
KELOMPOK
ECHO

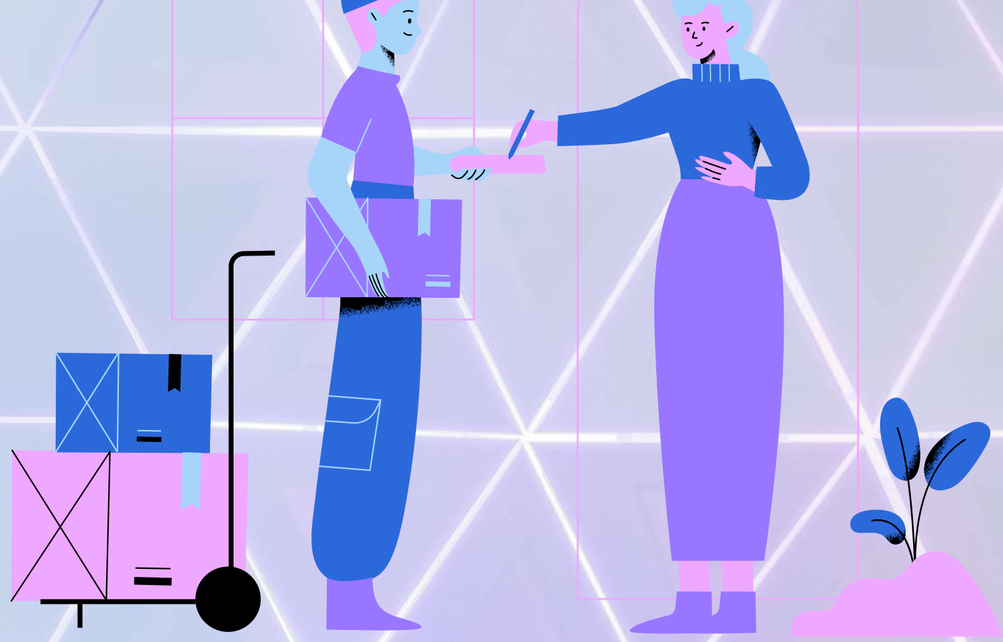
Lambda Function and Comprehensions

Tugas 3

Susun presentasi terkait penerapan *dict comprehension* dan *set comprehension* pada pemrograman python. Isi presentasi perlu mencakup (minimal) beberapa sub topik berikut:

- Overview
- Contoh penggunaan
- Kelebihan dan kekurangan

dict comprehension



OVERVIEW

--List comprehension bisa digunakan untuk mengisi elemen dari sebuah list secara cepat

-Format penulisannya adalah: dictname = {key: value(body of function) for variable in itterable}

```
{ key: value for vars in iterable }
  |    |       |          |
  |    |       |          |
{ num: num*num for num in range(1, 11) }
```

-Iterable bisa berupa sequence, list, string, tuple.

KELEBIHAN

-Baris kode lebih singkat dan padat jika dibandingkan dengan manual

-Memperpendek proses inisialisasi dictionary

KEKURANGAN

-Terkadang, dapat membuat kode berjalan lebih lambat

-Mengkonsumsi lebih banyak memori

-Lebih sulit untuk dibaca atau diterjemahkan

contoh penerapan

dict 1

```
#CONTOH 1: Dictionary Comprehension (Sequence)
#Manual
bil_pangkat = {}
for bil in range (1,11):
  bil_pangkat[bil] = bil**2
print(bil_pangkat)

#Comprehension
bil_pangkat2 = {bil: bil**2 for bil in range (1,11)}
print(bil_pangkat2)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100}
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100}
```

dict 2

```
#CONTOH 2: Dictionary Comprehension (List)
keys = [1,2,3,4,5,6,7,8,9,10]
values = ["Rike", "Ilmi", "Arifah", "Puput", "Salsabila", "Indah", "Hanandia", "Zen", "Dicky", "Alde"]
#Manual
myDict = dict(zip(keys, values))
print(myDict)

#Comprehension
myDict = { k:v for (k,v) in zip(keys, values)}
print(myDict)
```

```
{1: 'Rike', 2: 'Ilmi', 3: 'Arifah', 4: 'Puput', 5: 'Salsabila', 6: 'Indah', 7: 'Hanandia', 8: 'Zen', 9: 'Dicky', 10: 'Alde'}
{1: 'Rike', 2: 'Ilmi', 3: 'Arifah', 4: 'Puput', 5: 'Salsabila', 6: 'Indah', 7: 'Hanandia', 8: 'Zen', 9: 'Dicky', 10: 'Alde'}
```

contoh penerapan

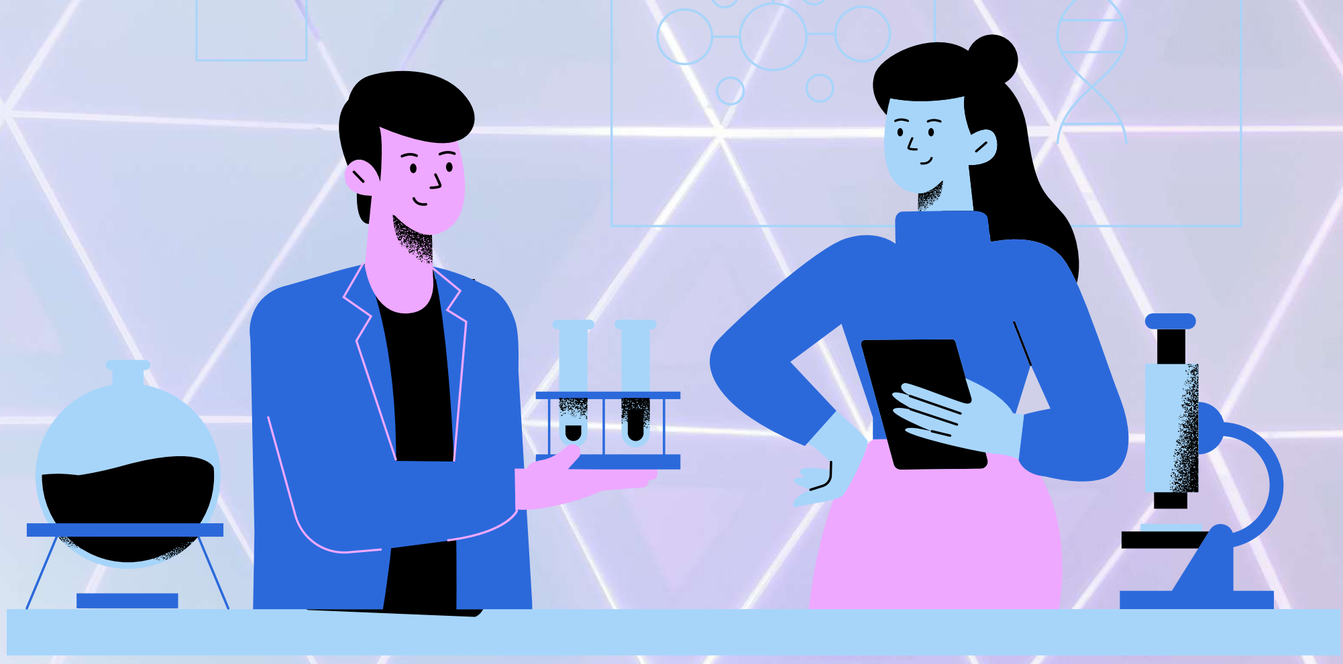
dict 3

```
#CONTOH 3: Dictionary Comprehension (Tuple)
myDict={1:"Rike", 2:"Ilmi", 3:"Arifah", 4:"Puput", 5:"Salsabila", 6:"Indah", 7:"Hanandia", 8:"Zen", 9:"Dicky", 10:"Alde" }
cari= (1, 2, 4, 10)
data = {}
bukan = {}
#Manual
for k,v in myDict.items():
  if k in cari :
    data[k] = v
  else:
    bukan[k] = v
print(data)
print(bukan)

#Comprehension
data={k:v for(k,v) in myDict.items() if k in cari}
print(data)
print(bukan)
```

```
{1: 'Rike', 2: 'Ilmi', 4: 'Puput', 10: 'Alde'}
{3: 'Arifah', 5: 'Salsabila', 6: 'Indah', 7: 'Hanandia', 8: 'Zen', 9: 'Dicky'}
{1: 'Rike', 2: 'Ilmi', 4: 'Puput', 10: 'Alde'}
{3: 'Arifah', 5: 'Salsabila', 6: 'Indah', 7: 'Hanandia', 8: 'Zen', 9: 'Dicky'}
```

set comprehension



OVERVIEW

- Set comprehension bisa digunakan untuk mengisi elemen dari sebuah set secara cepat, tanpa menuliskannya secara manual satu persatu.
- Format penulisan: Set_name = { expression for element in iterable if condition }
- Iterable bisa berupa sequence, list, set, maupun tuple.
- Expression: statement yang digunakan untuk memberikan nilai pada tiap elemen dari set baru yang dibuat.

KELEBIHAN

1. Membutuhkan waktu komputasi yang lebih cepat dibandingkan dengan manual loop.
2. Menggunakan baris kode yang lebih padat dibandingkan dengan cara manual.

KEKURANGAN

1. Sulit untuk dibaca / dipahami apabila logic yang digunakan terlalu kompleks.
2. Tidak dapat mengakses elemen set satu persatu karena set tidak memiliki index, sehingga outputnya pun akan ditampilkan secara acak.

contoh penerapan

set 1

```
# Membuat set baru
manual_set = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}
print(manual_set)

automatic_set = set(x for x in range (1,21))
print(automatic_set)
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}
```

set 2

```
# Membuat set dengan mengubah tipe data dari elemen tuple
mytuple = (3, 2.6, '4', 8, 0.14)
myset = {str(number) for number in mytuple }
print(myset)
```

```
{'0.14', '3', '4', '2.6', '8'}
```

contoh penerapan

set 4

```
# Memodifikasi elemen set
pets = {'Cat', 'rabBIt', 'doG', 'hAmster'}

your_pets = set()
for x in pets:
    your_pets.add(x.lower())
print(your_pets)

my_pets = {x.lower() for x in pets}
print(my_pets)
```

```
{'hamster', 'rabbit', 'cat', 'dog'}
{'hamster', 'rabbit', 'cat', 'dog'}
```

set 3

```
# Membuat set baru dari element list
Set = ['5', '@', 'rn', 8, 'rn', True]

urnewSet = set()
for element in Set:
   urnewSet.add(element*2)
print(urnewSet)

mynewSet = {element*2 for element in Set}
print(mynewSet)
```

```
{2, '@@', 16, 'rnrn', '55'}
{2, '@@', 16, 'rnrn', '55'}
```

contoh penerapan

set 5

```
#Conditional Statement dalam Set Comprehension
words1 = {"now", "all", "I", "see", "is", "colour", "like", "a", "rainbow", "in", "the", "sky"}

cute_words_ = set()
for word in words1:
  if len(word) <= 3:
    cute_words_.add(word)
print(cute_words_)

cute_words = {word for word in words1 if len(word) <= 3}
print(cute_words)
```

```
{'in', 'all', 'I', 'now', 'is', 'a', 'see', 'the', 'sky'}
{'in', 'all', 'I', 'now', 'is', 'a', 'see', 'the', 'sky'}
```

set 6

```
words2 = ("and", "like", "red", "and", "yellow", "and", "pink", "and", "green")

biggest_words_ = set()
for word in words2:
  if len(word) > 3:
    biggest_words_.add(word.capitalize())
  else:
    biggest_words_.add(word)
print(biggest_words_)

biggest_words = {word.capitalize() if len(word) > 3 else word for word in words2}
print(biggest_words)
```

```
{'and', 'red', 'Green', 'Like', 'Yellow', 'Pink'}
{'and', 'red', 'Green', 'Like', 'Yellow', 'Pink'}
```

contoh penerapan

set 7

```
words3 = ['you', 'make', 'everything', 'change', 'you', 'rearrange', 'so', "don't", 'ever', 'fade']

unique_words_ = set()
for word in words3:
  if len(word) >= 4:
    if word[-1] ==  'e':
      unique_words_.add(word.upper())
    else:
      unique_words_.add(word)
print(unique_words_)

unique_words = {word.upper() if word[-1] == 'e' else word for word in words3 if len(word) >= 4}
print(unique_words)
```

```
{'CHANGE', 'FADE', 'MAKE', "don't", 'ever', 'everything', 'REARRANGE'}
{'CHANGE', 'FADE', 'MAKE', "don't", 'ever', 'everything', 'REARRANGE'}
```

Member

Hanadia Navaz Kamalia
(3322600007)

Farrel Manazili Nabhan
(3322600011)

Ardiansyah Indra Febrianto
(3322600014)

Erina Nur Miftaqul Jannah
(3322600018)

Bayu Kurniawan
(3322600019)

THANK YOU