

Rekursi

Rekursi : kemampuan suatu sub program / fungsi untuk memanggil dirinya sendiri.

Ciri fungsi rekursi:

- Kasus penyetop.
Dalam kasus ini terdapat nilai konstan (return value)
- Kasus pemanggilan rekursif.
Dalam kasus ini terdapat pemanggilan fungsi itu sendiri, tetapi harus mengarah kepada kasus penyetop.

Contoh 1:

Fungsi Faktorial.

$n! = n(n-1)(n-2)\dots(1)$.

Dapat dituliskan sebagai :

$n! = n(n-1)!$

dimana :

$n! = 1$ jika $n=1$ atau $n=0$

Analisis :

Kasus penyetop (= nilai awal) $n = 0$ atau $n = 1$ yaitu bernilai konstan 1

Kasus rekursif :

faktorial (n) = $n * \text{faktorial}(n-1)$

```
def faktorial(n):  
    if n == 0:  
        return 1  
    else:  
        return n * faktorial(n - 1)
```

```
print(faktorial(4))
```

24

Contoh 2.

Bilangan fibonacci ke n didefinisikan :

$f_n = f_{n-1} + f_{n-2}$ dengan $f_0 = 0, f_1 = 1$

Misalnya bilangan fibonacci ke 2 :

$$\begin{aligned} f_2 &= f_1 + f_0 \\ &= 1 + 0 \\ &= 1 \end{aligned}$$

Misalnya bilangan fibonacci ke 3 :

$$\begin{aligned} f_3 &= f_2 + f_1 \\ &= 1 + 1 \\ &= 2 \end{aligned}$$

Analisis :

Kasus penyetop $n = 0$ maka $f_0 = 0$, $n = 1$ maka $f_1 = 1$,

Kasus rekursif :

$$Fib(n) = Fib(n - 1) + Fib(n - 2)$$

```
def fib(n):  
    if n==0:  
        return 0  
    elif n==1:  
        return 1  
    else:  
        return (fib(n-1)+fib(n-2))
```

```
fib(7)
```

13

Latihan :

1. Ubahlah fungsi fibonacci di atas ke dalam bentuk iterasi/perulangan menggunakan for atau while.
2. Diberikan fungsi untuk menghitung nilai dua pangkat x (2^x)
 - a. Buat analisis kasus penyetop dan kasih rekursinya
 - b. Buat program untuk menghitungnya dalam rekursi
 - c. Buat programnya dengan iterasi/perulangan.