

ALGORITMA DAN PEMROGRAMAN

KULIAH 6 : Pengulangan

Dosen Pengampu:

Hasanuddin, S.Si., M.Si., Ph.D.





Manfaat Pengulangan

- Dalam banyak kasus, kita membutuhkan instruksi secara berulang (contoh: menghitung jumlah bilangan dari 1 sampai 100)
- Komputer dapat melakukan suatu instruksi secara berulang (menguntungkan manusia agar tidak menuliskan instruksi sebanyak pengulangan yang diinginkan)

Pengulangan dengan 'for'

- Jumlah loop tetap dan dispesifikasi.
- Diperlukan suatu variabel pencacah (iterator)
- Struktur:

```
>>>for target in object:  # list, tuple, etc.  
...     statement_1  
...     else:              # optional  
...     statement_2
```

Struktur for Loop yang Lain

```
>>>for target in object:
...     statement_1
...     if test: break           # exit loop now, skip else
...     if test: continue      # continue to next item
... else:
...     statement_2
```

Contoh:

```
>>>sum = 0
```

```
>>>for x in [1,2,3,4,5]:
```

```
...     sum = sum + x
```

```
...
```

```
>>>s = 'strong'
```

```
>>>for c in s:
```

```
...     print(c,end='')
```

```
...
```

Contoh:

```
>>> for x in range(5):
...     print(x)
...
>>> T = [(1,2),(3,4),(5,6)]
>>> for (a,b) in T:
...     print (a,b)
...
>>> D = {'a':1,'b':2,'c':3}
>>> for key in D:
...     print(key,' => ',D[key])
...
```

Nested for Loop

```
>>> x = [1,2,3,4,5,6]
>>> y = [3,7]
>>> for j in y:
...     for i in x:
...         if j == i:
...             print(j, ' was found')
...             break
...     else:
...         print(j, ' was not found')
```


Contoh

```
>>> s = "start"
```

```
>>> while s:
```

```
...     print(s)
```

```
...     s=s[1:]
```

```
...
```

```
>>> a= 0; b =10
```

```
>>> while a < b:
```

```
...     print (a, end=' ')
```

```
...     a = a + 1
```

```
...
```



Struktur Umum While Loop

```
>>>while test :  
...     statements  
...     if test : break  
...     if test : continue  
... else:  
...     statements  
...  
...
```

continue

- Jump to the top of header loop and skip the following statements in loop block

- Example:

```
>>> x= 10
```

```
>>> while x:
```

```
...     x = x - 1
```

```
...     if x % 2 != 0: continue
```

```
...     print (x)
```

```
...
```

- A clearer one but without continue:

```
...     if x% 2 == 0:
```

```
...         print(x)
```

break

- Immediate exit of loop including else if there is.

- Example:

```
>>> while True:
```

```
...     name = input(' enter a name :')
```

```
...     if name == 'stop' : break
```

```
...     age = input(' enter your age:')
```

```
...     print('name: ', name, 'age:', age)
```

```
...
```

Repeat/do until

- Bahasa pemrograman seperti C++ dan Pascal memiliki struktur loop do until atau semacamnya.
- Struktur ini tidak terdapat dalam Python. Tetapi, struktur semacam ini bisa kita buat sendiri seperti:

```
>>>while True:
```

```
    statements #loop body
```

```
    if exit_test: break
```

Example:

- Menjumlahkan deret bilangan:

```
>>> sum = 0
```

```
>>> for x in range(1,11):
```

```
    . . .     sum = sum + x
```

```
    . . .     print(sum)
```

Euclid Algoritma

```
def euclid():
    a = int(input('input sebuah bilangan bulat '))
    b = int(input('input bilangan bulat lainnya '))
    if (a<b):
        tmp = a
        a = b
        b = tmp
    if b == 0 :
        print ('FPB = ',a)

    while b != 0 :
        r = a % b
        a = b
        b = r
    print ('FPB =',a)
```


Penjumlahan Deret

```
def one_over_n():  
    n = int(input('masukkan total penjumlahan (n) = '))  
    total = 0  
    for x in range(1,n+1):  
        total = total + 1/x  
    print ('hasil penjumlahan 1/n untuk n = ',n, ' adalah ', total)
```

TUGAS

- Buat program untuk menghitung deret di bawah ini dengan input suku ke-N.

$$\sum_{n=1}^N \frac{1}{n^2}$$

- Tentukan akar dari suatu bilangan x:
(postponed)