

210411100085_Modul6_Fungsi

November 26, 2021

1 Fungsi

Tulis ringkasan atau penjelasan hal-hal berikut, dengan kata-kata kalian sendiri :

1. Sebutkan alasan kenapa kita harus menggunakan Fungsi (Function), jika perlu tambahkan contoh
2. Sebutkan Jenis Fungsi dan perbedaannya, syntax umum dari fungsi tersebut, berikan contoh

1. Sebutkan alasan kenapa kita harus menggunakan Fungsi (Function), jika perlu tambahkan contoh

- Agar code lebih rapi dan mudah dibaca
- Membuat code lebih efisien

#contoh Luas Segitiga

```
def luasSegitiga(alas,tinggi) :  
    luas  = alas*tinggi/2  
    return luas
```

```
luasBangun = luasSegitiga(2,3)  
luasBangun2= luasSegitiga(3,5)  
print("Segitiga1 = %f, Segitiga2 = %f " %(luasBangun,luasBangun2))
```

2. Sebutkan Jenis Fungsi dan perbedaannya, syntax umum dari fungsi tersebut, berikan contoh

Jenis Fungsi :

- Fungsi biasa tanpa parameter

Fungsi adalah potongan code yang dibuat tanpa parameter/return value

```
def namaFungsi() :  
    #code
```

#contoh Luas Segitiga

```
def luasSegitiga() :  
    alas  = 3  
    tinggi = 4  
    luas  = alas*tinggi/2  
    print(luas)
```

```
luasSegitiga()
```

- Fungsi biasa dengan parameter

Fungsi adalah potongan code yang dibuat dengan parameter/argument

```
def namaFungsi(parameter) :  
    #code
```

```
#contoh Luas Segitiga
```

```
def luasSegitiga(alas,tinggi) :  
    luas    = alas*tinggi/2  
    print(luas)
```

```
luasSegitiga(2,3)
```

- Fungsi dengan return value dan tanpa parameter

Fungsi adalah potongan code yang dibuat tanpa parameter/argument dan disertai return value.

```
def namaFungsi() :  
    #code  
    return nilai
```

```
#contoh Luas Segitiga
```

```
def luasSegitiga() :  
    alas    = 3  
    tinggi  = 4  
    luas    = alas*tinggi/2  
    return luas
```

```
luasBangun = luasSegitiga(2,3)  
print(luasBangun)
```

- Fungsi dengan return value dan parameter

Fungsi adalah potongan code yang dibuat dengan parameter/argument dan disertai return value.

```
def namaFungsi(parameter) :  
    #code  
    return nilai
```

```
#contoh Luas Segitiga
```

```
def luasSegitiga(alas,tinggi) :  
    luas    = alas*tinggi/2  
    return luas
```

```
luasBangun = luasSegitiga(2,3)  
print(luasBangun)
```

- Fungsi dengan return value ,parameter, dan nilai default

Fungsi adalah potongan code yang dibuat dengan parameter/argument dan disertai return value.

```
def namaFungsi(parameter=nilai_default,parameter2) :  
    #code  
    return nilai
```

#contoh Luas Segitiga

```
def luasSegitiga(alas=3,tinggi=2) :  
    luas    = alas*tinggi/2  
    return luas
```

```
luasBangun = luasSegitiga(2)  
print(luasBangun)
```

- Fungsi Recursif

Fungsi Recursif adalah teknik dalam memecahkan suatu masalah menggunakan fungsi yang memanggil fungsi tersebut sendiri sampai suatu kondisi tertentu.

```
def namaFungsi(parameter=nilai_default,parameter2) :  
    #code  
    if stopping_condition :  
        #code  
    else :  
        return namaFungsi(parameter,parameter2-1)
```

#Fibonacci

```
def faktorialRecursif(number):  
    if number==1 or number==0:  
        val=1  
    else:  
        val=number*faktorialRecursif(number-1)  
    return(val)
```

```
num = faktorialRecursif(5)  
print(num)
```

Secara garis besar terdapat perbedaan penggunaan fungsi berdasarkan daftar diatas, untuk fungsi yang tidak mengembalikan nilai digunakan untuk memecahkan masalah saja dan tidak menyimpan nilai untuk dikembalikan lagi, sedangkan fungsi yang mengembalikan nilai/return value akan mengembalikan suatu nilai yang dapat digunakan untuk operasi selanjutnya

2 Implementasi

2.1 Cek List

Buatlah Fungsi dan code untuk memanggil fungsi dengan menggunakan Python, antara lain :

- Fungsi createList untuk membentuk suatu list dengan parameter atau argumen berupa ukuran list

- Fungsi isPrime untuk menentukan apakah suatu bilangan adalah bilangan prima, dengan argumen adalah bilangan yang akan dicek
- Fungsi createListPrime untuk membuat suatu list yang berisi bilangan prima, dengan ketentuan :
 - argumen atau paramater adalah list dimana di setiap anggotanya akan dicek termasuk bilangan prima
 - gunakan fungsi isPrime yang sudah dibuat, untuk cek apakah suatu bilangan adalah bilangan prima
 - anggota-anggota list yang dihasilkan dari fungsi ini, tidak boleh sama [Hint: gunakan in]

```
[14]: def createList(listLen) :
    listNums = []
    for i in range(listLen) :
        listNums.append(int(input("==> Masukkan data ke- %d ="%(i))))
    return listNums

def isPrime(num) :
    prime = True
    for i in range(2,num) :
        if num%i == 0 :
            prime = False
            break
    return prime

def createListPrime(nums) :
    primeList = []
    for num in nums :
        if isPrime(num) and num not in primeList :
            primeList.append(num)
    return primeList

data = createList(5)
print("Data List = ",data)

prima = createListPrime(data)
print("Data Prima = ",prima)
```

```
==> Masukkan data ke- 0 = 4
==> Masukkan data ke- 1 = 5
==> Masukkan data ke- 2 = 5
==> Masukkan data ke- 3 = 7
==> Masukkan data ke- 4 = 8
Data List = [4, 5, 5, 7, 8]
```

Data Prima = [5, 7]

2.2 Operasi List

Buatlah Fungsi dan code untuk memanggil fungsi dengan menggunakan Python, antara lain :

- Fungsi createList untuk membentuk suatu list dengan parameter atau argumen berupa ukuran list
- Fungsi avgList untuk menghitung rata-rata suatu list, dengan argumen adalah list
- Fungsi addList untuk menjumlahkan dua buah list yang memiliki ukuran yang sama

```
[1]: def createList(lenList) :  
    numsList = []  
    for i in range(lenList) :  
        numsList.append(int(input("==> Masukkan data ke- %d = " % (i))))  
    return numsList  
  
def addList(list1,list2) :  
    if len(list1) == len(list2) :  
        result = []  
        for i in range(len(list1)) :  
            result.append(list1[i]+list2[i])  
        return result  
    else :  
        return "Ukuran List Tidak sama"  
  
def avgList(numsList) :  
    return sum(numsList)/len(numsList)  
  
data1 = createList(6)  
print("Data - 1 = %s , Rata-rata List = %g " % (data1,avgList(data1)))  
  
data2 = createList(6)  
print("Data - 2 = %s , Rata-rata List = %g " % (data2,avgList(data2)))  
  
hasil = addList(data1,data2)  
print("%s + %s = %s " % (data1,data2,hasil))
```

==> Masukkan data ke- 0 = 4

==> Masukkan data ke- 1 = 2

==> Masukkan data ke- 2 = 4

==> Masukkan data ke- 3 = 7

==> Masukkan data ke- 4 = 8

==> Masukkan data ke- 5 = 2

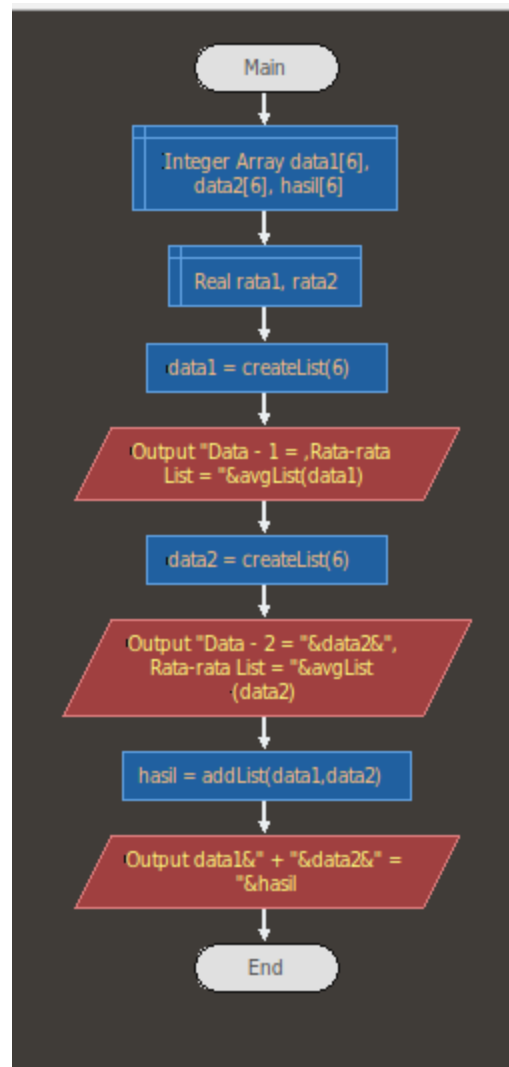
Data - 1 = [4, 2, 4, 7, 8, 2] , Rata-rata List = 4.5

==> Masukkan data ke- 0 = 1

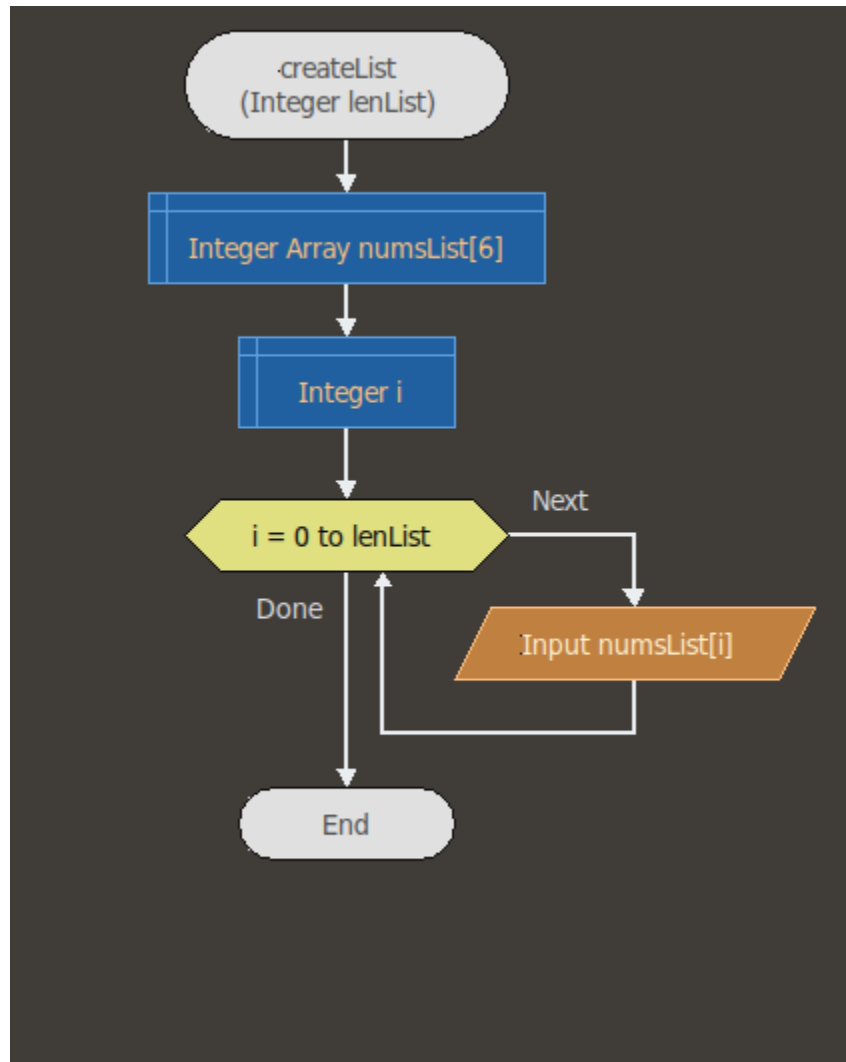
==> Masukkan data ke- 1 = 3
 ==> Masukkan data ke- 2 = 10
 ==> Masukkan data ke- 3 = 4
 ==> Masukkan data ke- 4 = 5
 ==> Masukkan data ke- 5 = 6
 Data - 2 = [1, 3, 10, 4, 5, 6] , Rata-rata List = 4.83333
 [4, 2, 4, 7, 8, 2] + [1, 3, 10, 4, 5, 6] = [5, 5, 14, 11, 13, 8]

Flowchart Function Buatlah flowchart dari fungsi dan main code yang sudah dikerjakan pada nomor sebelumnya (Operasi List).

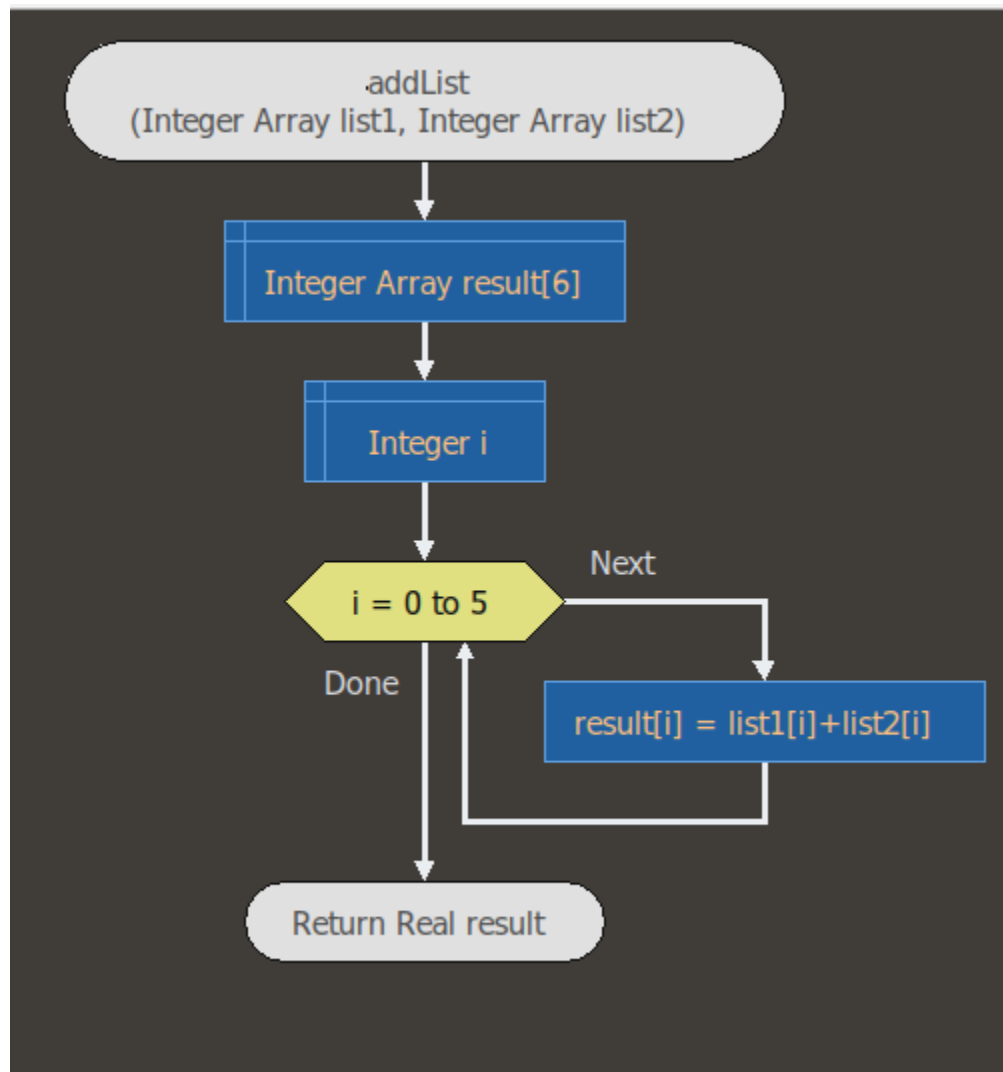
Main



Function createList



Function `addList`



Function avgList

