Laporan Praktikum Algoritma & Pemrograman

Semester Genap 2024/2025

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.

NIM	**<71241079>**
Nama Lengkap	**<Natalie Neysa Jessica Soesanto>**
Minggu ke / Materi	**12 / Dictionary**

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2025

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

Dictionary merupakan kumpulan kunci dan nilai yang saling berhubungan dimana setiap kunci tersebut mengaitkan dengan nilai tertentu. Setiap pasangan kunci:nilai disebut sebagai item. Sebagai contoh, kitab isa membuat dictionary yang memetakan kata-kata dari bahasa Inggris ke bahasa Spanyol dimana kata-kata tersebut menjadi kunci dan nilai dalam bentuk string. Dengan asumsi bahwa setiap kata dalam bahasa Inggris mempunyai satu arti dalam bahasa Spanyol. Dictionary mirip dengan list namun lebih fleksibel. Setiap item dalam dictionary terdiri dari kunci dan nilainya, dan setiap kunci harus unik. Ini memungkinkan pemetaan antara Kumpulan kunci dan nilai.

Fungsi dict digunakan untuk membuat dictionary baru. Penggunaannya perlu dihindari sebagai nama variabel karena dict adalah built-in function dari python.

Source Code:

```
1   eng2sp = {'one': 'uno', 'two': 'dos', 'three': 'tres'}
2   print(eng2sp)
```

Output:

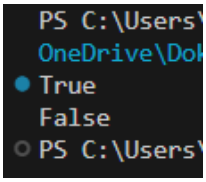
```
PS C:\Users\ASUS\OneDrive\Dokumen\Skull\Sem 2\Pr
OneDrive\Dokumen\Skull\Sem 2\PrakAlpro\Dict\lapr
{'one': 'uno', 'two': 'dos', 'three': 'tres'}
PS C:\Users\ASUS\OneDrive\Dokumen\Skull\Sem 2\Pr
```

Operator in dalam dictionary mengembalikan nilai true dan false sesuai dengan kunci yang ada pada dictionary. Pada list, operator in menggunakan algoritma pencarian linear. Yang artinya waktu pencarian bertambah seiring bertambahnya panjang list. Namun dalam dictionary, python menggunakan algoritma Hash Table yang memungkinkan operator in untuk memproses dengan cepat tanpa memperdulikan jumlah item di dalam dictionary.

Source Code:

```
1   eng2sp = {'one': 'uno', 'two': 'dos', 'three': 'tres'}
2   print ('one' in eng2sp)
3   print('uno' in eng2sp)
```

Output:



MATERI 2

Dictionary sebagai set penghitung (counters)

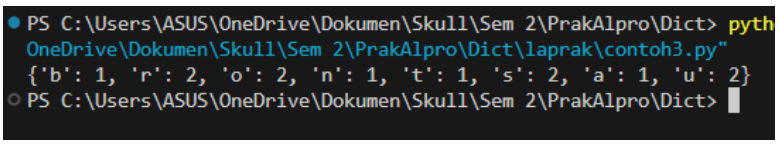
Penggunaan model dictionary dalam perhitungan dianggap lebih praktis tidak perlu diketahui karakter apa yang akan muncul.

Contoh perhitungan yang dapat dilakukan dengan model dictionary :

Source Code:

```
word = 'brontosaurus'
d = dict()
for c in word:
    if c not in d:
        d[c] = 1
    else:
        d[c] = d[c] + 1
print(d)
```

Output:



Terdapat beberapa pendekatan yang dapat digunakan untuk menghitung jumlah kemunculan huruf dalam sebuah string :

1. Pembuatan 26 variabel untuk setiap huruf dalam alfabet kemudian setiap karakter dalam string dimasukkan ke dalam variabel yang sesuai dan dilakukan penambahan perhitungan yang sesuai dengan setiap karakter. Pendekatan ini menggunakan kondisional berantai untuk megelola setiap karakter.
2. Pembuatan list dengan 26 elemen dimana setiap karakter dalam string dikonversi menjadi angka menggunakan fungsi bawaan. Kemudian angka tersebut digunakan sebagai indeks dalam list untuk menambahkan perhitungan yang sesuai.
3. Pembuatan sebuah dictionary dengan karakter sebagai kunci dan jumlah kemunculannya sebagai nilai yang sesuai. Setiap karakter dalam string ditambhakn

sebagai item ke dalam dictionary, dan nilai dari setiap item ditambahkan sesuai dengan kemunculan karakter tersebut.

Dictionary memiliki metode get. Metode ini berguna untuk mengambil nilai dari kunci yang spesifik dan dapat mengembalikan nilai default jika kunci tidak ada dalam dictionary. Jika kuncu ditemukan, metode get akan mengembalikan nilai yang terkait dengan kunci tersebut. Namun, jika kunci tidak ditemukan, maka akan mengembalikan nilai default yang telah ditentukan sebelumnya.

Source Code:

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

Output:

```
PS C:\Users\ASUS\OneDrive\Dokumen\Skull\Sem 2\PrakAlpro\Dict> pyth
OneDrive\Dokumen\Skull\Sem 2\PrakAlpro\Dict\laprak\contoh4.py"
{'b': 1, 'r': 2, 'o': 2, 'n': 1, 't': 1, 's': 2, 'a': 1, 'u': 2}
PS C:\Users\ASUS\OneDrive\Dokumen\Skull\Sem 2\PrakAlpro\Dict>
```

MATERI 3

Dictionary dan File

Salah satu penggunaan umum dari dictionary adalah untuk menghitung frekuensi kemunculan kata-kata dalam sebuah file teks. Misalnya, kita menggunakan file yang berisi teks yang disederhanakan dari Romeo daan Juliet tanpa tanda baca. Sebagai contoh awal, kita akan membaca setiap baris dari file tersebut lalu memecahkan menjadi daftar kata-kata dan mengiterasi melalui setiap kata dalam setiap baris. Dalam proses ini, kita akan menggunakan dictionary untuk menghitung jumlah kemunculan setiap kata.

Kita asumsikan, kita menggunakan dua perulangan for dimana perulangan luar untuk membaca setiap baris dalam file dan perulangan dalam untuk melakukan iterasi melalui setiap kata yang ada di baris tertentu. Itu merupakan contoh dari nested loop yaitu satu perulangan berada di dalam yang lain.

Untuk setiap iterasi dari perulangan luar, perulangan dalam akan dieksekusi. Dalam hal ini, perulangan dalam akan berjalan lebih cepat daripada perulangan luar. Setiap kata yang ada di setiap baris file akan dihitung dengan benar karena kombinasi dari kedua peristiwa ini.

Source Code:

```python
fname = input('Enter the file name: ')
try:
    fhand = open(fname)
except:
    print('File cannot be opened:', fname)
    exit()

counts = dict()
for line in fhand:
    words = line.split()
    for word in words:
        if word not in counts:
            counts[word] = 1
        else:
            counts[word] += 1

print(counts)
```

Output:

```
PS C:\Users\ASUS\OneDrive\Dokumen\Skull\Sem 2\PrakAlpro\Dict> python -u "c:\Users\ASUS\OneDrive\Dokumen\Skull\Sem 2\PrakAlpro\Dict\laprak\contoh5.py"
Enter the file name: laprak\romeo.txt
{'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}
PS C:\Users\ASUS\OneDrive\Dokumen\Skull\Sem 2\PrakAlpro\Dict>
```

Digunakan model penulisan yang lebih singkat pada statement else untuk menambahkan variabel. counts[word] += 1 sama dengan counts[word] = counts[word] + 1. Dapat digunakan metode lain juga untuk mengubah nilai suatu variabel dengan jumlah yang diinginkan, misalnya dengan -=, *=, dan /=. Akan didapatkan data dump jumlah semua dalam urutan hash ketika kita menjalankan program.

MATERI 4

Looping and Dictionary

Untuk mencetak kunci dalam urutan alfabet, langkah pertama adalah membuat daftar kunci di kamus dengan menggunakan metode kunci yang tersedia pada object dictionary. Langkah selanjutnya adalah melakukan pengurutan (sort), list, dan loop melalui sorted list. Langkah terakhir adalah memeriksa setiap kunci dan mencetak pasangan nilaai key yang telah diurutkan.

Source Code:

```python
counts = { 'chuck' : 1 , 'annie' : 42, 'jan': 100}
lst = list(counts.keys())
print(lst)
lst.sort()
for key in lst:
    print(key, counts[key])
```

Output:

```
PS C:\Users\ASUS\OneDrive\Do
ro\Dict\laprak\contoh6.py"
['chuck', 'annie', 'jan']
annie 42
chuck 1
jan 100
PS C:\Users\ASUS\OneDrive\Do
```

MATERI 5

Advenced Text Parsing

Fungsi split() pada python secara default membagi string berdasarkan spasi, menjadikan setiap kata sebagai token yang terpisah oleh spasi. Misalnya, kata ‘soft!” dan “soft” akan dianggap sebagai dua kata yang berbeda dan keduanya akan dhitung secara terpisah dalam dictionary.

Selain fungsi split(), kita juga dapat menggunakan metode string lain seperti lower(), punctuation, dan translate(). Metode string yang paling halus (paling tidak terlihat) adalah metode translate().

Dalam metode translate(), karakter pada fromstr diganti dengan karakter pada posisi yang sama pada tostr dan semua karakter yang ada dalam deletetr dihapus. Jika fromstr atau tostr kosong, karakter yang sesuai akan diabaikan. Kita tidak akan menentukan tostr dalam hal ini, tetapi kita akan menggunakan parameter deletetr untuk menghapus semua tanda baca. Python menggunakan karakter mana yang dianggap sebagai tanda baca akan diidentifikasi secara otomatis.

Source Code:

```python
import string

fname = input('Enter the file name: ')
try:
    fhand = open(fname)
except:
    print('File cannot be opened:', fname)
    exit()

counts = dict()
for line in fhand:
    line = line.rstrip()
    line = line.translate(line.maketrans('', '', string.punctuation))
    line = line.lower()
    words = line.split()
    for word in words:
        if word not in counts:
            counts[word] = 1
        else:
            counts[word] += 1

print(counts)
```

Output:

```
PS C:\Users\ASUS\OneDrive\Dokumen\Skull\Sem 2\PrakAlpro\Dict> python -u "c:\Users\ASUS\OneDrive\Dokumen\Skull\Sem 2\PrakAlp
ro\Dict\laprak\contoh7.py"
Enter the file name: laprak\romeo-full.txt
{'romeo': 40, 'and': 42, 'juliet': 32, 'act': 1, '2': 2, 'scene': 2, 'ii': 1, 'capulets': 1, 'orchard': 2, 'enter': 1, 'he'
: 5, 'jests': 1, 'at': 9, 'scars': 1, 'that': 30, 'never': 2, 'felt': 1, 'a': 24, 'wound': 1, 'appears': 1, 'above': 6, 'wi
ndow': 2, 'but': 18, 'soft': 1, 'what': 11, 'light': 5, 'through': 2, 'yonder': 2, 'breaks': 1, 'it': 22, 'is': 21, 'the':
34, 'east': 1, 'sun': 2, 'arise': 1, 'fair': 4, 'kill': 2, 'envious': 2, 'moon': 4, 'who': 5, 'already': 1, 'sick': 2, 'pal
e': 1, 'with': 8, 'grief': 2, 'thou': 32, 'her': 14, 'maid': 2, 'art': 7, 'far': 2, 'more': 9, 'than': 6, 'she': 9, 'be': 1
4, 'not': 18, 'since': 1, 'vestal': 1, 'livery': 1, 'green': 1, 'none': 1, 'fools': 1, 'do': 7, 'wear': 1, 'cast': 1, 'off'
: 1, 'my': 29, 'lady': 2, 'o': 11, 'love': 24, 'knew': 1, 'were': 9, 'speaks': 3, 'yet': 9, 'says': 1, 'nothing': 1, 'of':
20, 'eye': 2, 'discourses': 1, 'i': 61, 'will': 8, 'answer': 1, 'am': 7, 'too': 8, 'bold': 1, 'tis': 4, 'to': 34, 'me': 18,
 'two': 1, 'fairest': 1, 'stars': 2, 'in': 13, 'all': 6, 'heaven': 3, 'having': 1, 'some': 3, 'business': 1, 'entreat': 1,
'eyes': 6, 'twinkle': 1, 'their': 6, 'spheres': 1, 'till': 4, 'they': 5, 'return': 1, 'if': 12, 'there': 3, 'head': 2, 'bri
ghtness': 1, 'cheek': 4, 'would': 16, 'shame': 1, 'those': 2, 'as': 12, 'daylight': 1, 'doth': 2, 'lamp': 1, 'airy': 2, 're
gion': 1, 'stream': 1, 'so': 10, 'bright': 2, 'birds': 1, 'sing': 1, 'think': 2, 'night': 16, 'see': 2, 'how': 5, 'leans':
1, 'upon': 5, 'hand': 4, 'glove': 1, 'might': 1, 'touch': 1, 'ay': 2, 'speak': 4, 'again': 6, 'angel': 1, 'for': 12, 'glori
ous': 1, 'this': 9, 'being': 2, 'oer': 1, 'winged': 1, 'messenger': 1, 'unto': 1, 'whiteupturned': 1, 'wondering': 1, 'mort
als': 1, 'fall': 1, 'back': 4, 'gaze': 1, 'on': 4, 'him': 2, 'when': 2, 'bestrides': 1, 'lazypacing': 1, 'clouds': 1, 'sail
s': 1, 'bosom': 1, 'air': 1, 'wherefore': 2, 'deny': 2, 'thy': 20, 'father': 1, 'refuse': 1, 'name': 11, 'or': 4, 'wilt': 6
, 'sworn': 1, 'ill': 8, 'no': 6, 'longer': 1, 'capulet': 1, 'aside': 1, 'shall': 6, 'hear': 2, 'enemy': 2, 'thyself': 1, 't
hough': 1, 'montague': 5, 'whats': 2, 'nor': 5, 'foot': 2, 'arm': 1, 'face': 2, 'any': 4, 'other': 4, 'part': 2, 'belonging
': 1, 'man': 2, 'which': 7, 'we': 2, 'call': 3, 'rose': 1, 'by': 14, 'smell': 1, 'sweet': 8, 'calld': 1, 'retain': 1, 'dear
': 7, 'perfection': 1, 'owes': 1, 'without': 1, 'title': 1, 'doff': 1, 'thee': 24, 'take': 3, 'myself': 2, 'word': 4, 'new'
```

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

Source Code:

```
dictio_nilai = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
nomor = 1
print('key', 'value', 'item')

for key, values in dictio_nilai.items():
    print(key, ' ', values, '  ', nomor)
    nomor += 1
```

Output:

```
PS C:\Users\ASUS\OneDrive\
ro\Dict\laprak\soal1.py"
key value item
1   10    1
2   20    2
3   30    3
4   40    4
5   50    5
6   60    6
PS C:\Users\ASUS\OneDrive\
```

Penjelasan:

Pada program ini terdapat fungsi diction_nilai yang berisi sejumlah pasangan kunci:nilai. Selanjutnya variabel nomor diinisialisasi dengan 1. Program menggunakan pernyataan for untuk mengiterasi pasangan kunci:nilai dalam dictio_nilai pada setiap iterasi. Pada setiap iterasi, kunci, nilai, dan nomor urut pasangan kunci:nilai dicetak. Untuk mengatur nomor urut pasangan selanjutnya, variabel nomor ditingkatkan satu setelah mencetak. Hal ini memungkinkan kita untuk melihat bagaimana pasangan kunci:nilai dalam dictionary diurutkan secara relatif.

SOAL 2

Source Code:

```
list_data_a = ['red', 'green', 'blue']
list_data_b = ['#FF0000','#008000','#0000FF']
dictio = dict()
for key, value in zip(list_data_a, list_data_b):
    dictio[key] = value
print(dictio)
```

Output:

```
PS C:\Users\ASUS\OneDrive\Dokumen\Skull\Sem 2\PrakAlpro\Dic
ro\Dict\laprak\soal2.py"
{'red': '#FF0000', 'green': '#008000', 'blue': '#0000FF'}
PS C:\Users\ASUS\OneDrive\Dokumen\Skull\Sem 2\PrakAlpro\Dic
```

Penjelasan:

Program ini menggunakan fungsi zip() untuk menghubungkan dua daftar, satu sebagai kunci dan yang lainnya sebagai nilai. Dictionary “dictio” mencakup semua pasangan kunci:nilai, yang berarti bahwa kita dapat langsung mengakses nilai suatu kunci dalam dictionary jika kita ingin tahu nilainya. Proses ini memungkinkan pengaturan data yang terstruktur. Hubungan kunci:nilai yang diwakili oleh pasangan list_data_a dan list_data_b dalam situasi ini adalah contoh hubungan konsep yang lebih kompleks atau tersebar luas.

SOAL 3

Source Code:

```python
def baca_log(filename):
    daftar_email = {}
    total_pesan = 0

    with open(filename, 'r') as file:
        for line in file:
            if line.startswith('From '):
                email = line.split()[1]
                daftar_email[email] = daftar_email.get(email, 0) + 1
                total_pesan += 1

    return daftar_email, total_pesan

nama_file = input("Masukkan nama file: ")

email_, pesan = baca_log(nama_file)
print(email_)
```

Output:

```
PS C:\Users\ASUS\OneDrive\Dokumen\Skull\Sem 2\PrakAlpro\Dict> python -u "c:\Users\ASUS\OneDrive\Dokumen\Skull\Sem 2\PrakAlp
ro\Dict\laprak\soal3.py"
Masukkan nama file: laprak\mbox-short.txt
{'stephen.marquard@uct.ac.za': 2, 'louis@media.berkeley.edu': 3, 'zqian@umich.edu': 4, 'rjlowe@iupui.edu': 2, 'cwen@iupui.e
du': 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\ASUS\OneDrive\Dokumen\Skull\Sem 2\PrakAlpro\Dict>
```

Penjelasan:

Pada program ini, kita memiliki fungsi baca_log yang membaca file log dan menghitung jumlah pesan dari setiap alamat email yang ditemukan. Setiap alamat email dan jumlahnya disimpan dalam sebuah dictionary. Setelah membaca file, fungsi mengembalikan dictionary tersebut bersama dengan total pesan yang telah dihitung. Kemudian, pengguna diminta untuk memasukkan nama file, dan fungsi baca_log dipanggil dengan nama file tersebut. Hasilnya, dictionary yang berisi alamat email dan jumlahnya dicetak.

SOAL 4

Source Code:

```python
def hitung_domain(email):
    return email.split('@')[-1]

def hitung_domain_email(filename):
    counts = {}

    with open(filename, 'r') as file:
        for line in file:
            if line.startswith('From '):
                email = line.split()[1]
                domain = hitung_domain(email)
                counts[domain] = counts.get(domain, 0) + 1
    return counts

nama_file = input("Masukkan nama file: ")
counts = hitung_domain_email(nama_file)

print(counts)
```

Output:

```
PS C:\Users\ASUS\OneDrive\Dokumen\Skull\Sem 2\PrakAlpro\Dict> python -u "c:\Users\ASUS\OneDrive\Dokumen\Skull\Sem 2\PrakAlpro\Dict\laprak\soal4.py"
Masukkan nama file: laprak\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}
PS C:\Users\ASUS\OneDrive\Dokumen\Skull\Sem 2\PrakAlpro\Dict>
```

Penjelasan:

Fungsi hitung_domain dalam program ini mengambil dan mengembalikan domain dari alamat email. Fungsi hitung_domain_email kemudian membaca file log dan menghitung jumlah kemunculan domain dari alamat email yang ditemukan. Hasilnya, kamus yang berisi jumlah kali domain muncul dicetak.

Link Github: https://github.com/babydoll-05/Laprak-12.git