

BAHASA PEMROGRAMAN JAVA

Pertemuan VI

Rekursi

Oleh
Achmad Arrosyidi



TUJUAN PEMBELAJARAN

Mahasiswa dapat memperbandingkan perbedaan antara Rekursi dengan Looping.



MATERI

- A. Pengantar Rekursi
- B. Pengertian Rekursi
- C. Aplikasi terapan yang menggunakan Rekursif / Contoh
- D. Perbedaan Rekursif Dengan Looping
- E. Ringkasan
- F. Latihan



A. REKURSI - **PENGANTAR**

- Sebelum komputer dikenal manusia ilmu matematika telah berkembang selama ribuan tahun.
- Salah satu cara mendefinisikan sesuatu di dalam matematika adalah pendefinisian yang bersifat rekursif.
- Pada saat bahasa pemrograman komputer dirancang, fitur rekursif ini disertakan dengan tujuan untuk memudahkan transformasi formula matematika menjadi program komputer



A. REKURSI - **PENGERTIAN**

Rekursif adalah fungsi memanggil dirinya sendiri. Terdapat syarat dalam pembuatan fungsi sebagai rekursif, yaitu:

1. Instruksi didalam fungsi yang memanggil dirinya sendiri. Pemanggilan atas fungsi itu sendiri mengindikasikan bahwa fungsi ini tergolong sebagai rekursif
2. Terdapat kondisi yang sebagai anker (jangkar). Jangkar ini bertujuan agar fungsi yang memanggil dirinya sendiri dapat berhenti. Pemberhentian ini penting supaya rekursi tidak secara terus menerus berlangsung. Seperti layaknya looping maka harus dihentikan agar proses tidak terjadi hang dan alur program dapat melanjutkan tahapan selanjutnya.

C. REKURSI – CONTOH 1

- Penyelesaian masalah yang menggunakan cara rekursif dalam operasi aritmatika diantaranya adalah faktorial.
- $n!$ adalah hasil kali dari n dengan $(n-1)!$.
- Untuk menyelesaikan $(n-1)!$ adalah sama dengan $n!$,
- Sehingga $(n-1)!$ adalah $n-1$ dikalikan dengan $(n-2)!$, dan $(n-2)!$ adalah $n-2$ dikalikan dengan $(n-3)!$ dan seterusnya sampai dengan $n = 0$, kita menghentikan penghitungan $n!$



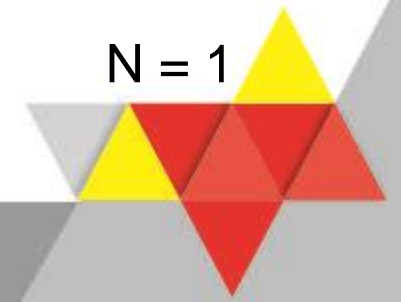
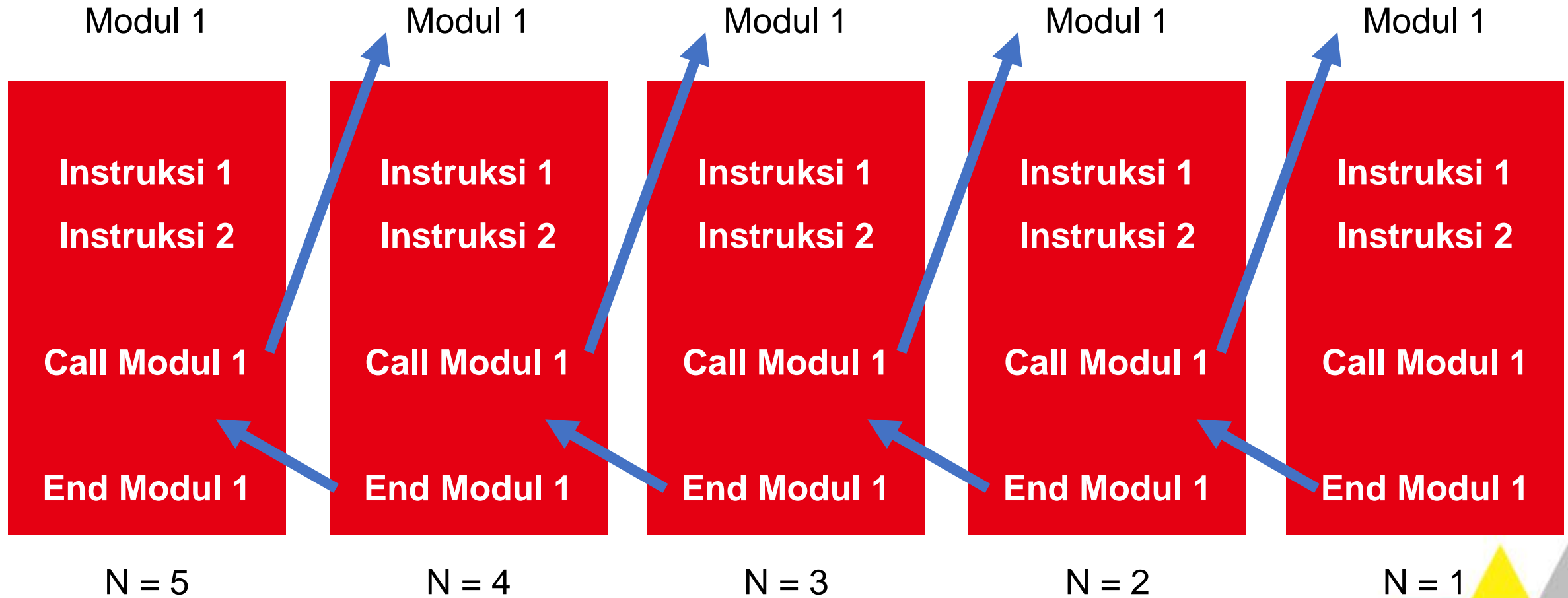
C. REKURSI – CONTOH 1

- KONDISI TUMPUKAN REKURSI
- Di dalam memori komputer saat rekursi ini dieksekusi akan membentuk stack (tumpukan).

N	FAKTORIAL
N = 1	1
N = 2	2 * Faktorial (1)
N = 3	3 * Faktorial (2)
N = 4	4 * Faktorial (3)
N = 5	5 * Faktorial (4)



C. REKURSI – CONTOH 1



C. REKURSI – CONTOH 1

```
1 //Files: Rekursi1.java
2 import java.util.Scanner;
3
4 public class Rekursi1 {
5     //Fungsi Rekursif faktorial
6     public static int faktorial(int n) {
7         if (n==0) { //kondisi anker/jangkar
8             return (1);
9         }
10        else
11            return(n*faktorial(n-1)); //instruksi yang memanggil dirinya sendiri
12    }
13
14    public static void main (String[] arg) {
15        Scanner sc = new Scanner(System.in);
16        System.out.print("Masukkan angka yang dicari faktorialnya: ");
17        int angka= sc.nextInt();
18
19        //Memanggil fungsi rekursif faktorial
20        int hasil=faktorial(angka);
21        System.out.println("Hasil dari faktorial " + angka + " adalah : " + + hasil);
22    }
23 }
```

Console

```
Masukkan angka yang dicari faktorialnya: 5
Hasil dari faktorial 5 adalah : 120
<<< Process finished (PID=10668). (Exit code 0)
===== READY =====
```



C. REKURSI – CONTOH 2

```
1 //File: SegitigaPascal.java
2 import java.util.Scanner;
3
4 public class SegitigaPascal {
5     //Fungsi rekursif pascal
6     public static int pascal(int i, int j) {
7         if ((j==0) || (i==j)) {
8             return (1);
9         }
10        else
11            return (pascal(i-1,j-1) + pascal(i-1,j));
12    }
13    public static void main (String[] args) {
14        Scanner sc = new Scanner(System.in);
15
16        System.out.print("Masukkan angka pembentuk segitiga pascal: ");
17        int angka = sc.nextInt();
18        System.out.println("Berikut adalah hasil segitiga pascal dengan angka pembentuk: " + angka);
19
20        //Memanggil fungsi rekursif pascal
21        for (int i=0; i<angka; i++) {
22            for (int j=0; j<angka-i; j++) {
23                System.out.print(" ");
24            }
25            for (int j=0; j<=i; j++) {
26                System.out.print(pascal(i,j) + " ");
27            }
28            System.out.println();
29        }
30    }
31 }
```

Console

```
Masukkan angka pembentuk segitiga pascal: 5
Berikut adalah hasil segitiga pascal dengan angka pembentuk: 5
  1
 1 1
1 2 1
1 3 3 1
1 4 6 4 1
<<< Process finished (PID=5988). (Exit code 0)
===== READY =====
```



D. REKURSI – **REKURSI VS LOOPING**

- Rekursif mempunyai kemiripan dengan looping karena memiliki prinsip berupa mengulangi tahapan yang sama, artinya didalamnya terjadi sebuah kegiatan yang dilakukan secara berulang-ulang.
- Kesamaan yang lain antara keduanya adalah mempunyai kondisi yang harus berhenti dari proses perulangan itu sendiri agar dapat menghindari perulangan yang tak berujung, karena semua langkah harus ada akhirnya.



D. REKURSI – REKURSI VS LOOPING

- Namun terdapat perbedaan antara rekursif jika dibandingkan dengan looping. Perbedaan tersebut adalah:
 1. looping bukan merupakan fungsi sedangkan rekursif merupakan fungsi.
 2. karena looping bukan merupakan fungsi maka tidak dapat dipanggil. Sedangkan rekursif adalah fungsi maka rekursif dapat dipanggil.
 3. Looping bukan merupakan fungsi sehingga tidak dapat dipanggil apalagi oleh looping itu sendiri, sedangkan rekursif adalah fungsi yang dipanggil oleh fungsi itu sendiri/dirinya sendiri.

E. REKURSI – RINGKASAN

- Rekursif adalah fungsi memanggil dirinya sendiri.
- Deklarasi dalam rekursi harus memenuhi syarat yaitu harus memanggil dirinya sendiri dan didalamnya terdapat kode program yang berfungsi sebagai **anker** (jangkar) untuk menghentikan proses rekursif.
- Perbedaan antar rekursif dengan looping yaitu rekursif adalah fungsi dan looping bukan fungsi.
- Namun kedua mempunyai kesamaan berupa didalamnya terdapat proses yang dilakukan berulang-ulang.



F. REKURSI – LATIHAN

- Buatlah program dengan yang masing-masing menggunakan rekursif dan looping untuk:
 1. Menghitung KPK dan FPB
 2. tampilkan deret fibonacci sebagai berikut:
 - Input = 10 \rightarrow Output = 1, 1, 2, 3, 5, 8, 13, 21, 34, 55
 - Input = 10 \rightarrow Output = 55, 34, 21, 13, 8, 5, 3, 2, 1, 1



**SELESAI
TERIMA KASIH**

