

## ▼ 1 Konsep

1. Jelaskan secara lengkap mengenai fungsi (Tujuan menggunakan Fungsi, Syntax tanpa parameter, syntax dengan parameter, syntax return value, Contoh masing-masing jenis fungsi tersebut)
2. Jelaskan tentang Fungsi Rekursif (komponen yang harus ada dalam fungsi rekursif, contoh fungsirekursif)

### Konsep\_1

1. Fungsi adalah sebuah blok kode (dapat diasumsikan variable) yang dapat dipanggil oleh bagian lain dari program untuk melakukan tugas tertentu yang bertujuan untuk membantu dalam mengelompokkan sebuah code/syntax menjadi bagian-bagian tertentu yang memudahkan dalam pengembangan ataupun pemecahan masalah.

- Syntax tanpa parameter

```
def namaFungsi():  
    syntax_1  
    syntax_2  
    ....  
    syntax_n  
namaFungsi()
```

- contoh

```
def cetakHello():  
    print("Hello")  
cetakHello()
```

- Syntax dengan parameter

```
def namaFun(param):  
    syntax_1  
    syntax_2  
    ....  
    syntax_n
```

- contoh

```
def cetakSelamat(nama):  
    print("Selamat Pagi",nama)  
cetakSelamat("Dody")
```

- Syntax dengan return value

```
def nameFunction():  
    syntax_1  
    ...  
    syntax_n  
    return nilai
```

- contoh

```
def penjumlahan(num1,num2):  
    return (num1 + num2)  
penjumlahan(3,7)
```

2. Recursive adalah sebuah konsep pemrograman dimana suatu fungsi dapat memanggil dirinya sendiri. Recursive digunakan untuk menyelesaikan sebuah persoalan yang dapat dipecahkan dengan cara membagi persoalan menjadi lebih kecil yang lebih mudah diselesaikan.

- Komponen yang harus ada dalam fungsi recursive:

1. Base case, kondisi yang digunakan untuk menghentikan recursive, biasanya base case adalah persoalan terkecil atau persoalan paling sederhana yang dapat langsung di pecahkan tanpa perlu recursive.
2. sintaks yang dapat merubah state dari permasalahan secara berulang sehingga semakin lama solusi permasalahan menuju base case yang sudah dibuat
3. Pemanggilan dirinya sendiri dengan parameter yang lebih sederhana atau lebih kecil sampai mencapai base case

- contoh :

```
def faktorialRecursive(num):
    if num == 1:
        return 1
    else:
        return num * faktorialRecursive(num-1)
faktorialRecursive(5)
```

## ▼ 2 Implementasi #1

Terdapat suatu deret aritmatika dengan persamaan  $Un = a + (n-1)b$ , dimana  $Un$  merupakan suku ke- $n$ ,  $a$  adalah suku awal, dan  $b$  merupakan beda atau selisih antara suku yang berdekatan. Buatlah suatu list berisi data-data yang memenuhi persamaan deret aritmatika tersebut dengan minimal banyaknya anggota adalah 500, dan buatlah fungsi-fungsi berikut ini :

1. Fungsi iteratif untuk menghitung total semua bilangan yang terdapat pada deret aritmatika tersebut
2. Fungsi Rekursif untuk menghitung total semua bilangan yang terdapat pada deret aritmatika tersebut

```
1 # 2 Implementasi #1
2
3 def deretAritmatika(a,b,n):
4     hasil = []
5     for i in range(a,a+n*b,b):
6         hasil.append(i)
7     return hasil
8
9 def sumIteratif(listData):
10    hasil=0
11    for i in listData:
12        hasil+=i
13    return hasil
14
15 def sumRecursive(listData):
16    if len(listData) == 1:
17        return listData[0]
18    else:
19        return listData[0] + sumRecursive(listData[1:])
20
21 deret = aritmatika(3,4,500)
22 print('List Deret Aritmatika : \n',deret)
23 print()
24 jumlahDeret = sumIteratif(deret)
25 print('Total Deret dengan Iterarif : ',jumlahDeret)
26 jumlahDeret2 = sumRecursive(deret)
27 print('Total Deret dengan Recursive : ',jumlahDeret2)
```

List Deret Aritmatika :

[3, 7, 11, 15, 19, 23, 27, 31, 35, 39, 43, 47, 51, 55, 59, 63, 67, 71, 75, 79, 83, 87, 91, 95, 99, 103, 107, 111, 115, 119, 123, 1

Total Deret dengan Iterarif : 500500

Total Deret dengan Recursive : 500500

## ▼ 3 Implementasi #2


Buatlah suatu list yang memiliki 500 bilangan secara acak. Buatlah fungsi-fungsi berikut ini

1. Buatlah fungsi iteratif untuk mencari nilai minimal atau maksimal dari suatu list (tambahkan parameter min atau maks untuk mencari nilai minimal atau maksimal)
2. Buatlah fungsi rekursif untuk mencari nilai minimal maksimal dari suatu list (tambahkan parametermin atau maks untuk mencari nilai minimal atau maksimal)

```

2
3 import random as rd
4 def listAcak(size):
5     data = []
6     for i in range(size):
7         data.append(rd.randint(0,999))
8     return data
9
10 def minMaxIteratif(data,find):
11     if find == 'maksimal':
12         maks = data[0]
13         for i in data:
14             if maks < i :
15                 maks=i
16         return maks
17
18     elif find == 'minimal':
19         mins = data[0]
20         for j in data:
21             if mins > j :
22                 mins=j
23         return mins
24
25     else:
26         return 'Parameter Salah'
27
28 def minMaxRecursive(data,find):
29     if find == 'maksimal':
30         if len(data) == 1:
31             return data[0]
32         else:
33             maks = minMaxRecursive(data[1:],find)
34             if data[0] > maks :
35                 return data[0]
36             else:
37                 return maks
38
39     if find == 'minimal':
40         if len(data) == 1:
41             return data[0]
42         else:
43             mins = minMaxRecursive(data[1:],find)
44             if data[0] < mins :
45                 return data[0]
46             else:
47                 return mins
48     else :
49         return 'Parameter Salah'
50
51 data = listAcak(500)
52 print('List Acak :\n',data)
53 print()
54 minList = minMaxIteratif(data,'minimal')
55 maxList = minMaxIteratif(data,'maksimal')
56 print('Menggunakan Iteratif :','\nMinimal dari list = ',minList,'\nMaksimal dari list = ',maxList)
57 minList1 = minMaxRecursive(data,'minimal')
58 maxList1 = minMaxRecursive(data,'maksimal')
59 print('Menggunakan Recursive :','\nMinimal dari list = ',minList1,'\nMaksimal dari list = ',maxList1)

```

 List Acak :  
[14, 944, 294, 540, 227, 855, 814, 6, 902, 152, 70, 410, 549, 83, 687, 131, 490, 681, 17, 987, 240, 778, 402, 598, 291, 190, 432,

Menggunakan Iteratif :  
Minimal dari list = 0  
Maksimal dari list = 999  
Menggunakan Recursive :  
Minimal dari list = 0  
Maksimal dari list = 999

