



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230997
Nama Lengkap	Christ Jevicto Ajimas Kirana
Minggu ke / Materi	08 / Membaca dan Menulis File

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

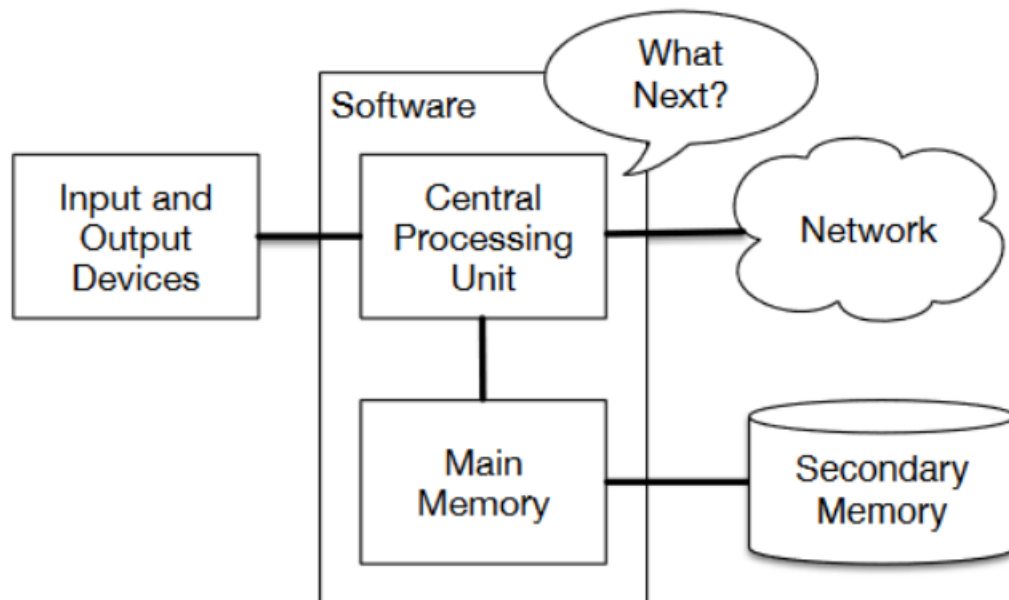
BAGIAN 1: MATERI MINGGU INI (40%)

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

A. Pengantar File

Program berjalan menggunakan memory primer di dalam computer yang bersifat tidak permanen (volatile). Yang berarti semua program akan menghilang setelah computer dimatikan, untuk menyimpan data yang permanen kita memerlukan penyimpanan sekunder (secondary memory). Contoh nya seperti gambar dibawah ini.



Secondary memory akan menyimpan file secara permanen dalam computer, file yang disimpan akan dapat menjadi berbagai jenis seperti file sistem, file program (binary), file multimedia, file teks, dan lain-lain. Setiap file memiliki atribut seperti nama file, ukuran, lokasi di harddisk, pemilik, izin akses, tanggal akses, dan lain-lain.

B. Pengaksesan File

Untuk dapat mengakses file, langkah-langkah yang harus dilakukan adalah:

1. Menyiapkan file dan path yang akan diakses
2. Open file
3. Lakukan sesuatu dengan file tersebut, seperti ditampilkan (read) isinya atau diubah / ditulis (write)
4. Close file

```
1 handle = open('mbox-short.txt')
2 print (handle)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
PS D:\lat mandiri> & C:/Users/chris/AppData/Local/Programs/Python/Python38-32/Python.exe "d:/lat mandiri/hehe.py"
<_io.TextIOWrapper name='mbox-short.txt' mode='r' encoding='cp1252'>
PS D:\lat mandiri>
```

Hasilnya berupa tampilan nama file, modusnya (r = read), dan encoding yang digunakan, yaitu Unicode UTF-8, dari sistem I/O pada Python. Jika nama file tidak ada atau tidak ditemukan, output akan menampilkan error. Pada file teks, biasanya file akan terdiri dari baris demi baris string. Cara pembacaan file teks biasanya juga menggunakan model baca baris demi baris untuk setiap string yang ditemukan sampai dengan EOF (End of File).

C. Manipulasi File

Untuk bisa memanipulasi file maka harus dimulai dari membaca file tersebut terlebih dahulu.

Cara membaca file pada Python adalah:

1. Siapkan file
2. Open file
3. Loop setiap baris pada file
4. Close file

Cara pembacaan file adalah

```
1 handle = open('mbox-short.txt')
2 count = 0
3 for line in handle:
4     count = count + 1
5 print ('line count= ', count)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\lat mandiri> & C:/Users/chris/AppData/Local/Programs/Python/Python38-32/Python.exe "d:/lat mandiri/hehe.py"
line count= 1910
PS D:\lat mandiri>
```

Cara menampilkan ukuran file teks dalam bytes, dapat digunakan fungsi len dari string yang ada pada file.

```
1 handle = open('mbox-short.txt')
2 hasil = handle.read()
3 print("Ukuran: +", len(hasil), "+ bytes")
4 print("Huruf dari belakang sendiri mundur 16 huruf adalah: " + hasil[-16::1])
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

PS D:\lat mandiri> & C:/Users/chris/AppData/Local/Programs/Python/Python311/python.exe "d:/lat mandiri/hehe.py"

Ukuran: + 94626 + bytes
Huruf dari belakang sendiri mundur 16 huruf adalah: > Preferences.

Program tersebut membuka file "mbox-short.txt", menampilkan jumlah karakter dalam file tersebut (ukuran file dalam byte), dan menampilkan string dari 16 karakter terakhir maju ke depan.

Program tersebut melakukan looping untuk membaca file "mbox" dan menampilkan hanya kalimat yang diawali dengan "Date:". Selama proses looping, kita dapat melakukan manipulasi pada string yang dibaca, seperti menampilkan bagian-bagian tertentu dari string tersebut:

```
1 handle = open('mbox-short.txt')
2 count = 1
3 for line in handle:
4     if line.startswith("Date:") and count <= 10:
5         count += 1
6         print(line)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

n.exe "d:/lat mandiri/hehe.py"

Date: Sat, 5 Jan 2008 09:12:18 -0500

Date: 2008-01-05 09:12:07 -0500 (Sat, 05 Jan 2008)

Date: Fri, 4 Jan 2008 18:08:57 -0500

Date: 2008-01-04 18:08:50 -0500 (Fri, 04 Jan 2008)

Date: Fri, 4 Jan 2008 16:09:02 -0500

Date: 2008-01-04 16:09:01 -0500 (Fri, 04 Jan 2008)

Date: Fri, 4 Jan 2008 15:44:40 -0500

Date: 2008-01-04 15:44:39 -0500 (Fri, 04 Jan 2008)

Date: Fri, 4 Jan 2008 15:01:38 -0500

Date: 2008-01-04 15:01:37 -0500 (Fri, 04 Jan 2008)

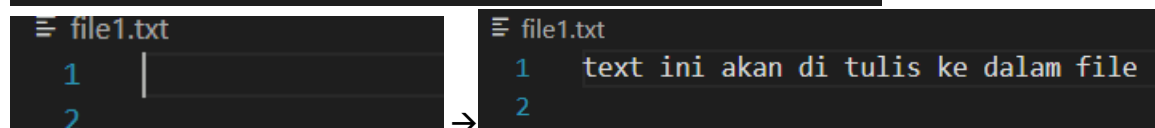
Baris kosong pada setiap baris di atas muncul karena file "mbox" sudah mengandung newline di setiap baris dan ditambah dengan perintah print yang menyebabkan double newline. Untuk menghindari double newline, kita dapat menggunakan perintah `rstrip` pada setiap baris atau menggunakan perintah `print` tanpa menambahkan newline. Dengan demikian, program akan mencetak setiap baris yang diawali dengan "Date:" tanpa menambahkan double newline.

D. Penyimpanan File

Dalam Python, cara untuk menulis ke file mirip dengan cara membuka file sebelumnya, hanya dengan mengubah mode pembukaannya dari 'r' menjadi 'w'. Perintah `write(<string>)` digunakan untuk menuliskan isi string ke dalam file yang telah dibuka. Setelah selesai menulis, jangan lupa untuk menutup file menggunakan perintah `close()`.

Contoh =

```
1 handle = open('file1.txt', 'w')
2 nulis = "text ini akan di tulis ke dalam file\n"
3 handle.write(nulis)
4 handle.close()
```



MATERI 2

Kegiatan Praktikum

Kegiatan praktikum akan dilakukan untuk memanipulasi lebih dalam lagi file teks mboxshort.txt sebagai berikut:

Kasus 8.1 Program harus mampu menerima nama file teks tertentu (mbox-short.txt atau mbox.txt) dan kemudian tampilkanlah semua baris yang mengandung string web pada file tersebut yang menggunakan domain berakhiran '*.ac.uk' dan berapa jumlahnya.

```
1 handle = open('mbox-short.txt')
2 c = 0
3 for line in handle:
4     if line.find("ac.uk") != -1:
5         c += 1
6         print("web domain 'ac.uk' ditemukan di" + line.strip() + "\n")
7         print("jumlah: ", c)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

jumlah: 154
jumlah: 154
web domain 'ac.uk' ditemukan di"Received: FROM paploo.uhi.ac.uk (app1.prod.collab.uhi.ac.uk [194.35.219.184])"
jumlah: 155
jumlah: 155
jumlah: 155
web domain 'ac.uk' ditemukan di"Received: from paploo.uhi.ac.uk (localhost [127.0.0.1])"
jumlah: 156
web domain 'ac.uk' ditemukan di"by paploo.uhi.ac.uk (Postfix) with ESMTP id DEC65ADC79;"
jumlah: 157
jumlah: 157
jumlah: 157
jumlah: 157
jumlah: 157
web domain 'ac.uk' ditemukan di"Received: from prod.collab.uhi.ac.uk ([194.35.219.182])"
jumlah: 158
web domain 'ac.uk' ditemukan di"by paploo.uhi.ac.uk (JAMES SMTP Server 2.1.3) with SMTP ID 917"
jumlah: 159
jumlah: 159
jumlah: 159
jumlah: 159
web domain 'ac.uk' ditemukan di"by shmi.uhi.ac.uk (Postfix) with ESMTP id 1FB8042B30"
jumlah: 160
jumlah: 160
jumlah: 160

Program harus dapat menerima nama file teks tertentu (misalnya "mbox-short.txt" atau "mbox.txt"), kemudian menampilkan berapa baris string pada file yang diawali dengan kata "Subject", dan mengubah semua baris tersebut menjadi "capitalized each word".

Untuk menerima input nama file, digunakan perintah 'input()' untuk meminta pengguna memasukkan nama file. File tersebut sebaiknya berada dalam satu folder yang sama agar mudah diakses. Kemudian, untuk menampilkan baris-baris file yang diawali dengan string "Subject:", digunakan metode 'startswith()'. Jumlah baris ditampilkan dengan menggunakan variabel counter untuk setiap baris yang ditemukan.

```
1 handle = open('mbox-short.txt')
2 c = 0
3 for line in handle:
4     if line.startswith('Subject: '):
5         c += 1
6         line = line.strip().title()
7         print(line)
8 print("jumlah: ", c)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
ssessment/Integration/Helper/Standalone Samigo-Services/Src/Java/Org/Sakaiproject
/Tool/Assessment/Shared/Impl/Grading
Subject: [Sakai] Svn Commit: R39754 - In Polls/Branches/Sakai_2-5-X: . Tool Tool/
Src/Java/Org/Sakaiproject/Poll/Tool Tool/Src/Java/Org/Sakaiproject/Poll/Tool/Evol
vers Tool/Src/Webapp/Web-Inf
Subject: [Sakai] Svn Commit: R39753 - In Polls/Trunk: . Tool Tool/Src/Java/Org/Sa
kaiproject/Poll/Tool Tool/Src/Java/Org/Sakaiproject/Poll/Tool/Evolvers Tool/Src/w
ebapp/Web-Inf
Subject: [Sakai] Svn Commit: R39752 - In Podcasts/Branches/Sakai_2-5-X/Podcasts-A
pp/Src/Webapp: Css Podcasts
Subject: [Sakai] Svn Commit: R39751 - In Podcasts/Branches/Sakai_2-5-X/Podcasts-A
pp/Src/Webapp: Css Images Podcasts
Subject: [Sakai] Svn Commit: R39750 - Event/Branches/Sak-6216/Event-Util/Util/Src
/Java/Org/Sakaiproject/Util
Subject: [Sakai] Svn Commit: R39749 - In Bspace/Site-Manage/Sakai_2-4-X/Site-Mana
ge-Tool/Tool/Src: Bundle Webapp/Vm/Sitesetup
Subject: [Sakai] Svn Commit: R39746 - In Bspace/Site-Manage/Sakai_2-4-X/Site-Mana
ge-Tool/Tool/Src: Bundle Webapp/Vm/Sitesetup
Subject: [Sakai] Svn Commit: R39745 - Providers/Trunk/Cm/Cm-Authz-Provider/Src/Ja
va/Org/Sakaiproject/Coursemanagement/Impl/Provider
Subject: [Sakai] Svn Commit: R39744 - Oncourse/Branches/Oncourse_Opc_122007
Subject: [Sakai] Svn Commit: R39743 - Gradebook/Branches/Oncourse_2-4-2/App/Ui/Sr
c/Java/Org/Sakaiproject/Tool/Gradebook/Ui
Subject: [Sakai] Svn Commit: R39742 - Gradebook/Branches/Oncourse_2-4-2/App/Ui/Sr
c/Java/Org/Sakaiproject/Tool/Gradebook/Ui
jumlah: 27
```

Program harus dapat menampilkan ukuran file dalam kilobit (KB) dari sebuah file teks, dan menangani error jika file yang diinputkan tidak ditemukan.

Untuk menerima input nama file, digunakan perintah 'input()' untuk meminta pengguna memasukkan nama file. File tersebut sebaiknya berada dalam satu folder yang sama agar mudah diakses. Untuk menghitung ukuran file, dapat digunakan perintah 'len()' untuk menghitung panjang file dalam byte, kemudian dibagi 1000 untuk mengonversinya menjadi kilobit (KB). Untuk menangani error, gunakan blok 'try' dan 'except' untuk menangkap 'FileNotFoundError'.

```
1 filename = input('nama file = ')
2 try:
3     handle = open(filename)
4     total = 0
5     for line in handle:
6         total += len(line)
7         kb = total / 1000
8     print("Ukuran: " + str(kb) + " KB")
9 except:
10    print("file tidak ditemukan!")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Ukuran: 93.821 KB
Ukuran: 93.892 KB
Ukuran: 93.964 KB
Ukuran: 93.965 KB
Ukuran: 94.034 KB
Ukuran: 94.107 KB
Ukuran: 94.192 KB
Ukuran: 94.193 KB
Ukuran: 94.203 KB
Ukuran: 94.254 KB
Ukuran: 94.327 KB
Ukuran: 94.4 KB
Ukuran: 94.401 KB
Ukuran: 94.402 KB
Ukuran: 94.425 KB
Ukuran: 94.549 KB
Ukuran: 94.625 KB
Ukuran: 94.626 KB
PS D:\lat mandiri> & C:/Users/chris/AppData/Local/Programs/Python/Python38-64/Python.exe "d:/lat mandiri/hehe.py"
nama file = mbox-short
file tidak ditemukan!

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

```
1 file1 = open ('file1.txt')
2 file2 = open ('file2.txt')
3 hasil1 = file1.read()
4 hasil2 = file2.read()
5
6 print (hasil1)
7 print (hasil2)
8
9 perbedaan= []
10 for i in (hasil1, hasil2):
11     if hasil1 != hasil2:
12         perbedaan = True
13
14 if perbedaan:
15     print("Ada perbedaan antara kedua file")
16 else:
17     print("Tidak ada perbedaan antara kedua file.")
18
iya sama iya
iya sama iya
Tidak ada perbedaan antara kedua file.
PS D:\lat mandiri> & C:/Users/chris/AppData/Local/Programs/Python/Python39-64/Python.exe "d:/lat mandiri/lat.py"
engga beda
iya sama iya
Ada perbedaan antara kedua file
```

Di soal ini saya diminta untuk membuat program yang akan membaca dua file lalu akan memberi tahu bahwa isinya sama atau beda. Lalu hasil1 dan hasil2 akan menjadi variable yang meBaca isi 2 file tersebut lalu sebelum isi variable hasil di apa apa kan kita akan mengeprint variable hasil dahulu, lalu saya membuat list kosong untuk nanti di isi dari perbedaan yang ditemukan di kedua file, lalu kita mengiterasi tiap file, lalu jika ada isi file berbeda saat di iterasi maka perbedaan akan menjadi True. Jika perbedaan benar maka akan menghasilkan output **"Ada perbedaan antara kedua file"** jika perbedaan mengeluarkan hasil lain maka hasilnya akan "Tidak ada perbedaan antara kedua file".

SOAL 2

```
namafile = "soal.txt"

file = open(namafile, 'r')
print ('nama file = ', namafile)

for line in file:
    pertanyaan, jawaban = line.strip().split('||')
    pertanyaan = pertanyaan.strip()
    jawaban = jawaban.strip()
    print(pertanyaan)
    jawabanpengguna = input ('jawaban anda = ').strip()
    if jawabanpengguna.lower() == jawaban.lower():
        print('jawaban anda benar')
    else:
        print('jawaban anda salah')
    print()
```

✓ 21.5s

```
nama file = soal.txt
1+1 =
jawaban anda benar

Bendera Indonesia?
jawaban anda salah

Kota gudeg adalah:
jawaban anda benar

Komponen PC untuk penyimpanan file adalah...
jawaban anda benar

50 * 20 =
jawaban anda benar
```

Penjelasan =

Program akan membuka file "soal.txt. lalu tiap baris akan di bersihkan oleh '.strip()'. kemudian setiap baris akan di bedakan menjadi dua bagian menggunakan '.split(||)'. lalu akan menampilkan pertanyaan ke pengguna lalu akan membersihkan spasi jawaban pengguna dengan '.strip()'. lalu jawaban akan di bandingkan dengan jawaban yang benar jika jawaban salah maka akan menghasilkan output "jawaban anda benar" lalu akan melanjutkan ke pertanyaan berikutnya, dan jika jawaban salah "jawab anda salah" lalu akan melanjutkan ke pertanyaan berikutnya.

GITHUB LINK = <https://github.com/Yeeemeki/praktikum-algoritma-dan-pemograman.git>