



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	<71230989>
Nama Lengkap	<Yohanes Nevan Adventus Wibawa>
Minggu ke / Materi	09 / 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.

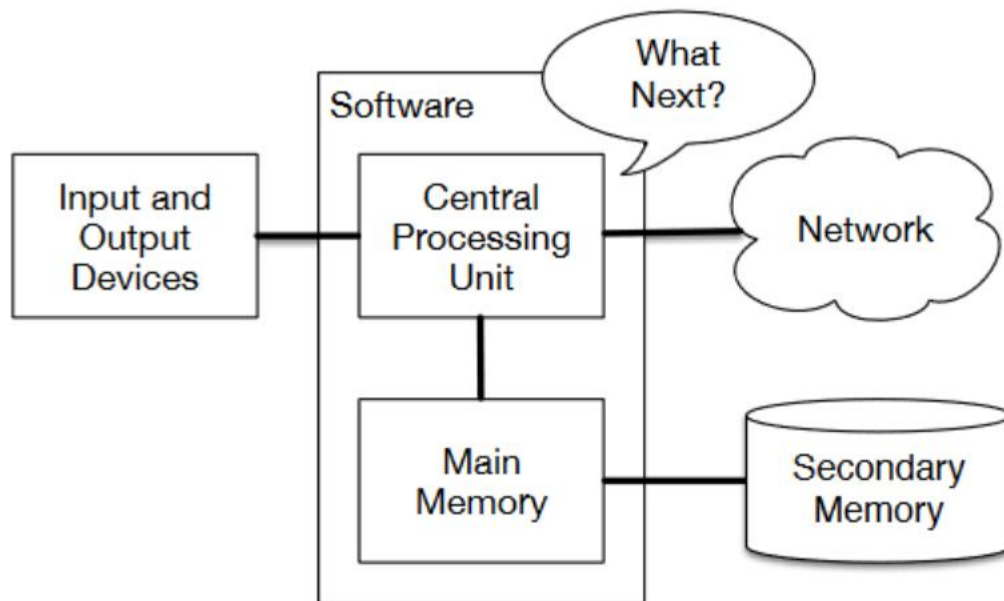
PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2024

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.

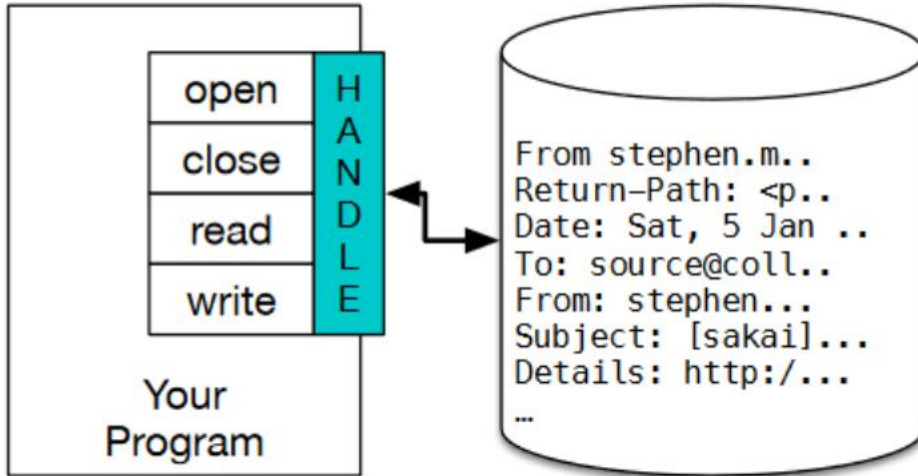
Pengantar File

Program yang sedang berjalan memanfaatkan memori utama di dalam komputer. Semua informasi yang terkait dengan program tersebut tersimpan di dalam memori, dan saat program diakhiri dan dimatikan, segala data yang ada di dalamnya juga lenyap. Memori ini bersifat tidak permanen atau mudah berubah (volatile). Karena sifat ini, program yang mengandalkan memori utama tidak dapat menyimpan data setelah program dinonaktifkan. Agar dapat menyimpan data, diperlukan penggunaan media penyimpanan tetap, yang disebut sebagai memori sekunder. Memori sekunder ini dapat diilustrasikan dalam gambar di bawah ini.



Pengaksesan File

```
handle = open('mbox-short.txt')
print(handle)
<_io.TextIOWrapper name='mbox-short.txt' mode='r' encoding='cp1252'>
```



Jadi hasilnya berupa tampilan nama file, modenya (`r = read`), dan encoding yang digunakan yaitu unicode UTF-8 dari sistem io pada Python. Jika nama file tidak ada / tidak ditemukan, maka output akan error:

```
File "c:\Users\LENOVO\Tugas PrakAlPro\Tugas 9\file.py", line 1, in <module>
    handle = open('mbox-short0.txt')
            ^^^^^^^^^^^^^^^^^^^^^^^
FileNotFoundError: [Errno 2] No such file or directory: 'mbox-short0.txt'
```

Memmanipulasi File

Menghitung Jumlah Line

```
handle = open('mbox-short.txt')
count = 0
for line in handle:
    count = count + 1
print('Line Count:', count)
Line Count: 1910
```

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

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

Program di atas akan membuka file mbox-short.txt, menampilkan ukuran berapa banyak huruf yang ada pada file tersebut (catatan: kalau dianggap 1 karakter = 1 byte, maka bisa disebut juga ukuran berapa banyak karakter = ukuran file tersebut dalam byte), dan terakhir menampilkan string dari huruf paling belakang maju 16 huruf kedepan.

Selama dilakukan looping kita juga dapat melakukan manipulasi terhadap file tersebut, seperti misalnya menangkap / menampilkan bagian dari string. Seperti pada contoh file mbox, saat looping kita dapat menampilkan hanya kalimat yang diawali dengan "tanggal" saja, yaitu "Date :". Perhatikan contoh program berikut:

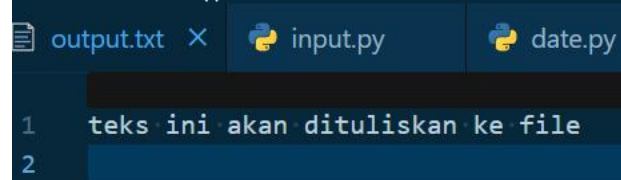
```
handle = open('mbox-short.txt')
count = 1
for line in handle:
    if line.startswith("Date:") and count <= 10:
        count += 1
        print(line)
```

Mengapa ada baris kosong disetiap baris di atas? hal ini karena dari file mbox sudah terdapat newline di setiap baris dan ditambah perintah print yang kita buat, sehingga terjadi dobel newline. Silahkan modifikasi agar enter tidak dobel! Kita bisa menggunakan perintah.rstrip pada line atau menggunakan perintah print tanpa newline.

Penyimpanan File

Dalam Python cara untuk menulis ke file adalah sama dengan cara membuka (open) file pada sub bab sebelumnya, hanya perlu mengubah metodenya saja yang tadinya r menjadi w sebagai berikut: fout=open('output.txt','w') Untuk menuliskan isi string ke dalam file output.txt langsung saja digunakan perintah write() dan jangan lupa tutup file dengan close() Contoh lengkap adalah sebagai berikut:

```
handle = open('output.txt','w')
tulisan = "teks ini akan dituliskan ke file\n"
handle.write(tulisan)
handle.close()
```



Kegiatan Praktikum

Kasus 8.1

Program harus bisa menerima nama file teks tertentu, seperti mbox-short.txt atau mbox.txt. Kemudian, program akan menampilkan semua baris dalam file tersebut yang memuat string web dengan domain yang berakhir pada '*.ac.uk', beserta jumlah baris yang memenuhi kriteria tersebut.

Untuk menerima input nama file, digunakan perintah input, lalu pengguna diminta untuk memasukkan nama file. File tersebut disarankan untuk ditempatkan dalam satu folder yang sama untuk kemudahan. Untuk menampilkan baris-baris dalam file yang mengandung URL web dengan domain yang berakhir dengan '*.ac.uk', perintah find() digunakan. Untuk menghitung jumlah baris yang ditemukan, penghitung digunakan untuk setiap baris yang memenuhi kriteria tersebut. Di bawah ini adalah kode programnya.

```
filename = input("nama file: ")
handle = open(filename)
c = 0
for line in handle:
    if line.find("ac.uk") != -1:
        c += 1
print("Web domain 'ac.uk' ditemukan di \"" + line.strip() + "\"")
print("Jumlah: ",c)
```

8.1.py
nama file: mbox-short.txt
Web domain 'ac.uk' ditemukan di ""
Jumlah: 166

Kode ini adalah sebuah program sederhana yang meminta pengguna untuk memasukkan nama file melalui input. Kemudian, program membuka file yang dimaksud dan menginisialisasi variabel 'c' (counter) dengan nilai 0. Program kemudian melakukan iterasi melalui setiap baris dalam file tersebut menggunakan loop for.

Pada setiap iterasi, program melakukan pencarian string "ac.uk" di dalam baris tersebut menggunakan metode `find()`. Jika string tersebut ditemukan dalam baris, nilai variabel 'c' akan ditambah 1. Setelah selesai iterasi, program mencetak baris terakhir dari file (tanpa newline) yang diakses menggunakan variabel 'line', serta jumlah kemunculan string "ac.uk" yang telah dihitung. Namun, perlu dicatat bahwa ada sedikit keanehan dalam kode ini.

Pernyataan `print("Web domain 'ac.uk' ditemukan di \"" + line.strip() + "\"")` seharusnya diletakkan di dalam loop `for`, sehingga bisa mencetak setiap baris yang mengandung string "ac.uk". Sebaliknya, kode ini hanya akan mencetak baris terakhir dari file, yang mungkin tidak mengandung string "ac.uk", dan juga bisa menyebabkan error jika file tersebut kosong.

Kasus 8.2

Program harus dapat menerima nama file teks tertentu, seperti mbox-short.txt atau mbox.txt. Setelah itu, program akan menampilkan jumlah baris yang dimulai dengan kata "Subject" dan mengubah setiap kata dalam baris tersebut menjadi huruf kapital di awal kata.

Untuk menerima input nama file, program menggunakan perintah input, yang meminta pengguna untuk memasukkan nama file. Disarankan agar file tersebut berada dalam satu folder yang sama untuk kemudahan akses. Untuk menampilkan baris-baris dalam file yang diawali dengan string 'Subject:', perintah startswith() digunakan. Kemudian, untuk menghitung jumlah baris yang memenuhi kriteria tersebut, digunakan penghitung untuk setiap baris yang ditemukan. Berikut adalah kode programnya.

```
handle = open("File/mbox-short.txt")
c = 0
for line in handle:
    if line.startswith("Subject:"):
        c += 1
        line = line.strip().title()
        print(line)
print("Jumlah: ",c)
```

Subject: [Sakai] Svn Commit: R39752 - In Podcasts/Branches/Sakai_2-5-X/Podcasts-App/Src/Webapp: Css Podcasts
Subject: [Sakai] Svn Commit: R39751 - In Podcasts/Branches/Sakai_2-5-X/Podcasts-App/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-Manage-Tool/Tool/Src: Bundle Webapp/Vm/Sitesetup
Subject: [Sakai] Svn Commit: R39746 - In Bspace/Site-Manage/Sakai_2-4-X/Site-Manage-Tool/Tool/Src: Bundle Webapp/Vm/Sitesetup
Subject: [Sakai] Svn Commit: R39745 - Providers/Trunk/Cm/Cm-Authz-Provider/Src/Java/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/Src/Java/Org/Sakaiproject/Tool/Gradebook/Ui
Subject: [Sakai] Svn Commit: R39742 - Gradebook/Branches/Oncourse_2-4-2/App/Ui/Src/Java/Org/Sakaiproject/Tool/Gradebook/Ui
Jumlah: 27

Kode ini membuka file "mbox-short.txt" dan menginisialisasi variabel 'c' (counter) dengan nilai 0. Selanjutnya, program melakukan iterasi melalui setiap baris dalam file tersebut menggunakan loop for.

Pada setiap iterasi, program memeriksa apakah baris tersebut dimulai dengan string "Subject:". Jika ya, maka nilai variabel 'c' akan ditambah 1, menandakan bahwa ditemukan satu subjek pesan baru. Setelah itu, program memproses baris tersebut dengan metode `strip()` untuk menghapus spasi di awal dan akhir, dan kemudian menggunakan metode `title()` untuk mengonversi teks menjadi huruf kapital di awal setiap kata. Hasilnya, baris subjek pesan yang sudah diproses ini dicetak.

Setelah selesai iterasi, program mencetak jumlah total subjek pesan yang telah ditemukan. Kode ini efektif untuk menghitung jumlah subjek pesan yang ada dalam file "mbox-short.txt" dan mencetak subjek-subjek tersebut dalam format yang dimodifikasi.

Kasus 8.3

Program harus mampu menampilkan ukuran file dalam KB dari sebuah file teks dan handle error jika file yang diinputkan tidak ditemukan.

Untuk menerima input nama file, digunakan perintah input yang meminta pengguna untuk memasukkan nama file. Disarankan agar file tersebut berada dalam satu folder yang sama untuk kemudahan akses. Untuk menghitung ukuran file, perintah len digunakan untuk setiap baris string yang dibaca dari file. Ukuran file yang diperoleh dalam byte kemudian dibagi 1000 untuk mengkonversinya menjadi KB. Untuk menangani kesalahan, blok try-catch digunakan. Berikut adalah kode programnya.

```
try:
    nama = input("Nama File: ")
    handle = open(nama, "r")
    total_byte = 0
    for line in handle:
        total_byte += len(line.strip())
    print(f"{total_byte/1000} KB")
except:
    print("File tidak ditemukan!")
```

PS C:\Users\LENOVO\Tugas Prak
8.3.py"
Nama File: mbox-short.txt
91.311 KB

Kode ini meminta pengguna untuk memasukkan nama file, kemudian mencoba membuka file tersebut. Jika file berhasil dibuka, program menghitung ukuran file dalam kilobyte dengan menghitung jumlah byte dari setiap baris dalam file dan mengambil rata-rata dari total byte. Akhirnya, program mencetak ukuran file dalam format kilobyte. Jika terjadi kesalahan saat membuka atau membaca file, program mencetak pesan "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

```
def membandingkan_file(file1, file2):  
    with open(file1, 'r') as f1, open(file2, 'r') as f2:  
        while True:  
            line1 = f1.readline()  
            line2 = f2.readline()  
            if not line1 and not line2:  
                break  
            if line1 != line2:  
                print(f"Difference at line {f1.tell()}:")  
                print(f"File 1: {line1.strip()}")  
                print(f"File 2: {line2.strip()}")  
  
membandingkan_file('p1.txt', 'p2.txt')
```

Latihan Mandiri 1.py
Difference at line 1:
File 1: Halo Dunia!
File 2: Hello World!

Fungsi `membandingkan_file` ini dibuat untuk membandingkan dua file teks yang diberikan baris per baris. Pertama, fungsi membuka kedua file menggunakan blok `with` untuk memastikan file ditutup setelah selesai digunakan. Kemudian, dengan menggunakan loop `while True`, fungsi membaca satu baris dari masing-masing file pada setiap iterasi. Selama iterasi, fungsi memeriksa apakah kedua baris yang dibaca tidak kosong (`if not line1 and not line2:`), yang menandakan bahwa kedua file telah selesai dibaca. Jika ya, loop dihentikan dengan `break`.

Jika kedua baris yang dibaca tidak sama (`if line1 != line2:`), fungsi mencetak pesan yang menunjukkan perbedaan pada baris tersebut, beserta isi baris dari masing-masing file. Penting untuk dicatat bahwa `f1.tell()` digunakan untuk mendapatkan posisi (offset) saat ini dalam file `file1`, yang memberikan indikasi baris ke berapa yang sedang dibaca.

Namun, perlu diperhatikan bahwa ini mungkin tidak memberikan hasil yang akurat dalam semua kasus, terutama jika file menggunakan karakter baru yang berbeda (`\r\n` versus `\n`) atau jika file menggunakan format teks yang terenkripsi. Selanjutnya, fungsi `membandingkan_file` dipanggil dengan dua nama file sebagai argumen (`membandingkan_file('p1.txt', 'p2.txt')`) untuk membandingkan file "p1.txt" dan "p2.txt". Fungsi ini memberikan kemampuan yang cukup sederhana untuk membandingkan dua file teks baris per baris, mencetak perbedaan jika ada, dan menghentikan proses saat salah satu file telah selesai dibaca.

SOAL 2

```
def tampilkan_soal(nama_file):
    with open(nama_file, 'r') as f:
        soal_jawaban = f.readlines()

    for line in soal_jawaban:
        soal,jawaban = line.strip().split('||')
        print(soal)
        jawaban_user = input("Jawab: ")
        if jawaban_user.strip().lower() == jawaban.strip().lower():
            print("Jawaban benar!")
        else:
            print("Jawaban salah!")

# Ganti dengan nama file teks Anda
nama_file = "soal.txt"
tampilkan_soal(nama_file)
1+1 =
Jawab: 2
Jawaban benar!
Bendera Indonesia?
Jawab: Merah Putih
Jawaban benar!
Kota gudeg adalah:
Jawab: Yogyakarta
Jawaban benar!
Komponen PC untuk penyimpanan file adalah...
Jawab: HARDDISK
Jawaban benar!
50 * 20 =
Jawab: 1000
Jawaban benar!
PS C:\Users\LENOVO\Tugas PrakAlPro\Tugas 9> 
```

Program ini mendefinisikan fungsi `tampilkan_soal` yang bertujuan untuk memuat dan menampilkan soal beserta jawabannya dari sebuah file teks. Fungsi ini menerima satu parameter `nama_file`, yang merupakan nama file teks yang berisi daftar soal dan jawabannya.

Pertama-tama, fungsi membuka file tersebut dalam mode baca `r` menggunakan blok `with`, dan membaca seluruh kontennya menggunakan `f.readlines()`. Setiap baris dari file tersebut dianggap sebagai satu set soal dan jawaban yang dipisahkan oleh delimiter `||`.

Selanjutnya, fungsi melakukan iterasi melalui setiap baris soal dan jawaban. Setiap baris dipisahkan menjadi dua bagian, yaitu `soal` dan `jawaban`, menggunakan metode `split('||')`. Kemudian, fungsi mencetak soal dan meminta pengguna untuk menjawab. Setelah pengguna memberikan jawaban, fungsi membandingkannya dengan jawaban yang benar (setelah di-stripped dan diubah menjadi huruf

kecil untuk menghindari perbedaan kapitalisasi). Jika jawaban pengguna sesuai dengan jawaban yang benar, fungsi mencetak "Jawaban benar!", dan jika tidak, mencetak "Jawaban salah!". Terakhir, fungsi ini dipanggil dengan menggunakan nama file "soal.txt" sebagai contoh.

Link Github

Link = <https://github.com/YohanesNevan/Tugas-Laporan-Alpro-9.git>