

## Pertemuan 6

### Sub Program dan Rekursif, variabel Global dan Lokal



#### CAPAIAN PEMBELAJARAN

---

1. Proses seleksi
2. sub program/fungsi rekursi, menggunakan variabel global, lokal



#### KEBUTUHAN ALAT/BAHAN/SOFTWARE

---

1. Sistem Operasi Linux/Window 10
2. Python
3. Teks editor Visual Studio Code



#### DASAR TEORI

---

##### Fungsi Recursive

Rekursif adalah salah satu metode dalam dunia matematika dimana definisi sebuah fungsi mengandung fungsi itu sendiri. Dalam dunia pemrograman, rekursi diimplementasikan dalam sebuah fungsi yang memanggil dirinya sendiri, contoh fungsi rekursif misalnya adalah fungsi pangkat, faktorial, dan barisan fibonacci.

Method yang memanggil dirinya sendiri baik secara langsung maupun secara tidak langsung.

$$f(0) = 0; f(x) = 2 f(x-1) + x^2$$

- $f(1) = 1;$
- $f(2) = 6;$
- $f(3) = 21;$
- $f(4) = 58$

```
Fib(0)=0;
fib(1)=1;
untuk
n>1: fib(n) = fib(n - 1) + fib(n - 2)
```

```
def fib(x):
    if (x == 0):
        return 0
    return 2 * fib (x - 1) + x * x
```

Fungsi yang memanggil dirinya, secara langsung atau lewat fungsi lain, disebut fungsi rekursif, proses pemanggilan diri itu disebut rekursi (recursion)

Contoh:

Memangkatkan bilangan real tak nol dengan suatu pangkat bilangan bulat

$$x^n = \begin{cases} 1 & \text{jika } n = 0 \\ x.x^{n-1} & \text{jika } n > 0 \\ 1.x^{-n} & \text{jika } n < 0 \end{cases}$$

```
def pangkatRekursif(x,n):
    if (n == 0):
        return 1.0
    elif (n > 0):
        return (x*pangkatRekursif (x, n - 1))
    else:
        return (1/pangkatRekursif (x, -n))
```

## Cakupan Variabel

Cakupan variabel tergantung pada lokasi di mana variabel dideklarasikan. Variabel yang dideklarasikan di satu bagian program mungkin tidak dapat diakses oleh bagian lain.

Dalam python, variabel didefinisikan dengan dua jenis cakupan.

- Variabel lokal
- Variabel global

Variabel lokal

```
#=====
# fungsi/contoh8.py
# variabel lokal
```

Hasil keluaran:

```
#-----
def print_pesan():
    #pesan hanya bisa diakses
    #di dalam fungsi
    pesan = "halo apa khabar"
    print(pesan)
print_pesan()
#var pesan tidak dikenal
print_pesan(pesan)
```

```
halo apa khabar
Traceback (most recent call last):
  File "D:/program/fungsi/contoh8.py",
    line 11, in <module> print_pesan(pesan)
NameError: name 'pesan' is not defined
```

Variabel global

```
=====
# fungsi/contoh8.py
# variabel global
#-----
-
def hitung(*nilai):
    jum=0
    for x in nilai:
        jum = jum + x
    print("Jumlah nilai",jum)
jum=0
hitung(20,20,30)
print("Jumlah di luar fungsi:",jum)
```

hasil keluaran:

```
Jumlah nilai 70
Jumlah di luar fungsi: 0
>>> |
```



## PRAKTIK

1. Buatlah program untuk menghitung bilangan Fibonacci adalah barisan yang didefinisikan secara rekursif sebagai berikut:

$$F(n) = \begin{cases} 0, & \text{jika } n = 0; \\ 1, & \text{jika } n = 1; \\ F(n-1) + F(n-2) & \text{jika tidak.} \end{cases}$$

berikut ini adalah barisan bilangan Fibonacci mulai dari n=1

1 1 2 3 5 8 13 21 34

```
# fungsi rekursif
# praktik51.py
```

```

#####
def fibonacci(n):
    if (n ==1) or (n==2):
        return 1
    else:
        return (fibonacci(n-1)+fibonacci(n-2))
n=int(input("Masukan n = "))
print ("F({:d})={:d}".format(n,fibonacci(n)))

```

coba dieksekusi dan amati hasilnya

2. Buatlah prpgram untuk menghitung faktorial,  $4! = 4*3*2*1$ , menggunakan iterasi (loop for) biasa seperti berikut:

```

#####
# menghitung faktorial tanpa
# tidak menggunakan rekursif
#####
n=int(input("Masukan N!"))
faktor=1;
for i in range(2,n+1):
    faktor=faktor*i;

print(n, ' Faktorial = ',faktor);

```

3. Selanjutnya menghitung faktorial menggunakan rekursif:

```

#####
# menghitung faktorial
# menggunakan rekursif
#####

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

n=int(input("Masukan N!"))
print(n, ' Faktorial = ',faktorial(n))

```

coba dieksekusi dan amati hasilnya

4. Buatlah program untuk menghitung kombinasi menggunakan rekursif, dengan rumus:

$$\binom{n}{k} = \frac{n!}{k! (n - k)!} \quad \text{for } 0 \leq k \leq n.$$

```
#####
# menghitung kombibinasi
# menggunakan rekursif
#####

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

def kombinasi(n, r):
    if (n < 1):
        return(0)
    else:
        return(faktorial(n)/(faktorial(r)* faktorial(n-r)));

print('>>> Kombinasi <<<<')
n = int(input('Masukkan bilangan bulat n : '))
r = int(input('Masukkan bilangan bulat r : '))

print ("Kombinasi C({:d},{:d})={:.2f}".format(n,r,
kombinasi(n,r)))
```

5. Membuat fungsi penjumlahan dengan menggunakan **return**, seperti berikut:

```
#-----
# fungsi tambah mengembalikan
# nilai hasil penjumlahan
# praktik53.py
#####

def tambah(x, y):
    return x + y
```

```
hasil = tambah(3, 5)
print("Hasil : ",hasil)
```

6. Berikut ini, jika dibuat fungsi tanpa return menggunakan variabel "hasil" lokal, berada di dalam fungsi tambah.

```
-----
# fungsi tambah mengembalikan
# nilai hasil penjumlahan
# praktik5.py
#*****
def tambah(x, y):
    hasil= x + y

tambah(3, 5)
print("Hasil : ",hasil)
```

apa hasilnya?

7. Mengacu pada program di atas dengan penambahan variable global di dalam fungsi:

```
#-----
# fungsi tambah mengembalikan
# nilai hasil penjumlahan
# praktik6.py
#*****
def tambah(x, y):
    global hasil
    hasil=x + y

hasil = tambah(3, 5)
print("Hasil : ",hasil)
```

Apa hasilnya?

8. Berikut buatlah program menggunakan variabel global yang dikenal di dalam atau di luar fungsi:

```
#*****
```

```

# variabel global
# dikenal di dalam atau di luar fungsi
#*****
def f():
    global x
    x=x+1
    return x

def g():
    global x
    x=x+3
    return x
x=10
print(f())
print(x)
print(g())
print(x)

```

Apa hasilnya?

9. Jika ingin mengembalikan hasil dua variabel atau lebih dari suatu fungsi. Ada beberapa pendekatan yang diambil. Pendekatannya dengan menggunakan kata kunci global.

```

1  #-----
2  # fungsi  mengembalikan
3  # nilai lebih dari satu hasil
4  # praktik57.py
5  #*****
6  alamat="Solo"
7  def profile():
8      global nama
9      global umur
10     nama = "Danny"
11     umur = 30
12     alamat="Yogyakarta"
13     print(alamat)
14
15 profile()
16 print(nama)
17 print(umur)
18 print(alamat)

```

Apa bedanya variabel "alamat" pada baris 12 dan baris 6

Apa hasilnya?

tambahkan di bawahnya

nama="Anto"

print(nama)

apa hasilnya?

10. Studi kasus : buatlah sub program untuk statistik, yang akan menghasilkan keluaran, jumlah, rata-rata, varian, standar deviasi, maksimal, minimal seperti pada kode program berikut:

$$s = \sqrt{\frac{\sum_{i=1}^n (xi - \bar{x})^2}{n-1}}$$

s = Standar Deviasi

n = jumlah sampel

x = rata-rata

xi = nilai x ke i

```
# *****
# fungsi statistik mengembalikan nilai
# jumlah, rata2, varian, standar deviasi
# maksimal, minimal
# tanpa menggunakan return
#*****
def stat(data):
    global jumlah
    global rata_rata
    global varian
    global std
    global max
    global min

    # menjumlahkan data
    jumlah=0
    for x in data:
        jumlah+=x

    # hitung rata-rata
    rata_rata = jumlah/len(data)

    # hitung varian
    jv=0
    for x in data:
        jv+=(x-rata_rata)**2
        varian=jv/len(data)-1

    # standar deviasi
```



```

std=varian**0.5
max=max(data)
min=min(data)

#Program Utama
data=(10,20,30,40,9,10)
print(">> Fungsi Statistik Menggunakan Variabel Global <<")
print("data =",data)
#menggil fungsi
stat(data)
print("Jumlah          =",jumlah)
print("Rata-rata        = ",rata_rata)
print("Varian           = ",varian)
print("Standar Deviasi = ",std)
print("Maximal          = ",max)
print("Minimal          = ",min)

```

amati hasilnya dan bantulah laporan analisisnya, apa yang terjadi jika tidak pakai `global jumlah`?



### LATIHAN

---

1. Buatlah program untuk menghitung permutasi  $n! = n * (n-1) * (n-2) * (n-3) * \dots * 3 * 2 * 1$
2. Buatlah program untuk menghitung deret  $S = 1+2+3+4+5$  menggunakan function rekursi
3. Buatlah program untuk menghitung deret  $S = 2+4+6+8+10$  menggunakan function rekursi



### TUGAS

---

Buatlah program untuk menghitung akar-akar persamaan kuadrat  $aX^2 + bX + c = 0$  dengan menggunakan 'Rumus ABC' sebagai berikut:

$$X_{1,2} = -b \pm \frac{\sqrt{b^2 - 4ac}}{2a}$$

dengan masukan nilai variabel nilai a,b,c, diskriminan  $D = (b*b) - (4*a*c)$ ,

Jika Diskriminan  $< 0$ , maka informasikan akar-akar persamaan IMAJINER,

Jika Diskriminan  $= 0$ , maka informasikan akar-akar persamaan tunggal  $= X = -b$

Hitung  $X_1 = -b + \left(\sqrt{\frac{D}{2a}}\right)$  dan  $X_2 = -b - \left(\sqrt{\frac{D}{2a}}\right)$



## REFERENSI

---