

# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024



NIM	71200645
Nama Lengkap	Edith Felicia Putri
Minggu ke / Materi	10 / 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  
2023

## 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 1

#### PENGANTAR DICTIONARY

*Dictionary* mempunyai kemiripan dengan *list*, tetapi lebih bersifat umum. Dalam *list*, indeks harus berupa *integer* sedangkan dalam *dictionary* indeks bisa dapat berupa apapun.

*Dictionary* memiliki anggota yang terdiri dari pasangan **kunci:nilai**. Kunci harus bersifat unik, tidak boleh ada dua kunci yang sama dalam satu *dictionary*. *Dictionary* dapat diartikan pula sebagai pemetaan antara sekumpulan indeks (yang disebut kunci) dan sekumpulan nilai.

Setiap kunci memetakan suatu nilai. Asosiasi kunci dan nilai tersebut disebut dengan pasangan nilai kunci (*key-value pair*) atau ada yang menyebutnya sebagai *item*. Nilai *dictionary* dapat berupa tipe apapun, namun *key* harus berupa tipe data yang tidak berubah seperti *string*, angka, atau *tuple*.

Fungsi *dict* digunakan untuk membuat dictionary baru yang kosong. Karena *dict* merupakan built-in *function* dari python, maka penggunaanya perlu dihindari sebagai nama variabel.

#### Membuat dictionary.

```
my_dict = {"Tuhan": "Yesus", "Dosa": "Maut"}
```

Tanda kurung kurawal { } digunakan untuk merepresentasikan dictionary kosong. Untuk menambahkan *item* dalam *dictionary*, dapat menggunakan kurung kotak [ ].

#### Mengakses item.

```
my_God = dict["Tuhan"]  
print(my_God) # Output: Yesus
```

#### Menambah dan meng-update item.

```
dict["Imam"] = "Eli"  
dict["Nabi"] = "Elia"
```

#### OUTPUT:

```
PS D:\71200645_PrakAlpro_10> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_10/71230994/71230994/week11/week11.py  
{'Tuhan': 'Yesus', 'Dosa': 'Maut', 'Imam': 'Eli', 'Nabi': 'Elia'}  
PS D:\71200645_PrakAlpro_10> □
```

**Menghapus item (del dan pop).**

```
del dict["Imam"]
dict.pop("Dosa")
```

**OUTPUT:**

```
PS D:\71200645_PrakAlpro_10> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_10/71230994/71230994/week11/week11.py
{'Tuhan': 'Yesus', 'Nabi': 'Elia'}
PS D:\71200645_PrakAlpro_10> □
```

**Memanggil dictionary.**

```
eng2sp = {'one': 'uno', 'two': 'dos', 'three': 'tres'}
print('one' in eng2sp) # True
print(eng2sp['one']) # uno
# Pemanggilan dict hanya bisa menggunakan key.
```

**Metode yang biasa digunakan:**

- keys(): mengembalikan *keys* dari *dictionary*.
- values(): mengembalikan *values* dari *dictionary*.
- items(): mengembalikan pasangan-pasangan *key-value* dari *dictionary*.

**Metode-metode lainnya:**

**Source:** <https://belajarpython.com/tutorial/dictionary-python/>

- clear(): menghapus semua elemen *dictionary*.
- copy(): mengembalikan salinan *dictionary*.
- fromkeys(): buat *dictionary* baru dengan kunci dari seq dan nilai yang disetel ke nilai.
- has\_key(key): mengembalikan *true* jika *key* dalam *dictionary*, *false* sebaliknya.
- update(dict2): menambahkan pasangan kunci kata kunci dict2 ke dict.

**Fungsi *build-in* pada *dictionary* python.**

- cmp(dict1, dict2): membandingkan unsur keduanya.
- len(dict): memberikan panjang total *dictionary*. Ini sama dengan jumlah *item* dalam *dictionary*.
- str(dict): menghasilkan representasi *string* yang dapat dicetak dari *dictionary*.
- type(variable): mengembalikan tipe variabel yang lulus. Jika variabel yang dilewatkan adalah *dictionary*, maka akan mengembalikan tipe *dictionary*.

**Menelusuri dictionary.**

```
for key, value in dict.items():
    print(f"{key}: {value}")
```

## OUTPUT:

```
PS D:\71200645_PrakAlpro_10> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_10/71230994/71230994/week11/week11.py
Tuhan: Yesus
Nabi: Elia
PS D:\71200645_PrakAlpro_10> █
```

Menggabungkan kedua list dengan dictionary.

```
keys = ['Izuku', 'Uraraka', 'Iida']
values = ['Deku', 'Uravity', 'Ingenium']
hero = dict(zip(keys, values))
print(hero)
```

Menggabungkan kedua list dengan dictionary melalui for loop.

```
keys = ['Izuku', 'Uraraka', 'Iida']
values = ['Deku', 'Uravity', 'Ingenium']

my_hero = {}
for i in range(len(keys)):
    my_hero[keys[i]] = values[i]

print(my_hero)
```

## OUTPUT:

```
PS D:\71200645_PrakAlpro_10> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_10/71230994/71230994/week11/week11.py
{'Izuku': 'Deku', 'Uraraka': 'Uravity', 'Iida': 'Ingenium'}
PS D:\71200645_PrakAlpro_10> █
```

Mengambil value melalui key dictionary.

```
print(hero.get('Izuku'))
```

Membuat fungsi-fungsi untuk menambah kontak, menghapus kontak, dan mencari kontak, menggunakan *dictionary*.

```
def tambah_kontak(buku_telepon, nama, nomor):
    buku_telepon[nama] = nomor
    print(f"Kontak {nama} ditambah/perbaharui.")

def hapus_kontak(buku_telepon, nama):
    if nama in buku_telepon:
        del buku_telepon[nama]
        print(f"Kontak {nama} dihapus.")
```

```

else:
    print(f"Kontak {nama} tidak ditemukan.")

def cari_kontak(buku_telepon, nama):
    return buku_telepon.get(nama, "Kontak tidak ditemukan.")

buku_telepon = {}
tambah_kontak(buku_telepon, 'Erwin', '0570-0000-911')
tambah_kontak(buku_telepon, 'Levi', '(03)5285-8181')
print(cari_kontak(buku_telepon, 'Eren'))
hapus_kontak(buku_telepon, 'Erwin')

```

## OUTPUT:

```

PS D:\71200645_PrakAlpro_10> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_10/71230994/71230994/week11/week11.py
Kontak Erwin ditambah/perbaharui.
Kontak Levi ditambah/perbaharui.
Kontak tidak ditemukan.
Kontak Erwin dihapus.
PS D:\71200645_PrakAlpro_10> 

```

## Dictionary sebagai set penghitung (counters).

Penggunaan komputasi model *dictionary* lebih praktis, dimana kita tidak perlu mengetahui huruf mana yang akan muncul dalam *string*. Hanya perlu memberikan ruang untuk huruf yang akan muncul.

```

word = "Hello Friends"
d = dict()
for i in word:
    if i not in d:
        d[i] = 1 # buat dict baru
    else:
        d[i] += 1
print(d)

```

Model komputasi di atas disebut dengan histogram, yang mana merupakan istilah statistika dari set perhitungan (atau frekuensi). *Dictionary* memiliki metode yang disebut *get* yang mengambil kunci dan nilai default. Jika kunci muncul di *dictionary*, akan mengembalikan nilai yang sesuai; jika tidak maka akan mengembalikan nilai *default*.

```

counts = {"chuck":1, "annie":42, "jan":100}
print(counts.get("doni")) #default isinya 0.
# bisa pula nilai default diletakkan di dalam. ("doni", 0)

```

Penggunaan *get* dapat dilakukan untuk menulis *loop histogram* secara lebih ringkas. Metode *get* secara otomatis menangani *case* dimana kunci tidak ada dalam *dictionary*.

## MATERI 2

### DICTIONARY DAN FILE

Salah satu kegunaan umum *dictionary* adalah untuk menghitung kemunculan kata-kata dalam file dengan beberapa teks tertulis.

```
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)
```

#### Looping dan Dictionary.

Dalam *statement for*, *dictionary* akan bekerja dengan cara menelusuri kunci yang ada di dalamnya. *Looping* ini akan melakukan pencetakan setiap kunci sesuai dengan hubungan nilainya.

```
counts = {"chuck":1, "annie":42, "jan":100, "rudi": 0}
for i in counts:
    if counts[i] > 10:
        print(i, counts[i])
```

```
annie 42
jan 100
```

## MATERI 3

### ADVANCED TEXT PARSING

Python sendiri mempunyai *function split* dimana fungsi tersebut akan mencari spasi dan mengubah kata-kata sebagai token yang dipisahkan oleh spasi. Mis: kata "soft!" dan "soft" merupakan dua kata yang berbeda dan dalam dictionary adakn dibuat terpisah untuk tiap kata tersebut. Hal tersebut juga berlaku pada penggunaan huruf kapital. Misal pada kata "who" dan "Who" dianggap sebagai kata berbeda dan dihitung secara terpisah. Model penyelesaian lain dapat juga dengan menggunakan method string lain seperti *lower*, *punctuation* dan *translate*. Metode *string translate* merupakan metode *string* yang paling *stuble* (hampir tidak kentara).

## 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

**Latihan 10.1** Bualah sebuah program untuk mendapatkan nilai key, value, dan item dari sebuah dictionary.

Contoh:

Dictionary : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

Output:

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

Pengerjaan soal dilakukan dengan membuat *dictionary* yang diinginkan, kemudian menampilkan *header* dengan fungsi *print()* untuk *key*, *value*, dan *item*. Lalu melakukan *for loop* untuk setiap *i* mengambil (*key*, *value*)-nya di dalam *Dictionary.items()* untuk mengembalikan pasangan *key* dan *value* dalam bentuk *tuple* dengan fungsi *enumerate()* dimulai dari satu, kemudian melakukan *print()* untuk menampilkan sesuai *for loop*.

### SOURCE CODE:

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

print("key value item")
for i, (key, value) in enumerate(Dictionary.items(), start=1):
    print(f"{key:}    {value:}    {i:}")
```

### OUTPUT:

```
/Python/Python311/python.exe "d:/71200645_PrakAlpro_10/Latihan 10
.1.py"
key value item
1 10 1
2 20 2
3 30 3
4 40 4
5 50 5
6 60 6
```

### SOAL 2

**Latihan 10.2** Bualah sebuah program untuk memetakan dua list mejadi satu dictionary.

Contoh:

Data List

Lista = ['red', 'green', 'blue']

Listb = ['#FF0000', '#008000', '#0000FF']

Output

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

Pengerjaan soal dilakukan dengan membuat kedua data *list*. Kemudian menggunakan fungsi *zip()* pada *dictionary* untuk kedua data *list* yang digunakan untuk menggabungkan *item*/elemen dari iterables (seperti *list*, *tuple*, atau *string*, sehingga kedua *list* menjadi satu *dictionary*.

#### SOURCE CODE:

```
Lista = ['red', 'green', 'blue']
Listb = ['#FF0000', '#008000', '#0000FF']

my_dict = dict(zip(Lista, Listb))
print(my_dict)
```

#### OUTPUT:

```
PS D:\71200645_PrakAlpro_10> & C:/Users/HP/AppData/Local/Program
s/Python/Python311/python.exe "d:/71200645_PrakAlpro_10/Latihan
10.2.py"
{'red': '#FF0000', 'green': '#008000', 'blue': '#0000FF'}
PS D:\71200645_PrakAlpro_10>
```

#### SOAL 3

**Latihan 10.3** 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.

Masukkan nama file : mbox-short.txt

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

Pengerjaan soal dilakukan dengan memasukkan nama *file* yang digunakan, kemudian membuat *dictionary* kosong untuk menyimpan alamat-alamat email yang ada dalam *file*. Lalu membuka *file* dalam mode *read* dengan fungsi *with open()*. Model komputasi dalam bentuk histogram (set perhitungan/frekuensi) dengan perulangan *for loop* untuk setiap *string* pada *file*. Setiap kali terjadi *looping*, jika baris dimulai dengan *string* 'From' terjadi *split* yang disimpan dalam variabel kata, kemudian mengambil dari variabel kata, index pertama (dimana alamat email ada: string setelah string 'From') dan menyimpannya di variabel email. Dilakukan percabangan *if-else* untuk memeriksa apakah alamat email (key) sudah disimpan dalam *dictionary* kemunculan\_email atau belum, apabila sudah maka *value* atau *count* meningkat satu, apabila belum maka alamat email dimasukkan ke dalam *dictionary* dengan *value* satu.

#### SOURCE CODE:

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

def hitung_kemunculan_email(nama_file):
```



```

kemunculan_email = {}

with open(nama_file, 'r') as file:
    for baris in file:
        if baris.startswith('From '):
            kata = baris.split()
            email = kata[1]
            if email in kemunculan_email:
                kemunculan_email[email] += 1
            else:
                kemunculan_email[email] = 1

    return kemunculan_email

hasil = hitung_kemunculan_email(nama_file)
print(hasil)

```

## OUTPUT:

```

PS D:\71200645_PrakAlpro_10> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "d:/71200645_PrakAlpro_10/Latihan 10.3.py"
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, '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}
PS D:\71200645_PrakAlpro_10>

```

## SOAL 4

**Latihan 10.4** 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.

Silakan cek bagian dibawah ini untuk contoh output dari programnya.

Masukkan nama file: mbox-short.txt

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

Pengerjaan soal dilakukan dengan memasukkan nama *file* yang digunakan, lalu membuat *dictionary* kosong untuk mengisi domain-domain dari alamat email yang ada dalam *file*. Setiap kali terjadi *looping*, jika baris dimulai dengan string 'From' terjadi *split* yang disimpan dalam variabel kata, kemudian mengambil dari variabel kata, index pertama (dimana alamat email ada: string setelah string 'From') dan menyimpannya di variabel email. Kemudian alamat email di-*split* lagi hanya untuk mendapatkan nama-nama domainnya berdasarkan '@' dan mengambil index pertama lagi. Dilakukan percabangan *if-else* untuk memeriksa apakah domain (key) sudah

disimpan dalam *dictionary* kemunculan\_email atau belum, apabila sudah maka *value* atau *count* meningkat satu, apabila belum maka alamat email dimasukkan ke dalam *dictionary* dengan *value* satu.

#### SOURCE CODE:

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

def hitung_kemunculan_email(nama_file):
    kemunculan_email = {}

    with open(nama_file, 'r') as file:
        for baris in file:
            if baris.startswith('From '):
                kata = baris.split()
                email = kata[1]
                domain = email.split('@')[1]
                if domain in kemunculan_email:
                    kemunculan_email[domain] += 1
                else:
                    kemunculan_email[domain] = 1

    return kemunculan_email

print(hitung_kemunculan_email(nama_file))
```

#### OUTPUT:

```
PS D:\71200645_PrakAlpro_10> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "d:/71200645_PrakAlpro_10/Latihan 10.4.py"
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}
PS D:\71200645_PrakAlpro_10> █
```

#### Link Github:

[https://github.com/EdithFelicia/71200645\\_Guided/tree/main/71200645\\_PrakAlpro\\_10](https://github.com/EdithFelicia/71200645_Guided/tree/main/71200645_PrakAlpro_10)