



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	Samuel Natanael
<b>Nama Lengkap</b>	71231050
<b>Minggu ke / Materi</b>	12 / Tuple

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%)

### Tuple

Tuple hampir sama dengan list karena keduanya bisa menyimpan nilai apa pun dan diakses menggunakan indeks angka. Namun, perbedaan utamanya adalah sifat tuple yang tidak dapat diubah (immutable). Artinya, elemen-elemennya tidak dapat diubah setelah tuple dibuat. Tuple juga dapat dibandingkan satu sama lain (compare) dan bersifat hashable

Sebuah objek disebut hashable jika nilai hashnya tidak berubah selama hidupnya dan dapat dibandingkan dengan objek lain (menggunakan method '.\_\_hash\_\_()' ). Hal ini membuat objek dapat digunakan sebagai kunci dalam kamus dan diatur dalam struktur data yang menggunakan nilai hash secara internal, seperti dictionary. Objek yang sudah ada dalam Python biasanya bersifat hashable, sedangkan tipe yang tidak dapat diubah, seperti list atau dictionary, tidak bersifat hashable. Objek yang merupakan instance dari kelas yang didefinisikan pengguna biasanya bersifat hashable secara default, dengan nilai hash yang dihasilkan dari id() mereka.

Struktur Data	Simbol	Sifat	Data Duplikat	Akses Data
List	[values]	mutable	tidak dihilangkan	value dapat diakses melalui index
Tuple	(values)	immutable	tidak dihilangkan	value dapat diakses melalui index
Set	{values}	mutable/immutable	dihilangkan	value tidak dapat diakses
Dictionary	{'key':value}	mutable	tidak dihilangkan (jika value yang duplikat)	value dapat diakses melalui key

Gambar 12.1 Perbedaan Setiap Struktur Data (diambil dari <https://muhammad-efan.medium.com/python-struktur-data-b4a027ec58ed>)

```
# Cara penulisan tuple:
tple = '1','2','3' #1
tple = ('1','2','3') #2

# Untuk menuliskan tuple dengan satu elemen ditambahkan koma
tple = "1",
print(type(tple))

<class 'tuple'>
```

```

# Jika tidak ada koma maka akan menjadi string
tple = '1'
print(type(tple))

<class 'str'>

# Jika argumennya berupa string, list, atau tuple, akan menghasilkan tuple baru
yang berisi elemen-elemen dari urutan tersebut dengan urutan yang sama.
tple = tuple('dutawacana')
print(tple)

('d', 'u', 't', 'a', 'w', 'a', 'c', 'a', 'n', 'a')

# Tuple adalah nama dari konstruktor dalam Python. Kita tidak dapat menggunakan
"tuple" sebagai nama variabel karena itu akan menimbulkan konflik dengan nama
bawaan. Sebagian besar operator yang berfungsi pada list juga berlaku pada tuple.
Tanda kurung kotak "[ ]" digunakan untuk menunjukkan indeks elemen dalam tuple,
meskipun sebenarnya digunakan pada list

tple = tuple('dutawacana')
print(tple[0])
d

# Untuk menampilkan rentang nilai dari element tuple dapat menggunakan:
tple = tuple('dutawacana')
print(tple[1:5])
('u', 't', 'a', 'w')

# Tuple bersifat immutable, yang berarti elemen-elemennya tidak dapat diubah
setelah tuple dibuat. Jika Anda mencoba mengubah tuple, Anda akan mendapatkan
sebuah error.
tple = tuple('dutawacana')
tple[0] = 'a'
~~~~^^^
TypeError: 'tuple' object does not support item assignment

# Anda tidak bisa merubah elemen pada sebuah tuple, namun Anda bisa membuat tuple
baru dengan elemen yang diubah atau diganti sesuai kebutuhan.
tple = ('A',) + tple[1:]
print(tple)

('A', 'u', 't', 'a', 'w', 'a', 'c', 'a', 'n', 'a')

```

## Membandingkan Tuple

Untuk tuple dan model sekuensial lainnya, seperti list, dictionary, dan set, operator perbandingan membandingkan elemen pertama dari setiap sekuensial. Ke elemen berikutnya akan dilanjutkan jika ada kesamaan. Jika ada perubahan, proses ini berhenti.

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

```
True
```

```
>>> (0,12,30,50) < (1,2,3,4)
```

```
True
```

Fungsi (sort) dalam Python umumnya beroperasi dengan pola yang konsisten. Langkah awalnya adalah melakukan pengurutan berdasarkan elemen pertama dari setiap item. Namun, ada situasi khusus di mana pengurutan dilakukan berdasarkan elemen lain seperti elemen kedua, ketiga, dan seterusnya. Proses ini dikenal dengan sebutan DSU - Decorate, Sort, Undercorate. Dalam DSU, urutan awalnya didekorasi (decorate) dengan elemen-elemen dari urutan tersebut, kemudian daftar yang telah didekorasi diurutkan (sort), dan akhirnya elemen-elemen yang telah diurutkan diekstraksi kembali (undercorate) ke dalam urutan semula.

Berikut adalah contoh code untuk mengurutkan kata dari yang Panjang sampai pendek

```
kalimat = 'aaa aaaa aaaaaaa aaaaaaaaaa aaaaaa'
kata = kalimat.split()
t = []

# Looping setiap kata
for x in kata:
    # Memasukan tuple yang berisikan daftar kata dan panjangnya
    t.append((len(x), x))

# sort(reverse=True) untuk mengurutkan data secara terbalik
t.sort(reverse=True)

urutan = list()
for x, y in t:
    # Looping ini untuk membuat tuple sesuai urutan yang sudah diurutkan
    urutan.append(y)

print(urutan)

['aaaaaaaaaaaa', 'aaaaaaa', 'aaaaaaa', 'aaaa', 'aaa']
```

## Penugasan Tuple

Salah satu hal unik tentang Python adalah kemampuannya untuk memungkinkan tuple di sebelah kiri dari pernyataan penugasan. Ini membuat penugasan lebih dari satu variabel secara berurutan di sisi kiri.

```
# Dua elemen berurutan akan dimasukan ke dalam variabel berbeda
x = [1,2]
y, z = x
print(y)
print(z)
1
2

# Bisa juga menggunakan tanda kurung
(y, z) = x

# Python akan menterjemahkan sintaks tuple dalam langkah-langkah sebagai berikut:
y = x[0]
z = x[1]
print(y)
print(z)
1
2

# Tuple juga bisa menukar nilai variable dalam satu statement
x = 1
y = 2
x, y = y, x
print(x)
print(y)
2
1

# Jika variabel kiri dan kanan tidak sama maka akan muncul error
x, y = 1, 2, 3
    x, y = 1, 2, 3
    ^^^^
ValueError: too many values to unpack (expected 2)

# Biasanya, pada sisi kanan terdapat data sekuensial seperti string, list, atau tuple. Sebagai contoh, kita bisa membagi alamat email menjadi username dan domain, seperti berikut ini.
email = 'samuelnatanael@gmail.com'
```

```
# Menggunakan split() untuk memisahkan username dan domain
username, domain = email.split("@")
print(f"username = {username}, domain = {domain}")
username = samuelnatanael, domain = gmail.com
```

## Dictionary dan Tuple

Pada Dictionary, terdapat metode 'items()' untuk mengembalikan nilai list dari tuple, dimana tuplenya adalah *key-value* pair

```
Dict = {"y" : 10, "x" : 20, "z" : 30}
tple = list(Dict.items())
print(tple)
[('y', 10), ('x', 20), ('z', 30)]
```

Perlu diingat bahwa list dari tuple adalah list, dan tuple dapat diurutkan. Oleh karena itu, kita dapat melakukan pengurutan pada tuple. Salah satu cara untuk menampilkan isi dictionary yang diurutkan berdasarkan kunci adalah dengan melakukan konversi dictionary ke dalam list tuple terlebih dahulu.

```
Dict = {"y" : 10, "x" : 20, "z" : 30}
tple = list(Dict.items())
tple.sort()
print(tple)
# List yang muncul diurutkan secara ascending berdasarkan alfabet dan key value.
[('x', 20), ('y', 10), ('z', 30)]
```

## 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.

### SOAL 1

```
11,1.py > ...
1  # Tuple yang dicek
2  tA = (90, 90, 9, 90)
3  |
4  # Mengubah Tuple menjadi List
5  ta = list(tA)
6  |
7  # Item pertama dalam list
8  first = ta[0]
9  con = True
10 |
11 # Looping setiap item dalam list
12 for x in ta:
13     # Jika ada satupun item yang tidak sama dengan item pertama maka akan False
14     if x != first:
15         con = False
16 print(con)
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS D:\Ngampus\Semester 2\PrakAIPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAIPro  
False

## SOAL 2

```
11,2.py > ...
1  Data = ('Samuel Natanael', '71231050', 'Kotabaru, DI Yogyakarta')
2  print(f'NIM = {Data[1]}')
3  print(f'NAMA = {Data[0]}')
4  print(f'ALAMAT = {Data[2]}')
5
6  nim = tuple(Data[1])
7  nama = Data[0].split()
8  nama_depan = nama[0]
9  nama_terbalik = nama[::-1]
10 print(nim)
11 print(tuple(nama_depan))
12 print(tuple(nama_terbalik))
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Seme
NIM = 71231050
NAMA = Samuel Natanael
ALAMAT = Kotabaru, DI Yogyakarta
('7', '1', '2', '3', '1', '0', '5', '0')
('S', 'a', 'm', 'u', 'e', 'l')
('Natanael', 'Samuel')
```



### SOAL 3

```
11,3.py > ...
1  inpt = input("Masukan Nama File = ")
2  def count_hours(input):
3      # Dict kosong
4      Dict = {}
5
6      # Membuka file
7      with open(input, "r") as file:
8          # Membaca setiap lines pada file
9          lines = file.readlines()
10
11         # Looping setiap kata dalam kalimat
12         for line in lines:
13             # Mencari kalimat waktu
14             if line.startswith("From") and len(line.split()) > 3:
15                 # Ketika kalimat sudah ditemukan maka akan displit
16                 kata = line.split()
17
18                 # Mencari jam
19                 waktu = kata[-2].split(":")
20                 hour = waktu[0]
21
22                 # Mengecek apakah jam sudah pernah ditambah ke dict
23                 if hour not in Dict:
24                     # Jika belum ada maka akan dibuat dict baru
25                     Dict[hour] = 1
26                 else:
27                     # Jika sudah ada maka akan ditambah satu
28                     Dict[hour] += 1
29
30             # Mengubah dict menjadi tuple
31             t = list(Dict.items())
32
33             # Mensortir tuple
34             t.sort()
35
36             # Looping dan print setiap tuple
37             for x, y in t:
38                 print(x, y)
39
40 count_hours(inpt)
```

```
Masukan Nama File = 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
```

Github : <https://github.com/SamuelN1508/PrakAIPro-11.git>