Modul Praktikum

Algoritma Pemrograman Dasar

8

“ Dictionary “

8.1. Tujuan Pembelajaran

- Mahasiswa mengetahui berbagai macam tipe data yang ada pada python dan menerapkannya.
- Mahasiswa mengetahui berbagai macam operator beserta fungsi dan penggunaannya.
- Mahasiswa dapat mengakses, mengubah, menambah, dan menghapus Key serta Value dari Dictionary.

8.2. Penjelasan

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.

8.3. Deklarasi Dictionary

Pendeklarasian pada dictionary akan menggunakan tanda kurung kurawal {} lalu isi dari dictionary tersebut akan berada didalamnya. Dan cara pemakaiannya akan seperti pada contoh dibawah ini.

Contoh 1:

```
daftar_buku = {
    "Buku1" : "Harry Potter",
    "Buku2" : "Percy Jackson",
    "Buku3" : "Twillight"
}

print(daftar_buku["Buku1"])
print(daftar_buku["Buku2"])
print(daftar_buku["Buku3"])
```

Output 1:

```
Harry Potter
Percy Jackson
Twillight
```

Contoh 2:

```
daftar_buku = {}

daftar_buku["Buku1"] = "Harry Potter"
daftar_buku["Buku2"] = "Percy Jackson"
daftar_buku["Buku3"] = "Twillight"

print(daftar_buku)
```

Output 2:

```
{'Buku1': 'Harry Potter', 'Buku2': ' Percy Jackson
', 'Buku3': ' Twillight '}
```

8.4. Sifat Dictionary

Dictionary memiliki 3 sifat, yaitu :

1. Unordered - Tidak Berurutan
2. Changeable - Bisa Diubah
3. Unique - Tidak memiliki kesamaan

Dari sifat yang pertama, bisa kita ketahui bahwa sebuah Dictionary belum tentu berurutan. Bisa saja apa yang kita taruh di awal akan menjadi yang "Pertama". Changeable berarti kita dapat mengubah value yang telah kita masukan dalam dictionary itu.

Dan terakhir Unique, tidak memiliki kesamaan disini maksudnya adalah key pada dictionary tidak boleh memiliki kesamaan dalam penamaannya. Karena hal tersebut akan membuat key yang didefinisikan terakhir akan menindih key yang didefinisikan di awal.

Contoh:

```
musik = {
    "judul" : "All we Know",
    "judul" : "Something Just Like This"
}

print(musik["judul"])
```

Output:

```
Somethings Just Like This
```

8.5. Membuat Dictionary

Terdapat beberapa Hal yang perlu diingat dan sangat diperlukan dalam sebuah dictionary, yaitu :

a) Nama Dictionary
b) Key / Atribut
c) Value / Nilai
d) Buka dan tutup kurung kurawal

Di antara Key dan value akan ada tanda titik dua (:) sebagai pemisahnya. Dan apabila terdapat lebih dari satu item dalam Dictionary maka dipisah dengan tanda koma (,)

Contoh satu item

```
nama_dict = {
    "key": "value"
}
```

Contoh multiple item

```
nama_dict = {
    "key1": "value",
    "key2": "value",
    "key3": "value"
}
```

Isi dari Dictionary dapat berupa berbagai macam tipe data seperti :

- String
- Integer
- Objek
- List
- Dictionary
- Dll..

Contoh 1:

```
Biodata = {
    "Nama" : "Aldy Ramadhan Syahputra",
    "NIM" : 2109106079,
    "KRS" : ["Program Web", "Struktur Data", "Basis Data"],
    "Mahasiswa_Aktif" :True,
    "Social Media" : {
        "Instagram" : "@aldyrmdhns_",
        "Discord" : "\'Izanami#6848"
    }
}
```

Dari contoh diatas dapat kita lihat bahwa :

- Key “Nama” berisi String
- Key “NIM” berisi Integer
- Key “KRS” berisi List dari string
- Key “Mahasiswa_Aktif” berisi Boolean
- Key “Social Media” berisi Dictionary

Selain itu, kita juga dapat membuat sebuah dictionary dengan konstruktor **dict()** dengan parameter key dan value

Contoh :

```
games = dict(Sekiro = "Action", Pokemon = "Adventure",
Valorant = "FPS")

print(games)
```

Output:

```
{'Sekiro': 'Action', 'Pokemon': 'Adventure', 'Valorant':
'FPS'}
```

8.6. Mengakses Item pada Dictionary

Terdapat dua cara untuk mengakses item yang ada pada dictionary. Yaitu dengan cara:

- Kurung Siku [ ]
- 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”]

Contoh:

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

Output:

```
nama saya adalah Aldy Ramadhan Syahputra
NIM Saya adalah 2109106079
Instagram : @aldyrmdhns_
```

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.get('Nama')}")
print(f"NIM Saya adalah {Biodata.get('NIM')}")
```

Output:

```
nama saya adalah Aldy Ramadhan Syahputra
NIM Saya adalah 2109106079
```

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"))
```

Output:

```
Aldy Ramadhan Syahputra
None
Key tersebut tidak ada
```

8.7. Perulangan pada Python

Jika kita menampilkan items pada dictionary dengan cara seperti sebelumnya, sudah pasti akan melelahkan jika data pada dictionary tersebut sangat banyak. Maka, disini kita dapat menggunakan perulangan untuk mengatasinya. Adapun caranya dengan menggunakan perulangan For.

Tapi kita juga perlu menambahkan sebuah fungsi tambahan untuk dapat memanggil keduanya. Yaitu dengan menggunakan **items()**. Jika tidak, maka yang muncul hanyalah key pada 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("")

#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
```

8.8. Menambahkan Item pada Dictionary

Untuk menambahkan sebuah item baru ke dictionary, kita dapat melakukannya dengan 2 cara. Bisa langsung kita masukkan saja atau kita bisa menggunakan fungsi **update()**

Contoh :

```
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)
```

Output:

```
#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'}
```

8.9. Mengubah Item Pada Dictionary

Mengubah item pada dictionary ini sama seperti cara menambahnya, namun kita hanya perlu memasukan key yang sama dengan sebelumnya untuk menindih value yang sudah ada.

Contoh:

```
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)
```

Output:

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

8.10. Menghapus Item Pada Dictionary

Untuk menghapus Item pada dictionary terdapat 3 cara yang bisa dilakukan, yaitu :

- **del**
- **pop()**
- **clear()**

Dalam Penggunaan **del**, item yang akan dihapus akan betul-betul terhapus dan tidak akan ada lagi sisanya.

Contoh:

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

#Sebelum Dihapus
print(data)

del data["Nama"]

#Setelah diubah
print(data)

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

Output:

```
{'Nama': 'Aldy', 'Umur': 19, 'Jurusan': 'Informatika'}
{'Umur': 19, 'Jurusan': 'Informatika'}
None
```

Jika kita menggunakan fungsi **pop()** untuk menghapus suatu item, kita tetap bisa melihat apa yang kita hapus tersebut pada sebuah variable yang kita gunakan untuk menampung sebagai tempat sampah dari item yang ingin kita hapus.

Contoh:

```
data = {
    "Nama" : "Aldy",
    "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)
```

Output:

```
{'Nama': 'Aldy', 'Umur': 19, 'Jurusan': 'Informatika'}
{'Umur': 19, 'Jurusan': 'Informatika'}
None
Aldy
```

Dan cara menghapus yang terakhir adalah dengan fungsi **clear()**. Dimana fungsi ini akan menghapus semua isi dari dictionary yang kita miliki. Baik itu value maupun key-nya.

Contoh:

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

#Sebelum Dihapus
print(data)

data.clear()

#Setelah dihapus
print(data)
```

Output:

```
{'Nama': 'Aldy', 'Umur': 19, 'Jurusan': 'Informatika'}
{}
```

8.11. Beberapa Fungsi yang bisa dipakai

A. Mengetahui Panjang dari Dictionary

Caranya sama seperti mengetahui Panjang dari list. Kita cukup menggunakan fungsi **len()** untuk mengetahuinya.

Contoh:

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

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

Ouput:

```
Jumlah Data =  3
```

B. Copy & Fromkeys

Seperti Namanya, fungsi **copy()** dapat membuat sebuah copy/salinan dari dictionary yang sudah kita buat di dalam sebuah variable.

Contoh:

```
Buku = {
    "No Longer Human" : "Osamu Dazai",
    "Harry Potter" : "J.K. Rowlings",
    "Hamlet" : "William Shakespeare"
}

pinjam = Buku.copy()

print("Dictionary yang Telah Disalin : ", pinjam)
```

Output:

```
Dictionary yang Telah Disalin :  {'No Longer Human':
'Osamu Dazai', 'Harry Potter': 'J.K. Rowlings',
'Hamlet': ‘William Shakespeare '}
```

Dan untuk kegunaan fungsi fromkeys() ini kita dapat membuat sebuah dictionary yang telah kita siapkan key dan valuenya pada sebuah variable 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)
```

Output:

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

C. 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)
```

Output:

```
Matematika
B. Indonesia
B. Inggris
Kimia
Fisika
```

```
80
90
81
78
80
```

D. Setdefault

Kegunaan dari fungsi **setdefault()** adalah untuk menambahkan key dan value baru ke dalam dictionary jika key tersebut tidak ada di dalam dictionary. Jika key tersebut sudah ada, maka value dari key tersebut tidak akan berubah.

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)
```

Output:

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


Nilai :  70

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

E. Dictionary of List and Nested Dictionary

Sebuah dictionary bisa juga berisi banyak list. Dan cara pemanggilannya pun akan lumayan berbeda dengan biasanya karena kita perlu melakukan perulangan dua kali untuk menampilkan key dari dictionary dan semua value dari list.

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("")
```

Output:

```
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
```

Dictionary juga bisa diisi oleh dictionary lagi sehingga akan membentuk suatu dictionary yang bersarang atau biasa disebut **nested dictionary**.

Cara pemanggilan yang akan dipakai juga berbeda karena terdapat 2 buah dictionary yang perlu diakses. Kita perlu melakukan dua kali perulangan untuk menampilkan semuanya. Perulangan pertama untuk menampilkan key dari dictionary yang berada di luar, dan perulangan kedua untuk menampilkan semua key dan value dari dictionary yang berada di dalam.

Contoh:

```
mahasiswa = {
    101 : {"Nama" : "Aldy", "Umur" : 19},
    111 : {"Nama" : "Abdul", "Umur" : 18}
}

for key, value in mahasiswa.items():
    print("ID Mahasiswa : ", key)
    for key_a, value_a in value.items():
        print (key_a, " : ", value_a)
    print("")
```

Output:

```
ID Mahasiswa :  101
Nama  :  Aldy
Umur  :  19

ID Mahasiswa :  111
Nama  :  Abdul
Umur  :  18
```

Untuk menambahkan, mengubah, atau menghapus item pada nested dictionary, dapat dilakukan dengan memasukkan key dictionary yang berada di luar, kemudian memasukkan key dictionary yang berada di dalam.

Contoh:

```
mahasiswa = {
    101 : {"Nama" : "Aldy", "Umur" : 19},
    111 : {"Nama" : "Abdul", "Umur" : 18}
}
#Sebelum Dilakukan Perubahan

print(mahasiswa)
```

```
#Menambahkan Item pada Nested Dictionary
mahasiswa[101]["Angkatan"] = 2023
print(mahasiswa)

#Mengubah Item pada Nested Dictionary
mahasiswa[101]["Nama"] = "Rizal"
print(mahasiswa)

#Menghapus Item pada Nested Dictionary
del mahasiswa[101]["Umur"]
print(mahasiswa)
```

Output:

```
#Sebelum Dilakukan Perubahan
{101: {'Nama': 'Aldy', 'Umur': 19}, 111: {'Nama':
'Abdul', 'Umur': 18}}

#Setelah Penambahan Item
{101: {'Nama': 'Aldy', 'Umur': 19, 'Angkatan': 2023},
111: {'Nama': 'Abdul', 'Umur': 18}}

#Setelah Perubahan Item
{101: {'Nama': 'Rizal', 'Umur': 19, 'Angkatan': 2023},
111: {'Nama': 'Abdul', 'Umur': 18}}

#Setelah Penghapusan Item
{101: {'Nama': 'Rizal', 'Angkatan': 2023}, 111:
{'Nama': 'Abdul', 'Umur': 18}}
```

Studi Kasus

1. Buatlah sebuah dictionary yang memiliki 5 key (Nama, Umur, NIM, Jurusan, Angkatan). Setelah itu buatlah dictionary ini dapat :
   -Menambahkan Item baru sesuai dengan inputan User
   -Mengubah salah satu key sesuai dengan inputan User
   -Menghapus salah satu key sesuai dengan Inputan user

2. Buatlah sebuah dictionary yang berisikan data berikut, kemudian hitunglah jumlah total dan rata-rata nilainya.

Mata Pelajaran	Nilai
Matematika	90
Fisika	80
Biologi	80
Kimia	70