



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	<71230972>
Nama Lengkap	<Oktavian Christ Putranto >
Minggu ke / Materi	11 / Tipe Data Dictionary

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.

### Dictionary

Dictionary terdiri dari Key:Value, Kunci/Key harus bersifat unik dan tidak boleh ada dua kunci yang sama dalam satu dictionary. Bila dalam list kita mengenal index memakai angka sedangkan pada dictionary indeksinya dapat berupa key yang menampung tipe data apapun.

Key dan value saling terikat dan berhubungan satusama lainnya dalam artian jika kita ingin mengambil value dari dictionary maka kita dapat memakai key dari value tersebut.

Dictionary sendiri pasti ditandai dengan kurung kurung awal {} yang didalamnya akan memuat Key:value yang bebas type datanya namun tiap key nya harus beda.

Contoh Dictionary

```
eng={"satu":"One", "dua":"Two", "tiga":"Three"}
print(eng)
```

Output :

```
{'satu': 'One', 'dua': 'Two', 'tiga': 'Three'}
```

Untuk mengakses Two maka kita dapat memakai key "dua", contoh :

```
eng={"satu":"One", "dua":"Two", "tiga":"Three"}
print(eng["dua"])
```

Output :

Two

Cara menambahkan key dan value dalam Dictionary :

```
eng={"satu":"One", "dua":"Two", "tiga":"Three"}

eng["empat"]="Four"
print(eng)
```

Output :

```
{'satu': 'One', 'dua': 'Two', 'tiga': 'Three', 'empat': 'Four'}
```

Jika kita memasukan key yang sama maka hanya akan mengganti valuenya saja, contoh:

```
eng={"satu":"One", "dua":"Two", "tiga":"Three"}
eng["empat"]="Four"
eng["satu"]="4"
print(eng)
```

Output :

```
{'satu': '4', 'dua': 'Two', 'tiga': 'Three', 'empat': 'Four'}
```

Keuntungan dari memakai dictionary dibandingkan list ialah kecepatan pencariannya karena pada list menggunakan algoritma pencarian linear. saat list bertambah, waktu pencarian yang dibutuhkan menjadi lebih lama sesuai dengan panjang list. sedangkan pada dictionary, Python menggunakan algoritma yang disebut Hash Table, dan tidak memperdulikan banyak item yang ada didalam dictionary.

### Counter Dictionary

```
word="brontosaurus"
d=dict()
for c in word:
    if c not in d:
        d[c]=1
    else:
        d[c]=d[c]+1
print (d)
```

Output :

```
{'b': 1, 'r': 2, 'o': 2, 'n': 1, 't': 1, 's': 2, 'a': 1, 'u': 2}
```

Model komputasi diatas disebut dengan histogram, yang mana merupakan istilah statistika dari set perhitungan (atau frekuensi). Perulangan for akan melewati string. Setiap kali terjadi looping, jika karakter c tidak ada dalam dictionary maka akan dibuat item baru dengan kunci c dan nilai awal 1 (asumsinya telah muncul sekali). Jika c sudah ada dalam dictionary secara otomatis akan melakukan penambahan d(c).

Dari model tersebut dikembangkan cara baru yang lebih mudah dalam menghitung nya yaitu dengan cara dibawah :

```
word = 'brontosaurus'

d = dict()
for c in word:
    d[c] = d.get(c,0) + 1
print(d)
```

Dimana akan menghasilkan Output yang sama yaitu :

```
{'b': 1, 'r': 2, 'o': 2, 'n': 1, 't': 1, 's': 2, 'a': 1, 'u': 2}
```

Keduanya memiliki fungsi count yang sama namun pada code ini jauh lebih ringkas karena memakai get dibandingkan if else.

Menghitung kata dalam sebuah file :

```
fname = input('Enter the file name: ')
try:
    fhand = open(fname)
except:
    print('File cannot be opened:', fname)
    exit()

counts = dict()
for line in fhand:
    words = line.split()
    for word in words:
        if word not in counts:
            counts[word] = 1
        else:
            counts[word] += 1
print(counts)
```

output :

```
{'Kita': 1, 'akan': 1, 'mencoba': 1, 'memnulis': 1, 'program': 1, 'Python': 1, 'untuk': 3, 'membaca': 2, 'baris-  
baris': 1, 'file': 1, 'memecah': 1, 'setiap': 4, 'baris': 4, 'menjadi': 1, 'daftar': 1, 'kata': 1, 'dan': 4,  
'kemudian': 1, 'melakukan': 1, 'perulangan': 6, 'melalui': 2, 'kata': 3, 'dalam': 2, 'menghitung': 1,  
'menggunakan': 2, 'dictionary.': 1, 'Asumsi': 1, 'kita': 1, 'dua': 1, 'for': 1, 'perulangan.': 1, 'Perulangan':  
1, 'bagian': 3, 'luar': 2, 'file': 1, 'pada': 2, 'mekalukan': 1, 'iterasi': 1, 'tertentu.': 1, 'Ini': 1, 'adalah': 3,  
'contoh': 1, 'dari': 1, 'pola': 1, 'yang': 1, 'disebut': 1, 'nested': 1, 'loop': 1, 'karena': 1, 'salah': 1, 'satu': 1,  
'lainnya': 1, 'dalam.': 1}
```

Dari code tersebut kita dapat mengetahui jumlah kata yang berulang kali keluar dalam file word.txt

Contoh lain atau bentuk lain dari code diatas (kita memakai get untuk menghitung perulangannya):

```
fname = input('Enter the file name: ')  
try:  
    fhand = open(fname)  
except:  
    print('File cannot be opened:', fname)  
    exit()  
  
counts = dict()  
for line in fhand:  
    words = line.split()  
    for word in words:  
        counts[word] = counts.get(word,0) + 1  
print(counts)
```

Output :

Enter the file name: word.txt

```
{'Kita': 1, 'akan': 1, 'mencoba': 1, 'memnulis': 1, 'program': 1, 'Python': 1, 'untuk': 3, 'membaca': 2, 'baris-  
baris': 1, 'file': 1, 'memecah': 1, 'setiap': 4, 'baris': 4, 'menjadi': 1, 'daftar': 1, 'kata': 1, 'dan': 4,  
'kemudian': 1, 'melakukan': 1, 'perulangan': 6, 'melalui': 2, 'kata': 3, 'dalam': 2, 'menghitung': 1,  
'menggunakan': 2, 'dictionary.': 1, 'Asumsi': 1, 'kita': 1, 'dua': 1, 'for': 1, 'perulangan.': 1, 'Perulangan':  
1, 'bagian': 3, 'luar': 2, 'file': 1, 'pada': 2, 'mekalukan': 1, 'iterasi': 1, 'tertentu.': 1, 'Ini': 1, 'adalah': 3,  
'contoh': 1, 'dari': 1, 'pola': 1, 'yang': 1, 'disebut': 1, 'nested': 1, 'loop': 1, 'karena': 1, 'salah': 1, 'satu': 1,  
'lainnya': 1, 'dalam.': 1}
```



## 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

Bualah sebuah program untuk mendapatkan nilai key, value, dan item dari sebuah dictionary

Input :

```
dictionary = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
print("key      value      item")
for key, value in dictionary.items():
    print(f"{key}      {value}      {key}")
```

Output :

key	value	item
1	10	1
2	20	2
3	30	3
4	40	4
5	50	5
6	60	6

Penjelasan :

Pada code ini kita hanya harus mem print key value dan key nya dalam urutan dan tampilan yang disesuaikan.

### SOAL 2

Bualah sebuah program untuk memetakan dua list mejadi satu dictionary

```
def list_to_dict(lista,listb):
    if len(lista) == len(listb):
        Warna =zip(lista, listb)
        print(dict(Warna))
    else:
        return print("kedua list tidak dalam jumlah yang sama")

list_to_dict(['red', 'green', 'blue'], ['#FF0000', '#008000', '#0000FF'])
```

```
list_to_dict(['red', 'green', 'blue', 'Merah'], ['#FF0000', '#008000', '#0000FF'])
```

Output :

```
{'red': '#FF0000', 'green': '#008000', 'blue': '#0000FF'}
```

kedua list tidak dalam jumlah yang sama

Penjelasan :

pada test yang pertama Dimana jumlah item di list 1 dan list 2 berjumlah sama maka code akan dijalankan dan membuat zip atau penyatuan dua list yang kemudian diubah typenya menjadi dictionary dan ditampilkan, sedangkan saat len/Panjang kedua list tidak sama maka akan mengeluarkan output “kedua list tidak dalam jumlah yang sama”

### SOAL 3

Dengan menggunakan file mbox-short.txt, buatlah program yang dapat membaca log email dan sajikan dalam histogram menggunakan dictionary. Kemudian hitung berapa banyak pesan yang masuk dari email dan sajikan dalam bentuk dictionary. Silakan cek bagian dibawah ini untuk contoh output dari programnya

Input :

```
Fname = input('Enter the file name: ')
try:
    with open(Fname, 'r') as file:
        count={}
        for line in file:
            if line.startswith('From '):
                email = line.split()[1]
                count[email] = count.get(email, 0) + 1

        print (count)
except:
    print('File tersebut tidak dapat ditemukan')
```

Output :

Enter the file name: mbox-short.txt

```
{'stephen.marquard@uct.ac.za': 2, 'louis@media.berkeley.edu': 3, 'zqian@umich.edu': 4,
'rljlowe@iupui.edu': 2, 'cwen@iupui.edu': 5, 'gsilver@umich.edu': 3, 'wagnermr@iupui.edu': 1,
'antranig@caret.cam.ac.uk': 1, 'gopal.ramasammycook@gmail.com': 1, 'david.horwitz@uct.ac.za': 4,
'ray@media.berkeley.edu': 1}
```

Penjelasan :



Pertama tama code akan meminta input dari user untuk memasukan nama file yang ingin di cek, kemudian dimasukan dalam try and except Dimana ketika nama file salah akan mengeluarkan output “file tersebut tidak dapat ditemukan” dan saat nama file ditemukan maka kemudian file akan dibaca menggunakan code :

```
with open(Fname, 'r') as file:
```

kemudian menambah kan dictionary kosong dengan nama count

```
count={}

for line in file:
    if line.startswith('From '):
        email = line.split()[1]
        count[email] = count.get(email, 0) + 1
```

lalu pada code tersebut maka akan membaca filenya perbaris dan ketika ditemukan baris yang diawali dengan “From “ akan di cek satu persatu yang kemudian di split berdasarkan spasinya lalu diambil index[1] nya untuk dimasukan dalam variable email.

Setelah itu di masukan menjadi sebuah dictionary dalam dictionary count yang sebelumnya telah diubah dengan email menjadu keynya dan perulangan key menjadi valuenya(setiap perulangan nama email yang sama)

Memakai prinsip code

```
counts[word] = counts.get(word,0) + 1
```

yang dimodifikasi menyesuaikan menjadi :

```
count[email] = count.get(email, 0) + 1
```

setelah itu ditampilkan .

#### SOAL 4

Dengan menggunakan file mbox-short.txt, buat program untuk mencatat data nama domain pengirim pesan. Hitunglah jumlah pesan yang dikirim masing-masing domain. sajikan dalam bentuk dictionary

Input :

```
Fname = input('Enter the file name: ')
try:
    with open(Fname, 'r') as file:
        count={}
        for line in file:
            line = line.strip()
            if line.startswith('From: '):
                email = line.split("@")[1]
                count[email] = count.get(email, 0) + 1
```

```
        print (count)
except:
    print('File tersebut tidak dapat ditemukan')
```

Output :

```
{'uct.ac.za': 6, 'media.berkeley.edu': 4, 'umich.edu': 7, 'iupui.edu': 8, 'caret.cam.ac.uk': 1, 'gmail.com': 1}
```

Penjelasan :

Struktur code ini hamper sama persis dengan code untuk soal nomor 3 dengan perbedaan di splitnya dan strip untuk menghilangkan lebihannya

Letak perbedaan :

```
line = line.strip()
    if line.startswith('From: '):
        email = line.split("@")[1]
```

Dimana line di strip terlebih dahulu untuk menghilangkan kelebihan spasi dll, kemudian penempatan split Dimana yang diambil ialah setelah @ untuk domain pengirim emailnya selebihnya sama dengan soal nomor 3.