

Lab 4 Dasar-Dasar Pemrograman 1

File and Exception

DEADLINE : 29 September 2017 pukul 15:30 waktu Scele

Selamat datang di Lab/Tutorial Dasar-Dasar Pemrograman 1!

Di Lab 4 ini kalian akan belajar tentang salah satu bagian penting dari pemrograman, yaitu

File and Exception

Oleh karena itu semangatnya Jangan Kasih Kendor !

I/O Files

- READ/WRITE FILES

Untuk bisa membuka / membuat files, bisa menggunakan fungsi :

```
<file_object> = open(<filename>,<op>)
```

Ket :

- **filename** merupakan file yang akan kita buka / buat, dan harus beserta formatnya ("input.txt", "diary.txt")
- **op** merupakan jenis operasi yang akan dilakukan pada method open tsb, ini yang menentukan apakah kita ingin membuka (**read**) atau membuat (**write**) berkas
- **filename** dan **op** berupa string
- File_object merupakan variable yang akan menyimpan file yang akan menyimpan file yang telah dibuka

Adapun beberapa jenis operasi yang harus diketahui antara lain :

r	Membaca file, akan error jika file yang ditentukan tidak ada.
----------	---

w	Menulis file, akan menimpa/overwrite file yang ditentukan.
a	Menulis file, akan menambahkan/merge/append file yang ditentukan.

Untuk menambahkan kalimat ke dalam file yang akan ditulis, bisa menggunakan fungsi :

print (<kalimat>, file = <file_object>)
<file_name> akan diisikan sebagai nama file tujuan yang akan kita isi/tulis. Pastikan file tersebut bersifat write .

PERHATIAN : Setiap file yang dibuka, harus ditutup agar bisa tersimpan dan tidak error (baik file yang di read maupun write) menggunakan method

<file_object>.close() # gunakan saat file sudah selesai dipakai / tidak dipakai lagi

Contoh penggunaan:

Transfer isi antar file

```
berkas_masukan = open("fileInput.txt", "r")
berkas_keluaran = open("fileOutput.txt", "w")

for kalimat in berkas_masukan:
    print(kalimat, file = berkas_keluaran)

berkas_masukan.close()
berkas_keluaran.close()
```

String Splitting with Regular Expression

Motivasi

String splitting dengan method str.split() hanya bisa menerima input berupa string biasa sebagai pemisahannya. Contoh:

```
>>> word = 'aku suka ddp'
```

```
>>> word.split('suka')
['aku ', ' ddp']
```

Problem

Bagaimana jika stringnya berupa:

```
>>> word = 'hari ini aku sedang-sangat,suka/sekali
kuliah;ddp'
```

Dan kita tidak butuh pengganggu seperti

```
, - / ;
```

Looping manual? Bisa saja. Tapi ada cara yang lebih sakti.

```
>>> import re #regular expression
>>> word = 'hari ini aku sedang-sangat,suka/sekali
kuliah;ddp'
>>> expression = ',+|-+|;+|/+|\s'
>>> re.split(expression, word)
['hari', 'ini', 'aku', 'sedang', 'sangat', 'suka',
'sekali', 'kuliah', 'ddp']
```

Kita cukup mengabaikan sampah-sampah yang ada di list. Sintaks (|) dalam expression berarti OR. Masih banyak sintaks lainnya dalam regular expression, silakan cek <https://regex101.com/>

Contoh lain penggunaan regular expression.

```
>>> import re #regular expression
>>> word = '<a>universitas<b>fasilkom<a>indonesia<b>'
>>> re.split('<b>.*<a>', word)
['<a>universitas', 'indonesia<b>']
```

Secret Agent 069 Benny

"Kriptografi caesar cipher terlalu sulit untuk diimplementasikan, aku pilih cara yang lebih mudah saja" -Agen Benny

Agen rahasia Benny menyembunyikan beberapa informasi rahasia di dalam suatu file yang tampak normal. Rahasia tersebut berada diantara sebuah tag <start> dan sebuah tag <end>. Karena Anda tidak suka Benny, maka Anda akan membuat program yang **mengekstrak informasi yang dirahasiakan** oleh Benny untuk kemudian dibocorkan ke pihak yang melawan Benny. Agen Lesa, musuh Benny, akan membantu Anda.

Agen Lesa memberikan Anda sebuah file berupa transkrip komunikasi agen Benny, program kalian harus bisa memindahkan semua string yang berada di antara tag <start> dan tag <end> ke sebuah file lain. Dijamin bahwa setiap tag <start>, tag yang selanjutnya akan muncul adalah tag <end>, jadi tidak akan ada semacam ...<start>...<start>...<end>...<end>...

Contoh input stdin:

Masukkan nama file input dan output: **input101.txt output66.txt**

Contoh output stdout:

Rahasia telah terbongkar, silakan cek file **output66.txt**

Contoh isi file input:

```
Selamat siang pak, saya ingin menginformasikan <start>Target
<end>Pengumpulan Dana sebesar<start>550<end>0000 sudah tercapai!
Uangnya sebagian besar<start> sudah di <end> tangan saya, sekarang
kita hanya perlu menentukan <start>lokasi<end> acara. Terima Kasih
```

Contoh isi file output:

```
Target 550 sudah di lokasi
```

BONUS! (yeay ada bonus)

Agen Lesa sebagai manusia tidak luput dari salah, jika nama file yang diberikan oleh agen Lesa salah, sehingga bisa jadi **tidak ada** atau terdapat **masalah lainnya**, **batal**kan eksekusi, dan laporkan pada standard output.

Contoh output stdout:

```
File input65536.txt bermasalah! Benny lolos kali ini.
```