Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	**71230982**
Nama Lengkap	**Rainie Fanita Chrisabel Hadisantoso**
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.

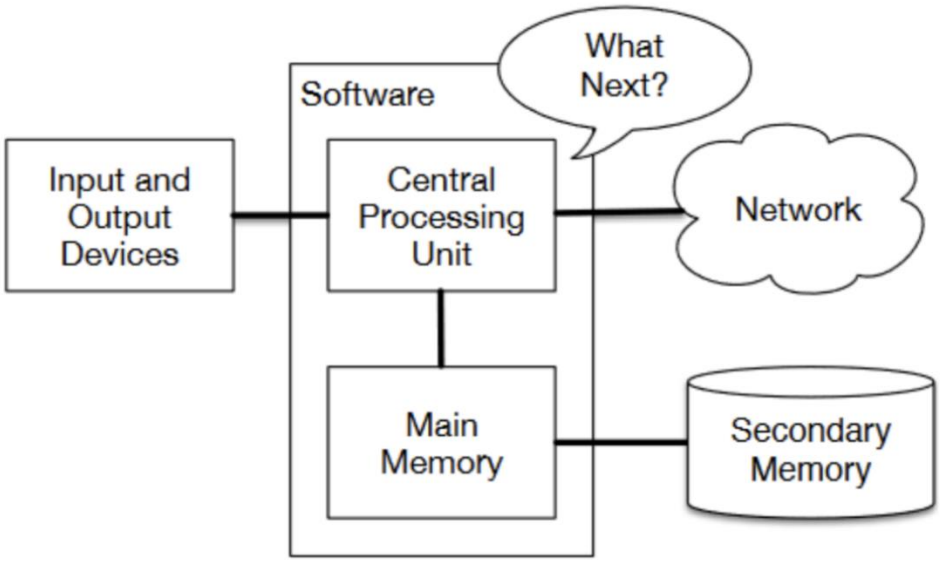
PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2024

MATERI : Membaca dan Menulis File

Tujuan dari pembelajaran ini adalah dapat menjelaskan tentang file, mengakses file melalui program dan memanipulasinya dengan perintah-perintah, serta menggunakan file untuk menyimpan data program.

Pengantar File

Dalam komputer ada 2 memori yang digunakan, memori primer dan memori sekunder. Memori primer adalah RAM dan ROM. Memori ini bersifat volatile atau tidak permanen. Yang kedua adalah memori sekunder, memori sekunder dapat teretak permanen dalam komputer (harddisk dan CD/DVD drive) dan ada yang dapat dipisahkan (USB dan blue ray DVD). Berbeda dengan memori primer, penyimpanan data dalam memori sekunder bersifat permanen atau non volatile.



Program dijalankan pada memori primer, sehingga semua data dalam program akan hilang ketika program selesai dijalankan dan dimatikan. Untuk menyimpan data secara permanen dari program perlu digunakan memori sekunder. Data dari program dapat disimpan berbentuk file. File adalah kumpulan bit yang tersimpan dalam memori sekunder. File dapat berupa teks, gambar, video dan lain-lain.

Pengaksesan File

Berikut beberapa langkah-langkah untuk mengakses file.

1. Menyiapkan file dan path file yang hendak diakses

   Sebagai contoh kita memiliki sebuah file bernama “nama.txt” dan berada dalam folder yang sama dengan file program, sehingga path tidak perlu digunakan. Namun jika tidak berada dalam folder yang sama maka perlu menggunakan path.

2. Membuka file

File dapat dibuka dengan cara disimpan dalam sebuah variabel secara langsung atau menggunakan barisan perintah untuk melakukannya.

- Secara langsung pada sebuah variabel. File dipanggil dengan perintah open() berisi nama file atau path serta metode akses yang digunakan.

```
file = open("nama.txt")
```

- Menggunakan with open untuk membuka dan untuk menyimpan dengan nama variabel pada program secara sementara. Dengan cara ini kita dapat mengolah file dalam perintah with dan olahan dapat digunakan oleh seluruh program.

```
with open ("nama.txt") as file:
```

- Ketika menggunakan path. Path digunakan ketika file yang hendak diakses tidak berada dalam folder yang sama dengan file program. Sehingga program perlu diberikan arahan di mana harus mengakses file.

```
path = "C:\Users\asus\belajar-git\nama.txt"

file = open (path)
```

Apabila file berhasil diakses, ketika variabel file di print maka akan menampilkan nama file, mode file dan encoding yang digunakan oleh file.

```
file = open("nama.txt", "r")

print(file)
```

Output:

```
PS C:\Users\asus\belajar-git> python -u "c:\Users\asus\belajar-git\belajar
 2\alpro8.2.py"
<_io.TextIOWrapper name='nama.txt' mode='r' encoding='cp1252'>
```

Namun jika file tidak dapat diakses oleh program maka akan menampilkan pesan error.

```
PS C:\Users\asus\belajar-git> python -u "c:\Users\asus\belajar-git\belajar python 2\a
Traceback (most recent call last):
  File "c:\Users\asus\belajar-git\belajar python 2\alpro8.2.py", line 1, in <module>
    file = open("error.txt", "r")
           ^^^^^^^^^^^^^^^^^^^^^^
FileNotFoundError: [Errno 2] No such file or directory: 'error.txt'
```

3. Memanipulasi file

Ada 3 metode mengolah file yang paling utama dan sering digunakan: read, write, append.

- Read: program hanya dapat membaca isi dari file, namun tidak dapat memanipulasi.
- Write: program dapat memanipulasi isi file, namun hanya berlaku sekali. Jika program write dijalankan lagi, maka data dari jalan sebelumnya akan hilang digantikan dengan yang baru.

- Append: program dapat memanipulasi file dengan menambahkan isi data baru. Data lama tidak akan hilang karena program hanya akan menambahkan data yang baru di bawah data lama.

Namun selain 3 itu ada banyak sekali metode mengolah file, ada yang menggunakan ketiga sebagai dasar ada yang beda. Namun untuk pembelajaran kali ini hanya digunakan ketiga metode tersebut saja.

4. Menutup file

Agar program berhenti mengakses file digunakan .close(). Biasanya digunakan ketika file akan diakses lagi dengan tujuan yang berbeda. Hal tersebut dilakukan agar program tidak bingung ketika melakukan perintah pada file yang sama. Seperti di refresh konsepnya.

```
file.close()
```

Manipulasi File

Program sederhana dapat memanipulasi file menggunakan metode read seperti membaca, mengambil bagian dari file, dan lain-lain. Untuk memanipulasi file pertama-tama program perlu mengakses file. Kemudian perlu juga dicantum kan metode akses file yang diinginkan: read, write, atau append. Apabila metode tidak dicantumkan maka program secara otomatis akan menggunakan metode read. Terakhir file ditutup untuk menghentikan program mengakses file.

1. Perulangan

Untuk mengakses dan membaca baris per baris dalam file dapat digunakan read() maupun perulangan. Read() membaca keseluruhan file sehingga dapat menyebabkan crash jika isi file sangat besar. Sehingga perulangan lebih disarankan apabila isi file sangat besar. Dengan menggunakan perulangan, memori yang digunakan lebih sedikit daripada menggunakan read(). Contoh menggunakan perulangan per baris seperti berikut, output program adalah jumlah baris dalam file.

```
file = open('nama.txt')
jumlah = 0
for baris in file:
    jumlah += 1
    print('Jumlah baris:', jumlah)
```

2. Metode string

Pada dasarnya file adalah sebuah string sehingga file juga dapat diakses menggunakan perintah-perintah untuk string biasa. Karena itu file juga memiliki indeks untuk tiap isinya, sehingga isi file dapat diakses menggunakan indeks.

```
file = open('nama.txt')
hasil = file.read()
print("Ukuran: " + len(hasil) + "bytes")
```

```
print("Huruf dari belakang sendiri mundur 16 huruf adalah: " + hasil[-16::1])
```

Contoh lain penggunaan metode string seperti startswith sebagai berikut:

```
file = open('nama.txt')

jumlah = 0

for baris in file:

    if baris.startswith("Date:") and jumlah <= 10:

        jumlah += 1

        print(baris)
```

Masih banyak lagi contoh penggunaan metode string maupun list maupun tuple untuk memanipulasi file.

Penyimpanan File

Program sederhana juga dapat mengganti, menghapus, menambah isi dari sebuah file menggunakan metode write maupun append. Meskipun sama-sama dapat menyimpan file ada perbedaan antara write dan append. Write dapat menulis ke dalam file, namun tidak menyimpan data yang sudah ada sebelumnya. Jadi write langsung menuliskan data baru di atas data lama, sehingga data lama hilang dan terhapus. Sedangkan append menambahkan data baru ke dalam file, sehingga data lama dalam file masih tersimpan. Write digunakan untuk data yang tidak tersimpan atau sekali pakai, seperti daftar pembelian. Append lebih digunakan untuk menyimpan data-data yang permanen seperti daftar nama dan daftar nilai.

1. Write:

   Menggunakan metode “w” dalam perintah open() setelah menuliskan path atau nama file.

   ```
   file = open('nama.txt', "w")

   tulisan = "teks ini akan dituliskan ke file\n"

   file.write(tulisan)

   file.close
   ```

   Hasilnya adalah, jika ada data lain sebelum ini, maka hanya akan ada tulisan “teks ini akan dituliskan ke file”

2. Append:

   Menggunakan metode “a” dalam perintah open() setelah menuliskan path atau nama file.

   ```
   file = open('nama.txt', "a")

   tulisan = "teks ini akan dituliskan ke file\n"
   ```

```
file.write(tulisan)

file.close
```

Output-nya jika ada data lain sebelumnya, maka data lama dan tulisan “teks ini akan dituliskan ke file” pada baris baru. Data masuk sebagai baris baru karena menggunakan \n untuk menggantikan baris.

Kegiatan Praktikum

a. Kasus 1

Contoh soal pertama untuk mencari url dalam sebuah file.

```
nama = input("nama file: ")

file = open(nama)

jumlah = 0

for baris in file:

    if baris.find("ac.uk"):

        jumlah += 1

        print("Web domain 'ac.uk' ditemukan di \"" + baris.strip() + "\"")

print("Jumlah: ",baris)
```

Nama file di input langsung oleh pengguna sehingga variabel nama digunakan untuk membuka isi file. Metode yang digunakan adalah read karena program hanya akan melihat isi file dan tidak mengubah-ubah isinya. Digunakan perulangan untuk melihat barisnya agar mengurangi penggunaan memori. Jika dalam baris ditemukan “ac.uk” dengan perintah find() , maka program akan menambah jumlah baris berisi url kemudian akan menampilkan di mana url ditemukan dan menampilkan isi barisnya menggunakan strip() untuk menghilangkan kelebihan spasi. Ketika baris selesai dibaca, program akan menampilkan berapa jumlah url yang digunakan.

b. Kasus 2

Program berikut digunakan untuk mencari judul-judul yang ada pada sebuah file kemudian menampilkannya dan menghitung jumlah judul.

```
nama = input("nama file: ")

file = open(nama)

jumlah = 0

for baris in file:

    if baris.startswith("Subject:"):

        jumlah += 1
```

```
        baris = baris.strip().title()
        print(baris)
print("Jumlah judul: ",baris)
```

Kurang lebih konsep membuka dan mencari per barisnya sama dengan kasus 1. Hanya saja percabangan hanya membaca jika ada “Subject:” di awal baris. Jika ya maka jumlah bertambah dan kemudian baris diambil, di hilangkan spasi tambahannya dan diubah dengan kapitalisasi title.

c. Soal 3

Program untuk menghitung besar byte sebuah file.

```
nama = input("nama file: ")
try:
    file = open(nama)
    total = 0
    for baris in file:
        total += len(baris)
        kb = total / 1000
    print("Ukuran: " + (kb) + " KB")
except:
    print("File tidak ditemukan!")
```

Program tidak menggunakan read() untuk mengurangi penggunaan memori, sehingga program menggunakan perulangan. Selain itu juga digunakan try except agar ketika file tidak ditemukan, program tidak menampilkan pesan error melainkan menampilkan “File tidak ditemukan!”. Ketika file ditemukan program akan membuka file. Tiap baris akan dihitung panjangnya kemudian total ditambahkan dengan jumlah panjang baris. Kemudian data diubah ke bentuk kb dengan dibagi 1000. Setelah perulangan selesai, program akan menampilkan jumlah total kb file.

LATIHAN 8

SOAL 1

Membuat sebuah fungsi untuk membandingkan 2 file dan mencari perbedaan keduanya secara perbaris

a. Source code

```
def banding(file1, file2):
    with open(file1, "r") as f1:
        ls1 = [line.rstrip() for line in f1]

    with open(file2, "r") as f2:
        ls2 = [line.rstrip() for line in f2]

    gabung = zip(ls1, ls2)
    no = 1
    for l1, l2 in gabung:
        if l1 != l2:
            print(f"Perbedaan di baris {no}:")
            print(f"File 1: {l1}")
            print(f"File 2: {l2}")
        no += 1

file1 = input("Masukkan nama file pertama : ")
file2 = input("Masukkan nama file kedua : ")
banding(file1, file2)
```

b. Output

```
PS C:\Users\asus\Downloads\File> python -u "c:\Users\asus\Downloads\F
8_71230982\Latihan_8.1.py"
Masukkan nama file pertama : file1.txt
Masukkan nama file kedua : file2.txt
Perbedaan di baris 1:
File 1: Haloo, world
File 2: Halo, world
Perbedaan di baris 2:
File 1: Hai, world
File 2: Haii, world
Perbedaan di baris 4:
File 1: Hai worldd
File 2: Hai world
```

c. Penjelasan

Pada fungsi bernama banding untuk memanggil, membaca, dan membandingkan perbedaan dari kedua file.

- With open (file 1, “r”) as f1 digunakan untuk membuka file dengan metode read atau membaca dan menyimpannya dengan nama f1. Kemudian file yang dibaca di pisahkan perbaris dengan

rstrip() untuk tiap baris dalam file pertama dan disimpan sebagai list dalam nama variabel ls1. Hal yang sama digunakan untuk file kedua.

- Kemudian kedua list baris dari kedua file dijadikan menjadi satu dengan zip() dalam bentuk tuple. Tuple adalah list dalam list, jadi ada list berisi baris pertama file 1 dan 2, list baris kedua dari kedua file, dan seterusnya semua dijadikan satu dalam sebuah list besar dengan nama gabung.
- Veriabel no untuk memberi nomor tiap baris.
- Dalam for ls1 dan ls2 dalam gabung, jadi for akan terus berulang selama baris dalam tuple gabung belum habis.
    a. Digunakan if untuk membandingkan line masing-masing file. Ketika tidak sama maka akan menampilkan dibaris berapa ada perbedaan menggunakan variabel no. Kemudian akan ditampilkan apa perbedaannya menggunakan baris yang dibandingkan.
    b. Kemudian nilai dalam variabel no akan ditambahkan 1
- Berulang terus hingga list-list dalam tuple gabung habis.

Diluar fungsi diberikan variabel file1 untuk input nama file pertama, dan file2 untuk nama file kedua. Terakhir fungsi banding dipanggil dengan kedua variabel sebagai parameter.

SOAL 2

Membuat sebuah program kuis berdasarkan file berisi pertanyaan dan kunci jawaban.

a. Source code

```
def kuis(file):
    with open(file, "r") as f:
        ls = [line.rstrip() for line in f]

    for l in ls:
        soal = l.split("||")
        print(f"Pertanyaan: {soal[0]}")
        kunci = soal[1].lower().strip()
        jawaban = input("Jawab: ").lower().strip()

        if kunci == jawaban:
            print("Jawaban benar!")
        else:
            print("Jawaban salah!")

file = input("Sebutkan file soal : ")
kuis(file)
```

b. Output

```
PS C:\Users\asus\Downloads\File> python -u "c:\Users\asus\Downloads\File\
8_71230982\Latihan_8.2.py"
Sebutkan file soal : kunci_soal.txt
Pertanyaan: 1+1 =
Jawab: 2
Jawaban benar!
Pertanyaan: Bendera Indonesia?
Jawab: merah putih
Jawaban benar!
Pertanyaan: Kota gudeg adalah:
Jawab: yogyakarta
Jawaban benar!
Pertanyaan: Komponen PC untuk penyimpanan file adalah...
Jawab: harddisk
Jawaban benar!
Pertanyaan: 50 * 20 =
Jawab: 20000000000000
Jawaban salah!
```

c. Penjelasan

Fungsi kuis digunakan untuk membuka dan membaca file berisi pertanyaan dan kunci jawaban, menampilkan soal pertanyaan, memasukkan jawaban dari user, dan menampilkan apakah jawaban benar atau salah.

- With open (file, “r”) as f digunakan untuk membuka file dengan metode read atau membaca dan menyimpannya dengan nama f. Kemudian file yang dibaca di pisahkan per baris dengan rstrip() untuk tiap baris dalam file dan disimpan sebagai list dalam nama variabel ls.
- Perulangan untuk tiap baris dalam list baris file.
    a. Pertama baris dipisah menjadi dua, pertama soal dan kedua kunci jawaban dengan split() dan “||” sebagai tanda pemisah. Soal menjadi bagian pertama memiliki indeks 0 dan kunci jawaban bagian kedua memiliki indeks 1.
    b. Soal di print dengan memanggil list soal[0] indeks pertama atau 0.
    c. Kemudian kunci jawaban disimpan dalam variabel kunci dengan soal[1]. Kunci di lower() agar dalam bentuk huruf kecil dan di strip() untuk menghilangkan spasi tambahan.
    d. Jawaban dari user disimpan dalam variabel kunci dengan input(). Input juga di lower() untuk mengubah jawaban dalam huruf kecil dan strip() untuk menghilangkan spasi tambahan.
    e. Percabangan if kunci dan jawaban user sama akan ditampilkan “Jawaban benar!”. Jika tidak sama maka akan menampilkan “Jawaban salah!”

Digunakan input() untuk memasukkan nama file dari user dalam variabel file. Kemudian memanggil fungsi kuis dengan parameter file.