

▼ 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, berikana contoh

1.
 - Dapat dipanggil kembali (berulang kali), dimanapun dibutuhkan fungsi tersebut
 - Code jadi lebih ringkas, jumlah baris lebih sedikit
 - Code menjadi lebih mudah untuk dibaca: edit, update, check error

2.
 - Fungsi 1 - Tanpa Parameter, Tanpa Return Value

Yaitu fungsi yang hanya mengerjakan syntax yang ada di fungsi tersebut (nilainya tidak dapat diubah-ubah) dan hasil dari fungsi tersebut tidak dapat dioperasikan kembali.

- Fungsi 2 - Dengan Parameter atau Argumen

Yaitu fungsi yang dapat diganti nilainya dengan parameter namun hasil dari fungsi tersebut tidak dapat dioperasikan kembali.

- Fungsi 3 - Dengan Parameter atau Argumen dan Dengan Return Value Yaitu fungsi yang dapat diganti nilainya dengan parameter dan hasil dari fungsi tersebut dapat dioperasikan kembali.

#Contoh Fungsi 1

```
def penjumlahan():  
    bil1=int(input('masukkan bil 1 = '))  
    bil2=int(input('masukkan bil 2 = '))  
    hasil=bil1+bil2  
    print(hasil)
```

penjumlahan()

```
    masukkan bil 1 = 5  
    masukkan bil 2 = 7  
    12
```

#Contoh Fungsi 2

```
def penjumlahan(bil1,bil2):  
    hasil=bil1+bil2
```

```
print(hasil)

penjumlahan(6,8)

14
```

#Contoh Fungsi 3

```
def penjumlahan(bil1,bil2):
    hasil=bil1+bil2
    print(hasil)
    return hasil

penjumlahan(4,5)*3

9
27
```

▼ 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 ataukah tidak
 - 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]

Contoh output yang dihasilkan dapat dilihat pada Gambar 1.

```
data=createList(10)
print('data list = ',data)
prima=createListPrime(data)
print('data prima = ',prima)
```

```
Masukkan data ke - 0 : 7
Masukkan data ke - 1 : 2
Masukkan data ke - 2 : 4
Masukkan data ke - 3 : 6
Masukkan data ke - 4 : 5
Masukkan data ke - 5 : 2
Masukkan data ke - 6 : 9
Masukkan data ke - 7 : 5
Masukkan data ke - 8 : 8
Masukkan data ke - 9 : 1
data list = [7, 2, 4, 6, 5, 2, 9, 5, 8, 1]
data prima = [7, 2, 5]
```

(a) Size List utama = 10

```
data=createList(15)
print('data list = ',data)
prima=createListPrime(data)
print('data prima = ',prima)
```

```
Masukkan data ke - 0 : 8
Masukkan data ke - 1 : 1
Masukkan data ke - 2 : 2
Masukkan data ke - 3 : 6
Masukkan data ke - 4 : 3
Masukkan data ke - 5 : 5
Masukkan data ke - 6 : 2
Masukkan data ke - 7 : 9
Masukkan data ke - 8 : 0
Masukkan data ke - 9 : 11
Masukkan data ke - 10 : 3
Masukkan data ke - 11 : 2
Masukkan data ke - 12 : 7
Masukkan data ke - 13 : 9
Masukkan data ke - 14 : 1
data list = [8, 1, 2, 6, 3, 5, 2, 9, 0, 11, 3, 2, 7, 9, 1]
data prima = [2, 3, 5, 11, 7]
```

(b) Size List utama = 15

Gambar 1: Fungsi createList dan createListPrime

```
def createList(sizeList):
    listCreate=[]
    for i in range(sizeList):
        temp='Masukkan data ke - '+str(i)+' : '
        listCreate.append(int(input(temp)))
    return listCreate

def isPrime(bil):
    pembagi=0
    if bil>=2:
        for i in range(1,bil+1):
            if bil%i==0:
                pembagi+=1
        if pembagi==2:
            return True
        else:
            return False
    else:
        return False

def createListPrime(list):
    listPrime=[]
    for n in list:
        prime=isPrime(n)
        if prime==True:
            if n not in listPrime:
```

```

        listPrime.append(n)
    return listPrime

data=createList(10)
print('data list = ',data)
prima=createListPrime(data)
print('data prima = ',prima)

Masukkan data ke - 0 : 7
Masukkan data ke - 1 : 2
Masukkan data ke - 2 : 4
Masukkan data ke - 3 : 6
Masukkan data ke - 4 : 5
Masukkan data ke - 5 : 2
Masukkan data ke - 6 : 9
Masukkan data ke - 7 : 5
Masukkan data ke - 8 : 8
Masukkan data ke - 9 : 1
data list = [7, 2, 4, 6, 5, 2, 9, 5, 8, 1]
data prima = [7, 2, 5]

```

```

data=createList(15)
print('data list = ',data)
prima=createListPrime(data)
print('data prima = ',prima)

Masukkan data ke - 0 : 8
Masukkan data ke - 1 : 3
Masukkan data ke - 2 : 2
Masukkan data ke - 3 : 6
Masukkan data ke - 4 : 3
Masukkan data ke - 5 : 5
Masukkan data ke - 6 : 2
Masukkan data ke - 7 : 9
Masukkan data ke - 8 : 0
Masukkan data ke - 9 : 11
Masukkan data ke - 10 : 3
Masukkan data ke - 11 : 2
Masukkan data ke - 12 : 7
Masukkan data ke - 13 : 9
Masukkan data ke - 14 : 1
data list = [8, 3, 2, 6, 3, 5, 2, 9, 0, 11, 3, 2, 7, 9, 1]
data prima = [3, 2, 5, 11, 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

Contoh output yang dihasilkan dapat dilihat pada Gambar 2.

```
data1=createList(4)
print('data - 1 = ',data1,' ; rata-rata list =', avgList(data1))
data2=createList(5)
print('data - 2 = ',data2,' ; rata-rata list =', avgList(data2))
hasil=addList(data1,data2)
print(data1,'+',data2,'=',hasil)
```

```
Masukkan data ke - 0 : 4
Masukkan data ke - 1 : 1
Masukkan data ke - 2 : 2
Masukkan data ke - 3 : 4
data - 1 = [4, 1, 2, 4] ; rata-rata list = 2.75
Masukkan data ke - 0 : 4
Masukkan data ke - 1 : 7
Masukkan data ke - 2 : 5
Masukkan data ke - 3 : 3
Masukkan data ke - 4 : 8
data - 2 = [4, 7, 5, 3, 8] ; rata-rata list = 5.4
[4, 1, 2, 4] + [4, 7, 5, 3, 8] = Ukuran List Tidak Sama
```

(a) Ukuran List tidak sama

```
data1=createList(6)
print('data - 1 = ',data1,' ; rata-rata list =', avgList(data1))
data2=createList(6)
print('data - 2 = ',data2,' ; rata-rata list =', avgList(data2))
hasil=addList(data1,data2)
print(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 : 4
Masukkan data ke - 2 : 10
Masukkan data ke - 3 : 4
Masukkan data ke - 4 : 5
Masukkan data ke - 5 : 6
data - 2 = [1, 4, 10, 4, 5, 6] ; rata-rata list = 5.0
[4, 2, 4, 7, 8, 2] + [1, 4, 10, 4, 5, 6] = [5, 6, 14, 11, 13, 8]
```

(b) Ukuran List Sama

```
data1=createList(3)
print('data - 1 = ',data1,' ; rata-rata list =', avgList(data1))
data2=createList(3)
print('data - 2 = ',data2,' ; rata-rata list =', avgList(data2))
hasil=addList(data1,data2)
hasil2=addList(hasil,hasil)
print(data1,'+',data2,'=',hasil)
print(hasil,'+',hasil,'=',hasil2)
```

```
Masukkan data ke - 0 : 2
Masukkan data ke - 1 : 3
Masukkan data ke - 2 : 4
data - 1 = [2, 3, 4] ; rata-rata list = 3.0
Masukkan data ke - 0 : 2
Masukkan data ke - 1 : 6
Masukkan data ke - 2 : 7
data - 2 = [2, 6, 7] ; rata-rata list = 5.0
[2, 3, 4] + [2, 6, 7] = [4, 9, 11]
[4, 9, 11] + [4, 9, 11] = [8, 18, 22]
```

(c) Ukuran List sama dan dilakukan Operasi Penjumlahan Kembali

Gambar 2: Operasi List

```
def createList(sizeList):
    listCreate=[]
    for i in range(sizeList):
        temp='Masukkan data ke - '+str(i)+' : '
        listCreate.append(int(input(temp)))
    return listCreate

def avgList(list):
    total=0
    for i in list:
        total+=i
    rata2=total/len(list)
```

```
return rata2
```

```
def addList(list1,list2):
    listJumlah=[]
    if len(list1)==len(list2):
        for i in range(len(list1)):
            tambah=list1[i]+list2[i]
            listJumlah.append(tambah)
        return listJumlah
    else:
        tdk sama='Ukuran List Tidak Sama'
        return tdk sama
```

```
data1=createList(4)
print('data - 1 = ',data1,' ; rata-rata list = ',avgList(data1))
data2=createList(5)
print('data - 2 = ',data2,' ; rata-rata list = ',avgList(data2))
hasil=addList(data1, data2)
print(data1,'+',data2,'=',hasil)
```

```
Masukkan data ke - 0 : 4
Masukkan data ke - 1 : 1
Masukkan data ke - 2 : 2
Masukkan data ke - 3 : 4
data - 1 = [4, 1, 2, 4] ; rata-rata list = 2.75
Masukkan data ke - 0 : 4
Masukkan data ke - 1 : 7
Masukkan data ke - 2 : 5
Masukkan data ke - 3 : 3
Masukkan data ke - 4 : 8
data - 2 = [4, 7, 5, 3, 8] ; rata-rata list = 5.4
[4, 1, 2, 4] + [4, 7, 5, 3, 8] = Ukuran List Tidak Sama
```

```
data1=createList(6)
print('data - 1 = ',data1,' ; rata-rata list = ',avgList(data1))
data2=createList(6)
print('data - 2 = ',data2,' ; rata-rata list = ',avgList(data2))
hasil=addList(data1, data2)
print(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 : 4
Masukkan data ke - 2 : 10
Masukkan data ke - 3 : 4
```

```

Masukkan data ke - 4 : 5
Masukkan data ke - 5 : 6
data - 2 = [1, 4, 10, 4, 5, 6] ; rata-rata list = 5.0
[4, 2, 4, 7, 8, 2] + [1, 4, 10, 4, 5, 6] = [5, 6, 14, 11, 13, 8]

```

```

data1=createList(3)
print('data - 1 = ',data1,' ; rata-rata list = ',avgList(data1))
data2=createList(3)
print('data - 2 = ',data2,' ; rata-rata list = ',avgList(data2))
hasil=addList(data1, data2)
hasil2=addList(hasil, hasil)
print(data1,'+',data2,'=',hasil)
print(hasil,'+',hasil,'=',hasil2)

```

```

Masukkan data ke - 0 : 2
Masukkan data ke - 1 : 3
Masukkan data ke - 2 : 4
data - 1 = [2, 3, 4] ; rata-rata list = 3.0
Masukkan data ke - 0 : 2
Masukkan data ke - 1 : 6
Masukkan data ke - 2 : 7
data - 2 = [2, 6, 7] ; rata-rata list = 5.0
[2, 3, 4] + [2, 6, 7] = [4, 9, 11]
[4, 9, 11] + [4, 9, 11] = [8, 18, 22]

```

▼ 2.3 List Comprehension

2.3.1 Konsep

Jelaskan yang dimaksud dengan list comprehension, dan berikan contohnya

Jawaban:

List Comprehension merupakan pendefinisian pada tipe data list secara singkat tanpa membuat sebuah fungsi. contohnya adalah sebagai berikut:

```

#mengkuadratkan anggota list1 dan dimasukkan kedalam list2 menggunakan for
list1=[1, 2, 3, 4, 5]
list2=[i**2 for i in list1]

print(list2)

[1, 4, 9, 16, 25]

```

```

#list comprehension menggunakan if
list1=[1, 2, 3, 4, 5]
list2=[i for i in list1 if i!=3]

```

```
print(list2)

[1, 2, 4, 5]

#list comprehension menggunakan if else
list1=[1, 2, 3, 4, 5]
list2=[i if i != 5 else 6 for i in list1]

print(list2)

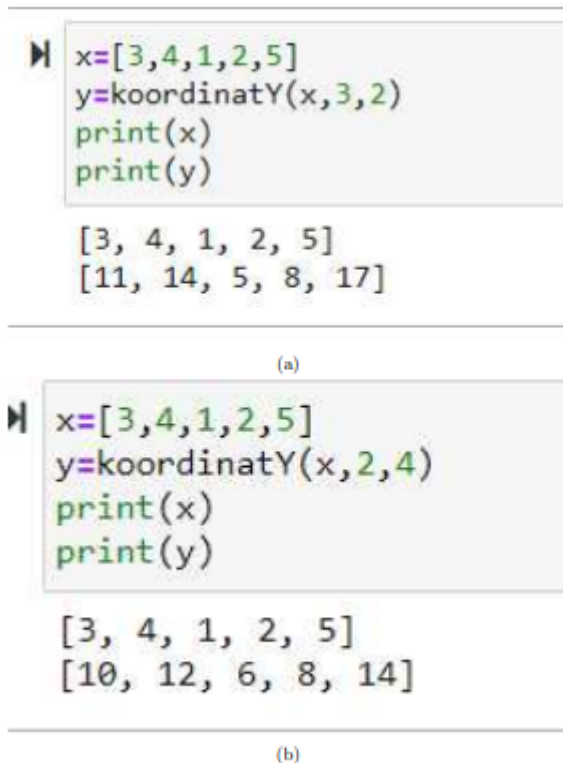
[1, 2, 3, 4, 6]
```

▼ 2.3.2 Garis Lurus

Buatlah fungsi dengan return value berupa list (y), dimana list tersebut dibuat dengan menggunakan list comprehension yang menghasilkan koordinat y dari suatu garis lurus dengan persamaan $y = mx + c$, dimana:

- x adalah list yang telah diketahui
- m adalah gradient, dan c adalah konstanta
- x, m, c adalah parameter fungsi

Contoh output yang dihasilkan dapat dilihat pada Gambar 3



Gambar 3: List Comprehension - Fungsi Garis Lurus

```
def koordinatY(x,m,c):
```



```
y=[]  
for xi in x:  
    pers=m*xi+c  
    y.append(pers)  
return y
```

```
x=[3,4,1,2,5]  
y=koordinatY(x,3,2)  
print(x)  
print(y)
```

```
[3, 4, 1, 2, 5]  
[11, 14, 5, 8, 17]
```

```
x=[3,4,1,2,5]  
y=koordinatY(x,2,4)  
print(x)  
print(y)
```

```
[3, 4, 1, 2, 5]  
[10, 12, 6, 8, 14]
```

[Produk berbayar Colab](#) - [Batalan kontrak di sini](#)

✓ 0 d selesai pada 20.58

