

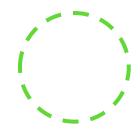
TIM DOSEN





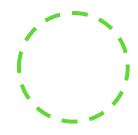
- a. Pengertian Rekursif
- b. Implementasi Rekursif dalam Bahasa C/C++

PENGERTIAN REKURSIF



- □ Salah satu konsep paling dasar dalam ilmu komputer dan pemrograman adalah pengunaan fungsi sebagai abstraksi untuk kode-kode yang digunakan berulang kali.
- □ Kedekatan ilmu komputer dengan matematika juga menyebabkan konsepkonsep fungsi pada matematika seringkali dijumpai.
- □ Salah satu konsep fungsi pada matematika yang ditemui pada ilmu komputer adalah fungsi rekursif: sebuah fungsi yang memanggil dirinya sendiri.
- □ Rekursif merupakan alat untuk memecahkan masalah dalam suatu fungsi atau procedure yang memanggil dirinya sendiri.

PENGERTIAN REKURSIF



- Menurut Niclaus Wirth: "An object is said be recursive if it partially consist or is defines in terms of itself"
- Penggunaan rekursif kadang-kadang harus mengorbankan efisiensi dan kecepatan.
- ☐ Tantangannya: eksekusi yang tidak pernah berhenti, akibatnya memori tumpukan akan habis dan computer hang.
- □ Pada dasarnya rekursif sering digunakan dalam perhitungan matematika, sebagai contoh pertimbangan fungsi factorial dan juga bilangan Fibonacci

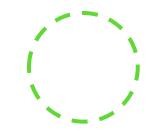






- ➤ Logika Rekursif adalah suatu fungsi berparameter yang memanggil dirinya sendiri dengan harga parameter yang berbeda
- Kelebihan logika bentuk ini mudah dipahami alurnya, namun kelemahannya pada penggunaan register stack yang sangat membebani kecepatan jalannya program.





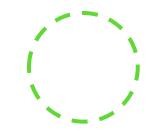
- Ada bagian base case dan ada bagian general case
- Paling sedikit mempunyai general base
- Selalu dalam bentuk fungsi-fungsi
- Selalu menggunakan statement percabangan





- **≻**Faktorial
- **▶**Bilangan Fibonacci

FAKTORIAL



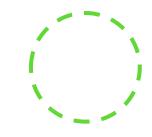
Fungsi factorial dari bilangan bulat positif n didefinisikan sebagai berikut:

```
n!= n.(n-1)!, jika n>1
n!= 1, jika n=0, 1
```

>contoh:

- · 3!= 3. 2!
- · 3!= 3. 2. 1!
- · 3!= 3. 2. 1
- · 3!= 6
- ➤ Ilustrasi: Faktorial(n) → hasilnya n*Faktorial(n-1), jika n > 1 {general case}
 - → hasilnya 1, jika n=1 atau 2 {base case}

ALGORITMA FAKTORIAL



```
Function Faktorial (input n: integer) → integer

Deklarasi
{tidak ada}

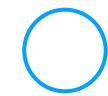
Deskripsi
if (n = 0) or (n = 1) then
return (1)
else
return (n * Faktorial(n-1))
endif
```



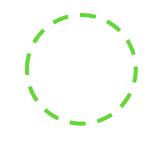
IMPLEMENTASI FUNGSI FAKTORIAL DALAM BAHASA C++

```
1. int Faktorial(int n)
2. {
3.    if ((n == 0) || (n == 1 ))
4.        return (1);
5.    else
6.        return (n * Faktorial(n-1));
7. }
```

- Pada *baris 3* dari fungsi diatas, nilai n dicek sama dengan 0 atau 1, jika ya, maka fungsi mengembalikan nilai 1 {*baris 4*}, jika tidak, fungsi mengembalikan nilai n * Faktorial (n -1) {*baris 6*}
- disinilah letak proses rekursif itu, perhatikan fungsi factorial ini memanggil dirinya sendiri tetapi dengan parameter (n-1)



BILANGAN FIBONACCI



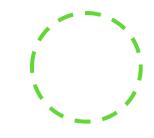
- Serangkaian deret angka sederhana yang susunan angkanya merupakan penjumlahan dari dua angka sebelumnya (0,1,1,2,3,5,8,13,21,...dst)
- Rumus deret Fibonacci bisa ditulis sebagai berikut Un = Un-2 + Un-1, artinya suku ke-n perupakan penjumlahan dari dua suku sebelumnya.
- > Fibonacci dapat didefinisikan sebagai berikut:

$$fn = fn-1 + fn-2 \text{ untuk } n>1$$

 $f0 = 0$
 $f1 = 1$

Contoh barisan bilangan Fibonacci mulai dari n=1

ALGORITMA FIBONACCI



```
Function Fibonacci(input n:integer) → integer

Deklarasi Lokal

{tidak ada}

Deskripsi

if (n ==1 || n==2) then

return (l)

else

return (Fibonacci(n-1)+Fibonacci(n-2))

endif
```

IMPLEMENTASI FUNGSI FIBONACCI DALAM BAHASA C++

```
1. int fibonacci(int n)
2. {
3.    if ((n == 0) || (n == 1 ))
4.        return (1);
5.    else
6.        return (Fibonacci(n-1)+Fibonacci(n-2));
7. }
```

TUGAS LMS