



# Rekursi

Pemrograman 3



# Definisi Rekursi

- Rekursi atau *recurrence* dapat didefinisikan sebagai metode yang mengulang prosesnya sendiri. Saat sebuah rekursi dipanggil, proses akan dijalankan dan diulang sampai kondisi berhenti terpenuhi.
- Dua hal yang selalu ada dalam sebuah rekursi adalah *Blok Rekursi* dan *Blok Basis*. *Blok Rekursi* merupakan kode program yang memuat kondisi mengulang. Blok inilah yang nantinya berfungsi mengulang prosesnya terus-menerus. *Blok Basis* merupakan kode program yang memuat kondisi berhenti. Blok inilah yang menyebabkan sebuah rekursi menghentikan prosesnya, sehingga diperoleh hasil akhir.



# Konsep Rekursi

- Untuk menyusun rekursi, terlebih dahulu kita harus menentukan *konsep*. Konsep bukanlah bagian program, melainkan hanya merupakan cara kerja atau proses dasar sebuah rekursi. Konsep ini akan memudahkan kita dalam menyusun rekursi kedalam kode program.
- Konsep berisikan urutan/langkah terperinci yang menjelaskan proses rekursi dari awal sampai akhir. Konsep terdiri atas *permisalan*, *langkah-langkah*, dan *hasil akhir*. *Permisalan* dapat berisikan nilai awal. *Langkah-langkah* meliputi urutan dan hasil sementara di setiap perulangan. *Hasil akhir* merupakan nilai dari proses rekursi

# Deklarasi Rekursi

- Telah kita ketahui bersama bahwa rekursi merupakan metode. Oleh sebab itu cara mendeklarasikan rekursi mirip dengan cara mendeklarasikan metode. Ada sedikit perbedaan dalam mendeklarasikan rekursi, yaitu kita harus memisahkan antara blok perulangan dan blok penghentian.

Algoritma	Java
<pre>method* <u>namarekursi</u> (<u>namavariabelin1</u> : <u>tipedatain1</u>,  <u>namavariabelin-n</u> : <u>tipedatain-n</u>) -&gt; <u>tipedataout</u>   1 { <u>if</u> (<u>kondisiberhenti</u>) <u>then</u>       return(<u>nilainetral</u>***)   2 { <u>else</u> {<u>kondisimengulang</u>}       return(<u>pemanggilanulang</u>)     <u>end if</u> <u>end method</u>****</pre>	<pre>public static <u>tipedataout</u>** <u>namarekursi</u> (<u>tipedatain1</u> <u>namavariabelin1</u>,  <u>tipedatain-n</u> <u>namavariabelin-n</u>) {   1 { <u>if</u> (<u>kondisiberhenti</u>)       {         return(<u>nilainetral</u>***);       }     <u>else</u> //<u>kondisimengulang</u>   2 { {         return(<u>pemanggilanulang</u>);       }     } }</pre>

ket:

- \* method = dapat berupa procedure atau function
- \*\* tipedataout = void untuk procedure dan tipe data tertentu untuk function
- \*\*\* nilainetral = merupakan nilai yang tidak mengubah hasil sebenarnya
- \*\*\*\* end method = dapat berupa end procedure atau end function
- 1 blok basis
- 2 blok rekursi

# Memanggil Rekursi

- Sama halnya dengan metode, rekursi juga akan dipanggil jika dibutuhkan. Saat rekursi dipanggil proses didalam rekursi akan berjalan, berulang, berhenti lalu memberikan hasil akhir. Cara memanggil rekursi juga mirip dengan cara memanggil metode.

Algoritma	Java
$namavariabel \leftarrow namarekursi (variabelinput)$	<code>namavariabel = namarekursi (variabelinput);</code>

# Contoh Program

```
class contoh_rekursi {  
    public static void main (String [] args) {  
        System.out.println(faktorial(5));  
    }  
  
    static int faktorial (int a) {  
        if (a==0) return 1;  
        else return a*faktorial(a-1);  
    }  
}
```