

# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2021/2022



<b>NIM</b>	<b>71210695</b>
<b>Nama Lengkap</b>	<b>Cahaya Sampebua</b>
<b>Minggu ke / Materi</b>	<b>10/Tipe Data Dictionary</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  
2022

## 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

Dictionary merupakan kumpulan data yang tidak berurutan di python. Dictionary dalam python memiliki anggota yang tersusun dari pasangan key:values. Dalam Python, dictionary dapat dibuat di dalam tanda kurung { } yang tiap pasangan item di dalamnya dipisahkan oleh tanda koma (,). Dict dapat berisi berbagai value baik integer, string maupun list.

Untuk membuat dictionary kosong, kita dapat menggunakan fungsi dict():

contohnya:

```
contoh = dict()
```

```
print(contoh)
```

Jika dijalankan, maka akan menghasilkan

```
{}
```

contoh pembuatan dictionary dengan isi;

```
contoh_dict = {'ayam': 'pertama', 'ayam': 'kedua', 'Ayam': 'ketiga'}
```

Tiap key (kunci)

'Ayam1' dan 'ayam1' maka keduanya merupakan kunci/item item yang berbeda.

Dari contoh diatas, jika dilakukan print,

```
print(contoh_dict)
```

maka akan menghasilkan output:

```
{'ayam': 'kedua', 'Ayam': 'ketiga'}
```

dimana key 'ayam' yang pertama sama dengan yang kedua, maka tidak bersifat valid.

Untuk menambahkan item ke dalam dictionary yang sudah ada, dapat digunakan tanda kurung kotak.

Contohnya:

```
contoh_dict = {'ayam': 'pertama', 'ayam': 'kedua', 'Ayam': 'ketiga'}
```

```
contoh_dict['AYAM'] = 'keempat'
```

```
print(contoh_dict)
```

Akan menghasilkan output:

```
{'ayam': 'kedua', 'Ayam': 'ketiga', 'AYAM': 'keempat'}
```

Melakukan Print value dari dictionary;

Dalam melakukan print / mengakses key tertentu dalam dictionary, kita dapat mengaksesnya dengan key dan bukan menggunakan indeks seperti yang ada pada list. Berikut merupakan contohnya;

```
print(contoh_dict['AYAM'])
```

Maka akan menghasilkan;

```
ketiga
```

Fungsi dan Pengecekan pada Dictionary;

Seperti pada list pada python, fungsi **len** juga dapat digunakan pada dict, untuk menghitung banyaknya item dalam dictionary; contohnya;

```
panjang = len(contoh_dict)
print(panjang)
```

Akan menghasilkan output;

```
3
```

Selain itu, kita juga dapat melakukan cek ke dalam dictionary; Misalnya untuk memeriksa ada atau tidaknya suatu key di dalam dictionary, berikut contohnya;

```
>>> contoh_dict = {'ayam': 'pertama', 'ayam': 'kedua', 'Ayam': 'ketiga'}
>>> contoh_dict['AYAM'] = 'keempat'
>>> 'AYAM' in contoh_dict
True
>>> 'ketiga' in contoh_dict
False
>>> |
```

Pada contoh pertama 'AYAM' merupakan key baru yang ditambahkan, sehingga ketika program melakukan pengecekan, memang ada key 'AYAM' dalam dictionary, Sedangkan 'ketiga' merupakan nilai/value dan bukan merupakan kunci, sehingga hasilnya false.

Mendapatkan semua item key:value dari dictionary;

Untuk mendapatkan return dari semua item yang ada di dalam dictionary, dapat digunakan fungsi items(). Berikut merupakan contohnya;

```
x1 = contoh_dict.items()
print(x1)
```

Jika dijalankan, maka akan menghasilkan output sebagai berikut:

```
dict_items([('ayam', 'kedua'), ('Ayam', 'ketiga'), ('AYAM', 'keempat')])
```

Mendapatkan nilai / value dari dictionary;

Untuk mendapatkan return dari value yang ada di dalam dictionary, dapat digunakan fungsi values(). Berikut merupakan contohnya;

```
x = contoh_dict.values()
print(x)
```

Jika dijalankan, maka akan menghasilkan output sebagai berikut

```
dict_values(['kedua', 'ketiga', 'keempat'])
```

Menghapus items dari dictionary;

Untuk menghapuskan item dari dictionary, yang perlu digunakan adalah built-in function yaitu pop() dengan parameter key yang akan dihapus.

Contohnya:

```
contoh_dict.pop("Ayam")
print(contoh_dict)
```

Maka output yang dihasilkan adalah;

```
{'ayam': 'kedua', 'AYAM': 'keempat'}
```

# Dictionary built-in methods

1. `dict.clear()` : Removes all elements of dictionary dict.
2. `dict.copy()` : Returns a shallow copy of dictionary dict.
3. `dict.fromkeys()` : Create a new dictionary with keys from seq and values set to value.
4. `dict.get(key, default = None)` : For key, returns value or default if the key is not in the dictionary.
5. `dict.has_key(key)` : Returns true if a key exists in the dictionary and false if otherwise.
6. `dict.items()` : Returns a list of dict's (key, value) tuple pairs.
7. `dict.keys()` : Returns a list of dictionary keys.
8. `dict.setdefault(key, default = None)` : Similar to `get()`, but it will set `dict[key]=default` if the key does not already exist in the dictionary.
9. `dict.update(dict2)` : Merges dictionary `dict2's` key-values pairs with `dict`.
10. `dict.values()` : Returns list of dictionary values.

*Gambar 1 built-in pada dictionary (sumber educative.io)*

*Dictionary sebagai set penghitung (counters)*

[ sumber : Modul-10 ALPRO ]

Contoh kasus: Diberikan sebuah string yang akan dihitung banyaknya huruf yang muncul, beberapa cara yang dapat digunakan antaranya adalah;

1. Dengan membuat 26 variabel, kemudian memasukkannya dalam string untuk masing masing karakter.
2. Membuat list berisi 26 elemen lalu melakukan konversi ke angka, lalu menggunakan angka sebagai indeks dalam list dan menambah perhitungan yang sesuai.
3. Membuat dictionary dengan karakter sebagai kunci dan perhitungan sebagai nilai yang sesuai pada satu karakter dapat ditambahkan ke dalam dictionary.

Dalam pengimplementasian-nya, opsi ketiga yaitu menggunakan dictionary dapat membuatnya lebih praktis, dimana kita tidak perlu mengetahui huruf mana saja yang akan muncul dalam string, tetapi hanya perlu memberikan tempat untuk huruf yang akan muncul. Contoh penerapannya adalah sebagai berikut;

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

Dimana d awalnya merupakan dictionary kosong, lalu untuk setiap c dalam variabel word akan dicacah.

Jika c yang ditemukan belum berada di dalam dict d, maka akan dibuat dengan key berisi 1 (yaitu perhitungan berapa kali huruf tsb muncul)

Lalu jika huruf yang ada sudah terdapat dalam dict, maka value pada key huruf tsb akan ditambahkan dengan 1 (+1 perhitungan huruf tersebut muncul)

Lalu outputnya akan mendapatkan;

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

Model program diatas disebut dengan *histogram* yaitu istilah statistika dari set perhitungan atau frekuensi.

Selain dengan source code diatas, kita juga dapat membuat code dengan lebih ringkas yaitu dengan menggunakan fungsi get pada dictionary. Fungsi get digunakan untuk memperoleh nilai dari key yang ada dalam dictionary. Jika kunci ada dalam dictionary, maka program akan mengembalikan nilai yang sesuai, sedangkan jika tidak, maka program akan mengembalikan nilai default.

Berikut source code menggunakan fungsi get;

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

Hasil outputnya pun akan menghasilkan nilai yang sama, yaitu;

```
7Downloads/Dict/coba.py  
{'b': 1, 'r': 2, 'o': 2, 'n': 1, 't': 1, 's': 2, 'a': 1, 'u': 2}
```

### *Looping dan Dictionary*

[ sumber : Modul-10 ALPRO ]

Dalam statement for di dictionary, program akan bekerja dengan cara menelusuri kunci yang ada di dalam dict, Looping akan melakukan output setiap kunci sesuai dengan valuenya.

contoh program;

```
counts = { 'mikro' : '1' , 'run' : 2, 'butter': 100}  
for key in counts:  
    print(key, counts[key])
```

Hasil outputnya adalah:

```
mikro 1  
run 2  
butter 100
```

### *Advanced Text Parsing*

[ sumber : Modul-10 ALPRO ]

```
But, soft! what light through yonder window breaks?  
It is the east, and Juliet is the sun.  
Arise, fair sun, and kill the envious moon,  
Who is already sick and pale with grief,
```

Test case yang ada untuk memisahkan text dengan tanda baca yang ada.

Untuk melakukan hal ini, akan digunakan metode string seperti lower,punctuation dan translate.Sebagai berikut;

```
line.translate(str.maketrans(fromstr, tostr, deletestr))
```

tostr tidak akan ditentukan, tetapi kita akan menggunakan parameter deletestr untuk menghapus semua tanda baca. Python akan secara otomatis memberitahu bagian mana yang dianggap sebagai tanda baca, yaitu dengan menggunakan import string dan menggunakan built-in .punctuation pada string.

Penggunaannya dalam string adalah sebagai berikut;

```
import string

fname = input('Enter the file name: ')
try:
    fhand = open(fname)

except:
    print('File cannot be opened:', fname)
    exit()

counts = dict()
for line in fhand:
    line = line.rstrip()
    line = line.translate(line.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
print(counts)
```

Setelah dilakukan run, maka akan menghasilkan output sebagai berikut;

```
Enter the file name: CONTOH ADVANCED.txt
{'but': 1, 'soft': 1, 'what': 1, 'light': 1, 'through': 1, 'yonder': 1, 'window': 1, 'breaks': 1, 'it': 1, 'is': 3, 'the': 3, 'east': 1, 'and': 3, 'juliet': 1, 'sun': 2, 'arise': 1, 'fair': 1, 'kill': 1, 'envious': 1, 'moon': 1, 'who': 1, 'already': 1, 'sick': 1, 'pale': 1, 'with': 1, 'grief': 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.

Link GitHub → <https://github.com/cahaya-sampebua/Latihan-Mandiri-PrakALPro-10.git>

### SOAL 1

#### #Soal 1

```
dictionary = {"1": 10, "2": 20, "3": 30, "4": 40, "5": 50, "6": 60}

print("key",end="\t")
print("value",end="\t")
print("items")

list = sorted(list(dictionary.items()))

count = 0
for key, value in list:
    count += 1
    print(count,end="\t")
    print(value,end="\t")
    print(key)
```

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

Masukkan items dalam dictionary ke dalam satu list yang sudah diurutkan. Setelah itu untuk kunci dan value, print sehingga menjadi bentuk seperti output.

## SOAL 2

```
#Soal 2

list_a = ["red","green","blue"]
list_b = ["#FF0000","008000","0000FF"]

dictionary = dict()

for i in range(0,len(list_a)):
    dictionary[list_a[i]] = list_b[i]

print(dictionary)

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

Buat perulangan dengan range berdasarkan jumlah indeks dalam list\_a yang berisi kunci. Karena dalam dictionary yang penting adalah kunci, maka kita buat kuncinya dulu. Untuk setiap i dalam range jumlah indeks dalam list\_a, masukkan key dan value dengan cara biasa **dict[key] = value**.

## SOAL 3

```
#Soal 3

import re

nama_file = input("Masukkan nama file : ")
file = open(nama_file)

baca_file = file.read()

email_dictionary = {}

email_list = re.findall("From: ([\w]+\.[\w]+@.+)", baca_file)

for i in email_list:
    if i not in email_dictionary:
        email_dictionary[i] = 1
    else:
        email_dictionary[i] += 1

print(email_dictionary)
print(sum(email_dictionary.values()))

file.close()
```

## Output

```
Masukkan nama file : mbox-short.txt
{'stephen.marquard@uct.ac.za': 2, 'louis@media.berkeley.edu': 3, 'zqian@umich.edu': 4, 'rjlowe@iupui.edu': 2, 'cwen@iupui.edu': 5, 'gsilver@umich.edu': 3, 'wagnerm@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}
27
```

Buat variabel untuk membuat list kata-kata dengan menggunakan regular expressions. Untuk setiap indeks dalam list tersebut, jika i belum menjadi key dalam dictionary, tambahkan i sebagai key dan 1 sebagai value pertama, jika sudah ada, value dari key tersebut ditambahkan dengan 1.

Regex di atas mencari yang diawali From: , namun hanya mengcapture emailnya saja.

## SOAL 4

```
#Soal 4

import re

file = input("Masukkan nama file : ")
file = open(file)
baca_file = file.read()

re_file = re.findall("From: [\w]+\.[\w]+@(.+)", baca_file)

dictionary_domain = dict()

list_baru = []
count = 0
for i in re_file:
    if i not in list_baru:
        list_baru.append(i)
        for j in re_file:
            if i == j:
                count += 1

        dictionary_domain[i] = count
        count = 0

print(dictionary_domain)

file.close()

Masukkan nama file : mbox-short.txt
{'uct.ac.za': 6, 'media.berkeley.edu': 4, 'umich.edu': 7, 'iupui.edu': 8, 'caret.cam.ac.uk': 1, 'gmail.com': 1}
```

Untuk setiap indeks dalam list yang dibuat dengan regex, jika indeks belum menjadi key dalam dictionary, buat indeks menjadi key dan 1 menjadi value pertama. Jika sudah ada, value dari key akan ditambahkan dengan 1.

Regex di atas mencari yang berawalan From: , namun hanya domainnya saja, yaitu setelah "@".

## Daftar Pustaka

educative. "How does the dictionary work in Python?", <https://www.educative.io/edpresso/how-does-the-dictionary-work-in-python>

Modul-10 ALPRO

W3Schools. "Python - Access Dictionary Items" ,  
[https://www.w3schools.com/python/python\\_dictionaries\\_access.asp](https://www.w3schools.com/python/python_dictionaries_access.asp)

W3Schools. "Python - Remove Dictionary Items" ,  
[https://www.w3schools.com/python/python\\_dictionaries\\_remove.asp](https://www.w3schools.com/python/python_dictionaries_remove.asp)