



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230987
Nama Lengkap	Prastha Pradipta Purusa
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.

Materi

Dictionary mirip dengan list. . Dalam list, indeks harus berupa integer sedangkan dalam dictionary indeks bisa berupa apapun. Dictionary terdiri dari pasangan kunci dan nilai. Kunci harus bersifat unik dan tidak boleh ada dua kunci yang sama dalam satu dictionary. Kunci dan nilai disebut dengan pasangan nilai kunci (key-value pair). Fungsi dict digunakan untuk membuat dictionary kosong baru dan penggunaannya perlu dihindari sebagai nama variabel. Tanda kurung kurawal { } digunakan sebagai dictionary kosong dan untuk menambahkan item dalam dictionary menggunakan kurung kotak []. Contoh program:

```
1 source = dict()
2 print(source)
3 {}
```

Output: [Running] {}

```
5 source = dict()
6 source['source'] = 'dict'
7 print(source)
```

Output: [Running] python -u "c:\Users\HP\prasss020904\bab {'source': 'dict'}"

Kode diatas membuat item memetakan kunci source ke nilai dict dan akan menjadi pasangan nilai dan kunci. Contoh print dictionary berisi untuk menampilkan keseluruhan data:

```
9 dict = {'satu' : 'dua', 'tiga' : 'empat', 'lima': 'enam'}
10 print(dict)
```

[Running] python -u "c:\Users\HP\prasss020904\bab {'satu': 'dua', 'tiga': 'empat', 'lima': 'enam'}"

Pengurutan pasangan kunci-nilai tidaklah sama. Dalam dictionary tidak pernah diberikan indeks dengan indeks integer dan dapat menggunakan kunci untuk mencari nilai yang sesuai.

```
9 dict = {'satu' : 'dua', 'tiga' : 'empat', 'lima': 'enam'}
10 print(dict['satu'])
```

[Running] dua

Fungsi len pada dictionary digunakan untuk menghitung jumlah key:

```
8
9 dict = {'satu' : 'dua', 'tiga' : 'empat', 'lima': 'enam'}
10 print(len(dict))
```

[Running] 3

Operator in dalam dictionary menghasilkan nilai true atau false sesuai dengan kunci yang ada di dictionary.

```

9 dict = {'satu' : 'dua', 'tiga' : 'empat', 'lima': 'enam'}
10 dict = 'satu' in dict
11 print(dict)

```

```

[Running] py
True

```

```

9 dict = {'satu' : 'dua', 'tiga' : 'empat', 'lima': 'enam'}
10 dict = 'enam' in dict
11 print(dict)

```

```

[Running] py
False

```

Untuk mengetahui nilai yang ada pada dictionary dapat menggunakan method values. Values memunculkan nilai sesuai dengan tipe data dan dikonversi ke dalam list menggunakan in.

```

9 dict = {'satu' : 'dua', 'tiga' : 'empat', 'lima': 'enam'}
10 dicts = list(dict.values())
11 dicts = 'dua' in dicts
12 print(dicts)

```

```

[Running] py
True

```

List dan dictionary menggunakan algoritma yang berbeda. List algoritma pencarian linear dan waktu pencariannya tergantung kepada panjang list. Python menggunakan struktur data yang disebut hash table untuk dictionary dengan operator in dan tidak memperdulikan banyak item yang ada didalam dictionary untuk di proses.

Dictionary sebagai set penghitung (counters)

Contoh program menghitung jumlah kata dengan counters:

```

14 kata = 'jflkdjsfkljsdfjl0tirier'
15 c = 0
16 d = {}
17 for c in kata:
18     d[c] = d.get(c, 0) + 1
19 print(d)

```

```

[Running] python -u "c:\Users\HP\prasss020904\bab11\source code.py"
{'j': 4, 'f': 3, 'l': 3, 'k': 2, 'd': 2, 's': 2, '0': 1, 't': 1, 'i': 2, 'r': 2, 'e': 1}

```

Fungsi `get` digunakan untuk mengambil nilai kunci dan nilai default.. Jika dibandingkan loop menggunakan pernyataan `if` dan operator `in` dengan loop menggunakan metode `get` melakukan hal yang persis sama, tetapi yang satu lebih ringkas.

Dictionary dan File

Contoh program yang memecah setiap baris menjadi daftar kata, dan kemudian melakukan perulangan melalui setiap kata dalam baris dan menghitung setiap kata menggunakan dictionary.

```
berita.py > ...
1 file = open('idk.txt', 'r')
2 c = dict()
3 c = {}
4 for line in file:
5     words = line.split()
6     for word in words:
7         if word not in c:
8             c[word] = 1
9         else:
10            c[word] += 1
11 file.close()
12 print(c)
```

```
[Running] python -u "c:\Users\HP\prasss020904\bab11\berita.py"
{'kebanyakan': 1, 'ya': 2, 'isi': 2, 'berita': 2, 'semua': 1, 'sama': 1, 'aja': 1, 'wkwk': 1, 'gitu': 1, 'deh': 1}
```

Perulangan bagian luar untuk membaca baris file dan perulangan pada dalam melakukan perulangan melalui setiap kata pada baris tertentu dan disebut nested loop karena salah satu perulangan adalah perulangan bagian luar dan perulangan lainnya adalah perulangan bagian dalam.

Looping dan Dictionary

Dictionary akan bekerja dengan cara menelusuri kunci yang ada didalamnya dengan fungsi `for` dan membuat output setiap kunci dengan nilainya. Contoh program:

```
dict = { 'wisnu' : 974 , 'jimmy' : 975, 'calvin': 988, 'prastha': 987}
for key in dict:
    print(key, dict[key])
```

```
wisnu 974
jimmy 975
calvin 988
prastha 987
```

Program tersebut membuat hanya key dan value yang muncul. Key berupa seperti 'wisnu' dan yang angka seperti :974 berupa value. Berikut adalah contoh program ada maksimal pengambilan valuenya:

```
dict = { 'wisnu' : 974 , 'jimmy' : 975, 'calvin': 988, 'prastha': 987}
for key in dict:
    if dict[key] <= 987:
        print(key, dict[key])
```

```
wisnu 974
jimmy 975
prastha 987
```

Untuk memunculkan output kunci dalam urutan alfabet dari dictionary dapat menggunakan list() pada dictionary kemudian mengurutkan dengan sort(). Contoh program:

```
dict = { 'wisnu' : 974 , 'jimmy' : 975, 'calvin': 988, 'prastha': 987}
sorted = list(dict.keys())
sorted.sort()
for key in sorted:
    print(key, dict[key])
```

```
[Running] pyth
calvin 988
jimmy 975
prastha 987
wisnu 974
```

Advanced Text Parsing

Python sendiri mempunyai fungsi split untuk mencari spasi dan mengubah kata dan dipisahkan oleh spasi dengan contoh kata 'Hai!' dan 'Hai' akan terpisah di dictionary karena dua kata berbeda. Perbedaan huruf kapital juga merupakan dua kata berbeda di dictionary dengan contoh 'Hai' dan 'hai'. Contoh program:

```
15 file = open('hai.txt', 'r')
16 c = {}
17 for i in file:
18     kata = i.split()
19     for j in kata:
20         c[j] = c.get(j, 0) + 1
21 file.close()
22 print(c)
```

```
[Running] python -u "c:\Users\HP\prasss020904\b
{'Hai': 1, 'Hai!': 1, 'hai?': 1, 'hai': 1}
```

Tetapi bisa menggunakan lower atau upper supaya 'Hai' dan 'hai' bisa menjadi satu kata yang sama di dalam dictionary. Contoh program:

```

25 file = open('hai.txt', 'r')
26 c = {}
27 for i in file:
28     kata = i.lower().split()
29     for j in kata:
30         c[j] = c.get(j, 0) + 1
31 file.close()
32 print(c)

```

```

[Running] python -u "c:\Users\HP\pras
{'hai': 2, 'hai!': 1, 'hai?': 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

The screenshot shows a Jupyter Notebook window titled "Prastha Pradipta Purusa(71230987)Laprak 11 Last Checkpoint:". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The code cell contains the following Python code:

```

In [1]: dictionary = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
print("Dictionary :",dictionary)
print("key\tvalue\titem")
for key , value in dictionary.items():
    print(key,"\t",value,"\t",key)

```

The output of the code is displayed below the code cell:

```

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

```

Line1: isi dari dictionary

Line2: mencetak output "Dictionary:" dan isi dari dictionary

Line3: mencetak output dari key, value, item dan fungsi /t adalah untuk tab antar output

Line4: loop for yang digunakan untuk mengakses setiap pasangan kunci-nilai dalam dictionary. items() digunakan untuk mendapatkan pasangan kunci-nilai dalam bentuk tuple.

Line5: memunculkan nilai key,value,items dan untuk items menggunakan key karena sama dengan key. /t untuk tab.

SOAL 2

```
In [2]: Lista = ['red', 'green', 'blue']
Listb = ['#FF0000', '#008000', '#0000FF']
output = {}
for i in range(len(Lista)):
    output[Lista[i]] = Listb[i]
print("Data List")
print("Lista =", Lista)
print("Listb =", Listb)
print("Output")
print(output)

Data List
Lista = ['red', 'green', 'blue']
Listb = ['#FF0000', '#008000', '#0000FF']
Output
{'red': '#FF0000', 'green': '#008000', 'blue': '#0000FF'}
```

Line1: isi dari Lista

Line2: isi dari Listb

Line3: membuat dictionary kosong dengan{}

Line4: Memulai loop for untuk melakukan perulangan sebanyak isi di Lista. Fungsi len(Lista) mengembalikan panjang list Lista dan range(len(Lista)) menghasilkan bilangan dari 0 hingga panjang list Lista dikurangi 1.

Line5: Dalam setiap perulangan loop kode ini membuat pasangan kunci-nilai baru dalam dictionary output dengan memetakan setiap "item" dalam Lista ke "nilai" yang sesuai dalam Listb menggunakan indeks i untuk mengakses keduanya.

Line6: Melakukan output kata "Data List"

Line7: Melakukan output 'Lista =', Lista untuk memunculkan isi di List a

Line8: Melakukan output 'Listb =', Listb untuk memunculkan isi di List b

Line9: Melakukan output kata 'Output'

Line10: Melakukan output isi dari dictionary output yang berisi Lista dan Listb

SOAL 3

```
In [ ]: filename = input("Masukkan nama file: ")
file = open(filename, 'r')
dict = {}

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

file.close()
print(dict)
```

```

./..\\debugpy\\launcher -59217 -- "c:\\Users\\HP\\prasss020904\\bab11\\no3.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 C:\\Users\\HP\\prasss020904\\bab11>

```

Line1: filename untuk input nama file

Line2: variabel file untuk membuka dan membaca file

Line3: dictionary kosong

Line4: Loop for membaca setiap baris dari file yang dibuka dan menyimpannya dalam variabel i satu per satu.

Line5: mengecek apakah baris dimulai dari string 'From '

Line6: Jika baris dimulai dengan 'From ' maka baris kode tersebut akan membagi baris tersebut menjadi terpisah dan menyimpan kata kedua (indeks 1) ke dalam variabel email.

Line7: Menambahkan alamat email ke dalam dict jika belum ada atau menambhkan jumlahnya jika sudah ada.

Line8: Menutup file setelah dibaca

Line9: Mengeluarkan output dict yang berisi jumlah kemunculan setiap alamat email.

SOAL 4

```

]: filename = input("Masukkan nama file: ")
file = open(filename, 'r')
dict = {}

for i in file:
    if i.startswith('From '):
        email = i.split()[1]
        domain = email.split('@')[-1]
        dict[domain] = dict.get(domain, 0) + 1

file.close()
print(dict)

```

```

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}

```

Line1: filename untuk input nama file

Line2: variabel file untuk membuka dan membaca file

Line3: dictionary kosong

Line4: Loop for membaca setiap baris dari file yang dibuka dan menyimpannya dalam variabel i satu per satu.

Line5: mengecek apakah baris dimulai dari string 'From '

Line6: Jika baris dimulai dengan 'From ' maka baris kode tersebut akan membagi baris tersebut menjadi terpisah dan menyimpan kata kedua (indeks 1) ke dalam variabel email.

Line7: Menambahkan alamat email ke dalam dict jika belum ada atau menambahkan jumlahnya jika sudah ada.

Line8: Membagi alamat email di variabel email menjadi dua bagian berdasarkan simbol '@' dan menyimpan bagian terakhir (indeks -1) ke dalam variabel domain dan akan menghasilkan domain dari alamat email.

Line9: Menutup file setelah dibaca

Line10: Mengeluarkan output dict yang berisi jumlah kemunculan setiap alamat email.

Link github:

<https://github.com/prasss020904/Laprak11-71230987.git>