



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

| | |
|---------------------------|----------------------------------|
| NIM | 71231052 |
| Nama Lengkap | GABRIEL SACHIO ATMADJAJA |
| 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

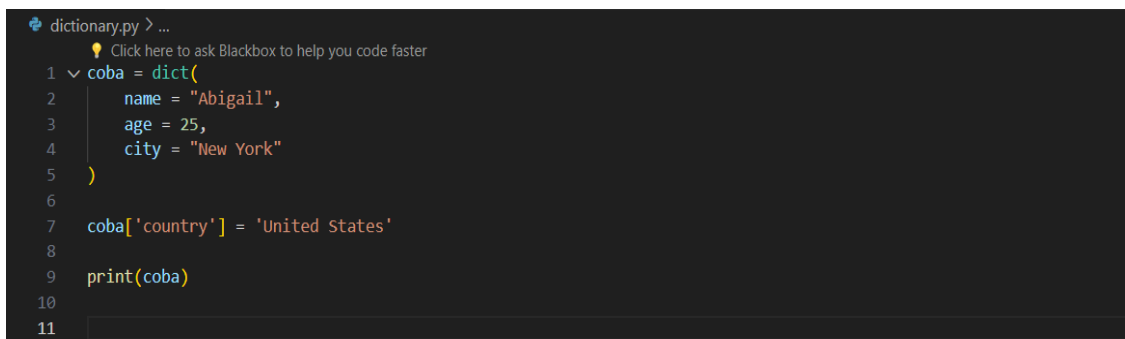
BAGIAN 1: MATERI MINGGU INI (40%)

Sumber: Modul Pratikum Algoritma dan Pemrograman,

<https://www.programiz.com/python-programming/methods/string/translate>

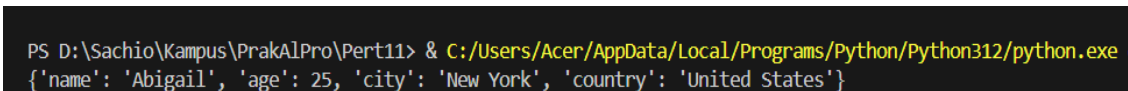
MATERI 1

Dictionary mirip dengan list, tetapi indeksnya bisa bermacam-macam. Dictionary memiliki anggota yang terdiri dari pasangan kunci:nilai. Kunci harus bersifat unik, tidak boleh ada dua kunci yang sama dalam satu dictionary. Fungsi dict untuk membuat dictionary baru yang kosong (hindari dict sebagai nama variabel). Untuk menambahkan item ke dalam dictionary, gunakan kurung siku. Contoh:



```
dictionary.py > ...
  Click here to ask Blackbox to help you code faster
1  ▾ coba = dict(
2      name = "Abigail",
3      age = 25,
4      city = "New York"
5  )
6
7  coba['country'] = 'United States'
8
9  print(coba)
10
11
```

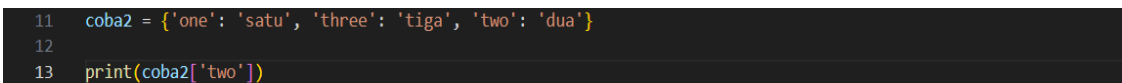
Gambar 1.1: Contoh dictionary



```
PS D:\Sachio\Kampus\PrakAlPro\Pert11> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
{'name': 'Abigail', 'age': 25, 'city': 'New York', 'country': 'United States'}
```

Gambar 1.2: Output

Urutan dalam dictionary tidak menjadi masalah karena tinggal memanggil kunci-nya saja, otomatis outputnya adalah nilai dari pasangan kunci tersebut, contoh:



```
11  coba2 = {'one': 'satu', 'three': 'tiga', 'two': 'dua'}
12
13  print(coba2['two'])
```

Gambar 1.3: Terkait urutan di dictionary

Jika mencoba memanggil dengan nilai, maka outputnya error.

FUNGSI LEN

Jika mencoba menggunakan fungsi len pada variable di gambar 1.3, maka outputnya itu jumlah pasangan kunci dan nilai. Jadi, outputnya nanti adalah 3.

OPERATOR IN

Intinya, hanya bisa mengecek apakah ada key tersebut didalam dictionary.. jika mencoba mengecek nilainya, maka akan mengembalikan False (jika memang tidak ada)

```
19 # OPERATOR IN
20 print('one' in coba2)
21 print('satu' in coba2)
```

Gambar 1.4: Penggunaan operator in di dictionary (variabel dari gambar 1.3)

```
PS D:\Sachio\Kampus\PrakAlPro\Pert11> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
True
False
```

Gambar 1.5: Output

DICTIONARY SEBAGAI COUNTER / SET PENGHITUNG

```
23 # DICTIONARY SBG COUNTER/SET PENGHITUNG
24 word = 'brontosaurus'
25 d = dict()
26 for char in word:
27     if char not in d:
28         d[char] = 1
29     else:
30         d[char] += 1
31 print(d)
```

Gambar 1.6: Contoh counter di dictionary

```
PS D:\Sachio\Kampus\PrakAlPro\Pert11> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
{'b': 1, 'r': 2, 'o': 2, 'n': 1, 't': 1, 's': 2, 'a': 1, 'u': 2}
```

Gambar 1.7: Ouput

METHOD GET

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

```
33 # METHOD GET
34 counts = {'chuck' : 1, 'annie' : 42, 'jan': 100}
35 print(counts.get('jan', 0))
36 print(counts.get('tim', 0)) # 0 sbg nilai default jika kunci tdk ada... jika tidak diberi 0, outputnya adalah none
37
```

Gambar 1.8: Penggunaan method get

```
PS D:\Sachio\Kampus\PrakAlPro\Pert11> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
100
0
```

Gambar 1.9: Output

PENYEDERHANAN LOOP FOR DGN METHOD GET

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

Gambar 1.10: Method get utk menyederhanakan loop

```
PS D:\Sachio\Kampus\PrakAlPro\Pert11> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d
{'b': 1, 'r': 2, 'o': 2, 'n': 1, 't': 1, 's': 2, 'a': 1, 'u': 2}
PS D:\Sachio\Kampus\PrakAlPro\Pert11>
```

Gambar 1.11: Output

DICTIONARY DAN FILE

```
44 # DICTIONARY DAN FILE
45 fname = input('Enter the file name: ')
46 try:
47     fhand = open(fname)
48 except:
49     print('File cannot be opened:', fname)
50     exit()
51
52 counts = dict()
53 for line in fhand:
54     words = line.split()
55     for word in words:
56         if word not in counts:
57             counts[word] = 1
58         else:
59             counts[word] += 1
60
61 print(counts)
```

Gambar 1.12: Program membaca file dalam dictionary

```
Enter the file name: txt-romeo/romeo.txt
{'Who': 1, 'is': 1, 'already': 1, 'sick': 1, 'and': 1, 'pale': 1, 'with': 1, 'grief': 1}
```

Gambar 1.13: Output

LOOPING DICTIONARY

```
63 # LOOPING DICTIONARY
64 counts = { 'chuck' : 1 , 'annie' : 42, 'jan': 100}
65 for key in counts:
66     if counts[key] > 10 :
67         print(key, counts[key])
```

Gambar 1.14: Perulangan di dictionary

```
PS D:\Sachio\Kampus\PrakAlPro\Pert11> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
annie 42
jan 100
PS D:\Sachio\Kampus\PrakAlPro\Pert11>
```

Gambar 1.15: Output

```
70 counts = { 'chuck' : 1 , 'annie' : 42, 'jan': 100}
71 lst = list(counts.keys())
72 print(lst)
73 lst.sort()
74 for key in lst:
75     print(key, counts[key])
```

Gambar 1.16: Contoh lain

```
PS D:\Sachio\Kampus\PrakAlPro\Pert11> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
['chuck', 'annie', 'jan']
annie 42
chuck 1
jan 100
```

Gambar 1.17: Output

ADVANCE TEXT PARSING

Method string selain split yaitu lower, punctuation dan translate. Method lower untuk mengubah string menjadi huruf non-kapital, method punctuation gampangnya untuk memberikan rangkaian tanda baca, dan method translate untuk mengganti/menerjemahkan karakter dalam string yang diberikan sesuai table pemetaan. Berikut ini merupakan dokumentasi dari method translate.

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

Ganti karakter pada fromstr dengan karakter pada posisi yang sama dengan tostr dan hapus semua karakter yang ada dalam deletetr. Untuk fromstr dan tostr dapat berupa string kosong dan untuk parameter pada deletetr dapat dihilangkan.

Parameter deletetr untuk menghapus semua tanda baca. Secara otomatis Python akan memberitahu bagian mana yang dianggap sebagai "tanda baca".

```
>>> import string
>>> string.punctuation
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

```

77 # ADVANCED TEXT PARSING
78 import string
79
80 fname = input('Enter the file name: ')
81 try:
82     fhand = open(fname)
83 except:
84     print('File cannot be opened:', fname)
85     exit()
86
87 counts = dict()
88 for line in fhand:
89     line = line.rstrip()
90     line = line.translate(line.maketrans('', '', string.punctuation))
91     line = line.lower()
92     words = line.split()
93     for word in words:
94         if word not in counts:
95             counts[word] = 1
96         else:
97             counts[word] += 1
98
99 print(counts)

```

Gambar 1.18: Program advance parsing text

```

PS D:\Sachio\Kampus\PrakAlPro\Pert11> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe
Enter the file name: txt-romeo/romeo.txt
{'who': 1, 'is': 1, 'already': 1, 'sick': 1, 'and': 1, 'pale': 1, 'with': 1, 'grief': 1}

```

Gambar 1.19: Outputs

BAGIAN 2: LATIHAN MANDIRI (60%)

Source: <https://github.com/gabrielsachioa/PrakAIPro10.git>

SOAL3 → prompt gemini: 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. Jangan gunakan import re dan jangan gunakan if __name__. Pastikan outputnya dalam bentuk dictionary

SOAL4 → prompt gemini: buat program untuk mencatat data nama domain pengirim pesan. Hitunglah jumlah pesan yang dikirim masing-masing domain. sajikan dalam bentuk dictionary. Jangan gunakan import re dan jangan gunakan if __name__. Pastikan outputnya dalam bentuk dictionary

SOAL 1

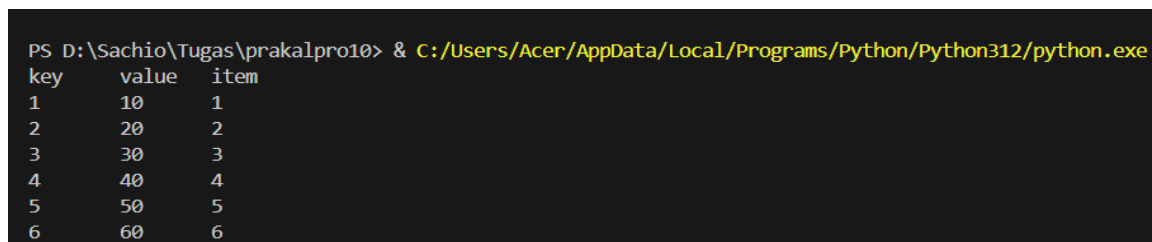


```
lat10-1.py > ...  
Click here to ask Blackbox to help you code faster  
1 sample = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}  
2  
3 item = 0  
4 print(f"key\tvalue\titem")  
5 for key in sample:  
6     item += 1  
7     print(f"{key}\t{sample[key]}\t{item}")
```

Gambar 2.1: Program

Keterangan program:

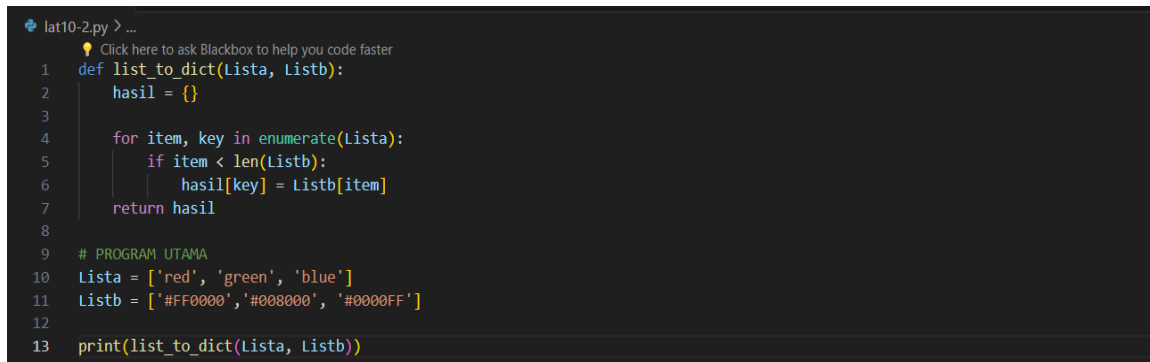
Intinya, sample itu menyimpan key dan valuenya (dictionary). Lalu variabel item = 0 untuk menghitung jumlah itemnya nanti. Lalu baris 4 itu akan menampilkan headernya (key, value, item). Lalu buat perulangan for key in sample.. Itemnya di-increment, lalu tinggal buat kode di baris 7 untuk menampilkan masing-masing key, value, dan itemnya



```
PS D:\Sachio\Tugas\prakalpro10> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe  
key    value  item  
1      10     1  
2      20     2  
3      30     3  
4      40     4  
5      50     5  
6      60     6
```

Gambar 2.2: Output

SOAL 2

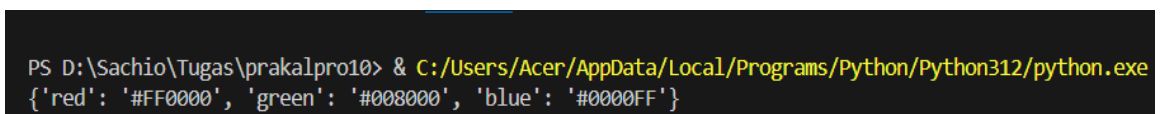
A screenshot of a code editor showing a Python script. The script defines a function `list_to_dict` that takes two lists, `Lista` and `Listb`, and returns a dictionary. The function uses `enumerate` to iterate over `Lista` and checks if the index is within the length of `Listb`. If so, it assigns `Listb[item]` to the value for `key` in the dictionary. The main program defines `Lista` as `['red', 'green', 'blue']` and `Listb` as `['#FF0000', '#008000', '#0000FF']`, then prints the result of `list_to_dict(Lista, Listb)`.

```
lat10-2.py > ...  
Click here to ask Blackbox to help you code faster  
1 def list_to_dict(Lista, Listb):  
2     hasil = {}  
3  
4     for item, key in enumerate(Lista):  
5         if item < len(Listb):  
6             hasil[key] = Listb[item]  
7     return hasil  
8  
9 # PROGRAM UTAMA  
10 Lista = ['red', 'green', 'blue']  
11 Listb = ['#FF0000', '#008000', '#0000FF']  
12  
13 print(list_to_dict(Lista, Listb))
```

Gambar 2.3: Program

Keterangan program:

Pertama, buat fungsi dengan nama `list_to_dict` dan parameternya diisi dari program utama nanti. Di dalam fungsinya, buat dictionary kosong dalam variable `hasil` untuk menampung outputnya nanti. Buat perulangan `for item, key in enumerate(Lista)`, `enumerate` itu akan mengakses elemen dan indeks dari `Lista`. Lalu di dalamnya buat `if item < len(listb)` (jika item yang ada di `Lista` kurang dari panjang elemen di `Listb`) → `hasil[key]` untuk mengatur key-nya dan `Listb[item]` untuk mengisi value-nya. Lalu `return` hasilnya. Baris 10 adalah list yang akan menjadi key-nya, lalu baris 11 adalah list yang nanti jadi value. Lalu baris 13 untuk mem-print fungsi itu

A screenshot of a terminal window showing the execution of the Python script. The command `C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe` is run, and the output is a dictionary: `{'red': '#FF0000', 'green': '#008000', 'blue': '#0000FF'}`.

```
PS D:\Sachio\Tugas\prakalpro10> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe  
{'red': '#FF0000', 'green': '#008000', 'blue': '#0000FF'}
```

Gambar 2.4: Output

SOAL 3

```
lat10-3.py > ...
Click here to ask Blackbox to help you code faster
1 def parse_email_log(filename):
2     """Membaca log email dan mengembalikan dictionary histogram dan dictionary jumlah pesan."""
3     histogram = {}
4     email_counts = {}
5
6     with open(filename, 'r') as file:
7         for line in file:
8             # Cari baris yang berisi alamat email
9             email_index = line.find("From:")
10            if email_index != -1:
11                email_end_index = line.find("\n", email_index)
12                email = line[email_index + 6:email_end_index].strip()
13
14                # Hitung jumlah pesan dari email ini
15                if email not in email_counts:
16                    email_counts[email] = 0
17                email_counts[email] += 1
18
19                # Dapatkan kata-kata dalam baris
20                words = line.split()
21                words = [word for word in words if word != "From:"] # Hapus "From:"
22                words = [word for word in words if word != email] # Hapus alamat email
23
24                # Perbarui histogram kata
25                for word in words:
26                    if word not in histogram:
27                        histogram[word] = 0
28                    histogram[word] += 1
29
30    return histogram, email_counts
```

Gambar 2.5: Fungsi

```
32 # Program utama
33 filename = "mbox-short.txt"
34 histogram, email_counts = parse_email_log(filename)
35
36 print(email_counts)
37
```

Gambar 2.6: Program Utama

```
{'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}
```

Gambar 2.7: Output

SOAL 4

```
lat10-4.py > ...
Click here to ask Blackbox to help you code faster
1 def parse_email_log(filename):
2     """Membaca log email dan mengembalikan dictionary nama domain dan jumlah pesan."""
3     domain_counts = {}
4
5     with open(filename, 'r') as file:
6         for line in file:
7             # Cari baris yang berisi alamat email
8             email_index = line.find("From:")
9             if email_index != -1:
10                 email_end_index = line.find("\n", email_index)
11                 email = line[email_index + 6:email_end_index].strip()
12
13                 # Dapatkan nama domain pengirim
14                 domain_start_index = email.find("@")
15                 if domain_start_index != -1:
16                     domain_end_index = email.find(" ", domain_start_index)
17                     if domain_end_index == -1:
18                         domain_end_index = len(email)
19                     domain = email[domain_start_index + 1:domain_end_index]
20
21                 # Hitung jumlah pesan dari domain ini
22                 if domain not in domain_counts:
23                     domain_counts[domain] = 0
24                 domain_counts[domain] += 1
25
26     return domain_counts
27
28 # Program utama
29 filename = "mbox-short.txt"
30 domain_counts = parse_email_log(filename)
31
32 print(domain_counts)
33
```

Gambar 2.9: Program

```
PS D:\Sachio\Tugas\prakalpro10> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d:/Sachio/Tugas/prakalpro10/lat10-4.py
{'uct.ac.za': 6, 'media.berkeley.edu': 4, 'umich.edu': 7, 'iupui.edu': 8, 'caret.cam.ac.uk': 1, 'gmail.com': 1}
```

Gambar 2.10: Output