



Laporan Praktikum Algoritma & Pemrograman

Semester Genap 2024/2025

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.

NIM	71220829
Nama Lengkap	TIMOTIUS EKANA TEJO KESUMA
Minggu ke / Materi	09 / Manajemen File

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2025

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 9.3.1 - Pengantar File

File merupakan kumpulan data dan informasi yang disimpan secara permanen dan dapat berupa teks, gambar, audio, program, dan lainnya. File juga memiliki properti berupa nama, Lokasi, hak akses, tanggal akses, dan lain lain. Random Access Memory (RAM) memiliki sifat *volatile* dan data akan hilang komputer dimatikan atau tidak permanen. Terdapat penyimpanan permanen (*Secondary Memory*) yang dapat digunakan untuk menyimpan file supaya tidak hilang saat komputer atau program dimatikan.

MATERI 9.3.2 - Pengaksesan File

Untuk mengakses file pada python dapat mengikuti langkah berikut:

1. Siapkan file dan path file.
2. Buka file dengan menggunakan:

```
handle = open('mbox.txt')
```
3. Operasikan file dengan menambahkan 'r' atau 'w' untuk read atau write:

```
handleread = open('mbox.txt', 'r')  
handlewrite = open('mbox.txt', 'w')
```
4. Close file.

MATERI 9.3.3 - Manipulasi File

Untuk memanipulasi file, dibutuhkan fungsi loop untuk membaca tiap barisnya. Contoh:

```
handle = open('mbox-short.txt')  
count = 0  
for line in handle:  
    count += 1  
print('Line Count:', count)
```

Untuk membaca seluruh isi sekaligus, dapat menggunakan:

```
handle = open("mbox.txt")
hasil = handle.read()
print("Ukuran:", len(hasil), "bytes")
print("Huruf terakhir mundur 16 huruf:", hasil[-16::1])
```

Untuk menampilkan tiap baris, dapat menggunakan:

```
handle = open("mbox.txt", "r")
for line in handle:
    if line.startswith("Date:"):
        print(line, end="")
```

MATERI 9.3.4 - Penyimpanan File

Untuk menyimpan file, dapat menggunakan write, contoh:

```
handle = open('output.txt', 'w')
tulisan = "teks ini akan dituliskan ke file\n"
handle.write(tulisan)
handle.close()
```

Nantinya, output.txt akan terisi:

```
teks ini akan dituliskan ke file
```

Fungsi write akan selalu melakukan overwrite.

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

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

Jawab:

```
def bedafile(file1_path, file2_path):
    try:
        with open(file1_path, 'r') as file1, open(file2_path, 'r') as file2:
            lines1 = file1.readlines()
            lines2 = file2.readlines()

            max_lines = max(len(lines1), len(lines2))
            differences_found = False

            for i in range(max_lines):
                line1 = lines1[i].strip() if i < len(lines1) else ""
                line2 = lines2[i].strip() if i < len(lines2) else ""
                words1 = line1.split()
                words2 = line2.split()
                different_words1 = []
                different_words2 = []
                max_words = max(len(words1), len(words2))
                for j in range(max_words):
                    w1 = words1[j] if j < len(words1) else "kosong"
                    w2 = words2[j] if j < len(words2) else "kosong"
                    if w1.lower() != w2.lower():
                        different_words1.append(w1)
                        different_words2.append(w2)
                if different_words1 or different_words2:
                    differences_found = True
                    print(f"\nPerbedaan pada baris {i+1}:")
                    print(f"File 1: {line1} | Perbedaan terdapat pada kata {' '.join(different_words1)}")
                    print(f"File 2: {line2} | Perbedaan terdapat pada kata {' '.join(different_words2)}")
            if not differences_found:
                print("Tidak ada perbedaan antara kedua file.")
    except FileNotFoundError as e:
        print("File tidak ditemukan:", e)

bedafile('file1.txt', 'file2.txt')
```

File1.txt:

```
1  Hallo nama saya Timotius
2  Saya adalah mahasiswa Informatika
3  a b c d e
```

File2.txt:

```
1  Hallo nama saya Asep
2  Saya adalah mahasiswa Hukum
```

Output:

```
Perbedaan pada baris 1:
File 1: Hallo nama saya Timotius | Perbedaan terdapat pada kata Timotius
File 2: Hallo nama saya Asep | Perbedaan terdapat pada kata Asep

Perbedaan pada baris 2:
File 1: Saya adalah mahasiswa Informatika | Perbedaan terdapat pada kata Informatika
File 2: Saya adalah mahasiswa Hukum | Perbedaan terdapat pada kata Hukum

Perbedaan pada baris 3:
File 1: a b c d e | Perbedaan terdapat pada kata a, b, c, d, e
File 2: | Perbedaan terdapat pada kata kosong, kosong, kosong, kosong, kosong
PS D:\Kuliah\Semester_6\Alpro\Laporan_9> █
```

Penjelasan:

Menggunakan try and except untuk melihat apakah file yang diinputkan tersedia atau tidak, kemudian menggunakan fungsi open(path,'r') untuk membuka dan membaca file, kemudian dilihat jumlah baris mana yang paling panjang dari baris yang ada di kedua file dengan fungsi max(len(line)). Tambahkan parameter variabel difference_found dan berikan value default sebagai False sebagai parameter terdapat perbedaan atau tidak. Kemudian, pada perulangan, gunakan nilai maksimal dari baris antara ke 2 buah file sebagai indeks, gunakan strip untuk menghilangkan whitespace dan \n di setiap barisnya, gunakan fungsi split juga untuk memisahkan kata. Pada nested loop, fungsi berjalan untuk melihat apakah terdapat kata yang kosong atau tidak, jika terdapat, maka append kosong. Setelah nested loop, jika terdapat perbedaan kata di file 1 atau 2, ubah parameter differences_found menjadi True, dan print baris dari file 1 dan 2 dan letak perbedaan katanya.

Jika perulangan sudah selesai dan tidak ditemukan perbedaan (variabel differences_found = False), maka print tidak ada perbedaan antara kedua file.

SOAL 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!

Jawab:

```
2.py > ...
1 def soal(nama_file):
2     print("Nama file:", nama_file)
3     try:
4         with open(nama_file, 'r') as file:
5             for line in file:
6                 if '||' in line:
7                     bagian = line.split(' || ')
8                     soal = bagian[0]
9                     jawaban_benar = bagian[1]
10
11                     if jawaban_benar:
12                         jawaban_benar = jawaban_benar.replace('\n', '')
13                     if '\n' in soal:
14                         soal = soal.replace('\n', '')
15                     print(soal)
16
17                     jawaban_user = input("Jawab: ")
18                     if jawaban_user.lower() == jawaban_benar.lower():
19                         print("Jawaban benar!\n")
20                     else:
21                         print("Jawaban salah!\n")
22             except FileNotFoundError:
23                 print("File tidak ditemukan.")
24
25 soal('soal.txt')
```

Output:

```
Nama file: soal.txt
1+1 =
Jawab: 2
Jawaban benar!

Bendera Indonesia?
Jawab: Merah Putih
Jawaban benar!

Kota gudeg adalah:
Jawab: Solo
Jawaban salah!

Komponen PC untuk penyimpanan file adalah...
Jawab: Harddisk
Jawaban benar!

50 * 20 =
Jawab: 60 bang
Jawaban salah!

PS D:\Kuliah\Semester_6\Alpro\Laporan_9> 
```

Penjelasan:

Kode berjalan diawali dengan print nama file, kemudian gunakan try and except untuk mendeteksi apakah file ditemukan atau tidak. Gunakan fungsi open untuk membuka file dan parameter r untuk membaca, pada perulangan, untuk setiap baris (*line*) di dalam file, maka, jika terdapat '|' didalam baris, maka gunakan split untuk memisahkan ||, gunakan split(||) karena terdapat white space dalam baris, dapat digunakan juga fungsi strip(), namun disini saya tidak menggunakannya. Kemudian bagi baris tersebut, soal untuk indeks ke 0 dan jawaban_benar untuk indeks ke 1, jika jawaban_benar, maka replace \n dengan " untuk menghilangkan \n dan pada soal, lakukan hal yang sama, kemudian print soal untuk menunjukkan soal kepada user, user akan menginputkan jawaban dan kemudian jika jawaban_user sama dengan jawaban_benar, maka akan print Jawaban benar!, jika tidak, maka akan print Jawaban salah!. Disini saya menggunakan .lower() untuk menghilangkan case sensitive.

Link GitHub:

https://github.com/TimotiusEkanaT/Alpro/tree/main/Laporan_9