

# TIPE DATA LANJUT DI PYTHON (COLLECTION)

OLEH : ANDRI HERYANDI, M.T.



02

# TIPE DATA DI PYTHON

01158 - ALGORITMA DAN STRUKTUR DATA 2

## ■ Tipe data bawaan (built-in) yang ada di python :

- int - Integer
- float
- bool - Boolean
- str - String

1 objek hanya menampung 1 data saja

- list
- tuple
- set
- dictionary

1 objek bisa menampung banyak data (collection)



# COLLECTION

## 01158 - ALGORITMA DAN STRUKTUR DATA 2

- Collection adalah penampung (containers) yang digunakan untuk menyimpan sekumpulan data.
- Berikut adalah beberapa tipe data collection yang sudah built-in di python (tidak perlu install/import module), yaitu :
  - List
  - Tuple
  - Set
  - Dictionary
- Selain collection tersebut di atas, ada juga collection tambahan yang bisa dipakai yang ada di module **collections**.



# LIST

## 01158 - ALGORITMA DAN STRUKTUR DATA 2

- List adalah struktur data python yang mempunyai karakteristik sebagai berikut :
  - Mutable → Datanya boleh dimodifikasi (tambah, update, hapus) setelah objeknya dibuat
  - Ordered → Urutan data sesuai dengan urutan pengisiannya.
  - Allow Duplication → Datanya boleh ada yang sama



# MEMBUAT LIST

## 01158 - ALGORITMA DAN STRUKTUR DATA 2

- Membuat list dapat dilakukan dengan menuliskan deretan data diapit dengan tanda kurung siku ( **[ ]** ).

```
data = [] # list kosong  
print(data)  
data = [5, 9, 7, 8, 7, 2]  
print(data)
```

Hasil Eksekusi :

```
[]  
[5, 9, 7, 8, 7, 2]
```

1. Ordered (urutan sama)
2. Boleh ada data yang sama



# MENGAKSES ELEMEN LIST

01158 - ALGORITMA DAN STRUKTUR DATA 2

- Untuk mengakses elemen yang berada dalam list, maka gunakan indexnya.
- Index selalu dimulai dari 0. Index 0 menunjuk ke data pertama.

```
data = [5, 9, 7, 8, 7, 2]
print("Data 0 : ", data[0])
print("Data 1 : ", data[1])
print("Data 2 : ", data[2])

data[2] = 99 # mengubah elemen
print("Data : ", data)
```

Hasil Eksekusi :

```
Data 0 : 5
Data 1 : 9
Data 2 : 7
Data : [5, 9, 99, 8, 7, 2]
```



# PENELUSURAN ELEMEN LIST

01158 - ALGORITMA DAN STRUKTUR DATA 2

- Penelusuran dilakukan dengan mengunjungi elemen satu per satu dari elemen pertama sampai elemen terakhir

```
data = [5, 9, 7, 8, 7]
# cara 1 (index)
for i in range(len(data)):
    print("Data ke-",i," = ", data[i])
# cara 2
for i in data:
    print(i)
```

Hasil Eksekusi :

```
Data ke- 0   = 5
Data ke- 1   = 9
Data ke- 2   = 7
Data ke- 3   = 8
Data ke- 4   = 7
5
9
7
8
7
```



# MENAMBAH ELEMEN LIST

01158 - ALGORITMA DAN STRUKTUR DATA 2

- Penambahan elemen baru ke suatu list data dilakukan dengan melakukan salah satu dari cara berikut :
  1. Menggunakan operator + (penggabungan list/*list concatenation*)  
Gunakan operator + untuk melakukan penggabungan 2 list menjadi 1 list baru.
  2. Menggunakan function list.append  
Digunakan untuk melakukan penambahan elemen baru di posisi akhir.
  3. Menggunakan function list.insert  
Digunakan untuk melakukan penambahan elemen baru di posisi tertentu.
  4. Menggunakan function list.extend  
Digunakan untuk menambahkan satu atau banyak data (iterable) ke dalam list.





# MENAMBAH ELEMEN LIST

01158 - ALGORITMA DAN STRUKTUR DATA 2

## ■ Contoh penambahan elemen list

```
data= [] # kosong
print(data)
data = data + [1, 4] #penggabungan list
print(data)
data.append(5)
print(data)
data.insert(0,2) #sisip 2 di index 0
print(data)
data.insert(2,3) #sisip 4 di index 2
print(data)
data.extend([8, 9, 7])
print(data)
```

Hasil Eksekusi :

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



# MENGHAPUS ELEMEN LIST

01158 - ALGORITMA DAN STRUKTUR DATA 2

- Menghapus elemen yang ada dalam list bisa dilakukan dengan cara :

1. Menggunakan function list.**remove**

Digunakan untuk menghapus suatu elemen tertentu dari list. Jika ada data yang sama lebih dari 1 maka yang dihapus adalah elemen yang cocok pertama.

2. Menggunakan function list.**pop**

Digunakan untuk menghapus suatu elemen dari list pada posisi tertentu dan mengembalikan elemen datanya.

3. Menggunakan perintah **del**.

Digunakan untuk menghapus suatu elemen dari list.



# MENGHAPUS ELEMEN LIST

01158 - ALGORITMA DAN STRUKTUR DATA 2

## ■ Contoh penghapusan elemen list

```
data = [5, 7, 5, 3, 2, 6, 8]
print(data)
data.remove(7)
print(data)
data.remove(5)
print(data)
dihapus = data.pop(1) # hapus index 1
print("Dihapus : ", dihapus)
print(data)
del data[2] # hapus index 2
print(data)
```

## Hasil Eksekusi :

```
[5, 7, 5, 3, 2, 6, 8]
[5, 5, 3, 2, 6, 8]
[5, 3, 2, 6, 8]
Dihapus : 3
[5, 2, 6, 8]
[5, 2, 8]
```



# TUPLE

## 01158 - ALGORITMA DAN STRUKTUR DATA 2

- Tuple adalah struktur data python yang mempunyai karakteristik sebagai berikut :
  - Immutable → Datanya **tidak bisa** dimodifikasi (tambah, update, hapus) setelah objeknya dibuat
  - Ordered → Urutan data sesuai dengan urutan pengisiannya.
  - Allow Duplication → Datanya boleh ada yang sama
- Tuple tidak bisa diedit elemennya, tetapi tuple bisa diganti/replace.



# MEMBUAT TUPLE

01158 - ALGORITMA DAN STRUKTUR DATA 2

- Membuat tuple dapat dilakukan dengan menuliskan deretan data diapit dengan tanda kurung ( **( )** ).

```
t1 = () # tuple kosong
print(t1)
t2 = (5, ) # tuple 1 data
print(t2)
t3 = (5, 7, 8, 6, 7)
print(t3)
t3 = (6, 3, 5) # Replace
print(t3)
```

Hasil Eksekusi :

```
()
(5,)
(5, 7, 8, 6, 7)
(6, 3, 5)
```

1. Ordered (urutan sama)
2. Boleh ada data yang sama



# MENGAKSES ELEMEN TUPLE

01158 - ALGORITMA DAN STRUKTUR DATA 2

- Untuk mengakses elemen yang berada dalam tuple, maka gunakan indexnya.
- Index selalu dimulai dari 0. Index 0 menunjuk ke data pertama.

```
data = (5, 9, 7, 8, 7, 2)
print("Data : ", data)
print("Data 0 : ", data[0])
print("Data 1 : ", data[1])
print("Data 2 : ", data[2])
```

Hasil Eksekusi :

```
Data : (5, 9, 7, 8, 7, 2)
Data 0 : 5
Data 1 : 9
Data 2 : 7
```



# PENELUSURAN ELEMEN TUPLE

## 01158 - ALGORITMA DAN STRUKTUR DATA 2

- Penelusuran dilakukan dengan mengunjungi elemen satu per satu dari elemen pertama sampai elemen terakhir

```
data = (5, 9, 7, 8, 7)
# cara 1 (index)
for i in range(len(data)):
    print("Data ke-",i," = ", data[i])
# cara 2 (tanpa index)
for i in data:
    print(i)
```

Hasil Eksekusi :

```
Data ke- 0   = 5
Data ke- 1   = 9
Data ke- 2   = 7
Data ke- 3   = 8
Data ke- 4   = 7
5
9
7
8
7
```



# MEMODIFIKASI ELEMEN TUPLE

01158 - ALGORITMA DAN STRUKTUR DATA 2

- Operasi yang melakukan modifikasi terhadap elemen tuple akan menghasilkan pesan kesalahan.

```
data = (6, 3, 7)
print(data)
print(data[1]) # cetak
data[1] = 20
```

Hasil Eksekusi :

```
(6, 3, 7)
3
Traceback (most recent call last):
  File "C:/Users/../../x.py", line 4, in <module>
    data[1] = 20
TypeError: 'tuple' object does not support item
assignment
```





# SET

## 01158 - ALGORITMA DAN STRUKTUR DATA 2

- Set = himpunan.
- Set adalah struktur data python yang mempunyai karakteristik sebagai berikut :
  - Mutable → Elemennya bisa ditambah atau dihapus, tapi **tidak bisa** diedit setelah objeknya dibuat
  - Unordered → Urutan data bisa sesuai dengan urutan pengisiannya.
  - Not Allow Duplication → Datanya unik (tidak boleh ada yang sama)



# MEMBUAT SET

01158 - ALGORITMA DAN STRUKTUR DATA 2

- Membuat tuple dapat dilakukan dengan menuliskan deretan data diapit dengan tanda kurung kurawal ( `{ }` ).

```
s1 = set({}) # set kosong
print(s1)
print(type(s1))
s2 = {5, 1, 3, 5, 2, 0}
print(s2)
s3 = {'A', 'B', 'B', 'a', 'C'}
print(s3)
```

Hasil Eksekusi :

```
set()
<class 'set'>
{0, 1, 2, 3, 5}
{'B', 'A', 'C', 'a'}
```

1. Unordered (urutan tidak sama)
2. Tidak boleh ada data yang sama



# PENELUSURAN ELEMEN SET

01158 - ALGORITMA DAN STRUKTUR DATA 2

- Penelusuran dilakukan dengan mengunjungi elemen satu per satu dari elemen pertama sampai elemen terakhir
- Elemen set **tidak bisa** diakses menggunakan index.

```
data = {5, 9, 7, 8, 7}
for i in data:
    print(i)
```

Hasil Eksekusi :

```
8
9
5
7
```



# PENAMBAHAN ELEMEN SET

01158 - ALGORITMA DAN STRUKTUR DATA 2

- Penambahan elemen set bisa dilakukan dengan cara :

1. Menggunakan function set.**add**

Digunakan untuk menambahkan sebuah elemen ke dalam set.

2. Menggunakan function set.**update**

Digunakan untuk menambahkan sebuah atau banyak elemen. Elemen yang ditambahkan berupa iterable object (seperti list, tuple).



# MENAMBAH ELEMEN SET

01158 - ALGORITMA DAN STRUKTUR DATA 2

## ■ Contoh penambahan elemen set

```
anggota = set({}) # set kosong
print(anggota)
anggota.add("ade")
print(anggota)
anggota.add("budi")
print(anggota)
anggota.update(["cepi", "erni", "budi"])
print(anggota)
```

Hasil Eksekusi :

```
set()
{'ade'}
{'ade', 'budi'}
{'ade', 'cepi', 'budi', 'erni'}
```



# PENGHAPUSAN ELEMEN SET

01158 - ALGORITMA DAN STRUKTUR DATA 2

## ■ Penghapusan elemen set bisa dilakukan dengan cara :

### 1. Menggunakan function set.**remove**

Digunakan untuk menambahkan sebuah elemen ke dalam set. Function ini bisa menghasilkan KeyError jika elemen yang akan dihapus tidak ditemukan.

### 2. Menggunakan function set.**discard**

Digunakan untuk menambahkan sebuah elemen ke dalam set. Function ini tidak menghasilkan KeyError jika elemen yang akan dihapus tidak ditemukan.

### 3. Menggunakan function set.**pop**

Menghapus elemen pertama dari set dan mereturnkan isinya.



# PENGHAPUSAN ELEMEN SET

01158 - ALGORITMA DAN STRUKTUR DATA 2

## ■ Contoh penghapusan elemen set :

```
data = { 2, 3, 9, 4, 6, 5}
print(data)
data.remove(9)
#data.remove(99) #error
print(data)
data.discard(6)
data.discard(66) #tidak error
print(data)
dihapus = data.pop()
print("Dihapus ", dihapus)
print(data)
```

## Hasil Eksekusi :

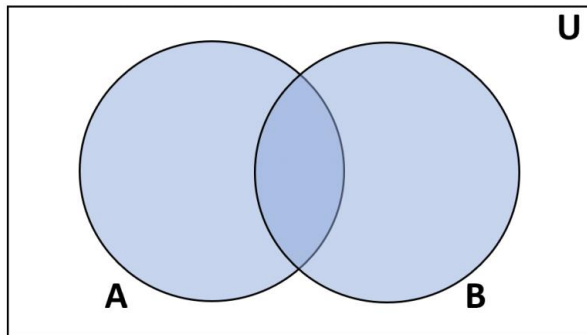
```
{2, 3, 4, 5, 6, 9}
{2, 3, 4, 5, 6}
{2, 3, 4, 5}
Dihapus 2
{3, 4, 5}
```



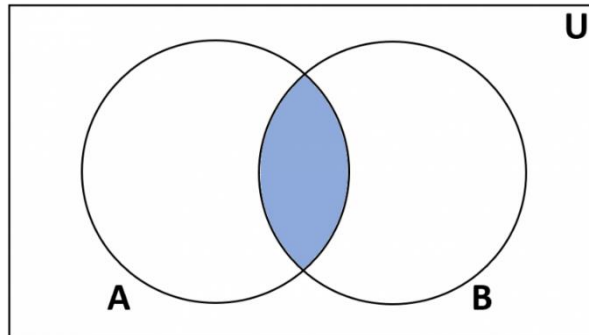
# OPERASI SET (OPERASI HIMPUNAN)

01158 - ALGORITMA DAN STRUKTUR DATA 2

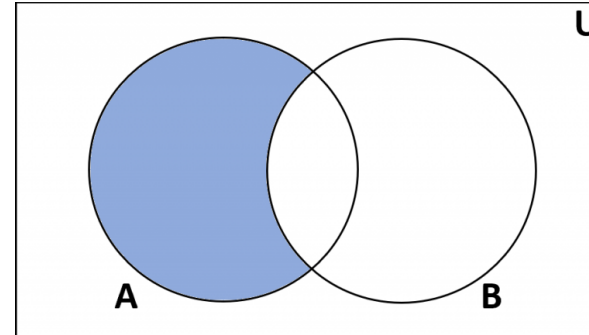
- Operasi-operasi khusus dalam set (himpunan)
  - Union (penggabungan)
  - Intersection (irisan)
  - Difference (perbedaan)
  - Symmetric Difference



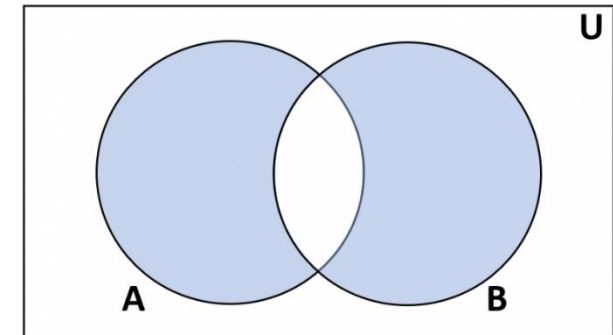
Union



Intersection



Difference



Symmetric Difference



# OPERASI KHUSUS DALAM SET (OPERASI HIMPUNAN)

01158 - ALGORITMA DAN STRUKTUR DATA 2

## ■ Contoh operasi-operasi pada set :

```
a = {1, 2, 3, 4, 5}
b = {4, 5, 6, 7, 8}
print("Himpunan A: ", a)
print("Himpunan B: ", b)
union = a.union(b) # a | b
print("Union : ",union)
intersect = a.intersection(b) # a & b
print("Intersection : ", intersect)
diff = a.difference(b) # a - b
print("Difference : ", diff)
syndiff=a.symmetric_difference(b) # a ^ b
print("Symmetric Diff : ",syndiff)
```

## Hasil Eksekusi :

```
Himpunan A:  {1, 2, 3, 4, 5}
Himpunan B:  {4, 5, 6, 7, 8}
Union :  {1, 2, 3, 4, 5, 6, 7, 8}
Intersection :  {4, 5}
Difference :  {1, 2, 3}
Symmetric Diff :  {1, 2, 3, 6, 7, 8}
```



# DICTIONARY (DICT)

01158 - ALGORITMA DAN STRUKTUR DATA 2

- Dictionary (**dict**) adalah struktur data python yang mempunyai karakteristik sebagai berikut :
  - Mutable → Elemen dictionary dapat dimodifikasi (tambah, edit, hapus) setelah objeknya dibuat.
  - Not Allow Duplication → Tidak boleh memiliki elemen yang mempunyai key yang sama.
  - Ordered → Urutan elemen sesuai urutan pengisiannya
- Setiap elemen dictionary harus berupa pasangan antara key dan value.
- Key dan value dipisahkan dengan tanda titik 2 ( : )
- Pemisah antara elemen adalah tanda koma ( , )



# MEMBUAT DICTIONARY

01158 - ALGORITMA DAN STRUKTUR DATA 2

- Dictionary dibuat dengan menuliskan elemen-elemennya diapit dengan tanda kurung kurawal ( { } )

```
mhs1 = {'nama': 'Asep', 'umur': 20, 'kota': 'Garut'}  
print("Mhs1 : ", mhs1)  
mhs2 = {'nama' : 'Budi',  
        'umur'  : 21,  
        'kota'  : 'Bandung'  
        }  
print("Mhs2 : ", mhs2)
```

Hasil Eksekusi :

```
Mhs1 :  {'nama': 'Asep', 'umur': 20, 'kota': 'Garut'}  
Mhs2 :  {'nama': 'Budi', 'umur': 21, 'kota': 'Bandung'}
```



# MENGAKSES ELEMEN DICTIONARY

01158 - ALGORITMA DAN STRUKTUR DATA 2

- Untuk mengakses elemen dictionary dapat dilakukan dengan cara berikut :
  - Menggunakan tanda kurung siku. Key ditulis diantara kurung siku.
  - Menggunakan function dict.**get**. Key ditulis sebagai parameter ke function tersebut
- Nama key bersifat case-sensitive.

```
mhs = {'nama' : 'Budi',  
      'umur' : 21,  
      'kota' : 'Bandung'}  
print("Mhs : ", mhs)  
print("Nama : ", mhs['nama'])  
print("Umur : ", mhs['umur'])  
print("Kota : ", mhs.get('kota'))
```

Hasil Eksekusi :

```
Mhs : {'nama': 'Budi', 'umur': 21,  
      'kota': 'Bandung'}  
Nama : Budi  
Umur : 21  
Kota : Bandung
```



# PENELUSURAN ELEMEN DICTIONARY

01158 - ALGORITMA DAN STRUKTUR DATA 2

- Penelusuran elemen dictionary dapat dilakukan dengan menggunakan for.
- Variable iterator pengulangan for akan memberikan nilai berupa key dari elemen dictionary.

```
mhs = {'nama' : 'Budi',  
      'umur' : 21,  
      'kota' : 'Bandung'}  
  
for key in mhs:  
    print(key)  
print("-----")  
for key in mhs:  
    print(key, '=>', mhs[key])
```

Hasil Eksekusi :

```
nama  
umur  
kota  
-----  
nama => Budi  
umur => 21  
kota => Bandung
```



# MENAMBAH/MENGUPDATE ELEMEN DICTIONARY

01158 - ALGORITMA DAN STRUKTUR DATA 2

- Penambahan elemen ke dalam dictionary dapat dilakukan dengan cara :

1. Menggunakan tanda kurung siku

Digunakan untuk menambahkan elemen baru, jika key-nya belum terdaftar. Jika key sudah terdaftar, maka akan dianggap mengupdate elemen.

2. Menggunakan function `dict.update`

Digunaan untuk menambah elemen baru, jika key-nya belum terdaftar. Jika key sudah terdaftar, maka akan dianggap update elemen. Parameter untuk function ini harus berupa dictionary (pasangan key:value).



# MENAMBAH/MENGUPDATE ELEMEN DICTIONARY

01158 - ALGORITMA DAN STRUKTUR DATA 2

## ■ Contoh penambahan/pengupdatean elemen ke dalam dictionary

```
mhs = {'nama' : 'Budi',  
      'kota' : 'Bandung'  
      }  
print("Mhs : ", mhs)  
mhs['kelas'] = 'IF-1' # menambah  
print("Mhs : ", mhs)  
mhs['kelas'] = 'IF-2' # mengupdate  
print("Mhs : ", mhs)  
mhs.update({'nilai':86}) # menambah  
print("Mhs : ", mhs)  
mhs.update({'nilai':77, 'nama': 'Acep'})  
print("Mhs : ", mhs)
```

### Hasil Eksekusi :

```
Mhs : {'nama': 'Budi', 'kota': 'Bandung'}  
Mhs : {'nama': 'Budi', 'kota': 'Bandung',  
      'kelas': 'IF-1'}  
Mhs : {'nama': 'Budi', 'kota': 'Bandung',  
      'kelas': 'IF-2'}  
Mhs : {'nama': 'Budi', 'kota': 'Bandung',  
      'kelas': 'IF-2', 'nilai': 86}  
Mhs : {'nama': 'Acep', 'kota': 'Bandung',  
      'kelas': 'IF-2', 'nilai': 77}
```



# PENGHAPUSAN ELEMEN DICTIONARY

01158 - ALGORITMA DAN STRUKTUR DATA 2

- Penghapusan elemen dictionary dapat dilakukan dengan cara :

1. Menggunakan function `dict.pop`

Digunakan untuk menghapus sebuah elemen berdasarkan nama key-nya. Function ini akan mereturnkan nilai yang akan dihapus.

2. Menggunakan perintah `del`

Digunaan untuk menghapus elemen dictionary atau keseluruhan dictionary.





# PENGHAPUSAN ELEMEN DICTIONARY

01158 - ALGORITMA DAN STRUKTUR DATA 2

## ■ Contoh penghapusan elemen dari dictionary

```
mhs = {'nama' : 'Budi',  
      'kota' : 'Bandung',  
      'nilai' : 77  
      }  
print("Mhs : ", mhs)  
dihapus = mhs.pop('kota')  
print("Dihapus : ", dihapus)  
print("Mhs : ", mhs)  
del mhs['nilai']  
print("Mhs : ", mhs)
```

### Hasil Eksekusi :

```
Mhs : {'nama': 'Budi', 'kota': 'Bandung',  
      'nilai': 77}  
Dihapus : Bandung  
Mhs : {'nama': 'Budi', 'nilai': 77}  
Mhs : {'nama': 'Budi'}
```



# IMPLEMENTASI DARI COLLECTION

01158 - ALGORITMA DAN STRUKTUR DATA 2

- Dengan collection anda bisa menyimpan banyak data dalam sebuah variable.
- Didalam sebuah collection, boleh memiliki collection lain.



# IMPLEMENTASI DARI COLLECTION

01158 - ALGORITMA DAN STRUKTUR DATA 2

## ■ Contoh implementasi collection

```
kelas = ({ 'nama'      : "IF-1",
           'siswa'     : [{ 'nim': '10119001', 'nama': 'Asep' },
                          { 'nim': '10119002', 'nama': 'Budi' },
                          { 'nim': '10119003', 'nama': 'Cecep' }
            ],
          { 'nama'      : "IF-2",
           'siswa'     : [{ 'nim': '10119004', 'nama': 'Dede' },
                          { 'nim': '10119005', 'nama': 'Erna' }
            ]
          }
)

for k in kelas:
    print("Nama Kelas : ", k['nama'])
    print("Banyak Siswa : ", len(k['siswa']))
    print("Daftar Siswa : ")
    for s in k['siswa']:
        print(s["nim"], " - ", s["nama"])
    print("-----")
```

## Hasil Eksekusi :

```
Nama Kelas : IF-1
Banyak Siswa : 3
Daftar Siswa :
10119001 - Asep
10119002 - Budi
10119003 - Cecep
-----
```

```
Nama Kelas : IF-2
Banyak Siswa : 2
Daftar Siswa :
10119004 - Dede
10119005 - Erna
-----
```



# CATATAN AKHIR

## 01158 - ALGORITMA DAN STRUKTUR DATA 2

- Jika anda ingin menjaga urutan pengisian data, maka gunakan list atau tuple
- Jika anda ingin memastikan elemen-elemen dalam suatu collection bersifat unik dan urutan tidak dianggap penting, maka gunakanlah set
- Jika anda ingin memastikan bahwa suatu data tidak boleh diubah setelah didefinisikan/dibuat, maka buatlah dalam bentuk tuple.
- Jika anda membutuhkan collection yang ingin diakses dengan menggunakan suatu key, maka gunakanlah dictionary,



# FORUM DISKUSI

01158 - ALGORITMA DAN STRUKTUR DATA 2



**LMS UNIKOM**

**<https://lms.unikom.ac.id>**



**Group Whatsapp  
Perkuliahahan**



**Youtube Comments**

