



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230987
Nama Lengkap	Prastha Pradipta Purusa
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

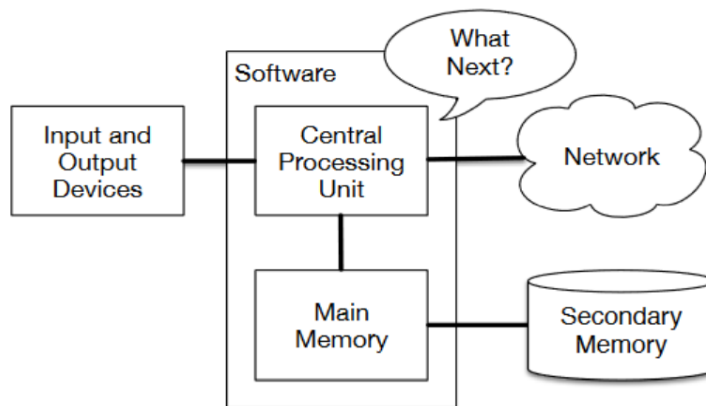
BAGIAN 1: MATERI MINGGU INI (40%)

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.

Pengantar File

Program yang menggunakan memory primer menyimpan semua datanya di dalam memory komputer. Namun, data tersebut bersifat tidak permanen (volatile), sehingga saat program dimatikan, data di dalamnya juga hilang. Sifat tersebut membuat program tidak dapat menyimpan data setelah dimatikan. Secondary memory digunakan untuk menyimpan data pada program.

Contoh secondary memory:



File disimpan dalam secondary memory untuk mempertahankan data walaupun komputer dimatikan. File terdiri dari bit-bit data yang tersimpan secara permanen. File bisa berupa berbagai jenis seperti file sistem, program (binary), multimedia, atau teks. Setiap file memiliki properti seperti nama, ukuran, letak di harddisk, owner, hak akses, dan tanggal akses.

Pengaksesan File

Langkah-langkah untuk mengakses file:

1. Menyiapkan file dan path yang akan di akses
2. Buka file
3. Lakukan sesuatu dengan file tersebut seperti ditampilkan isinya (read) atau mengubah isinya (write)
4. Tutup file

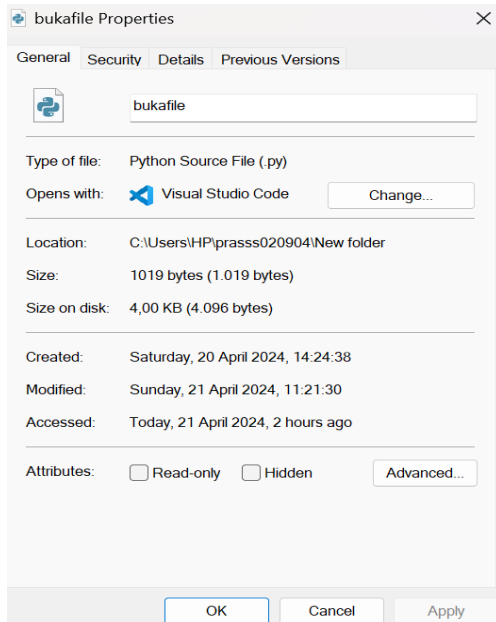
Contoh penggunaan di Python:

```
1 bukafile = open("file1.txt")
2 print(bukafile)
```

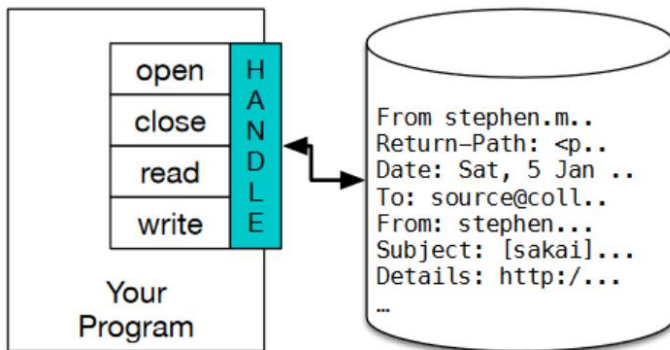
Output:

```
<_io.TextIOWrapper name='file1.txt' mode='r' encoding='cp1252'>
```

Contoh file property:



Contoh ilustrasi handle file:



Hasilnya adalah tampilan nama file, mode (r = read), dan encoding yang digunakan yaitu cp1252 yang merupakan system io di Python. Pembacaan teks di Python membaca baris demi baris di setiap string yang ditemukan sampai End of File (EOF).

Manipulasi File

Untuk memanipulasi file kita harus membaca file terlebih dahulu.

Cara membaca file pada Python adalah:

1. Menyiapkan file
2. Buka file

3. Jumlah kata di file

4. Tutup file

Berikut adalah contoh membuat program untuk menghitung ada berapa kata:

```
1  nama_file = "file1.txt"
2  bukafile = open(nama_file, 'r')
3  isi_file = bukafile.read()
4  bukafile.close()
5  kata = isi_file.split()
6  jumlah_kata = len(kata)
7  print("Jumlah kata dalam file", nama_file, "adalah:", jumlah_kata)
```

Output:

```
[Running] python -u "c:\Users\HP\prasss020904\New folder\bukafile.py"
Jumlah kata dalam file file1.txt adalah: 1
```

Pertama kita buat variable nama_file untuk nama file yang mau dibuka. Variable bukafile digunakan untuk membuka file dan 'r' read untuk membaca file. Variable isi_file digunakan untuk membaca bukafile dengan read(). Untuk bukafile.close() digunakan untuk menutup file setelah membaca. Untuk kata di split() untuk memisahkan kata. Menghitung jumlah kata dengan len. Untuk memunculkan output menggunakan print.

Contoh menampilkan jumlah file teks dalam file dengan len:

```
file = open('file1.txt')
baca = file.read()
print(len(baca))
```

```
1  hai!
2  hai?
```

Output:

```
[Running] python -u "c:\Users\HP\prasss020904\New folder\bukafile.py"
9
```

Kita membuka file1.txt terlebih dahulu kemudian membaca file dengan variable baca = file.read(). Fungsi len() digunakan untuk menghitung ada berapa jumlah kata yang ada di file tersebut dan ukuran untuk 1 karakter = 1 byte maka terdapat 9 byte termasuk line 2.

Perintah read() pada file sangat boros memory, lebih baik menggunakan looping untuk memori yang lebih hemat.

Penyimpanan File

Buat variable bukafile untuk "file1.txt" dengan penulisan ('w'), menulis string "hai" ke dalam file, dan kemudian ditutup. Membuat atau mengganti isi file "file1.txt" dengan string "hai". Dengan kode ini dan dieksekusi isi dari "file1.txt" akan menjadi "hai". Jika file sudah ada ASDisi sebelumnya, isi sebelumnya akan digantikan dengan string "hai". Jika file tersebut belum ada, maka file baru akan dibuat dengan isi "hai".

```
1 bukafile = open("file1.txt","w")
2 tulisan = "hai"
3 bukafile.write(tulisan)
4 bukafile.close()
```

Output:

```
1 hai
```

Kegiatan Praktikum

Kita akan mencoba membuat program dimana program itu akan mencari semua baris yang memiliki angka 1 menggunakan find() untuk mencarinya dan menampilkannya. Gunakan jumlah untuk menampilkan setiap baris yang ditemukan angka1.

```
bukafile = open('file2.txt')
jumlah = 0
for baris in bukafile:
    if baris.find("1") != -1:
        jumlah += 1
        print("1:\n" + baris.strip() + "\n")
print(jumlah)
```

Output:

```
[Running] python -u "c:\Users\HP\prasss020904\New folder\bukafile.py"
1:"aku1"
1:"dia1"
1:"dia sama yang lain 1"
3
```

Karakter \ digunakan untuk menyertakan tanda kutip ganda (") di dalam string yang dicetak tanpa menutup string.

Program kedua adalah program untuk menampilkan baris di file yang diawali kata "aku" dengan menggunakan command startswith() dan cara untuk menampilkan jumlahnya dengan variabel jumlah untuk jumlah baris yang ditemukan.

```
bukafile = open("file2.txt")
jumlah = 0
for baris in bukafile:
    if baris.startswith("aku"):
        jumlah += 1
        print("aku:" + baris.strip() + "\n")
print(jumlah)
```

Output:

```
[Running] python -u "c:\Users\h...
aku:"aku1"
aku:"aku2"
aku:"aku sama dia"
3
```

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

jupyter Prastha Pradipta Purusa (71230987) Laprak 8 Last Checkpoint: beberapa detik yang

File Edit View Insert Cell Kernel Widgets Help

Run

```
In [1]: file1 = 'file1.txt'
file2 = 'file2.txt'

f1 = open(file1, 'r')
f2 = open(file2, 'r')

lines1 = f1.readlines()
lines2 = f2.readlines()

f1.close()
f2.close()

isi_line = max(len(lines1), len(lines2))

for i in range(isi_line):
    if i < len(lines1) and i < len(lines2):
        line1 = lines1[i].strip()
        line2 = lines2[i].strip()
        if line1 != line2:
            print(f'Baris {i+1}:')
            print(f'File 1: {line1}')
            print(f'File 2: {line2}')

    elif i < len(lines1):
        print(f'Baris {i+1}:')
        print(f'File 1: {lines1[i].strip()}')
        print(f'File 2: {lines2[i].strip()}')

    elif i < len(lines2):
        print(f'Baris {i+1}:')
        print(f'File 1: {lines1[i].strip()}')
        print(f'File 2: {lines2[i].strip()}')
```

```
[Running] python -u "c:
Baris 1:
File 1: A
File 2: o
Baris 2:
File 1: kjpkohkphok
File 2: skdkasm
Baris 3:
File 1: hkofk
File 2: samdklasnd
```

Line 1: Membuat variable file1 untuk file1.txt

Line2: Membuat variable file2 untuk file2.txt

Line3: Membuat variable f1 untuk membuka file1 dengan command 'r' yaitu read

Line4: Membuat variable f2 untuk membuka file2 dengan command 'r' yaitu read

Line5: Membuat variable lines1 untuk membaca lines di file1 dengan readlines()

Line6: Membuat variable lines2 untuk membaca lines di file2 dengan readlines()

Line7: f1.close() digunakan untuk menutup file setelah membaca file1

Line8: f2.close() digunakan untuk menutup file setelah membaca file2

Line9: variable isi_line digunakan untuk membaca semua line yang ada di file dan menggunakan len untuk menghitungnya dan max untuk semua line.

Line10: Loop for ini akan berjalan sebanyak isi_line yaitu panjang maksimum dari kedua file.

Line11: Memastikan bahwa perbandingan dilakukan pada baris yang ada di kedua file dengan memastikan bahwa indeks masih dalam rentang panjang kedua file yang dibandingkan.

Line12: Menghapus karakter whitespace di awal dan akhir setiap baris dan menyimpan di line1

Line13: Menghapus karakter whitespace di awal dan akhir setiap baris dan menyimpan di line2

Line14: kondisi jika line1 tidak sama dengan line2

Line15: Menambahkan index sebanyak 1 karena mulai dari 0

Line16: Memunculkan output isi baris dari file pertama (line1) yang telah di strip()

Line17: Memunculkan output isi baris dari file kedua(line2) yang telah di strip()

Line18: Jika kita masih dalam rentang indeks yang valid untuk lines1 tetapi tidak untuk lines2, maka akan mencetak baris dari lines1 saja

Line19: Menambahkan index sebanyak 1 karena mulai dari 0

Line20: Memunculkan output dari isi dari baris i dari file pertama (file1). `lines1[i].strip()` mengambil baris ke-i dari `lines1` kemudian menghapus karakter whitespace di awal dan akhir baris tersebut sebelum print dan proses perbandingan baris antara kedua file.

Line21: Memunculkan output isi dari baris i dari file kedua (file2). `lines2[i].strip()` mengambil baris ke-i dari `lines2` lalu menghapus karakter whitespace di awal dan akhir baris.

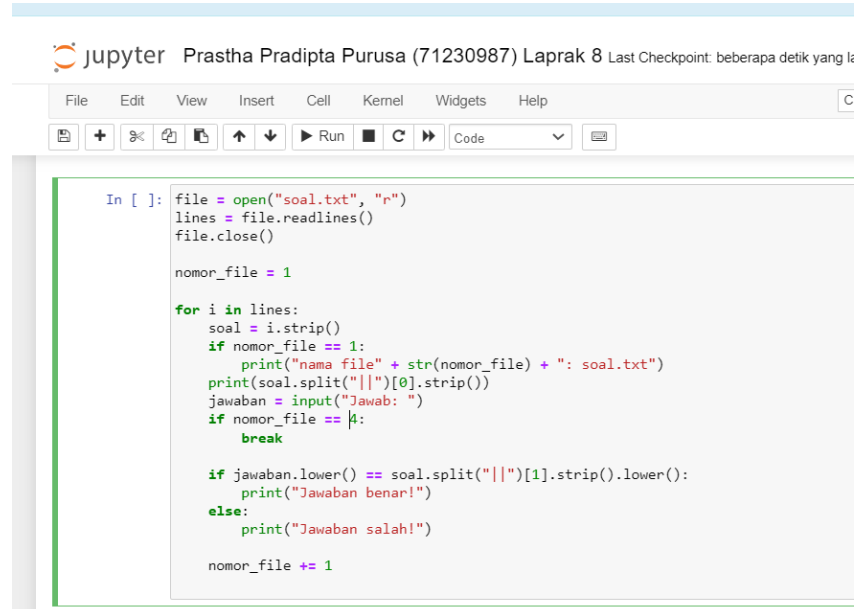
Line22: Jika kita masih dalam rentang indeks yang valid untuk `lines2` tetapi tidak untuk `lines1`, maka akan mencetak baris dari `lines2` saja

Line23: Menambahkan index sebanyak 1 karena dimulai dari 0

Line24: Memunculkan output isi dari baris i dari file kedua (file1). `Lines1[i].strip()` mengambil baris ke-i dari `lines1` lalu menghapus karakter whitespace di awal dan akhir baris.

Line25: Memunculkan output dari isi dari baris i dari file pertama (file2). `lines2[i].strip()` mengambil baris ke-i dari `lines2` kemudian menghapus karakter whitespace di awal dan akhir baris tersebut sebelum print dan proses perbandingan baris antara kedua file.

SOAL 2



The screenshot shows a Jupyter Notebook interface with a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The code cell contains the following Python script:

```
In [ ]: file = open("soal.txt", "r")
lines = file.readlines()
file.close()

nomor_file = 1

for i in lines:
    soal = i.strip()
    if nomor_file == 1:
        print("nama file" + str(nomor_file) + ": soal.txt")
        print(soal.split("|")[0].strip())
        jawaban = input("Jawab: ")
        if nomor_file == 4:
            break

    if jawaban.lower() == soal.split("|")[1].strip().lower():
        print("Jawaban benar!")
    else:
        print("Jawaban salah!")

    nomor_file += 1
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GIT LENS
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
PS C:\Users\HP\prasss020904\Laprak8-71230987>
```

Line1: membuat variabel file dan membuka file soal.txt dan r untuk read

Line2: membaca file tiap line dengan readlines()

Line3: menutup file setelah membaca file

Line4: menginisiasi sebuah variabel nomor_file untuk melacak nomor file.

Line5: Memulai looping melalui setiap baris dalam daftar lines.

Line6: Menghapus karakter spasi (whitespace) di awal dan di akhir setiap baris dan disimpan di variabel soal.

Line7: Kondisi jika nomor_file sama dengan 1

Line8: Jika nomor file adalah 1 akan mencetak informasi tentang nama file yang sedang diproses.

Line9: Mengeluarkan output pertanyaan yang terdapat dalam baris. Pertanyaan diambil dengan membagi baris menggunakan pemisah "|" dan mengambil elemen pertama dari hasilnya.

Line10: untuk menjawab pertanyaan yang ditampilkan

Line11: kondisi jika nomor_file sama dengan 4

Line12: Jika nomor_file sama dengan 4 maka program berhenti

Line13: Memeriksa apakah jawaban benar dan membandingkan jawaban yang dimasukkan dengan jawaban yang diambil dari baris dalam file "soal.txt".

Line14: jika jawaban benar akan muncul output "Jawaban benar!"

Line15: untuk jawaban salah:

Line16: Jika jawaban salah maka akan muncul output "Jawaban salah!"

Line17: Menambahkan 1 ke nomor file untuk memproses baris berikutnya dalam file dan mempersiapkan program untuk memproses pertanyaan berikutnya dalam file "soal.txt".

Link github:

<https://github.com/prasss020904/Laprak8-71230987.git>