Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	**Samuel Natanael**
Nama Lengkap	**71231050**
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
2024

BAGIAN 1: MATERI MINGGU INI (40%)

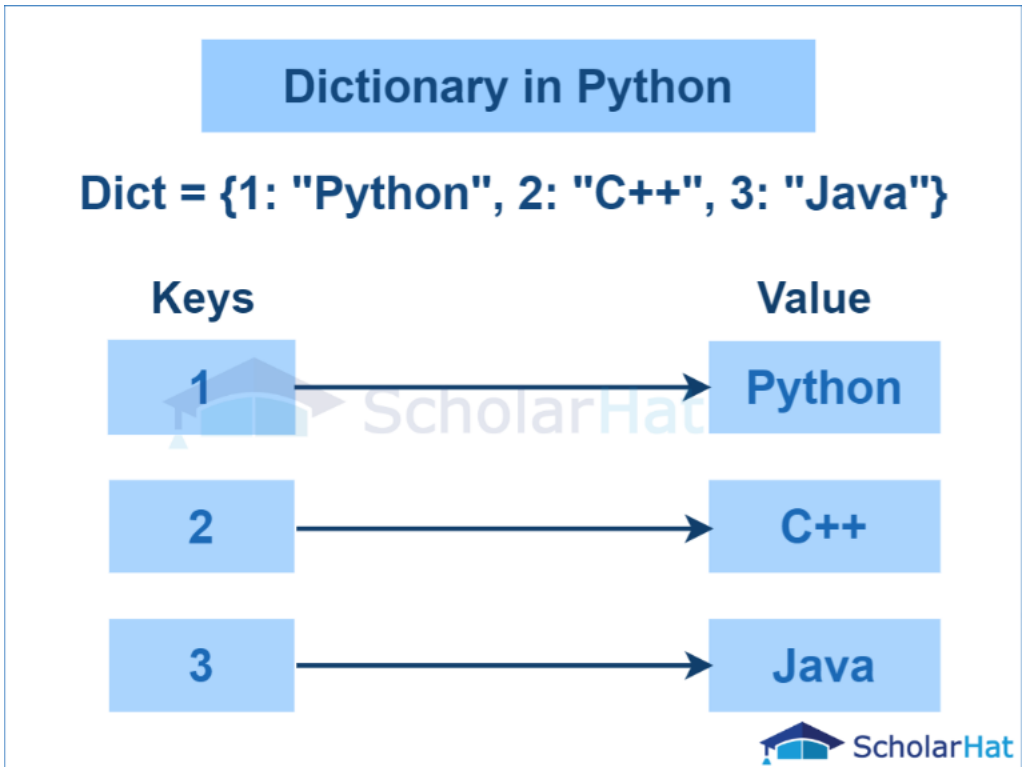
Apa itu Dictionary

Dictionary mirip dengan list dalam hal menyimpan data, tetapi lebih fleksibel karena indeksnya bisa apa saja. Di list, indeksnya berupa angka, sedangkan di dictionary, indeksnya bisa beragam, seperti kata, frasa, atau bahkan objek.

Setiap item dalam dictionary terdiri dari pasangan kunci:nilai, di mana kuncinya harus unik. Bayangkan dictionary sebagai kamus di mana setiap kata memiliki arti tertentu. Dalam Python, kata tersebut adalah kunci dan artinya adalah nilai.

Contoh:

- Kita bisa menggunakan dictionary untuk memetakan kata bahasa Inggris ke bahasa Indonesia, di mana kata bahasa Inggris menjadi kunci dan artinya dalam bahasa Indonesia menjadi nilai.
- Dictionary juga bisa digunakan untuk menyimpan profil pengguna, di mana nama pengguna menjadi kunci dan informasi seperti alamat email dan nomor telepon menjadi nilai.



Gambar 10.1 Cara Kerja Dictionary (diambil dari https://www.scholarhat.com/tutorial/python/dictionary-in-python)

Berikut adalah contoh dan cara penggunaan Dictionary dalam Python:

```
# Menggunakan fungsi dict() untuk membuat dictionary baru yang kosong (karena
'dict adalah fungsi built-in python, maka 'dict' sebaiknya tidak digunakan
sebagai nama variabel)
dictionary = dict()
print(dictionary)

# output = {}
```

```
# Tanda kurung kurawal ‘{ }’ berarti sebuah dictionary itu kosong. Tanda kurung
kotak ‘[ ]’ digunakan untuk menambahkan item ke dictionary tersebut
dictionary[1] = '1'
print(dictionary)

# output = {1: '1'}
```

```
# Format output dan input yang digunakan Dictionary sama, maka ketika kamu
membuat suatu Dict yang bar dan umencetaknya, kamu akan mendapatkan representasi
lengkap data dictionary, termasuk kunci dan nilainya.
dictionary = {'one': 1, 'two': 2, 'three': 3}
print(dictionary)

# output = {'one': 1, 'two': 2, 'three': 3}
```

Pasangan keys-values dalam dictionary di Python belum tentu berurutan. Hal ini dikarenakan dictionary bukan struktur data yang berurutan seperti list, melainkan urutannya tidak bisa diprediksi. Urutan item dalam dictionary bisa berubah saat menambahkan, menghapus, atau memodifikasi elemennya.

Meskipun tidak memiliki urutan yang terjamin, kita bisa mengakses values dengan cara mengetahui keysnya.

```
# Key 'two' dipasangkan dengan nilai 2, jadi urutan pada item tidak terlalu
dibutuhkan. Jika menggunakan kunci yang tidak ada dalam Dict makan akan terjadi
Error
print(dictionary['two'])

# output = 2

print(dictionary['satu'])
# output =     print(dictionary['satu'])
           ~~~~~~~~~~^^^^^^^^
KeyError: 'satu'
```

Dict juga memiliki beberapa fungsi dan operator yang berguna, berikut adalah kegunaan dan contoh penggunaannya:

Method	Description
clear()	Removes all the elements from the dictionary
copy()	Returns a copy of the dictionary
fromkeys()	Returns a dictionary with the specified keys and value
get()	Returns the value of the specified key
items()	Returns a list containing a tuple for each key value pair
keys()	Returns a list containing the dictionary's keys
pop()	Removes the element with the specified key
popitem()	Removes the last inserted key-value pair
setdefault()	Returns the value of the specified key. If the key does not exist: insert the key, with the specified value
update()	Updates the dictionary with the specified key-value pairs
values()	Returns a list of all the values in the dictionary

Gambar 10.2 Dictionary Methods (diambil dari https://www.w3schools.com/python/python_ref_dictionary.asp)

```
# Fungsi len untuk menghitung jumlah pasangan keys dan values
print(len(dictionary))

# output = 3

# Operator in untuk mengetahui apakah sebuah key berada pada dictionary
print('satu' in dictionary)
print('one' in dictionary)

# Tidak berlaku untuk value
print(1 in dictionary)

# output =
False
True
False
```

Untuk memeriksa apakah nilai tertentu ada dalam dictionary, kita dapat menggunakan metode ‘values()’, yang akan memberikan semua values dalam dictionary. Kemudian kita dapat mengonversi nilai-nilai tersebut menjadi daftar list menggunakan fungsi ‘list()’, dan menggunakan operator ‘in’ untuk memeriksa apakah nilai tertentu ada di dalam daftar tersebut.

```
values = list(dictionary.values())
print(1 in values)

# output = True
```

Operator ‘in’ pada list menggunakan pencarian linear, yang berarti semakin panjang listnya, semakin lama waktu yang dibutuhkan untuk mencari nilai tertentu. Namun, pada dictionary, Python menggunakan *hash table* yang memberikan waktu pencarian yang konstan, seberapa banyaknya item yang ada dalam dictionary.

Dictionary sebagai set penghitung (counters)

Ada beberapa cara untuk menghitung banyak huruf dalam sebuah string yaitu:

- Menggunakan 26 variabel untuk setiap huruf dalam alfabet, kemudian memasukkan setiap karakter dalam string ke variabel yang sesuai, menambahkan perhitungan yang tepat, dan menggunakan kondisi berantai.
- Membuat list dengan 26 elemen, kemudian mengonversi setiap karakter ke angka menggunakan fungsi bawaan, menggunakan angka sebagai indeks dalam list, dan menambahkan perhitungan yang sesuai.
- Membuat dictionary dengan karakter sebagai kunci dan jumlah kemunculan sebagai nilai. Setiap karakter ditambahkan sebagai item ke dalam dictionary, dan nilai dari setiap item kemudian ditambahkan sesuai dengan kemunculan karakter tersebut.

Kita juga bisa menggunakan komputasi model Dictionary untuk menghitung banyaknya huruf yang muncul dalam string. Menggunakan model Dictionary lebih praktis karena tidak perlu mengetahui tentang huruf mana yang akan muncul sebelumnya dalam string. Kita hanya perlu menyediakan ruang untuk semua huruf yang mungkin muncul.

Berikut adalah contoh penggunaan Dictionary untuk menghitung huruf dalam kata:

```
# Model komputasi dibawah ini disebut histogram (statistik set perhitungan)
kata = 'duta wacana'
dictionary = dict()

# Perulangan untuk setiap huruf dalam kata
for huruf in kata:
    if huruf not in dictionary:
        # Jika huruf belum ada di dict maka akan ditambah yang baru
        dictionary[huruf] = 1
    else:
        # Jika sudah ada maka keys dict akan ditambah 1 valuenya
        dictionary[huruf] += 1
print(dictionary)

output = {'d': 1, 'u': 1, 't': 1, 'a': 4, ' ': 1, 'w': 1, 'c': 1, 'n': 1}
```

Dictionary juga memiliki methods seperti ‘get()’ untuk mengambil key dan nilai default. Jika key ada pada Dict maka akan mengembalikan value sesuai keynya. Jika tidak ada, maka akan diberikan value default:

```
Dict = {'sam' : 1 , 'aldo' : 2, 'jon' : 3}
print(Dict.get('sam', 0))
print(Dict.get('amel', 0))

output = 1, 0
```

Methods ‘get()’ juga bisa digunakan dalam loop histogram.

```
 nbvkata = 'duta wacana'
dictionary = dict()

# Dengan penggunaan get(), tidak perlu digunakan 'if' karena metode 'get()' bisa
menangani kasus key tidak ada dalam Dict secara otomatis
for huruf in kata:
    dictionary[huruf] = dictionary.get(huruf, 0) + 1

print(dictionary)
output = {'d': 1, 'u': 1, 't': 1, 'a': 4, ' ': 1, 'w': 1, 'c': 1, 'n': 1}
```

Dictionary dan File

Salah satu kegunaan Dictionary yang umum adalah untuk menghitung jumlah kata-kata dalam sebuah file txt.

```
# Membuka file gtxt
handle = open("fileromeo/romeo.txt", 'r')

# Dict kosong
counts = dict()

# Looping tiap baris dalam txt
for line in handle:
    # Split baris menjadi kata
    words = line.split()

    # Looping setiap kata dalam kalimat
    for word in words:
        if word not in counts:
            # Jika kata belum ada di dict maka akan ditambah yang baru
            counts[word] = 1
        else:
            # Jika sudah ada maka keys dict akan ditambah 1 valuenya
```

```
            counts[word] += 1

print(counts)
```

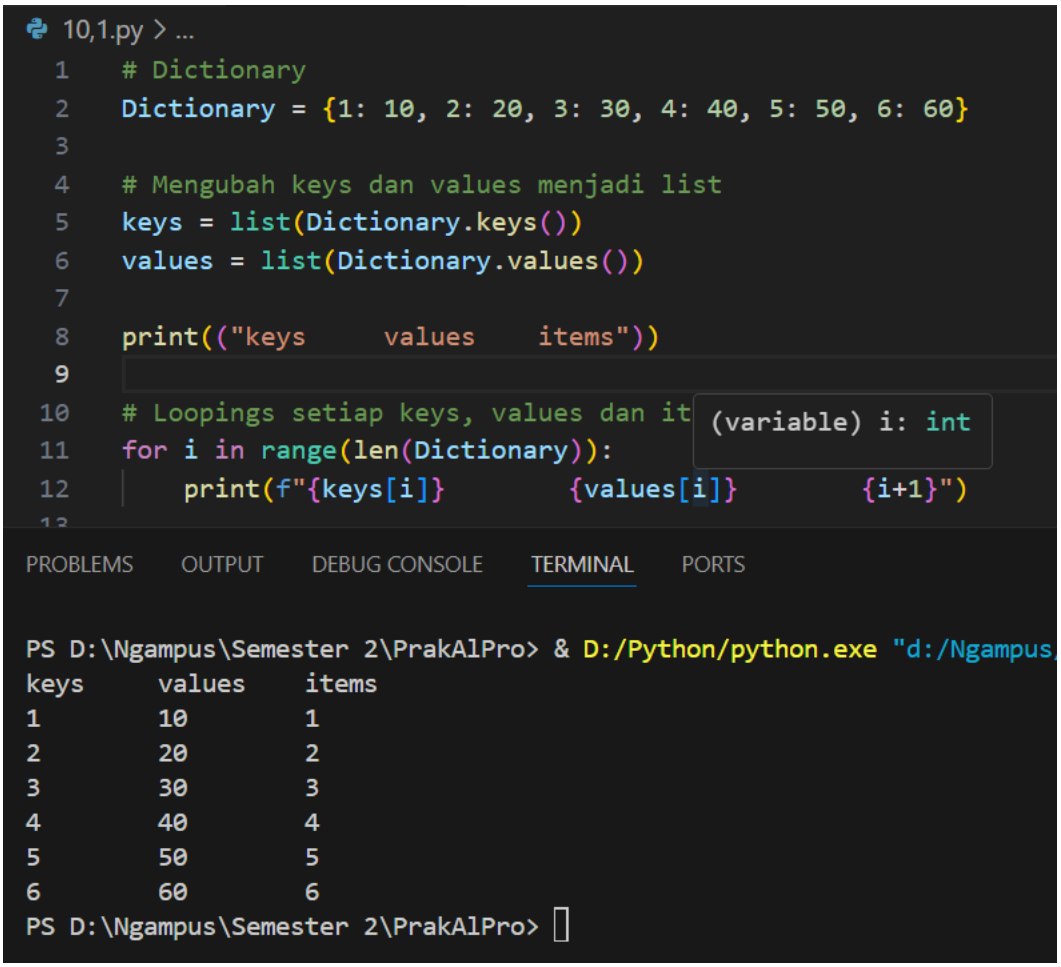
output : {'But': 1, 'soft': 1, 'what': 1, 'light': 1, 'through': 1, 'yonder': 1, 'window': 1, 'breaks': 1, 'It': 1, 'is': 3, 'the': 3, 'east': 1, 'and': 3, 'Juliet': 1, 'sun': 2, 'Arise': 1, 'fair': 1, 'kill': 1, 'envious': 1, 'moon': 1, 'Who': 1, 'already': 1, 'sick': 1, 'pale': 1, 'with': 1, 'grief': 1}

Dalam statement ‘else’, digunakan model penulisan yang lebih singkat untuk menambahkan nilai ke variabel. Misalnya, counts[word] += 1 sama dengan counts[word] = counts[word] + 1. Metode lain adalah -= untuk pengurangan, *= untuk perkalian, dan /= untuk pembagian.

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



SOAL 2

```python
# List yang ingin digabung
Lista = ['red', 'green', 'blue']
Listb = ['#FF0000','#008000', '#0000FF']

# Dict kosong
dict = {}

# Looping untuk memasukan list b ke a
i = 0
for x in Lista:
    dict[x] = Listb[i]
    i += 1

print(dict)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/
{'red': '#FF0000', 'green': '#008000', 'blue': '#0000FF'}
PS D:\Ngampus\Semester 2\PrakAlPro>
```

SOAL 3

```python
# Meminta input
input = input("Masukan nama file : ")
# Dict kosong
dict = {}
# Membuka file
with open(input, "r") as file:
    # Membaca setiap lines pada file
    lines = file.readlines()

    # Looping setiap kata dalam kalimat
    for line in lines:

        # Mencari email
        if line.startswith("From:"):
            # Ketika email sudah ditemukan maka akan displit
            kata = line.split()

            # Mengecek apakah email sudah pernah ditambah ke dict atau blm
            if kata[1] not in dict:
                # Jika belom ada maka akan dibuat dict baru
                dict[kata[1]] = 1
            else:
                # Jika sudah ada maka akan ditambah satu
                dict[kata[1]] += 1

print(dict)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Masukan nama file : 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, 'wagner
mr@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:\Ngampus\Semester 2\PrakAlPro>
```

SOAL 4

10,4.py > domain_name

```python
# Meminta input
input = input("Masukan nama file : ")

# Fungsi untuk mencari alamat domain (kata-kata setelah @)
def domain_name(email):
    # List kosong
    list = []
    # String kosong
    domain = ''

    # Looping setiap huruf di alamat email
    for x in email:
        # Memasukan setiap huruf dalam list dengan index berbeda
        list.append(x)
    # Mencari index "@"
    index = list.index("@")

    # String domain akan ditambah setiap huruf setelah "@"
    for x in list[index+1:]:
        domain += x
    return domain
```

10,4.py > domain_name

```python

# Fungsi menghitung domain
def count_domain(input):
    # Dict kosong
    dict = {}

    # Membuka file
    with open(input, "r") as file:
        # Membaca setiap lines pada file
        lines = file.readlines()

        # Looping setiap kata dalam kalimat
        for line in lines:
            # Mencari email
            if line.startswith("From:"):
                # Ketika email sudah ditemukan maka akan displit
                kata = line.split()

                # Mencari nama domain menggunakan fungsi domain_name()
                domain = domain_name(kata[1])

                # Mengecek apakah email sudah pernah ditambah ke dict
                if domain not in dict:
                    # Jika belom ada maka akan dibuat dict baru
                    dict[domain] = 1
                else:
                    # Jika sudah ada maka akan ditambah satu
                    dict[domain] += 1
    print(dict)

count_domain(input)
```

```
PS D:\Ngampus\Semester 2\PrakAlPro> & D:/Python/python.exe "d:/Ngampus/Semester 2/PrakAlPro/10,4.py"
Masukan nama file : 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}
```

Github : https://github.com/SamuelN1508/PrakAlPro10.git