Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	**<71231051>**
Nama Lengkap	**<Amida A Ronsumbre>**
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
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 1

Penjelasan materi 1

- 10.3 Materi

Ringkasan Dictionary Python

Dictionary adalah tipe data yang digunakan untuk menyimpan data dalam bentuk pasangan kunci-nilai (key-value pairs). Setiap kunci harus unik dan tidak boleh ada dua kunci yang sama dalam satu dictionary. Nilai yang terkait dengan kunci dapat berupa tipe data apa pun.

Dictionary dapat dibuat dengan menggunakan kurung kurawal ({}).

- Mudah digunakan untuk menyimpan data dalam bentuk pasangan kunci-nilai.
- Memungkinkan akses data yang cepat dan efisien menggunakan kunci.
- Dapat menyimpan berbagai jenis data, termasuk string, angka, dan list.
- Fleksibel dan dapat digunakan untuk berbagai aplikasi.

- Urutan item tidak dijamin.
- Kunci harus unik, yang dapat menyulitkan jika ada data yang memiliki kunci yang sama.

Dictionary adalah tipe data yang sangat berguna dan serbaguna di Python. Mudah digunakan, efisien, dan dapat digunakan untuk berbagai aplikasi.

- 10.3.1 Dictionary sebagai set penghitung (counters)

Implementasi histogram dalam Python dengan menggunakan metode .get() untuk menyederhanakan loop penghitungan dapat direpresentasikan dalam bentuk berikut:

latihan.py > ...

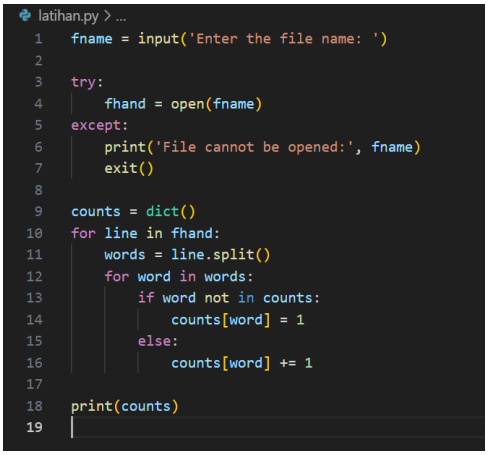
```
1  word = 'brontosaurus'
2  d = dict()
3  for c in word:
4      d[c] = d.get(c, 0) + 1
5  print(d)
6
```

Dalam implementasi ini, metode .get() digunakan untuk mengambil nilai yang terkait dengan kunci c. Jika c tidak ada dalam dictionary d, .get() akan mengembalikan nilai default yaitu 0. Dengan demikian, kita bisa menambahkan 1 ke nilai default ini, sesuai dengan frekuensi munculnya karakter c.

- 10.3.2 Dictionary dan File

Program Python di atas membaca file teks dan menghitung jumlah kemunculan setiap kata dalam file tersebut menggunakan dictionary. Setiap kata dipisahkan dari setiap baris, kemudian dictionary digunakan untuk menghitung kemunculan kata-kata. Hasilnya adalah dictionary yang berisi kata-kata sebagai kunci dan jumlah kemunculannya sebagai nilai.

Contohnya:



Dengan kode di atas, Anda dapat memasukkan nama file teks yang ingin Anda analisis kemunculan kata-katanya. Program akan membaca file tersebut, menghitung jumlah kemunculan setiap kata, dan menampilkan hasilnya dalam bentuk dictionary.

- 10.3.3 Looping dan Dictionary

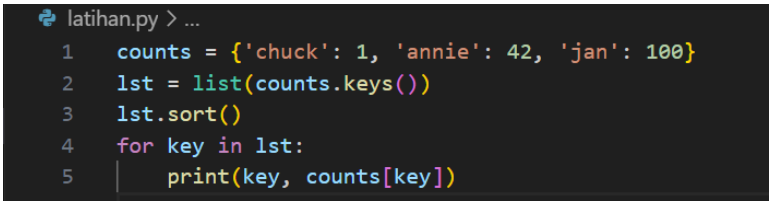
Pernyataan ini menjelaskan cara menggunakan looping for untuk menelusuri kunci-kunci dalam dictionary dan mencetak pasangan kunci-nilai. Terdapat tiga contoh yang diberikan:

1. Menelusuri dan mencetak semua kunci dan nilai dalam dictionary.

2. Menelusuri dan mencetak kunci-kunci yang memiliki nilai di atas 10.

3. Mencetak kunci-kunci dalam urutan alfabet.

Kesimpulannya, looping for digunakan untuk menelusuri kunci-kunci dalam dictionary, dan dengan menggunakan kunci, kita dapat mengakses nilainya. Selanjutnya, contoh ketiga menunjukkan bagaimana mengurutkan kunci-kunci sebelum mencetaknya.Contohnya:



untuk menggunakan pendekatan tertentu tergantung pada kebutuhan Anda:Jika Anda hanya perlu menelusuri seluruh dictionary, Anda dapat menggunakan looping langsung dengan for key in counts.

Jika Anda perlu melakukan pengecekan pada nilai sebelum mencetak, seperti dalam contoh kedua, Anda dapat menambahkan pernyataan kondisional dalam looping.

Jika Anda ingin mencetak kunci-kunci dalam urutan tertentu, Anda dapat membuat list kunci, mengurutkannya, dan kemudian melakukan looping melalui list tersebut.

Setiap pendekatan memiliki kegunaannya tergantung pada situasi yang spesifik. Misalnya, jika urutan tidak penting, pendekatan pertama paling sederhana dan paling langsung. Sedangkan, jika Anda perlu mengurutkan output, pendekatan ketiga akan lebih sesuai

- 10.3.4 Advenced Text Parsing

Program tersebut membersihkan teks dari tanda baca dan mengubah semua huruf menjadi huruf kecil sebelum menghitung jumlah kemunculan setiap kata dalam file. Ini dilakukan dengan menggunakan metode `translate()` untuk menghapus tanda baca dan `lower()` untuk mengubah huruf menjadi huruf kecil. Setelah itu, program menggunakan dictionary untuk menghitung jumlah kemunculan setiap kata dalam teks.contohnya:

untuk membersihkan teks dari tanda baca dan mengubah semua huruf menjadi huruf kecil sebelum menghitung kemunculan kata adalah agar program dapat menganggap kata-kata yang sama tetapi ditulis dengan huruf besar atau kecil atau memiliki tanda baca yang berbeda sebagai kata yang sama. Hal ini membantu dalam melakukan analisis yang lebih akurat terhadap teks.

MATERI 2

Penjelasan materi 2,

Kegiatan Pratikum:

- Kasus 10.1 Berikut adalah program yang menghasilkan dan mencetak dictionary yang berisi angka dari 1 sampai n beserta kuadratnya:

```python
n = int(input("Input data = "))
dictionary = {}

for x in range(1, n + 1):
    dictionary[x] = x * x

print("Dictionary =", dictionary)

```

Program ini meminta pengguna untuk memasukkan nilai n, kemudian membuat dictionary yang berisi angka dari 1 sampai n beserta kuadratnya. Setiap angka dari 1 hingga n dimasukkan sebagai kunci, dan kuadrat dari angka tersebut dimasukkan sebagai nilai. Setelah itu, dictionary tersebut dicetak.

- Kasus 10.2

Berikut adalah program untuk mencetak semua nilai (value) unik yang ada dalam list of dictionaries:

```python
data = [
    {"V": "S001"}, {"V": "S002"}, {"VI": "S001"}, {"VI": "S005"},
    {"VII": "S005"}, {"V": "S009"}, {"VIII": "S007"}
]

print("Data asli:", data)

unique_values = set(val for dic in data for val in dic.values())

print("Nilai Unik:", unique_values)

```

Program ini menginisialisasi list data yang berisi beberapa dictionaries. Kemudian, menggunakan nested comprehension, program mengumpulkan semua nilai unik dari dictionaries dalam list tersebut ke dalam sebuah set menggunakan ekspresi set(val for dic in data for val in dic.values()). Set digunakan karena hanya nilai unik yang diinginkan. Akhirnya, program mencetak set nilai unik tersebut.

- Kasus 10.3

Berikut adalah program yang membaca file words.txt, menyimpan kunci (keys) dalam sebuah dictionary, dan melakukan pengecekan apakah suatu kata yang diinputkan ada dalam daftar tersebut:

```python
count = 0
dictionary_words = dict()
fname = input('Masukkan nama file : ')
fword = input('Kata yang dicari : ')

try:
    fhand = open(fname)
except FileNotFoundError:
    print('File tidak bisa dibuka !!', fname)
    exit()

for line in fhand:
    words = line.split()
    for word in words:
        count += 1
        if word in dictionary_words:
            continue  # cek duplikasi
        dictionary_words[word] = count  # kata yg pertama muncul

print('\nDaftar Kamus : \n')
print(dictionary_words)

if fword in dictionary_words:
    print('\nKata %s ditemukan dalam kamus' % fword)
else:
    print('\nKata %s tidak ditemukan dalam kamus' % fword)

```

Program ini membaca file words.txt, kemudian untuk setiap kata dalam file tersebut, program memasukkan kata tersebut ke dalam dictionary dictionary_words, dengan key merupakan kata dan value merupakan nomor urut kemunculan kata tersebut. Setelah selesai membaca file, program mencetak dictionary tersebut dan melakukan pengecekan apakah kata yang diinputkan ada dalam dictionary.

- Kasus 10.4

Berikut adalah program yang membaca file mbox-short.txt, membagi kategori dari setiap email berdasarkan hari perintah commit dilakukan, dan menghitung berapa banyak hari di mana perintah commit tersebut sering dilakukan:

latihan.py > ...

```python
dictionary_days = dict()  # dictionary baru
fname = input('Enter a file name: ')

try:
    fhand = open(fname)
except FileNotFoundError:
    print('File cannot be opened:', fname)
    exit()

for line in fhand:
    words = line.split()
    if len(words) < 3 or words[0] != 'From':
        continue
    else:
        if words[2] not in dictionary_days:
            dictionary_days[words[2]] = 1  # pertama
        else:
            dictionary_days[words[2]] += 1  # +1

print(dictionary_days)
```

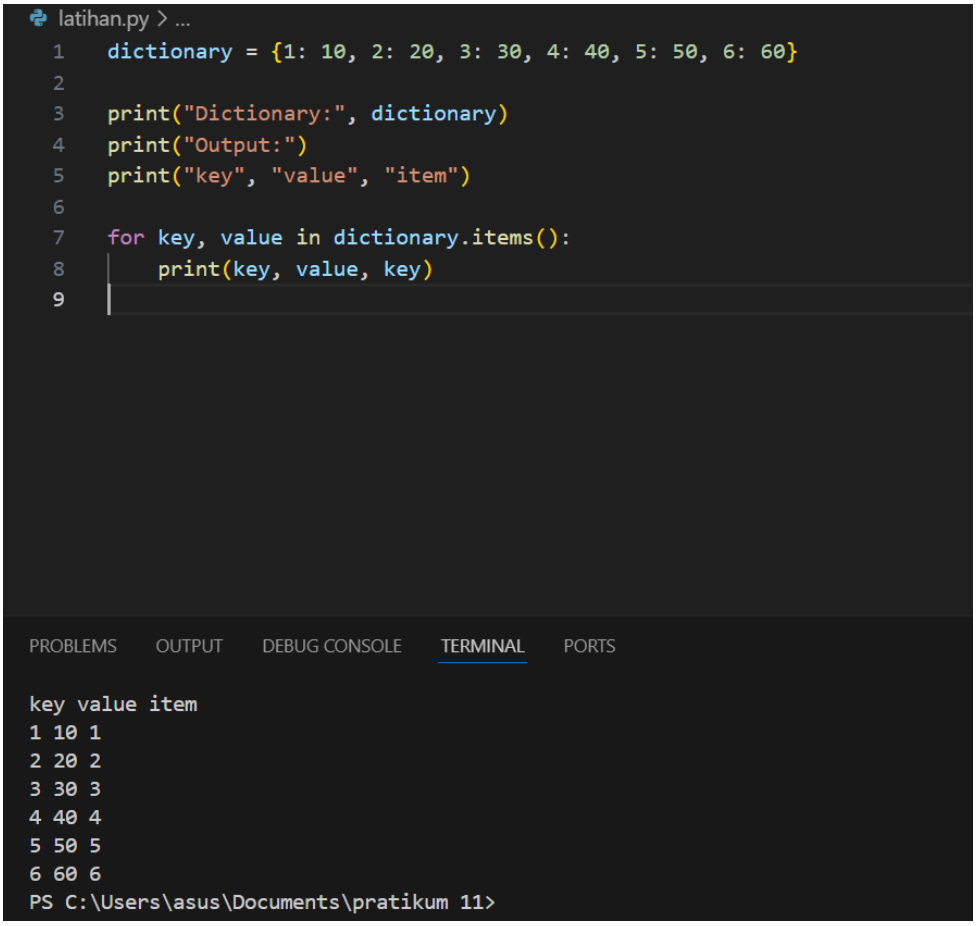
Program ini membaca file mbox-short.txt dan mencari baris yang dimulai dengan "From". Kemudian, program membagi kategori dari setiap email berdasarkan hari perintah commit dilakukan (kata ketiga pada setiap baris), dan menghitung berapa banyak hari di mana perintah commit tersebut sering dilakukan. Hasilnya dicetak dalam bentuk dictionary yang menunjukkan jumlah perintah commit yang dilakukan pada setiap hari.

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

Tulis jawaban anda untuk soal nomor 1 di sini



Tentu! Program tersebut menghasilkan output yang sesuai dengan kebutuhan karena:

Memahami Struktur Dictionary: Program memahami struktur dictionary dengan memasukkan pasangan key-value yang diberikan ke dalam sebuah dictionary.

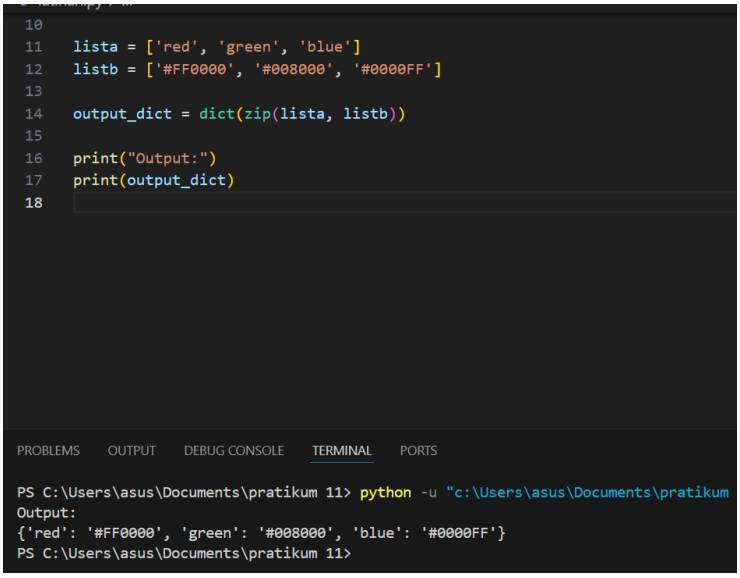
Menggunakan Metode items(): Dengan menggunakan metode items() pada dictionary, program dapat mengakses setiap pasangan key-value dalam dictionary secara berurutan.

Mencetak Nilai Key, Value, dan Item: Program mencetak nilai key, value, dan item dari setiap pasangan key-value dalam dictionary. Nilai key dan value langsung diakses menggunakan variabel dalam loop, sementara nilai item diambil dari key itu sendiri karena pada contoh yang diberikan, item diinginkan sesuai dengan nilai key.

Dengan demikian, program tersebut menghasilkan output yang sesuai dengan format yang diminta, yaitu mencetak nilai key, value, dan item dari setiap pasangan key-value dalam dictionary.

SOAL 2

Tulis jawaban anda untuk soal nomor 2 di sini.



Dua list lista dan listb berisi elemen-elemen yang ingin dipetakan ke dalam dictionary.Fungsi zip() digunakan untuk menggabungkan elemen-elemen kedua list ke dalam pasangan-pasangan yang sesuai. Setiap elemen pertama dari lista akan menjadi kunci, dan setiap elemen pertama dari listb akan menjadi nilai dalam dictionary.Hasil dari zip() kemudian dikonversi menjadi dictionary menggunakan fungsi dict().

Hasil dictionary kemudian dicetak.Output dari program ini akan sesuai dengan contoh yang diberikan, yaitu sebuah dictionary yang memetakan elemen-elemen dari lista sebagai kunci dan elemen-elemen dari listb sebagai nilai.

SOAL 3

Tulis jawaban anda untuk soal nomor 3 di sini.

latihan.py > ...

```python
fname = input("Masukkan nama file: ")
try:
    fhand = open(fname)
except FileNotFoundError:
    print('File tidak dapat dibuka:', fname)
    exit()

email_histogram = {}

for line in fhand:
    if line.startswith('From '):
        words = line.split()
        email = words[1]
        email_histogram[email] = email_histogram.get(email, 0) + 1

print(email_histogram)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\asus\Documents\pratikum 11> python -u "c:\Users\asus\Documents\pratiku
Masukkan nama file: mbox,short
{'stephen.marquard@uct.ac.za': 1}
PS C:\Users\asus\Documents\pratikum 11>
```

Program ini membaca setiap baris dalam file "mbox-short.txt". Jika baris dimulai dengan "From ", yang menunjukkan awal sebuah email, program akan memisahkan baris tersebut menjadi kata-kata dan mengambil alamat email pengirimnya. Kemudian, program akan memperbarui dictionary email_count, dengan menggunakan alamat email sebagai kunci dan meningkatkan jumlah pesan masuk untuk setiap alamat email yang ditemukan.Contoh output dari program ini akan menampilkan histogram jumlah pesan yang masuk dari setiap alamat email yang terdapat dalam file "mbox-short.txt", sesuai dengan contoh output yang Anda berikan.

SOAL 4

Tulis jawaban anda untuk soal nomor 4 di sini.

Program ini membaca setiap baris dalam file "mbox-short.txt". Jika baris dimulai dengan "From ", yang menunjukkan awal sebuah email, program akan memisahkan baris tersebut menjadi kata-kata dan mengambil alamat email pengirimnya. Kemudian, program akan memisahkan domain dari alamat email menggunakan "@" dan menghitung jumlah pesan yang dikirim oleh masing-masing domain.Contoh output dari program ini akan menampilkan dictionary yang berisi nama domain pengirim pesan beserta jumlah pesan yang dikirim oleh masing-masing domain, sesuai dengan contoh output yang Anda berikan.