

MODUL MATA KULIAH

BAHASA PEMROGRAMAN DASAR

PG168 – 3 SKS



**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

**JAKARTA
SEPTEMBER 2021**

TIM Penyusun

Agus Umar Hamdani, M.Kom
Tri Ika Jaya Kusumawati, M.Kom¹



MODUL PERKULIAHAN #4 OPERASI PADA LIST, TUPLE, DAN DICTIONARY

Capaian Pembelajaran	:	Mahasiswa Mampu: 1. Memahami operasi manipulasi elemen nilai pada List, Tuple dan Dictionary.
Sub Pokok Bahasan	:	1. Operasi manipulasi elemen nilai pada List, Tuple dan Dictionary. 2. Penggunaan fungsi untuk manipulasi elemen nilai pada List, Tuple dan Dictionary.
Daftar Pustaka	:	<ol style="list-style-type: none">1. Zarman, Wendi dan Wicaksono, Mochamad Fajar. <i>"Implementasi Algoritma dalam bahasa Python"</i>. Edisi Pertama. Bandung : Penerbit Informatika. 2020.2. Kurniawati, Arik. <i>"Algoritma dan Pemrograman menggunakan Python"</i>. Edisi Pertama. Yogyakarta : Depublish. 2016.3. Ismah. <i>"Pemrograman Komputer Dasar-dasar Python"</i>. Jakarta : Fakultas Ilmu Pendidikan Universitas Muhammadiyah Jakarta. 2017.4. Irfani, M. Haviz dan Dafid. <i>"Modul Praktikum Dasar Pemrograman dengan bahasa Python"</i>. Palembang : Sekolah Tinggi Manajemen Informatika Global Informatika Multidata Palembang. 2016.5. Fikri, Rijalul. <i>"Praktikum Algoritma dan Pemrograman Komputer"</i>. Surabaya : Program Studi Teknik Komputer dan Telematika Jurusan Teknik Elektro Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember. 2010.6. Sugiana, Owo. <i>Membuat Aplikasi Bisnis menggunakan bahasa Python dan database berbasis SQL</i>. Jakarta. 2003.7. Septian, Ridwan Fadjar. <i>Buku Serial Open Source Belajar Pemrograman Python Dasar</i>. Versi 1. Bandung : POSS-UPI. 2013.8. Hendri. <i>Cepat Mahir Python</i>. Ilmu Komputer.com. 2003.9. Herho, Sandy H.S. <i>Tutorial Pemrograman Python 2 Untuk pemula</i>. Bandung: WCPL Press. 2017.10. <i>Welcome to Python.org</i> (diakses pada 29 September 2021 pukul : 21.00 WIB)11. <i>Programiz.com</i>. Python Operators (https://www.programiz.com/python-programming/operators diakses pada 29 September 2021 pukul 21.32 WIB)

PRAKTIKUM 4

OPERASI PADA LIST, TUPLE DAN DICTIONARY

3.1 Teori Singkat

Data yang kompleks biasanya dapat menampung obyek atau entitas dengan jumlah yang banyak. Dalam bahasa Python terdapat tiga struktur data yang digunakan untuk menampung obyek atau entitas dengan jumlah yang banyak, yaitu : **List**, **Tuple** dan **Dictionary**. Ketiga tipe data tersebut adalah obyek Python, dimana anda dapat menggunakan method-method yang berhubungan dengan pengolahan pada struktur data tersebut. Selain itu, terdapat pula fungsi (function) untuk mengolah ketiga struktur data tersebut, seperti : mencari nilai Max, Min, hitung Panjang dan perbandingan isi.

Untuk mendefinisikan sebuah List, anda cukup membuat variabel, kemudian isi variabel tersebut. List diawali dengan tanda '[' dan diakhiri dengan tanda ']'. Isi List dapat beragam dapat berupa : string, number, object dan bahkan List juga. List dapat ditambah, dirubah data pada elemennya, dapat dihapus data pada elemennya dan menghapus seluruh isi List.

Sintak penulisan List : **nama_variabel = [nilai1, nilai2]**

Daftar operasi yang bisa dijalankan pada List dapat dilihat pada tabel 4.1.

Tabel 4.1 Daftar Operasi pada List

Nama Operasi	Deskripsi	Sintak Penulisan
Any()	Bernilai True, jika salah satu item dalam list bernilai Boolean True.	variabel_list.any()
All()	Bernilai True, jika semua item dalam list bernilai Boolean True.	variabel_list.all()

Append()	Digunakan untuk menambahkan elemen nilai ke List secara berkelompok.	variabel_list.append(object)
Extend()	Digunakan untuk menambahkan elemen nilai ke List secara tunggal.	variabel_list.extend(object)
Copy()	Digunakan untuk menyalin semua isi list.	variabel_list.copy()
Clear()	Digunakan untuk menghapus semua isi list.	variabel_list.clear()
Count()	Digunakan untuk menghitung jumlah item / object pada List.	variabel_list.count()
Del	Digunakan untuk menghapus semua isi list.	del variabel_list (object)
Insert()	Digunakan untuk menambahkan elemen baru ke dalam list.	variabel_list.insert(object)
Pop()	Digunakan untuk menghapus isi list pada index tertentu.	variabel_list.pop(index)
Sort()	Digunakan untuk mengurutkan isi list secara menaik (ascending).	variabel_list.sort()
Sum()	Digunakan untuk menghitung total isi elemen pada List.	variabel_list.sum()
Remove()	Digunakan untuk menghapus isi list.	variabel_list.remove()
Reverse()	Digunakan untuk membalikkan urutan isi list	variabel_list.reverse()

	secara menurun (descending).	
--	---------------------------------	--

Hampir sama seperti List, Tuple diawali dengan tanda '(' dan diakhir dengan tanda ')'. Elemen pada Tuple tidak dapat dihapus dan diubah.

Sintak penulisan Tuple : **nama_variabel = (nilai1, nilai2)**

Daftar operasi yang bisa dijalankan pada Tuple dapat dilihat pada tabel 4.2.

Tabel 4.2 Daftar Operasi pada Tuple

Nama Operasi	Deskripsi	Sintak Penulisan
Any()	Bernilai True, jika salah satu item dalam Dictionary bernilai Boolean True.	variabel_tuple.any()
All	Bernilai True, jika semua item dalam Dictionary bernilai Boolean True.	variabel_tuple.all()
Extend()	Digunakan untuk menambahkan elemen nilai ke List secara tunggal.	variabel_tuple.extend(object)
Index()	Digunakan untuk pencarian nilai dalam elemen Tuple dan mengembalikan nilai index-nya.	variabel_tuple.index(object)
Len()	Digunakan untuk menghitung jumlah item / object pada Dictionary.	variabel_dict.len()
Max()	Digunakan untuk menghitung nilai terbesar dalam elemen Tuple.	variabel_list.max()

Min()	Digunakan untuk menghitung nilai terkecil dalam elemen Tuple.	variabel_list.min()
Sum()	Digunakan untuk menghitung total isi elemen pada List.	variabel_list.sum()

Dictionary berbeda halnya dengan List dan Tuple, setiap data akan memiliki pengenalnya masing-masing. Pengenal tersebut dinamakan dengan "*Key*" dan isi datanya dinamakan "*Value*". Dictionary diawali dengan tanda '{' dan diakhir dengan tanda '}'.

Khusus untuk key pada Dictionary, nilainya harus berupa tipe data yang tidak dapat diganti seperti Tuple, String dan Number. Tetapi umumnya key berisi number dan string. Karena jika anda mencoba memasukkan tipe data yang "*mutable*", maka akan keluar peringatan "*unhashable type*" pada saat mendefinisikan Dictionary yang key-nya berupa tipe data "*mutable*".

Sintak penulisan Dictionary :

nama_variabel = {"Key1": "value", "Key2": "value"}

Daftar operasi yang bisa dijalankan pada Dictionary dapat dilihat pada tabel 4.3.

Tabel 4.3 Daftar Operasi pada Dictionary

Nama Operasi	Deskripsi	Sintak Penulisan
Any()	Bernilai True, jika salah satu item dalam Dictionary bernilai Boolean True.	variabel_dict.any()
All()	Bernilai True, jika semua item dalam Dictionary bernilai Boolean True.	variabel_dict.all()
Append()	Digunakan untuk menambahkan elemen nilai	variabel_dict.append(object)

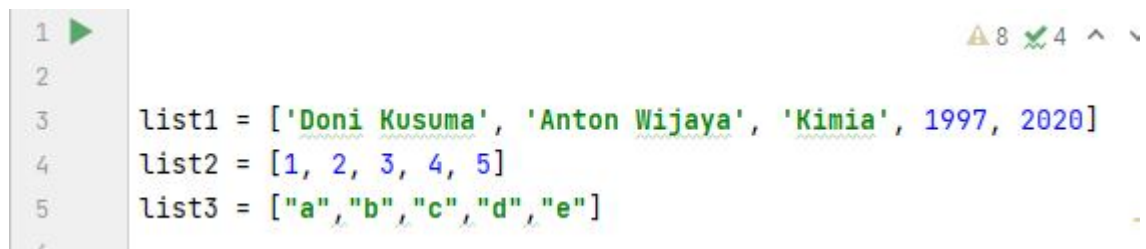
	ke Dictionary secara berkelompok.	
Copy()	Digunakan untuk menyalin semua isi Dictionary.	variabel_dict.copy()
Clear()	Digunakan untuk menghapus semua isi Dictionary.	variabel_dict.clear()
Count()	Digunakan untuk menghitung jumlah item / object pada Dictionary.	variabel_dict.count()
Del	Digunakan untuk menghapus semua isi Dictionary.	del variabel_dict (object)
Insert()	Digunakan untuk menambahkan elemen baru ke dalam Dictionary.	variabel_dict.insert(object)
Pop()	Digunakan untuk menghapus isi elemen Dictionary pada key tertentu.	variabel_dict.pop(index)
Popitem()	Digunakan untuk menghapus isi elemen Dictionary menggunakan key dan value tertentu.	variabel_dict.pop(key, value)
Sort()	Digunakan untuk mengurutkan isi Dictionary secara menaik (ascending).	variabel_dict.sort()
Sum()	Digunakan untuk menghitung total elemen pada Dictionary.	variabel_dict.sum()

Remove()	Digunakan untuk menghapus isi Dictionary.	variabel_dict.remove()
Reverse()	Digunakan untuk membalikkan urutan isi elemen Dictionary secara menurun (descending).	variabel_dict.reverse()

3.2 Manipulasi Elemen Nilai pada List

a. Membuat dan Mengakses Nilai Pada List

Berikut ini adalah contoh pembuatan variabel yang bertipe List, yaitu : list1, list2 dan list3.



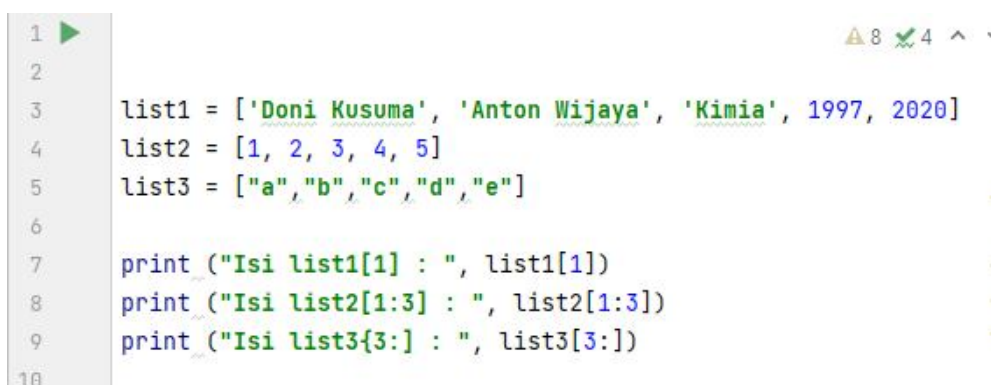
```

1  ▶
2
3  list1 = ['Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020]
4  list2 = [1, 2, 3, 4, 5]
5  list3 = ["a", "b", "c", "d", "e"]
6

```

Gambar 4.1 Mendeklarasikan List

Untuk mengakses elemen nilai dalam List, gunakanlah tanda kurung siku [] dengan memberikan nilai indeks sesuai elemen yang akan diambil.



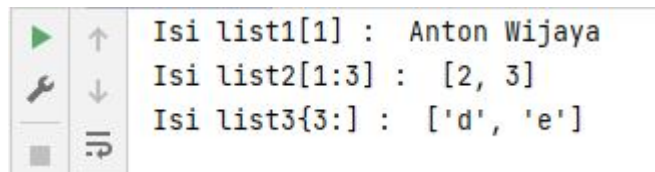
```

1  ▶
2
3  list1 = ['Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020]
4  list2 = [1, 2, 3, 4, 5]
5  list3 = ["a", "b", "c", "d", "e"]
6
7  print ("Isi list1[1] : ", list1[1])
8  print ("Isi list2[1:3] : ", list2[1:3])
9  print ("Isi list3[3:] : ", list3[3:])
10

```

Gambar 4.2 Mengakses Isi List

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :

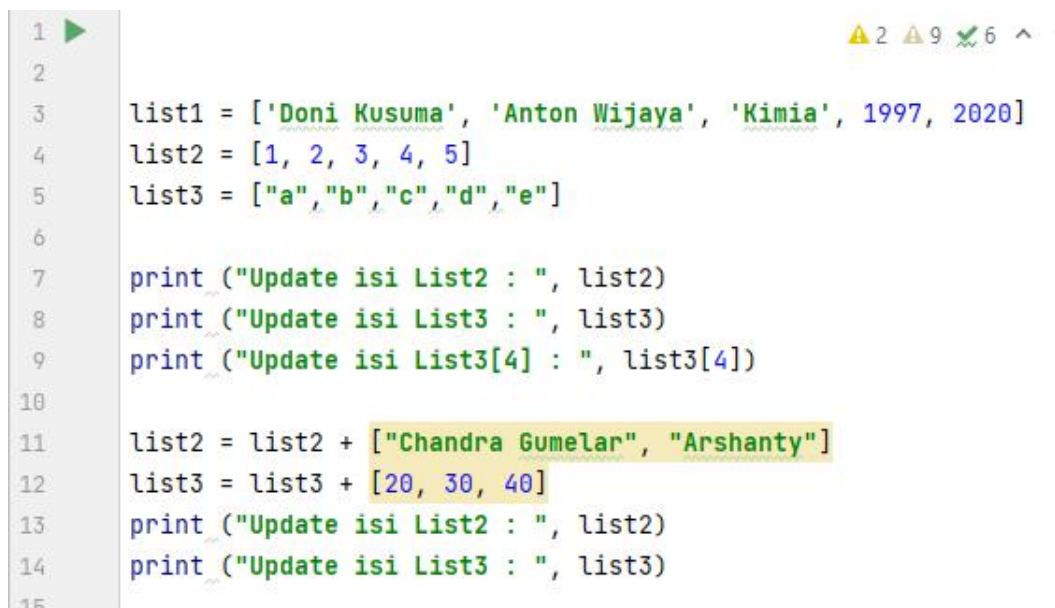


```
Isi list1[1] : Anton Wijaya
Isi list2[1:3] : [2, 3]
Isi list3{3:} : ['d', 'e']
```

Gambar 4.3 Hasil Keluaran Program

b. Menambahkan Nilai pada List

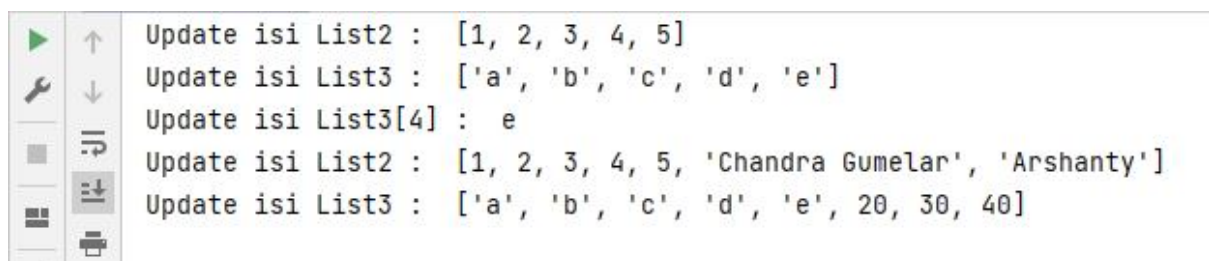
Berikut ini adalah contoh penggunaan operator penggabungan (+) untuk menambahkan elemen nilai bar ke dalam List2 dan List3.



```
1 list1 = ['Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020]
2
3 list2 = [1, 2, 3, 4, 5]
4
5 list3 = ["a", "b", "c", "d", "e"]
6
7 print ("Update isi List2 : ", list2)
8 print ("Update isi List3 : ", list3)
9 print ("Update isi List3[4] : ", list3[4])
10
11 list2 = list2 + ["Chandra Gumelar", "Arshanty"]
12 list3 = list3 + [20, 30, 40]
13 print ("Update isi List2 : ", list2)
14 print ("Update isi List3 : ", list3)
15
```

Gambar 4.4 Menambahkan Elemen Baru ke List

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :

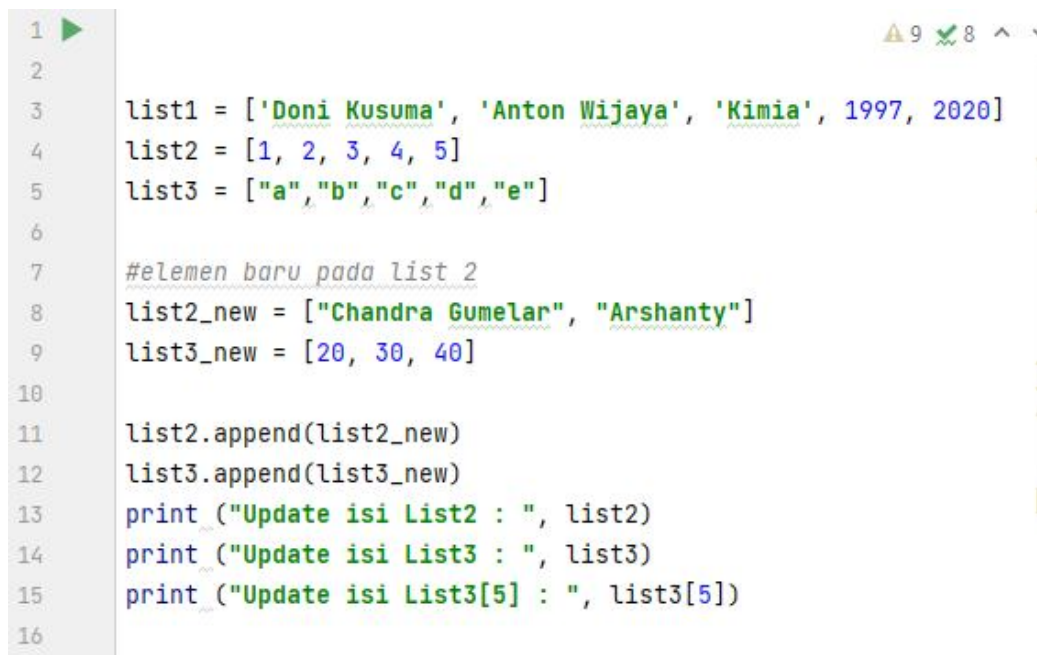


```
Update isi List2 : [1, 2, 3, 4, 5]
Update isi List3 : ['a', 'b', 'c', 'd', 'e']
Update isi List3[4] : e
Update isi List2 : [1, 2, 3, 4, 5, 'Chandra Gumelar', 'Arshanty']
Update isi List3 : ['a', 'b', 'c', 'd', 'e', 20, 30, 40]
```

Gambar 4.5 Hasil Keluaran Program

Berikut ini adalah contoh penggunaan fungsi Append() untuk menambahkan elemen nilai yang baru ke dalam List2 dan List3.

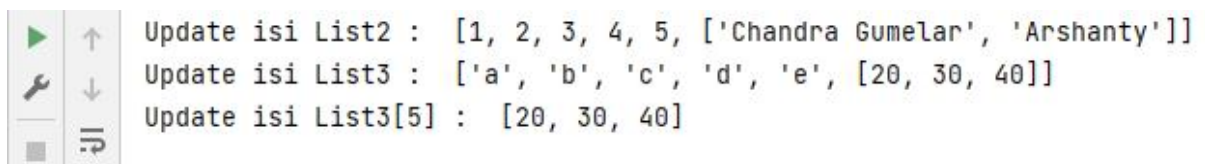
Sintak penulisan : **variabel_list.append(object)**



```
1  ▶
2
3  list1 = ['Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020]
4  list2 = [1, 2, 3, 4, 5]
5  list3 = ["a", "b", "c", "d", "e"]
6
7  #elemen baru pada list 2
8  list2_new = ["Chandra Gumelar", "Arshanty"]
9  list3_new = [20, 30, 40]
10
11 list2.append(list2_new)
12 list3.append(list3_new)
13 print ("Update isi List2 : ", list2)
14 print ("Update isi List3 : ", list3)
15 print ("Update isi List3[5] : ", list3[5])
16
```

Gambar 4.6 Penggunaan fungsi Append() pada List

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :



```
▶ ↑ Update isi List2 : [1, 2, 3, 4, 5, ['Chandra Gumelar', 'Arshanty']]
⚙ ↓ Update isi List3 : ['a', 'b', 'c', 'd', 'e', [20, 30, 40]]
☐ ≡ Update isi List3[5] : [20, 30, 40]
```

Gambar 4.7 Hasil Keluaran Program

Berikut ini adalah contoh penggunaan fungsi Extend() untuk menambahkan elemen nilai yang baru ke dalam List2 dan List3.

Sintak penulisan : **variabel_list.extend(object)**

```

1  ▶
2
3  list1 = ['Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020]
4  list2 = [1, 2, 3, 4, 5]
5  list3 = ["a", "b", "c", "d", "e"]
6
7  #elemen baru pada list 2
8  list2_new = ["Chandra Gumelar", "Arshanty"]
9  list3_new = [20, 30, 40]
10
11 list2.extend(["Chandra Gumelar", "Arshanty"])
12 list3.extend([20, 30, 40])
13
14 print ("Update isi List2 : ", list2)
15 print ("Update isi List3 : ", list3)
16

```

Gambar 4.8 Penggunaan fungsi Extend() pada List

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :

```

▶ ↑ Update isi List2 : [1, 2, 3, 4, 5, 'Chandra Gumelar', 'Arshanty']
⚙ ↓ Update isi List3 : ['a', 'b', 'c', 'd', 'e', 20, 30, 40]

```

Gambar 4.9 Hasil Keluaran Program

Berikut ini adalah contoh penggunaan fungsi **Insert()** untuk menambahkan elemen nilai yang baru ke dalam List2 dan List3.

Sintak penulisan : **variabel_list.insert(index, object)**

```

1  ▶
2
3  list1 = ['Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020]
4  list2 = [1, 2, 3, 4, 5]
5  list3 = ["a", "b", "c", "d", "e"]
6
7  #elemen baru pada list
8
9  list2.insert(2, "Chandra Gumelar")
10 list3.insert(3, 20)
11
12 print("Update isi List2 : ", list2)
13 print("Update isi List3 : ", list3)
14

```

Gambar 4.10 Penggunaan fungsi Insert() pada List

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :

```

▶ ↑ Update isi List2 : [1, 2, 'Chandra Gumelar', 3, 4, 5]
⚙ ↓ Update isi List3 : ['a', 'b', 'c', 20, 'd', 'e']

```

Gambar 4.11 Hasil Keluaran Program

c. Menyalin Elemen Nilai pada List

Fungsi **Copy()** : digunakan untuk menyalin isi dari variabel List asal ke variabel List baru.

Sintak penulisan : **original_list.copy()**.



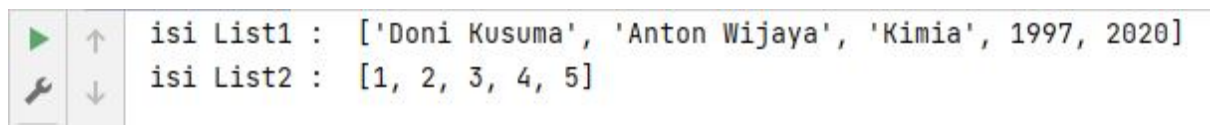
```

1  ▶
2
3  list1 = ['Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020]
4  list2 = [1, 2, 3, 4, 5]
5
6  #menyalin elemen pada list
7  list1_new = list1.copy()
8  list2_new = list2.copy()
9
10 print ("isi List1 : ", list1)
11 print ("isi List2 : ", list2)
12

```

Gambar 4.12 Penggunaan fungsi Copy() pada List

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :



```

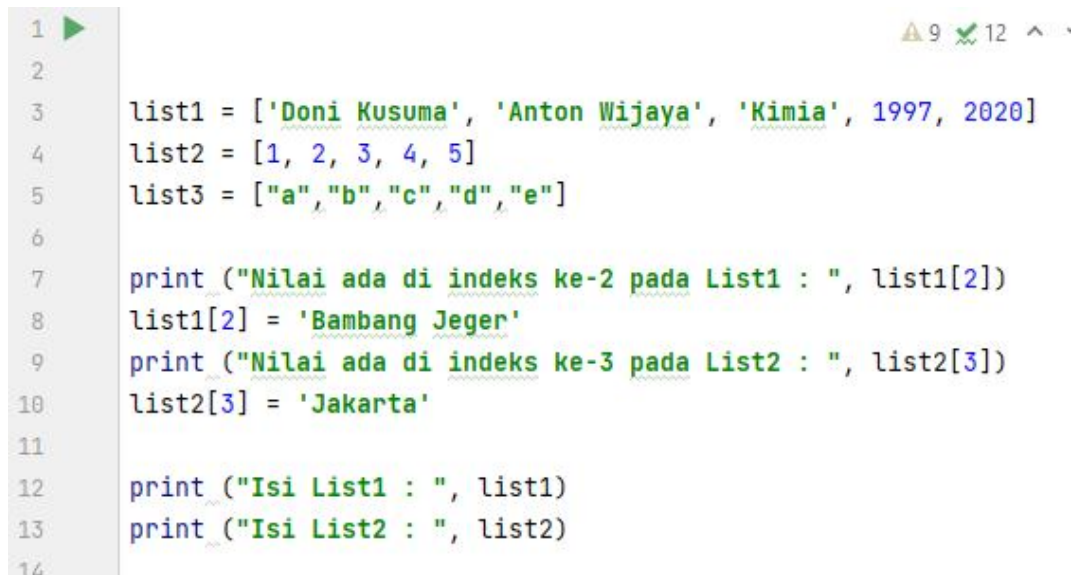
▶ ↑ isi List1 : ['Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020]
⚙ ↓ isi List2 : [1, 2, 3, 4, 5]

```

Gambar 4.13 Hasil Keluaran Program

d. Mengubah Elemen Nilai pada List

Anda dapat mengubah satu atau beberapa elemen nilai dari List dengan memberikan indeks elemen di dalam tanda kurung siku di sebelah kiri operator penugasan (=). Syarat utama elemen nilai pada List dapat diubah adalah elemen tersebut harus berisi nilai terlebih dahulu.



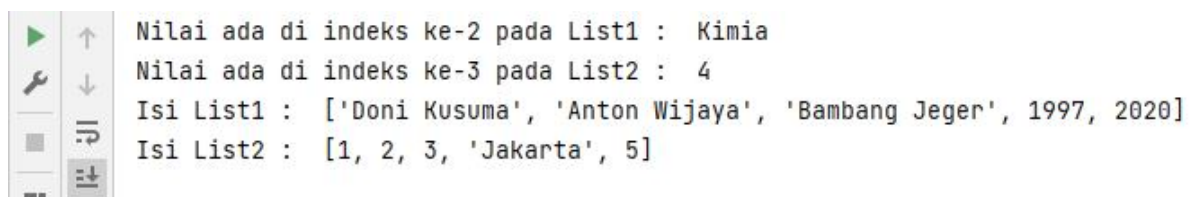
```

1  ▶
2
3  list1 = ['Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020]
4  list2 = [1, 2, 3, 4, 5]
5  list3 = ["a", "b", "c", "d", "e"]
6
7  print ("Nilai ada di indeks ke-2 pada List1 : ", list1[2])
8  list1[2] = 'Bambang Jeger'
9  print ("Nilai ada di indeks ke-3 pada List2 : ", list2[3])
10 list2[3] = 'Jakarta'
11
12 print ("Isi List1 : ", list1)
13 print ("Isi List2 : ", list2)
14

```

Gambar 4.14 Mengubah Elemen pada List

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :



```

▶ ↑ Nilai ada di indeks ke-2 pada List1 :  Kimia
⚙ ↓ Nilai ada di indeks ke-3 pada List2 :  4
⌵ Isi List1 :  ['Doni Kusuma', 'Anton Wijaya', 'Bambang Jeger', 1997, 2020]
⌵ Isi List2 :  [1, 2, 3, 'Jakarta', 5]

```

Gambar 4.15 Hasil Keluaran Program

e. Penggunaan Fungsi Del(), Remove(), Pop() dan Clear() pada List

Untuk menghapus elemen atau item nilai dengan perintah **del**, maka sebutkan nomor indeks yang akan dihapus. Syarat utama elemen nilai pada List dapat dihapus adalah elemen tersebut harus berisi nilai terlebih dahulu.

Sintak penulisan : **del variabel_list (object)**


```

1  ▶
2
3  list1 = ['Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020]
4  list2 = [1, 2, 3, 4, 5]
5  list3 = ["a", "b", "c", "d", "e"]
6
7  #menghapus elemen pada list
8  del list2[2]
9  del list3[3]
10
11 print ("isi List2 : ", list2)
12 print ("isi List3 : ", list3)
13

```

Gambar 4.16 Penggunaan perintah del pada List

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :

```

▶ ↑ isi List2 : [1, 2, 4, 5]
⚙ ↓ isi List3 : ['a', 'b', 'c', 'e']

```

Gambar 4.17 Hasil Keluaran Program

Untuk menghapus elemen atau item nilai dengan fungsi **remove**, maka sebutkan nama object atau elemen yang akan dihapus.

Sintak penulisan : **variabel_list.remove(object)**

```

1  ▶
2
3  list1 = ['Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020]
4  list2 = [1, 2, 3, 4, 5]
5  list3 = ["a", "b", "c", "d", "e"]
6
7  #menghapus elemen pada list
8  list2.remove(3)
9  list3.remove("c")
10 print ("isi List2 : ", list2)
11 print ("isi List3 : ", list3)
12

```

Gambar 4.18 Penggunaan fungsi remove pada List

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :

```
↑ isi List2 : [1, 2, 4, 5]
↓ isi List3 : ['a', 'b', 'd', 'e']
```

Gambar 4.19 Hasil Keluaran Program

Untuk menghapus elemen atau item nilai dengan fungsi **pop()**, maka sebutkan nomor urut indeks yang akan dihapus. Jika anda tidak menyebutkan nomor urut indeks yang akan dihapus, maka elemen yang akan terhapus adalah object atau elemen terakhir yang ada di dalam List.

Sintak penulisan : **variabel_list.pop(index)**

```
1 ▶ list1 = ['Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020]
2
3 list2 = [1, 2, 3, 4, 5]
4 list3 = ["a", "b", "c", "d", "e"]
5
6
7 #menghapus elemen pada list
8 list2.pop(3)
9 list3.pop()
10 print ("isi List2 : ", list2)
11 print ("isi List3 : ", list3)
12
```

Gambar 4.20 Penggunaan fungsi pop pada List

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :

```
▶ ↑ isi List2 : [1, 2, 3, 5]
⚙ ↓ isi List3 : ['a', 'b', 'c', 'd']
```

Gambar 4.21 Hasil Keluaran Program

Untuk menghapus elemen atau item nilai dengan fungsi **clear()**, maka sebutkan nomor urut indeks yang akan dihapus. Setelah perintah ini dijalankan, maka elemen List menjadi kosong.

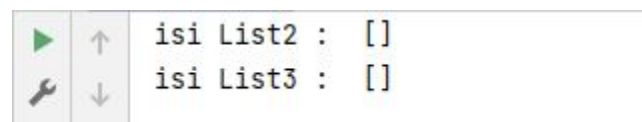
Sintak penulisan : **variabel_list.clear()**



```
1  ▶
2
3  list1 = ['Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020]
4  list2 = [1, 2, 3, 4, 5]
5  list3 = ["a", "b", "c", "d", "e"]
6
7  #menghapus elemen pada list
8  list2.clear()
9  list3.clear()
10 print ("isi List2 : ", list2)
11 print ("isi List3 : ", list3)
12
```

Gambar 4.22 Penggunaan fungsi clear() pada List

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :



```
▶ ↑ isi List2 : []
⚙ ↓ isi List3 : []
```

Gambar 4.23 Hasil Keluaran Program

f. Penggunaan Fungsi Sort pada List

Untuk mengurutkan elemen atau item nilai yang ada di List, gunakanlah fungsi **sort()**. Setelah perintah ini dijalankan, maka elemen List akan diurutkan secara menaik (ascending).

Sintak penulisan : **variabel_list.sort()**



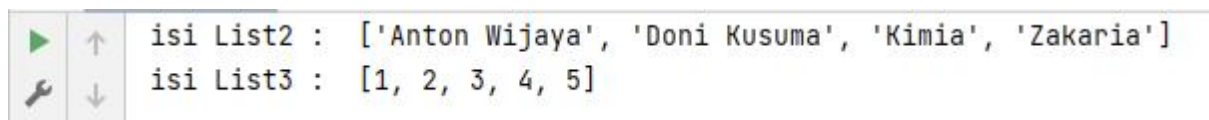
```

1  ▶
2
3  list1 = ['Doni Kusuma', 'Anton Wijaya', 'Kimia', 'Zakaria']
4  list2 = [1, 3, 5, 4, 2]
5
6  #mengurutkan elemen pada list
7  list1.sort()
8  list2.sort()
9  print ("isi List2 : ", list1)
10 print ("isi List3 : ", list2)
11

```

Gambar 4.24 Penggunaan fungsi sort() pada List

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :



```

▶ ↑ isi List2 : ['Anton Wijaya', 'Doni Kusuma', 'Kimia', 'Zakaria']
⚙ ↓ isi List3 : [1, 2, 3, 4, 5]

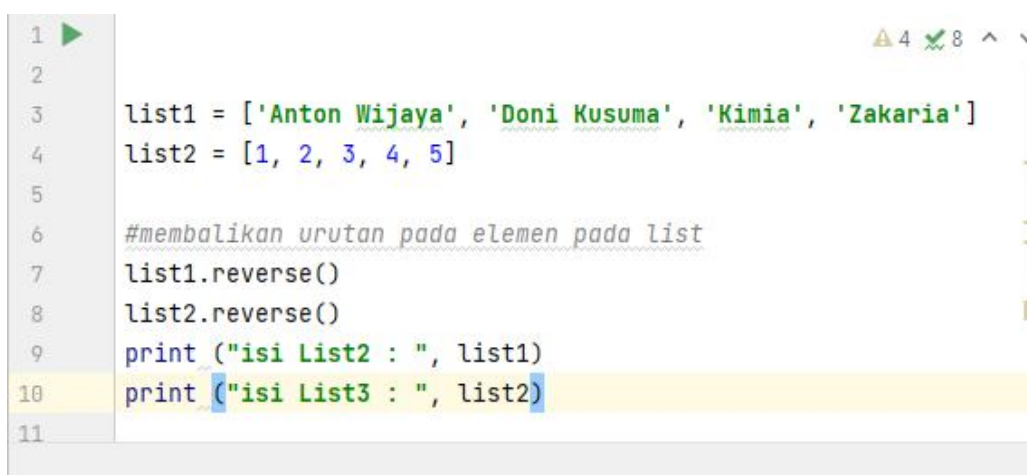
```

Gambar 4.25 Hasil Keluaran Program

g. Penggunaan fungsi Reverse() pada List

Kebalikan dari fungsi Sort() adalah **reverse**. Untuk membalikkan urutan elemen / item yang ada di List, gunakanlah fungsi **reverse()**. Setelah perintah ini dijalankan, maka elemen List akan diurutkan secara descending.

Sintak penulisan : **variabel_list.reverse()**



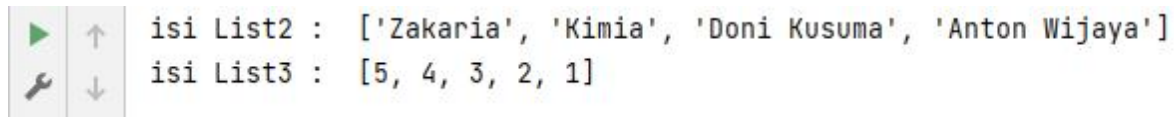
```

1  ▶
2
3  list1 = ['Anton Wijaya', 'Doni Kusuma', 'Kimia', 'Zakaria']
4  list2 = [1, 2, 3, 4, 5]
5
6  #membalikkan urutan pada elemen pada list
7  list1.reverse()
8  list2.reverse()
9  print ("isi List2 : ", list1)
10 print ("isi List3 : ", list2)
11

```

Gambar 4.26 Penggunaan fungsi reverse() pada List

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :



```
isi List2 : ['Zakaria', 'Kimia', 'Doni Kusuma', 'Anton Wijaya']
isi List3 : [5, 4, 3, 2, 1]
```

Gambar 4.27 Hasil Keluaran Program

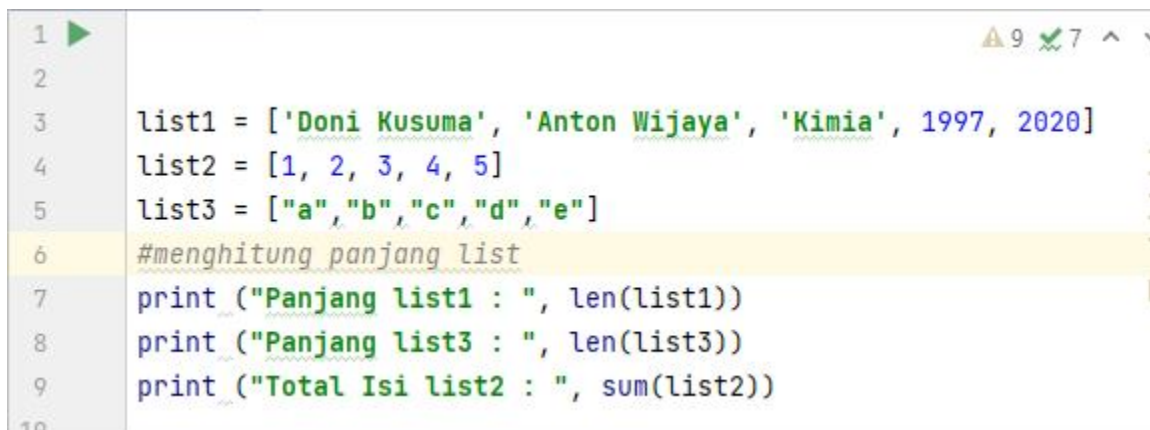
h. Penggunaan Fungsi Len() dan Sum() pada List

Untuk menghitung panjang (length) sebuah List, gunakanlah fungsi **len()**.

Sintak penulisan : **len(variabel_list)**

Untuk menjumlahkan elemen pada List, maka gunakanlah fungsi **sum()**.

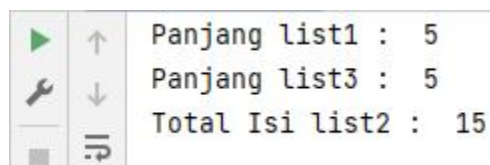
Sintak penulisan : **sum(variabel_list)**



```
1 list1 = ['Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020]
2
3 list2 = [1, 2, 3, 4, 5]
4
5 list3 = ["a", "b", "c", "d", "e"]
6 #menghitung panjang list
7 print ("Panjang list1 : ", len(list1))
8 print ("Panjang list3 : ", len(list3))
9 print ("Total Isi list2 : ", sum(list2))
10
```

Gambar 4.28 Penggunaan fungsi len() dan sum() pada List

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :



```
Panjang list1 : 5
Panjang list3 : 5
Total Isi list2 : 15
```

Gambar 4.29 Hasil Keluaran Program

i. Penggunaan Fungsi Any() dan All() pada List

Fungsi **Any()** : Jika salah satu item dalam tuple bernilai Boolean True, maka fungsi ini mengembalikan True. Jika salah satu item tidak bernilai Boolean True, maka ia mengembalikan False.

Sintak penulisan : **Any(variabel_list)**.

```
1  ▶ #semua elemen di list bernilai true
2  list1 = [0, 1, 3, 4]
3  x = any(list1)
4  print(x)
5
6  #semua elemen di list bernilai false
7  list1 = [0, False]
8  y = any(list1)
9  print(y)
10 #beberapa elemen di list bernilai true,
11 #jika elemen lain bernilai false
12 list1 = [0, False, 5]
13 y = any(list1)
14 print(y)
```

Gambar 4.30 Penggunaan fungsi Any() pada List

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :

```
▶ ↑ True
⚙ ↓ False
>> >> True
```

Gambar 4.31 Hasil Eksekusi Program

Fungsi **All()** : Tidak seperti **any()**, **all()** mengembalikan True hanya jika semua item bernilai Boolean True. Jika tidak, ia mengembalikan False.

Sintak penulisan : **All(variabel_list)**.

Berikut ini contoh penggunaan fungsi **All()** pada List dapat dilihat pada gambar 4.32.

```

1  ▶ #semua elemen di list bernilai true
2  list1 = [0, 1, 3, 4]
3  x = all(list1)
4  print (x)
5
6  #semua elemen di list bernilai false
7  list1 = [0, False]
8  y = all(list1)
9  print (y)
10 #beberapa elemen di list bernilai true,
11 #jika elemen lain bernilai false
12 list1 = [0, False, 5]
13 y = all(list1)
14 print (y)

```

Gambar 4.32 Penggunaan All() pada List

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :

```

▶ ↑ False
⚙ ↓ False
☐ ⏏ False

```

Gambar 4.33 Hasil Eksekusi Program

3.3 Manipulasi Elemen Nilai pada Tuple

a. Membuat dan Mengakses Nilai Pada Tuple

Berikut ini adalah contoh pembuatan variabel yang bertipe List, yaitu : tuple, tuple2 dan tuple3.

```

1  ▶ tuple1 = ('Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020)
2
3  tuple2 = (1, 2, 3, 4, 5)
4
5  tuple3 = ("a", "b", "c", "d", "e")
6

```

Gambar 4.34 Mendeklarasikan Tuple

Untuk mengakses nilai dalam Tuple, gunakanlah tanda kurung () dengan memberikan nilai indeks sesuai elemen yang akan diambil.

```
1  ▶
2
3  tuple1 = ('Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020)
4  tuple2 = (1, 2, 3, 4, 5)
5  tuple3 = ("a", "b", "c", "d", "e")
6
7  #mengakses isi Tuple
8  print ("Isi tuple[1] : ", tuple1)
9  print ("Isi tuple[1:3] : ", tuple2)
10 print ("Isi tuple[3:] : ", tuple3)
11
```

Gambar 4.35 Mengakses Isi Tuple

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :

```
▶ Isi tuple[1] : ('Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020)
⚙ Isi tuple[1:3] : (1, 2, 3, 4, 5)
⚙ Isi tuple[3:] : ('a', 'b', 'c', 'd', 'e')
```

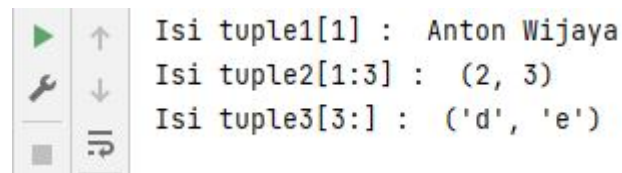
Gambar 4.36 Hasil Keluaran Program

Untuk mengakses elemen nilai dalam Tuple, gunakanlah tanda kurung () dengan memberikan nilai indeks sesuai elemen yang akan diambil.

```
1  ▶
2
3  tuple1 = ('Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020)
4  tuple2 = (1, 2, 3, 4, 5)
5  tuple3 = ("a", "b", "c", "d", "e")
6
7  print ("Isi tuple1[1] : ", tuple1[1])
8  print ("Isi tuple2[1:3] : ", tuple2[1:3])
9  print ("Isi tuple3[3:] : ", tuple3[3:])
10
11
```

Gambar 4.37 Mengakses Isi Tuple

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :



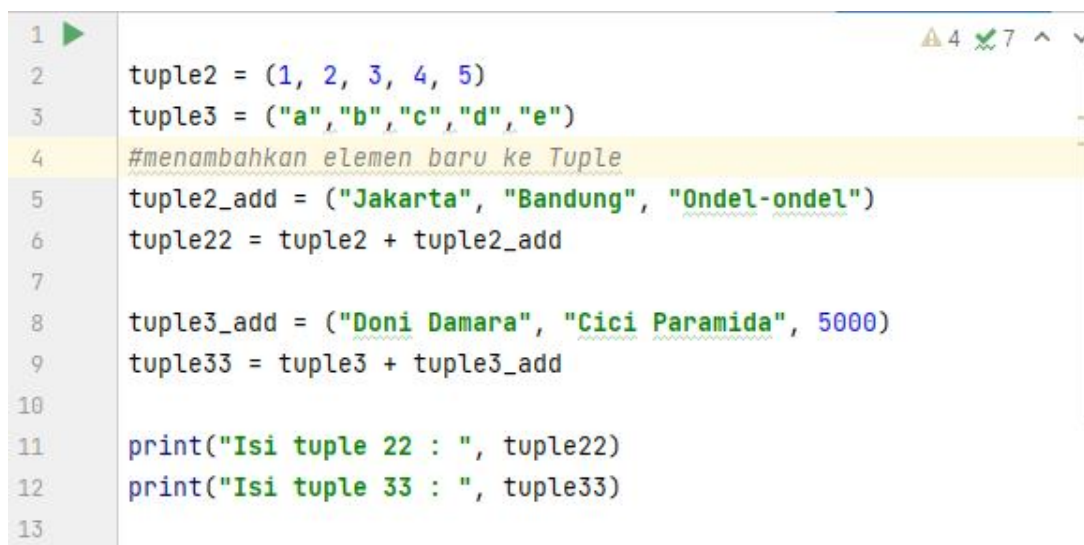
```
Isi tuple1[1] : Anton Wijaya
Isi tuple2[1:3] : (2, 3)
Isi tuple3[3:] : ('d', 'e')
```

Gambar 4.38 Hasil Keluaran Program

b. Menambahkan Nilai pada Tuple

Tuple memiliki sifat immutable, yaitu object atau tipe variabel yang telah dibuat tidak dapat diubah (read only). Untuk menambahkan elemen nilai ke Tuple, maka anda dapat membuat object baru dan menambahkan object baru ke Tuple yang akan ditambahkan dengan menggunakan operator penggabungan (+).

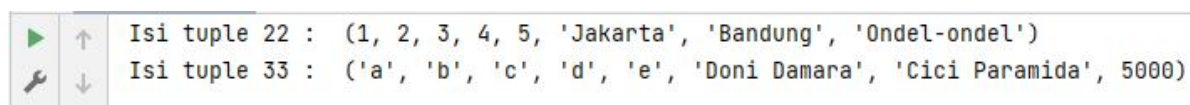
Berikut ini contoh menambahkan elemen nilai baru ke Tuple dengan menggunakan object baru dapat dilihat pada gambar 4.39.



```
1 tuple2 = (1, 2, 3, 4, 5)
2 tuple3 = ("a", "b", "c", "d", "e")
3 #menambahkan elemen baru ke Tuple
4 tuple2_add = ("Jakarta", "Bandung", "Ondel-ondel")
5 tuple22 = tuple2 + tuple2_add
6
7
8 tuple3_add = ("Doni Damara", "Cici Paramida", 5000)
9 tuple33 = tuple3 + tuple3_add
10
11 print("Isi tuple 22 : ", tuple22)
12 print("Isi tuple 33 : ", tuple33)
13
```

Gambar 4.39 Menambahkan Elemen pada Tuple menggunakan object baru

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :



```
Isi tuple 22 : (1, 2, 3, 4, 5, 'Jakarta', 'Bandung', 'Ondel-ondel')
Isi tuple 33 : ('a', 'b', 'c', 'd', 'e', 'Doni Damara', 'Cici Paramida', 5000)
```

Gambar 4.40 Hasil Keluaran Program

Berikut ini contoh menambahkan elemen nilai baru ke Tuple menggunakan fungsi **Extend()** dapat dilihat pada gambar 4.11.

```
1  ▶ tuple2 = (1, 2, 3, 4, 5)
2
3  tuple3 = ("a", "b", "c", "d", "e")
4
5  #mengubah tipe Tuple menjadi List
6  tuple22 = list(tuple2)
7  tuple33 = list(tuple3)
8
9  tuple2_add = ["Jakarta", "Bandung", "Ondel-ondel"]
10 tuple3_add = ["Doni Damara", "Cici Paramida", 5000]
11
12 tuple22.extend(tuple2_add)
13 tuple33.extend(tuple3_add)
14
```

Gambar 4.41 Menambahkan Elemen pada Tuple menggunakan fungsi Extend()

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :

```
▶ ↑ Isi tuple 22 : [1, 2, 3, 4, 5, 'Jakarta', 'Bandung', 'Ondel-ondel']
⚙ ↓ Isi tuple 33 : ['a', 'b', 'c', 'd', 'e', 'Doni Damara', 'Cici Paramida', 5000]
```

Gambar 4.42 Hasil Keluaran Program

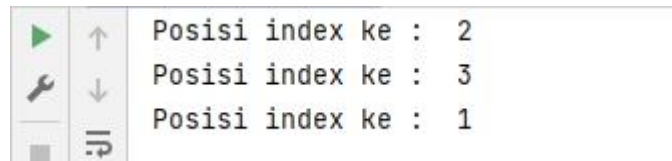
c. Penggunaan fungsi Index() pada Tuple

Untuk mengitung mengetahui posisi index sebuah elemen nilai pada Tuple, maka gunakanlah fungsi **Index()**. Sintak penulisan : **Index(object)**

```
1  ▶ tuple1 = ('Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020)
2
3  tuple2 = (1, 2, 3, 4, 5)
4
5  tuple3 = ("a", "b", "c", "d", "e")
6  #mencari posisi index pada tuple
7  print ("Posisi index ke : ", tuple1.index("Kimia"))
8  print ("Posisi index ke : ", tuple2.index(4))
9  print ("Posisi index ke : ", tuple3.index("b"))
10
```

Gambar 4.43 Mencari posisi index pada Tuple menggunakan fungsi Index()

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :



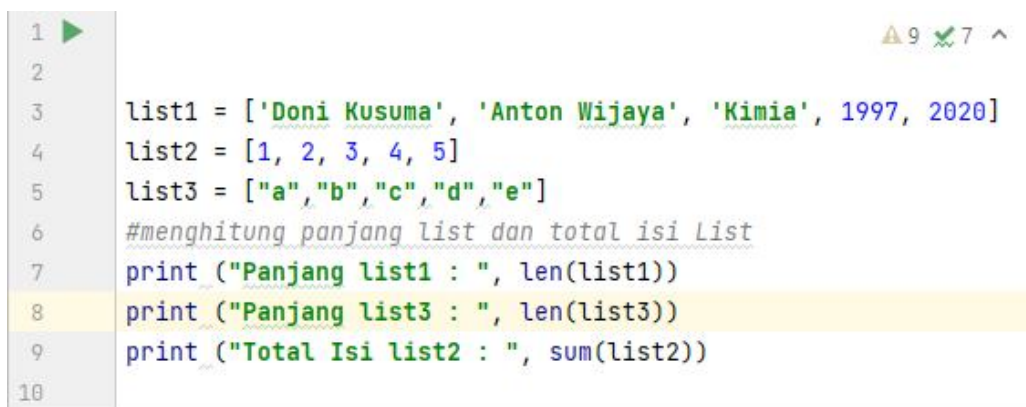
Gambar 4.44 Hasil Keluaran Program

d. Penggunaan Fungsi Len() dan Sum() pada Tuple

Untuk menghitung jumlah karakter dalam string pada Tuple, termasuk spasi dan tanda baca dan mengembalikan angka sebagai hasilnya, maka gunakanlah fungsi **Len()**. Sintak penulisan : **len(variabel_tuple)**

Sedangkan untuk menghitung jumlah seluruh item pada Tuple, gunakanlah perintah **Sum()**. Sintak penulisan : **sum(variabel_tuple)**

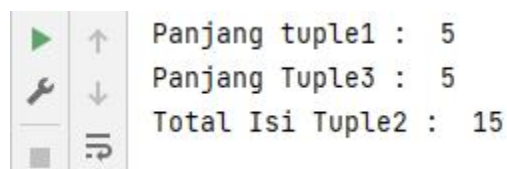
Berikut ini contoh menghitung jumlah karakter dalam string pada Tuple dapat dilihat pada gambar 4.45.



```
1  ▶
2
3  list1 = ['Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020]
4  list2 = [1, 2, 3, 4, 5]
5  list3 = ["a", "b", "c", "d", "e"]
6  #menghitung panjang list dan total isi List
7  print ("Panjang list1 : ", len(list1))
8  print ("Panjang list3 : ", len(list3))
9  print ("Total Isi list2 : ", sum(list2))
10
```

Gambar 4.45 Penggunaan fungsi Len() dan Sum() pada Tuple

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :



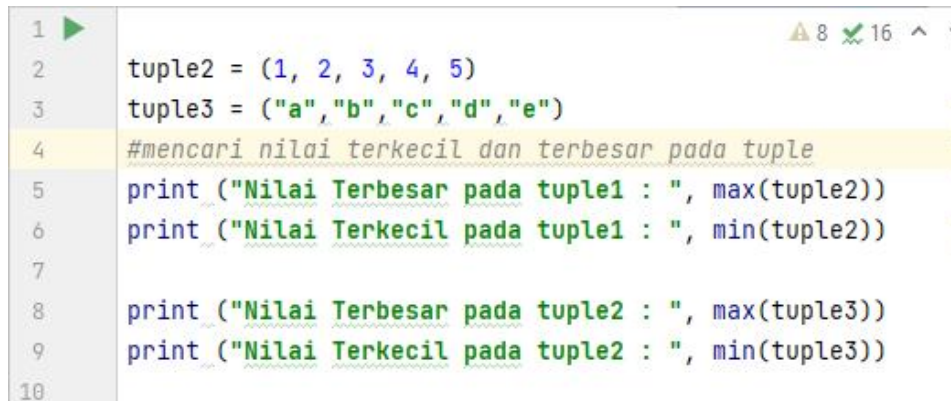
Gambar 4.46 Hasil Keluaran Program

e. Penggunaan Fungsi Max() dan Min() pada Tuple

Untuk mengetahui nilai terbesar dalam elemen Tuple, maka gunakanlah fungsi **Max()**, Sintak penulisan : **Max(variabel_tuple)**.

Sedangkan untuk mengetahui nilai terkecil dalam elemen Tuple, maka gunakanlah fungsi **Min()**, Sintak penulisan : **Min(variabel_tuple)**.

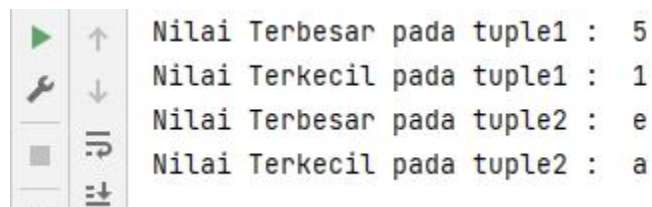
Berikut ini contoh penggunaan fungsi **Max()** dan **Min()** pada Tuple dapat dilihat pada gambar 4.47.



```
1  ▶
2  tuple2 = (1, 2, 3, 4, 5)
3  tuple3 = ("a", "b", "c", "d", "e")
4  #mencari nilai terkecil dan terbesar pada tuple
5  print ("Nilai Terbesar pada tuple1 : ", max(tuple2))
6  print ("Nilai Terkecil pada tuple1 : ", min(tuple2))
7
8  print ("Nilai Terbesar pada tuple2 : ", max(tuple3))
9  print ("Nilai Terkecil pada tuple2 : ", min(tuple3))
10
```

Gambar 4.47 Penggunaan fungsi Max() dan Min() pada Tuple

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :



```
▶ ↑ Nilai Terbesar pada tuple1 : 5
⚙ ↓ Nilai Terkecil pada tuple1 : 1
□ ↻ Nilai Terbesar pada tuple2 : e
⚙ ↻ Nilai Terkecil pada tuple2 : a
```

Gambar 4.48 Hasil Keluaran Program

a. Penggunaan Fungsi Any() dan All() pada Tuple

Fungsi **Any()** : Jika salah satu item dalam tuple bernilai Boolean True, maka fungsi ini mengembalikan True. Jika tidak, ia mengembalikan False.

Sintak penulisan : **Any(variabel_tuple)**.

```

1  ▶ #semua elemen di tuple bernilai true
2  tuple1 = (1, 2, 3, 4)
3  print("Any ke-1 bernilai : ", any(tuple1))
4  #semua elemen di tuple bernilai false
5  tuple1 = [0, False]
6  print("Any ke-2 bernilai : ", any(tuple1))
7  #beberapa elemen di list bernilai true,
8  #jika elemen lain bernilai false
9  tuple1 = [0, False, 5]
10 print("Any ke-3 bernilai : ", any(tuple1))
11 #variabel tuple kosong bernilai false
12 tuple1 = {}
13 print("Any ke-4 bernilai : ", any(tuple1))
14
15

```

Gambar 4.49 Penggunaan fungsi Any() pada Tuple

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :

```

▶ ↑ Any ke-1 bernilai : True
⚙ ↓ Any ke-2 bernilai : False
■ ↕ Any ke-3 bernilai : True
⏮ ⏭ Any ke-4 bernilai : False

```

Gambar 4.50 Hasil Keluaran Program

Fungsi **All()** : Tidak seperti **any()**, **all()** mengembalikan True hanya jika semua item bernilai Boolean True. Jika tidak, ia mengembalikan False.

Sintak penulisan : **All(variabel_tuple)**.

```

1  ▶ #semua elemen di tuple bernilai true
2  tuple1 = (1, 2, 3, 4)
3  print("All ke-1 bernilai : ", all(tuple1))
4  #semua elemen di tuple bernilai false
5  tuple1 = [0, False]
6  print("All ke-2 bernilai : ", all(tuple1))
7  #semua elemen di list bernilai false
8  tuple1 = [0, False, 5]
9  print("All ke-3 bernilai : ", all(tuple1))
10 #variabel tuple kosong bernilai false
11 tuple1 = {}
12 print("All ke-4 bernilai : ", all(tuple1))
13
14

```

Gambar 4.51 Penggunaan fungsi All() pada Tuple

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :

```

▶ ↑ All ke-1 bernilai : True
⚙ ↓ All ke-2 bernilai : False
⏏ ↕ All ke-3 bernilai : False
⏏ ⇅ All ke-4 bernilai : True

```

Gambar 4.52 Hasil Keluaran Program

3.4 Manipulasi Elemen Nilai pada Dictionary

a. Membuat dan Mengakses Nilai Pada Dictionary

Berikut ini adalah contoh pembuatan variabel yang bertipe Dictionary, yaitu : mydict.

```

1  ▶ mydict = {
2  'Asia': {'86': 'China', '62': 'Indonesia', '966': 'Saudi Arab'},
3  'America': {'1': 'USA', '58': 'Venezuela', '51': 'Peru'},
4  'Africa': {'20': 'Mesir', '251': 'Ethiopia', '249': 'Sudan'}}
5
6

```

Gambar 4.53 Mendeklarasikan Dictionary

Ada beberapa cara untuk mengakses elemen nilai dalam Dictionary, yaitu : mengakses langsung ke **key name**, menggunakan fungsi **get()** atau menggunakan fungsi **setdefault()**.

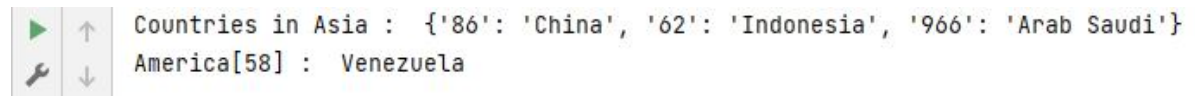
Berikut ini adalah contoh pengaksesan elemen nilai pada Dictionary menggunakan **Key Name** :



```
1 ▶ mydict = {
2   'Asia': {'86': 'China', '62': 'Indonesia', '966': 'Saudi Arab'},
3   'America': {'1': 'USA', '58': 'Venezuela', '51': 'Peru'},
4   'Africa': {'20': 'Mesir', '251': 'Ethiopia', '249': 'Sudan'}}
5
6 #mengakses nilai dalam Dictionary
7 print('Countries in Asia : ', mydict["Asia"])
8 print('America[58] : ', mydict["America"]["58"])
9
```

Gambar 4.54 Mengakses Elemen Dictionary dengan Key Name

Berikut ini adalah hasil keluaran (output) setelah kode program diatas dijalankan :



```
▶ ↑ Countries in Asia : {'86': 'China', '62': 'Indonesia', '966': 'Arab Saudi'}
⚙ ↓ America[58] : Venezuela
```

Gambar 4.55 Hasil Keluaran Program

b. Mengubah Elemen Nilai pada Dictionary

Untuk mengubah nilai item yang ada, maka gunakanlah operator penugasan. Jika kunci (key name) sudah ada, maka nilai yang ada dapat diperbarui. Jika kunci (key name) tidak ada, elemen baru (kunci: nilai) ditambahkan ke kamus.


```

1  ▶
2  mydict = {
3      'Asia': {'86': 'China', '62': 'Indonesia', '966': 'Saudi Arab'},
4      'America': {'1': 'USA', '58': 'Venezuela', '51': 'Peru'},
5      'Africa': {'20': 'Mesir', '251': 'Ethiopia', '249': 'Sudan'}}
6      #mengubah nilai dalam Dictionary
7      mydict["Asia"]["966"] = "Arab Saudi"
8      mydict["America"]["1"] = "United States of America"
9
10     print(['"Asia"']['966'] : ', mydict["Asia"]["966"])
11     print(['"America"']['1'] : ', mydict["America"]["1"])
12

```

Gambar 4.56 Menambahkan Elemen ke Dictionary

Berikut ini adalah hasil keluaran (output) setelah kode program diatas dijalankan :

```

▶ ↑ ["Asia"]["966"] : Arab Saudi
⚙ ↓ ["America"]["1"] : United States of America

```

Gambar 4.57 Hasil Keluaran Program

Berikut ini adalah kode program untuk mengupdate isi elemen pada Dictionary.

```

1  ▶
2  mydict = {
3      'Asia': {'86': 'China', '62': 'Indonesia', '966': 'Saudi Arab'},
4      'America': {'1': 'USA', '58': 'Venezuela', '51': 'Peru'},
5      'Africa': {'20': 'Mesir', '251': 'Ethiopia', '249': 'Sudan'}}
6
7  mydict.update([("Asia", "SEA GAMES")])
8  print("my dict : ", mydict)
9

```

Gambar 4.58 Mengubah Isi Elemen Dictionary

Berikut ini adalah hasil keluaran (output) setelah kode program diatas dijalankan :

```
my dict : {'Asia': 'SEA GAMES', 'America': {'1': 'USA', '58': 'Vene
```

Gambar 4.59 Hasil Keluaran Program

c. Menambah Elemen Nilai pada Dictionary

Untuk menambahkan item baru, maka gunakanlah operator penugasan. Jika kunci (key name) sudah ada, maka nilai yang ada dapat diperbarui. Jika kunci (key name) tidak ada, elemen baru (kunci: nilai) ditambahkan ke kamus.

```
1  ▶
2  mydict = {
3      'Asia': {'86': 'China', '62': 'Indonesia', '966': 'Saudi Arab'},
4      'America': {'1': 'USA', '58': 'Venezuela', '51': 'Peru'},
5      'Africa': {'20': 'Mesir', '251': 'Ethiopia', '249': 'Sudan'}}
6  #menambahkan elemen baru dalam Dictionary
7  mydict["Asia"]["93"] = "Afganistan"
8  mydict["America"]["52"] = "Mexico"
9
10 print('Countries in Asia : ', mydict["Asia"])
11 print('Countries in America : ', mydict["America"])
12
```

Gambar 4.60 Menambahkan Elemen ke Dictionary

Berikut ini adalah hasil keluaran (output) setelah kode program diatas dijalankan :

```
Countries in Asia : {'86': 'China', '62': 'Indonesia', '966': 'Saudi Arab', '93': 'Afganistan'}
Countries in America : {'1': 'USA', '58': 'Venezuela', '51': 'Peru', '52': 'Mexico'}
```

Gambar 4.61 Hasil Keluaran Program

d. Menghapus Elemen Nilai pada Dictionary

Untuk menghapus satu elemen nilai pada Dictionary, maka gunakan fungsi Pop() disertai dengan menyebutkan nilai pada kunci (**key value**).

Sintak penulisan : **Pop(key value)**.

```

1  ▶
2  mydict = {
3      'Asia': {'86': 'China', '62': 'Indonesia', '966': 'Saudi Arab'},
4      'America': {'1': 'USA', '58': 'Venezuela', '51': 'Peru'},
5      'Africa': {'20': 'Mesir', '251': 'Ethiopia', '249': 'Sudan'}}
6      #Menghapus elemen dalam Dictionary
7      print(mydict["Asia"].pop('966'))
8      print(mydict["America"].pop("58"))
9
10     print('Countries in Asia : ', mydict["Asia"])
11     print('Countries in America : ', mydict["America"])
12

```

Gambar 4.62 Menghapus Elemen pada Dictionary

Berikut ini adalah hasil keluaran (output) setelah kode program diatas dijalankan :

```

▶ ↑ Saudi Arab
⚙ ↓ Venezuela
— — Countries in Asia : {'86': 'China', '62': 'Indonesia'}
— — Countries in America : {'1': 'USA', '51': 'Peru'}
— —

```

Gambar 4.63 Hasil Keluaran Program

Kita juga bisa menghapus elemen nilai terakhir pada Dictionary, menggunakan fungsi Popitem() disertai dengan menyebutkan pasangan kunci dan nilainya **(key, value)**.

Sintak penulisan : **Popitem(key, value)**.


```

1  ▶
2  mydict = {
3      'Asia': {'86': 'China', '62': 'Indonesia', '966': 'Saudi Arab'},
4      'America': {'1': 'USA', '58': 'Venezuela', '51': 'Peru'},
5      'Africa': {'20': 'Mesir', '251': 'Ethiopia', '249': 'Sudan'}}
6      #Menghapus elemen dalam Dictionary
7      print(mydict["Asia"].popitem())
8      print(mydict["America"].popitem())
9
10     print('Countries in Asia : ', mydict["Asia"])
11     print('Countries in America : ', mydict["America"])
12

```

Gambar 4.64 Menghapus Elemen pada Dictionary

Berikut ini adalah hasil keluaran (output) setelah kode program diatas dijalankan :

```

▶ ↑ ('966', 'Saudi Arab')
⚙ ↓ ('51', 'Peru')
🔍 Countries in Asia : {'86': 'China', '62': 'Indonesia'}
🔍 Countries in America : {'1': 'USA', '58': 'Venezuela'}

```

Gambar 4.65 Hasil Keluaran Program

Kita juga menghapus seluruh elemen nilai pada Dictionary, maka gunakan fungsi **clear()**.

Sintak penulisan : **variabel_Dictionary.clear()**.

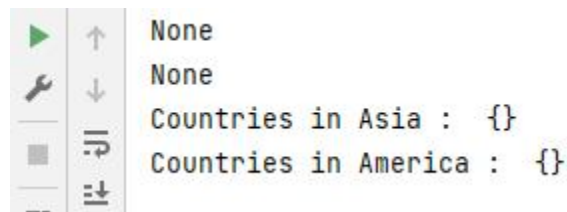
```

1  ▶
2  mydict = {
3      'Asia': {'86': 'China', '62': 'Indonesia', '966': 'Saudi Arab'},
4      'America': {'1': 'USA', '58': 'Venezuela', '51': 'Peru'},
5      'Africa': {'20': 'Mesir', '251': 'Ethiopia', '249': 'Sudan'}}
6      #Menghapus seluruh elemen dalam Dictionary
7      print(mydict["Asia"].clear())
8      print(mydict["America"].clear())
9
10     print('Countries in Asia : ', mydict["Asia"])
11     print('Countries in America : ', mydict["America"])
12

```

Gambar 4.66 Menghapus Elemen pada Dictionary

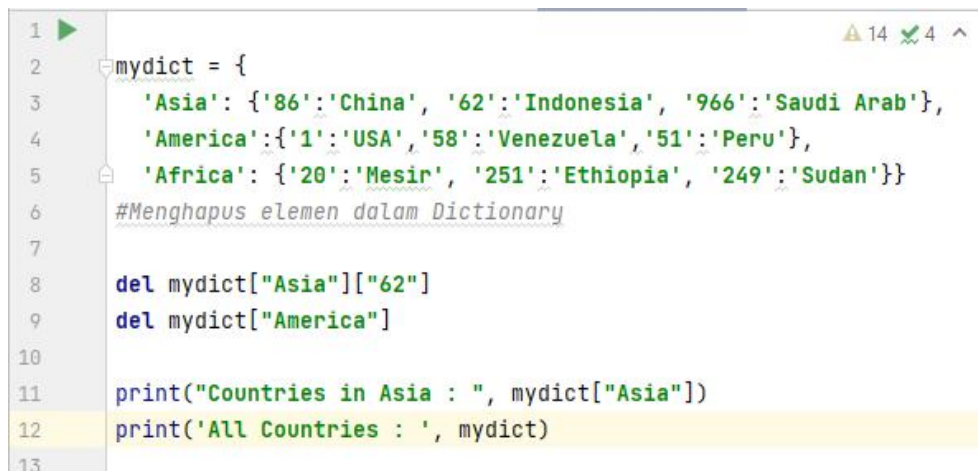
Berikut ini adalah hasil keluaran (output) setelah kode program diatas dijalankan :



```
None
None
Countries in Asia : {}
Countries in America : {}
```

Gambar 4.67 Menghapus Elemen pada Dictionary

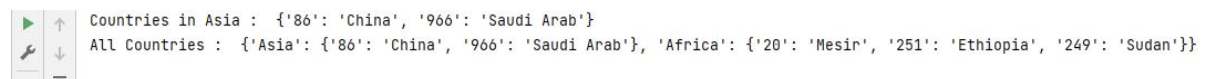
Kita juga menghapus seluruh elemen nilai pada Dictionary, maka gunakan perintah **del**.



```
1 mydict = {
2     'Asia': {'86': 'China', '62': 'Indonesia', '966': 'Saudi Arab'},
3     'America': {'1': 'USA', '58': 'Venezuela', '51': 'Peru'},
4     'Africa': {'20': 'Mesir', '251': 'Ethiopia', '249': 'Sudan'}}
5 #Menghapus elemen dalam Dictionary
6
7
8 del mydict["Asia"]["62"]
9 del mydict["America"]
10
11 print("Countries in Asia : ", mydict["Asia"])
12 print('All Countries : ', mydict)
13
```

Gambar 4.68 Menghapus Elemen pada Dictionary

Berikut ini adalah hasil keluaran (output) setelah kode program diatas dijalankan :



```
Countries in Asia : {'86': 'China', '966': 'Saudi Arab'}
All Countries : {'Asia': {'86': 'China', '966': 'Saudi Arab'}, 'Africa': {'20': 'Mesir', '251': 'Ethiopia', '249': 'Sudan'}}
```

Gambar 4.69 Hasil Keluaran Program

e. Menyalin Elemen pada Dictionary

Fungsi **Copy()** : digunakan untuk menyalin isi dari variabel Dictionary asal ke variabel Dictionary baru.

Sintak penulisan : **original_dictionary.copy()**.

```

1  ▶
2  mydict = {
3      'Asia': {'86': 'China', '62': 'Indonesia', '966': 'Saudi Arab'},
4      'America': {'1': 'USA', '58': 'Venezuela', '51': 'Peru'},
5      'Africa': {'20': 'Mesir', '251': 'Ethiopia', '249': 'Sudan'}}
6      #menyalin elemen dalam variabel Dict ke variabel Dict yg lain
7      mydict_copy = mydict.copy()
8
9      print('mydict: ', mydict)
10     print('mydict_copy: ', mydict_copy)
11

```

Gambar 4.70 Menyalin Elemen pada Dictionary

Berikut ini adalah hasil keluaran (output) setelah kode program diatas dijalankan :

```

▶ ↑ mydict: {'Asia': {'86': 'China', '62': 'Indonesia', '966': 'Saudi Arab'}, 'America': {'1': 'USA', '58': 'Venezuela', '51': 'Peru'}, 'Africa': {'20': 'Mesir', '251': 'Ethiopia', '249': 'Sudan'}}
⚙ ↓ mydict_copy: {'Asia': {'86': 'China', '62': 'Indonesia', '966': 'Saudi Arab'}, 'America': {'1': 'USA', '58': 'Venezuela', '51': 'Peru'}, 'Africa': {'20': 'Mesir', '251': 'Ethiopia', '249': 'Sudan'}}
=

```

Gambar 4.71 Hasil Keluaran Program

j. Penggunaan Fungsi Len() dan Sum() pada Dictionary

Untuk menghitung panjang (length) sebuah Dictionary, gunakanlah fungsi **len()**.

Sintak penulisan : **len(variabel_dictionary)**

Untuk menjumlahkan elemen pada Dictionary, maka gunakanlah fungsi **sum()**.

Sintak penulisan : **sum(variabel_dictionary)**

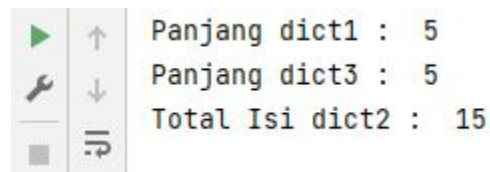
```

1  ▶
2
3      dict1 = ['Doni Kusuma', 'Anton Wijaya', 'Kimia', 1997, 2020]
4      dict2 = [1, 2, 3, 4, 5]
5      dict3 = ["a", "b", "c", "d", "e"]
6      #menghitung panjang dan total elemen pada Dictionary
7      print ("Panjang dict1 : ", len(dict1))
8      print ("Panjang dict3 : ", len(dict3))
9      print ("Total Isi dict2 : ", sum(dict2))
10

```

Gambar 4.72 Penggunaan fungsi len() dan sum() pada Dictionary

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :



```
Panjang dict1 : 5
Panjang dict3 : 5
Total Isi dict2 : 15
```

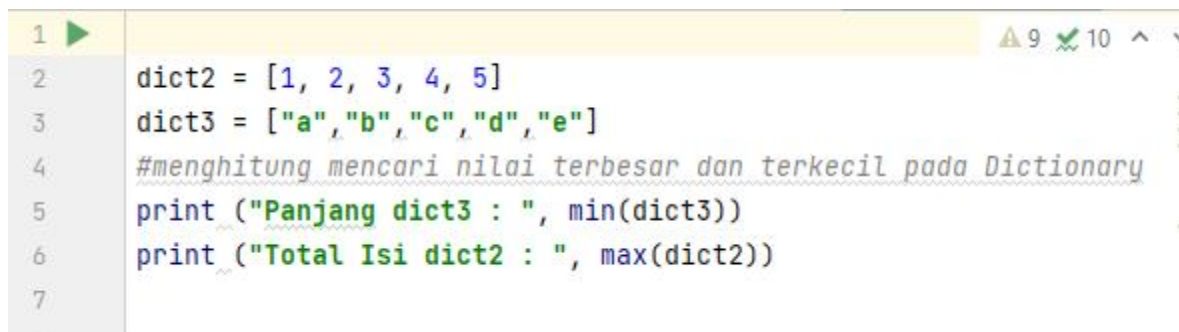
Gambar 4.73 Hasil Keluaran Program

k. Penggunaan Fungsi Max() dan Min() pada Dictionary

Untuk mengetahui nilai terbesar dalam elemen Dictionary, maka gunakanlah fungsi **Max()**, Sintak penulisan : **Max(variabel_dictionary)**.

Sedangkan untuk mengetahui nilai terkecil dalam elemen Dictionary, maka gunakanlah fungsi **Min()**, Sintak penulisan : **Min(variabel_dictionary)**.

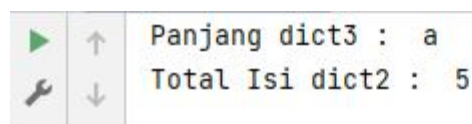
Berikut ini contoh penggunaan fungsi **Max()** dan **Min()** pada Dictionary dapat dilihat pada gambar 4.74.



```
1 dict2 = [1, 2, 3, 4, 5]
2 dict3 = ["a", "b", "c", "d", "e"]
3 #menghitung mencari nilai terbesar dan terkecil pada Dictionary
4 print ("Panjang dict3 : ", min(dict3))
5 print ("Total Isi dict2 : ", max(dict2))
```

Gambar 4.74 Penggunaan fungsi Max() dan Min() pada Dictionary

Setelah kode program diatas dijalankan, maka hasil keluaran yang ditampilkan sebagai berikut :



```
Panjang dict3 : a
Total Isi dict2 : 5
```

Gambar 4.75 Hasil Keluaran Program

l. Penggunaan Fungsi Any() dan All() pada Dictionary

Fungsi **Any()** : Jika salah satu item dalam tuple bernilai Boolean True, maka fungsi ini mengembalikan True. Jika tidak, ia mengembalikan False.

Sintak penulisan : **Any(variabel_Dictionary)**.

```

1  ▶
2  #semua key bernilai True
3  mydict = {1 : 'Jakarta', 2: 'Medan'}
4  print("Any ke-1 bernilai : ", any(mydict))
5  #semua key bernilai False
6  mydict = {0 : 'Jakarta', False: 'Medan'}
7  print("Any ke-2 bernilai : ", any(mydict))
8  #satu key bernilai True, '0' bernilai tidak bernilai false
9  #karena '0' adalah format string
10 mydict = {0 : 'Jakarta', 1: 0}
11 print("Any ke-3 bernilai : ", any(mydict))
12 #variabel Dict kosong bernilai false
13 mydict = {}
14 print("Any ke-4 bernilai : ", any(mydict))
15

```

Gambar 4.76 Penggunaan fungsi Any pada Dictionary

Berikut ini adalah hasil keluaran (output) setelah kode program diatas dijalankan :

```

▶ ↑ Any ke-1 bernilai : True
⚙ ↓ Any ke-2 bernilai : False
□ ⏏ Any ke-3 bernilai : True
⏏ ⏏ Any ke-4 bernilai : False

```

Gambar 4.77 Hasil Keluaran Program

Fungsi **All()** : Tidak seperti **any()**, **all()** mengembalikan True hanya jika semua item bernilai Boolean True. Jika tidak, ia mengembalikan False.

Sintak penulisan : **All(variabel_dictionary)**.


```

1  ▶
2  #semua key bernilai True
3  mydict = {1 : 'Jakarta', 2: 'Medan'}
4  print("All ke-1 bernilai : ", all(mydict))
5  #semua key bernilai False
6  mydict = {0 : 'Jakarta', False: 'Medan'}
7  print("All ke-2 bernilai : ", all(mydict))
8  #semua bernilai False
9  mydict = {0 : 'Jakarta', 1: 0}
10 print("All ke-3 bernilai : ", all(mydict))
11 #variabel dict kosong, bernilai true
12 mydict = {}
13 print("All ke-4 bernilai : ", all(mydict))
14

```

Gambar 4.78 Penggunaan fungsi All pada Dictionary

Berikut ini adalah hasil keluaran (output) setelah kode program diatas dijalankan :

```

▶ ↑ All ke-1 bernilai : True
⚙ ↓ All ke-2 bernilai : False
☐ ≡ All ke-3 bernilai : False
⇅ ⇅ All ke-4 bernilai : True

```

Gambar 4.79 Hasil Keluaran Program

Praktikum

Langkah-langkah Praktikum

1. Buka Editor Python (IDLE / Pycharm / VSCode).
2. Buatlah file baru dengan membuka menu File > New > Source File atau dengan shortcut Ctrl + N.
3. Tulislah kode program berikut ini :

Program 4.1 : List4. Py

1. Tuliskan Program 4.1 berikut ini pada editor :

```

2 my_list = ["John", "Bill", "Maria", "Jenny", "Jack"]
3
4 my_list1 = [1, "Monica"]
5 my_list.append(my_list1)
6 print("my_list 1 : ", my_list)
7

```

Gambar 4.80 Kode Program 4.1

2. Simpan Program ini dengan nama List4.py
3. Jalankan program 4.1 di atas, lalu tuliskan apa yang tercetak di layar.

4. Tambahkan kode program berikut ini pada program 4.1 :

```

8 my_list2 = [1, "Monica"]
9 my_list.extend(my_list2)
10 print("my_list 2 : ", my_list)
11

```

Gambar 4.81 Penambahan Kode Program

5. Jalankan program 4.1 di atas, lalu tuliskan apa yang tercetak di layar.

6. Apakah hasil keluaran program di nomor 3 dan nomor 5 memiliki hasil yang sama, jelaskan!.
7. Tambahkan kode program berikut ini pada program 4.1 :

```

12 del my_list
13 print("my_list : ", my_list)
14

```

Gambar 4.82 Penambahan Kode Program

8. Berdasarkan program 4.2 nomor 7 diatas, apakah ada yang kurang atau tidak tampil dari kode program tersebut? Jelaskan!

Program 4.2 : Tuple4. Py

1. Tuliskan kode program 4.2 berikut ini pada editor :

```
2  
3 my_tuple = (4, 2, 3, 6, 5, 9, 8, 10, 11)  
4 my_tuple[1] = 9  
5 print(my_tuple)  
6
```

Gambar 4.83 Kode Program 4.2

2. Simpan kode program 4.2 dengan nama Tuple2.py
3. Jalankan kode program 4.2 di atas, lalu tuliskan apa yang tercetak di layar.

Berdasarkan program 4.2, apakah proses merubah elemen nilai pada indeks ke-1 berhasil atau tidak? Jelaskan!

4. Tambahkan kode program berikut ini pada program 4.2 : :

```
2  
3 my_tuple = (4, 2, 3, 6, 5, 9, 8, 10, 11)  
4  
5 my_tuple = ('b', 'u', 'd', 'i', 'l', 'u', 'h', 'u', 'r')  
6 print(my_tuple)  
7
```

Gambar 4.84 Kode Program 4.1

5. Jalankan Kembali program 4.2 di atas, lalu tuliskan apa yang tercetak di layar.

6. Tambahkan kode program berikut ini pada program 4.2 :

```
5  
6 my_tuple = my_tuple + (1, 9, 7, 9)  
7 print(my_tuple)  
8
```

Gambar 4.85 Penambahan Kode Program

7. Jalankan Kembali kode program 4.2 di atas, lalu tuliskan apa yang tercetak di layar.

8. Tambahkan kode program berikut ini pada program 4.2 :

```
5  
6 del my_tuple  
7 print(my_tuple)  
8
```

Gambar 4.86 Penambahan Kode Program

9. Jalankan Kembali kode program 4.2 di atas, lalu tuliskan apa yang tercetak di layar.

Program 4.3 : Dictionary4. Py

1. Tuliskan program 4.3 berikut ini pada editor :

```
my_dict = {"student":{"username": "Ahsan", "email": "ahsan@gmail.com", "usia" : 25}}  
print("my dict : ", my_dict)
```

Gambar 4.87 Kode Program 4.3

2. Simpan program 4.3 dengan nama Dictionary4.py
3. Jalankan program 4.3 di atas lalu tuliskan apa yang tercetak di layar.

4. Tambahkan kode program erikut ini pada program 4.3 :

```
my_dict["student"]["id"] = 535  
print("my dict : ", my_dict)
```

Gambar 4.87 Menambahkan Kode Program

5. Jalankan program 4.3 di atas lalu tuliskan apa yang tercetak di layar.

6. Tambahkan kode program 4.2 berikut ini pada editor :

```
my_dict.update({"address": "Jakarta"})  
print("my dict : ", my_dict)
```

Gambar 4.88 Menambahkan Kode Program

7. Jalankan program 4.3 di atas lalu tuliskan apa yang tercetak di layar.

8. Tambahkan kode program berikut ini pada program 4.3 :

```
del my_dict  
print("my dict : ", my_dict)
```

Gambar 4.89 Penambahan Kode Program

9. Jalankan program 4.3 di atas, lalu tuliskan apa yang tercetak di layar.

Rangkuman

1. List dan Dictionary bersifat mutable, dimana variabel yang dibuat menggunakan kedua tipe data tersebut masih dapat diubah.
2. Tuple bersifat immutable, dimana variabel yang dibuat menggunakan kedua tipe tersebut tidak dapat diubah.
3. Manipulasi isi elemen pada List, Tuple dan Dictionary dapat dilakukan menggunakan perintah dan fungsi-fungsi yang terdapat pada bahasa Python, seperti : del, append(), remove(), dll.

Latihan

Setelah membuat program di atas (praktik operasi pada List, Tuple dan Dictionary), buatlah duplikasi dari contoh praktik sebagai berikut :

1. Buat sebuah program sederhana untuk menyimpan kenalanmu, dengan ketentuan sebagai berikut :
 - a. Isi list sebanyak 5.
 - b. Tampilkan isi list hanya khusus untuk indeks nomor 2 dan 4.
 - c. Tampilkan semua teman anda.
 - d. Tampilkan panjang list.

sehingga akan menghasilkan keluaran program sebagai berikut :

```
Isi my_friends indeks ke-2 adalah: Agung
Isi my_friends indeks ke-4 adalah: Adam
Semua teman berjumlah : 5
Semua teman itu adalah : ['Anggun', 'Dian', 'Agung', 'Adi', 'Adam']
```

Gambar 4.90 Tampilan keluaran program

2. Buatlah program menggunakan python dengan mengganti tipe data pada soal tugas 1 menjadi bertipe Tuple sehingga akan menghasilkan keluaran program sebagai berikut :

```
Isi my_friends indeks ke-2 adalah: Agung
Isi my_friends indeks ke-4 adalah: Adam
Semua teman berjumlah : 5
Semua teman itu adalah : ['Anggun', 'Dian', 'Agung', 'Adi', 'Adam']
```

Gambar 4.91 Tampilan keluaran program

3. Buatlah program sederhana menggunakan tipe Dictionary python untuk merekam biodata mahasiswa yang terdiri atas :

Nama : Rudi Hartono
Umum : 22
Email : rudi.hartono@gmail.com
Hobi : Koding, Membaca, Travelling, Kulier
Menikah : False
Email : rudi.hartono@gmail.com
Sosmed : untuk "Facebook" : "RudiHartono", untuk "Twitter" : @RudiH
Sehingga menampilkan informasi sebagai berikut :

```
Nama saya adalah Rudi Hartono
Email saya adalah rudi.hartono@gmail.com
Hobi saya adalah ['Koding', 'membaca', 'Travelling']
Twitter: @rudih
```

Gambar 4.92 Tampilan keluaran program

Tugas Mandiri

Kerjakan soal-soal berikut :

1. Jelaskan karakteristik yang dimiliki oleh List, Tuple dan Dictionary.
2. Tuple memiliki sifat immutable, bagaimana caranya agar dapat menambahkan isi elemen pada variabel yang bertipe Tuple? Jelaskan!
3. Buatlah program dengan Python menggunakan List, Tuple dan Dictionary untuk mencetak nama-nama student sebagai berikut :

ID	Nama Mahasiswa	Telepon
1	Bambang	081230305656
2	Richard	087123455777
3	Budi	087809298787
4	Kelvin	085678787878
5	Meli	021393900097

Berikut ini adalah data yang ditambahkan pada Dictionary diatas.

NIM	Nama Mahasiswa	Telepon
6	Bambang	081230305656
7	Richard	087123455777



FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR

Jl. Raya Ciledug, Petukangan Utara, Pesanggrahan
Jakarta Selatan, 12260
Telp: 021-5853753 Fax : 021-5853752
<http://fti.budiluhur.ac.id>