

# LAPORAN PRAKTIKUM

## Jobsheet 12

### Fungsi Rekursif

Nama : Sandy kurniawan

NIM : 244107020029

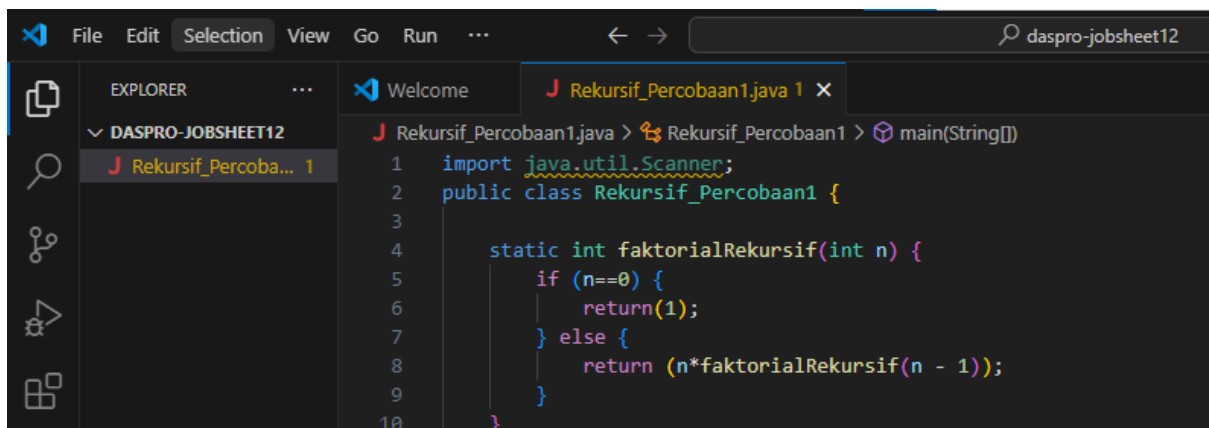
Kelas : 1D (27)

### PERCOBAAN1

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));  
    }  
}
```

### KODE PROGRAM



The screenshot shows an IDE window with the following structure:

- File Explorer: **DASPRO-JOBSHEET12** > **Rekursif\_Percobaan1.java**
- Editor: **Rekursif\_Percobaan1.java** > **Rekursif\_Percobaan1** > **main(String[])**

```
1  import java.util.Scanner;  
2  public class Rekursif_Percobaan1 {  
3  
4      static int faktorialRekursif(int n) {  
5          if (n==0) {  
6              return(1);  
7          } else {  
8              return (n*faktorialRekursif(n - 1));  
9          }  
10 }
```

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;  
}
```

#### KODE PROGRAM

```
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 }
```

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));  
}
```

#### KODE PROGRAM

```
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 }
```

5. Jalankan program tersebut. Amati apa yang terjadi!

```
PS D:\github\jobsheet12\daspro-jobsheet12>  
PS D:\github\jobsheet12\daspro-jobsheet12> d::; cd 'd:\github\jobsheet12\daspro-jobsheet12';  
sInExceptionMessages' '-cp' 'C:\Users\ACER\AppData\Roaming\Code\User\workspaceStorage\ab8cd60  
0f180\bin' 'Rekursif_Percobaan1'  
120  
1  
PS D:\github\jobsheet12\daspro-jobsheet12>  
Java: Ready
```

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

5 \* hitungFaktorialRekursif(4) →  
4 \* hitungFaktorialRekursif(3) →  
3 \* hitungFaktorialRekursif(2) →  
2 \* hitungFaktorialRekursif(1)  
return 1  
return 2 \* 1 = 2  
return 3 \* 2 = 6  
return 4 \* 6 = 24  
return 5 \* 24 = 120

## PERTANYAAN

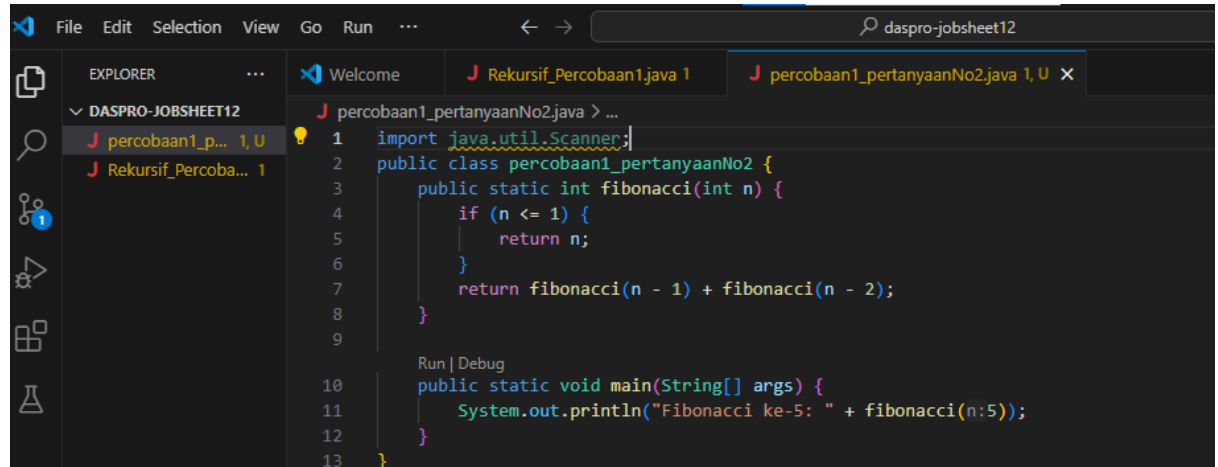
1. Apa yang dimaksud dengan fungsi rekursif?

fungsi yang memanggil dirinya sendiri untuk menyelesaikan suatu masalah. Rekursi digunakan untuk menyelesaikan masalah yang dapat dibagi menjadi sub-masalah yang lebih kecil dengan cara yang sama

2. Bagaimana contoh kasus penggunaan fungsi rekursif ?

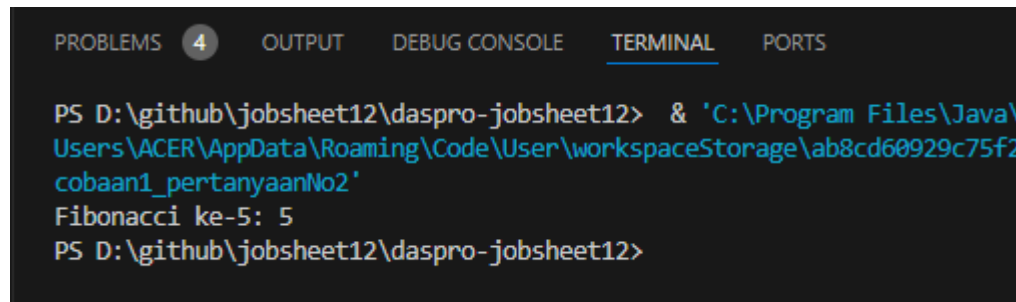
Fungsi rekursif sering digunakan untuk menghitung angka Fibonacci

## KODE PROGRAM



```
1 import java.util.Scanner;
2 public class percobaan1_pertanyaanNo2 {
3     public static int fibonacci(int n) {
4         if (n <= 1) {
5             return n;
6         }
7         return fibonacci(n - 1) + fibonacci(n - 2);
8     }
9
10    Run | Debug
11    public static void main(String[] args) {
12        System.out.println("Fibonacci ke-5: " + fibonacci(5));
13    }
```

## HASIL KETIKA DI RUN



```
PS D:\github\jobsheet12\daspro-jobsheet12> & 'C:\Program Files\Java\
Users\ACER\AppData\Roaming\Code\User\workspaceStorage\ab8cd60929c75f2
cobaan1_pertanyaanNo2'
Fibonacci ke-5: 5
PS D:\github\jobsheet12\daspro-jobsheet12>
```

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!

Hasilnya tidak sama, dalam loop faktorialIteratif, seharusnya mengalikan faktor dengan i, tetapi malah mengalikan dengan angka 1 saja, sehingga hasilnya selalu 1.

Fungsi faktorial rekursif, setiap pemanggilan fungsi memecah masalah menjadi lebih kecil, yaitu dengan mengurangi nilai n hingga mencapai nilai dasar (base case), yaitu  $n == 0$ . Sedangkan Fungsi iteratif untuk menghitung faktorial. Variabel faktor diinisialisasi dengan 1, dan kemudian dalam loop mengalikannya dengan i dari n hingga 1.

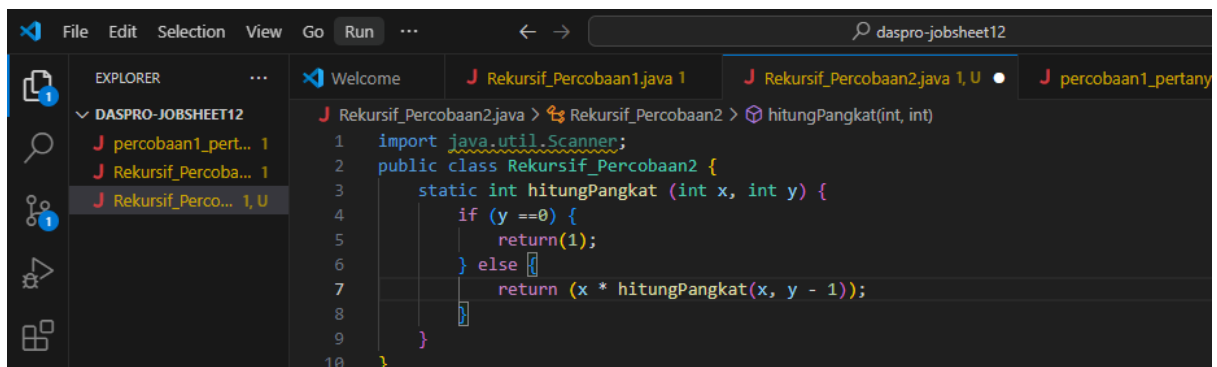
## PERCOBAAN2

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**

## KODE PROGRAM



```
File Edit Selection View Go Run ... daspro-jobsheet12  
EXPLORER  
DASPRO-JOBSHEET12  
  J percobaan1_pert... 1  
  J Rekursif_Percoba... 1  
  J Rekursif_Percoba... 1, U  
  J Rekursif_Percobaan2.java 1  
  J Rekursif_Percobaan2.java 1, U  
  J percobaan1_pertany...  
  J Rekursif_Percobaan2.java > Rekursif_Percobaan2 > hitungPangkat(int, int)  
1  import java.util.Scanner;  
2  public class Rekursif_Percobaan2 {  
3      static int hitungPangkat (int x, int y) {  
4          if (y ==0) {  
5              return(1);  
6          } else {  
7              return (x * hitungPangkat(x, y - 1));  
8          }  
9      }  
10 }
```

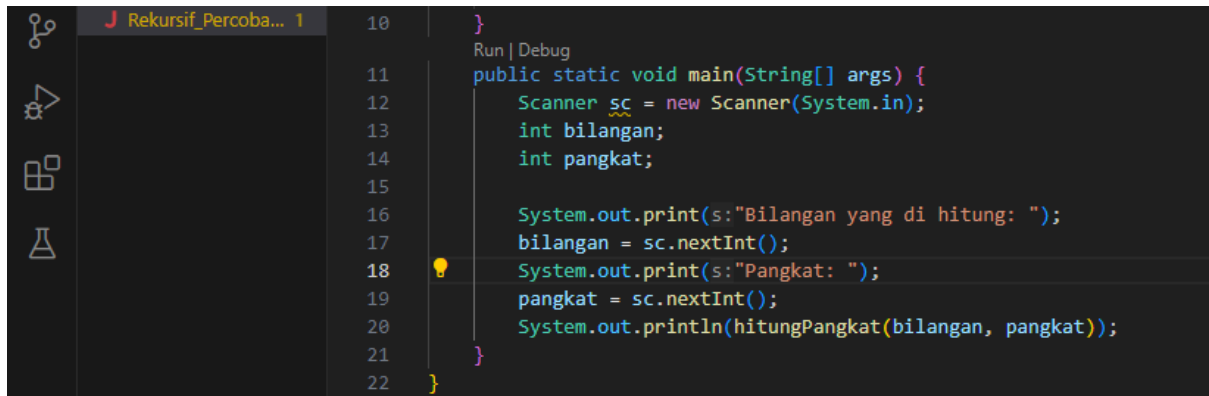
5. Tambahkan kode berikut ini untuk menerima input dari keyboard

```
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));
```

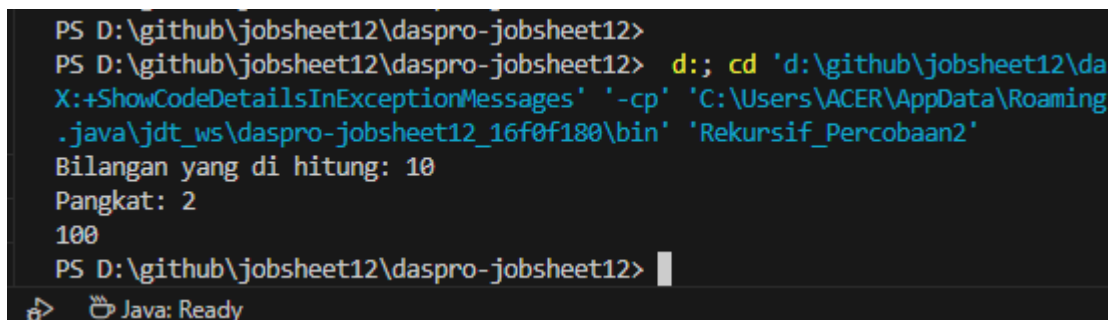
## KODE PROGRAM



```
10 }  
11 Run | Debug  
12 public static void main(String[] args) {  
13     Scanner sc = new Scanner(System.in);  
14     int bilangan;  
15     int pangkat;  
16  
17     System.out.print(s:"Bilangan yang di hitung: ");  
18     bilangan = sc.nextInt();  
19     System.out.print(s:"Pangkat: ");  
20     pangkat = sc.nextInt();  
21     System.out.println(hitungPangkat(bilangan, pangkat));  
22 }
```

7. Jalankan program tersebut. Amati apa yang terjadi!

## HASIL KETIKA DI RUN



```
PS D:\github\jobsheet12\daspro-jobsheet12>  
PS D:\github\jobsheet12\daspro-jobsheet12> d:; cd 'd:\github\jobsheet12\da  
X:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ACER\AppData\Roaming  
.java\jdt_ws\daspro-jobsheet12_16f0f180\bin' 'Rekursif_Percobaan2'  
Bilangan yang di hitung: 10  
Pangkat: 2  
100  
PS D:\github\jobsheet12\daspro-jobsheet12>
```

## 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!  
pemanggilan akan berjalan berulang-ulang sampai mencapai base case, yaitu kondisi yang mengakhiri pemanggilan fungsi lebih lanjut. Dalam kasus ini, base case-nya adalah ketika nilai pangkat (y) mencapai 0, bahwa hasil dari setiap bilangan yang dipangkatkan dengan 0 adalah 1
2. Tambahkan kode program untuk mencetak deret perhitungannya. Contoh : hitungPangkat(2,5) dicetak  $2 \times 2 \times 2 \times 2 \times 1 = 32$

## KODE PROGRAM

```
File Edit Selection View Go Run ...
daspro-jobsheet12

EXPLORER
DASPRO-JOBSHEET12
  percobaan1_pert... 1
  Rekursif_Percoba... 1
  Rekursif_Percoba... 1, M

J Rekursif_Percobaan2.java > Rekursif_Percobaan2 > hitungPangkat(int, int)
1 import java.util.Scanner;
2 public class Rekursif_Percobaan2 {
3
4     static int hitungPangkat (int x, int y) {
5         if (y == 0) {
6             return 1;
7         } else {
8             int result = x * hitungPangkat(x, y - 1);
9             if (y!=1) {
10                 System.out.print(x + "x" );
11             }
12             return result;
13         }
14     }
15
16     Run | Debug
17     public static void main(String[] args) {
18         Scanner sc = new Scanner(System.in);
19         int bilangan;
20         int pangkat;
21
22         System.out.print(s:"Bilangan yang di hitung: ");
23         bilangan = sc.nextInt();
24         System.out.print(s:"Pangkat: ");
25         pangkat = sc.nextInt();
26         System.out.println(x:"Deret perhitungan: ");
27         int hasil = hitungPangkat(bilangan, pangkat);
28         System.out.println(" = " + hasil);
29     }
30 }
```

HASIL KETIKA DI RUN

```
PS D:\github\jobsheet12\daspro-jobsheet12>
PS D:\github\jobsheet12\daspro-jobsheet12> d: Open folder in new window
X:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ACER\AppData\Local\Temp\16f0f180\bin' 'Rekursif_Percobaan2'
Bilangan yang di hitung: 2
Pangkat: 5
Deret perhitungan:
2x2x2x2x = 32
PS D:\github\jobsheet12\daspro-jobsheet12>
```

## PERCOBAAN3

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));
    }
}
```

KODE PROGRAM

```

1  import java.util.Scanner;
2  public class Rekursif_Percobaan3 {
3      static double hitungLaba (double saldo, int tahun) {
4          if (tahun == 0) {
5              return saldo;
6          } else {
7              return (1.11 * hitungLaba(saldo, tahun));
8          }
9      }

```

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(:"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.

#### KODE PROGRAM

```

1  import java.util.Scanner;
2  public class Rekursif_Percobaan3 {
3      static double hitungLaba (double saldo, int tahun) {
4          if (tahun == 0) {
5              return saldo;
6          } else {
7              return (1.11 * hitungLaba(saldo, tahun));
8          }
9      }
10     public static void main(String[] args) {
11         Scanner sc = new Scanner(System.in);
12         double saldoAwal;
13         int tahun;
14
15         System.out.println(x:"Jumlah saldo awal: ");
16         saldoAwal = sc.nextInt();
17         System.out.println(x:"Lamanya investasi (tahun): ");
18         tahun = sc.nextInt();
19     }
20 }

```

7. Jalankan program tersebut. Amati apa yang terjadi!

#### HASIL KETIKA DI RUN

```

PS D:\github\jobsheet12\daspro-jobsheet12>
PS D:\github\jobsheet12\daspro-jobsheet12> d:; cd 'd:\github\jobsheet12\daspro-jobsheet12'; java -X:+ShowCodeDetailsInExceptionMessages -cp 'C:\Users\ACER\AppData\Roaming\OpenJDK\jdk-11.0.10\bin\java\jdt ws\daspro-jobsheet12 16f0f180\bin' 'Rekursif_Percobaan3'
Jumlah saldo awal: 50000
Lamanya investasi (tahun): 1
jumlah saldo setelah 1 tahun:
55500.000000000001
PS D:\github\jobsheet12\daspro-jobsheet12>

```

## PERTANYAAN

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

“if (tahun == 0) { System.out.print(saldo); return saldo; }” Blok kode tersebut adalah Base Case

Sedangkan “double result = 1.11 \* hitungLaba(saldo, tahun - 1);” Blok kode tersebut adalah recursion call

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

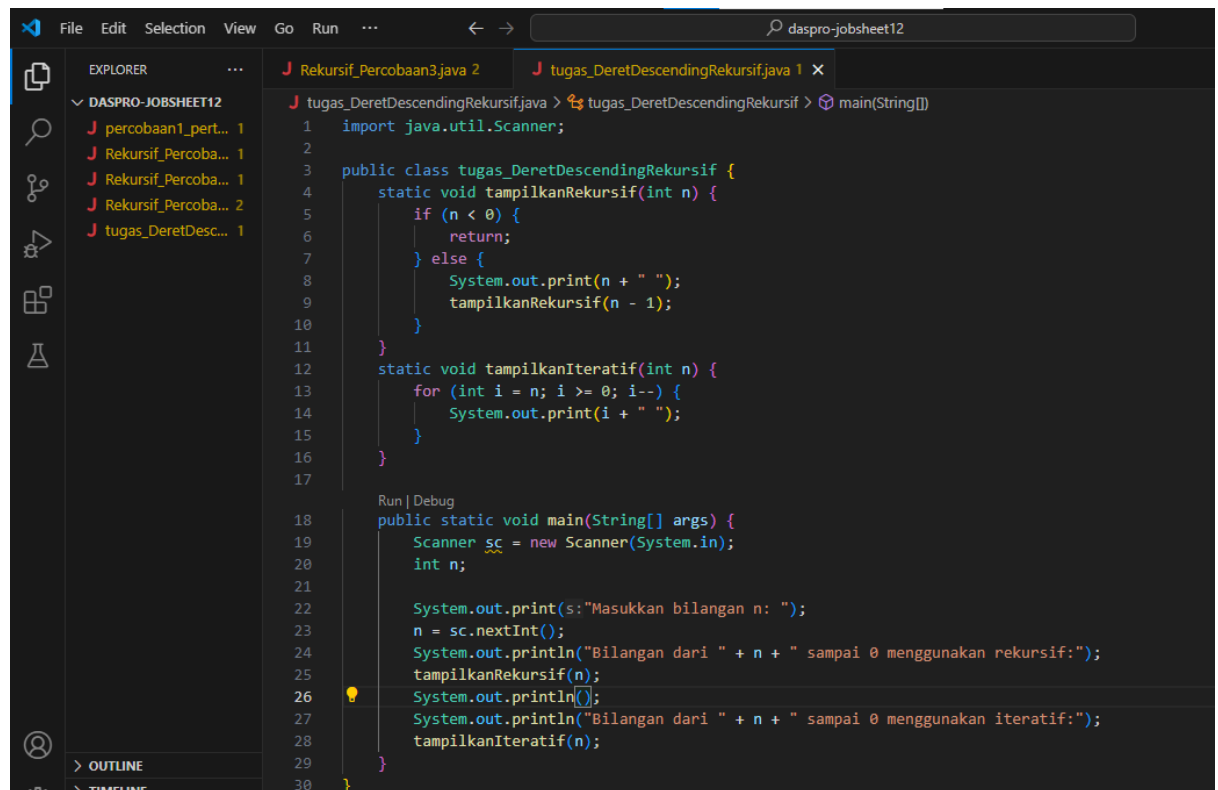
1. Saat memanggil hitungLaba(100000, 3), output sementara akan mencetak 100000x1.11x.
2. Saat memanggil hitungLaba(100000, 2), output sementara akan mencetak 100000x1.11x11000x1.11x.
3. Saat memanggil hitungLaba(100000, 1), output sementara akan mencetak 100000x1.11x11000x1.11x123210x1.11.

Akhirnya, setelah base case tercapai dan hasil akhir dihitung, output akhirnya adalah: 100000x1.11x11000x1.11x123210x1.11 = 136367.1

## TUGAS

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

### KODE PROGRAM



```
File Edit Selection View Go Run ... daspro-jobsheet12

EXPLORER
DASPRO-JOBSHEET12
  percobaan1_pert... 1
  Rekursif_Percoba... 1
  Rekursif_Percoba... 1
  Rekursif_Percoba... 2
  tugas_DeretDesc... 1

J Rekursif_Percobaan3.java 2 J tugas_DeretDescendingRekursif.java 1 x
J tugas_DeretDescendingRekursif.java > tugas_DeretDescendingRekursif > main(String[])
1 import java.util.Scanner;
2
3 public class tugas_DeretDescendingRekursif {
4     static void tampilkanRekursif(int n) {
5         if (n < 0) {
6             return;
7         } else {
8             System.out.print(n + " ");
9             tampilkanRekursif(n - 1);
10        }
11    }
12    static void tampilkanIteratif(int n) {
13        for (int i = n; i >= 0; i--) {
14            System.out.print(i + " ");
15        }
16    }
17
18    Run | Debug
19    public static void main(String[] args) {
20        Scanner sc = new Scanner(System.in);
21        int n;
22
23        System.out.print(s:"Masukkan bilangan n: ");
24        n = sc.nextInt();
25        System.out.println("Bilangan dari " + n + " sampai 0 menggunakan rekursif:");
26        tampilkanRekursif(n);
27        System.out.println();
28        System.out.println("Bilangan dari " + n + " sampai 0 menggunakan iteratif:");
29        tampilkanIteratif(n);
30    }
}
```

### HASIL KETIKA DI RUN



```
PS D:\github\jobsheet12\daspro-jobsheet12>
PS D:\github\jobsheet12\daspro-jobsheet12> d:; cd 'd:\github\jobsheet12\daspro-jobsheet12'; & 'C:\Program Files\Java\jdk-22\bin\java.exe' '-XX:ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ACER\AppData\Roaming\Code\User\workspaceStorage\ab8cd60929c75f23fad13815b8bff96e\redhat.java\jdt_ws\daspro-jobsheet12_16f0f180\bin' 'tugas_DeretDescendingRekursif'
Masukkan bilangan n: 11
Bilangan dari 11 sampai 0 menggunakan rekursif:
11 10 9 8 7 6 5 4 3 2 1 0
Bilangan dari 11 sampai 0 menggunakan iteratif:
11 10 9 8 7 6 5 4 3 2 1 0
PS D:\github\jobsheet12\daspro-jobsheet12>
```

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).

#### KODE PROGRAM

```
File Edit Selection View Go Run ... daspro-jobsheet12
EXPLORER
DASPRO-JOBSHEET12
  percobaan1_pert... 1
  Rekursif_Percoba... 1
  Rekursif_Percoba... 1
  Rekursif_Percoba... 2
  tugas_DeretDesc... 1
  tugas_Penjumla... 1, U
  tugas_PenjumlahanRekursif.java 2
  tugas_DeretDescendingRekursif.java 1
  tugas_PenjumlahanRekursif.java 1, U x

tugas_PenjumlahanRekursif.java
1 import java.util.Scanner;
2
3 public class tugas_PenjumlahanRekursif {
4     static int hitungPenjumlahan(int n, StringBuilder deret) {
5         if (n == 1) {
6             deret.append(str:"1");
7             return 1;
8         } else {
9             int hasil = n + hitungPenjumlahan(n - 1, deret);
10            deret.insert(offset:0, n + " + ");
11            return hasil;
12        }
13    }
14
15    Run | Debug
16    public static void main(String[] args) {
17        Scanner sc = new Scanner(System.in);
18        int bil;
19        System.out.print(s:"Masukkan bilangan yang ingin dijumlah: ");
20        bil = sc.nextInt();
21        StringBuilder deret = new StringBuilder();
22        int hasilPenjumlahan = hitungPenjumlahan(bil, deret);
23        System.out.println("Deret penjumlahan: " + deret.toString() + " = " + hasilPenjumlahan);
24    }
25 }
```

#### HASIL KETIKA DI RUN

```
PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL PORTS Run: tugas_Per
PS D:\github\jobsheet12\daspro-jobsheet12> & 'C:\Program Files\Java\jdk-22\bin\java.exe' '-XX:ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ACER\AppData\Roaming\Code\User\workspaceStorage\ab8cd60929c75f23fad13815b8bff96e\redhat.java\jdt_ws\daspro-jobsheet12_16f0f180\bin' 'tugas_PenjumlahanRekursif'
Masukkan bilangan yang ingin dijumlah: 8
Deret penjumlahan: 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 = 36
PS D:\github\jobsheet12\daspro-jobsheet12>
```

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

## KODE PROGRAM

```

1  public class tugas_Fibonacci {
2      public static int jumlahPasanganMarmut(int bulan) {
3          if (bulan <= 2) {
4              return 1;
5          } else {
6              return jumlahPasanganMarmut(bulan - 1) + jumlahPasanganMarmut(bulan - 2);
7          }
8      }
9  }

10 public static void main(String[] args) {
11     int bulan = 12;
12     int jumlahPasangan = jumlahPasanganMarmut(bulan);
13     System.out.println("Jumlah pasangan marmut pada bulan ke-" + bulan + " adalah: " + jumlahPasangan);
14 }
15

```

## HASIL KETIKA DI RUN

```

PS D:\github\jobsheet12\daspro-jobsheet12> & 'C:\Program Files\Java\jdk-22\bin\java.exe' '-XX:+ShowCodeDetails -ea' -cp .\classes\ tugas_Fibonacci
Jumlah pasangan marmut pada bulan ke-12 adalah: 144
PS D:\github\jobsheet12\daspro-jobsheet12>

```

Jadi jumlah pasangan marmut yang ada pada akhir bulan ke-12 jawabannya adalah **144**