

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)
<div data-bbox="92 297 743 354"><u>DEFINISI DAN SPESIFIKASI</u></div> <div data-bbox="92 389 687 446">fac : <u>integer</u> $\geq 0 \rightarrow$ <u>integer</u> > 0</div> <div data-bbox="92 461 1164 518"><i>{ fac (n) = n! sesuai dengan definisi rekursif factorial }</i></div> <div data-bbox="92 525 591 582"><u>REALISASI (VERSI-1)</u></div> <div data-bbox="92 604 1541 661"><i>{ Realisasi dengan definisi factorial sebagai berikut jika fac(n) adalah n!:</i></div> <div data-bbox="392 661 662 711">$n = 0 : n! = 1$</div> <div data-bbox="392 718 900 768">$n \geq 1 : n! = (n-1) ! * n \}$</div> <div data-bbox="166 768 1275 946">fac (n) : <u>if</u> n = 1 <u>then</u> {Basis 1} 1 <u>else</u> {Rekurens : definisi faktorial} fac (n-1) * n</div> <div data-bbox="92 961 604 1018"><u>REALISASI (VERSI-2)</u></div> <div data-bbox="92 1039 1541 1096"><i>{ Realisasi dengan definisi factorial sebagai berikut jika fac(n) adalah n!:</i></div> <div data-bbox="392 1096 653 1146">$n = 0 : n! = 1$</div> <div data-bbox="392 1153 888 1203">$n \geq 1 : n! = (n-1) ! * n \}$</div> <div data-bbox="166 1203 1275 1382">fac (n) : <u>if</u> n = 0 <u>then</u> {Basis 0} 1 <u>else</u> {Rekurens : definisi faktorial} n * fac (n-1)</div>	

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)

DEFINISI DAN SPESIFIKASI

Fib : integer $\geq 0 \rightarrow$ integer ≥ 0

{ Definisi rekursif fungsi fibonacci : }

{ Fib (n) = sesuai dengan definisi deret fibonacci :

n = 0 : Fib(0) = 0

n = 1 : Fib(1) = 1

n > 1 : Fib(n) = Fib(n-1) + Fib(n-2) }

REALISASI

Fib(n) : depend on (n)

n = 0 : 0 {Basis 0}

n = 1 : 1 {Basis 1}

n > 1 : Fib(n-1) + 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)
```