

Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024



NIM	71200645
Nama Lengkap	Edith Felicia Putri
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
2023

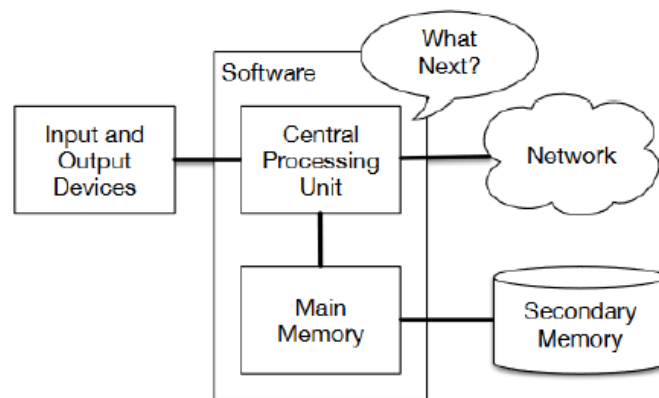
BAGIAN 1: MATERI MINGGU INI (40%)

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.

MATERI 1

PENGANTAR FILE

Program yang berjalan membutuhkan **memori primer** di dalam komputer. Semua data yang ada di program tersebut disimpan dalam memori dan ketika program diakhiri, sehingga semua data di dalam program tersebut juga ikut hilang. Penyimpanan data di dalam memori bersifat tidak permanen (*volatile*), untuk kasus program yang menggunakan memori primer.



Untuk dapat menyimpan data pada program digunakan **penyimpanan tetap**, yaitu *secondary memory*, sehingga file dapat digunakan untuk menyimpan data dari program, tidak hanya data tidak akan hilang walaupun komputer dimatikan, tetapi data juga dapat disalin ke USB dan dipindahkan ke komputer lain.

File pada dasarnya adalah bit-bit data yang disimpan di dalam *secondary memory* secara permanen, berupa **kumpulan informasi yang saling berelasi satu sama lain** sebagai satu kesatuan. File memiliki properti seperti nama file, ukuran, letak di hard-disk, *owner*, hak akses, tanggal akses, dan lain-lain.

Pada Python, file hanya dikelompokkan menjadi dua tipe:

1. File teks: Berisi teks. Setiap baris teks memiliki EOL (*End of Line*).
Contoh: TXT, MD, CSV, JSON, dan sebagainya.
2. File program (binary): Bukan berisi teks dan hanya bisa diproses oleh program tertentu yang memahami strukturnya.
Contoh: EXE, JPG, MKV, M4A, 3GP, dan sebagainya.
3. File bisa berupa file system, file multimedia, dan sebagainya.

source: <https://www.petanikode.com/python-file/>

PENGAKSESAN FILE

- PRINT FILE

Teks files terdiri dari rentetan baris yang dipisahkan dengan karakter *newline* (). Untuk menampilkan data menggunakan fungsi *print()*.

```
print("Yellow all!")
```

- OPEN FILE

Membuka *file* sesuai dengan *path*-nya, kemudian menampilkan isinya dengan fungsi *read()* atau mengubah atau menulis dalam *file*-nya dengan fungsi *write()*.

SOURCE CODE:

```
fi = open('End Times.txt')  
print(fi)
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_9> & C:/Users/  
HP/AppData/Local/Programs/Python/Python  
311/python.exe d:/71200645_PrakAlpro_9/  
Modul/modul.py  
<_io.TextIOWrapper name='End Times.txt'  
mode='r' encoding='cp1252'>  
PS D:\71200645_PrakAlpro_9>
```

- OPEN AND READ FILE

Membuka *path* dalam mode *read* dan menyimpannya dalam variabel *fi*.

```
fi = open(filename, "r")
```

- OPEN FILE WITH INPUT

```
filename = input("File yang ingin dibuka: ")  
file_read = open(filename)
```

MEMBACA KONTEN DARI FILE:

1. Mode operasi *read()* ini mengembalikan seluruh teks dan bisa menentukan berapa banyak karakter yang ingin dikembalikan. Ada tiga cara dalam melakukan pembacaan file: Mulai dari *read([n])*, *readline([n])*, *readlines()*. Mode operasi ini biasanya dipakai dalam membaca data pada file yang sama. Sebelum membaca file, ada dua jenis file yang bisa ditangani untuk melakukan operasi *read()* yaitu *normal text files* dan *binary files*.

```
fi.read(20) # Membaca 20 karakter  
fi.read() # Membaca semua karakter  
fi.readline() # Mengembalikan seluruh baris  
fi.readlines() # Mengembalikan sebuah list dari seluruh baris
```

2. Fungsi *readline()* membaca satu baris file dan mengembalikannya dalam bentuk string. Dibutuhkan parameter *n*, yang menentukan jumlah maksimum *byte* yang akan dibaca. Namun, tidak membaca lebih dari satu baris, meskipun *n* melebihi panjang baris, sehingga

efisien saat membaca file besar karena alih-alih mengambil semua data sekaligus, ini mengambil baris demi baris. `readline()` mengembalikan baris berikutnya dari file yang berisi karakter baris baru pada akhirnya. Juga, jika akhir file tercapai, itu akan mengembalikan string kosong.

3. `Readlines()` untuk membaca semua baris sekaligus, kemudian mengembalikannya sebagai setiap baris elemen string dalam daftar. Digunakan untuk file kecil, karena membaca seluruh konten file ke memori, lalu membaginya menjadi beberapa baris terpisah.

Setiap baris dalam proses dibaca, dengan jarak *new-line* di akhir line karena fungsi `print(i)`.

```
filename = "End Times.txt"
fi = open(filename, "r")

contents = fi.readlines()
for i in contents:
    print(i)

fi.close()
```

Membuat setiap kata pada setiap line terpisah karena fungsi `split()`.

SOURCE CODE:

```
filename = "End Times.txt"
fi = open(filename, "r")

contents = fi.readlines()
for i in contents:
    reading = i.split()
    print(reading)

fi.close()
```

OUTPUT:

```
['Teman', 'semuanya,', 'apa', 'kabar']
['Kalian', 'hadir', 'di', 'sini', 'untuk', 'mendengarkan',
', 'kisah', 'menjelang', 'hari', 'terakhir']
['Semua', 'lutut', 'akan', 'berlutut', 'dan', 'semua', 'lidah',
'mengaku', 'Yesus', 'Kristus', 'adalah', 'Tuhan', 'dan', 'Raja']
['Aku', 'tidak', 'sabar', 'akan', 'kedatangan', 'hari', 'itu']
['Untuk', 'sekarang,', 'hari', 'ini', 'adalah', 'hari', 'keselamatan', 'itu']
['Carilah', 'Dia,', 'selama', 'Dia', 'masih', 'dapat', 'ditemukan',
'dan', 'penghakiman', 'belum', 'tertimpa', 'atasmu.']
```

Mengambil string di dalam list dengan index 0.

```
print(reading[0])
```

OUTPUT:

```
PS D:\71200645_PrakAlpro_9> & C:/Users/HP/AppData/Local/
Programs/Python/Python311/python.exe d:/71200645_PrakAlp
ro_9/Modul/modul.py
Teman
Kalian
Semua
Aku
Untuk
Carilah
```

4. WRITING A FILE

Membuka *path* dengan mode *write* dan menyimpannya dalam variabel *outfile*.

```
outfile = open(filename, 'w')
```

Menulis satu baris teks di dalam *file*.

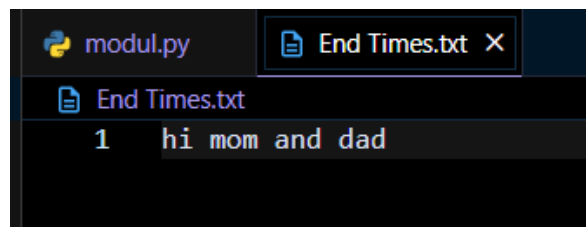
```
outfile.write(line+ '\n')
```

Menulis setiap elemen di dalam *list* ke dalam *file*, sehingga terjadi *overwriting*.

SOURCE CODE:

```
lines = ["hi ", "mom ", "and ", "dad"]
outfile.writelines(lines)
```

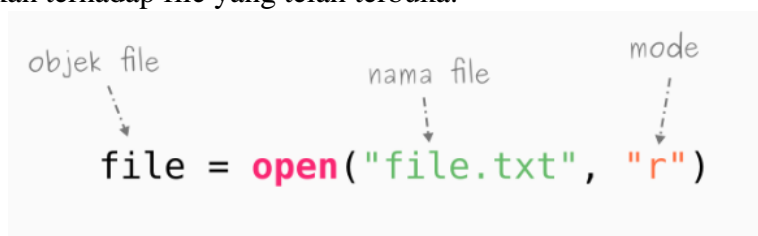
OUTPUT:



MATERI 2

MANIPULASI FILE

Untuk dapat memanipulasi file, harus dimulai dari membaca file. Fungsi `open()` menerima 2 parameter, yaitu nama dari file yang ingin dibuka/dibuat dan mode yang berkaitan dengan aksi yang ingin dilakukan terhadap file yang telah terbuka.



PENYIMPANAN FILE

```
file = open('Modul/Keselamatan.txt', 'w')
isi = "Hanya dengan darah Yesus\n"
file.write(isi)

file.close()
```

Perintah `file.close()` untuk menutup file yang telah dibuka. Cara membuka file dengan `with open` tidak memerlukan perintah tutup file karena secara otomatis menutup file yang dibuka setelah selesai prosesnya.

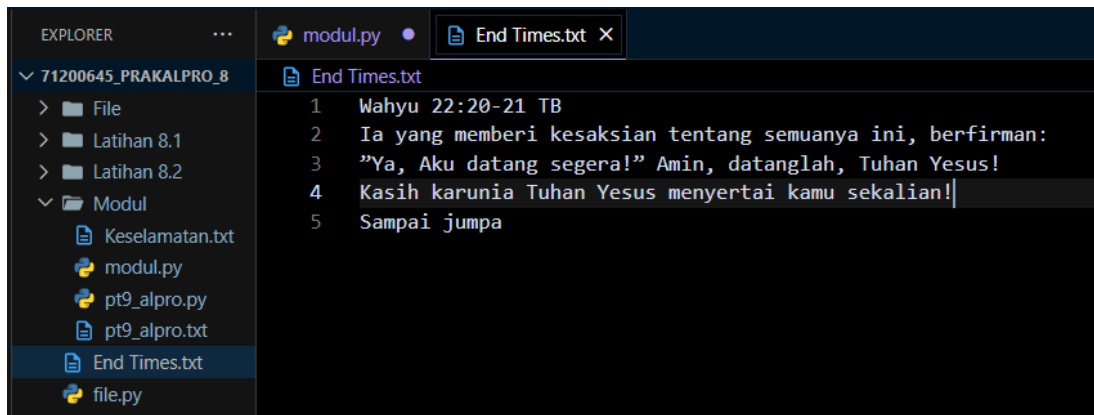
MODE-MODE UNTUK MEMBACA FILE:

1. `w`: write, digunakan untuk menulis ke dalam sebuah berkas teks. Jika berkas tidak tersedia, maka Python akan secara otomatis membuat sebuah berkas baru dengan nama yang telah dispesifikasikan. Saat menulis dengan menggunakan mode ini, jika file semula tidak kosong, maka isi yang sebelumnya terdapat di dalam berkas akan terhapus.
2. `a`: append, digunakan untuk menambahkan isi dari sebuah berkas teks. Mode ini juga akan membuat sebuah berkas teks baru dengan nama yang telah dispesifikasikan jika berkas teks tidak tersedia.
3. `w+`: write+, digunakan untuk membaca ataupun menulis isi dari sebuah berkas teks.
4. `a+`: append+, digunakan untuk membaca ataupun menambahkan isi dari sebuah berkas teks.

Menambahkan data ke dalam file.

```
my_file = open("End Times.txt", "a+")
my_file.write("\nSampai jumpa")
```

OUTPUT:



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

Latihan 8.1 Buatlah sebuah program yang dapat membandingkan 2 buah file teks dan kemudian menampilkan perbedaan antar kedua teks per barisnya jika ada perbedaan! ■

Pengerjaan soal dilakukan dengan memasukkan path kedua file ke dalam dua variabel, lalu membukanya dengan fungsi open, dengan fungsi readlines() dilakukan pembacaan baris per baris di dalam file. Dibuat variabel max_lines untuk mengetahui maksimal baris yang ada di kedua file yang digunakan untuk fungsi for loop. Jika baris i berisi teks pada len di kedua file, maka kedua baris akan dibandingkan. Jika baris i file1 != baris i file2, maka perbedaan itu ditampilkan. Jika i hanya sesuai untuk file pertama, maka menampilkan sisa baris dari file pertama, dan sebaliknya.

SOURCE CODE:

```
def perbandingan_file():
    file_1 = "Latihan 8.1/teks_1.txt"
    file_2 = "Latihan 8.1/teks_2.txt"
    with open(file_1, 'r') as f1, open(file_2, 'r') as f2:
        f1_lines = f1.readlines()
        f2_lines = f2.readlines()

    max_lines = max(len(f1_lines), len(f2_lines))

    for i in range(max_lines):
        if i < len(f1_lines) and i < len(f2_lines):
            if f1_lines[i] != f2_lines[i]:
                print(f"Perbedaan di baris ke-{i + 1}:")
                print(f"File 1: {f1_lines[i].strip()}")
                print(f"File 2: {f2_lines[i].strip()}")
            elif i < len(f1_lines):
                print(f"Perbedaan di baris ke-{i + 1}:")
                print(f"File 1: {f1_lines[i].strip()}")
            elif i < len(f2_lines):
                print(f"Perbedaan di baris ke-{i + 1}:")
                print(f"File 2: {f2_lines[i].strip()}")

perbandingan_file()
```

OUTPUT:

```
Perbedaan di baris ke-1:
File 1: Teman semuanya, apa kabar
File 2: Teman semuanya, aku terhibur s
ekarang

Perbedaan di baris ke-6:
File 1: Carilah Dia, selama Dia masih
dapat ditemukan dan penghakiman belum
tertimpa atasmu.
File 2: Ikutlah Dia yang menyebut diri
-Nya, jalan, kebenaran, dan hidup
```

SOAL 2

Latihan 8.2 Buatlah sebuah program untuk menampilkan soal sederhana yang diambil dari file teks soal.txt yang memiliki format sebagai berikut:

```
1+1 = || 2
Bendera Indonesia? || Merah Putih
Kota gudeg adalah: || Yogyakarta
Komponen PC untuk penyimpanan file adalah... || harddisk
50 * 20 = || 1000
```

Dari soal tersebut tampilkan sbb:

```
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!
```

Pengerjaan soal dilakukan dengan membuka file dalam mode *read*, lalu menggunakan fungsi *readlines()* untuk membaca tiap barisnya. Kemudian menggunakan for loop pada setiap baris dari teks untuk memisahkan bagian soal dan jawaban menggunakan fungsi *split()* dengan pemisah "||". Fungsi *strip()* untuk menghapus spasi di awal dan akhir setiap baris. Lalu jawaban dari pengguna didapatkan dengan fungsi *input()* dan dilakukan penghapusan spasi, serta mengubah input-an menjadi huruf kecil untuk menyamakan atau mempermudah perbandingan dengan jawaban. Jika jawaban pengguna sesuai, maka benar, dan sebaliknya salah.

SOURCE CODE:

```
def tanya_jawab():
    with open("Latihan 8.2/soal.txt", "r") as file:
        hasil = file.readlines()

        for i in hasil:
            soal, jawaban = i.strip().split("||")
```



```
print(soal.strip())
jawaban_pengguna = input("jawab: ").strip().lower()

if jawaban_pengguna == jawaban.strip().lower():
    print("Jawaban benar!")
else:
    print("Jawaban salah!")

tanya_jawab()
```

OUTPUT:

```
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
```

Link Github:

https://github.com/EdithFelicia/71200645_Guided/tree/main/71200645_PrakAlpro_8