

Rekursif

Rekursif

Definisi entitas (type, fungsi) disebut rekursif jika definisi tersebut mengandung terminologi dirinya sendiri.

Ekspresi rekursif dalam pemrograman fungsional didasari oleh Analisa rekurens, yaitu penalaran berdasarkan definisi **fungsi rekursif**, yang biasanya juga berdasarkan “type” yang juga terdefinisi secara rekursif-induktif.

Fungsi Rekursif

Fungsi didefinisikan rekursif, jika **ekspresi** yang merealisasi fungsi tersebut mengandung **aplikasi** terhadap fungsi tersebut. :

REALISASI

F (<list-param>) :

depend on

<kondisi-basis > : <ekspresi-1 >

<kondisi-rekurens > : **F** (<ekspresi-2 >)

Dengan catatan, bahwa ekspresi-2 biasanya dinyatakan dengan domain yang sama dengan <list-param>, namun “mendekati” kondisi basis sehingga suatu saat akan terjadi kondisi basis yang menyebabkan aplikasi berhenti.

Faktorial

Faktorial	fac(n)
<u>DEFINISI DAN SPESIFIKASI</u>	
<p>fac : $\text{integer} \geq 0 \rightarrow \text{integer} > 0$</p> <p>$\{ \text{fac}(n) = n! \text{ sesuai dengan definisi rekursif factorial} \}$</p>	
<u>REALISASI (VERSI-1)</u>	
<p>$\{ \text{Realisasi dengan definisi factorial sebagai berikut jika fac(n) adalah n!} :$</p> <p style="padding-left: 40px;">$n = 0 : n! = 1$</p> <p style="padding-left: 40px;">$n \geq 1 : n! = (n-1)! * n \}$</p> <p>fac(n) : <u>if</u> $n = 1$ <u>then</u> {Basis 1}</p> <p style="padding-left: 40px;">1</p> <p style="padding-left: 40px;"><u>else</u> {Rekurens : definisi faktorial}</p> <p style="padding-left: 40px;">fac (n-1) * n</p>	
<u>REALISASI (VERSI-2)</u>	
<p>$\{ \text{Realisasi dengan definisi factorial sebagai berikut jika fac(n) adalah n!} :$</p> <p style="padding-left: 40px;">$n = 0 : n! = 1$</p> <p style="padding-left: 40px;">$n \geq 1 : n! = (n-1)! * n \}$</p> <p>fac(n) : <u>if</u> $n = 0$ <u>then</u> {Basis 0}</p> <p style="padding-left: 40px;">1</p> <p style="padding-left: 40px;"><u>else</u> {Rekurens : definisi faktorial}</p> <p style="padding-left: 40px;">n * fac (n-1)</p>	

Iterasi Vs Rekursif

```
fak=1
i=1
while i<=5:
    fak=fak*i
    i=i+1
print(fak)
```

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

Fibonacci

Fibonacci	Fib(n)
<u>DEFINISI DAN SPESIFIKASI</u>	
Fib : <u>integer</u> $\geq 0 \rightarrow$ <u>integer</u> ≥ 0	
{ Definisi rekursif fungsi fibonacci : }	
{ Fib (n) = sesuai dengan definisi deret fibonacci :	
$n = 0 : \text{Fib}(0) = 0$	
$n = 1 : \text{Fib}(1) = 1$	
$n > 1 : \text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2) \}$	
<u>REALISASI</u>	
Fib (n) : <u>depend on</u> (n)	
$n = 0 : 0$ {Basis 0}	
$n = 1 : 1$ {Basis 1}	
$n > 1 : \text{Fib}(n-1) + \text{Fib}(n - 2)$ {Rekurens}	

Iterasi Vs Rekursif

```
>>> def fib(n):    # write Fibonacci series up to n
...     """Print a Fibonacci series up to n."""
...     a, b = 0, 1
...     while a < n:
...         print a,
...         a, b = b, a+b
...
>>> # Now call the function we just defined:
... fib(2000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597
```

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