



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231050
Nama Lengkap	Samuel Natanael
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.

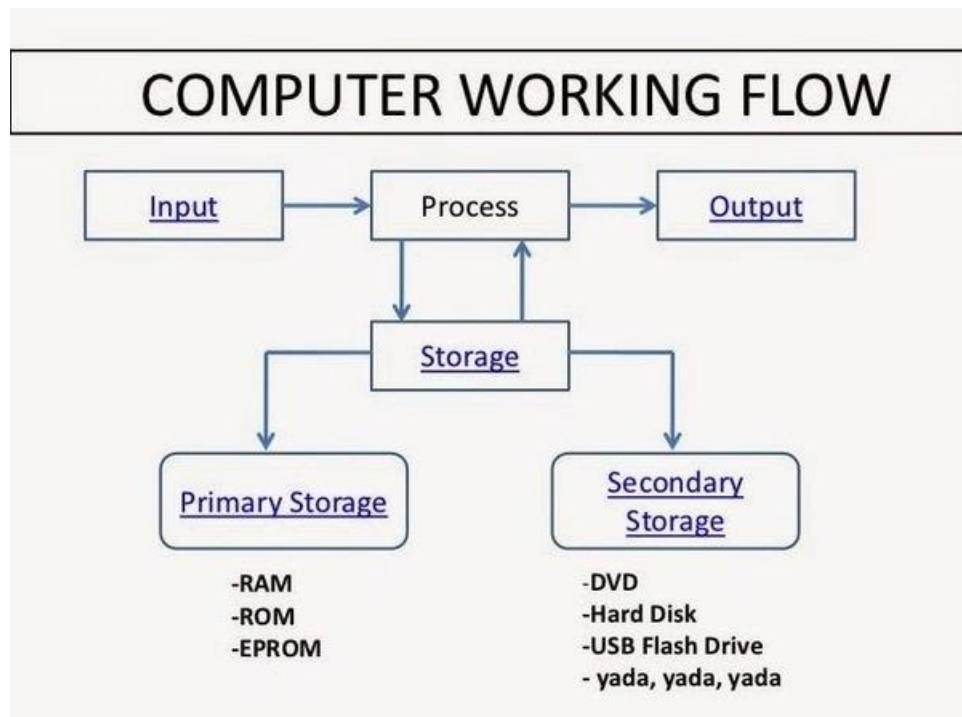
SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

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

BAGIAN 1: MATERI MINGGU INI (40%)

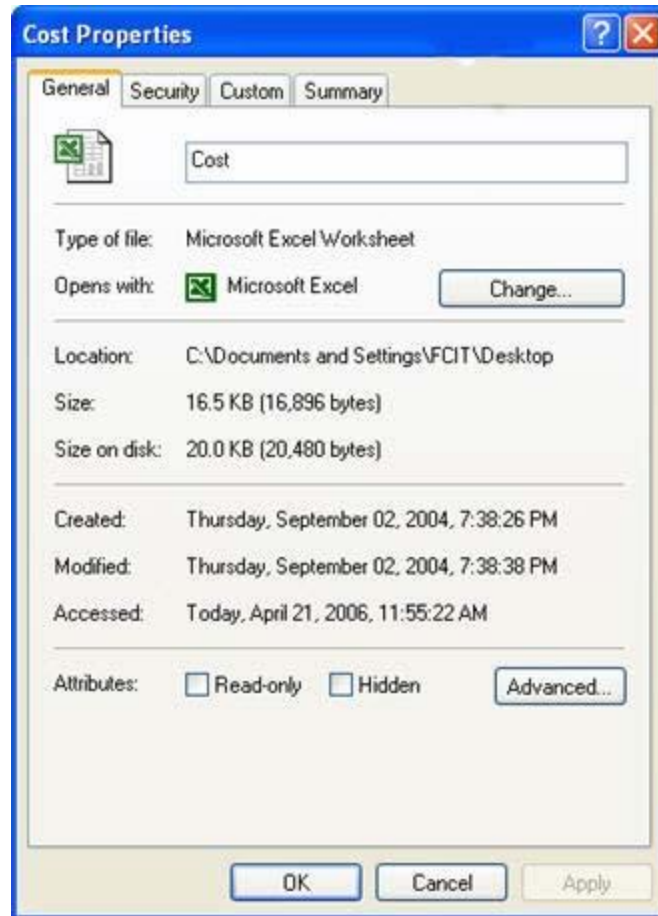
Pengantar File

Komputer memerlukan memori primer untuk menjalankan program. Data program disimpan di memori, dan ketika program selesai dijalankan dan dimatikan, semua data juga hilang. Program yang menggunakan memori primer tidak akan dapat menyimpan data setelah program dimatikan karena penyimpanan data di dalam memori bersifat tidak permanen atau volatile. Penyimpanan tetap, atau secondary memory, diperlukan untuk menyimpan data pada program.



Gambar 8.1 Secondary Storage di Komputer (Diambil dari <https://id.quora.com/Apa-pengaruh-RAM-pada-laptop>)

Pada dasarnya, file adalah bit-bit data yang disimpan secara permanen di secondary memory, berupa kumpulan informasi yang saling berhubungan satu sama lain, dan dapat digunakan untuk menyimpan data dari program sehingga tidak akan hilang bahkan jika komputer dimatikan. Jenis file termasuk file system, file program (binary), file multimedia, file teks, dan sebagainya. Nama, ukuran, lokasi, pemilik, hak akses, tanggal akses, dan atribut lainnya. Contoh property sebuah file dapat dilihat dari gambar 8.2



Gambar 8.2 Contoh Properti File (diambil dari <https://etc.usf.edu/techease/win/files-sharing/how-do-i-view-the-properties-of-a-file-or-folder/>)

Pengaksesan File

Ada beberapa langkah yang perlu dilakukan jika ingin mengakses sebuah file:

1. Menyiapkan file dan path
2. Menggunakan fungsi `open()`
3. Menentukan mode file ("w" atau "r" mode yang paling sering dipakai)
4. Close file

Berikut adalah contoh kodenya

```
handle = open("file/mbox-short.txt", "r")
print(handle)
handle.close()
```

output:

```
<_io.TextIOWrapper name='file/mbox-short.txt' mode='r' encoding='cp1252'>
```

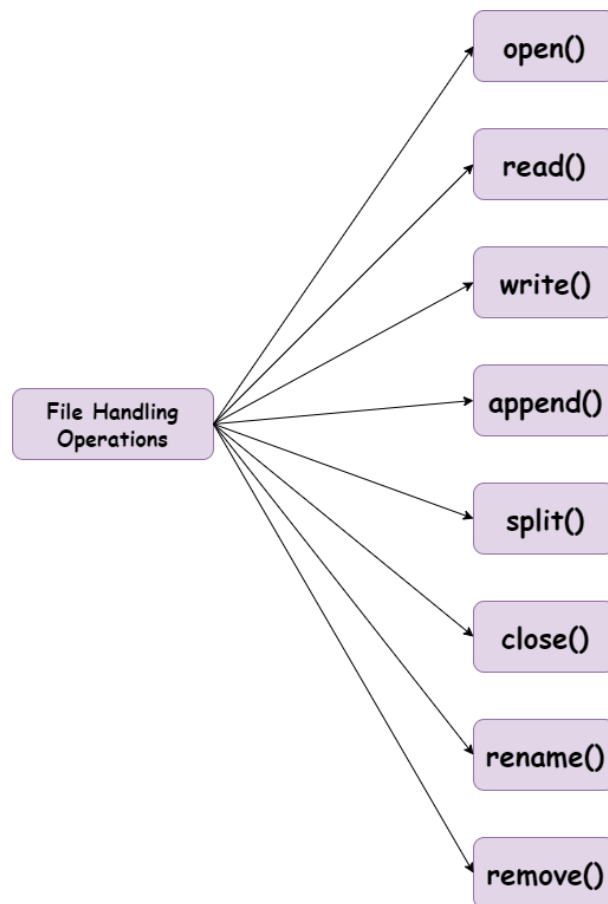
Hasilnya menunjukkan tampilan nama file, modenya ("r" = read), dan encoding yang digunakan, yaitu unicode "cp1252" yang digunakan oleh sistem io Python. (digunakan ("file/mbox-short.txt") karena file mbox-short.txt terdapat dalam file "file")

Jika file tidak dapat ditemukan maka akan muncul pesan

FileNotFoundError: [Errno 2] No such file or directory: 'file/mbx-short.txt'

Mode	Description
r	Read Mode (default value)
w	Write Mode (file is opened in write-only mode)
a	Append Mode (Opens a file for appending at the end of the file without truncating)
x	Create Mode (Creates a new file but will return an error if the file already exists.
t	Open the file in text mode.
b	Opens the file in binary mode. Binary mode returns bytes. It is mainly used while dealing with the non-text file such as images.
+	Opens the file for updating (reading and writing)

Gambar 8.3 Mode-Mode Akses File Dalam Python (diambil dari <https://www.shiksha.com/online-courses/articles/file-handling-in-python/#:~:text=Apart%20from%20the%20most%20commonly,file-handling%20functions%20in%20Python.>)



Gambar 8.4 Python File Handling (diambil dari <https://www.askpython.com/python/python-file-handling>)

Function	Description
open()	Used to open a file.
readable()	Used to check whether the file can be read or not.
readline()	Used to read a single line from the file.
readlines()	Used to read all the content in the form of lines.
read()	Used to read whole content at a time.
writable()	Used to check whether the file is writable or not.
write()	Used to write a string to the file.
writelines()	Used to write multiple strings at a time.
close()	Used to close the file from the program.

Gambar 8.5 Fungsi File Handling (diambil dari <https://www.shiksha.com/online-courses/articles/file-handling-in-python/#:~:text=Apart%20from%20the%20most%20commonly,file-handling%20functions%20in%20Python.>)

Manipulasi File

Cara memanipulasi file mengikuti prosedur yang sama dengan mengakses file. Tetapi sebelum closing file, kita akan melakukan Langkah yang berbeda. Dengan looping setiap baris dalam file kita bisa mengetahui jumlah barisnya tanpa melihat filenya.

```
handle = open("file.txt", "r")
count = 0 # count = jumlah baris
for line in handle:
    count += 1
print(count)
```

Ada cara lain untuk menghitung file yaitu dengan cara otomatis melalui readlines dan len, tetapi cara ini dapat memakan ram dan waktu yang banyak untuk file besar.

```
handle = open("file.txt", "r")
count = len(handle.readlines())
print(count)
```

1 karakter = 1 byte, maka bisa dibayangkan kalau file yang memiliki 100 huruf mempunyai 100 byte

Kita juga bisa melakukan looping untuk hanya mengambil bagian tertentu dari string. Di dalam looping kita akan menggunakan function seperti "line.startswith()" untuk hanya menangkap string yang mulai dengan kata/huruf tersebut.

```
handle = open("file.txt", "r")
count = 0
for line in file:
    if line.startswith("hi") and count <= 10:
        count += 1
        print(line)
```

Code diatas akan mencari baris kata yang mulai dari "hi" sekaligus memprint barisnya dan count akan menghitung jumlah baris yang mulai dengan "hi". Setiap baris berkemungkinan dipisah dengan baris kosong, hal ini karena dalam file txtnya sudah terdapat newline di setiap barisnya, plus dengan perintah print. Untuk menghilangkannya, kita bisa menggunakan strip() atau print tanpa newline

Penyimpanan File

Setelah mengopen file dan menulis atau mengubah data dalam file, kita perlu menyimpannya kembali dengan menggunakan command close(), yaitu menutup lagi filenya.

```
# Ini adalah cara kedua untuk membuka file menggunakan "with" dan "as"
with open("file.txt", "w") as file:
    kalimat_ditambah = "tes"
    handle.write(kalimat_ditambah)
    file.close()
```

Jangan lupa untuk menggunakan mode "w" untuk menulis isi string ke dalam file

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

```
8,1.py > ...
1  # Membuka file pertama
2  with open("8,11.txt", "r", encoding="utf8") as filex:
3      # membaca seluruh baris
4      lines1 = filex.readlines()
5
6  # Membuka file kedua
7  with open("8,12.txt", "r", encoding="utf8") as filey:
8      lines2 = filey.readlines()
9
10 def compare(lines1, lines2):
11     # Mencari jumlah baris terbanyak
12     max_len = max(len(lines1), len(lines2))
13
14     # Loop dalam range baris terbanyak
15     for i in range(max_len):
16         # Jika jumlah baris lines1 lebih sedikit, maka baris yang kosong akan diisi dengan None
17         if i < len(lines1):
18             baris1 = lines1[i].strip()
19         else:
20             baris1 = None
21
22         # Jika jumlah baris lines2 lebih sedikit, maka baris yang kosong akan diisi dengan None
23         if i < len(lines2):
24             baris2 = lines2[i].strip()
25         else:
26             baris2 = None
27
28         # Mengecek perbaris apakah ada yang berbeda
29         if baris1 != baris2:
30             print(f'Perbedaan pada baris {i+1}:')
31             print(f'File 1: {baris1}')
32             print(f'File 2: {baris2}')
33
34     compare(lines1, lines2)
```

File 1

```
8,11.txt
1  abcd
2  abcd
```

File 2

≡	8,12.txt
1	abcd
2	abcde
3	abcd

Output:

```
Perbedaan pada baris 2:  
File 1: abcd  
File 2: abcde  
Perbedaan pada baris 3:  
File 1: None  
File 2: abcd
```


SOAL 2

```
8,2.py > ...
1  # Masukan nama file
2  nama_file = input("nama file1 : ")
3
4  # Mengecek apakah nama file benar
5  try :
6      # Membuka file
7      with open(nama_file, "r", encoding="utf8") as file:
8          # Pertanyaan dan jawaban sebagai list kosong
9          pertanyaan = []
10         jawaban = []
11
12         # Loop dalam file
13         for line in file:
14             # Dibagi menjadi list yang ada 2 isi yaitu pertanyaan dan jawaban
15             tanya_jawab = line.strip().split("||")
16
17             # Masing-masing dimasukkan ke dalam list tersendiri
18             pertanyaan.append(tanya_jawab[0])
19             jawaban.append(tanya_jawab[1])
20
21         # Mencari jumlah baris tertinggi
22         max_len = max(len(pertanyaan), len(jawaban))
23
24         # Loop dalam jangkauan max_len
25         for i in range(max_len):
26             # Print pertanyaan
27             print(pertanyaan[i])
28
29             # Meminta jawaban
30             user_input = input()
31
32             # mengecek apakah jawaban benar dan sesuai dengan pertanyaan
33             if user_input.strip().lower() == jawaban[i].strip().lower():
34                 print("Jawaban benar!")
35             else:
36                 print("Jawaban salah!")
37     except:
38         print("masukan file yang benar")
39
```

Output :

```
nama file1 : soal.txt
1+1 =
2
Jawaban benar!
Bendera Indonesia?
a
Jawaban salah!
Kota gudeg adalah:
YOGYAKARTA
Jawaban benar!
Komponen PC untuk penyimpanan file adalah...
Harddisk
Jawaban benar!
50 * 20 =
10000000
Jawaban salah!
```

Github : <https://github.com/SamuelN1508/PrakAIPro8.git>