Nama: Kamila Zahwa(13)

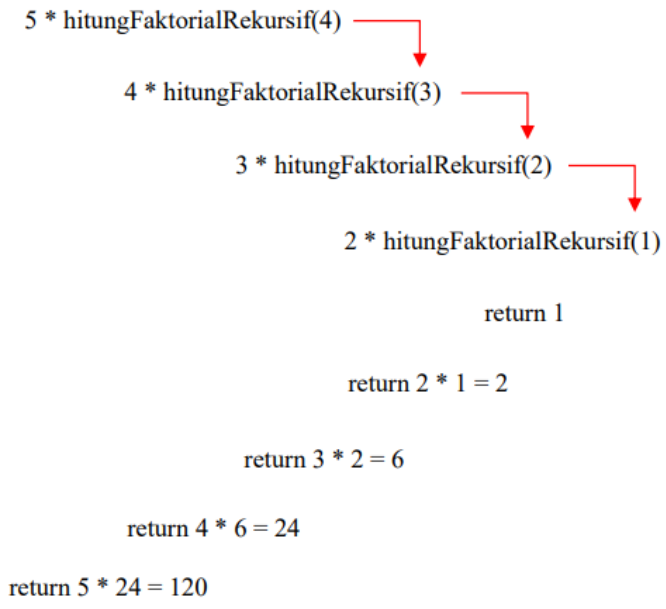
NIM: 244107020111

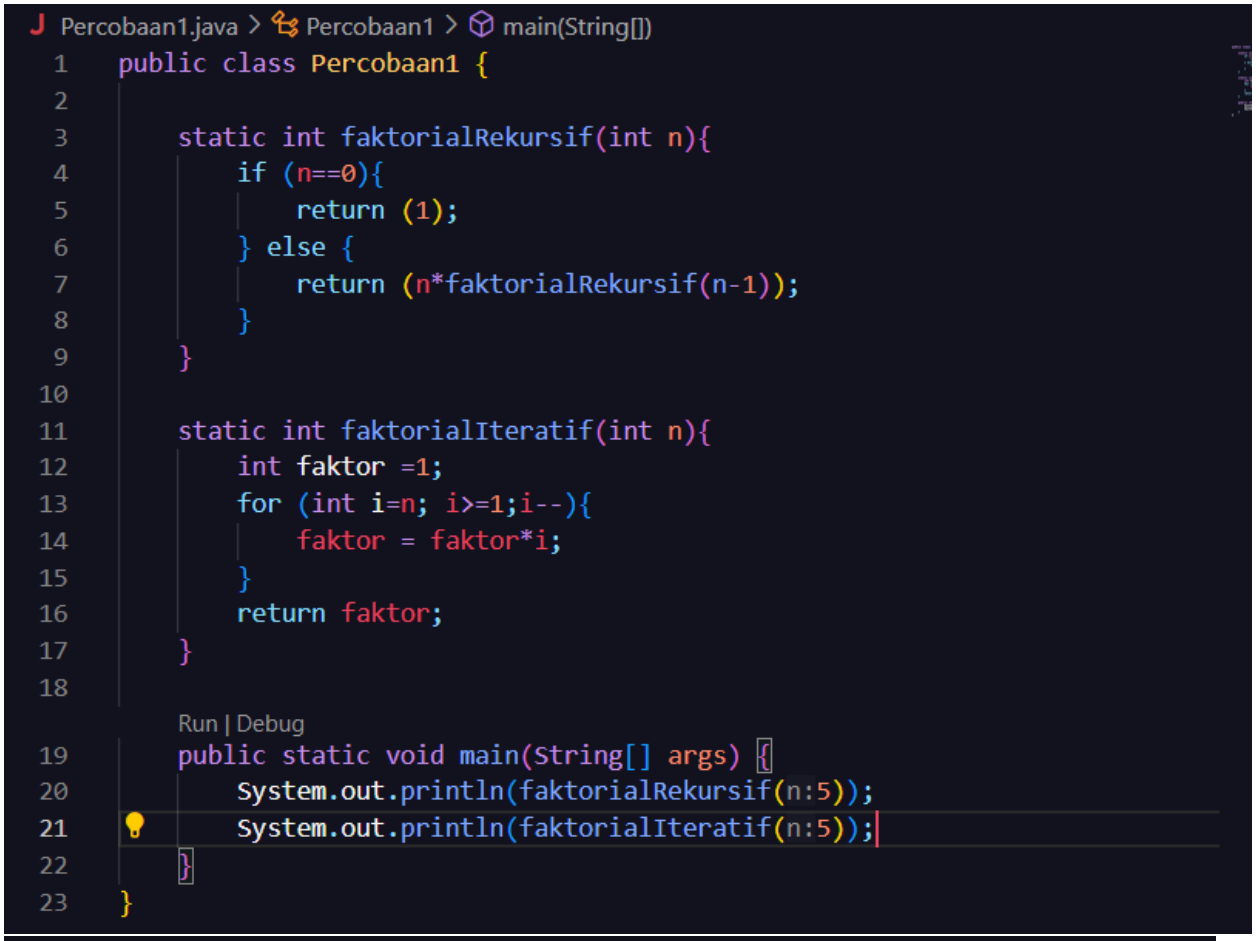
Kelas: TI 1D

JOBSHEET 12

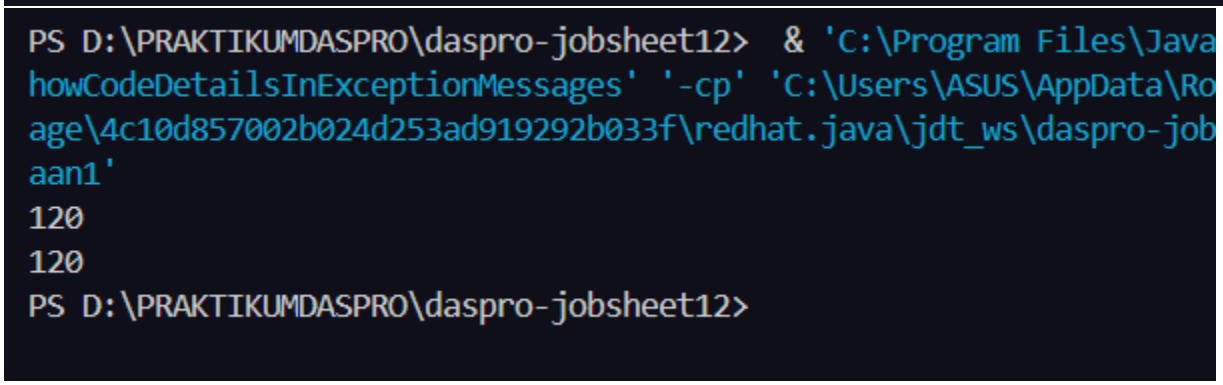
Percobaan 1

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.
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:





```
Percobaan1.java > Percobaan1 > main(String[])
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
        Run | Debug
19      public static void main(String[] args) {
20          System.out.println(faktorialRekursif(n:5));
21          System.out.println(faktorialIteratif(n:5));
22      }
23  }
```



```
PS D:\PRAKTIKUMDASPRO\daspro-jobsheet12>  & 'C:\Program Files\Java
howCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ASUS\AppData\Ro
age\4c10d857002b024d253ad919292b033f\redhat.java\jdt_ws\daspro-job
aan1'
120
120
PS D:\PRAKTIKUMDASPRO\daspro-jobsheet12>
```

Jawaban:

1. Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri untuk menyelesaikan suatu masalah. Biasanya, fungsi ini memiliki kondisi dasar (base case) yang menghentikan pemanggilan rekursif dan mencegah terjadinya infinite loop.
2. Contoh kasus penggunaan fungsi rekursif adalah perhitungan faktorial dari sebuah bilangan, di mana faktorial dari n (n!) didefinisikan sebagai n * (n-1)! dengan kondisi dasar 0! = 1.
3. Ya, hasil yang diberikan oleh faktorialRekursif() dan faktorialIteratif() sama. Perbedaan alur jalannya adalah fungsi rekursif memanggil dirinya sendiri berulang kali hingga mencapai

kondisi dasar, menyimpan status pemanggilan di stack. Sedangkan fungsi iteratif menggunakan loop (seperti for atau while) untuk menghitung hasil tanpa memanggil fungsi itu sendiri, sehingga tidak memerlukan stack tambahan.

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 bertipa 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.

Percobaan2.java > ...

```java
import java.util.Scanner;

public class Percobaan2 {
    static int hitungPangkat(int x, int y){
        if (y==0){
            return (1);
        } else {
            return (x*hitungPangkat(x, y-1));
        }
    }
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int bilangan, pangkat;
        System.out.print(s:"Bilangan yang dihitung: ");
        bilangan = sc.nextInt();
        System.out.print(s:"Pangkat: ");
        pangkat = sc.nextInt();

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

7. Jalankan program tersebut. Amati apa yang terjadi!

```
PS D:\PRAKTIKUMDASPRO\Rekursif>  & 'C:\Program Files
tailsInExceptionMessages' '-cp' 'C:\Users\ASUS\AppDa
9885da809a2d34e71a11033bf05\redhat.java\jdt_ws\Rekur
Bilangan yang dihitung: 2
Pangkat: 3
8
PS D:\PRAKTIKUMDASPRO\Rekursif>
```

Jawaban:

1. Proses pemanggilan fungsi hitungPangkat(bilangan, pangkat) akan terus dijalankan hingga nilai pangkat mencapai 0, di mana pada saat itu fungsi akan mengembalikan nilai 1 (karena setiap bilangan pangkat 0 adalah 1).
2.

Percobaan2.java > Percobaan2

```java
import java.util.Scanner;

public class Percobaan2 {
    static int hitungPangkat(int x, int y){
        if (y==0){
            return (1);
        } else {
            System.out.print(x + (y>1? "x" : "x1 = "));
            return (x*hitungPangkat(x, y-1));
        }
    }
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int bilangan, pangkat;
        System.out.print(s:"Bilangan yang dihitung: ");
        bilangan = sc.nextInt();
        System.out.print(s:"Pangkat: ");
        pangkat = sc.nextInt();

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

Output:

```
7c0b72a\bin' 'Percobaan2'
Bilangan yang dihitung: 2
Pangkat: 3
2x2x2x1 = 8
PS D:\PRAKTIKUMDASPRO\Rekursif>
```

Menambahkan x untuk memisahkan setiap bilangan hingga mencapai pangkat ke-0, di mana dicetak 1. Rekursi terakhir menambahkan x1 = sebelum hasil dikalkulasikan.

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 * saldo, dan saldo dianggap 1 * saldo, sehingga 1 * saldo + 0.11 * saldo dapat diringkas menjadi 1.11 * saldo untuk perhitungan saldo setelah ditambah laba (dalam setahun).
3. Buatlah fungsi main dan deklarasikan Scanner dengan nama sc
4. Buatlah sebuah variabel bertipa 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.

Percobaan3.java > Percobaan3 > main(String[])

```java
import java.util.Scanner;

public class Percobaan3 {
    static double hitungLaba(double saldo, int tahun){
        if(tahun == 0){
            return (saldo);
        } else {
            return (1.11*hitungLaba(saldo, tahun-1));
        }
    }

    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        double saldoAwal;
        int tahun;
        System.out.print(s:"Jumlah saldo awal: ");
        saldoAwal = sc.nextDouble();
        System.out.print(s:"Lamanya investasi (tahun): ");
        tahun = sc.nextInt();

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

7. Jalankan program tersebut. Amati apa yang terjadi!

```
rsif_47c0b72a\bin' 'Percobaan3'
Jumlah saldo awal: 1000000
Lamanya investasi (tahun): 2
Jumlah saldo setelah 2 tahun: 1232100.0
PS D:\PRAKTIKUMDASPRO\Rekursif>
```

Jawaban:

1. Base case adalah kondisi di mana investasi mencapai tahun 0, ditulis sebagai if (tahun == 0) return saldo;. Recursion call adalah pemanggilan fungsi hitungLaba(saldo, tahun - 1) yang akan menghitung laba untuk tahun sebelumnya.
2. Trace fase ekspansi dan fase subtitusi algoritma perhitungan laba nilai hitungLaba(100000,3)
   a) Fase ekspansi:
      hitungLaba(100000, 3) → 1.11 * hitungLaba(100000, 2)
      hitungLaba(100000, 2) → 1.11 * hitungLaba(100000, 1)
      hitungLaba(100000, 1) → 1.11 * hitungLaba(100000, 0)
      hitungLaba(100000, 0) → 100000 (base case)
   b) Fase substitusi:
      hitungLaba(100000, 1) → 1.11 * 100000 = 111000
      hitungLaba(100000, 2) → 1.11 * 111000 = 123210
      hitungLaba(100000, 3) → 1.11 * 123210 = 136763.1

Tugas

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

Tugas1.java > Tugas1 > main(String[])

```java
import java.util.Scanner;

public class Tugas1 {
    static void descendingRekursif(int n){
        if (n< 0){
            return;
        }
        System.out.print(n + " ");
        descendingRekursif(n-1);
    }

    static void descendingIteratif(int n){
        for (int i=n; i>=0; i--){
            System.out.print(i + " ");
        }
    }
```

```
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
23          System.out.println(x:"Menggunakan fungsi rekursif: ");
24          descendingRekursif(n);
25          System.out.println(x:"\nMenggunakan fungsi iteratif: ");
26          descendingIteratif(n);
27      }
28  }
29
```

Output:

```
rsif_47c0b72a\bin' 'Tugas1'
Masukkan nilai n: 7
Menggunakan fungsi rekursif:
7 6 5 4 3 2 1 0
Menggunakan fungsi iteratif:
7 6 5 4 3 2 1 0
PS D:\PRAKTIKUMDASPRO\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).

```
Tugas2.java > Tugas2 > main(String[])
1   import java.util.Scanner;
2
3   public class Tugas2 {
4       static int hitungPenjumlahan(int f){
5           if (f==0){
6               return 0;
7           }
8           return f + hitungPenjumlahan(f-1);
9       }
10
        Run | Debug
11      public static void main(String[] args) {
12          Scanner sc = new Scanner(System.in);
13          System.out.print(s:"Masukkan nilai f: ");
14          int f = sc.nextInt();
15
16          for (int i=1; i<=f; i++){
17              System.out.print(i + (i<f? "+" : " = "));
18          }
19
20          System.out.println(hitungPenjumlahan(f));
21      }
22  }
```

Output:

```
a\Roaming\Code\User\workspaceStorage\1d12b
rsif 47c0b72a\bin' 'Tugas2'
Masukkan nilai f: 8
1+2+3+4+5+6+7+8 = 36
PS D:\PRAKTIKUMDASPRO\Rekursif>
```

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

Percobaan1.java | Tugas1.java 1 | Tugas2.java 1 | Tugas3.java 1, U | Percobaan2.java 1 | Percobaan3.java 1

Tugas3.java > Tugas3 > main(String[])

```java
import java.util.Scanner;

public class Tugas3 {
    static int fibonacci(int bulan){
        if (bulan == 1 || bulan==2) {
            return 1;
        }
        return fibonacci(bulan-1) + fibonacci(bulan-2);
    }

    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Masukkan bulan ke-n: ");
        int bulan = sc.nextInt();

        int totalPasangan = fibonacci(bulan);
        System.out.println("Jumlah total pasangan marmut pada bulan ke-" + bulan + " adalah: " + totalPasangan);
    }
}
```

Output:

```
ers\ASUS\AppData\Roaming\Code\User\workspaceStorage\1d12b9883da809a
3'
Masukkan bulan ke-n: 12
Jumlah total pasangan marmut pada bulan ke-12 adalah: 144
PS D:\PRAKTIKUMDASPRO\Rekursif>
```