



---

# Laporan Praktikum Algoritma & Pemrograman

Semester Genap 2024/2025

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.

NIM	71220829
Nama Lengkap	TIMOTIUS EKANA TEJO KESUMA
Minggu ke / Materi	13 / Tipe Data Tuples

PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA  
YOGYAKARTA  
2025

## 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.

### MATERI 12.3.1

Tuple merupakan salah satu tipe data sekuensial di Python yang sangat mirip dengan list. Bedanya, tuple bersifat immutable atau sekali dibuat, elemennya tidak dapat diubah. Justru sifat inilah yang menjadikan tuple hashable, sehingga bisa dipakai sebagai kunci dalam dictionary.

```
# Contoh Penulisan Tuple:
t = ('a', 'b', 'c')
t1 = ('a',)      # Tuple satu elemen butuh koma
t2 = ('a')       # Ini string, bukan tuple
t3 = tuple('python') # ('p','y','t','h','o','n')

# Mengakses Elemen:
print(t[0])      # Mengakses elemen pertama
print(t[1:3])    # Slicing tuple

# Sifat Immutable:
t[0] = 'x'       # Akan error
t = ('X',) + t[1:]
print(t)         # Menampilkan tuple yang telah dimodifikasi
```

### MATERI 12.3.2

Tuple dapat dibandingkan menggunakan operator perbandingan seperti <, >, ==, dll. Proses perbandingan dilakukan dari elemen pertama ke berikutnya hingga ditemukan perbedaan.

```
(0, 1, 2) < (0, 3, 4)    # True
```

Konsep DSU (Decorate, Sort, Undecorate)

Teknik ini digunakan untuk menyortir elemen berdasarkan kriteria tertentu menggunakan tuple.

```
kalimat = 'but soft what light in yonder window breaks'
kata = kalimat.split()
t = [(len(k), k) for k in kata]
t.sort(reverse=True)
hasil = [k for _, k in t]
```

#### MATERI 12.3.4

Python mendukung multiple assignment dengan tuple, yaitu menetapkan beberapa variabel sekaligus dalam satu baris kode.

```
a, b = ['hello', 'world']
a, b = b, a # Menukar nilai dua variabel
```

```
email = 'mahasiswa@kampus.ac.id'
username, domain = email.split('@')
print(f'Username: {username}, Domain: {domain}') # Memisahkan email menjadi username dan domain
```

Output:

```
Username: mahasiswa, Domain: kampus.ac.id
```

#### MATERI 12.3.5

Python menyediakan method .items() pada dictionary yang mengembalikan daftar tuple berisi pasangan key-value.

```
d = {'a': 10, 'b': 1}
t = list(d.items()) # [('a', 10), ('b', 1)]
t.sort()           # Mengurutkan berdasarkan key
```

```
d = {'a': 10, 'b': 1}
t = [(v, k) for k, v in d.items()]
t.sort(reverse=True)

print(t)
```

Output:

```
[(10, 'a'), (1, 'b')]
```

Kita dapat menggabungkan for, items(), dan penugasan tuple untuk mengakses key dan value dictionary secara langsung.

```
d = {'a': 10, 'b': 1}
for key, val in d.items():
    print(val, key)
```

Output:

```
10 a
1 b
```

Dan jika ingin mengurutkan dictionary berdasarkan nilai:

```
pairs = [(val, key) for key, val in d.items()]
pairs.sort(reverse=True)
print(pairs)
```

Output:

```
[(10, 'a'), (1, 'b')]
```

## MATERI 12.3.6

Program sederhana untuk mencari kata yang paling sering muncul dalam file teks:

```
import string
fhand = open('romeo-full.txt')
counts = {}

for line in fhand:
    line = line.translate(str.maketrans('', '', string.punctuation)).lower()
    words = line.split()
    for word in words:
        counts[word] = counts.get(word, 0) + 1

lst = [(val, key) for key, val in counts.items()]
lst.sort(reverse=True)

for val, key in lst[:10]:
    print(val, key)
```

Output:

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

### MATERI 12.3.7

Karena tuple bersifat hashable, kita bisa menggunakannya sebagai composite key dalam dictionary.

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

Output:

```
dida nendya 088112266
```

### 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

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

Contoh:

```
tA= (90, 90, 90, 90)
```

Output

```
True
```

Jawab:

```
def semua_sama(t):  
    for x in t:  
        if x != t[0]:  
            return False  
    return True  
  
data = (90, 90, 90, 90)  
if semua_sama(data):  
    print("Semua anggota tuple sama")  
else:  
    print("Ada anggota tuple yang berbeda")
```

Output:

```
Semua anggota tuple sama
```

Penjelasan:

Fungsi dinamakan `semua_sama(t)` dengan 1 parameter yaitu `t`. Fungsi melakukan perulangan dimana seluruh value tuple di dalam parameter akan di cocokkan dengan `t` indeks ke 0, jika tidak cocok, maka return adalah `False`, dan jika tidak, maka return adalah `True`. Kemudian jika fungsi dipanggil dan fungsi `semua_sama()` mengeluarkan output `True`, maka output kepada user adalah "Semua anggota tuple sama", jika `False`, maka "Ada anggota tuple yang berbeda".

## SOAL 2

Buatlah program dengan menggunakan tuple yang dapat melakukan proses seperti pada kasus 11.1, Gunakan data diri anda masing-masing dan lakukan perubahan supaya didapatkan output seperti contoh berikut ini :

Contoh:

Data: ('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')

Jawab:

```
def data_diri():
    data = ('Matahari Bhakti Nendya', '22064091', 'Bantul, DI Yogyakarta')
    nim = data[1]
    nama = data[0]
    alamat = data[2]

    nim_tuple = tuple(nim)
    nama_depan = nama.split()[0]
    nama_depan_tuple = tuple(nama_depan)
    nama_terbalik = tuple(nama.split()[::-1])
    print(f'NIM: {nim}\nNAMA: {nama}\nALAMAT: {alamat}')
    print(f'NIM: {nim_tuple}')
    print(f'NAMA DEPAN: {nama_depan_tuple}')
    print(f'NAMA TERBALIK: {nama_terbalik}')

data_diri()
```

Output:

```
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:

Fungsi dinamakan data\_diri(). Fungsi ini akan memuat data dimana data akan dibaca sesuai urutan indeks 0 adalah nama, 1 adalah nim, dan 2 adalah alamat. Nama depan diambil melalui nama yang di



split kemudian diambil indeks pertama. Kemudian nim, nama depan dan nama terbalik dibuat menjadi tuple supaya output dapat langsung terpisah dengan koma atau berbentuk tuple.

### SOAL 3

Buatlah program untuk menghitung distribusi jam dalam satu hari dimana ada pesan yang diterima dari setiap email yang masuk. Gunakan file mbox-short.txt untuk sebagai datanya. Berikut ini adalah contoh output dari programnya.

```
fhandle = open("mbox-short.txt")
counts = dict()
for line in fhandle:
    if line.startswith('From '):
        words = line.split()
        if len(words) > 5:
            time = words[5]
            hour = time.split(':')[0]
            counts[hour] = counts.get(hour, 0) + 1

for hour in sorted(counts):
    print(hour, counts[hour])
```

Output:

```
00 22
01 9
02 8
02 8
02 8
03 18
04 15
05 8
06 23
07 17
08 46
09 120
10 116
11 107
12 81
13 75
14 105
15 94
16 112
17 57
18 38
19 32
20 30
21 32
22 24
23 13
```

Penjelasan:

Program ini membaca file bernama "mbox-short.txt" dan menghitung distribusi jam dari setiap pesan email yang masuk. Pertama, file dibuka dan setiap baris dibaca satu per satu. Jika baris tersebut diawali dengan "From ", maka baris tersebut dipecah menjadi beberapa kata. Jika jumlah kata lebih dari 5, maka kata ke-6 (yang berisi waktu) diambil, lalu dipisahkan berdasarkan tanda titik dua (:) untuk

mendapatkan jamnya saja. Jam tersebut kemudian dihitung jumlah kemunculannya dan disimpan dalam sebuah dictionary. Setelah semua baris diproses, program akan mencetak jam beserta jumlah pesan yang diterima pada jam tersebut secara berurutan.

Link GitHub:

[https://github.com/TimotiusEkanaT/Alpro/tree/main/Laporan\\_13](https://github.com/TimotiusEkanaT/Alpro/tree/main/Laporan_13)