

LAPORAN PRAKTIKUM

STRUKTUR DATA 2



MUHAMMAD AVERROES ALIF AKBAR

25071103405

S1 TEKNIK INFORMATIKA

GENAP 2025/2026

BAB I

PENDAHULUAN

1.1. Latar Belakang

List adalah struktur data bawaan yang digunakan untuk menyimpan kumpulan item (data) secara berurutan dalam satu variabel. List bersifat *mutable* (dapat diubah isinya), terurut, dinamis (ukuran bisa berubah), dan dapat menampung berbagai tipe data (heterogen) seperti string, angka, atau objek. List ditandai dengan kurung siku []. List adalah salah satu struktur data yang paling serbaguna dan sering digunakan dalam pemrograman Python.

Tuple merupakan tipe data koleksi yang terurut (ordered) dan tidak dapat diubah (*immutable*) setelah dibuat. Tuple didefinisikan menggunakan kurung biasa () dan memuat elemen yang dipisahkan koma. Karena sifatnya yang tidak bisa diubah, tuple lebih cepat dan efisien dibandingkan list untuk menyimpan data konstan. Meskipun isinya tidak bisa diubah, tuple mendukung operasi seperti *indexing* (pengambilan indeks), *slicing* (pemotongan), dan penggabungan, yang menghasilkan tuple baru. Tuple tidak dapat diubah (*immutable*) di Python karena didesain sebagai koleksi data yang tetap, aman, dan efisien. Sifat ini menjamin integritas data (tidak

bisa diubah tidak sengaja), memungkinkan penggunaan sebagai kunci *dictionary* yang *hashable*, serta meningkatkan performa memori dibandingkan *list*.

Set merupakan struktur data bawaan yang digunakan untuk menyimpan koleksi item yang unik (tidak memiliki duplikat), tidak berurutan (*unordered*), dan juga dapat diubah (*mutable*). Set ditandai dengan kurung kurawal {} atau fungsi set, dan juga ideal untuk menghilangkan duplikat dan melakukan operasi himpunan seperti irisan (*intersection*) atau gabungan (*union*).

Dictionary adalah tipe data koleksi yang terurut, dapat diubah (*mutable*), dan menyimpan data dalam format pasangan kunci-nilai (key-value pair). Dictionary ditandai dengan kurung kurawal {} dan digunakan untuk memetakan nilai ke kunci unik, mirip seperti kamus bahasa, memungkinkan akses data yang cepat dan efisien menggunakan kunci. Dictionary dipakai untuk merepresentasikan data terstruktur, konfigurasi aplikasi, dan manipulasi data JSON.

1.2. Tujuan

Mahasiswa mampu menggunakan List, Tuple, Set, dan Dictionary dalam Python dan mengimplementasikan ke dalam program dan memahami kelebihan dan kegunaannya untuk membantu dalam menyusun data

BAB II

PEMBAHASAN

Di Pertemuan Praktikum Kedua, kita belajar tentang jenis-jenis penyimpanan data (database), yaitu List, Tuple, Set, dan Dictionary. Berikut merupakan penjelasan di Pertemuan Praktikum Kedua:

1. List

List adalah struktur data bawaan yang digunakan untuk menyimpan kumpulan item (data) secara berurutan dalam satu variabel. List bersifat *mutable* (dapat diubah isinya), terurut, dinamis (ukuran bisa berubah), dan dapat menampung berbagai tipe data (heterogen) seperti string, angka, atau objek. List ditandai dengan kurung siku []. List adalah salah satu struktur data yang paling serbaguna dan sering digunakan dalam pemrograman Python. Python memiliki serangkaian metode bawaan yang dapat Anda gunakan pada list, diantaranya :

Method	Description
<u>append()</u>	Adds an element at the end of the list
<u>clear()</u>	Removes all the elements from the list
<u>copy()</u>	Returns a copy of the list
<u>count()</u>	Returns the number of elements with the specified value
<u>extend()</u>	Add the elements of a list (or any iterable), to the end of the current list
<u>index()</u>	Returns the index of the first element with the specified value
<u>insert()</u>	Adds an element at the specified position
<u>pop()</u>	Removes the element at the specified position
<u>remove()</u>	Removes the item with the specified value
<u>reverse()</u>	Reverses the order of the list
<u>sort()</u>	Sorts the list

```

thislist = ["apple", "banana", "cherry", "apple", "cherry"]
print(thislist) # menampilkan seluruh item dalam list
print(thislist.count("cherry")) # menghitung berapa kali item "cherry" muncul dalam list
print(len(thislist)) # menghitung jumlah item dalam list
print(thislist[-1]) # menampilkan item terakhir dalam list
if "apple" in thislist:
    print("Yes, 'apple' is in the fruits list") # memeriksa apakah "apple" ada dalam list


list1 = ["apple", "banana", "cherry"]
list2 = [1, 5, 7, 9, 3]
list3 = [True, False, False]
list1 = ["abc", 34, True, 40, "male"]
print(list1) # menampilkan list dengan tipe data campuran


mylist = ["apple", "banana", "cherry"]
print(type(mylist)) # menampilkan tipe data list
thislist = list(("apple", "banana", "cherry")) # note the double round-brackets
print(thislist) # menampilkan list yang dibuat dengan constructor list()
thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[2:5]) # menampilkan item dari index 2 sampai index 4


thislist = ["apple", "banana", "cherry", "orange", "kiwi", "mango"]
thislist[1:3] = ["blackcurrant", "watermelon"]
print(thislist) # mengganti item pada index 1 dan 2 dengan "blackcurrant" dan "watermelon"

```

```
thislist = ["apple", "banana", "cherry", "banana", "kiwi"]
thislist.remove("banana")
print(thislist) # menghapus item "banana" dari list (hanya yang pertama ditemukan)

thislist = ["apple", "banana", "cherry"]
thislist.pop(1)
print(thislist) # menghapus item pada index 1 (yaitu "banana")

thislist = ["apple", "banana", "cherry"]
thislist.pop()
print(thislist) # menghapus item terakhir dalam list

thislist = ["apple", "banana", "cherry"]
del thislist[0]
print(thislist) # menghapus item pada index 0 (yaitu "apple")

thislist = ["apple", "banana", "cherry"]
thislist.clear()
print(thislist) # menghapus semua item dalam list sehingga list menjadi kosong
| 

thislist = ["apple", "banana", "cherry"]
del thislist # menghapus seluruh list
# print(thislist) # ini akan menimbulkan error karena list telah dihapus
```

```
thislist = ["apple", "banana", "cherry"]
i = 0
while i < len(thislist):
    print(thislist[i])
    i = i + 1 # menampilkan semua item dalam list menggunakan loop while

fruits = ["apple", "banana", "cherry", "kiwi", "mango"]
newlist = []

for x in fruits:
    if "a" in x:
        newlist.append(x)

print(newlist) # membuat list baru yang hanya berisi item yang mengandung huruf "a"

fruits = ["apple", "banana", "cherry", "kiwi", "mango"]
newlist = ['hello ' + x for x in fruits]

print(newlist) # membuat list baru dengan menambahkan kata "hello " di depan setiap item dalam list

thislist = ["orange", "mango", "kiwi", "pineapple", "banana"]
thislist.sort()
print(thislist) # mengurutkan item dalam list secara ascending (A-Z)
```

```
thislist = [100, 50, 65, 82, 23]
thislist.sort()
print(thislist) # mengurutkan item dalam list angka secara ascending (dari kecil ke besar)

thislist = [100, 50, 65, 82, 23]
thislist.sort(reverse = True)
print(thislist) # mengurutkan item dalam list angka secara descending (dari besar ke kecil)

def myfunc(n):
| return abs(n - 50)

thislist = [100, 50, 65, 82, 23]
thislist.sort(key = myfunc)
print(thislist) # mengurutkan item dalam list berdasarkan jarak dari angka 50
```

2. Tuple

Tuple merupakan tipe data koleksi yang terurut (ordered) dan tidak dapat diubah (immutable) setelah dibuat. Tuple didefinisikan menggunakan kurung biasa () dan memuat elemen yang dipisahkan koma. Karena sifatnya yang tidak bisa diubah, tuple lebih cepat dan efisien dibandingkan list untuk menyimpan data konstan. Meskipun isinya tidak bisa diubah, tuple mendukung operasi seperti *indexing* (pengambilan indeks), *slicing* (pemotongan), dan penggabungan, yang menghasilkan tuple baru. Tuple tidak dapat diubah (*immutable*) di Python karena didesain sebagai koleksi data yang tetap, aman, dan efisien. Sifat ini menjamin integritas data (tidak bisa diubah tidak sengaja), memungkinkan penggunaan sebagai kunci *dictionary* yang *hashable*, serta meningkatkan performa memori dibandingkan *list*. Python memiliki dua metode bawaan yang dapat Anda gunakan pada tuple, seperti :

Method	Description
<code>count()</code>	Returns the number of times a specified value occurs in a tuple
<code>index()</code>	Searches the tuple for a specified value and returns the position of where it was found

3. Set

Set merupakan struktur data bawaan yang digunakan untuk menyimpan koleksi item yang unik (tidak memiliki duplikat), tidak berurutan (*unordered*), dan juga dapat diubah (*mutable*). Set ditandai dengan kurung kurawal {} atau fungsi set, dan juga ideal untuk menghilangkan duplikat dan melakukan operasi himpunan seperti irisan (*intersection*) atau gabungan (*union*). Python memiliki serangkaian metode bawaan yang dapat Anda gunakan pada Set, diantaranya :

Method	Shortcut	Description
<code>add()</code>		Adds an element to the set
<code>clear()</code>		Removes all the elements from the set
<code>copy()</code>		Returns a copy of the set
<code>difference()</code>	-	Returns a set containing the difference between two or more sets
<code>difference_update()</code>	-=	Removes the items in this set that are also included in another, specified set
<code>discard()</code>		Remove the specified item
<code>intersection()</code>	&	Returns a set, that is the intersection of two other sets
<code>intersection_update()</code>	&=	Removes the items in this set that are not present in other, specified set(s)
<code>isdisjoint()</code>		Returns whether two sets have a intersection or not
<code>issubset()</code>	<=	Returns True if all items of this set is present in another set
	<	Returns True if all items of this set is present in another, <i>larger</i> set
<code>issuperset()</code>	>=	Returns True if all items of another set is present in this set
	>	Returns True if all items of another, <i>smaller</i> set is present in this set
<code>pop()</code>		Removes an element from the set
<code>remove()</code>		Removes the specified element
<code>symmetric_difference()</code>	^	Returns a set with the symmetric differences of two sets
<code>symmetric_difference_update()</code>	^=	Inserts the symmetric differences from this set and another
<code>union()</code>		Return a set containing the union of sets
<code>update()</code>	=	Update the set with the union of this set and others

```
'''Set adalah kumpulan data yang tidak berurutan dan tidak dapat diubah (immutable).  
Set digunakan untuk menyimpan beberapa item dalam satu variabel.'''  
  
thisset = {"apple", "banana", "cherry", False, True, 0, 1}  
  
print(thisset) # menampilkan seluruh item dalam set (urutan tidak terjamin)  
  
# false dan 0 dianggap item yang sama dalam set  
# true dan 1 dianggap item yang sama dalam set  
  
print(len(thisset)) # menghitung jumlah item dalam set  
  
  
thisset = {"apple", "banana", "cherry"}  
tropical = {"pineapple", "mango", "papaya"}  
  
thisset.update(tropical)  
  
print(thisset) # menambahkan item dari set tropical ke dalam thisset
```

4. Dictionary

Dictionary adalah tipe data koleksi yang terurut, dapat diubah (*mutable*), dan menyimpan data dalam format pasangan kunci-nilai (key-value pair). Dictionary ditandai dengan kurung kurawal {} dan digunakan untuk memetakan nilai ke kunci unik, mirip seperti kamus bahasa, memungkinkan akses data yang cepat dan efisien menggunakan kunci. Dictionary dipakai untuk merepresentasikan data terstruktur, konfigurasi aplikasi, dan manipulasi data JSON. Python memiliki serangkaian metode bawaan yang dapat Anda gunakan pada Dictionary, diantaranya :

Method	Description
<u>clear()</u>	Removes all the elements from the dictionary
<u>copy()</u>	Returns a copy of the dictionary
<u>fromkeys()</u>	Returns a dictionary with the specified keys and value
<u>get()</u>	Returns the value of the specified key
<u>items()</u>	Returns a list containing a tuple for each key value pair
<u>keys()</u>	Returns a list containing the dictionary's keys
<u>pop()</u>	Removes the element with the specified key
<u>popitem()</u>	Removes the last inserted key-value pair
<u>setdefault()</u>	Returns the value of the specified key. If the key does not exist: insert the key, with the specified value
<u>update()</u>	Updates the dictionary with the specified key-value pairs
<u>values()</u>	Returns a list of all the values in the dictionary

```

car = {
"brand": "Ford",
"model": "Mustang",
"year": 1964
}

x = car.keys()

print(x) # menampilkan keys dalam dictionary

car["color"] = "white"

print(x) # menampilkan keys dalam dictionary, setelah menambahkan item baru keys akan terupdate

```

```

child1 = {
    "name" : "Emil",
    "year" : 2004
}
child2 = {
    "name" : "Tobias",
    "year" : 2007
}
child3 = {
    "name" : "Linus",
    "year" : 2011
}

myfamily = {
    "child1" : child1,
    "child2" : child2,
    "child3" : child3
}
print(myfamily) # menampilkan dictionary yang berisi dictionary lain sebagai nilai
print(myfamily["child2"]["name"]) # menampilkan nilai dari key "name" dalam dictionary child2

```

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
print(thisdict["brand"]) # menampilkan nilai dari key "brand" dalam dictionary  
print(len(thisdict)) # menghitung jumlah item dalam dictionary  
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964,  
    "year": 2020  
}  
print(thisdict) # menampilkan dictionary (key "year" akan memiliki nilai terakhir yaitu 2020)
```

BAB III

PENUTUP

Dengan laporan praktikum di atas, kita akhirnya sampai pada penutup, bahwa List, Tuple, Set, dan Dictionary merupakan tools yang berguna bagi para pengembang dalam menyusun data-data yang ada di dalam program, dan memiliki keunggulan dan kelebihan di setiap kegunaannya. Praktikum kedua ini mengajarkan kita tentang pentingnya menyusun data yang baik, serta mengurus kode program yang mudah dibaca oleh kita, bahkan orang lain.

DAFTAR PUSTAKA

<https://www.revou.co/id/panduan-teknis/python-set>

<https://mimo.org/glossary/python/set>

<https://dqlab.id/belajar-python-list-tuples-set-dan-dictionary>