



DIGITAL
TALENT
SCHOLARSHIP

DIGITAL TALENT SCHOLARSHIP 2019





DIGITAL
TALENT
SCHOLARSHIP

Sesi 9

Pengenalan

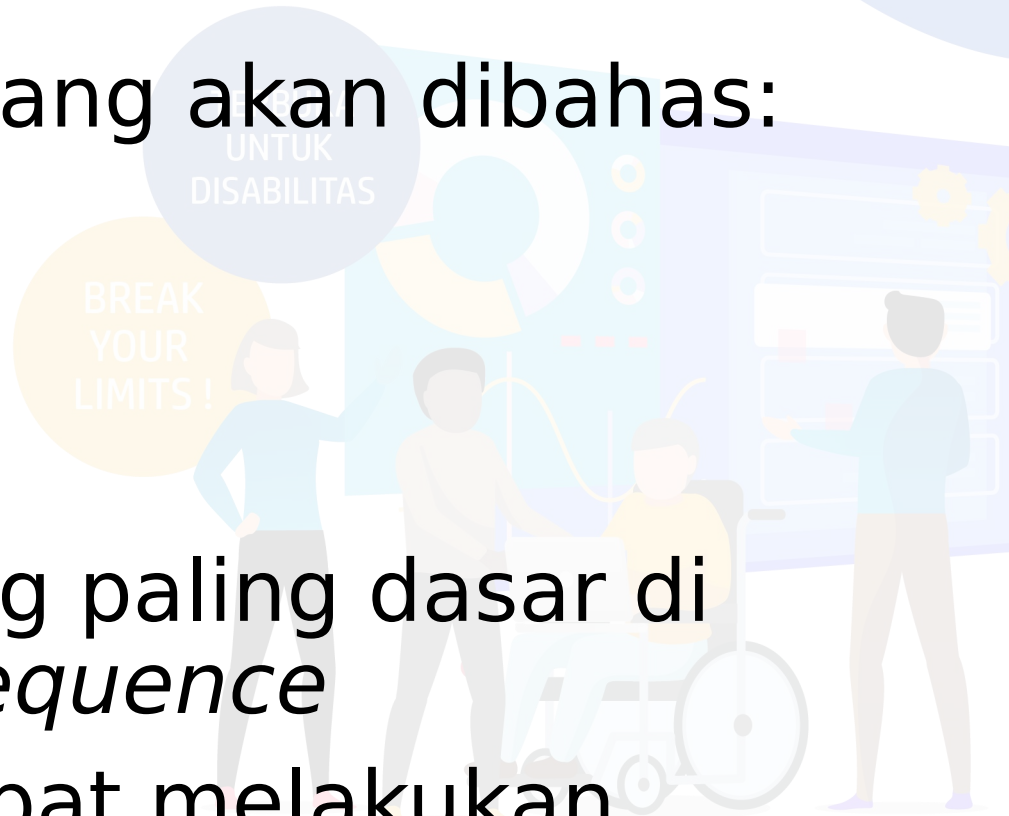
Python (9)

Big Data Analytics



Struktur Data

- 4 struktur data yang akan dibahas:
 - *List*
 - *Tuple*
 - *Dictionary*
 - *Set*
- Struktur data yang paling dasar di Python adalah *sequence*
- Struktur data dapat melakukan penyimpanan, pengurutan, pengelompokan dan menampilkan suatu data



Struktur Data - *List*

- List merupakan struktur data pada Python yang mampu menyimpan lebih dari satu data; seperti *array*
- Dimulai dengan index 0; ditulis menggunakan *square brackets* []

```
list1 = ['physics', 'chemistry', 1997, 2000];  
list2 = [1, 2, 3, 4, 5 ];  
list3 = ["a", "b", "c", "d"]
```

List - Accessing Values

- Untuk mengakses nilai di *list*, gunakan kurung siku [] bersamaan dengan indeks

```
list1 = ['physics', 'chemistry', 1997, 2000];  
list2 = [1, 2, 3, 4, 5, 6, 7];  
print "list1[0]: ", list1[0]  
print "list2[1:5]: ", list2[1:5]
```

Output

```
list1[0]: physics  
list2[1:5]: [2, 3, 4, 5]
```

List - *Updating List*

Output

```
list = ['physics', 'chemistry', 1997, 2000];  
print "Value available at index 2 : "  
print list[2]  
list[2] = 2001;  
print "New value available at index 2 : "  
print list[2]
```

```
Value available at index 2 :  
1997  
New value available at index 2 :  
2001
```

List - *Delete List Element*

Output

```
list1 = ['physics', 'chemistry', 1997, 2000];  
print list1  
del list1[2];  
print "After deleting value at index 2 : "  
print list1
```

```
['physics', 'chemistry', 1997, 2000]  
After deleting value at index 2 :  
['physics', 'chemistry', 2000]
```


List - Adding List

1. `prepend(item)`
menambahkan
item dari depan

```
#list mula-mula  
buah = ["jeruk", "apel", "mangga", "duren"]  
buah.prepend("anggur")  
["anggur", "jeruk", "apel", "mangga", "duren"]
```

2. `append(item)`
menambahkan
item dari
belakang

```
#list mula-mula  
buah = ["jeruk", "apel", "mangga", "duren"]  
# Tambahkan manggis  
buah.append("manggis")  
["jeruk", "apel", "mangga", "duren", "manggis"]
```

3. `insert(index, item)`
menambahkan
item dari indeks
tertentu

```
#list mula-mula  
buah = ["jeruk", "apel", "mangga", "duren"]  
buah.insert(2, "duren")
```



DIGITAL
TALENT
SCHOLARSHIP

List - *Basic Operations*

- + Penggabungan
- * Pengulangan

TERBUKA
UNTUK
DISABILITAS

BREAK

Python Expression	Results	Description
<code>len([1, 2, 3])</code>	3	Length
<code>[1, 2, 3] + [4, 5, 6]</code>	<code>[1, 2, 3, 4, 5, 6]</code>	Concatenation
<code>['Hi!'] * 4</code>	<code>['Hi!', 'Hi!', 'Hi!', 'Hi!']</code>	Repetition
<code>3 in [1, 2, 3]</code>	True	Membership
<code>for x in [1, 2, 3]: print x,</code>	1 2 3	Iteration

List - Multidimensi

```
# List minuman dengan 2 dimensi
list_minuman = [
    ["Kopi", "Susu", "Teh"],
    ["Jus Apel", "Jus Melon", "Jus Jeruk"],
    ["Es Kopi", "Es Campur", "Es Teler"]
]

# Cara mengakses list multidimensi
# misalkan kita ingin mengambil "es kopi"
print list_minuman[2][0]
```

Output "Es Kopi"

Tabel Minuman

"Kopi"	"Susu"	"Teh"
"Jus Apel"	"Jus Melon"	"Jus Jeruk"
"Es Kopi"	"Es Campur"	"Es Teler"

- *List* multidimensi biasanya digunakan untuk menyimpan struktur data yang kompleks seperti tabel, matriks, *graph*, *tree*, dsb.

List h Progra m

```

1  # Fig. 5.3: fig05_03.py
2  # Creating, accessing and changing a list.
3
4  aList = []      # create empty list
5
6  # add values to list
7  for number in range( 1, 11 ):
8      aList += [ number ]
9
10 print "The value of aList is:", aList
11
12 # access list values by iteration
13 print "\nAccessing values by iteration:"
14
15 for item in aList:
16     print item,
17
18 print
19
20 # access list values by index
21 print "\nAccessing values by index:"
22 print "Subscript    Value"
23
24 for i in range( len( aList ) ):
25     print "%9d %7d" % ( i, aList[ i ] )
26
27 # modify list
28 print "\nModifying a list value..."
29 print "Value of aList before modification:", aList
30 aList[ 0 ] = -100
31 aList[ -3 ] = 19
32 print "Value of aList after modification:", aList

```



DIGITAL
TALENT
SCHOLARSHIP

List - Contoh Program

OUTPUT

The value of aList is: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Accessing values by iteration:

1 2 3 4 5 6 7 8 9 10

Accessing values by index:

Subscript	Value
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10

Modifying a list value...

Value of aList before modification: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Value of aList after modification: [-100, 2, 3, 4, 5, 6, 7, 19, 9, 10]

TERBUKA



DIGITAL
TALENT
SCHOLARSHIP

List - Built-in List Functions & Methods

Sr.No.	Function with Description
1	cmp(list1, list2) Compares elements of both lists.
2	len(list) Gives the total length of the list.
3	max(list) Returns item from the list with max value.
4	min(list) Returns item from the list with min value.
5	list(seq) Converts a tuple into list.

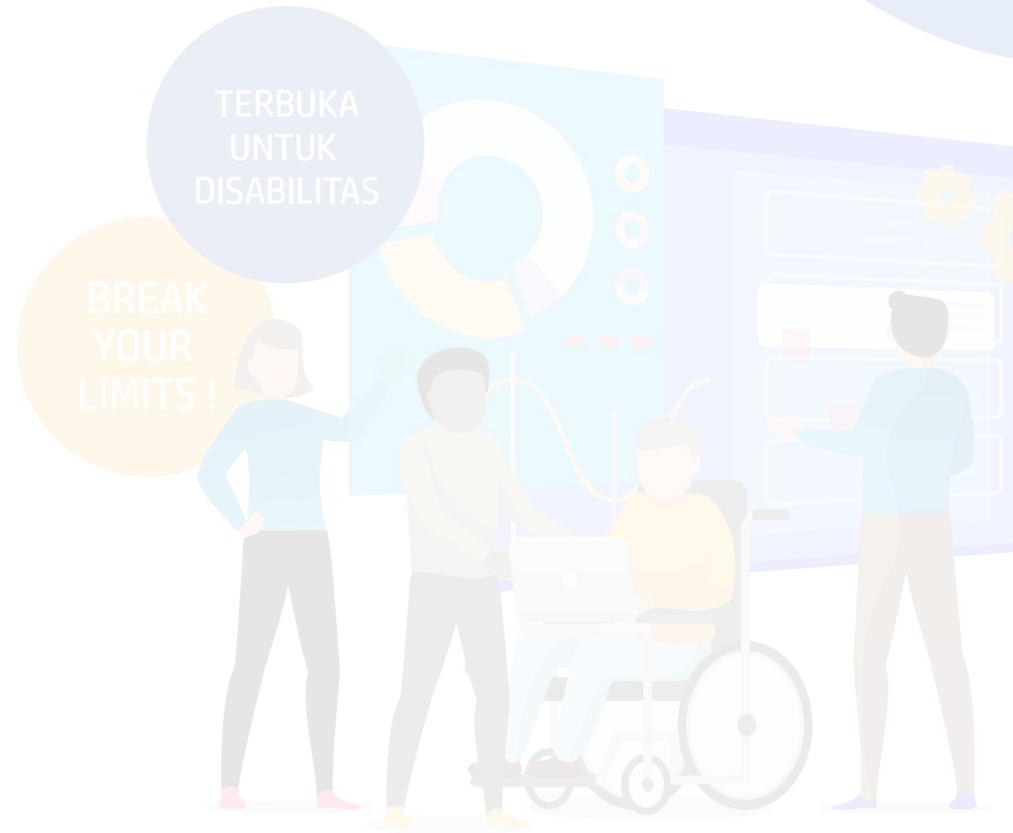
Sr.No.	Methods with Description
1	list.append(obj) Appends object obj to list
2	list.count(obj) Returns count of how many times obj occurs in list
3	list.extend(seq) Appends the contents of seq to list
4	list.index(obj) Returns the lowest index in list that obj appears
5	list.insert(index, obj) Inserts object obj into list at offset index
6	list.pop(obj=list[-1]) Removes and returns last object or obj from list
7	list.remove(obj) Removes object obj from list
8	list.reverse() Reverses objects of list in place
9	list.sort([func]) Sorts objects of list, use compare func if given

REAK
OUR
MITS



DIGITAL
TALENT
SCHOLARSHIP

Lab Session - List



Lab

- Buatlah program untuk mencetak histogram dari nilai sebuah list, se

```
Enter 10 integers:
Enter integer 1: 19
Enter integer 2: 3
Enter integer 3: 15
Enter integer 4: 7
Enter integer 5: 11
Enter integer 6: 9
Enter integer 7: 13
Enter integer 8: 5
Enter integer 9: 17
Enter integer 10: 1

Creating a histogram from values:
Element      Value  Histogram
      0         19  *****
      1          3   ***
      2         15  *****
      3          7   *****
      4         11  *****
      5          9   *****
      6         13  *****
      7          5   *****
      8         17  *****
      9          1   *
```


Struktur Data - *Tuples*

- *Tuple* hampir sama dengan *list*
- Perbedaannya adalah *tuple* tidak dapat diubah bahkan dihapus
- Dibuat menggunakan tanda kurung ()

```
tup1 = ('physics', 'chemistry', 1997, 2000);  
tup2 = (1, 2, 3, 4, 5 );  
tup3 = "a", "b", "c", "d";
```

- *Empty tuple* dapat ditulis dengan 2 tanda kurung yang tidak mengandung apapun
Tup1 = ()
- Untuk mengisi *tuple* dengan nilai tunggal harus memasukan dengan menggunakan koma *Tup1* = (50,);

Tuples - Accessing Values

- Seperti halnya list, kita bisa mengakses anggota tuple lewat indeksnya menggunakan format `namatuple[indeks]`
- Indeks dimulai dari 0 untuk anggota pertama. Indeks negatif juga bisa dipakai mulai dari -1 untuk anggota terakhir *tuple*

```
▶ #!/usr/bin/python  
  
tup1 = ('physics', 'chemistry', 1997, 2000);  
tup2 = (1, 2, 3, 4, 5, 6, 7 );  
print ("tup1[0]: ", tup1[0])  
print ("tup2[1:5]: ", tup2[1:5])
```

```
tup1[0]: physics  
tup2[1:5]: (2, 3, 4, 5)
```

Tuples - Accessing Values

```
my_tuple = ('p','y','t','h','o','n')
# Output: 'p'
print(my_tuple[0])

# Output: 'y'
print(my_tuple[1])

# Output: 'n'
print(my_tuple[-1])

# Output: 'o'
print(my_tuple[-2])

# IndexError
print(my_tuple[6])
```

p
y
n
o

```
IndexError                                Traceback (most recent call last)
<ipython-input-12-93a6bca08ce4> in <module>
    13
    14 # IndexError
--> 15 print(my_tuple[6])
```

IndexError: tuple index out of range



Tuples - *Updating Values*

- *Tuples* tidak dapat diubah yang artinya kita tidak dapat memperbarui atau mengubah nilai elemen *tuple*
- Kita dapat mengambil porsi *tuple* yang ada untuk membuat tupel baru seperti yang diperlihatkan contoh berikut

```
#!/usr/bin/python
tup1 = (12, 34.56),
tup2 = ('abc', 'xyz');

# Following action is not valid for tuples
# tup1[0] = 100;

# So let's create a new tuple as follows
tup3 = tup1 + tup2;
print (tup3)
```

```
(12, 34.56, 'abc', 'xyz')
```

Tuples - *Updating Values*

```

M my_tuple = (2, 3, 4, [5, 6])
# kita tidak bisa mengubah anggota tuple
# bila kita hilangkan tanda komentar # pada baris ke 6
# akan muncul error: # TypeError: 'tuple' object does not support item assignment

# my_tuple[1] = 8

# tapi list di dalam tuple bisa diubah
# output: (2, 3, 4, [7, 6])
my_tuple[3][0] = 7
print(my_tuple)

# tuple bisa diganti secara keseluruhan dengan penugasan kembali
# output: ('p', 'y', 't', 'h', 'o', 'n')
my_tuple = ('p', 'y', 't', 'h', 'o', 'n')
print(my_tuple)

# anggota tuple juga tidak bisa dihapus menggunakan del
# perintah berikut akan menghasilkan error TypeError
# kalau Anda menghilangkan tanda komentar #

#del my_tuple[0]

# kita bisa menghapus tuple keseluruhan

del my_tuple

```

```

(2, 3, 4, [7, 6])
('p', 'y', 't', 'h', 'o', 'n')

```

- Bila anggota *tuple*-nya adalah *tuple* bersarang dengan anggota seperti *list*, maka item pada *list* tersebut dapat diubah. Jelasnya ada pada contoh berikut:

Tuples - *Delete Element*

- Menghapus elemen *tuple* secara individual tidak dimungkinkan
- Kita dapat dengan menyusun *tuple* lain dengan elemen yang tidak diinginkan dibuang
- Untuk menghapus seluruh *tuple* secara eksplisit, cukup (

```
#!/usr/bin/python
```

```
tup = ('physics', 'chemistry', 1997, 2000);  
print (tup);  
del tup;  
print ("After deleting tup : ");  
print (tup);
```

```
('physics', 'chemistry', 1997, 2000)  
After deleting tup :
```

NameError

Traceback (most recent call last)

```
<ipython-input-11-26e9ad50a7bc> in <module>  
      5 del tup;  
      6 print ("After deleting tup : ");  
----> 7 print (tup);
```

NameError: name 'tup' is not defined

Menguji Keanggotaan *Tuple*

- Seperti halnya *string* dan *list*, kita bisa menguji apakah sebuah objek adalah anggota dari *tuple* atau tidak, yaitu dengan menggunakan operator *in* atau *out* untuk kebalik

```
my_tuple = (1, 2, 3, 'a', 'b', 'c')
```

```
# menggunakan in
# output: True
print('3' in my_tuple)

# output: False
print('e' in my_tuple)

# menggunakan not in
# output True
print('k' not in my_tuple)
```

```
False
False
True
```

Metode dan Fungsi Bawaan *Tuple*

- Metode *count(x)* berfungsi mengembalikan jumlah item yang sesuai dengan x pada tuple
- Metode *index(x)* berfungsi mengembalikan indeks dari item pertama yang sama dengan

X ▶ `my_tuple = ('p','y','t','o','n','i','n','d','o')`

`# count`

`# output: 2`

`print(my_tuple.count('n'))`

`# index`

`# Output 4`

`print(my_tuple.index('n'))`

2

4



DIGITAL
TALENT
SCHOLARSHIP

Metode dan Fungsi Bawaan Tuple

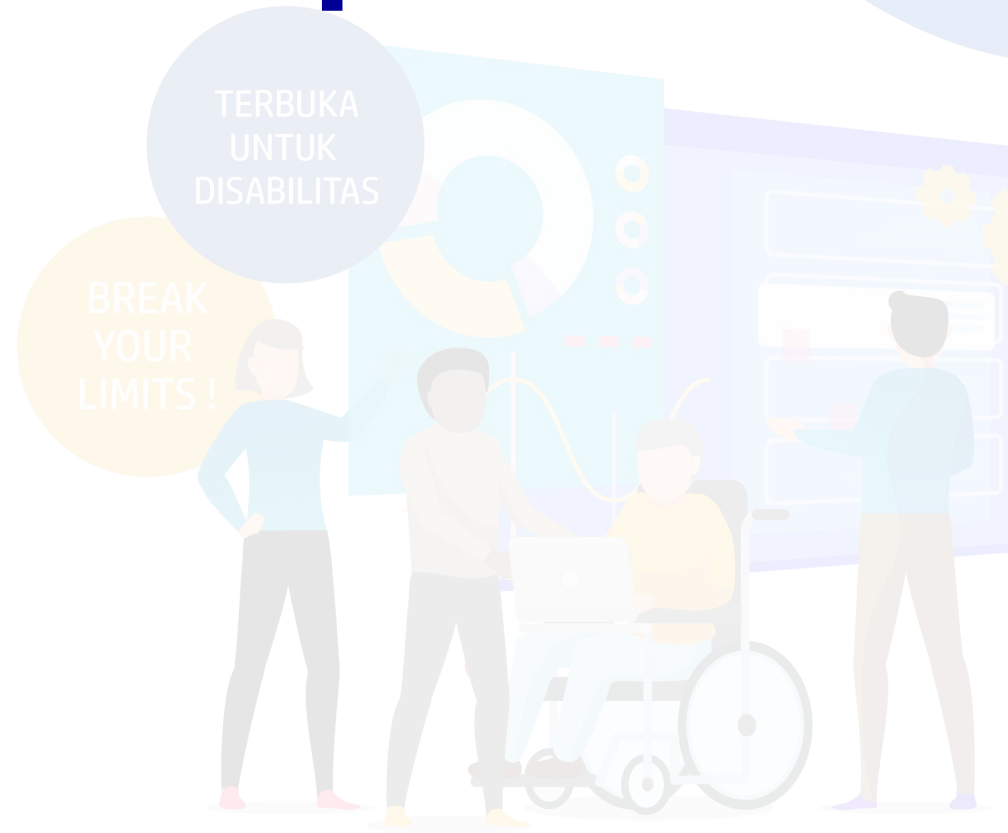
Fungsi	Deskripsi
all()	Mengembalikan True jika semua anggota tuple adalah benar (tidak ada yang kosong)
any()	Mengembalikan True jika salah satu atau semua bernilai benar. Jika tuple kosong, maka akan mengembalikan False.
enumerate()	Mengembalikan objek enumerasi. Objek enumerasi adalah objek yang terdiri dari pasangan indeks dan nilai.
len()	Mengembalikan panjang (jumlah anggota) tuple
max()	Mengembalikan anggota terbesar di tuple
min()	Mengembalikan anggota terkecil di tuple
sorted()	Mengambil anggota tuple dan mengembalikan list baru yang sudah diurutkan
sum()	Mengembalikan jumlah dari semua anggota tuple
tuple()	Mengubah sequence (list, string, set, dictionary) menjadi tuple





DIGITAL
TALENT
SCHOLARSHIP

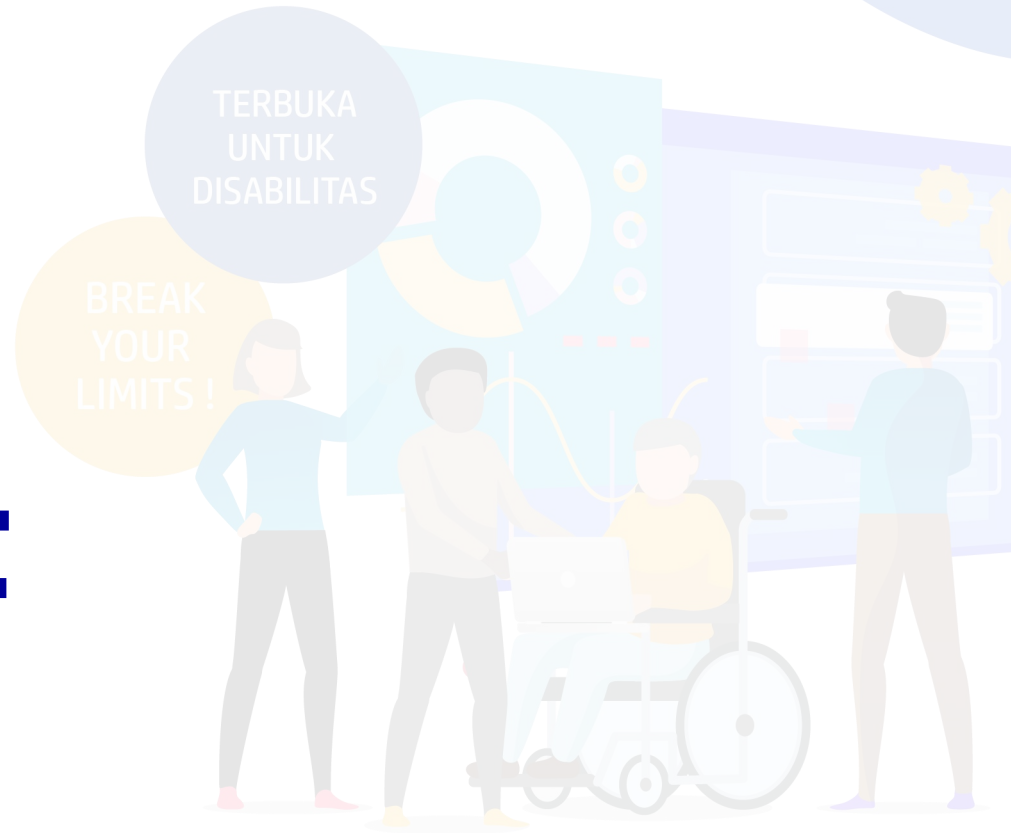
Lab Session - Tuple





DIGITAL
TALENT
SCHOLARSHIP

Istirahat



Struktur Data - *Dictionary*

- *Dictionary* adalah struktur data yang bentuknya seperti kamus. Ada kata **kunci** kemudian ada **nilainya**. Kata kunci harus unik, sedangkan nilai boleh diisi dengan apa saja
- Setiap *key* dipisahkan dengan colon (:), semuanya tertutup dengan kurung kurawal
- *Key* harus *unique* sementara *value* tidak harus. Nilai bisa terdiri dari berbagai tipe

TERBUKA
UNTUK
KEBERKESAMAAN

Dictionary - Accessing Values

```
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}  
print "dict['Name']: ", dict['Name']  
print "dict['Age']: ", dict['Age']
```

Output

```
dict['Name']:  Zara  
dict['Age']:   7
```

Akan terjadi *error* jika mengakses data *key* yang tidak terdapat di *dictionary*

```
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}  
print "dict['Alice']: ", dict['Alice']
```

Output

```
dict['Alice']:  
Traceback (most recent call last):  
  File "test.py", line 4, in <module>  
    print "dict['Alice']: ", dict['Alice'];  
KeyError: 'Alice'
```



Dictionary - *Updating List*

```
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}  
dict['Age'] = 8; # update existing entry  
dict['School'] = "DPS School"; # Add new entry  
  
print "dict['Age']: ", dict['Age']  
print "dict['School']: ", dict['School']
```

Output

```
dict['Age']: 8  
dict['School']: DPS School
```

Dictionary - *Delete List Element*

```
dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}  
del dict['Name']; # remove entry with key 'Name'  
dict.clear();     # remove all entries in dict  
del dict;         # delete entire dictionary  
  
print "dict['Age']: ", dict['Age']  
print "dict['School']: ", dict['School']
```





Output

```
dict['Age']:  
Traceback (most recent call last):  
  File "test.py", line 8, in <module>  
    print "dict['Age']: ", dict['Age'];  
TypeError: 'type' object is unsubscriptable
```



DIGITAL
TALENT
SCHOLARSHIP

Dictionary - Built-in List Functions & Methods

Sr.No.	Function with Description
1	cmp(dict1, dict2)  Compares elements of both dict.
2	len(dict)  Gives the total length of the dictionary. This would be equal to the number of items in the dictionary.
3	str(dict)  Produces a printable string representation of a dictionary
4	type(variable)  Returns the type of the passed variable. If passed variable is dictionary, then it would return a dictionary type.





DIGITAL
TALENT
SCHOLARSHIP

Dictionary - Built-in List Functions & Methods

Sr.No.	Methods with Description
1	<code>dict.clear()</code> Removes all elements of dictionary <i>dict</i>
2	<code>dict.copy()</code> Returns a shallow copy of dictionary <i>dict</i>
3	<code>dict.fromkeys()</code> Create a new dictionary with keys from <i>seq</i> and values set to <i>value</i> .
4	<code>dict.get(key, default=None)</code> For <i>key</i> <i>key</i> , returns value or default if <i>key</i> not in dictionary
5	<code>dict.has_key(key)</code> Returns <i>true</i> if <i>key</i> in dictionary <i>dict</i> , <i>false</i> otherwise
6	<code>dict.items()</code> Returns a list of <i>dict</i> 's (key, value) tuple pairs
7	<code>dict.keys()</code> Returns list of dictionary <i>dict</i> 's keys
8	<code>dict.setdefault(key, default=None)</code> Similar to <i>get()</i> , but will set <i>dict[key]=default</i> if <i>key</i> is not already in <i>dict</i>
9	<code>dict.update(dict2)</code> Adds dictionary <i>dict2</i> 's key-values pairs to <i>dict</i>
10	<code>dict.values()</code> Returns list of dictionary <i>dict</i> 's values





DIGITAL
TALENT
SCHOLARSHIP

Dictionary - Contoh Program

```
1  # Fig. 5.09: fig05_09.py
2  # Creating, accessing and modifying a dictionary.
3
4  # create and print an empty dictionary
5  emptyDictionary = {}
6  print "The value of emptyDictionary is:", emptyDictionary
7
8  # create and print a dictionary with initial values
9  grades = { "John": 87, "Steve": 76, "Laura": 92, "Edwin": 89 }
10 print "\nAll grades:", grades
11
12 # access and modify an existing dictionary
13 print "\nSteve's current grade:", grades[ "Steve" ]
14 grades[ "Steve" ] = 90
15 print "Steve's new grade:", grades[ "Steve" ]
16
17 # add to an existing dictionary
18 grades[ "Michael" ] = 93
19 print "\nDictionary grades after modification:"
20 print grades
21
22 # delete entry from dictionary
23 del grades[ "John" ]
24 print "\nDictionary grades after deletion:"
25 print grades
```





DIGITAL
TALENT
SCHOLARSHIP

Dictionary - Contoh Program Output

The value of emptyDictionary is: {}

All grades: {'Edwin': 89, 'John': 87, 'Steve': 76, 'Laura': 92}

Steve's current grade: 76

Steve's new grade: 90

Dictionary grades after modification:

{ 'Edwin': 89, 'Michael': 93, 'John': 87, 'Steve': 90, 'Laura': 92 }

Dictionary grades after deletion:

{ 'Edwin': 89, 'Michael': 93, 'Steve': 90, 'Laura': 92 }

TERBUKA
UNTUK



Dictionary - Contoh Program

```

1  # Fig. 5.11: fig05_11.py
2  # Student poll program.
3
4  responses = [ 1, 2, 6, 4, 8, 5, 9, 7, 8, 10,
5                1, 6, 3, 8, 6, 10, 3, 8, 2, 7,
6                6, 5, 7, 6, 8, 6, 7, 5, 6, 6,
7                5, 6, 7, 5, 6, 4, 8, 6, 8, 10 ]
8
9  print "Rating      Frequency"
10
11  for i in range( 1, 11 ):
12      print "%6d %13d" % ( i, responses.count( i ) )

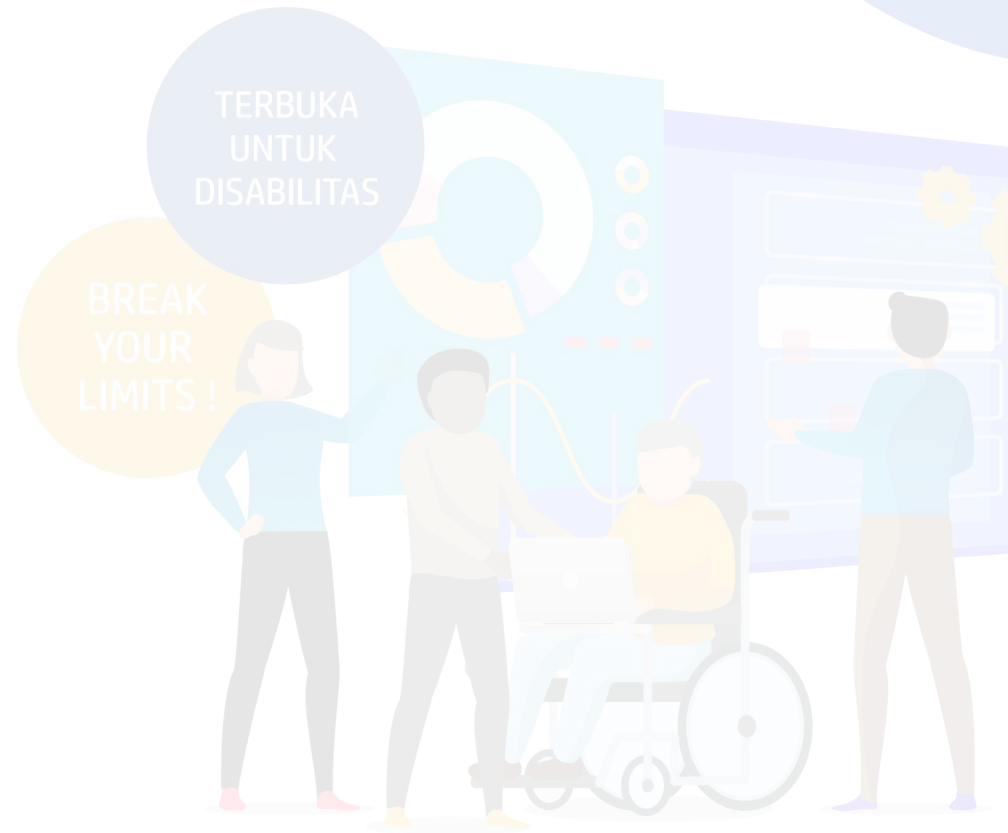
```

Rating	Frequency
1	2
2	2
3	2
4	2
5	5
6	11
7	5
8	7
9	1
10	3



DIGITAL
TALENT
SCHOLARSHIP

Lab Session - Dictionaries



Struktur Data - Set

- Bersifat *unique*. Apabila menginput sebuah nilai yang sama pada **set**, maka salah satu akan di-*replace*
- **Unordered**
- **Unindexed** - tidak ada indeks yang dilampirkan
- **Immutable** - Elemen dalam set tidak seluruhan

```
Days=set(["Mon","Tue","Wed","Thu","Fri","Sat","Sun"])
Months={"Jan","Feb","Mar"}
Dates={21,22,17}
print(Days)
print(Months)
print(Dates)
```

Output

```
set(['Wed', 'Sun', 'Fri', 'Tue', 'Mon', 'Thu', 'Sat'])
set(['Jan', 'Mar', 'Feb'])
set([17, 21, 22])
```

Set - Accessing Values

Output

Tidak bisa mengakses *individual value* suatu set. Hanya bisa mengakses keseluruhan elemen. Mendapatkan *individual element* dengan menggunakan *looping*

```
Days=set(["Mon","Tue","Wed","Thu","Fri","Sat","Sun"])  
  
for d in Days:  
    print(d)
```

Wed
Sun
Fri
Tue
Mon
Thu
Sat

Set - Adding

```
Days=set(["Mon","Tue","Wed","Thu","Fri","Sat"])
Days.add("Sun")
print(Days)
```

Output

```
set(['Wed', 'Sun', 'Fri', 'Tue', 'Mon', 'Thu', 'Sat'])
```

TERBUKA
UNTUK
DISABILITAS

BREAK
LIMITS!

Set - Removing

Ada beberapa cara yang digunakan untuk menghapus data di set:

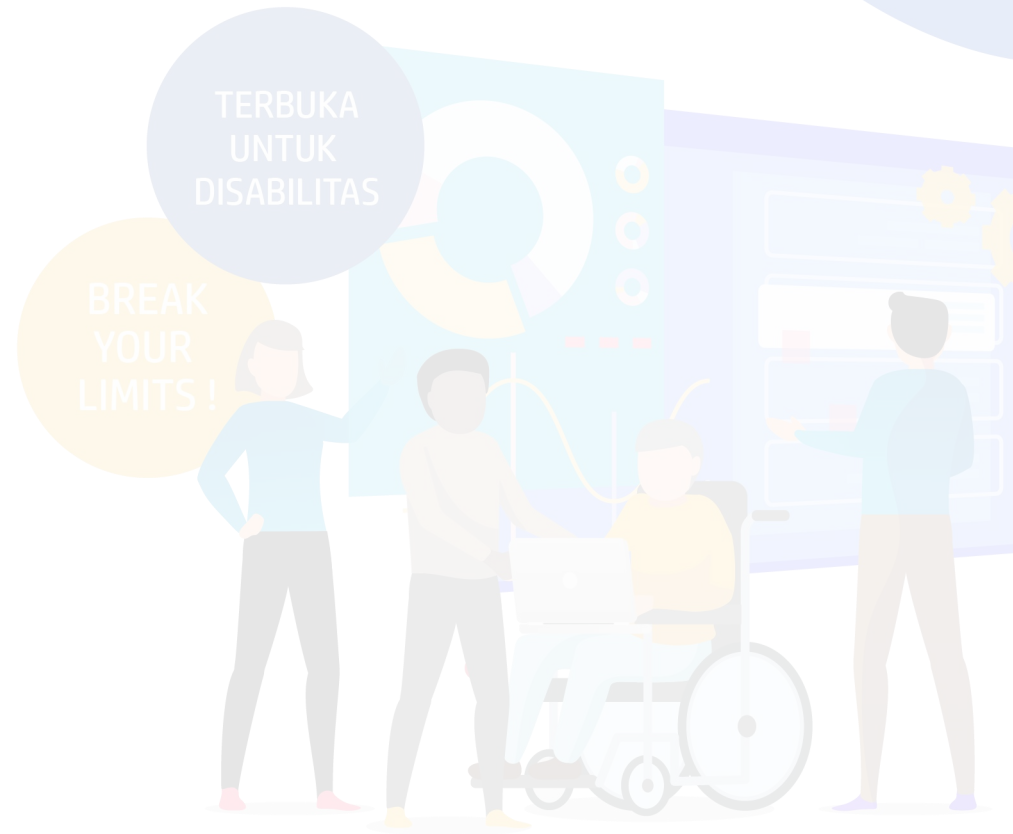
1. `remove()` - menghapus suatu data pada set. Terjadi *error* jika data yang ingin dihapus tidak terdapat di set
2. `discard()` - menghapus suatu data pada set. Tidak akan terjadi *error* jika data yang ingin dihapus tidak terdapat di set

3. `pop()` - data pertama dari set akan dihapus



DIGITAL
TALENT
SCHOLARSHIP

Lab Session - Sets



Referensi

- <https://courses.cognitiveclass.ai>
- Mueller, John Paul. *Beginning Programming with Python for Dummies*, John Wiley & Sons, Incorporated, 2014
- Deitel. *How to program Python*
- <https://www.petanikode.com>
- <https://www.tutorialspoint.com/python>
- <https://developers.google.com/edu/python/lists>
- <https://hiwijaya.com/>
- <https://www.pythonindo.com>



DIGITAL
TALENT
SCHOLARSHIP

IKUTI KAMI



DIGITAL
TALENT
SCHOLARSHIP

- digitalent.kominfo
- digitalent.kominfo
- DTS_kominfo
- Digital Talent Scholarship 2019

Pusat Pengembangan Profesi dan Sertifikasi
Badan Penelitian dan Pengembangan SDM
Kementerian Komunikasi dan Informatika
Jl. Medan Merdeka Barat No. 9
(Gd. Belakang Lt. 4 - 5)
Jakarta Pusat, 10110



digitalent.kominfo.go.id



digitalent.kominfo.go.id