(faktorialIteratif(n:5));
21    }
22 }
```

5. Jalankan program tersebut. Hasil kode program:



```
PS C:\Users\LENOVO\OneDrive\Documents\daspro-jobsheet12> & 'C:\Program
ava.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\LENOV
User\workspaceStorage\21f4d1b10cffac564e95d72517484d69\redhat.java\jdt_w
d7334\bin' 'Percobaan1'
120
120
PS C:\Users\LENOVO\OneDrive\Documents\daspro-jobsheet12>
```

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?

Jawab:

Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri dalam proses eksekusinya hingga mencapai kondisi dasar (*base case*). Fungsi ini digunakan untuk menyelesaikan masalah yang dapat dipecah menjadi sub-masalah yang serupa dengan struktur yang lebih kecil.

2. Bagaimana contoh kasus penggunaan fungsi rekursif?

Jawab:

Contoh kasus penggunaan fungsi rekursif adalah menghitung faktorial. Misalnya, untuk menghitung  $5!5!5!$ , kita bisa memecahnya menjadi  $5 \times 4!5 \times 4!5 \times 4!$ , lalu  $4 \times 3!4 \times 3!4 \times 3!$ , hingga kondisi dasar  $1! = 1! = 1! = 1$ . Rekursi juga sering digunakan untuk masalah lain seperti bilangan Fibonacci, di mana untuk menghitung  $F(5)F(5)F(5)$ , fungsi akan memanggil  $F(4)F(4)F(4)$  dan  $F(3)F(3)F(3)$ , lalu  $F(3)F(3)F(3)$  akan memanggil  $F(2)F(2)F(2)$ , hingga kondisi dasar  $F(1) = 1F(1) = 1F(1) = 1$ . Selain itu, rekursi juga cocok digunakan untuk mencetak deret menurun. Sebagai contoh, jika ingin mencetak angka dari 5 ke 0 (yaitu 5, 4, 3, 2, 1, 0), kita bisa memanggil fungsi untuk mencetak angka  $nnn$ , lalu memanggil dirinya sendiri dengan nilai  $n-1n-1n-1$ , hingga mencapai angka 0 sebagai kondisi dasar.

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!

Jawab:

Ya, hasil dari fungsi faktorialRekursif() dan faktorialIteratif() adalah sama, misalnya untuk  $5!5!5!$ , keduanya menghasilkan 120120120. Namun, cara kerjanya berbeda. Pada rekursif, fungsi memanggil dirinya sendiri berulang kali hingga mencapai kondisi dasar ( $n=0n=0n=0$ ), kemudian menghitung hasilnya secara berbalik dari bawah ke atas. Di sisi lain, pada iteratif, hasil dihitung langsung menggunakan perulangan, dimulai dari angka 1 hingga  $nnn$ , dengan mengalikan angka satu per satu. Perbedaan utama adalah bahwa rekursif menggunakan lebih banyak memori karena setiap pemanggilan fungsi disimpan di stack, sementara iteratif lebih sederhana dalam penggunaan memori dan cenderung lebih cepat.

## Percobaan 2

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

```
J Percobaan2.java > Java > Percobaan2
1  import java.util.Scanner;
2
3  public class Percobaan2 {
4      static int hitungPangkat (int x, int y) {
5          if (y == 0) {
6              return (1);
7          } else {
8              return (x * hitungPangkat(x, y-1));
9          }
10     }
11
12     Run main | Debug main | Run | Debug
13     public static void main(String[] args) {
14         Scanner sc = new Scanner(System.in);
15         int bilangan, pangkat;
16
17         System.out.print(s:"Bilangan yang dihitung: ");
18         bilangan = sc.nextInt();
19         System.out.print(s:"Pangkat: ");
20         pangkat= sc.nextInt();
21
22         System.out.println(hitungPangkat(bilangan, pangkat));
23     }
```

7. Jalankan program tersebut. Amati apa yang terjadi!

```
PROBLEMS 1 DEBUG CONSOLE OUTPUT TERMINAL PORTS
PS C:\Users\LENOVO\OneDrive\Documents\Rekursif> & 'C:\Program Files\Java\jdk-11.0.10\bin\java.exe' -XX:+ShowCodeDetailsInExceptionMessages -cp 'C:\Users\LENOVO\OneDrive\Documents\Rekursif\src' -Djava.class.path=. 'C:\Users\LENOVO\OneDrive\Documents\Rekursif\src\Percobaan2.java'
Bilangan yang dihitung: 6
Pangkat: 2
36
PS C:\Users\LENOVO\OneDrive\Documents\Rekursif>
```

## 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!

Jawab:

Proses pemanggilan fungsi rekursif `hitungPangkat(bilangan, pangkat)` pada kode akan terus berjalan hingga kondisi dasar (*base case*) tercapai, yaitu ketika nilai pangkat sama dengan 0. Pada kondisi ini, fungsi akan berhenti memanggil dirinya sendiri dan mengembalikan nilai 1. Setiap pemanggilan fungsi mengurangi nilai pangkat sebesar 1, sehingga proses ini akan selesai setelah semua langkah rekursi selesai. Sebagai contoh, untuk input `hitungPangkat(2, 5)`, pemanggilan berlanjut sebagai  $2 \times \text{hitungPangkat}(2, 4)$   $\times$   $\text{hitungPangkat}(2, 4)$   $\times$   $\text{hitungPangkat}(2, 4)$ ,  $2 \times \text{hitungPangkat}(2, 3)$   $\times$   $\text{hitungPangkat}(2, 3)$   $\times$   $\text{hitungPangkat}(2, 3)$ , dan seterusnya hingga mencapai `hitungPangkat(2, 0)` `hitungPangkat(2, 0)` `hitungPangkat(2, 0)`, yang mengembalikan 1.

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

Jawab:

```
Percobaan2.java > Language Support for Java(TM) by Red Hat > Percobaan2
1  import java.util.Scanner;
2
3  public class Percobaan2 {
4      static int hitungPangkat (int x, int y) {
5          if (y == 0) {
6              return 1;
7          } else {
8              System.out.print(x);
9              if (y > 1) {
10                 System.out.print("x");
11             }
12             return (x * hitungPangkat(x, y-1));
13         }
14     }
15
16     Run | Debug | Run main | Debug main
17     public static void main(String[] args) {
18         Scanner sc = new Scanner(System.in);
19         int bilangan, pangkat;
20
21         System.out.print(s:"Bilangan yang dihitung: ");
22         bilangan = sc.nextInt();
23         System.out.print(s:"Pangkat: ");
24         pangkat = sc.nextInt();
25
26         int hasil = hitungPangkat(bilangan, pangkat);
27         System.out.println(" = " + hasil);
28     }
29 }
```

Hasil kode program:

```
PS C:\Users\LENOVO\OneDrive\Documents\Rekursif> & 'C:\Pr
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\
spaceStorage\c35533570616f43280b224a018359e6d\redhat.java\
aan2'
Bilangan yang dihitung: 2
Pangkat: 5
2x2x2x2x2 = 32
PS C:\Users\LENOVO\OneDrive\Documents\Rekursif> █
```

### Percobaan 3

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

```
J Percobaan3.java > ...
1  import java.util.Scanner;
2
3  public class Percobaan3 {
4      static double hitungLaba(double saldo, int tahun) {
5          if (tahun == 0) {
6              return (saldo);
7          } else {
8              return (1.11 * hitungLaba(saldo, tahun - 1));
9          }
10     }
11
12     Run | Debug | Run main | Debug main
13     public static void main(String[] args) {
14         Scanner sc = new Scanner(System.in);
15         double saldoAwal;
16         int tahun;
17
18         System.out.print(s:"Jumlah saldo awal : ");
19         saldoAwal = sc.nextInt();
20         System.out.print(s:"Lamanya investasi (tahun) : ");
21         tahun = sc.nextInt();
22
23         System.out.print("Jumlah saldo setelah " + tahun + " tahun : ");
24         System.out.print(hitungLaba(saldoAwal, tahun));
25     }
26 }
```

7. Jalankan program tersebut. Amati apa yang terjadi!

```
PROBLEMS 2 DEBUG CONSOLE OUTPUT TERMINAL PORTS

PS C:\Users\LENOVO\OneDrive\Documents\Rekursif> & 'C:\Pr
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users
spaceStorage\c35533570616f43280b224a018359e6d\redhat.java
an3'
Jumlah saldo awal : 500000
Lamanya investasi (tahun) : 5
Jumlah saldo setelah 5 tahun : 842529.0775500003
PS C:\Users\LENOVO\OneDrive\Documents\Rekursif> █
```

## Pertanyaan

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

Jawab:

- 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)

Jawab:

- Fase ekspansi

1. hitungLaba(100000,3): memanggil 1.11×hitungLaba(100000,2)
2. hitungLaba(100000,2): memanggil 1.11×hitungLaba(100000,1)
3. hitungLaba(100000,1): memanggil 1.11×hitungLaba(100000,0)
4. hitungLaba(100000,0): base case tercapai, mengembalikan 100000.

- Fase substitusi

Dari hitungLaba(100000,0): hasilnya 100000.

Substitusi ke hitungLaba(100000,1):

$1.11 \times 100000 = 111000$ .

Substitusi ke hitungLaba(100000,2):

$1.11 \times 111000 = 123210$ .

Substitusi ke hitungLaba(100000,3):

$1.11 \times 123210 = 136753.1$ .

## Tugas

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

Jawab:

Kode program:

```
J DeretDescendingRekursif.java > ...  
1  import java.util.Scanner;  
2  
3  public class DeretDescendingRekursif {  
4      static void deretRekursif(int n) {  
5          if (n < 0) {  
6              return;  
7          }  
8          System.out.print(n + " ");  
9          deretRekursif(n - 1);  
10     }  
11  
12     static void deretIteratif(int n) {  
13         for (int i = n; i >= 0; i--) {  
14             System.out.print(i + " ");  
15         }  
16     }  
17  
18     Run | Debug | Run main | Debug main  
19     public static void main(String[] args) {  
20         Scanner sc = new Scanner(System.in);  
21  
22         System.out.print(s:"Masukkan bilangan n: ");  
23         int n = sc.nextInt();  
24  
25         System.out.println(x:"Deret menggunakan rekursif:");  
26         deretRekursif(n);  
27         System.out.println(x:"\nDeret menggunakan iteratif:");  
28         deretIteratif(n);  
29     }  
}
```

Hasil:

```
Masukkan bilangan n: 5
Deret menggunakan rekursif:
5 4 3 2 1 0
Deret menggunakan iteratif:
5 4 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).

Jawab:

Kode program:

```
PenjumlahanRekursif.java > Language Support for Java(TM) by Red Hat > PenjumlahanRekursif
1  import java.util.Scanner;
2
3  public class PenjumlahanRekursif {
4      static int hitungPenjumlahan(int n) {
5          if (n == 1) {
6              return 1;
7          }
8          return n + hitungPenjumlahan(n - 1);
9      }
10
11     static void cetakDeret(int n) {
12         if (n == 1) {
13             System.out.print(s:"1");
14             return;
15         }
16         cetakDeret(n - 1);
17         System.out.print(" + " + n);
18     }
19
20     Run | Debug | Run main | Debug main
21     public static void main(String[] args) {
22         Scanner sc = new Scanner(System.in);
23
24         System.out.print(s:"Masukkan bilangan f: ");
25         int f = sc.nextInt();
26
27         System.out.print(s:"Deret penjumlahan: ");
28         cetakDeret(f);
29         System.out.println(" = " + hitungPenjumlahan(f));
30     }
```

Hasil:

```
Masukkan bilangan f: 8
Deret penjumlahan: 1+2+3+4+5+6+7+8 = 36
```

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.

Jawab:

Kode program:

```
J Marmut.java > ...
1  public class Marmut {
2      static int jumlahMarmut(int bulan) {
3          if (bulan == 1 || bulan == 2) {
4              return 1;
5          }
6          return jumlahMarmut(bulan - 1) + jumlahMarmut(bulan - 2);
7      }
8  }

Run | Debug | Run main | Debug main
9  public static void main(String[] args) {
10     int bulan = 12;
11     int hasil = jumlahMarmut(bulan);
12     System.out.println("Jumlah pasangan marmut pada bulan ke-" + bulan + " adalah: " + hasil);
13 }
14 }
```

Hasil:

```
Jumlah pasangan marmut pada bulan ke-12 adalah: 144
```