**JOBSHEET 11**

**Fungsi Rekursif**

1. **Tujuan**

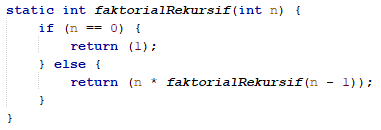
* Mahasiswa memahami konsep fungsi rekursif
* Mahasiswa mampu mengimplementasikan fungsi rekursif dalam kode program

1. **Praktikum**
   1. **Percobaan 1**

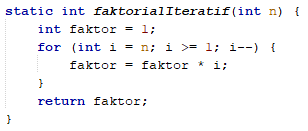
**Waktu Percobaan : 60**

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.

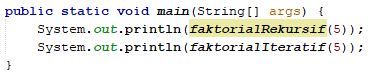
1. Buat project baru bernama **Rekursif**, dan buat file Java dengan nama **Percobaan1**
2. Buat fungsi static dengan nama **faktorialRekursif()**, dengan tipe data kembalian fungsi **int** dan memiliki 1 parameter dengan tipe data **int** berupa bilangan yang akan dihitung nilai faktorialnya.



1. 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.



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



1. Jalankan program tersebut. Amati apa yang terjadi!
2. 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

1. Bagaimana contoh kasus penggunaan fungsi rekursif ?

Bisa digunakan untuk menghitung factorial

1. 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,

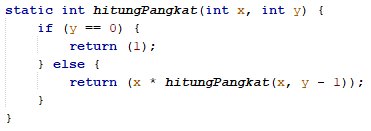
Fungsi rekursif menggunakan proses pemanggilan diri sendiri secara berulang. Sedangkan, Fungsi iteratif menggunakan proses perulangan.

* 1. **Percobaan 2**

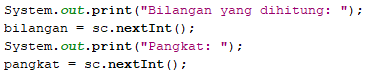
**Waktu Percobaan : 60**

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.



1. Buatlah fungsi main dan deklarasikan Scanner dengan nama **sc**
2. Buatlah dua buah variabel bertipa int dengan nama **bilangan** dan **pangkat**
3. Tambahkan kode berikut ini untuk menerima input dari keyboard



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



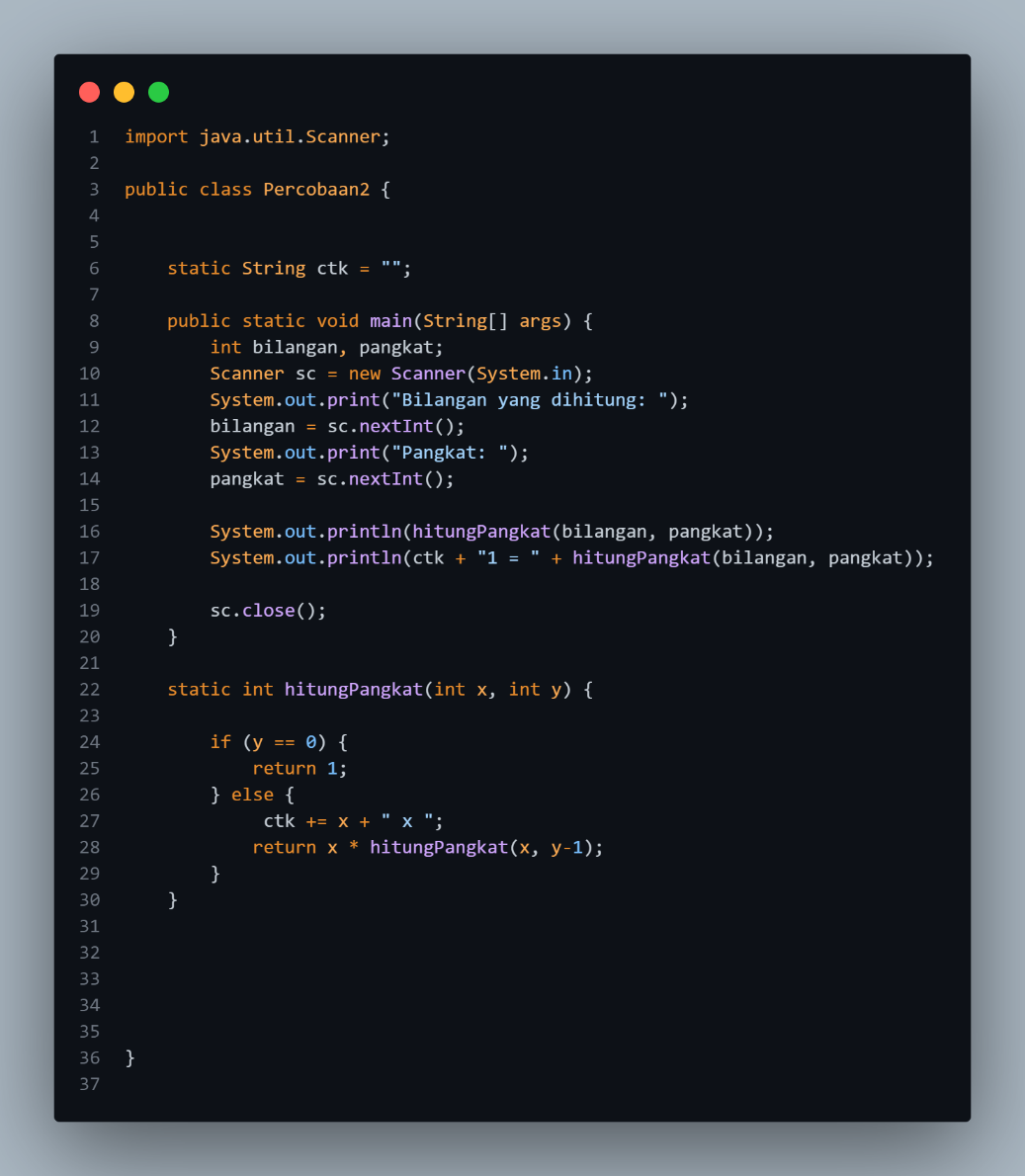
1. Jalankan program tersebut. Amati apa yang terjadi!

**Pertanyaan**

1. Pada **Percobaan2**, terdapat pemanggilan fungsi rekursif **hitungPangkat(bilangan, pangkat)** pada fungsi main, kemudian dilakukan pemanggilan fungsi **hitungPangkat()** secara berulangkali.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.

1. Tambahkan kode program untuk mencetak deret perhitungan pangkatnya. Contoh : **hitungPangkat(2,5)** dicetak 2x2x2x2x2x1 = 32



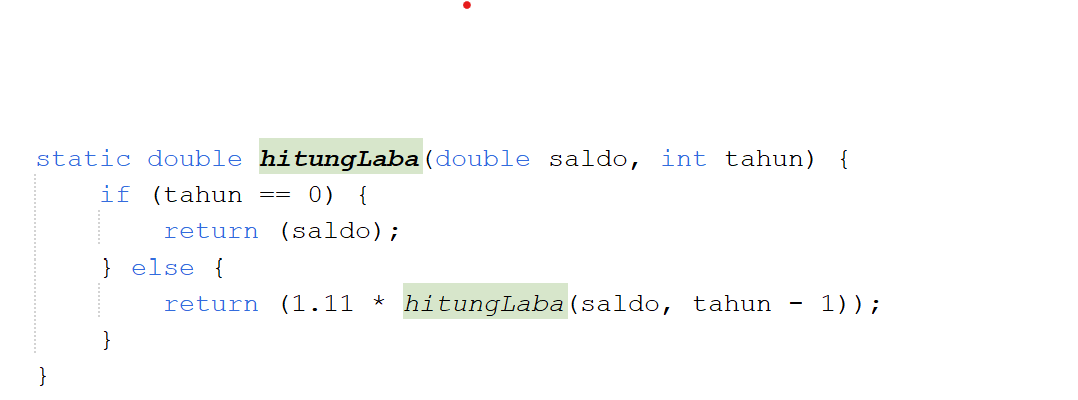
* 1. **Percobaan 3**

**Waktu Percobaan : 60**

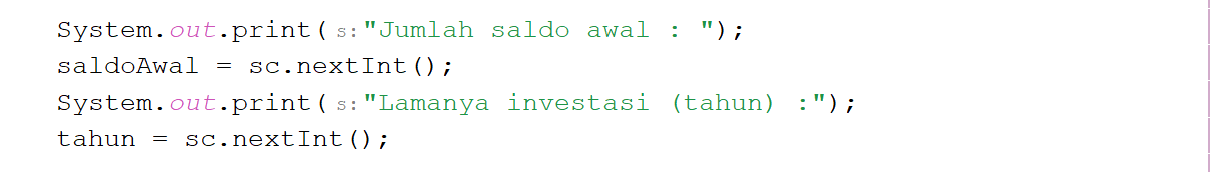
Pada percobaan ini akan dilakukan pembuatan program untuk menghitung jumlah uang investor yang digunakan sebagai investasi setelah mendapatkan laba selama beberapa tahun dengan menggunakan fungsi rekursif.

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.

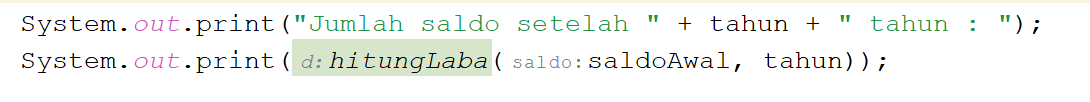
Pada kasus ini dianggap laba yang ditentukan adalah 11% per tahun. Karena perhitungan laba adalah **laba \* saldo**, sehingga untuk menghitung besarnya uang setelah ditambah laba adalah **saldo + laba \* saldo**. Dalam hal ini, besarnya laba adalah 0.11 \* saldo, dan saldo dianggap 1 \* saldo, sehingga 1 \* saldo + 0.11 \* saldo dapat diringkas menjadi **1.11 \* saldo** untuk perhitungan saldo setelah ditambah laba (dalam setahun).



1. Buatlah fungsi main dan deklarasikan Scanner dengan nama **sc**
2. Buatlah sebuah variabel bertipa double dengan nama **saldoAwal** dan sebuah variabel bertipe int bernama **tahun**
3. Tambahkan kode berikut ini untuk menerima input dari keyboard

****

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



1. 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.

1. Jabarkan trace fase ekspansi dan fase subtitusi 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. 2 | 111000
3. 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.

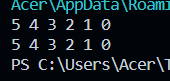
Pada tahun ketiga, fungsi hitungLaba() akan dipanggil dengan nilai saldo awal sebesar 123.211 dan jumlah tahun sebesar 0. Fungsi ini akan mencapai kondisi dasar, karena jumlah tahun telah menjadi 0. Oleh karena itu, fungsi akan mengembalikan nilai saldo, yaitu 123.211.

1. **Tugas**

**Waktu Pengerjaan 120 menit**

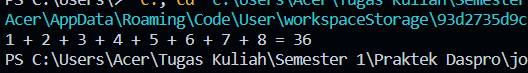
1. Buatlah program untuk menampilkan bilangan n sampai 0 dengan menggunakan fungsi rekursif dan fungsi iteratif. (**DeretDescendingRekursif**).



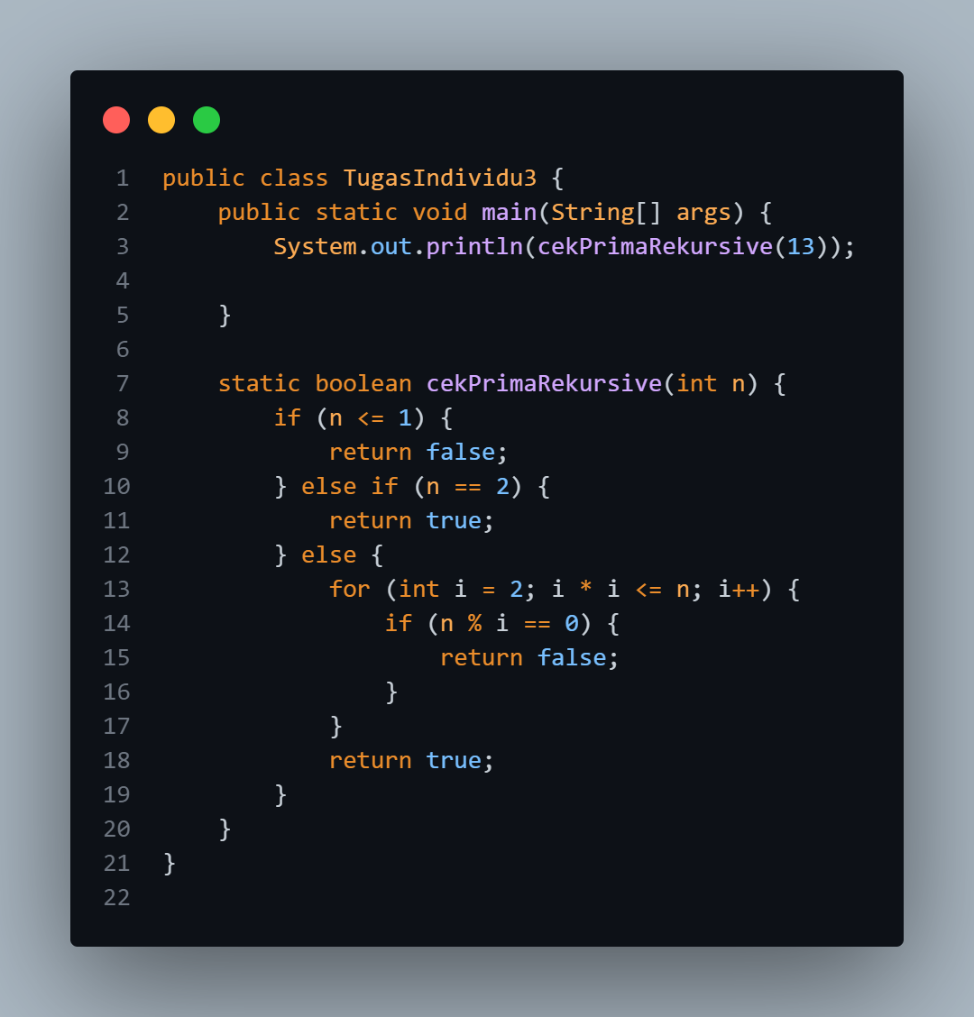


1. Buatlah program yang di dalamnya terdapat fungsi rekursif untuk menghitung penjupsmlahan bilangan. Misalnya f = 8, maka akan dihasilkan 1+2+3+4+5+6+7+8 = **36** (**PenjumlahanRekursif**).





1. 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. 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 |

