

Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024



NIM	71231006
Nama Lengkap	Fransiskus Tedyanto seran
Minggu ke / Materi	10 / 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
2023

BAGIAN 1: MATERI MINGGU INI (40%)

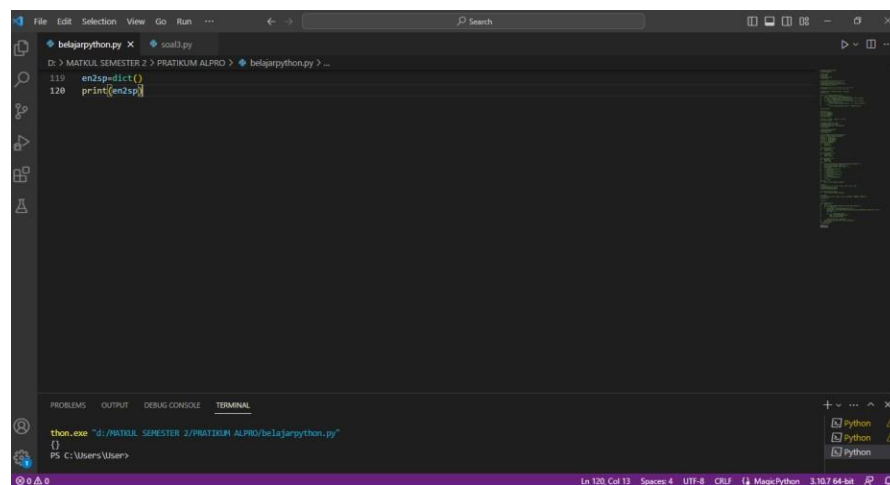
Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

MATERI 1

Materi

Tipe data dictionary merupakan salah satu tipe data yang digunakan dalam pemrograman untuk menyimpan data dalam bentuk yaitu key-value. Dalam dictionary setiap elemen terdiri dari sepasang key dan nilai yang terkait satu sama lain. key dapat berupa sebuah string atau bilangan bulat, sedangkan nilai dapat berupa tipe data apapun seperti, string, bilangan bulat, float, list, tuple, atau bahkan dictionary lainnya.

Fungsi dict digunakan untuk membuat suatu dictionary baru yang kosong. Karena dict merupakan built-in-function dari python, maka penggunaannya perlu dihindari sebagai nama variabel. Seperti ini contohnya:



```
File Edit Selection View Go Run ... Search
belajarpython.py x xall.py
D:\> MATKUL_SEMESTER_2\PRATIUM ALPRO> belajarpython.py ...
119 en2sp=dict()
120 print(en2sp)
```

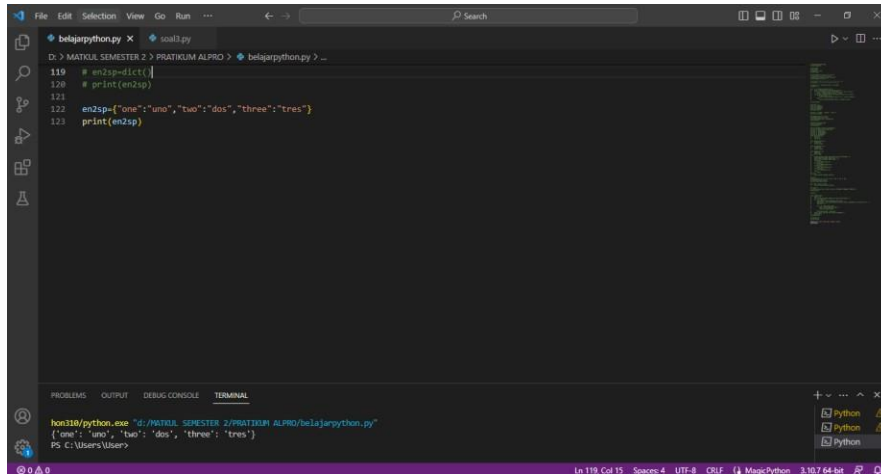
thon.exe "d:/MATKUL_SEMESTER_2/PRATIUM ALPRO/belajarpthon.py"

{}
PS C:\Users\User>

Ln 120, Col 13 Spaces: 4 UTF-8 CRLF MagicPython 3.10.7 64-bit

Program di atas membuat sebuah dictionary kosong dengan nama en2sp, dengan menggunakan fungsi dict(), dan kemudian mencetak isi dictionary tersebut menggunakan fungsi print(). Dikarenakan dictionary masih kosong, maka output yang dihasilkan akan menjadi {}, atau dictionary kosong.

Sebagai contoh yang lain, kita akan mengembangkan kamus yang memetakan dari kata-kata dalam bahasa Inggris ke bahasa spanyol, jadi bisa dikatakan yang menjadi kunci dan nilai adalah data string.

A screenshot of a Python IDE window. The main editor shows a Python script with the following code:

```
119 # en2sp=dict()  
120 # print(en2sp)  
121  
122 en2sp={"one":"uno","two":"dos","three":"tres"}  
123 print(en2sp)
```

The terminal at the bottom shows the output of the script:

```
nonim/python.exe "C:\PROGRA.../PRATIKUM ALPRO/belajarpem.py"  
{'one': 'uno', 'two': 'dos', 'three': 'tres'}  
PS C:\Users\User>
```

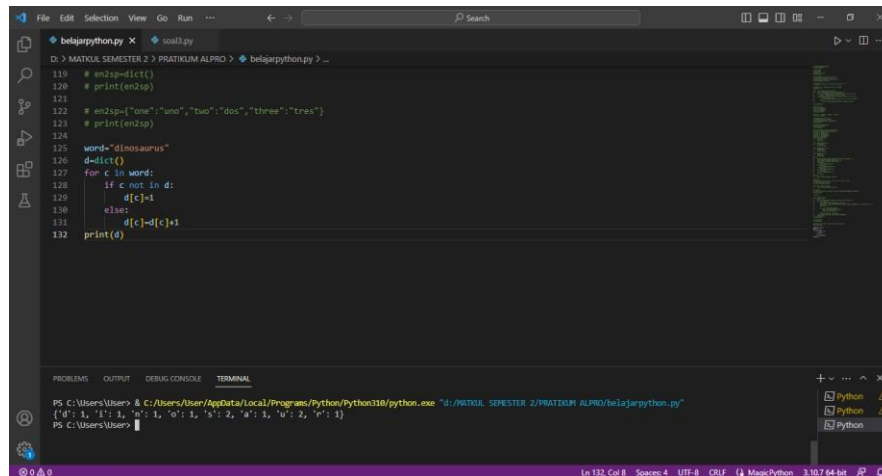
Pada program di atas membuat sebuah dictionary dengan nama `en2sp`, dan diisi dengan tiga pasangan kunci-nilai menggunakan format `{kunci:nilai}`. kunci `"one"` dengan memiliki nilai `"uno"`, dan seterusnya. Setelah dictionary terisi dengan pasangan kunci-nilai, program kemudian mencetak isi dictionary tersebut dengan menggunakan perintah `print`. Hasil yang dikeluarkan menunjukkan isi dari dictionary `en2sp`. Kunci-kunci dictionary dituliskan sebagai string, dan dipisahkan dengan nilai-nilai yang sesuai oleh tanda `":"` seluruh pasangan kunci-nilai tersebut dikelilingi oleh tanda `"{"}` yang menandakan bahwa itu adalah dictionary.

MATERI 2

Dictionary sebagai set penghitung (counters)

Dictionary dapat juga digunakan sebagai set penghitung (counters) karena dictionary memungkinkan kita untuk menyimpan pasangan kunci-nilai dimana kunci berupa elemen-elemen dari suatu set, sedangkan nilai berupa frekuensi kemunculan elemen tersebut dalam set. Dalam penggunaannya sebagai set penghitung dictionary juga disebut sebagai `"hash table"`

Sebagai contoh, kita dapat menggunakan dictionary sebagai set penghitung untuk menghitung kemunculan kata-kata dalam suatu teks. Dalam hal ini, set kata-kata dapat dianggap sebagai kunci, sedangkan frekuensi kemunculan kata-kata tersebut dianggap sebagai nilai dalam dictionary. Contohnya sebagai berikut:

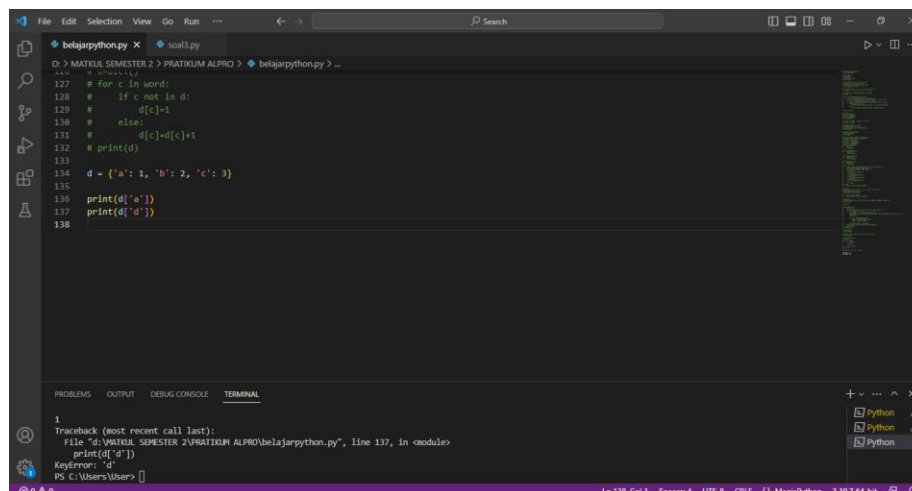


```
119 # en2ip=dict()
120 # print(en2ip)
121
122 # en2ip={"one": "uno", "two": "dos", "three": "tres"}
123 # print(en2ip)
124
125 word="dinosaur"
126 d=dict()
127 for c in word:
128     if c not in d:
129         d[c]=1
130     else:
131         d[c]=d[c]+1
132 print(d)
```

```
PS C:\Users\User & C:\Users\User\AppData\Local\Programs\Python\Python38\python.exe "D:\MATKUL SEMESTER 2\PRATIUM ALPRO\belajarpython.py"
{'d': 1, 'i': 1, 'n': 1, 'o': 1, 's': 2, 'a': 1, 'u': 2, 'r': 1}
PS C:\Users\User >
```

Pada program di atas membuat sebuah dictionary kosong bernama “d”. kemudian program mengiterasi setiap karakter dalam kata “dinosaur” menggunakan loop for. Untuk setiap karakter program memeriksa apakah karakter tersebut sudah ada di dalam dictionary “d”. jika karakter belum ada, maka program akan menambahkan karakter tersebut sebagai kunci baru di dalam dictionary “d”, dan menetapkan nilai 1 sebagai nilai awal untuk kunci tersebut. namun jika karakter sudah ada di dalam dictionary, maka program akan menambahkan 1 pada nilai yang sudah ada untuk kunci tersebut. setelah program selesai menghitung frekuensi masing-masing huruf, program mencetak hasilnya dalam bentuk dictionary yang berisi kunci yaitu (huruf), dan nilai yaitu (frekuensi masing-masing huruf dalam kata “dinosaur”)

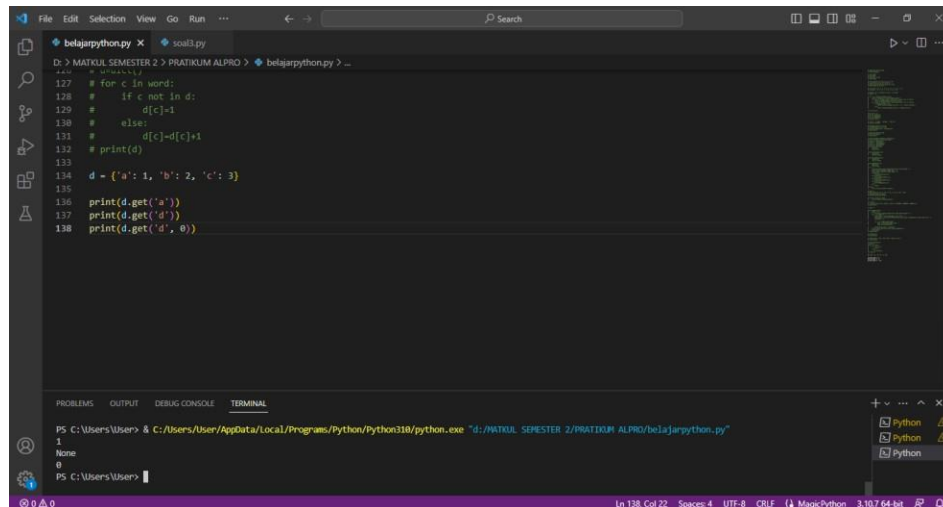
Dictionary juga ada namanya get metode ini digunakan untuk mendapatkan nilai yang terkait dengan kunci tertentu. Metode ini mengambil satu argumen yang merupakan kunci yang akan dicari di dalam dictionary. Perbedaan antara metode get dengan cara mengakses nilai langsung menggunakan kunci, adalah jika kunci tidak ditemukan di dalam dictionary, metode get akan mengembalikan nilai default, sedangkan cara mengakses nilai secara langsung akan menghasilkan keyerror jika kunci tidak ditemukan. Seperti contoh di bawah ini merupakan kita mengakses nilai secara langsung dengan menggunakan kunci, dan kita menggunakan metode get:



```
127 # for c in word:
128 #     if c not in d:
129 #         d[c]=1
130 #     else:
131 #         d[c]=d[c]+1
132 # print(d)
133
134 d = {'a': 1, 'b': 2, 'c': 3}
135
136 print(d['a'])
137 print(d['d'])
138
```

```
1
Traceback (most recent call last):
  File "D:\MATKUL SEMESTER 2\PRATIUM ALPRO\belajarpython.py", line 137, in module
    print(d['d'])
KeyError: 'd'
PS C:\Users\User >
```

Mengakses nilai secara langsung



```
127 # for c in word:
128 #     if c not in d:
129 #         d[c]=1
130 #     else:
131 #         d[c]=d[c]+1
132 # print(d)
133
134 d = {'a': 1, 'b': 2, 'c': 3}
135
136 print(d.get('a'))
137 print(d.get('d'))
138 print(d.get('d', 0))
```

Terminal Output:

```
PS C:\Users\User> & C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe "d:/MATKUL SEMESTER 2/PRATIUM ALPRO/belajarpypthon.py"
1
None
0
PS C:\Users\User>
```

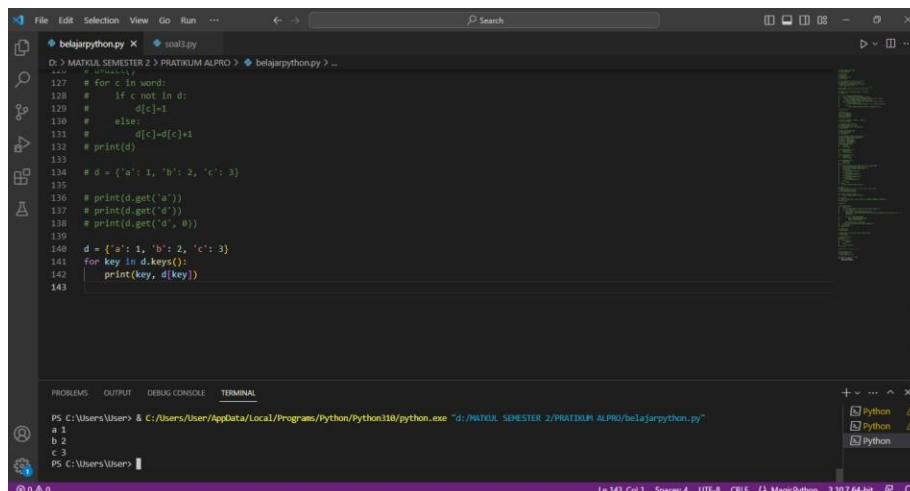
Menggunakan metode get

Pada kedua gambar di atas saat kunci a dicari, kedua cara (mengakses nilai langsung dan metode get) menghasilkan output yang sama. Namun saat kunci d dicari, cara mengakses nilai langsung menghasilkan error keyerror,s edangkan metode get mengembalikan nilai none (default jika kunci tidak ditemukan). Selain itu, dengan menggunakan argumen kedua pada metode get pengguna dapat menentukan nilai default yang akan dikembalikan jika kunci tidak ditemukan pada contoh di atas, jika kunci d tidak ditemukan, maka metode get akan mengembalikannya sebagai nilai 0.

MATERI 3

Looping dan Dictionary

Looping merupakan proses perulangan instruksi atau blok kode secara terus-menerus samapi kondisi yang ditentukan terpenuhi. Sedangkan dictionary adalah struktur data yang digunakan untuk menyimpan pasangan knci-nilai yang tidak berurut. Looping, dan dictionary dapat digunakan bersamaan dalam pemograman untuk melakukan iterasi pada setipa kunci atau nilai dalam sebuah dictionary. Contoh untuk melakukan looping pada dictionary:



```
136 # print(d.get('a'))
137 # print(d.get('d'))
138 # print(d.get('d', 0))
139
140 d = {'a': 1, 'b': 2, 'c': 3}
141
142 for key in d.keys():
143     print(key, d[key])
```

Terminal Output:

```
PS C:\Users\User> & C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe "d:/MATKUL SEMESTER 2/PRATIUM ALPRO/belajarpypthon.py"
a 1
b 2
c 3
PS C:\Users\User>
```

Pada program di atas adalah contoh penggunaan looping pada dictionary. Sebuah dictionary didefinisikan dengan nama `d` yang berisi tiga pasang kunci-nilai. kunci `a` memiliki nilai 1, kunci `b` memiliki nilai 2, dan kunci `c` memiliki nilai 3. Lalu program menggunakan loop `for` untuk mengiterasi setiap kunci dalam dictionary `d`. pada setiap iterasi, program mencetak kunci yang sedang diproses dan nilai yang terkait dengan kunci tersebut menggunakan format string yaitu `print(key,d[key])`. Dalam hal ini program memperoleh nilai terkait dengan kunci menggunakan sintaks `d[key]`.