Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	**71231058**
Nama Lengkap	**Michael Chandra Mahanaim**
Minggu ke / Materi	**11 / Tipe Data Tuples**

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA

YOGYAKARTA
2024

BAGIAN 1: MATERI MINGGU INI (40%)

MATERI 1: Tuple

Tuple bersifat immutable, yang berarti isinya tidak bisa diubah / dimodifikasi. Data dari dalam tuple dapat diambil dan di konversi menjadi data lain. Berikut contoh tuple dan pencobaan untuk mengubah isi tuple:

```
>>> t = ('a', 'b', 'c', 'd', 'e')

>>> t[0] = 'A'
TypeError: object doesn't support item assignment
```

Gambar contoh sifat tuple sebagai immutable

Isi tuple tidak dapat diubah, tetapi data masih bisa diambil, dimasukkan ke variable lain, dan mengubah variabel itu. Tuple juga dapat digunakan untuk memecah sebuah string menjadi masing-masing hurufnya tersendiri, seperti contoh di bawah ini:

```
>>> t = tuple('dutawacana')
>>> pint(t)
('d','u','t','a','w','a','c','a','n','a')
```

Gambar tuple memecah string menjadi item tersendiri

Data tuple tidak dapat diubah, namun dapat diganti menggunakan slicing dan tuple lain. Contohnya seperti di bawah ini:

```
>>> t = ('A',) + t[1:]
>>> print(t)
('A', 'b', 'c', 'd', 'e')
```

Gambar penggantian 'a' menjadi 'A' dengan slicing

MATERI 2: Pembandingan Tuple

Operator perbandingan (==, !=, >, <) dapat digunakan ke tuple. Dalam kasus tuple, perbandingan akan dilakukan per indeks tuple yang dibandingkan. Maka tuple1 dengan tuple2 akan dibandingkan indeks ke-0, lalu ke-1, dan seterusnya satu per satu hingga terjadi perbandingan yang mencukupi kondisi dari perbandingan. Contoh sebagai berikut:

```
>>> (0, 1, 2) < (0, 3, 4)
True
>>> (0, 1, 2000000) < (0, 3, 4)
True
```

Gambar perbandingan tuple

Tuple juga dapat digunakan untuk menyimpan 2 data yang berhubungan, contohnya yaitu adalah sebuah kalimat dengan kata yang panjangnya berbeda-beda. Dengan tuple, dapat digunakan untuk menyimpan panjang kata dan kata tersebut dalam 1 variable. Variabel itu kemudian dapat dimasukkan ke dalam list menjadi tuple list. Contohnya seperti dibaw

```
kalimat = 'but soft what light in yonder window breaks'
dafkata = kalimat.split()
t = list()
for kata in dafkata:
    t.append((len(kata), kata))

t.sort(reverse=True)

urutan = list()
for length, kata in t:
    urutan.append(kata)

print(urutan)
```

Dari contoh di atas, output yang keluar akan diurutkan dari kata paling panjang ke kata paling pendek.

```
['yonder', 'window', 'breaks', 'light', 'what', 'soft', 'but', 'in']
```

Gambar output pengurutan kata panjang ke pendek

MATERI 3: Dictionary dan Tuple

Tuple, List, dan Dictionary berhubungan erat, dengan menggunakan tuple, dictionary dapat dirubah menjadi tuple in a list dan bisa dipanggil dengan angka indeks. Contoh sebagai berikut:

```
>>> d = {'a':10, 'b':1, 'c':22}
>>> t = list(d.items())
>>> print(t)
[('b', 1), ('a', 10), ('c', 22)]
```

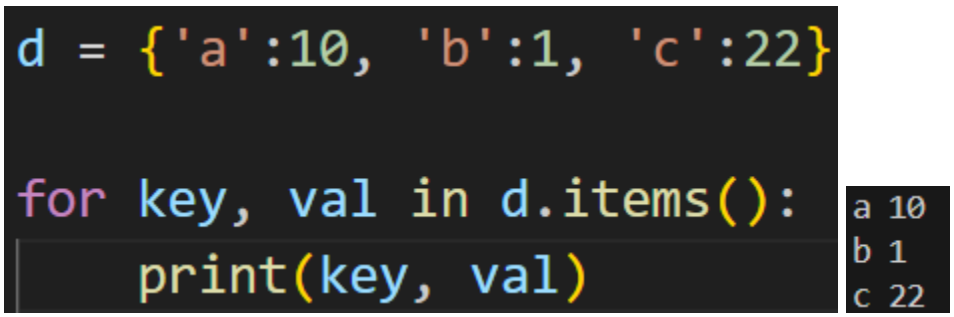
Gambar pengubahan dictionary menjadi tuple list

Jika list tersebut tidak dalam bentuk terurut, maka dapat diurutkan menggunakan list methods .sort(). Sort dengan tuple akan mengacu pada item pertama dalam tuple atau index ke-0 tuple. Penggunaannya seperti berikut

```
>>> t.sort()
>>> t
[('a', 10), ('b', 1), ('c', 22)]
```

Gambar penggunaan list methods sort() dengan tuple

Dengan dictionary methods .items() juga dapat digunakan untuk mengambil key dan value secara langsung dengan looping. Dalam sintaks for, diberikan 2 variabel untuk bentuk tuple dari .items(), dimana indeks 0 adalah key dan indeks 1 adalah value. Contohnya sebagai berikut:



Contoh penggunaan dan output for loop untuk mengambil key dan value dari dictionary yang diubah ke tuple list

MATERI 4: Tuple dan Dictionary Keys

Tuple bisa digunakan sebagai key sebuah dictionary. Format yang digunakan sama seperti key int atau str, tetapi menggunakan tuple. Contoh mudah seperti di bawah ini:

```
>>> last = 'nendya'
>>> first = 'dida'
>>> number = '088112266'
>>> directory = dict()
>>> directory[last, first] = number
>>> for last, first in directory:
...     print(first, last, directory[last,first])
...
dida nendya 088112266
>>>
```

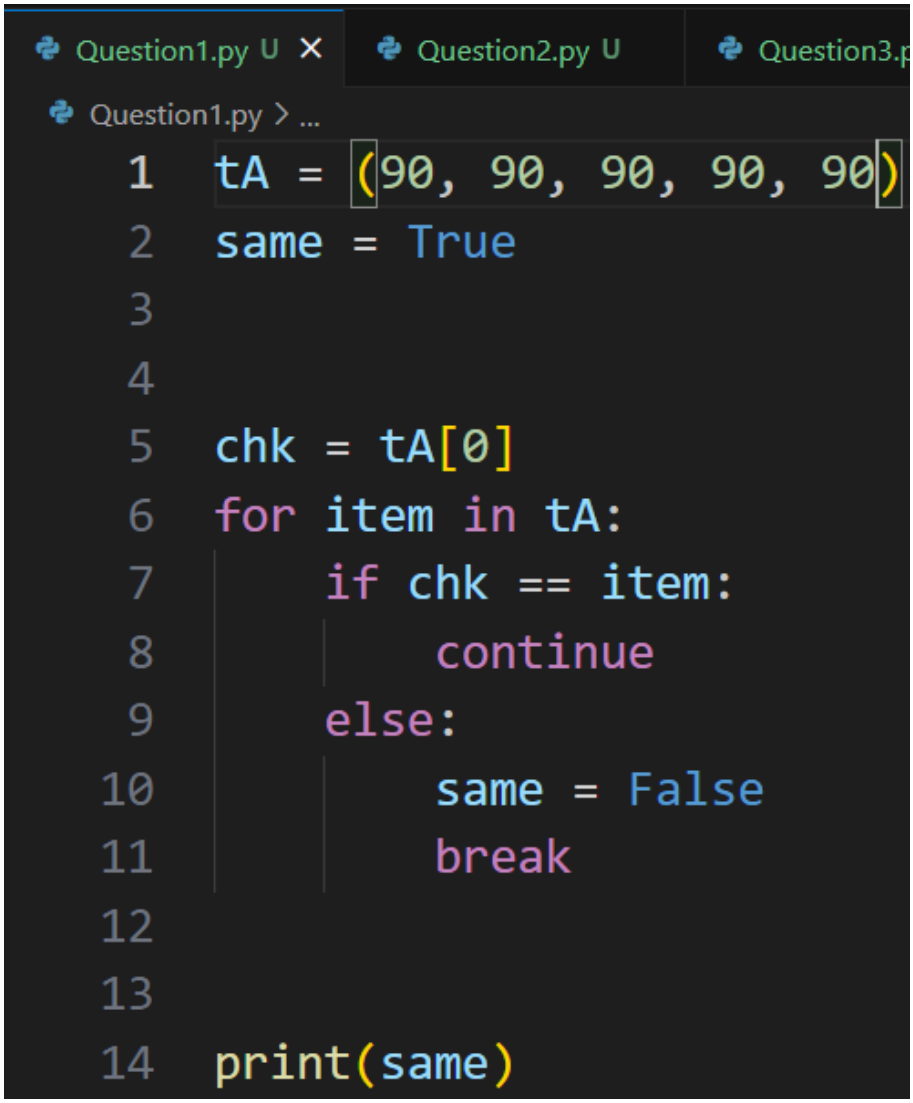
Gambar contoh penggunaan tuple sebagai key untuk mengambil sebuah value dalam dictionary

Dari contoh di atas menggunakan variabel first dan last sebagai key dengan number sebagai value yang kemudian disimpan ke dalam dictionary directory. Saat value akan dipanggil, maka hanya perlu menggunakan dictionary biasa serta pemanggilan key dengan koma sebagai penanda pemanggilan key tuple.

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

A. Source Code



B. Output Result

C. Explanation

Kode di atas hanya menggunakan for loop untuk mengambil setiap item di dalam tuple dan membandingkannya dengan item pertama. JIka semua item sama, maka same akan tetap True, jika ada yang berbeda, maka same akan berubah menjadi False.

SOAL 2

A. Source Code

```python
Data = ('Matahari Bhakti Nendya', '22064091', 'Bantul, DI Yogyakarta')

nim = []
for num in Data[1]:
    nim.append(num)
nim = tuple(nim)

name = Data[0].split()
initial = name[0]

alpha = []
for i in range(1, len(initial)):
    alpha.append(initial[i])
alpha = tuple(alpha)

name.reverse()
rever = tuple(name)

print(f"{Data}\n")
print(f"NIM\t\t:\t{Data[1]}")
print(f"NAMA\t:\t{Data[0]}")
print(f"ALAMAT\t:\t{Data[2]}\n")
print(f"NIM:\t{nim}\n")
print(f"NAMA DEPAN:\t{alpha}\n")
print(f"NAMA TERBALIK:\t{rever}")
```

B. Output Result

```
('Matahari Bhakti Nendya', '22064091', 'Bantul, DI Yogyakarta')

NIM     :       22064091
NAMA    :       Matahari Bhakti Nendya
ALAMAT  :       Bantul, DI Yogyakarta

NIM:    ('2', '2', '0', '6', '4', '0', '9', '1')

NAMA DEPAN:     ('a', 't', 'a', 'h', 'a', 'r', 'i')

NAMA TERBALIK:  ('Nendya', 'Bhakti', 'Matahari')
```

C. Explanation

Kode di atas hanya berfungsi jika Nama, NIM, dan Alamat ada dalam tuple dengan urutan tersebut. Kode di atas menggunakan indeks untuk print masin-masing nama, nim, dan alamat. Pemisahan NIM digunakan dengan membuat for loop untuk setiap angka di NIM dan memasukkannya ke dalam list yang kemudian dikonversi menjadi tuple. Nama depan dilakukan dengan for loop yang dimulai dari indeks 1 dan per huruf dimasukkan ke dalam list yang dikonversi menjadi tuple. Nama terbalik menggunakan indeks nama yang di split dan kemudian dimasukkan ke list yang ordernya di reverse menggunakan list method yang kemudia dikonversi menjadi tuple.

SOAL 3

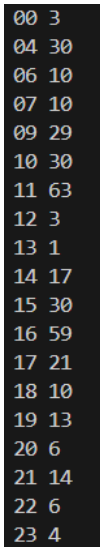
A. Source Code

Question1.py U | Question2.py U | Question3.py U | mbox-short.txt U

Question3.py > ...

```python
handle = open("mbox-short.txt", "r")

lines = handle.readlines()
d = {}
for line in lines:
    sentence = line.split()
    for word in sentence:
        if ":" in word and word[0].isdigit():
            numlst = word.split(":")
            if len(numlst[0]) <= 2:
                hour = numlst[0]
                d[hour] = d.get(hour, 0) + 1

lst = []
for key, val in d.items():
    lst.append((key, val))
lst.sort()

for i in range(len(lst)):
    print(lst[i][0], lst[i][1])

handle.close()
```

B. Output Result



C. Explanation

Kode di atas akan memecah setiap baris menjadi kata yang dipisah spasi, yang kemudian akan dicari tanda ":" untuk menandakan jam. Setelah ditemukan, akan dicek jika indeks pertama adalah angka atau bukan. Jika angka maka akan di split berdasarkan ":", lalu akan di cek panjang indeks pertama. Jika panjang indeks lebih dari 2 maka bukan tanda jam dan di skip, jika indeks panjangnya 2, maka itu adalah tanda jam yang akan dimasukkan ke dalam dicitonary sebagai penghitung. Proses diulangi hingga semua tanda jam tercatat. Yang kemudian menggunakan for loop dan dictionary methods .items() digunakan untuk memprint jam dan jumlah munculnya jam tersebut.