

PRAKTIKUM 5

EKSPRESI REKURSIF

Ekspresi rekursif adalah sebuah ekspresi yang di dalam definisinya mengandung terminology dirinya sendiri.

1. Tujuan Praktikum

Setelah praktikum ini, diharapkan mahasiswa mampu merealisasikan ekspresi rekursif ke dalam bahasa pemrograman Python.

2. Tools

Tools yang dibutuhkan untuk melakukan praktikum ini adalah interpreter Python yang telah terinstal di komputer.

3. Materi Praktikum

Berikut ini diberikan contoh realisasi fungsi rekursif ke dalam bahasa Python.

3.1. Ekspresi rekursif: Factorial (Versi 1)

```
#Nama file: fac.py
#Deskripsi: menghitung factorial dari sebuah bilangan integer secara rekursif
#Pembuat: Khadijah
#Tanggal: 20 September 2019

#DEFINISI DAN SPESIFIKASI
#fac : integer > 0 --> integer > 0
# fac(n) = n! sesuai dengan definisi rekursif factorial, versi 1 dengan basis=1

#REALISASI
#Realisasi dengan definisi factorial fac(n) = n! dimana
# jika n = 1 : n! = 1
# jika n > 1 : n! = n * (n-1)!

def fac(n):
    if n == 1: #basis 1
        return 1
    else: #rekurens: definisi factorial
        return n * fac(n-1)

#APLIKASI
fac(1)
fac(3)
fac(6)
```

3.2. Ekspresi rekursif: Factorial (Versi 2)

```
#Nama file: fac_v2.py
#Deskripsi: menghitung factorial dari sebuah bilangan integer secara rekursif
#Pembuat: Khadijah
#Tanggal: 20 September 2019

#DEFINISI DAN SPESIFIKASI
#fac : integer >= 0 --> integer > 0
# fac(n) = n! sesuai dengan definisi rekursif factorial, versi 2 dengan basis=0

#REALISASI
#Realisasi dengan definisi factorial fac(n) = n! dimana
# jika n = 0 : n! = 1
# jika n > 0 : n! = n * (n-1)!
def fac(n):
    if n == 0: #basis 0
        return 1
    else: #rekurens: definisi factorial
        return n * fac(n-1)

#APLIKASI
fac(0)
fac(1)
fac(3)
fac(6)
```

3.3. Ekspresi rekursif: Factorial (Versi 3)

```
#Nama file: fac_v3.py
#Deskripsi: menghitung factorial dari sebuah bilangan integer secara rekursif (INI ADALAH CONTOH YANG SALAH)
#Pembuat: Khadijah
#Tanggal: 20 September 2019

#DEFINISI DAN SPESIFIKASI
#fac : integer > 0 --> integer > 0
# fac(n) = n! sesuai dengan definisi rekursif factorial versi 3, yaitu RUMUS BENAR tapi PROGRAM SALAH

#REALISASI
#Realisasi dengan definisi factorial fac(n) = n! dimana
# jika n = 1 : n! = 1
# jika n > 1 : n! = (n+1)! / (n+1)
def fac(n):
    if n == 1: #basis 1
        return 1
    else: #rekurens: definisi factorial
        return fac(n+1) / (n+1)

#APLIKASI
fac(1)
fac(3)
fac(6)
```

Perhatikan bahwa contoh program tersebut akan error jika dijalankan, karena menyebabkan pemanggilan fungsi yang tidak pernah berhenti, atau dengan kata lain kondisi basis tidak bernilai True, sehingga pemanggilan fungsi rekurens berjalan terus.

3.4. Ekspresi rekursif: Factorial (Versi 4)

```
#Nama file: fac_v4.py
#Deskripsi: menghitung factorial dari sebuah bilangan integer versi iteratif
#Pembuat: Khadijah
#Tanggal: 20 September 2019

#DEFINISI DAN SPESIFIKASI
#faciter: 3 integer > 0 --> integer > 0
#faciter(n,count,hasil) menghitung n! sesuai dengan definisi: fac(n) = 1*2*3*...*n = hasil
# ketika count sudah mencapai n, dengan 1*2*3*...*n-1 adalah hasil sebelumnya:
# jika n = count : hasil (sebelumnya) * count
# jika n < count : faciter(n,count+1,hasil*count)

#fac : integer > 0 --> integer > 0
# fac(n) menghitung n! versi iteratif menggunakan fungsi antara faciter(n,count,hasil)

#REALISASI
def faciter(n,count,hasil):
    if n == count:
        return hasil * count
    else:
        return faciter(n,count+1,hasil*count)

def fac(n):
    return faciter(n,1,1)

#APLIKASI
fac(1)
fac(3)
fac(6)
```

3.5. Ekspresi rekursif: Bilangan Fibonacci

```
#Nama file: fibonacci.py
#Deskripsi: menghitung bilangan Fibonacci ke-n
#Pembuat: Khadijah
#Tanggal: 20 September 2019

#DEFINISI DAN SPESIFIKASI
#fibonacci : integer >= 0 --> integer > 0
# fibonacci(n) menghitung bilangan fibonacci ke-n, sesuai definisi barisan fibonacci 0,1,1,2,3,5,...
# jika n = 0 : fibonacci(n) = 0
# jika n = 1 : fibonacci(n) = 1
# jika n > 1 : fibonacci(n) = fibonacci(n-1) + fibonacci(n-2)

def fibonacci(n):
    if n == 0: #basis 0
        return 0
    elif n == 1: #basis 1
        return 1
    else: #rekurens
        return fibonacci(n-1) + fibonacci(n-2)

#APLIKASI
fibonacci(0)
fibonacci(1)
fibonacci(2)
```

3.6. Ekspresi rekursif: Penjumlahan bilangan integer

```
#Nama file: plus.py
#Deskripsi: penjumlahan dua bilangan bulat
#Pembuat: Khadijah
#Tanggal: 20 September 2019

#DEFINISI DAN SPESIFIKASI
#plus : 2 integer > 0 --> integer > 0
# plus(x,y) menjumlahkan x dan y
# plus(x,y) = x + y
#           = x + 1 + y(-1)

def plus(x,y):
    if y == 0: #basis 0
        return x
    else: #rekurens
        return 1 + plus(x,y-1)

#APLIKASI
plus(4,0)
plus(4,1)
plus(5,5)
```

4. Latihan

Buatlah fungsi rekursif untuk:

1. Operasi aritmatika pengurangan
2. Operasi aritmatika perkalian
3. Operasi aritmatika pembagian
4. Operasi aritmatika pembagian
5. Operasi aritmatika perpangkatan
6. Menghitung perkalian dengan 3 atau $f(n) = 3 * n$
 $f(1) = 3$
 $f(n+1) = f(n) + 3$
7. Menghitung deret bilangan integer: $1 + 2 + 3 + 4 + \dots$
 $S(1) = 1$
 $S(2) = 1 + 2$
 $S(3) = 1 + 2 + 3$
8. Menghitung deret aritmatika: $3 + 6 + 9 + 12 + \dots$
 $S(1) = 3$
 $S(2) = 3 + 6$
 $S(3) = 3 + 6 + 9$
9. Menghitung deret geometri: $1 + 3 + 9 + 27 + \dots$
 $S(1) = 1$
 $S(2) = 1 + 3$
 $S(3) = 1 + 3 + 9$
10. Menghitung deret : $1 + 4 + 9 + 16 + \dots$
 $S(1) = 1$
 $S(2) = 1 + 4$
 $S(3) = 1 + 4 + 9$