



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>71210695</b>
<b>Nama Lengkap</b>	<b>Cahaya Sampebua</b>
<b>Minggu ke / Materi</b>	<b>11/Tipe Data Tuples</b>

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%)

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.

### *Tuple Immutable*

Data jenis tuple memiliki sifat *immutable*, yaitu tidak bisa merubah anggota yang ada dalam tuple. Nilai tuple dapat berubah. Jadi, jika ingin mengubah anggota dalam tuple harus sekaligus diganti semua nilainya.

Contoh

```
tupel = (1, 2, 3)
tupel[1] = 10
```

akan menghasilkan output error.

```
TypeError                                Traceback (most recent call last)
<ipython-input-1-17d311b7a498> in <cell line: 2>()
      1 tupel = (1, 2, 3)
----> 2 tupel[1] = 10

TypeError: 'tuple' object does not support item assignment
```

Beda halnya jika

```
tupel = (1, 2, 3)
tupel = tupel[0],10,3

print(tupel)
```

akan menghasilkan output

```
(1, 10, 3)
```

Cara penulisan tuple

```
tupel = 1,2,3
print(type(tupel))

# dapat juga menggunakan tanda kurung
```

```
print(type(tupel))
```

akan menghasilkan output

```
<class 'tuple'>
```

Tuple dengan satu elemen ditambahkan koma (",") di belakang nilai karena jika tidak, tidak akan dianggap sebagai tuple.

```
tupel = (1,)
print(type(tupel))

tupel = (1)
```

```
<class 'tuple'>
<class 'int'>
```

Metode pengaksesan tuple sama dengan metode pengaksesan list, yaitu dengan cara:

tuple[index] – untuk mengakses salah satu elemen dalam tuple

tuple[a:b] – untuk mengakses elemen tuple dengan range a sampai b, atau untuk mengakses elemen tuple dari a sampai b-1.

Tuple tidak memiliki sintaks untuk menghapus salah satu dari anggotanya. Alasannya adalah karena sifatnya yang immutable. Jika ingin menghapus anggota dalam tuple, salah satu caranya adalah dengan mengubahnya menjadi list terlebih dahulu.

```

>>> tup = (1,2,3,4,5,6)
>>> tup = list(tup)
>>> tup
[1, 2, 3, 4, 5, 6]
>>> tup.remove(4)
>>> tup
[1, 2, 3, 5, 6]
>>> tup = tuple(tup)
>>> tup
(1, 2, 3, 5, 6)

```

Setelah itu dapat menghapus anggota yang dikehendaki dalam list tuple tersebut dengan perintah yang sudah ada, lalu mengubah kembali list tersebut menjadi tuple.

```

>>> t = (1,2,3,4,5,6)

```

Ubah jadi list

```

>>> t = list(t)

```

1. remove(n) – untuk menghapus elemen yang bernilai n

```

>>> t.remove(3)
>>> t = tuple(t)
>>> t
(1, 2, 4, 5, 6)

```

2. pop(n) – untuk menghapus value ke-n dan mendapatkan nilainya

```

>>> t.pop(3)
4

```

3. del list[n]– untuk menghapus index ke-n

```

>>> del t[0:3]
>>> t = tuple(t)
>>> t
(4, 5, 6)

```

4. clear() – untuk menghapus semua element

```

>>> t.clear()
>>> t = tuple(t)
>>> t
()

```

Tuple juga tidak memiliki perintah sort. Alasannya adalah karena perintah sort akan mengubah semua anggota dalam tuple, sedangkan tuple bersifat immutable.

```
>>> t = ("j","e","r","a","p","a","h")
>>> t.sort()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'tuple' object has no attribute 'sort'
```

Untuk mengurutkan data dalam tuple, salah satu caranya adalah dengan menggunakan sorted(tuple). Karena hasil yang dikembalikan adalah dalam bentuk list, kita harus mengubahnya kembali menjadi bentuk tuple.

```
>>> t = ("j","e","r","a","p","a","h")
>>> t = sorted(t)
>>> t
['a', 'a', 'e', 'h', 'j', 'p', 'r']
>>> t = tuple(t)
>>> t
('a', 'a', 'e', 'h', 'j', 'p', 'r')
```

Perintah list yang hasilnya mengubah elemen tidak akan berlaku pada tuple.

### Membandingkan *Tuple*

Cara kerja perbandingan pada tuple menggunakan model sekuensial, yaitu dengan cara mencari elemen pertama yang berbeda dalam tuple, lalu membandingkannya.

```
>>> ("R","a","d","a") < ("R","o","d","a")
True
```

Cara kerjanya adalah Python membandingkan elemen pertama dalam kedua tuple dan membandingkannya. Karena nilainya sama ("R" dan "R"), maka berlanjut ke elemen berikutnya. Elemen berikutnya adalah "a" dan "o". Karena berbeda, maka kedua elemen ini akan dibandingkan dan akan langsung menghasilkan True atau False. Karena "o" lebih besar dari "a", output bernilai True.

## Penugasan *Tuple*

Di dalam bahasa Python, kita dapat memberi nilai kepada beberapa variable sekaligus dengan menggunakan tuple. Misal kita mau memberi nilai kepada 5 variabel.

```
>>> a, b, c, d, e = 1, 2, 3, 4, 5
>>> a
1
>>> b
2
>>> c
3
>>> d
4
>>> e
5
```

Atau bisa menggunakan list:

```
>>> list = [1,2,3,4,5]
>>> a, b, c, d, e = list
>>> a
1
>>> b
2
>>> c
3
>>> d
4
>>> e
5
```

Tuple dapat digunakan juga untuk menukar nilai dari variable.

```
>>> list = [1,2,3,4,5]
>>> a, b, c, d, e = list
>>> a
1
>>> b
2
>>> c
3
>>> d
4
>>> e
5
>>> a, b, c, d, e = e, d, c, b, a
>>> a
5
>>> b
4
>>> c
3
>>> d
2
>>> e
1
```

Satu hal yang harus diperhatikan adalah jumlah variabel di sisi kiri dengan jumlah nilai di sisi kanan harus sama.

```
>>> a, b, c = 1, 2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: not enough values to unpack (expected 3, got 2)
```

### *Dictionary and Tuple*

items dalam dictionary dapat kita jadikan list yang setiap index nya berisi tuple yang berisi key dan value dari dictionary.

```
# Membuat dictionary
dictionary = {"Cahaya": ["20", "Sulawesi Selatan"], "Setyo": ["21", "Lampung"], "Kaleb": ["20", "Tegal"]}

# Mengubah dictionary menjadi list
tupel = list(dictionary.items())

# Outputnya
print(tupel)
```

```
[('Cahaya', ['20', 'Sulawesi Selatan']), ('Setyo', ['21', 'Lampung']), ('Kaleb', ['20', 'Tegal'])]
```

## 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.

Lin GitHub → <https://github.com/cahaya-sampebua/Latihan-Mandiri-PrakALPro-11.git>

### SOAL 1

```
def cek_tuple(data):  
    val = data[0]  
    for i in data:  
        if i == val:  
            status = True  
        else:  
            status = False  
            break  
  
    return status
```

### Output

True

### Penjelasan

Gunakan acuan index pertama dari data untuk dibandingkan, masukkan sebagai variabel val. Untuk setiap index dalam data, jika semua index bernilai sama dengan val, output akan menunjukkan hasil True. Jika ada salah satu saja index yang tidak sama dengan val, program akan langsung keluar dari perulangan dan menampilkan hasil False.



## SOAL 2

```
def data_diri(data):
    for i in data:
        if i.isdigit():
            nim = i

        elif "," in i:
            alamat = i

        else:
            nama = i.split(" ")
            nama_depan = nama[0]
            nama.reverse()

    print(f"NIM: {tuple(nim)}")
    print(f"NAMA DEPAN: {tuple(nama_depan[1:])}")
    print(f"NAMA TERBALIK: {tuple(nama)}")
```

### Output

```
NIM: ('2', '2', '0', '6', '4', '0', '9', '1')
NAMA DEPAN: ('a', 't', 'a', 'h', 'a', 'r', 'i')
NAMA TERBALIK: ('Nendya', 'Bhakti', 'Matahari')
```

### Penjelasan

Untuk setiap index dalam data, jika index adalah angka, maka index tersebut adalah NIM. Jika ada tanda koma “,” dalam index, maka itu adalah alamat (asumsi setiap alamat ada tanda koma karena menunjukkan daerah dan provinsi). Selain itu, index tersebut adalah nama.

Nama diubah menjadi bentuk list yang berisi setiap kata pada nama. Buat variabel baru yaitu nama depan, dimana nama depan adalah index pertama dari list nama. Lalu, list nama dibalik urutannya.

Output pertama menampilkan nim dalam bentuk tuple yang berisi setiap digit. Output kedua menampilkan tuple yang berisi setiap huruf pada nama depan, kecuali yang kapital. Output ketiga menghasilkan tuple yang berisi nama yang sudah dibalik.

### SOAL 3

```
import re

nama_file = input("Enter a file name: ")
file = open(nama_file)

baca_file = file.read()

re_file = re.findall("From .+ \d+ (\d+):.+", baca_file)
re_file.sort()

dict_jam = dict()
for i in re_file:
    if i not in dict_jam:
        dict_jam[i] = 1
    else:
        dict_jam[i] += 1
```

### Output

```
Enter a file name: mbox-short.txt
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

Read semua file dengan `.read( )`, lalu buat list jam dengan menggunakan `re.findall( )`. Regex mencari baris yang ada kata From, dan hanya mengcapture jamnya, yaitu 2 angka pertama sebelum tanda titikdua (**04**:12:47) dan urutkan mulai dari angka terkecil. Buat dictionary baru.

Untuk setiap index dalam file, jika index belum ada dalam dictionary, masukkan index ke dalam dictionary dan beri value 1. Jika sudah ada dalam dictionary, tambah value dari key tersebut dengan 1.