



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230970
Nama Lengkap	Gregorius Daniel Jodan Perminas
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 KULIAHINI.**

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2024

BAGIAN 1: MATERI MINGGU INI (40%)

Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

Apa itu Tuple?

Konsep tuple mirip dengan list, yaitu sebuah tipe data yang mampu menyimpan lebih dari satu nilai dalam satu variable. Tuple ini memiliki sifat ordered (terurut/memiliki indeks) dan bersifat immutable yang artinya setelah didefinisikan maka data di dalamnya tidak akan bisa diubah. Tuple juga dapat dibandingkan dan bersifat hashable sehingga dapat dimasukkan dalam list sebagai key pada dictionary Python.

Ada beberapa sintaks dalam penulisan tuple dalam Python:

```
# Cara Standar
tpl1 = ('Mantap', 'Joss', 'Elok')

# Tanpa Kurung
tpl2 = 'Mantap', 'Joss', 'Elok'

# Menggunakan Fungsi Tuple()
tpl3 = tuple(['Mantap', 'Joss', 'Elok'])
```

Keterangan =

- Cara Standar = Merupakan cara yang paling sering digunakan
- Cara Tanpa Kurung = Merupakan cara lain menulis tuple di python
- Menggunakan Fungsi Tuple() = Cara membuat tuple dengan memasukkan list sebagai parameternya

Catatan Khusus = Tuple dengan satu elemen perlu ditambahkan koma, jika tidak maka tipe datanya akan dianggap sama sesuai dengan value yang dimasukkan. Kecuali jika menggunakan fungsi tuple()

PROGRAM PYTHON

```
tpl4 = ("a")
print("tpl4 type =", type(tpl4))

tplFunction = tuple("a")
print("tplFunction type =", type(tplFunction))

tpl5 = ("a",)
print("tpl5 type =", type(tpl5))
```

OUTPUT

```
tpl4 type = <class 'str'>
tplFunction type = <class 'tuple'>
tpl5 type = <class 'tuple'>
```

Sebagian besar dari operator yang bekerja di list dapat bekerja juga pada tuple. Hanya saja karena sifatnya yang immutable maka elemen tuple tidak dapat diubah setelah didefiniskan. Satu-satunya cara adalah mengganti suatu elemen tuple dengan elemen lain.

PROGRAM PYTHON

```
# Penggunaan Indeks
tpl6 = ("a", "b", "c", "d", "e")
print("tpl6[0] =",tpl6[0])
print("tpl6[1:3] =",tpl6[1:3])

# Mengganti Elemen
tpl6 = ("Z",) + tpl6[1:]
print("tpl6 metode ganti elemen =",tpl6)

# Mengganti Indeks --> ERROR
tpl6[0] = "A"
print("tpl6 metode indeks =",tpl6)
```

OUTPUT

```
tpl6[0] = a
tpl6[1:3] = ('b', 'c')
tpl6 metode ganti elemen = ('Z', 'b', 'c', 'd', 'e')
TypeError: 'tuple' object does not support item assignment
```

Membandingkan Tuple

Operator perbandingan dapat bekerja pada tuple dan tipe data sekuensial lainnya seperti list, dictionary, dan set. Python akan membaca elemen pertama dari tiap sekuensial dan membandingkannya, jika ditemukan adanya kesamaan maka Python akan lanjut membaca elemen berikutnya dan begitu seterusnya hingga ditemukan perbedaan.

```
tpla = (0,1,2)
tplb = (0,3,4)
check = tpla < tplb
print(check)           OUTPUT = True
```

```

tpla = (0,1,2000000)
tplb = (0,3,4)
check = tpla < tplb
print(check)           OUTPUT = True

```

Fungsi (sort) pada Python juga bekerja dengan cara yang sama:

1. Decorate – urutan (sekuensial) membangun daftar tuple dengan satu atau lebih key pengurutan sebelum elemen dari urutan (sekuensial).
2. Sort – list tuple menggunakan sort (fungsi bawaan di python).
3. Undercorate – melakukan ekstraksi pada elemen yang telah diurutkan pada satu sekuensial

Penugasan Tuple

Python memiliki keunikan dalam hal penanganan tuple, di mana kita dapat melakukan penugasan multi variabel dalam satu baris kode. Misalnya, jika kita memiliki dua elemen dalam suatu urutan, kita dapat menugaskan elemen pertama dan kedua dari urutan tersebut ke dalam variabel x dan y. Ini adalah salah satu fitur yang membuat Python menjadi bahasa pemrograman yang efisien dan mudah digunakan.

PROGRAM 1

```

m = [ 'have', 'fun' ]
x, y = m
print("x =",x)           x = have
print("y =",y)           OUTPUT = y = fun

```

PROGRAM 2

```

m = [ 'have', 'fun' ]
x = m[0]
y = m[1]
print("x =",x)           x = have
print("y =",y)           OUTPUT = y = fun

```

PROGRAM 3

```

email = 'danieljodan@ti.ukdw.ac.id'
username, domain = email.split('@')
print("username =",username)      username = danieljodan
print("domain =",domain)         OUTPUT = domain = ti.ukdw.ac.id

```

Catatan Khusus = Jumlah variable sisi kiri dan kanan harus sama

PROGRAM 4

```
a,b = 1,2,3  
print(a,b)
```

OUTPUT =

```
ValueError: too many values to unpack (expected 2)
```

Dictionaries dan Tuple

Dictionaries mempunyai metode yang disebut items untuk mengembalikan nilai list dari tuple dimana tiap tuple-nya merupakan key-value pair. Namun seperti dictionary pada umumnya, item yang dikembalikan nilainya tidak berurutan. Hal ini tidak menjadi masalah karena bentuknya adalah list dan tuple bisa dibandingkan maka kita dapat melakukan pengurutan (sort) pada tuple. Melakukan konversi dictionary dari list tuple merupakan cara untuk menampilkan isi dictionary yang diurutkan berdasarkan kuncinya.

PROGRAM

```
d = {'a':10, 'b':1, 'c':22}  
t = list(d.items())  
t.sort()  
print("t sorted =",t)
```

OUTPUT

```
t sorted = [('a', 10), ('b', 1), ('c', 22)]
```

Multi Penugasan Dengan Dictionaries

Kombinasi antara items, tuple assignment dan for akan menghasilkan kode tertentu pada keys dan values dari dictionary dalam satu loop.

PROGRAM

```
d = {'a':10, 'b':1, 'c':22}  
l = list()  
for key, val in d.items():  
    l.append((val, key))  
print("l =",l)  
  
l.sort(reverse=True)  
print("l reverse =",l)
```

OUTPUT

```
l = [(10, 'a'), (1, 'b'), (22, 'c')]
l reverse = [(22, 'c'), (10, 'a'), (1, 'b')]
```

Tuple Sebagai Kunci Dictionaries

Tuple merupakan hashable dan list tidak. Ketika kita ingin membuat composite key yang digunakan dalam dictionary, kita dapat menggunakan tuple sebagai key. Misalnya menggunakan composite key jika ingin membuat direktori telepon yang memetakan dari pasangan last-name, first-name ke nomor telepon. Dengan asumsi bahwa kita telah mendefinisikan variabel last, first, dan nomor.

PROGRAM

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

OUTPUT

```
dida nendya 088112266
```

REFERENSI

[Python Dasar: Tipe Data Tuple](#) | Jago Ngoding

BAGIAN 2: LATIHAN MANDIRI (60%)

Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

LINK GITHUB = https://github.com/danieljodan/PrakAIPro11_A_71230970.git

SOAL 1

PROGRAM LATIHAN MANDIRI 11.1

```
1 # Program untuk pengecekan apakah semua anggota tuple sama
2 def pengecekan(data):
3     if type(data) == tuple:
4         flag = True
5         for x in data:
6             if x == data[0]:
7                 flag = True
8             else:
9                 flag = False
10        else:
11            flag = False
12    print(flag)
13
14 tA= (90, 90, 90, 90)
15 pengecekan(tA)
```

OUTPUT PROGRAM

True

PENJELASAN PROGRAM

Program untuk melakukan pengecekan apakah semua anggota yang ada didalam tuple sama.

- Fungsi pengecekan(data) didefinisikan dengan satu parameter yaitu data.
- Di dalam fungsi, pertama-tama dilakukan pengecekan apakah data adalah sebuah tuple dengan type(data) == tuple.
- Jika data adalah sebuah tuple, maka variabel flag diatur menjadi True dan program akan melakukan iterasi melalui setiap elemen x dalam tuple tersebut.
- Selama iterasi, program memeriksa apakah elemen x sama dengan elemen pertama dalam tuple (data[0]). Jika sama, flag tetap True. Jika berbeda, flag diatur menjadi False.
- Jika data bukan tuple, maka flag diatur menjadi False.

SOAL 2

PROGRAM LATIHAN MANDIRI 11.2

```

1 # Program Memasukkan Identitas
2 def identitas(info):
3     print('NIM      :', info[1])
4     print('NAMA     :', info[0])
5     print('ALAMAT   :', info[2],"\n")
6
7     tplNim = tuple(info[1])
8     print("NIM:", tplNim, "\n")
9     tplNamaDepan = tuple(info[0].split()[0])
10    print("NAMA DEPAN:", tplNamaDepan, "\n")
11    tplNamaTerbalik = tuple(info[0].split()[::-1])
12    print("NAMA TERBALIK:", tplNamaTerbalik)
13
14 data = ('Matahari Bhakti Nendya', '22064091', 'Bantul, DI Yogyakarta')
15 identitas(data)

```

OUTPUT PROGRAM

```

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

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

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

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

```

PENJELASAN PROGRAM

- Program Python ini mendefinisikan sebuah fungsi bernama `identitas` yang menerima satu parameter `info`. Parameter `info` ini diasumsikan sebuah list atau tuple yang berisi tiga elemen: nama, NIM (Nomor Induk Mahasiswa), dan alamat.
- Fungsi `identitas(info)`: Fungsi ini mencetak NIM, NAMA, dan ALAMAT dari list `info` yang diberikan sebagai input.
- Membuat tuple dari NIM: `tplNim = tuple(info[1])` mengubah string NIM menjadi tuple yang berisi setiap karakter dari NIM sebagai elemen terpisah.
- Membuat tuple dari nama depan: `tplNamaDepan = tuple(info[0].split()[0])` mengambil nama depan (elemen pertama dari nama yang dipisahkan oleh spasi) dan mengubahnya menjadi tuple yang berisi setiap karakter dari nama depan sebagai elemen terpisah.
- Membuat tuple dari nama terbalik: `tplNamaTerbalik = tuple(info[0].split()[::-1])` membalik urutan nama dan mengubahnya menjadi tuple yang berisi setiap kata dari nama yang telah dibalik sebagai elemen terpisah.

SOAL 3

PROGRAM LATIHAN MANDIRI 11.3

```

1 # Program penghitung distribusi jam dimana ada email masuk
2 def distribusi(data):
3     count = dict()
4     for line in data:
5         line = line.strip()
6         if line.startswith("From "):
7             lineFrom = line.split()
8             jam = lineFrom[5].split(":")[0]
9             if jam not in count:
10                 count[jam] = 1
11             else:
12                 count[jam] += 1
13     for key, value in sorted(count.items()):
14         print(key, value)
15
16 file = open("mbox-short.txt", "r")
17 distribusi(file)

```

OUTPUT PROGRAM

```

04 3
06 1
07 1
09 2
10 3
11 6
14 1
15 2
16 4
17 2
18 1
19 1

```

PENJELASAN PROGRAM

Program Python ini dibuat untuk menghitung distribusi jam ketika email diterima.

- Fungsi distribusi(data):
 1. Membuat dictionary count untuk menyimpan jumlah email yang diterima per jam.
 2. Melakukan iterasi melalui setiap baris dalam data yang diberikan.
 3. Menghapus whitespace di awal dan akhir baris dengan strip().
 4. Mengecek apakah baris dimulai dengan "From ", yang menandakan awal dari data email.
 5. Jika ya, memisahkan baris tersebut menjadi list dengan split() dan mengambil bagian waktu.
 6. Dari waktu, mengambil jam menggunakan split(":")[0].
 7. Jika jam tersebut belum ada dalam dictionary count, menambahkannya dengan nilai 1.
 8. Jika sudah ada, menambahkan nilai tersebut dengan 1.

- Setelah semua data diproses, fungsi akan mengurutkan dan mencetak pasangan kunci-nilai dalam count berdasarkan jam secara ascending.
- Di luar fungsi, program membuka file “mbox-short.txt” yang berisi data email.
- Fungsi distribusi dipanggil dengan file tersebut sebagai argumen.