Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	**71231051**
Nama Lengkap	**Amida A Ronsumbre**
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.

MATERI 1

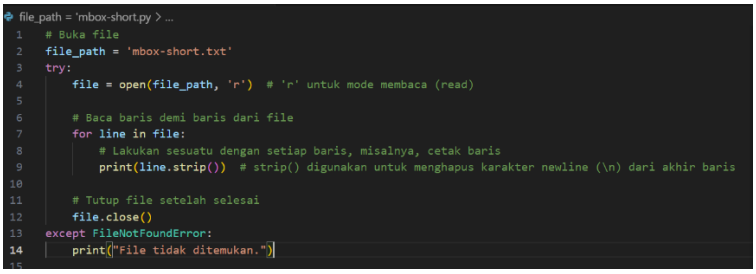
Penjelasan materi 1

- 8.3.1 Pengantar File

File disimpan pada secondary memory, dan berbeda dengan data dalam memory primer, file-file tersebut bersifat permanen dan tidak akan hilang ketika komputer dimatikan. Setiap file memiliki property seperti nama file, ukuran, letak di harddisk, owner, hak akses, tanggal akses, dan lain-lain. Property sebuah file sangat penting untuk mengelola dan mengakses file tersebut. Contoh property
Misalnya, nama file, ukuran file, letak di harddisk, owner file, hak akses , dan tanggal akses merupakan beberapa property yang umum terkait dengan sebuah file. Dengan menggunakan secondary memory, program dapat menyimpan data dalam file secara permanen dan dapat diakses kembali setelah program dimatikan atau komputer dimatikan.

- 8.3.2 Pengaksesan File

Pastikan path yang Anda berikan sesuai dengan lokasi sebenarnya dari file tersebut. Dalam banyak bahasa pemrograman, seperti Python, Anda dapat menggunakan fungsi `open` untuk membuka file. Ini penting untuk memastikan bahwa sumber daya sistem yang terkait dengan file dibebaskan dan tidak terjadi kebocoran sumber daya. Contoh program Python untuk membuka file `mbox-short.

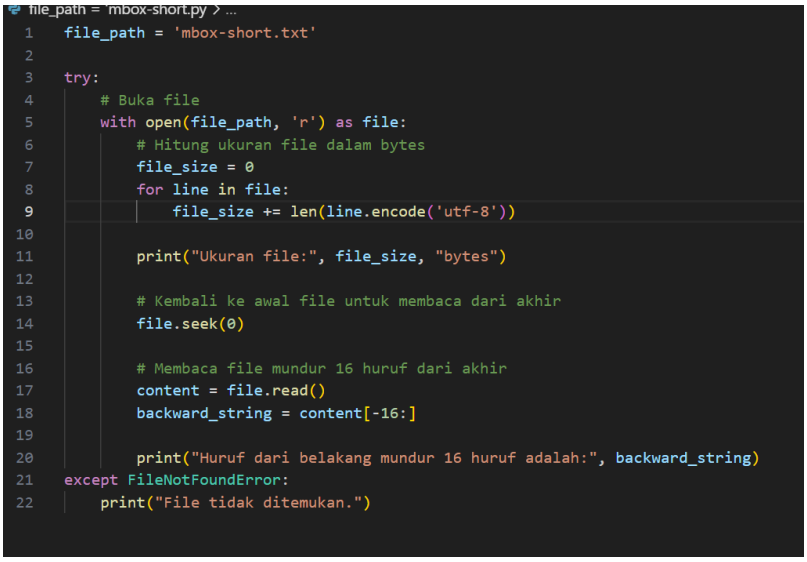


Dalam contoh ini, kita membuka file mbox-short.txt dengan mode membaca ('r'). Kemudian, kita menggunakan loop for untuk membaca baris demi baris dari file tersebut. Pada setiap iterasi, kita melakukan sesuatu dengan baris tersebut, dalam contoh ini hanya mencetaknya. strip() digunakan untuk menghapus karakter newline (\n) dari akhir setiap baris. Setelah selesai membaca file, kita menutup file dengan menggunakan metode close().Dengan cara ini, kita dapat membaca dan melakukan operasi pada setiap baris dalam file mbox-short.txt secara berurutan tanpa harus memuat seluruh isi file ke dalam memori sekaligus.

- 8.3.3 Manipulasi File

Untuk menghindari adanya baris kosong antara setiap baris yang ditampilkan, Anda dapat menggunakan metode `.rstrip` pada setiap baris sebelum mencetaknya. Metode `.rstrip` digunakan untuk menghapus

karakter spasi kosong, termasuk karakter newline , dari akhir sebuah string.
```python handle = open count = 1 for line in handle: if line.Program yang dimaksud membaca file mbox-short.txt, menampilkan ukuran file dalam bytes, dan menampilkan string dari huruf paling belakang maju 16 huruf kedepan, dapat ditulis sebagai berikut:




Dalam program ini:

File dibuka dan ukurannya dihitung dalam bytes dengan menggunakan loop untuk membaca setiap baris dari file.

Setelah itu, file dikembalikan ke awal dengan menggunakan file.seek(0) untuk membaca dari awal lagi.

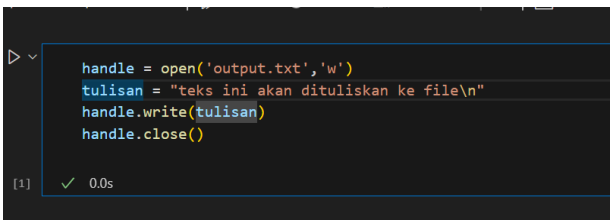
Selanjutnya, menggunakan fungsi file.read() untuk membaca seluruh isi file ke dalam string content.

Dari string tersebut, diambil substring dari 16 karakter terakhir untuk menampilkan string dari huruf paling belakang maju 16 huruf kedepan.

Ukuran file dan string tersebut kemudian ditampilkan.

- 8.3.4 Penyimpanan File

String ini dapat berisi teks apa pun yang ingin Anda tulis ke dalam file. Mode menulis akan membuat file baru jika belum ada, atau akan menghapus isi file jika sudah ada sebelumnya. Ini memastikan bahwa sumber daya file dibebaskan dengan benar setelah digunakan. Jika ada kesalahan, seperti masalah saat membuka atau menulis ke file, pesan kesalahan akan ditampilkan.
Dengan menggunakan kode seperti ini, Anda dapat dengan mudah menulis teks atau data lain ke dalam file menggunakan Python.


```

Dalam contoh ini:

String string_to_write berisi teks yang akan ditulis ke dalam file.

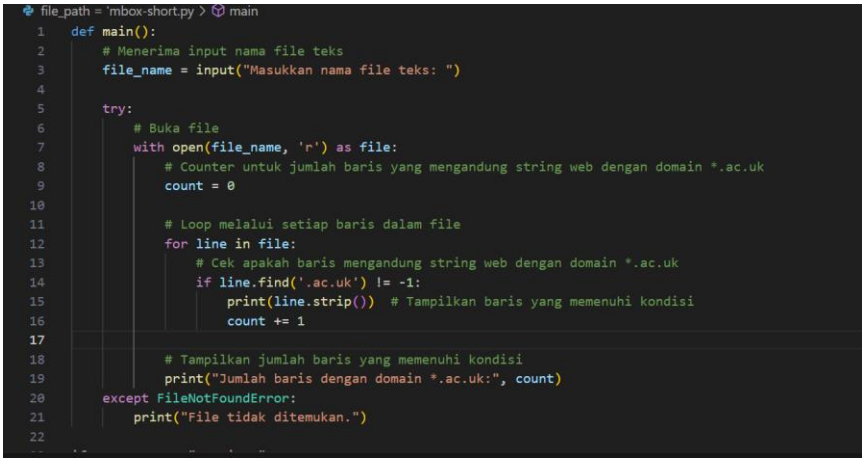
File output.txt dibuka atau dibuat (jika belum ada) dalam mode menulis ('w'). Mode menulis akan membuat file baru jika file tidak ada, atau akan menghapus isi file jika sudah ada sebelumnya.String string_to_write ditulis ke dalam file menggunakan metode write() dari objek file.File ditutup secara otomatis setelah blok with selesai dieksekusi.Pesan berhasil atau gagal ditampilkan tergantung dari apakah operasi penulisan ke dalam file berhasil atau tidak.

MATERI 2

Penjelasan materi 2, dst… sesuai format ini.

- 8.4 Kegiatan Praktikum

Berikut adalah contoh kode program Python yang menerima input nama file teks, menampilkan semua baris yang mengandung string web dengan domain berakhiran '*.ac.uk', dan menghitung jumlah baris yang memenuhi kondisi tersebut menggunakan metode find():



Dalam contoh program ini, untuk mengecek apakah sebuah baris mengandung string web dengan domain berakhiran '*.ac.uk', kita menggunakan metode find('.ac.uk'). Jika nilai yang dikembalikan oleh find() tidak sama dengan -1, maka artinya string '.ac.uk' ditemukan dalam baris tersebut. Jika ditemukan, baris tersebut akan ditampilkan dan counter jumlah baris yang memenuhi kondisi tersebut diinkrementasi. Setelah selesai membaca seluruh file, jumlah baris yang memenuhi kondisi tersebut akan ditampilkan.

- Kasus 8.2

Jika sebuah baris diawali dengan kata «Subject» , maka baris tersebut akan dimodifikasi menggunakan fungsi `capitalize_each_word` untuk mengubah setiap kata menjadi kapital. - Selama proses pembacaan file, program juga menghitung jumlah baris yang diawali dengan kata «Subject». Dengan demikian, program ini memungkinkan pengguna untuk memasukkan nama file teks tertentu, dan kemudian akan menampilkan berapa banyak baris dalam file yang diawali dengan kata «Subject» serta mengubah semua baris tersebut menjadi «capitalized each word» sebelum menampilkannya.

```python
filename = input("nama file: ")
handle = open(filename)
c = 0
for line in handle:
    if line.startswith("Subject:"):
        c += 1
        line = line.strip().title()
        print(line)
print("Jumlah: ",c)
```

Dengan demikian, program ini memungkinkan pengguna untuk memasukkan nama file teks tertentu, dan kemudian akan menampilkan berapa banyak baris dalam file yang diawali dengan kata «Subject» serta mengubah semua baris tersebut menjadi «capitalized each word» sebelum menampilkannya.

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

Tulis jawaban anda untuk soal nomor 1

file_path = 'mbox-short.py > compare_files

```python
def compare_files(file1, file2):
    # Buka kedua file
    with open(file1, 'r') as f1, open(file2, 'r') as f2:
        # Baca isi kedua file ke dalam list
        file1_lines = f1.readlines()
        file2_lines = f2.readlines()

        num_lines_file1 = len(file1_lines)
        num_lines_file2 = len(file2_lines)

        # Tentukan jumlah baris yang akan dibandingkan (ambil yang paling sedikit)
        num_lines_to_compare = min(num_lines_file1, num_lines_file2)

        for i in range(num_lines_to_compare):
            line_file1 = file1_lines[i].strip()
            line_file2 = file2_lines[i].strip()

            if line_file1 != line_file2:
                print(f"Perbedaan pada baris {i+1}:")
                print(f"{file1}: {line_file1}")
                print(f"{file2}: {line_file2}")
                print()

        if num_lines_file1 != num_lines_file2:
            print("Perhatian: Jumlah baris kedua file berbeda.")
            print(f"Jumlah baris {file1}: {num_lines_file1}")
            print(f"Jumlah baris {file2}: {num_lines_file2}")
```

Penjelasan singkat program ini:

Fungsi compare_files(file1, file2): Menerima dua nama file sebagai argumen, membuka kedua file tersebut, membaca baris-barisnya ke dalam list, dan membandingkan setiap baris dari kedua file. Jika ada perbedaan, maka program akan mencetak baris yang berbeda beserta nomor barisnya.

Fungsi main(): Meminta pengguna untuk memasukkan nama dua file yang ingin dibandingkan, kemudian memanggil fungsi compare_files() untuk membandingkan kedua file tersebut.

Dengan menggunakan program ini, Anda dapat membandingkan isi dari dua file teks dan menampilkan perbedaan antara kedua file per barisnya.

SOAL 2

Tulis jawaban anda untuk soal nomor 2 di sini.

```python
def load_questions(file_name):
    # Membaca file dan memuat soal ke dalam dictionary
    questions = {}
    with open(file_name, 'r') as file:
        for line in file:
            # Memisahkan pertanyaan dan jawaban menggunakan separator '||'
            question, answer = line.strip().split('||')
            # Menyimpan pertanyaan dan jawaban ke dalam dictionary
            questions[question.strip()] = answer.strip()
    return questions
def main():
    file_name = 'soal.txt'  # Nama file soal
    questions = load_questions(file_name)

    # Menampilkan nama file soal
    print(f"Nama file1: {file_name}")

    # Loop melalui setiap soal dan tampilkan
    for question, answer in questions.items():
        print(question)
        user_answer = input("Jawab: ").strip().lower()
        if user_answer == answer.lower():
            print("Jawaban benar!")
        else:
            print("Jawaban salah!")
```

Penjelasan singkat program ini:

Fungsi load_questions(file_name): Membaca file yang berisi soal dan jawaban, memisahkan pertanyaan dan jawaban menggunakan separator '||', dan memuatnya ke dalam dictionary dengan pertanyaan sebagai kunci dan jawaban sebagai nilai.

Fungsi main(): Memuat soal dari file 'soal.txt' menggunakan fungsi load_questions(), kemudian menampilkan setiap pertanyaan, menerima jawaban dari pengguna, dan membandingkannya dengan jawaban yang benar. Setelah itu, program akan memberikan feedback apakah jawaban pengguna benar atau salah.Dengan menggunakan program ini, Anda dapat memuat soal dari file teks 'soal.txt', menampilkan setiap soal, menerima jawaban dari pengguna, dan memberikan feedback apakah jawaban yang diberikan benar atau salah.