



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>71231042</b>
<b>Nama Lengkap</b>	<b>Revaldo Fransisco Hohary</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%)

### MATERI 1

#### A. Tuple Immutable

Tuple merupakan model urutan sekuensial yang dalam penggunaannya hampir sama dengan list. Dalam tuple memiliki elemen dengan indeks yang dimulai dengan 0. Tuple tidak bisa diubah, tuple dapat dibandingkan (comparable). Jika sebuah objek memiliki nilai hash yang tidak pernah berubah selama hidupnya (menggunakan method `__hash__()`), dan dapat dibandingkan dengan benda-benda lain (menggunakan method `__eq__()` atau `__cm__()`) itu biasanya disebut hashable. Tuple dapat dimasukkan dalam list sebagai nilai kunci (key value) pada dictionary.

##### a. Sintaks Penulisan

Sintaks penulisan tuple :

```
>>> t1 = 'd','b','c','f','e'
```

Bisa juga jika menggunakan tanda kurung.

```
>>> t1 = ('d','b','c','f','e')
```

Dalam penulisan tuple harus ada penggunaan tanda koma. Sedangkan ketika tidak menggunakan tanda koma maka, komputer akan menganggap itu string.

```
>>> t2 = ('z')
>>> type(t2)
<type 'str'>
```

Dalam penggunaan lain bisa juga tuple tidak diisi oleh argument apapun, dan tuple tersebut akan berbentuk tuple kosong.

```
>>> t2 = tuple()
>>> type(t2)
()
```

Jika dalam argument tersebut ada string, list atau tuple, maka akan mengembalikan nilai tuple dengan elemen yang berurutan.

```
>>> t = tuple('informatika')
>>> print(t)
('i','n','f','o','r','m','a','t','i','k','a')
```

Pada tuple juga kita bisa memanggil argument dalam tuple tersebut dengan kurung kotak []

```
>>> t = ('s', 'a', 'b', 'a', 'r')
>>> print(t[0])
's'
```

Untuk menampilkan rantang nilai :

```
>> print(t[1:3])  
( 'a', 'b' )
```

Jika ada yang berusaha mengubah tuple maka akan muncul **“TypeError: object doesn't support item assignment”**. Pada tuple tidak dapat dirubah dan diganti oleh elemen lain.

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

## B. Membandingkan Tuple

Pada operator perbandingan (compare) ini dapat bekerja pada tuple dan model sekuensial lainnya seperti list, dictionary, set). Dalam penggunaannya ini dengan membandingkan elemen utama dari setiap sekuensial, jika terdapat kesamaan pada elemen tersebut maka akan berlanjut pada elemen selanjutnya dan proses ini berlangsung terus menerus sampai menemukan perbedaan pada elemen tersebut.

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

Fungsi sort berkerja dengan cara sama juga seperti python. Pada pertama kali bekerja akan mengurutkan pada elemen yang pertama, dan pada test tertentu akan mengurutkan dari elemen yang kedua atau sebagainya. Biasanya fitur ini disebut DSU ( Decorate, Sort, Undercorate).

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.

Untuk memahami hal tersebut berikut ada sebuah contoh pengurutan kata dalam kalimat, dimulai dari yang paling panjang ke paling pendek.

```
klmt = 'but soft what light in yonder window breaks'  
daftarkat = klmt.split()  
t = list()  
for kata in daftarkata:  
    t.append((len(kata), kata))  
t.sort(reverse=True)  
  
urtan = list()  
for length, kata in t:  
    urtan.append(kata)
```

```
print(urutan)
```

Looping yang pertama akan membuat daftar tuple, yang berisi daftar kata sesuai dengan panjangnya. Fungsi sort akan membandingkan elemen pertama dari list dari panjang kata yang ada dan kemudian akan menuju ke elemen kedua jika kondisi sesuai. Keyword reverse=True digunakan untuk melakukan urutan secara terbalik (descending).

Looping kedua akan membangun daftar tuple dalam sesuai urutan alfabet diurutkan berdasarkan panjangnya. Pada contoh diatas Empat kata dalam kalimat akan diurutkan secara alfabet terbalik. Kata "what" akan muncul sebelum "soft" didalam list.

Output dari program tersebut sebagai berikut :

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

Kata-kata yang muncul diurutkan dalam bentuk list dengan urutan kata yang panjang ke yang paling kecil

### C. Penugasan Tuple

Python memiliki kemampuan untuk memiliki tuple di sisi kiri statement penugasan, yang memungkinkan untuk menetapkan lebih dari satu variabel secara berurutan di sisi kiri.

Sebagai contoh, kita akan mencoba menetapkan elemen dari urutan dua daftar ke dalam variabel x dan y dalam satu pernyataan.

```
>>> m = [ 'have', 'fun' ]  
>>> x, y = m  
>>> x  
'have'  
>>> y  
'fun'  
>>>
```

Dalam langkah-langkah berikut, Python akan menterjemahkan sintaks tuple:

```
>>> m = [ 'have', 'fun' ]  
>>> x = m[0]  
>>> y = m[1]  
>>> x  
'have'  
>>> y  
'fun'  
>>>
```

Contoh di atas menggunakan statement penugasan disebelah kiri dan tidak menggunakan tanda kurung. Contoh yang sama dengan menggunakan tanda kurung (parentheses) diberikan di bawah ini.

```
>>> m = [ 'have', 'fun' ]  
>>> (x, y) = m  
>>> x
```

```
'have'
>>> y
'fun'
>>>
```

Tuple memungkinkan pula untuk menukar nilai variabel dalam satu statement.

```
>>> a, b = b, a
```

Contoh di atas menunjukkan bahwa kedua statemen adalah tuple. Masing-masing variable di sebelah kiri menerima atau diberi nilai dari tuple di bagian kanan, sedangkan variabel di sebelah kiri menerima nilai dari tuple di bagian kiri. Semua ekspresi di kiri dievaluasi sebelum ditugaskan.

Yang harus diperhatikan adalah bahwa jumlah variabel di kedua sisi harus sama.

```
>>> a, b = 1, 2, 3
ValueError: too many values to unpack
```

Data sekuensial string, list, atau tuple biasanya berada di sisi kanan. Sebagai contoh, kita akan membagi alamat email menjadi domain dan user name seperti yang ditunjukkan di bawah ini.

```
>>> email = 'didanendya@ti.ukdw.ac.id'
>>> username, domain = email.split('@')
```

Nilai yang dihasilkan dari split terdiri dari dua elemen yang terpisah dengan tanda "@"; elemen pertama mengandung username, dan elemen kedua mengandung domain.

```
>>> print(username)
didanendya
>>> print(domain)
ti.ukdw.ac.id
```

#### D. Dictionaries and Tuple

Dictionaries menggunakan metode yang disebut items untuk mengembalikan nilai daftar dari tuple yang masing-masing berisi pasangan nilai dan kunci.

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

Nilai item yang dikembalikan tidak berurutan, seperti yang terlihat dalam dictionary biasa. Bagaimanapun juga, melakukan konversi dictionary dari list tuple menampilkan isi dictionary yang diurutkan berdasarkan kuncinya, karena list tuple membandingkannya sehingga kita dapat melakukan pengurutan (sort) pada tuple.

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

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

#### E. Multipenugasan Dengan Dictionaries

Dalam satu loop, kombinasi item, tuple assignment, dan for akan menghasilkan kode khusus untuk tombol dan nilai dictionary.

```
for key, val in list(d.items()):
    print(val, key)
```

bagian looping ini, item mengembalikan list dari tuple, dan key. val adalah penugasan tuple yang melakukan iterasi berulang melalui pasangan key-value pada dictionary.

Key dan value akan bergerak maju menuju pasangan key-value berikutnya di dictionary pada setiap iterasi loop. Mereka tetap dalam urutan hash. Output dari loop sebelumnya

```
10 a
22 c
1 b
```

Dengan menggabungkan kedua metode di atas, kita dapat mencetak isi dictionary yang diurutkan berdasarkan nilai yang disimpan di setiap pasangan nilai kunci. Langkah pertama adalah membuat daftar tuple di mana masing-masing tuple beranggotakan (value,key). Metode item akan membuat list tuple dengan value, key, dan melakukan pengurutan berdasarkan value, bukan key. Setelah daftar list dengan tuple value-key terbentuk, langkah berikutnya adalah mengurutkan list tersebut dengan urutan terbalik dan mencetak list baru yang disortir.

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

Dengan menyusun daftar tuple dengan value sebagai elemen pertama, Mengurutkan daftar tuple dan mendapatkan isi dictionary yang akan menjadi tugas yang mudah. diurutkan menurut nilainya.

#### F. Kata yang sering muncul

Kita akan mencoba menampilkan kata-kata yang sering muncul dalam teks Act 2, Scene 2, dari Romeo and Juliet pada bagian ini. Silakan download romeo-full.txt di bagian materi. Kami akan mencoba membuat program dengan list tuple yang menampilkan sepuluh kata yang paling sering digunakan.

```

import string
fhand = open('romeo-full.txt')
counts = dict()
for line in fhand:
    line = line.translate(str.maketrans('', '',
string.punctuation))
    line = line.lower()
    words = line.split()
    for word in words:
        if word not in counts:
            counts[word] = 1
        else:
            counts[word] += 1
# urutkan dictionary by value
lst = list()
for key, val in list(counts.items()):
    lst.append((val, key))
lst.sort(reverse=True)
for key, val in lst[:10]:
    print(key, val)

```

Bagian pertama program digunakan untuk membaca file dan melakukan komputasi terhadap dictionary, yang memetakan setiap kata ke sejumlah kata yang tidak berubah dalam dokumen. List tuple (val, key) dibuat dan diurutkan dalam list dengan urutan dengan menambahkan print out counts.

Untuk membandingkan, nilai pertama digunakan. Jika ada lebih dari satu tupel yang memiliki nilai yang sama, elemen kedua—atau tombol—akan dilihat, sehingga tuple yang memiliki nilai yang sama akan diurutkan berdasarkan urutan abjad sesuai tombol.

Pada bagian terakhir, looping melakukan penugasan ganda sekali lagi dan mencetak sepuluh kata yang paling sering muncul dengan melakukan perulangan pada list lst:10.

Output dari program tersebut adalah :

```

61 I
42 and
40 romeo
34 to
34 the
32 thou
32 juliet
30 that
29 my
24 thee

```

Dengan 19 baris program Python yang dipahami, penguraian dan analisis data yang kompleks ini dapat dilakukan. Ini adalah salah satu alasan mengapa python adalah bahasa yang bagus untuk mengeksplorasi data.

## G. Tuple Sebagai Kunci Dictionaries

Tuple dapat dihash, sementara list tidak. Dengan demikian, jika kita ingin membuat kunci kombinasi yang dapat digunakan dalam dictionary, kita dapat menggunakan tuple sebagai kunci. Salah satu contohnya adalah jika kita ingin membuat direktori telepon yang memetakan.

Dari pasangan nama terakhir dan nama pertama ke nomor telepon, dengan asumsi kita telah menentukan variabel nama terakhir, pertama, dan nomor. Pernyataan penugasan harus ditulis dalam dictionary sebagai berikut:

```
directory[last,first] = number
```

Tuple adalah ekspresi yang ditemukan di dalam kurung kotak. Selanjutnya, berikan penugasan tuple pada looping for yang terkait dengan kamus.

```
for last, first in directory:  
    print(first, last, directory[last,first])
```

Tuple berfungsi sebagai looping yang berhubungan dengan tombol di direktori sebelumnya. Setelah elemen pertama dari masing-masing tuple ditetapkan, nama dan nomor telepon yang sesuai dicetak.

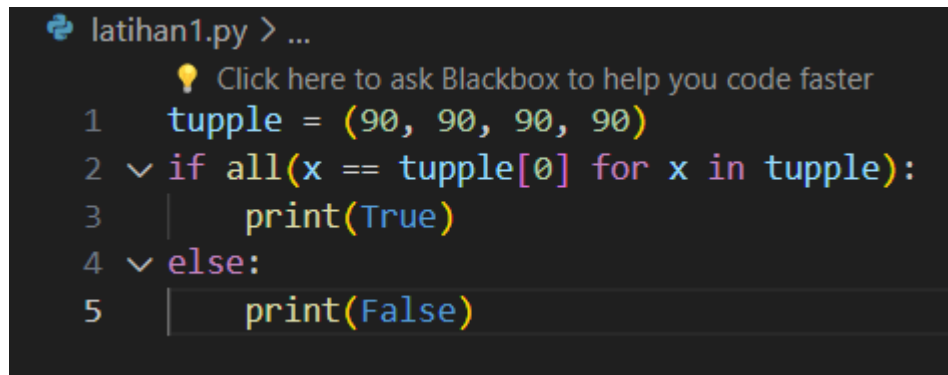
```
>>> last = 'adlo'  
>>> first = 'hoha'  
>>> number = '088112277'  
>>> directory = dict()  
>>> directory[last, first] = number  
>>> for last, first in directory:  
...     print(first, last, directory[last,first])  
...  
adlo hoha 088112277  
>>>
```



## BAGIAN 2: LATIHAN MANDIRI (60%)

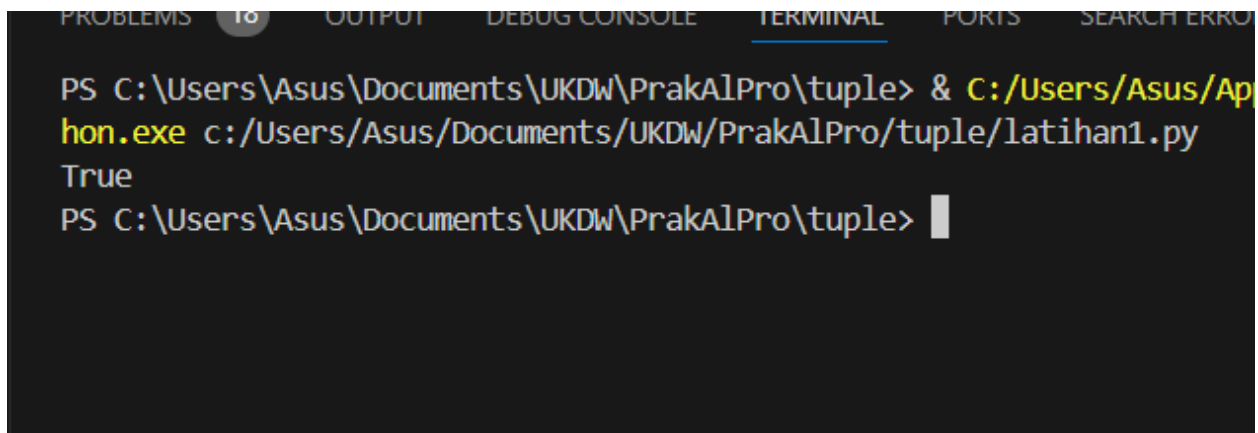
### SOAL 1

Pada soal berikut ini kita akan mengecek apakah semua anggota yang ada dalam tuple yang sama. Gambar berikut bagian pertama kita membuat sebuah variable bernama tuple yang berisi empat 90, dan selanjutnya untuk mengecek semua itu kita memerlukan if jika x sama dengan tuple[0] dan selanjutnya akan mengulangi x di dalam tuple jika sama semua maka akan print true tapi jika tidak maka false



```
latihan1.py > ...  
  Click here to ask Blackbox to help you code faster  
1  tuple = (90, 90, 90, 90)  
2  if all(x == tuple[0] for x in tuple):  
3      print(True)  
4  else:  
5      print(False)
```

Gambar 11.1 : kode Pemograman



```
PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERRORS  
PS C:\Users\Asus\Documents\UKDW\PrakAlPro\tuple> & C:/Users/Asus/AppData/Local/Programs/Python/Python39-64/Python.exe c:/Users/Asus/Documents/UKDW/PrakAlPro/tuple/latihan1.py  
True  
PS C:\Users\Asus\Documents\UKDW\PrakAlPro\tuple> |
```

Gambar 11.2 : output hasil dari program diatas

### SOAL 2

Pada bagian Ini akan pertama saya akan membuat variable data\_diri untuk selanjutnya akan print data\_diri perindeks supaya dapat memunculkan apa yang kita mau untuk dipanggil. Pada bagian selanjutnya nim akan dibuatg dalam tuple dan selanjutnya menggunakan print untuk memunculkannya, selanjutnya untuk nama juga buat seperti itu. Dan terakhir akan membalik nama yaitu dengan reversed.

```

latihan2.py > ...
  Click here to ask Blackbox to help you code faster
1  data_diri = ('Revaldo Fransisco Hohary', 71231042, 'Biak, Papua')
2
3  print('Data:', data_diri)
4  print('NIM :', data_diri[1])
5  print('NAMA', data_diri[0])
6  print('ALAMAT', data_diri[2])
7
8  nim = tuple(str(data_diri[1]))
9  print('NIM (2, 2; 0; 6;4;*0; 9,11):', nim)
10
11 nama = tuple(data_diri[0].split()[0])
12 print('NAMA DEPAN:', nama)
13
14 terbalik = tuple(reversed(data_diri[0].split()))
15 print('NAMA TERBALIK:', terbalik)

```

Gambar 11.3 : kode program

```

PS C:\Users\Asus\Documents\UKDW\PrakAlPro\tuple> & C:/Users/Asus/AppD
hon.exe c:/Users/Asus/Documents/UKDW/PrakAlPro/tuple/latihan2.py
Data: ('Revaldo Fransisco Hohary', 71231042, 'Biak, Papua')
NIM : 71231042
NAMA : Revaldo Fransisco Hohary
ALAMAT : Biak, Papua
NIM : ('7', '1', '2', '3', '1', '0', '4', '2')
NAMA DEPAN : ('R', 'e', 'v', 'a', 'l', 'd', 'o')
NAMA TERBALIK : ('Hohary', 'Fransisco', 'Revaldo')
PS C:\Users\Asus\Documents\UKDW\PrakAlPro\tuple>

```

Gambar 11.4 : Ouputnya dari program

### SOAL 3

```
1  nama = input("Masukkan nama file: ")
2  ✓ try:
3      |   file = open(nama)
4  ✓ except:
5      |   print(f"{nama} is not found!")
6      |   quit()
7
8  jam_dict = dict()
9  ✓ for garis in file:
10 ✓     |   if garis.startswith('From '):
11         |       words = garis.split()
12         |       waktu = words[5]
13         |       jam = waktu[:2]
14         |       jam_dict[jam] = jam_dict.get(jam, 0) + 1
15
16  hours_list = sorted(jam_dict.items())
17
18  ✓ for jam, count in hours_list:
19      |   print(jam, count)
```

Gambar 11.5 : kode program

```
PS C:\Users\Asus\Documents\UKDW\PrakAlPro\tuple> & C:/Users/Asus/A  
hon.exe c:/Users/Asus/Documents/UKDW/PrakAlPro/tuple/latihan3.py  
Masukkan nama file: 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  
PS C:\Users\Asus\Documents\UKDW\PrakAlPro\tuple> █
```

Gambar 11.6 : Output dari program diatas

LINK GITHUB : <https://github.com/Frealy0901/Tugas-PrakAlPro12.git>