

Dasar-Dasar Pemrograman: Lab 5 Kelas A

Strings, Slicing and Files

Jumat, 5 Oktober 2018



FAKULTAS
ILMU
KOMPUTER

String

String merupakan salah satu tipe data yang dapat merepresentasikan data sebagai susunan atau urutan karakter-karakter yang pada umumnya kita sebut sebagai kata atau kalimat. Penulisan *string* sedikit berbeda dari penulisan data bertipe *int* atau *float*. *String* di Python boleh ditulis menggunakan tanda kutip `'` atau pun tanda kutip dua `"`. Kutip pembuka suatu *string* harus sama dengan kutip penutupnya. Berikut adalah contoh penulisan data *string* pada Python Shell.

```
>>> name = "Dek Depy" # string dengan kutip dua "  
>>> name  
'Dek Depy'  
>>> nama = 'Dedek Peye' # string dengan kutip satu '  
>>> nama  
'Dedek Peye'  
>>> "23"  
'23'  
>>> "23" == 23  
False  
>>> 'lalala"  
SyntaxError: EOL while scanning string literal  
>>>
```

Beberapa karakter khusus, misalnya tanda kutip, harus ditulis dengan cara khusus agar bisa tetap dianggap sebagai karakter (sehingga tidak dianggap sebagai tanda khusus). Kita menggunakan *backslash* pada karakter yang khusus sehingga karakter khusus itu bisa dianggap sebagai karakter biasa atau karakter yang tidak bisa direpresentasikan oleh keyboard. Berikut contohnya.

```
>>> str1 = 'Don\'t eat Dek Depy\'s cake' # membuat karakter ku
tip
>>> print(str1)
Don't eat Dek Depy's cake
>>> str2 = 'Separate this \n from \n this' # membuat karakter
new line
>>> print(str2)
Separate this
    from
    this
>>> str3 = 'Increase distance between this \t and \t that' # m
embuat karakter tab
>>> print(str3)
Increase distance between this          and          that
>>>
```

Kita dapat melakukan beberapa operasi pada string, contohnya penambahan dan perkalian. Operasi penambahan (concatenation) hanya dapat dilakukan antar string, sedangkan perkalian dilakukan antara string dengan suatu integer. Berikut adalah contohnya.

```
>>> str1 = "Super"
>>> str2 = "man"
>>> str3 = str1 + str2 # penambahan string
>>> print(str3)
Superman
>>> str4 = str3*3 # perkalian string
```

```
>>> print(str4)
SupermanSupermanSuperman
>>> str5 = 3*str3 # perkalian string
>>> print(str5)
SupermanSupermanSuperman
>>>
```

Kita juga dapat mengambil karakter pada suatu indeks dari sebuah string (disebut juga *indexing*). Sebagai salah satu objek yang *iterable*, string memiliki indeks untuk setiap karakter terurutnya. Kita mengambil karakter pada indeks tertentu dengan notasi `stringnya[index]`. Kita juga dapat mengecek indeks dari kemunculan pertama sebuah karakter pada sebuah string dengan notasi: `stringnya.index(karakter)`. Berikut adalah contohnya.

```
>>> str1 = "Dek Depy"
>>> print(str1[3]) # mengeprint spasi

>>> print(str1[5]) # mengeprint huruf 'e'
e
>>> print(str1.index('e')) # mencari index kemunculan pertama
karakter 'e'
1
>>> print(str1.index('e', 2)) # mencari index 'e' pertama dimu
lai dari index ke-2
5
>>>
```

Kalian dapat membaca lebih lanjut mengenai string di [sini](#).

String Slicing

Sebelumnya kita telah mengetahui bahwa kita bisa mengambil satu karakter dari sebuah string (*indexing*). Ternyata kita juga dapat mengambil substring (bisa lebih dari satu karakter) dari sebuah string. Ini dilakukan dengan “memotong” string, atau

kita sebut dengan *slicing*. Kita melakukan *slicing* dengan cara yang mirip dengan *indexing*.

Jika *indexing* dilakukan dengan 1 parameter, *slicing* dilakukan dengan lebih dari 1 parameter. Notasi `stringnya[awal:akhir]` akan memotong string dari indeks ke-awal hingga indeks sebelum akhir. Parameter ke2 (akhir) di-*exclude* seperti pada `range()`. Kita juga dapat melakukan *slicing* dengan 3 parameter. Parameter ke-3 adalah langkah yang diambil saat memotong string (banyak karakter yang dilompati). Notasinya `string[awal:akhir:langkah]`. Berikut adalah contohnya.

```
>>> str1 = "Fakultas Ilmu Keperawatan Universitas Indonesia"
>>> print(str1[5]) # indexing (1 parameter)
t
>>> print(str1[1:5]) # slicing dari index ke 1 hingga 5
akul
>>> print(str1[5:15])
tas Ilmu K
>>> print(str1[3:20:3]) # slicing dari index 3 hingga 20 dengan
langkah 3 karakter
uaIuer
>>> print(str1[:]) # slicing dari awal hingga akhir
Fakultas Ilmu Keperawatan Universitas Indonesia
>>> print(str1[:100]) # tidak error!?
Fakultas Ilmu Keperawatan Universitas Indonesia
>>> print(str1[24:0:-1]) # langkah bisa negatif
natawarepeK umlI satluka
>>> print(str1[::-2])
asndIstseiUntwrpKul alkF
>>>
```

Kalian dapat membaca lebih lanjut mengenai string dan slicing di [sini](#).

Files

Pada lab-lab sebelumnya, kalian sudah belajar untuk mengambil input dari user melalui fungsi `input()`. Tentunya mengambil input dengan cara ini akan melelahkan user jika data yang perlu diinput ada begitu banyak, misalnya daftar NPM mahasiswa Fasilkom UI. Data sebanyak itu (atau hanya sedikit pun juga tidak masalah) bisa disimpan dalam sebuah file saja, lalu kita mengambil data dari file tersebut. Ini akan sangat memudahkan user dibandingkan harus menginput data satu per satu.

Untuk **membuka sebuah file**, kita menggunakan fungsi `open(alamat_nama_file, mode)` yang mengembalikan **objek file** dari nama file yang bersangkutan. `alamat_nama_file` adalah alamat file yang ingin dibaca, atau bisa langsung nama filenya saja jika file tersebut berada di direktori yang sama dengan file Python yang membukanya. `mode` adalah mode pekerjaan yang dilakukan terhadap file yang dibuka. Ada 3 mode, yaitu `'r'` untuk membaca saja, `'w'` untuk menulis ulang isi file, dan `'a'` untuk menambahkan isi file. Pada lab kali ini, kita hanya akan menggunakan mode `'r'` untuk membaca. Setelah selesai menggunakan file tersebut, kita HARUS ingat untuk menutup file itu dengan `variable_file.close()`. Berikut adalah contoh membuka dan menutup file di Python.

```
>>> file = open('Test.txt', 'r') # objek file dengan mode 'r'
    untuk membaca file
>>> for line in file: # mencetak setiap baris isi dari file Te
    st.txt
        print(line)
```

```
Fakultas Ilmu Komputer UI
```

```
Fakultas Ilmu Keperawatan UI
```

```
>>> file.close() # menutup file agar tidak bisa dibaca (atau d
    itulis) lagi
```

Di atas, kita sudah mengakses setiap baris dalam file dengan melakukan *looping*. Kita dapat membaca isi file dengan cara lain yang lebih baik dan hasil bacaannya

dapat disimpan dalam variable. Ada dua method yang umum dipakai, yaitu `.readline()` dan `.readlines()`.

Pada awal membuka file, **posisi baca ada di posisi paling awal dari file**. Method `.readline()` akan membaca satu baris selanjutnya dan mengembalikannya sebagai string. Posisi setelah membaca jadi ada di awal baris selanjutnya. Method ini baik digunakan untuk mengakses beberapa baris dalam sebuah file. Jika method `.readline()` dipanggil ketika posisi baca sudah di ujung akhir dari file, maka akan mengembalikan string kosong karena tidak ada lagi yang bisa dibaca. Berikut adalah contoh untuk membaca file menggunakan method `.readline()`.

```
>>> file = open('Test.txt', 'r')
>>> # saat ini posisi baca ada di paling awal dari file
>>> current_line = file.readline() # membaca line pertama dari
file
>>> current_line
'Fakultas Ilmu Komputer UI'
>>> current_line = file.readline() # membaca line selanjutnya
dari file
>>> current_line
'Fakultas Ilmu Keperawatan UI'
>>> # saat ini posisi baca ada di paling akhir dari file
>>> # saat ini file sudah habis dibaca semua
>>> current_line = file.readline() # membaca line selanjutnya
dari file
>>> current_line
''
>>> current_line = file.readline() # membaca line selanjutnya
dari file
>>> current_line
''
>>> file.close()
```

Method `.readlines()` akan langsung membaca seluruh isi file dan mengembalikan sebuah list berisi string-string. String-string tersebut adalah baris-baris dari file yang dibaca. String-string itu berurutan dalam list sesuai urutan baris di file. Setelah menggunakan method ini, posisi baca akan langsung berada di ujung akhir file. Berikut adalah contoh membaca file menggunakan method `.readlines()`.

```
>>> file = open('Test.txt', 'r')
>>> # saat ini posisi baca ada di paling awal dari file
>>> all_lines = file.readlines()
>>> # saat ini posisi baca ada di paling akhir dari file
>>> # saat ini file sudah habis dibaca semua
>>> for line in all_lines:
    print(line)

Fakultas Ilmu Komputer UI
Fakultas Ilmu Keperawatan UI
>>> print(all_lines)
['Fakultas Ilmu Komputer UI', 'Fakultas Ilmu Keperawatan UI']
>>> current_line = file.readline()
>>> current_line
''
>>> current_line = file.readline()
>>> current_line
''
>>> file.close()
```

Penjelasan lain mengenai membaca file dapat dibaca di [sini](#).

Rey's Imitation Game

The Story

Setelah membuat game pada minggu lalu, Rey menyatakan rasa terimakasih ke Kamu dan Dek Depy dengan mengajak nonton film berjudul "The Imitation Game". Saat menonton film tersebut, Rey terkesima saat melihat seseorang dapat mengubah huruf-huruf yang acak menjadi sebuah pesan rahasia.

Kekreatifan Rey muncul. Rey mengajak Dek Depy untuk saling bertukar pesan rahasia yang dikirimkan dalam suatu file dengan ekstensi .txt. Mereka saling bertukar file dan mulai memecahkannya menjadi suatu pesan rahasia.

File akan terdiri dari beberapa baris string. Baris pertama akan berisi sebuah angka N (dijamin angka). N baris berikutnya terdiri dari suatu string dengan panjang tak tentu (ada kemungkinan termasuk spasi). Setiap baris mengandung minimal **satu** angka. Baris terakhir merupakan suatu string yang bisa berisi karakter apapun (baik angka, huruf, simbol, spasi).

Sistem pemecahan pesan rahasia yang dibuat oleh Rey dan Dek Depy adalah untuk setiap baris Rey harus mencatat jumlahan setiap angka yang ada pada baris tersebut. Lalu, setelah setiap baris memperoleh jumlahan angka-angkanya, Rey akan mengambil satu karakter untuk setiap hasil jumlahan angka-angka tersebut dari string terakhir dalam input file. Lalu, untuk setiap karakter yang diperoleh, Rey akan menggabungkan karakter-karakter tersebut menjadi sebuah pesan rahasia yang telah dipecahkan.

Rey mengalami kesulitan untuk memenuhi keinginannya. Sebagai teman seangkatan yang tidak apatis dan senang berkontribusi untuk angkatan kamu, bantulah Rey untuk memenuhi keinginannya!

Contoh dari file yang mengandung pesan rahasia. Sebut saja nama file rahasia tersebut **mantan.txt** , mantan.txt akan berisi :

```
6                                #menunjukkan ada berapa baris pesan yang m
engandung angka rahasia
m4nta::n                        #4
t3r1$!nd4h                     #8
4k4n%^ s3l4lu--                #15
ter""1ng4t,                    #5
w4l4u 1n1:(                     #10
m3ny4k1tk4n):                  #12
Q=yRdpeiMuk!Pcn+rw-jfL        #kode acak yang terdapat pesan ra
hasia didalamnya
```

OUTPUT : Rindu!

Output dikeluarkan dalam Shell dengan format:

```
#Contoh untuk output di atas

Pesan rahasia dari Dek Depy adalah: Rindu!
```

Deadline Pengumpulan: 5 Oktober pukul 17.40 WIB

Komponen Penilaian

Komponen	Penjelasan	Bobot
Baca File	Implementasi yang tepat untuk membaca file	35%
Sistem Kode	Pengolahan data dengan baik	40%
<i>Correctness</i>	Ketepatan	15%

<i>Code Style</i>	Kerapihan kode	5%
Dokumentasi	Pemberian dokumentasi yang sesuai	5%

Format Pengumpulan

[Kode_Asdos]_[Nama]_[NPM]_[Kelas]_Lab05.py

Contoh: DN_Rey_1806123456_A_Lab05.py

Acknowledgements

- Daya Adianto

Authors

- KC
- UCI
- DN