



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	71200609
<b>Nama Lengkap</b>	Airell Aristo Subagia
<b>Minggu ke / Materi</b>	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  
2024

# Dictionary

## Apa Itu List

Dalam pemrograman Python, dictionary merupakan struktur data yang digunakan untuk menyimpan pasangan kunci-nilai yang bersifat tidak berurutan. Ini berarti kita dapat menggunakan kunci untuk mengakses nilai-nilai dalam dictionary. Dictionary ditulis dalam kurung kurawal {}, dengan setiap pasangan kunci-nilai dipisahkan oleh koma.

Contoh :

```
identitas = {"nama": "Airell",  
            "umur": 22,  
            "domisili": "Yogyakarta"}
```

## Mengakses Nilai/Kunci Didalam Dictionary

Ada beberapa cara yang dapat dilakukan untuk mengakses nilai pada dictionary. Berikut contohnya :

1. Menggunakan keynya secara langsung

Kita dapat mengakses nilai dalam dictionary dengan menggunakan kunci sebagai indeks. Misalnya:

```
identitas = {"nama": "Airell",  
            "umur": 22,  
            "domisili": "Yogyakarta"}  
  
print(identitas["nama"])
```

Outputnya :

**Airell**

2. Metode get()

Kita juga dapat menggunakan metode get() untuk mengakses nilai dalam dictionary. Metode ini mengembalikan nilai yang terkait dengan kunci yang diberikan. Misalnya :

```
identitas = {"nama": "Airell",  
            "umur": 22,  
            "domisili": "Yogyakarta"}
```

```
print(identitas.get("umur"))
```

Outputnya :

```
22
```

### 3. Mengakses Kunci Dictionary menggunakan looping

Kita dapat menggunakan looping untuk mengakses semua kunci yang ada dalam dictionary. Misalnya :

```
identitas = {"nama": "Airell",  
            "umur": 22,  
            "domisili": "Yogyakarta"}  
  
for key in identitas :  
    print(key)
```

### 4. Metode mengakses kunci dan value menggunakan keys() dan values()

Metode keys() digunakan untuk mendapatkan semua kunci dalam dictionary, sedangkan metode values() digunakan untuk mendapatkan semua nilai.

```
identitas = {"nama": "Airell",  
            "umur": 22,  
            "domisili": "Yogyakarta"}  
  
print(identitas.keys())  
print(identitas.values())
```

Outputnya :

```
dict_keys(['nama', 'umur', 'domisili'])  
dict_values(['Airell', 22, 'Yogyakarta'])
```

## Contoh-contoh Penggunaan Dictionary

Ada Beberapa kegunaan Dictionary yang sering digunakan. Beberapa contohnya seperti pemetaan data, penyimpanan konfigurasi, analisis data (contoh seperti pada teks).

Contoh :

### 1. Pemetaan Data

Dictionary digunakan untuk memetakan satu set nilai ke nilai lainnya. Ini sangat berguna ketika kita perlu menghubungkan atau mengaitkan dua jenis data yang berbeda.

```
kodePos = {'Bogor': 16110,  
           'Yogyakarta': 55000,  
           'Makassar': 90111}  
  
print(kodePos['Makassar'])
```

Outputnya :

```
90111
```

## 2. Penyimpanan Konfigurasi

Dictionary sering digunakan untuk menyimpan konfigurasi aplikasi atau pengaturan program karena mereka memungkinkan kita untuk dengan mudah mengakses berbagai opsi dan nilai-nilai mereka.

```
konfigurasi = {'ukuran_font': 12, 'tema': 'gelap', 'notifikasi': True}
```

## 3. Analisis Data

Dalam analisis data, dictionary sering digunakan untuk menyimpan hasil agregasi atau hasil penghitungan, seperti jumlah kemunculan setiap kata dalam sebuah teks.

```
teks = "Ini adalah contoh teks yang panjang panjang banget"  
hitung_kata = {}  
for kata in teks.split():  
    if kata in hitung_kata:  
        hitung_kata[kata] += 1  
    else:  
        hitung_kata[kata] = 1  
print(hitung_kata)
```

Outputnya :

```
{'Ini': 1, 'adalah': 1, 'contoh': 1, 'teks': 1, 'yang': 1, 'panjang': 2, 'banget': 1}
```

## Looping Pada Dictionary

Looping pada dictionary di Python menggunakan beberapa cara. Contoh :

### 1. Looping Untuk Mendapatkan Key

Kita dapat menggunakan perulangan for untuk mengakses setiap key dalam dictionary.

```
identitas = {"nama": "Airell",  
            "umur": 22,  
            "domisili": "Yogyakarta"}  
  
for key in identitas :  
    print(key)
```

Outputnya :

```
nama  
umur  
domisili
```

### 2. Looping Untuk Mendapatkan Value

Kita dapat menggunakan perulangan for untuk mengakses setiap value dalam dictionary dengan bantuan dari key.

```
identitas = {"nama": "Airell",  
            "umur": 22,  
            "domisili": "Yogyakarta"}  
  
for key in identitas :  
    print(identitas[key])
```

Outputnya :

```
Airell  
22  
Yogyakarta
```

### 3. Looping menggunakan .items()

Anda dapat menggunakan metode items() untuk melakukan perulangan melalui semua pasangan kunci-nilai dalam dictionary.

```
identitas = {"nama": "Airell",  
             "umur": 22,  
             "domisili": "Yogyakarta"}  
  
for key, value in identitas.items() :  
    print(f"{key} : {value}")
```

Outputnya :

```
nama : Airell  
umur : 22  
domisili : Yogyakarta
```

## BAGIAN 2: LATIHAN MANDIRI (60%)

### Soal 1

#Latihan 10.1

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

```
def keyValue(dictionary) :  
    print("Key      Value      Item")  
    for key in dictionary :  
        print(key, "      ", dictionary[key], "      ", key)
```

```
keyValue(Dict)
```

Outputnya :

Key	Value	Item
1	10	1
2	20	2
3	30	3
4	40	4
5	50	5
6	60	6

Penjelasannya :

1. Pertama-tama, dibuat dulu function bernama **keyValue()** yang dimana parameternya akan berupa dictionary.
2. Setelah itu, didalamnya dilakukan dulu printing sesuai dengan soal .
3. Kemudian, tinggal dilakukan **for loop** untuk mengakses key dan value dari dictionary
4. Terakhir, tinggal dilakukan print key dan value dengan cara menggunakan variable **key** dari looping sebelumnya dan data dictionary dipanggil menggunakan variable **key**.

## SOAL 2

#Latihan 10.2

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

def combList(a,b) :
    dict = {}
    for i in range(len(a)) :
        dict[a[i]] = b[i]
    return dict

print(combList(Lista,Listb))
```

Outputnya :

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

Penjelasannya :

1. Pertama-tama, dibuat function bernama **combList()** dengan parameternya berupa 2 list. Yang akan digunakan untuk digabungkan valuenya nanti.
2. Setelah itu, dibuat **dictionary** kosong bernama **dict**. Digunakan untuk menampung hasil penggabungan kedua list.
3. Kemudian, dibuat **for loop**. Yang digunakan untuk mengakses value dari kedua list **a** dan **b** sesuai dengan panjang dari list **a**.
4. Terakhir, tinggal dilakukan penambahan nilai kedalam **dict**, dengan cara menggunakan list **a** dengan indeks dari looping **i** sebagai key dan **b** dengan indeks dari looping **i** sebagai value. Dan sisa di return saja



### SOAL 3

```
#Latihan 10.3
fileName = input("Masukkan Nama File : ")

artikel = open("c:\\Users\\Airell\\Documents\\Tugas Kampus\\Semester 8\\Praktikum AlPro\\Pertemuan11\\71200609_Pertemuan11\\"+fileName,"r")
sender = {}
for baris in artikel :

    x = baris.lower()
    y = x.split()
    if len(y) > 1 :
        if y[0] == "author:" :
            if y[1] not in sender.keys() :
                sender[y[1]] = 1
            else :
                sender[y[1]] += 1

print(sender)
```

Outputnya :

```
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, 'gsilven@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}
```

Penjelasannya :

1. Pertama-tama, dibuat variable bernama **fileName** untuk menyimpan nama file yang akan dibuka.
2. Setelah itu, dibuka di varibale bernama **artikel** dengan mode **read**.
3. Kemudian, dibuat **dictionary** bernama **sender** untuk meyimpan author dari pengirim surat nantinya.
4. Setelah itu , dilakukan **for loop** yang dimana digunakan untuk mendapatkan baris pada artikel dan juga membuat **lower()** setiap huruf lalu **split()** menggunakan spasi.
5. Kemudian, dilakukan **filtering menggunakan if statement** kembali untuk menghilangkan semua baris kosong dengan check panjang dari listnya.
6. Terakhir, dilakukan checking lagi apakah pada baris tersebut kata pertama mengandung huruf **"author:"**. Jika iya maka akan dilanjutkan ke **if statement berikutnya** untuk check apakah **y[1] = (email)** dari pengirim sudah ada didalam dictionary atau belum, jika sudah tinggal ditambahkan 1 jika belum dibuat key dan value yang baru.

## SOAL 4

```
#Latihan 10.4
fileName = input("Masukkan Nama File : ")

artikel = open("c:\\Users\\Airell\\Documents\\Tugas Kampus\\Semester 8\\Praktikum AlPro\\Pertemuan11\\71200609_Pertemuan11\\"+fileName+".txt")
sender = {}
for baris in artikel :
    x = baris.lower()
    y = x.split()
    if len(y) > 1 :
        if y[0] == "author:" :
            x1 = y[1].split("@")
            # print(x1)
            if x1[1] not in sender.keys() :
                sender[x1[1]] = 1
            else :
                sender[x1[1]] += 1
print(sender)
```

Outputnya :

```
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}
```

Penjelasannya :

1. Pertama-tama, dibuat variable bernama **fileName** untuk menyimpan nama file yang akan dibuka.
2. Setelah itu, dibuka di varibale bernama **artikel** dengan mode **read**.
3. Kemudian, dibuat **dictionary** bernama **sender** untuk meyimpan author dari pengirim surat nantinya.
4. Setelah itu , dilakukan **for loop** yang dimana digunakan untuk mendapatkan baris pada artikel dan juga membuat **lower()** setiap huruf lalu **split()** menggunakan spasi.
5. Kemudian, dilakukan **filtering menggunakan if statement** kembali untuk menghilangkan semua baris kosong dengan check panjang dari listnya.
6. Setelah itu, dilakukan checking lagi apakah pada baris tersebut kata pertama mengandung huruf **"author:"**. Jika iya maka akan dibuat variable baru bernama **x1** untuk menampung nilai dari domain si pengirimnya. Yang dimana isi dari **y[1].split("@")** itu domainnya karena indeks 0 merupakan usernamenya.

7. Terakhir, **if statement berikutnya** untuk check apakah **x1[1] = (domain)** dari pengirim sudah ada didalam dictionary atau belum, jika sudah tinggal ditambahkan 1 jika belum dibuat key dan value yang baru.

Link Github :

[https://github.com/AirellAristo/TugasPrakAlpro/tree/main/71200609\\_Pertemuan10\(Dictionary\)](https://github.com/AirellAristo/TugasPrakAlpro/tree/main/71200609_Pertemuan10(Dictionary))