

Lab05

CSGE601020

16 Oktober, 2020

Dasar - Dasar Pemrograman 1

Read & Write Files Dengan Python

Baca dokumen ini dengan saksama sebelum membuat program

Gambaran Umum

Tutorial Lab05 merupakan kelanjutan dari Lab04. Pada kesempatan kali ini, anda akan belajar membuat sebuah program konverter yang serupa program pada Lab04, namun program anda akan menerima input dari file teks dan menuliskan outputnya ke dalam file teks yang lain. Silakan baca Slide 04 dan buku rujukan kalau anda belum membacanya.

File Read and Write Pada Python

Untuk membaca sebuah file, anda dapat memanfaatkan fungsi bawaan Python yaitu `open()`. Secara umum, fungsi `open()` dapat menerima dua argumen yaitu **nama file** dan **mode**.

```
fileInput = open("theFile.txt", "r")
```

Perintah di atas akan membuka file bernama "theFile.txt" dan membuat **stream** yang menghubungkan file tsb dengan program ybs, kemudian memberikannya ke variabel bernama **fileInput**. Selanjutnya pembacaan file dilakukan lewat variable **fileInput**. Secara umum **stream** merupakan sebuah object pada Python yang dapat membantu kita mengakses dan memodifikasi sebuah file. **Stream** disebut juga sebagai **file object**.

Argumen pertama menunjukkan **nama file** yang ingin dibaca dalam bentuk string. Apabila file yang ingin dibaca berada pada direktori (folder) yang sama dengan program, maka anda dapat menuliskan nama file secara langsung seperti pada contoh di atas.

Argumen kedua menunjukkan mode yang diinginkan; "r" untuk membaca file (**read**), "w" untuk menulis file (**write**), "a" untuk menambahkan konten ke ujung file (**append**), "x" untuk membuat file baru dan membukanya untuk ditulis.

Setelah file tidak dibutuhkan lagi dalam program, anda bisa menutup file tersebut dengan perintah `close()`.

```
fileInput.close()
```

Setelah file dibuka, anda dapat melakukan iterasi atas setiap baris pada file teks dengan melakukan `for statement` terhadap `fileInput`.

```
for line in fileInput:  
    print(line, end='')
```

Baris kode di atas akan mencetak setiap baris teks pada `fileInput`, tanpa menambahkan `'\n'`.

Sedangkan untuk menulis file pada python, anda dapat memanfaatkan fungsi bawaan python yaitu `print()` dengan menambahkan beberapa argumen tertentu.

```
teks = "Hello World"  
print(teks, end = "\n", file = fileInput)
```

Baris kode di atas akan menuliskan isi variabel `teks` ke dalam `fileInput`. Tapi, sebelum dapat menuliskannya, anda harus membuka file ybs dengan mode `"w"` atau `"a"` atau `"x"`. Apa bedanya?

Argumen `end` berfungsi menambahkan string tambahan ke dalam string original yang akan diprint. Pada contoh di atas, setiap perintah `print()` akan berakhiran `"\n"` (default) sehingga setiap `print()` akan mencetak baris baru.

Argumen `file` berfungsi untuk menentukan target file yang akan ditulis.

Deskripsi Tugas

1. Buatlah sebuah direktori yang akan berisikan:
 - **Kode anda (.py)**
 - **Sebuah file teks untuk input** (misal input.txt),
 - **Sebuah file untuk output** (misal output.txt). File ini akan dibuat oleh program anda nanti.

Jadi di dalam direktori tersebut akan terdapat **3 file**.

2. Isikan file input dengan format:
 - a. Baris pertama berisi integer **jumlah bit**
 - b. 10 baris berikutnya berisi integer **desimal**
 - c. 10 baris berikutnya berisi integer **biner (2's complement)** dengan panjang sesuai baris pertama.

Anda bisa menggunakan file **input.txt** yg disediakan di SCeLE untuk testing.

3. Ketika program dijalankan:
 - a. Program anda akan meminta **input** berupa nama file yang akan **dibaca** (misal **input.txt**)
 - b. Program anda akan meminta **input** berupa nama file yang akan **dituliskan** (misal **output.txt**)
 - c. Program menampilkan teks terima kasih dan menunggu pengguna menekan tombol ENTER sebelum mengakhiri program
4. Di balik layar, program anda akan melakukan konversi setiap integer pada baris 2-11 input.txt ke dalam bentuk 2's complement kemudian menuliskan hasilnya pada baris yang sama pada output.txt. Begitu juga dengan baris 12-21 pada input.txt yang akan dikonversikan ke dalam bentuk desimalnya dan dituliskan pada baris yang sama pada output.txt
5. Setelah program dijalankan, file output akan berisi:
 - a. Baris pertama berisi integer **jumlah bit** (misal **8**)
 - b. 10 baris berikutnya berisi integer **desimal dan biner (2's complement) hasil konversinya**, contoh: **-9 = 11110111**
 - c. 10 baris berikutnya berisi integer **biner (2's complement) dan desimal hasil konversinya**, contoh: **00001001 = 9**

Contoh Interaksi

Lab05: Converter with read/write feature

Input File: **inp.txt**

Output File: **out.txt**

Thanks for using this program, check the "out.txt" to see the output.
Press Enter to continue ...

Contoh File Teks

inp.txt	out.txt
1 16	1 16
2 -6740	2 -6740 = 1110010110101100
3 12345	3 12345 = 0011000000111001
4 -1	4 -1 = 1111111111111111
5 5259	5 5259 = 0001010010001011
6 161	6 161 = 0000000010100001
7 0	7 0 = 0000000000000000
8 -1183	8 -1183 = 1111101101100001
9 958	9 958 = 0000001110111110
10 -5868	10 -5868 = 1110100100010100
11 -1025	11 -1025 = 1111101111111111
12 1011101111111001	12 1011101111111001 = -17415
13 000000000010111	13 000000000010111 = 23
14 1011011000000001	14 1011011000000001 = -18943
15 1110010101111001	15 1110010101111001 = -6791
16 0000001010000101	16 0000001010000101 = 645
17 1100010000010010	17 1100010000010010 = -15342
18 0001111001100001	18 0001111001100001 = 7777
19 1111101110000110	19 1111101110000110 = -1146
20 000000001011000	20 000000001011000 = 88
21 1111111111111110	21 1111111111111110 = -2
22	

Petunjuk Pengumpulan

- Beri nama source code dengan format penamaan <KODEASDOS>_<NamaLengkap>_<NPM>_lab05.py
- Penamaan input dan output dibebaskan, namun gunakan nama yang informatif seperti contoh di atas.
- Compress ketiga file tersebut menjadi sebuah file zip dengan format penamaan <KODEASDOS>_<NamaLengkap>_<NPM>_lab05.zip
Pastikan file dalam format zip (bukan rar)

That's all for today. Happy programming! 'Met ngoding!

L. Y. Stefanus & The Asdos Team