

Modul 7 Praktikum Pemrograman Berbasis Fungsi

March 31, 2023

Tujuan Praktikum:

1. Mahasiswa dapat menerapkan fungsi Map, Reduce, dan Filter pada pemrosesan data

1 Map

Syntax:

```
[ ]: #map(func, iterable)
#iterable has to be a list, dict, tuple, set
#map normally happens with a predefined function but can be used with lambda as well

def squared(num, num2):
    if num >10 and num2 < 10:
        return num**2, num2**2

    else:
        return num, num2

numbers = [4, 11, 15, 23, 20, 10, 100]
more_nums = [2, 3, 4, 5, 7, 9]

squared_nums_map = list(map(squared, numbers, more_nums))
print(squared_nums_map)
```

[(4, 2), (121, 9), (225, 16), (529, 25), (400, 49), (10, 9)]

```
[ ]: #map(lambda x: do something, iterable)
#using lambda in map happens inline on one line
#list(map(lambda x: x * 2, nums))

squared_nums_lamb = list(map(lambda x, y: (pow(x, 2), y**2 ) if x > 10 and y < 10 else(x,y), numbers, more_nums))

print(squared_nums_lamb)
```

```
print(list(map(lambda x, y: (pow(x, 2), y**2 ) if x > 10 and y < 10 else(x,y),
↪numbers, more_nums)))
```

```
[(4, 2), (121, 9), (225, 16), (529, 25), (400, 49), (10, 9)]
```

```
[(4, 2), (121, 9), (225, 16), (529, 25), (400, 49), (10, 9)]
```

Gunakan fungsi map untuk menggandakan setiap angka dan mengurangnya dengan satu dalam list dengan menggunakan fungsi lambda

```
[ ]: print(list(map(lambda x, y: (x, y), numbers, more_nums)))
print(list(map(lambda x, y: (x *2 -1, y*2-1 ), numbers, more_nums)))
```

```
[(4, 2), (11, 3), (15, 4), (23, 5), (20, 7), (10, 9)]
```

```
[(7, 3), (21, 5), (29, 7), (45, 9), (39, 13), (19, 17)]
```

2 Filter

Filter mirip dengan fungsi map, di mana Anda dapat meneruskan argumen fungsi dan argumen list dan memfilter sesuatu dari list berdasarkan kondisi yang diteruskan. Mirip dengan fungsi map, yang mengembalikan objek filter, jadi Anda perlu mengetikkan convert to a list()

Syntax:

```
[ ]: names = ["April", "Tasha", "Fred", "Tony", "Sandra", "Jose", "Soloman",
↪"Ashliegh"]
```

```
def aNames(names):
    if names[0].lower() == "a":
        return True
    elif names[0].lower() == "t":
        return True
    else:
        return False

new_names_list = list(filter(aNames, names))
print(new_names_list)
print(aNames(names))
```

```
['April', 'Tasha', 'Tony', 'Ashliegh']
```

```
False
```

```
[ ]: new_names_lambda = list(filter(lambda name: True if name[0].lower() == "a" else
↪False, names))
print(new_names_lambda)
```

```
['April', 'Ashliegh']
```

Filter semua angka yang berada di bawah nilai rata-rata dalam list

```
[ ]: numbers = [23, 45, 15, 10, 9, 27, 5, 13, 2, 16, 17, 25, 49]

from statistics import mean

print(f"This is the mean value: {mean(numbers):.2f}")

mean_value = mean(numbers)
mean_numbers = list(filter(lambda numbers: True if numbers > mean_value else
↪False, numbers))
print(f"These are the values above the mean: {mean_numbers}")
```

This is the mean value: 19.69

These are the values above the mean: [23, 45, 27, 25, 49]

3 Reduce()

Syntax:

```
[ ]: from functools import reduce

#reduce(func, iterable)

list_1 = [1,2,3,5,9,10]

def addNums(num1, num2):
    return num1 + num2

result_add = reduce(addNums, list_1)
print(result_add)

#subtract a list of numbers

def subNums(num1, num2):
    return num1 - num2

result_sub = reduce(subNums, list_1)
print(result_sub)
```

30

-28

```
[ ]: result_lam = reduce(lambda x, y: x + y, list_1)
print(result_lam)
```

30

Gunakan fungsi reduce untuk mengalikan angka dalam daftar di bawah ini bersama dengan fungsi lambda.

```
[ ]: my_list = [1,3,5,7,9]

mult_lam = reduce(lambda x, y: x * y, my_list)
print(mult_lam)
```

945

4 Contoh penggunaan Map, Reduce, dan Filter

1. Tulis program yang mana fungsi map() dapat membuat list yang berisi pangkat dari isi list tersebut.[1,2,3,4,5,6,7,8,9,10].

```
[ ]: from functools import reduce

lst = [1,2,3,4,5,6,7,8,9,10]
print(list(map(lambda x: x**2, lst)))
```

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

2. Tulislah program yang mana fungsi map() and filter() dapat membuat list yang berisikan pangkat dari list tersebut yang bernilai genap [1,2,3,4,5,6,7,8,9,10].

```
[ ]: evenFiltered = list(filter(lambda x: x % 2 == 0, lst))
print(list(map(lambda x: x**2, evenFiltered)))
```

[4, 16, 36, 64, 100]

3. Tulislah program yang mana fungsi filter() dapat membuat list yang berisikan nilai genap antara angka 1 sampai 20.

```
[ ]: evenFilteredBelowTwenty = list(filter(lambda x: x % 2 == 0, range(1,21)))
print(evenFilteredBelowTwenty)
```

[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

4. Tulislah program yang mana fungsi map() dapat membuat list yang berisikan nilai antara angka 1 sampai 20.

```
[ ]: print(list(map(lambda x: x**2, range(1,21))))
```

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400]

5. Tentukan nilai maksimum dari sebuah list dari nilai numerik dengan menggunakan fungsi reduce.

```
[ ]: print(reduce(lambda largest, current: largest if (largest > current) else
    ↪current, range(1,21)))
```

20

6. Hitunglah jumlah angka dari 1 sampai 100, menggunakan reduce.

```
[ ]: print(reduce(lambda previous, current: previous + current, range(1,101)))
```

5050

7. Ubahlah contoh sebelumnya untuk menghitung faktorial dari 1 hingga nilai berapapun, tetapi tidak menggunakan angka 50.

```
[ ]: print(reduce(lambda previous, current: previous * current, range(1,21)))
```

2432902008176640000

8. Tulislah program python untuk menyaring nilai positif dari sebuah list.

```
[ ]: print(list(filter(lambda x: x > 0, range(-10, 10))))
```

[1, 2, 3, 4, 5, 6, 7, 8, 9]

9. Carilah nilai intersection dari dua list, gunakan fungsi filter.

```
[ ]: lst2 = [5,6,7,8,9,10,11,12,13,14,15]
print(list(filter(lambda x: x in lst2, lst)))
print([x for x in lst if x in lst2])
```

[5, 6, 7, 8, 9, 10]

[5, 6, 7, 8, 9, 10]