Set & Dictionary

Set

salah satu tipe data bawaan yang digunakan untuk menyimpan sekumpulan elemen unik (tidak boleh duplikat) dan bersifat tidak berurutan (unordered).

6.1.1 Ciri-ciri Set

1. **Unordered** → data tidak punya indeks tetap (tidak bisa diakses dengan set[0]).
2. **Unik** → otomatis menghapus elemen duplikat.
3. **Mutable** → bisa ditambah atau dihapus.
4. **Bisa dipakai untuk operasi himpunan** → union, intersection, difference, dsb.
5. **Hanya bisa berisi data yang immutable** → angka, string, tuple (tidak bisa list atau dictionary).

Contoh Penggunaan

```
# Membuat set
buah = {"apel", "jeruk", "mangga", "apel"}

print(buah)
```

6.1.2 Pengoperasian Set

Akses Elemen

```
angka_ganjil = {1, 3, 5, 7, 9}

for angka in angka_ganjil:
    print(angka)
```

Terdapat beberapa cara pengoperasian set yaitu :

1. Menambahkan elemen menggunakan metode **add()**.

2. Menghapus elemen menggunakan metode **remove()** atau **discard()**, perbedaan dari kedua ini adalah **remove()** akan menghasilkan error jika elemen yang ingin dihapus tidak ditemukan, sedangkan **discard()** tidak.
3. Menggabungkan set bisa dengan beberapa cara yaitu :
    a. metode **union()** menggabungkan semua elemen dari set bersangkutan dan set yang lain diberikan, tanpa duplikasi.
    b. **update()** sama seperti **union()** tetapi bedanya menggabungkan set lain kedalam set yang bersangkutan.
    c. **intersection()** menemukan elemen yang sama antara 2 set.
    d. **difference()** menemukan elemen yang ada di satu set tapi tidak ada di set lainnya.

6.1.3 Kegunaan Set

1. Menghilangkan duplikasi data.
2. Mencari elemen unik dalam dua kumpulan data.
3. Kategorisasi data.
4. Mencari elemen bersama antara 2 kumpulan data

6.1.4 Batasan Penggunaan Set

1. Elemen tidak berurutan.
2. Tidak ada indeks.
3. Hanya menampung elemen yang immutable (tidak dapat berubah).
4. Tidak ada duplikasi.

Dictionary

Dictionary adalah suatu tipe data pada python yang berfungsi untuk menyimpan kumpulan data/nilai. Berbeda dengan list, dimana list bisa dibilang terbatas untuk cara pemanggilannya, yaitu hanya dapat dipanggil menggunakan index-nya saja..

Dictionary ini sendiri sesuai kalimat asalnya yaitu kamus. Dimana dalam kamus ada sebuah Kunci/Atribut dan Nilai/Informasinya. Sehingga kita cukup menggunakan kata kunci yang dimiliki oleh suatu informasi yang diinginkan untuk memanggilnya.

6.2.1 Ciri-ciri Dictionary

1. **Unordered** → Tidak berurutan
2. **Changeable** → Bisa diubah
3. Menyimpan pasangan key : value atau kunci : nilai.
4. Key harus unik dan immutable (tidak bisa sama).
5. Value bisa apa saja.
6. Diakses dengan key.
7. Mutable

6.2.2 Deklarasi Dictionary

Cara mendeklarasi dictionary dengan menggunakan tanda kurung kurawal {} lalu di isi dengan format “key : value” atau “kunci : nilai”.

Contoh Dictionary

```
Daftar_buku = {
      "Buku1" : "Bumi Manusia",
      "Buku2" : "Laut Bercerita"
}
```

Setelah mendeklarasikan dictionary, kita dapat memanggil dictionary dengan 2 cara yaitu,

```
print(Daftar_buku["Buku1"])

print(Daftar_buku)
```

Perbedaan dari kedua output tersebut adalah, output 1 akan menampilkan value atau nilai dari dictionary Daftar_buku, sedangkan output 2 menampilkan key dan value atau kunci dan nilai dari dictionary kita. Seperti dibawah berikut ini.

```
Bumi Manusia
{'Buku1': 'Bumi Manusia', 'Buku2': 'Laut Bercerita'}
```

6.2.3 Membuat Dictionary

Kita harus mengingat beberapa hal untuk membuat dictionary, yaitu :

1. Buka dan tutup kurung kurawal.
2. Nama dictionary.
3. Key atau kunci.
4. Value atau nilai.
5. Pemisah key dan value yaitu “ : “.
6. Jika item lebih dari 1 maka pisahkan menggunakan koma “ , ”.

Contoh

```
Biodata = {
      "Nama" : "Ananda Daffa Harahap",
      "NIM" : 2409106050,
      "KRS" : ["Pemrograman Web", "Struktur Data", "Basis Data"],
      "Mahasiswa_Aktif" : True,
      "Social Media" : {
```

```
            "Instagram" : "daffahrhap"
        }
}
```

Selain itu kita dapat membuat dictionary menggunakan **dict()** dengan parameter yang sama yaitu key (kunci) dan value (nilai).

Contoh

```
List_game = {"GTA 5" : "Open World", "Valorant" : "FPS", "Honkai Star Rail"
: "RPG"}
```

6.2.4 Mengakses Item Pada Dictionary

Mengakses item pada dictionary dapat dilakukan dengan 2 cara, yaitu :

1. Kurung siku [].
2. Fungsi **get()**.

Kedua cara ini sama-sama dapat mengakses dan menampilkan item dari dictionary yang kita inginkan, namun tentu saja ada perbedaannya. Untuk menggunakan kurung siku, kita cukup memanggil dictionary yang kita inginkan dan menambahkan kurung siku setelahnya, lalu didalam kurung siku tersebut kita masukan key yang telah kita masukan tadi. Contohnya seperti **Biodata[“Nama”]**. Dan untuk menggunakan fungsi get() kita dapat melakukannya dengan cara memanggil nama dictionary dan menambahkan get() setelahnya lalu masukan key yang kita inginkan. Contohnya seperti **Biodata.get(“Nama”)**

Contoh

```
print(f"nama saya adalah {Biodata["Nama"]}")
print(f"Instagram : {Biodata['Social Media']['Instagram']}")

print(f"nama saya adalah {Biodata.get("Nama")}")
print(Biodata.get("Nama"))
```

```
nama saya adalah Ananda Daffa Harahap
Instagram : daffahrhap

nama saya adalah Ananda Daffa Harahap
Ananda Daffa Harahap
```

Walau terlihat sama-sama memanggil item dari dictionary. Namun kedua cara ini memiliki perbedaan pada fungsi get(). Yaitu jika sebuah key yang dipanggil tidak ada sama sekali pada dictionary. Maka get() dapat mengembalikan nilai none.

Contoh

```
print(Biodata.get("Nama"))
print(Biodata.get("Alamat"))
print(Biodata.get("Alamat", "Key tersebut tidak ada"))

Ananda Daffa Harahap
None
Key tersebut tidak ada
```

6.2.5 Perulangan

Akan melelahkan jika kita harus menampilkan data dengan cara seperti sebelumnya, terlebih jika datanya ada sangat banyak. Maka disini kita akan menggunakan perulangan, tetapi kita juga perlu menambahkan fungsi tambahan untuk memanggil keduanya, yaitu **Items()**. Jika tidak, maka yang akan muncul hanya key dari dictionary tersebut.

Contoh

```
Nilai = {
    "Matematika": 80,
    "B. Indonesia": 90,
```

```
    "B. Inggris": 81,
    "Kimia": 78,
    "Fisika": 80
}

# Tanpa menggunakan items()
for i in Nilai:
    print(i)

print("")  # pemisah

# Menggunakan items()
for i, j in Nilai.items():
    print(f"Nilai {i} anda adalah {j}")

# output
Matematika
B. Indonesia
B. Inggris
Kimia
Fisika

Nilai Matematika anda adalah 80
Nilai B. Indonesia anda adalah 90
Nilai B. Inggris anda adalah 81
Nilai Kimia anda adalah 78
Nilai Fisika anda adalah 80
```

6.2.6 Menambahkan Item

Untuk menambahkan item, kita dapat melakukan dengan 2 cara yaitu langsung dimasukan saja atau menggunakan fungsi **update()**.

```
Film = {
"Avenger Endgame" : "Action",
"Sherlock Holmes" : "Mystery",
"The Conjuring" : "Horror"
```

```
}
#Sebelum Ditambah
print(Film)

Film["Zombieland"] = "Comedy"
Film.update({"Hours" : "Thriller"})
#Setelah Ditambah
print(Film)

#Sebelum Ditambah
{'Avenger Endgame': 'Action', 'Sherlock Holmes': 'Mystery',
'The Conjuring': 'Horror'}

#Setelah Ditambah
{'Avenger Endgame': 'Action', 'Sherlock Holmes': 'Mystery',
'The Conjuring': 'Horror', 'Zombieland': 'Comedy', 'Hours':
'Thriller'}
```

6.2.7 Mengubah Item

Mengubah item sama seperti menambahkan item, karena sistem akan menimpa value yang ada sesuai yang kita mau. Kita hanya perlu memasukan key yang sama untuk menimpanya.

```
Film = {
"Avenger Endgame" : "Action",
"Sherlock Holmes" : "Mystery",
"The Conjuring" : "Horror"
}

#Sebelum Diubah
print(Film)
Film["Sherlock Holmes"] = "Action"
Film.update({"The Conjuring" : "Tragedy"})

#Setelah diubah
```

```
print(Film)

{'Avenger Endgame': 'Action', 'Sherlock Holmes': 'Mystery',
'The Conjuring': 'Horror'}

{'Avenger Endgame': 'Action', 'Sherlock Holmes': 'Action',
'The Conjuring': 'Tragedy'}
```

6.2.8 Menghapus Item

Untuk menghapus item kita dapat menggunakan 3 cara, yaitu :

1. **Del**.
2. **Pop()**.
3. **clear()**.

Perbedaan dari ketiga ini adalah :

1. **Del** akan menghapus secara keseluruhan tanpa ada tersisa.
2. **pop()** akan menghapus tetapi masih dapat kita lihat apa yang kita hapus dengan cara kita tampung pada sebuah variabel.
3. **clear()** akan menghapus baik itu key (kunci) maupun value (nilai).

Contoh Del

```
data = {
"Nama" : "Daffa",
"Umur" : 19,
"Jurusan" : "Informatika"
}
#Sebelum Dihapus
print(data)

del data["Nama"]

#Setelah diubah
```

```
print(data)

#memanggil data yang telah dihapus
print(data.get("Nama"))

{'Nama': 'Daffa', 'Umur': 19, 'Jurusan': 'Informatika'}

{'Umur': 19, 'Jurusan': 'Informatika'}

None
```

Contoh Pop

```
data = {
"Nama" : "Daffa",
"Umur" : 19,
"Jurusan" : "Informatika"
}

#Sebelum Dihapus
print(data)

cache = data.pop("Nama")

#Setelah dihapus
print(data)

#memanggil data yang telah dihapus pada dictionary
print(data.get("Nama"))

#memanggil data yang telah dihapus pada variabel cache
print(cache)

{'Nama': 'Daffa', 'Umur': 19, 'Jurusan': 'Informatika'}
{'Umur': 19, 'Jurusan': 'Informatika'}
None

Daffa
```

Contoh Clear

```
data = {
"Nama" : "Daffa",
"Umur" : 19,
"Jurusan" : "Informatika"
}

#Sebelum Dihapus
print(data)

data.clear()

#Setelah dihapus
print(data)

{'Nama': 'Daffa', 'Umur': 19, 'Jurusan': 'Informatika'}

{}
```

6.2.9 Beberapa Fungsi Yang Bisa Dipakai

Mengetahui Panjang Dari Dictionary

Caranya mudah yaitu menggunakan fungsi **len()**.

Contoh

```
data = {
"Nama" : "Daffa",
"Umur" : 19,
"Jurusan" : "Informatika"
}
```

```
print("Jumlah Data = ", len(data))

Jumlah Data = 3
```

Copy

Caranya mudah yaitu menggunakan fungsi **Copy()** untuk membuat salinan dari dictionary yang sudah kita buat.

Contoh

```
buku = {
      "Buku1" : "Bumi Manusia",
      "Buku2" : "Laut Bercerita"
}

pinjam = buku.copy()

print("Dictionary yang telah disalin : ", pinjam)

Dictionary yang telah disalin : {"Buku1" : "Bumi Manusia", "Buku2" : "Laut
Bercerita"}
```

Fromkeys

fungsi fromkeys() ini kita dapat membuat sebuah dictionary yang telah kita siapkan key dan valuenya pada sebuah variabel terlebih dahulu. Fromkeys juga bisa kita gunakan untuk mengubah semua value pada sebuah dictionary menjadi satu value yang kita inginkan.

Contoh

```
key = "apel", "jeruk", "mangga"

value = 1

buah = dict.fromkeys(key, value)

print(buah)

{'apel': 1, 'jeruk': 1, 'mangga': 1}
```

Keys & Value

Terkadang kita hanya ingin menampilkan keys atau valuenya saja dari sebuah dictionary. Jika kita menggunakan **items()** maka semua bagian dari dictionary akan ditampilkan, dan jika kita tidak menggunakan **items()** maka yang tampil hanyalah key nya saja. Nah disini dapat kita atasi dengan fungsi **keys()** dan **value()**. Seperti namanya sendiri, fungsi ini digunakan untuk memanggil keys atau valuenya saja.

Contoh

```
Nilai = {
"Matematika" : 80,
"B. Indonesia" : 90,
"B. Inggris" : 81,
"Kimia" : 78,
"Fisika" : 80
}

#menggunakan keys
for i in Nilai.keys():
```

```
 print(i)
print("")

#menggunakan value
for i in Nilai.values():
 print(i)

Matematika
B. Indonesia
B. Inggris
Kimia
Fisika

80
90
81
78
80
```

SetDefault

SetDefault() digunakan untuk menambahkan key dan value baru ke dalam dictionary jika key tersebut tidak ada di dalam dictionary, jika key tersebut sudah ada maka value tidak akan berubah. Ini juga dapat digunakan untuk mencegah error dari tidak ditemukannya key di dalam dictionary.

Contoh

```
Nilai = {
"Matematika" : 80,
"B. Indonesia" : 90,
"B. Inggris" : 81
}

#sebelum Setdefault
print(Nilai)

print("")

#menggunakan setdefault
print("Nilai : ", Nilai.setdefault("Kimia", 70))

print("")

#setelah menggunakan setdefault
print(Nilai)

{'Matematika': 80, 'B. Indonesia': 90, 'B. Inggris':
81}

Nilai : 70

{'Matematika': 80, 'B. Indonesia': 90, 'B. Inggris':
81, 'Kimia': 70}
```

Dictionary of List and Nested Dictionary

Seperti yang kita ketahui, Dictionary dapat diisi dengan **List** ataupun **Dictionary** yang disebut sebagai **Nested Dictionary** jika di dalam sebuah dictionary berisi dictionary, jika hal ini terjadi maka cara memanggilnya jelas berbeda karena harus memerlukan dua kali pemanggilan agar key dan value dapat terbaca.

Contoh

```
Musik = {
"The Chainsmoker" : ["All we Know", "The Paris"],
"Alan Walker" : ["Alone", "Lily"],
"Neffex" : ["Best of Me", "Memories"]
}

for i, j in Musik.items():
 print(f"Musik milik {i} adalah : ")
 for song in j:
  print(song)
 print("")

Musik milik The Chainsmoker adalah :
All we Know
The Paris

Musik milik Alan Walker adalah :
Alone
Lily

Musik milik Neffex adalah :
Best of Me
Memories
```

Studi Kasus

1. Buatlah sebuah **nested dictionary** yang berisi minimal **2 data mahasiswa**, di mana setiap mahasiswa memiliki **NIM** sebagai key, dan value-nya berupa dictionary dengan 3 data (key) : **nama, kelas, dan IPK**. Setelah itu, gunakan **nested loop (perulangan dalam perulangan)** untuk menampilkan seluruh data mahasiswa dengan rapi.