



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231058
Nama Lengkap	Michael Chandra Mahanaim
Minggu ke / Materi	08 / Membaca & 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.

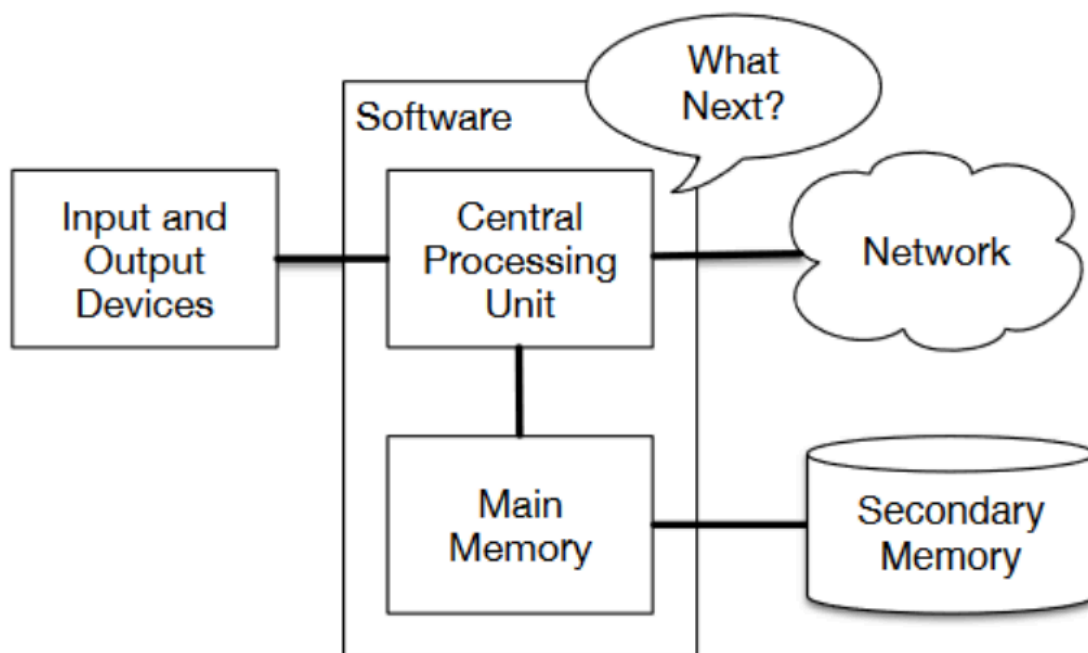
SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA

BAGIAN 1: MATERI MINGGU INI (40%)

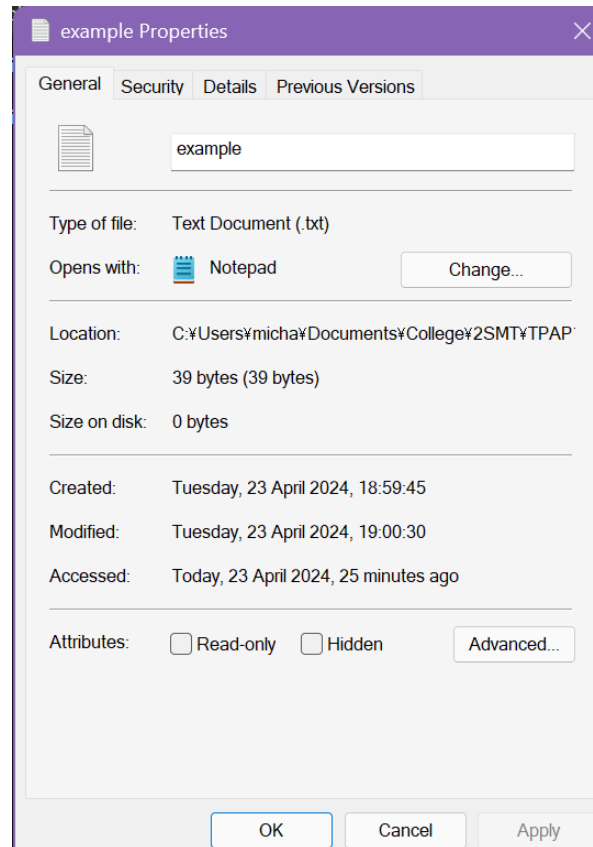
MATERI 1: Pengantar File

Program yang berjalan membutuhkan memory. Semua data program disimpan dalam memory dan ketika program selesai dijalankan dan dimatikan, maka semua data di dalam program tersebut juga akan hilang. Penyimpanan data di dalam memory bersifat tidak permanen (volatile). Karena itu, program yang menggunakan memory tidak akan dapat menyimpan data setelah program dimatikan. Maka harus digunakan penyimpanan tetap yaitu secondary memory.



Gambar Secondary Memory di Komputer

File disimpan pada secondary memory supaya file dapat digunakan untuk menyimpan data dari program dan tidak hilang jika komputer dimatikan. File bisa berupa file system, file program (binary), file multimedia, file teks, dan lain sebagainya. File memiliki property seperti nama file, ukuran, letak di harddisk, owner, hak akses, tanggal akses, dan lain-lain. Contoh sebuah file property adalah sebagai berikut:



MATERI 2: Pengaksesan File

Untuk mengakses file .txt dengan python, hanya perlu menggunakan beberapa kondisi terlebih dahulu. Kondisi tersebut termasuk:

1. Sebuah file .txt
2. Pastikan file itu ada di folder yang sama jika ingin menggunakan relative path
3. Gunakan Absolute path jika file .txt berada di folder lain

Contoh Absolute path: (C:\\user\\name\\file1\\folder2\\name.txt)

Jika sudah dipersiapkan, maka untuk mengakses file hanya memerlukan sintaks `open()` dengan format berikut:

```
handle = open("example.txt", "r")

print(handle)

handle.close()
```

Gambar contoh mengakses file menggunakan python

handle sebagai variabel untuk menyimpan `open("example.txt", "r")`. `example.txt` adalah nama file, sedangkan `r` adalah format penggunaan file. Jika file berhasil dibuka, maka output print akan tampil seperti berikut:

```
<_io.TextIOWrapper name='example.txt' mode='r' encoding='cp932'>
```

Gambar hasil open text yang berhasil

Jika nama file `.txt` yang dimasukkan tidak ada, maka output yang ditampilkan akan seperti berikut ini:

```
Traceback (most recent call last):
  File "c:\Users\micha\Documents\College\2SMT\TPAP1\tugaspralpro9\SrcCd.py", line 2, in <module>
    handle = open("none.txt", "r")
             ^^^^^^^^^^^^^^^^^^^^^
FileNotFoundError: [Errno 2] No such file or directory: 'none.txt'
```

Gambar error file tidak ditemukan

Ada beberapa format penggunaan file, yang utama hanya 2, yaitu:

- Read dengan tanda `r`
- Write dengan tanda `w`

Read hanya digunakan untuk membaca dan menggunakan file untuk perbandingan, manipulasi, atau file sebagai database. Write digunakan untuk menulis ulang atau memanipulasi isi dari file.

MATERI 3: Manipulasi File

Manipulasi file adalah menggunakan file untuk menghitung, membandingkan, ataupun sebagai database seperti menyimpan soal dan jawaban. Contoh pertama yaitu untuk menghitung jumlah baris yang ada seperti berikut:

```
count = 0
for line in handle:
    count += 1
print("Line count = ", count)
```

Gambar penghitung baris yang ada di sebuah `.txt` file

Kode di atas sudah diawali dengan `handle = open()` dan diakhiri dengan `handle.close()`. `for` loop dapat digunakan untuk looping sebanyak baris yang ada di `handle`. Dengan cara ini dapat diketahui berapa banyak baris yang ada di sebuah file `.txt`.

Ada juga manipulasi untuk mengetahui ukuran sebuah file dengan sintaks seperti berikut:

```
handle_read = handle.read()
print(f"Ukuran: {len(handle_read)} bytes")
```

Gambar contoh pengukuran besar file

Terakhir ada juga kode untuk menghitung dan menampilkan baris tertentu yang bermula dengan tulisan tertentu, seperti contoh di bawah ini:

```
count = 0
for line in handle:
    if line.startswith("Hello"):
        count += 1
        print(line)
print(count)
```

Gambar contoh pencarian baris yang bermula dengan suatu kata

Kode akan mencari semua baris yang bermula dengan "Hello" menggunakan .startswith(), jika ditemukan maka akan masuk IF dan menjalankan sesuai sintaks.

MATERI 4: Penyimpanan File

File dapat disimpan dengan menggunakan 'w' sebagai perintah open(). Contoh:

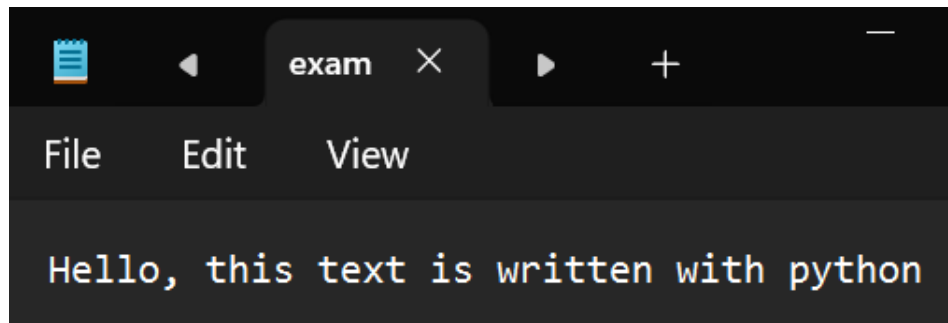
```
handle = open("example.txt", "w")

# print(handle)
handle.write("Hello, this text is written with python")

handle.close()
```

Contoh penulisan text file dengan python

Tulisan di atas akan disimpan dalam file example.txt dan muncul sebagai berikut:



Gambar hasil penyimpanan file yang sudah di write

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

A. Source Code

```
Question1.py x Question2.py SrcCd.py soal.txt Text2.txt
Question1.py > ...
1 with open("Text1.txt", "r") as file:
2     with open("Text2.txt", "r") as file2:
3         count = 1
4         handle = file.readlines()
5         handle2 = file2.readlines()
6         for i in range(len(handle)):
7             for j in range(len(handle2)):
8                 if i == j:
9                     if handle[i] == handle2[j]:
10                        print(f"Line {count} are the SAME\n")
11                    else:
12                        print(f"Line {count} are NOT SAME")
13                        print(f"File1 is {handle[i].strip()}")
14                        print(f"File2 is {handle2[j].strip()}\n")
15                count +=1
```

```
Text1.txt
1 Hello World
2 My Name
3 Age
4 Lived with
5
6 Don't Understand
7 Wait
```

```
Text2.txt
1 Hello World
2 My name
3 Age
4 Doesn't live
5
6 Don't Understand
7 Huh?
```

B. Output Result

```
PS C:\Users\micha\Docu
/College/2SMT/TPAP1/tu
Line 1 are the SAME

Line 2 are NOT SAME
File1 is My Name
File2 is My name

Line 3 are the SAME

Line 4 are NOT SAME
File1 is Lived with
File2 is Doesn't live

Line 5 are the SAME

Line 6 are the SAME

Line 7 are NOT SAME
File1 is Wait
File2 is Huh?
```

C. Explanation

Kode di atas berfungsi dengan membaca 2 file .txt(Text1.txt & Text2.txt) dan mencari perbedaan dari kedua file tersebut baris per baris. Penjelasan setiap langkah sebagai berikut:

1. Membuka kedua file .txt dengan with open() as file sebagai read
2. count = 1 digunakan untuk menghitung berada di baris ke berapa
3. handle = file.readlines() digunakan untuk kedua file supaya menjadi list yang dipisahkan baris per baris
4. for loop digunakan untuk looping sebanyak baris yang ada di list handle
5. Saat i == j, maka pada baris yang sama (1 == 1, 2==2, 3==3, etc)
6. Jika isi pada baris itu sama maka akan dikeluarkan output baris yang sama
7. Jika isi berbeda maka akan dikeluarkan output baris berbeda dan ditampilkan baris yang berbeda dari kedua file tersebut

SOAL 2

A. Source Code

```
Question1.py  Text1.txt  Question2.py x  SrcCd.py  Text2.txt
Question2.py > ...
1  filename = input("nama file1:")
2
3  try:
4      handle = open(filename, "r")
5      handle_data = handle.readlines()
6      for i in range(len(handle_data)):
7          line = handle_data[i]
8          split_line = line.split("||")
9          ans_chk = split_line[1].lower().strip()
10         print(split_line[0])
11         ans = input("Jawab: ")
12         ans = ans.lower()
13         if ans == ans_chk:
14             print("Jawaban benar!")
15         else:
16             print("Jawaban salah!")
17     handle.close()
18 except:
19     print("File Not Found")
```

B. Output Result

```
PS C:\Users\micha\Documents\College\2SMT\TPAP1
/College/2SMT/TPAP1/tugaspralpro9/Question2.py
nama file1:soal.txt
1+1 =
Jawab: 2
Jawaban benar!
Bendera Indonesia?
Jawab: merah putih
Jawaban benar!
Kota gudeg adalah:
Jawab: yogya
Jawaban salah!
Komponen PC untuk penyimpanan file adalah...
Jawab: HARDDISK
Jawaban benar!
50 * 20 =
Jawab: 1000
Jawaban benar!
```

C. Explanation

Kode di atas menggunakan file soal.txt sebagai penyimpan soal dan jawaban yang dipisahkan dengan || yang kemudian digunakan untuk memberi pertanyaan sesuai dengan isi file. Penjelasan setiap langkah sebagai berikut:

1. filename = input() adalah untuk meminta nama dari file soal & jawaban
2. try except digunakan jika nama file yang diberikan tidak ditemukan
3. handle = open() untuk membuka file sebagai read
4. handle.readlines() untuk memisahkan isi file baris per baris dalam bentuk list
5. for loop untuk looping sebanyak baris yang ada di handle_data
6. line = handle_data[i] untuk menyimpan baris terbaru ke dalam sebuah variabel
7. line.split(['|']) digunakan karena sebelah kiri adalah soal dan kanan adalah jawaban, maka split ini akan memisahkan pada bagian || dan dibuat ke dalam sebuah list 2 item
8. ans_chk digunakan untuk mengambil jawaban, di lowercase, dan di strip untuk menghilangkan \n
9. print(split_line[0]) untuk menampilkan soal
10. ans = input() untuk meminta jawaban dari user
11. ans.lower() untuk di lowercase supaya sama dengan ans_chk
12. If Else digunakan untuk mengecek jika jawaban user sama dengan kunci jawaban