

LAPORAN PRAKTIKUM PEMROGRAMAN

Pertemuan 25 - Higher order function and recursive function



Dosen Pengampu : Tri Hadiah Muliawati S. ST., M. Kom

Disusun Oleh :

Najma Akmalina Ni'Ma	3322600015
Bagas Cahya Fajar Bastian	3322600016
Bayu Kurniawan	3322600019
Muhammad Rafi Jauhar Pralistio	3322600029
Yusuf Budi Raharjo Yusuf Budi	3322600030

PRODI SAINS DATA TERAPAN
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
NOVEMBER 2022

LATIHAN 1

MAP()

1.

```
# LATIHAN 1
# MAP FUNCTION
print("-"*5+"PROGRAM PENAMBAHAN ANGKA BERURUTAN"+"-"*5)
while True:
    angka1 = int(input("Masukkan panjang angka pertama yang ingin dijumlahkan: "))
    if angka1 >= 0:
        break
    else:
        print("Tolong masukkan angka yang benar!!!")
lst = []
lst1 = [x+1 for x in range(angka1)]
print(lst1)

while True:
    angka2 = int(input("Masukkan panjang angka kedua yang ingin dijumlahkan: "))
    if angka2 >= 0:
        break
    else:
        print("Tolong masukkan angka yang benar!!!")
lst = []
lst2 = [y+1 for y in range(angka2)]
print(lst2)

def penambahan(satu,dua):
    return satu + dua

if lst1 > lst2 or lst2 > lst1:
    print("Perlu diketahui, hasil angka yang dijumlahkan adalah angka dengan urutan terkecil.")

hasil = list(map(penambahan, lst1, lst2))
print("\n Hasil penambahan adalah:",hasil)
print("-"*5+"PROGRAM SELESAI DIJALANKAN"+"-"*5)

-----PROGRAM PENAMBAHAN ANGKA BERURUTAN-----
Masukkan panjang angka pertama yang ingin dijumlahkan: 15
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
Masukkan panjang angka kedua yang ingin dijumlahkan: 12
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Perlu diketahui, hasil angka yang dijumlahkan adalah angka dengan urutan terkecil.

Hasil penambahan adalah: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24]
-----PROGRAM SELESAI DIJALANKAN-----
```

Analisis: Membuat 2 list dari user input, setelahnya disimpan dalam variabel bernama lst. Membuat fungsi def untuk menambah kedua variabel tersebut. Setelahnya buat list dari map function dengan isi adalah def function, dan kedua list

2.

```
def jumlahkata(kata):
    if len(kata)%2 == 0:
        return "Ini Genab loh"
    else:
        return "Ganjil kali"

kata = ['hsgdjskius','jshsduuaj','kamu']
listkata =list(map(jumlahkata,kata))
print(listkata)

['Ini Genab loh', 'Ganjil kali', 'Ini Genab loh']
```

Analisis: Membuat user defined function jumlah kata dengan parameter kata, jika panjang kata modulus 2 sama dengan nol akan retron string "Ini Genab loh" dan jika tidak akan retron string "Ganjil kali" dari list kata yang berisi kata = ['hsgdjskius','jshsduuaj','kamu']

menggunakan fungsi map dari user defined function di listkata untuk mengecek dari variabel kata apakah jumlah kata genap atau tidak

FILTER()

3.

```
# LATIHAN 1
# FILTER FUNCTION
print("-"*5+"PROGRAM PENYARINGAN KATA-KATA"+"-"*5)
user = input("Masukkan kata yang diinginkan: ")
user1 = user.upper()
vocal = input("Masukkan kata pemfilter: ")
vocal1 = vocal.upper()
def saring_kata(kata):
    if kata not in vocal1:
        return True

sdh_difilter = list(filter(saring_kata, user1))
print(sdh_difilter)
print("-"*5+"PROGRAM SELESAI DIJALANKAN"+"-"*5)
```

-----PROGRAM PENYARINGAN KATA-KATA-----
 Masukkan kata yang diinginkan: saya bahagia sekali
 Masukkan kata pemfilter: aieuo
 ['S', 'Y', ' ', 'B', 'H', 'G', ' ', 'S', 'K', 'L']
 -----PROGRAM SELESAI DIJALANKAN-----

Analisis: User menginput kata yang diinginkan untuk digunakan, dan kata filter. Membuat fungsi def jika kata yang diinginkan tidak terdapat pada kata filter maka simpan sebagai True, membuat list fungsi dari filter dengan isi adalah fungsi def serta input user kata yang diinginkan.

4.

```
angka = [-12,-22,-43,131,-231,102,120,9,12,31,-1,-2,90]
angka_positif= filter(lambda n: n > 0, angka)
print(list(angka_positif))

[131, 102, 120, 9, 12, 31, 90]
```

Analisis: membuat program untuk mengetahui nilai positif apa saja dari list angka, Dengan lambda akan mengambil nilai yang hanya $n > 0$ dan akan mencetak nilai yang lebih dari nol saja

REDUCE()

5.

```
# LATIHAN 1
# REDUCE FUNCTION

print("-"*5+"PROGRAM UNTUK MENENTUKAN NILAI FAKTORIAL"+"-"*5)
from functools import reduce
while True:
    angka = int(input("Masukkan nilai faktorial yang diinginkan: "))
    if angka > 0:
        break
    else:
        print("Tolong masukkan angka yang benar!!!")
lst = []
lst_satu = [x+1 for x in range(angka)]

def faktorial(a, b):
    return a*b

maka = reduce(faktorial, lst_satu)
print("Hasil Faktorial dari", angka, "adalah",maka)
print("-"*5+"PROGRAM SELESAI DIJALANKAN"+"-"*5)

-----PROGRAM UNTUK MENENTUKAN NILAI FAKTORIAL-----
Masukkan nilai faktorial yang diinginkan: 6
Hasil Faktorial dari 6 adalah 720
-----PROGRAM SELESAI DIJALANKAN-----
```

Analisis: Membuat list dari input user yang dimana harus berupa integer atau angka yang lebih dari 0, jika tidak maka user akan diminta untuk menginput kembali. Membuat fungsi untuk mengalikan a dengan b, setelahnya buat variabel reduce dengan isi adalah fungsi def dan variabel list yang telah dibuat. Cetak variabel tersebut.

6.

```
#REDUCE ()
from functools import reduce

daftar_barang = [
    {"nama": "telor", "harga": 59900, "berat_kg": 1.312},
    {"nama": "beras", "harga": 19550, "berat_kg": 2.300 },
    {"nama": "gula", "harga": 9130, "berat_kg": 0.150},
    {"nama": "cabai", "harga": 3049, "berat_kg": 0.789},
    {"nama": "bawang merah", "harga": 33022, "berat_kg": 3.578 },
    {"nama": "bawang putih", "harga": 67079, "berat_kg": 9.920 },
    {"nama": "kacang tanah", "harga": 45031, "berat_kg": 1.123 },
    {"nama": "kedelai", "harga": 93027, "berat_kg": 0.521},
    {"nama": "kunyit", "harga": 10090, "berat_kg": 1.300}
]

total_price = reduce(lambda sum, product: sum + product["harga"], daftar_barang, 0)
print("Total tagihan belanja anda adalah",total_price,"Rupiah")

Total tagihan belanja anda adalah 339878 Rupiah
```

Analisis: Membuat program jumlah keseluruhan dari total dari key harga untuk mengetahui total tagihan belanja yang harus dibayar.Menggunakan lambda untuk menjumlahkan dari keseluruhan key dari value harga.

LATIHAN 2

Soal 3

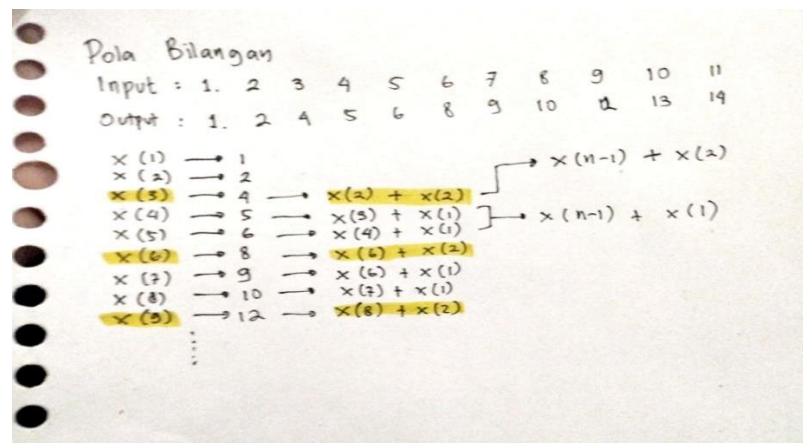
```
# kasus 3
# input = 1 2 3 4 5 6 7 8 9 10 11
# output= 1 2 4 5 6 8 9 10 12 13 14

def sum(n):
    if n == 1:
        return 1
    elif n==2:
        return 2
    elif n%3 == 0:
        return sum(n-1) + sum(2)
    else:
        return sum(n-1) + sum(1)

num = [1,2,3,4,5,6,7,8,9,10,11]
hasil = list(map(sum,num))
print(hasil)
```

[1, 2, 4, 5, 6, 8, 9, 10, 12, 13, 14]

Analisis:



$x(1)$ dan $x(2)$ dijadikan basis atau patokan

Pola yang pertama menunjukkan $x(n-1) + x(2)$, dimana pola tersebut terjadi ketika kondisi nilai n bisa dibagi 3 atau $n\%3$

Sedangkan pola kedua menunjukkan $x(n-1) + x(1)$, dimana pola ini terjadi ketika nilai $n!=1, n!=2$, nilai n bukan $\%3$, maka dimasukkan kondisi else

Soal 4

```
#soal 4
# input = 1 2 3 4 5 6 7 8 9 10 11 12
# output= 1 2 3 2 3 4 5 4 5 6 7 6

x = int(input("masukkan angka : "))

def soal4 (x):
    if x < 4 :
        return x
    else :
        return (x // 4)*2 + x%4

print(soal4(x))
```

masukkan angka : 5
3

masukkan angka : 9
5

masukkan angka : 12
6

masukkan angka : 4
2

Analisis:

- Membuat variabel x yang bernilai integer dan memberiiinput agar user bisa input angka
- Men def soal4 agar menjadi fungsi lalu memberikan parameter x
- Selanjutnya menambahkan kondisi jika x kurang dari 4 maka nilainya akan kembali x. Jika lainnya maka nilainya akan kembali tetapi menjadi x dibagi 4(dibulatkan kebawah) dikali 2 lalu ditambah x terakhir diambil sisa bagi dengan 4
- Untuk mencetak hasil dari fungsi soal4(x), print(soal4(x))

Soal 5

```
#latihan 2 soal 5
def ubah(angka):
    if angka % 2 == 0:
        return '*'
    else:
        return str(angka)

data = int(input("Masukkan angka: "))
lst=[]
for i in range(1,data+1):
    lst.append(i)
even_num = list(map(ubah, lst))
even_num = ' '.join(even_num)
print(even_num)

Masukkan angka: 10
1 * 3 * 5 * 7 * 9 *
```

Analisis: User diminta untuk input nilai data, nilai data tersebut dijadikan parameter untuk looping, looping tersebut bertujuan untuk menyimpan pada lst nilai indeks +1 dari nilai data. Setelah itu terdapat map dengan parameter fungsi ubah dan lst, untuk fungsi ubah bertujuan mengubah nilai lst, ketika nilai lst % 2 = 0 maka nilai tersebut akan diubah menjadi symbol '*' dan ketika nilainya bukan % 2 = 0 maka akan tercetak sesuai dari nilai lst tersebut. Karena hasil dari map masih berupa list, maka perlu menggunakan method join dengan nilai space untuk menampilkan space antar nilai dari lst.

Soal 6

```
#latihan 2 soal 6
def ubah(angka):
    if angka % 2 == 0:
        return '*'
    else:
        return str((angka+1)//2)

data = int(input("Masukkan angka: "))
lst=[]
for i in range(1,data+1):
    lst.append(i)
even_num = list(map(ubah, lst))
even_num = ' '.join(even_num)
print(even_num)

Masukkan angka: 10
1 * 2 * 3 * 4 * 5 *
```

Analisis: User diminta untuk input nilai data, nilai data tersebut dijadikan parameter untuk looping, looping tersebut bertujuan untuk menyimpan pada lst nilai indeks +1 dari nilai data. Setelah itu terdapat map dengan parameter fungsi ubah dan lst, untuk fungsi ubah bertujuan mengubah nilai lst, ketika nilai lst % 2 = 0 maka nilai tersebut akan diubah menjadi symbol '*'

dan ketika nilainya bukan $\%2=0$ maka akan tercetak nilai + 1 dan dibagi 2 $((1+1)/2=1)$. Karena hasil dari map masih berupa list, maka perlu menggunakan method join dengan nilai space untuk menampilkan space antar nilai dari lst.

Soal 8

```
#kasus 8
#input  = 1 2 3 4 5 6 7 8 9 10
#output = 1 1 3 3 15 15 105 105 945 945
def jaga(n):
    if n == 1:
        return 1
    elif n%2 == 1 :
        return jaga(n-2)*n
    else:
        return jaga(n-1)

num = [1,2,3,4,5,6,7,8,9,10]
hasil = list(map(jaga,num))
print(hasil)

[1, 1, 3, 3, 15, 15, 105, 105, 945, 945]
```

Analisis:

Pola Bilangan

Input = 1 3 5 7 9

Output = 1 3 15 105 945

$x(1) \rightarrow 1$
 $x(3) \rightarrow 3 \rightarrow x(1) \cdot 3$
 $x(5) \rightarrow 15 \rightarrow x(3) \cdot 5$
 $x(7) \rightarrow 105 \rightarrow x(5) \cdot 7$
 $x(9) \rightarrow 945 \rightarrow x(7) \cdot 9$

Bentuk umum = $x(n-2) \cdot n$

Pola bilangan tersebut adalah hasil dari bilangan sebelumnya dikali dengan nilai n, **return $x(n-2)*n$** tersebut sedangkan untuk input bilangan genapnya hasilnya sama seperti hasil bilangan sebelumnya, maka rumus yang benar untuk **return (n-1)**