

Fungsi

Perkuliahan Algoritma Pemrograman pada Semester Ganjil 2020

Contoh list bukan Fungsi

In []:

```
kalimat=input('masukkan kalimat =')
kata=input('masukkan kata =')
jumlah=kalimat.count(kata)
print(jumlah)
```

In []:

```
genap=[]
num=int(input('masukkan banyaknya bilangan = '))
bil=0
for i in range(num):
    genap.append(bil)
    bil=bil+2
print(genap)
```

Jika suatu code memerlukan suatu sub code yang sama, maka cara yang paling mudah adalah mengcopy-paste sub code tersebut. Akan tetapi cara ini akan menyulitkan programmer untuk debugging, maintenance suatu code. Oleh karena itu jika suatu code akan semakin kompleks, diperlukan fitur tambahan untuk code yang semakin kompleks ini dengan tujuan :

- mempermudah programmer ketika proses debugging code
- mempermudah programmer untuk maintenance code (edit, update code)
- memperkecil ukuran code
- mempermudah programmer untuk menggunakan kembali sub code yang sama pada code

Fitur tambahan ini dinamakan dengan **Fungsi/Function/Procedure**. Fungsi adalah suatu fitur yang dimiliki oleh bahasa pemrograman yang membungkus suatu sub code dengan nama fungsi atau prosedur tertentu.

Di dalam python, untuk membuat fungsi, diperlukan keywords **def**.

Untuk memanggil fungsi pada python, perlu didefinisikan terlebih dahulu sebelumnya.

Beberapa hal yang harus diperhatikan pada fungsi ini:

- argument/parameter : nilai yang dikirim oleh pemanggil fungsi
- return value : nilai balik yang dihasilkan oleh fungsi
- local atau Global Variable

Fungsi tanpa argument/parameter

```
** def namaFungsi ():**
```

In []:

```
def displayData():
    print('Hello World')

def sumNumber():
    a=int(input('bilangan 1= '))
    b=int(input('bilangan 2= '))
    c=a+b
    print('Hasil penjumlahan = ',c)

#Main Code#
displayData()
sumNumber()
```

Fungsi dengan Argument/Parameter

```
def namaFungsi(parameter1,parameter2,...):
```

In []:

```
def displayData2(a):
    print(a)
```

In []:

```
displayData2('Surabaya')
```

In []:

```
def sumNumber(a,b):
    c=a+b
    print('Hasil penjumlahan = ',c)

#-----Main Program-----#
num1=int(input('bilangan 1= '))
num2=int(input('bilangan 2= '))
sumNumber(num1,num2)

hasil=c*5
print(hasil)
```

In []:

```
sumNumber(10,4)
```

Fungsi tanpa return value

Pada bahasa pemrograman lain, suatu fungsi tanpa return value disebut dengan **Procedure/prosedur**. Pada python, tetap disebut dengan fungsi. Fungsi-fungsi yang dibuat sebelumnya adalah contoh fungsi tanpa return value. Berikut adalah contoh lain fungsi tanpa menggunakan return value (nilai balik)

In []:

```
def penjumlahan(a,b):  
    c=a+b  
    print(c)
```

In []:

```
penjumlahan(6,7)
```

Fungsi dengan return value

Return value atau nilai balik dibuat jika programmer membutuhkan nilai hasil fungsi untuk operasi-operasi selanjutnya pada code. Berikut contoh fungsi dengan return value

In []:

```
def sumNumber(a,b):  
    c=a+b  
    print('Hasil penjumlahan = ',c)  
  
def sumNumberReturn(a,b):  
    c=a+b  
    return c  
  
#-----Main Program-----#  
num1=int(input('bilangan 1= '))  
num2=int(input('bilangan 2= '))  
sumNumber(num1,num2)  
#hasil=7*sumNumber(num1,num2)  
hasil=7*sumNumberReturn(num1,num2)  
print(hasil)
```

In [1]:

```
def cekGenap(bil):  
    if bil%2==0:  
        print('Genap')  
    else:  
        print('Ganjil')
```

In []:

```
cekGenap(9)
```

In [3]:

```
def cekGenap(bil):  
    if bil%2==0:  
        return 'Genap'  
    else:  
        return 'Ganjil'
```

In [7]:

```
def cekGenap(bil):  
    if bil%2==0:  
        return True  
    else:  
        return False
```

In [10]:

```
def cekPrima(bil):  
    count=0  
    for i in range(bil):  
        if bil%(i+1)==0:  
            count=count+1  
    print(count)  
    if count==2:  
        return True  
    else:  
        return False
```

In [12]:

```
cekPrima(5)
```

2

Out[12]:

True

In [8]:

```
num=int(input('masukkan bilangan ='))  
print(cekGenap(num))  
if cekGenap(num):  
    hasil=num*3  
else:  
    hasil=num*2  
print(hasil)
```

masukkan bilangan = 7

False

14

Berikut contoh fungsi dengan return value yang lain

Kalimat dan teks yang ingin dicari merupakan data konstan, oleh karena itu diperlukan suatu parameter atau argumen agar kalimat ataupun teks merupakan inputan dari user

In []:

```
def numberOfWord(word,text):
    print(text)
    newText=text.split()
    i=0;
    count=0

    while i<len(newText):
        if newText[i]==word:
            count=count+1;
        i=i+1
    return(count)

#Main Program
fullText="Where is Waldo ? I think Waldo is hiding behind the wall, we have to find Waldo b
targetText="Waldo"
num=numberOfWord(targetText,fullText)
print ('there are ', num, 'Waldo in the text')
```

Local Variable

Local variabel adalah variabel yang dikenali didalam fungsi itu sendiri saja

In []:

```
def setPassword():
    password='tanggalLahirku'

def displayPassword():
    print('the password is',password)

#Main
setPassword()
displayPassword()
```

Latihan

Perbaiki fungsi-fungsi tersebut agar tidak terjadi error

In []:

```
def setPassword():
    password='tanggalLahirku'
    return password

def displayPassword(password):
    print('the password is',password)

#Main
passWd=setPassword()
displayPassword(passWd)
```

Global Variable

Suatu variabel yang dapat dikenali oleh semua bagian dari code termasuk semua fungsi

In []:

```
total=0
def jumlah(a,b):
    c=a+b
    global total
    total=total+c

jumlah(6,7)
print(total)
```

In []:

```
password='tanggalLahirku'

def displayPassword2():
    print('the password is',password)

#Main

displayPassword2()
```

In []:

```
password='tanggalLahirku'

def setPassword3():
    password='namaku'
def displayPassword3():
    print('the password is',password)

#Main

setPassword3()
displayPassword3()
```

In []

