Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	**71230985**
Nama Lengkap	**TOMAS BECKET**
Minggu ke / Materi	**09 / Membaca & 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.

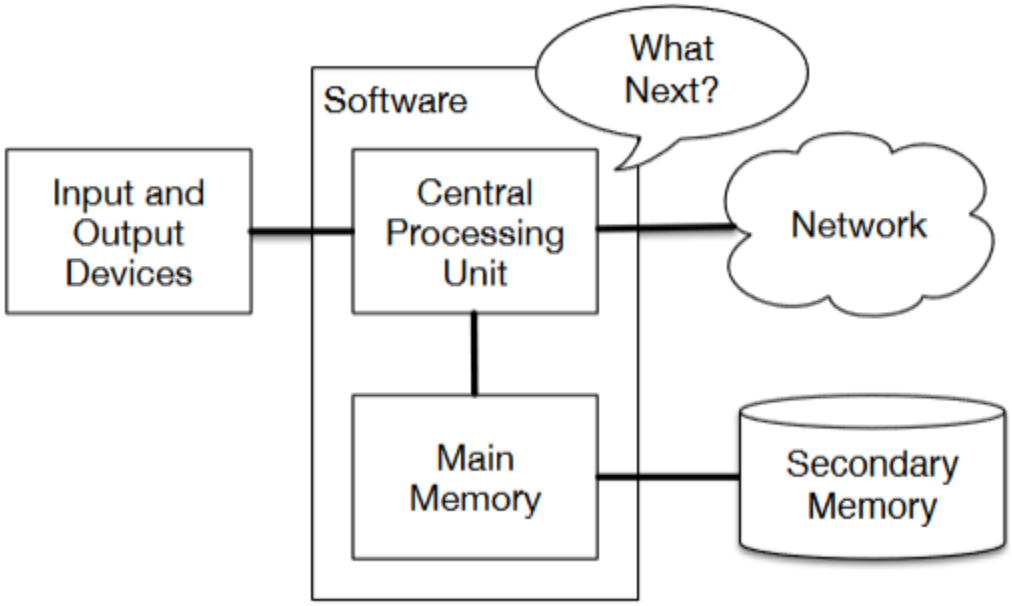
PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2024

BAGIAN 1: MATERI MINGGU INI (40%)

MATERI 1

PENGANTAR FILE

Pada Materi Minggu ke-9 Yaitu membaca dan menulis File pada sebuah program membutuhkan memory primer di dalam komputer. Yang dimana penyimpanan data di dalam sebuah memory bersifat tidak permanen (volatile). Dan kerena bersifat Volatile, memory primer tidak akan menyimpan data setelah program dimatikan



Dengan begitu jika ingin menyimpan sebuah data pada program harus menggunakan penyimpanan tetap yaitu secondary memory.

Sehingga jika dihat lebih detail Memory primer (atau yang sering disebut sebagai RAM, singkatan dari Random Access Memory) adalah jenis memori komputer yang digunakan untuk menyimpan data yang sedang aktif dan digunakan oleh program-program yang berjalan pada saat itu. RAM merupakan tempat penyimpanan sementara yang sangat cepat dan dapat diakses secara acak oleh prosesor komputer. Data yang disimpan di dalam RAM akan tetap ada selama komputer tetap dalam keadaan menyala. Namun, saat komputer dimatikan atau direstart, semua data yang disimpan di dalam RAM akan hilang.

Dan ini disebabkan oleh sifat volatile dari RAM. Volatile berarti bahwa data yang disimpan di dalamnya tidak bersifat permanen dan akan hilang saat daya listrik atau sumber daya yang menyuplai RAM terputus. Dibandingkan dengan jenis memori lain seperti penyimpanan

sekunder (seperti hard disk atau SSD), RAM memiliki kecepatan akses yang lebih tinggi namun kapasitas penyimpanan yang lebih terbatas dan sifatnya yang tidak permanen.

Karena sifatnya yang tidak permanen, program-program yang menggunakan memory primer tidak dapat menyimpan data setelah program dimatikan atau komputer dimatikan. Oleh karena itu, jika data perlu disimpan untuk digunakan kembali di masa mendatang, perlu menggunakan media penyimpanan yang permanen seperti hard disk, SSD, atau cloud storage.

File yang disimpan pada secondary memory akan digunakan untuk menyipan data dari program dan tidak akan hilang walaupun komputer dimatikan.

MATERI 2

```
handle = open('mbox-short.txt')
print(handle)
```

Ini adalah contoh dari fungsi open() dalam Python yang digunakan untuk membuka file. Saat fungsi open() dieksekusi, file 'mbox-short.txt' dibuka dalam mode 'r', yang berarti mode baca (read). Mode 'r' menunjukkan bahwa file tersebut dibuka untuk membaca isi dari file.

Ketika print(handle) dieksekusi, ini mencetak objek file yang disebut sebagai "file handle" atau "file object". Objek file ini adalah representasi dari file yang telah dibuka dalam program Python.

```
<_io.TextIOWrapper name='mbox-short.txt' mode='r' encoding='cp1252'>
```

<_io.TextIOWrapper ini Menunjukkan bahwa ini adalah objek yang membungkus operasi I/O (input/output) teks. Sedangkan 'mbox-short.txt' menunjukkan nama file yang dibuka

Dan 'r' menunjukkan mode pembukaan file, dalam hal ini mode baca ('r').

```
handle = open('mbox-short.txt')
print(handle)
# Hitung jumlah Baris/line
count = 0
for line in handle:
    count += 1

print(f"Jumah line : {count}")
```

Berikut adalah contoh Source Code pada minggu 9 dimana Fungsi dari potongan kode ini adalah membuka file dengan nama 'mbox-short.txt', kemudian menghitung jumlah baris atau line di dalam file tersebut.

Pada baris pertama open('mbox-short.txt'), file 'mbox-short.txt' dibuka dalam mode default, yaitu mode baca ('r'). Objek file yang dihasilkan disimpan dalam variabel handle. Kemudian dilanjut dengan print(handle), objek file yang dihasilkan kemudian dicetak. Ini akan mencetak informasi tentang objek file, seperti yang dijelaskan sebelumnya.

Dan Pada baris ketiga, count = 0, variabel count diinisialisasi dengan nilai 0. Variabel ini akan digunakan untuk menghitung jumlah baris.

Kemudian di baris keempat, for line in handle: dilakukan iterasi melalui setiap baris di dalam file yang dibuka. Setiap baris dari file disimpan dalam variabel line.

Dan melakukan count += 1, setiap kali iterasi dilakukan, nilai variabel count ditambah satu, sehingga menghitung jumlah baris. Untuk mencetak hasil out put maka menuliskan kode print(f"Jumlah line : {count}"), setelah iterasi selesai, jumlah total baris yang telah dihitung dan dicetak menggunakan f-string.

Sehingga, potongan kode ini membuka file, menghitung jumlah baris di dalamnya, dan mencetak jumlah baris tersebut.

MATERI 3

Pada Python, untuk menulis ke file, langkah-langkah umumnya mirip dengan cara membuka (open) file, namun dengan menggunakan mode menulis ('w') sebagai parameter kedua saat membuka file. Setelah file dibuka, code tersebut dapat menggunakan metode .write() untuk menuliskan isi ke dalamnya. Setelah selesai menulis, penting untuk menutup file dengan menggunakan metode .close().

```
14  handle = open('pertemuan9.txt', 'w')
15  handle.write('Hello world!\n')
16  handle.write('pertemuan9')
17
18  handle.close()
```

Fungsi handle.write sendiri digunakan dalam Python untuk menuliskan data ke dalam sebuah file yang sudah dibuka. Ketika membuka file dalam mode menulis ('w').

Dan fungsi handle.close hanya digunakan ketika tidak menggunakan fungsi With yang akan dijelaskan pada Sub bab materi 3 bahwa handel.close digunakan untuk menutup file yang telah dibuka sebelumnya dengan menggunakan fungsi open(). Fungsi lainnya Setelah menutup file, itu berarti memory membebaskan sumber daya yang digunakan oleh file tersebut, seperti memori sistem dan penggunaan sistem file. Ini membantu mencegah terjadinya kebocoran sumber daya dan memastikan bahwa program Anda bekerja dengan efisien.

Dalam beberapa situasi, seperti saat menulis ke file, menutup file memastikan bahwa semua perubahan yang di buat telah disimpan dengan benar. Pada beberapa sistem operasi, menutup file juga penting untuk memastikan bahwa perubahan yang Anda buat telah ditulis ke disk secara fisik.

Selain itu, menutup file membantu mencegah terjadinya konflik akses dengan program atau proses lain yang ingin mengakses file yang sama. Sehingga dengan menutup file setelah selesai menggunakannya akan membebaskan file tersebut untuk digunakan oleh program lain.

MATERI 4

Fungsi with open() adalah cara yang disarankan untuk membuka dan menggunakan file dalam Python. Karena fungsi with open menyediakan cara yang lebih bersih dan aman untuk berinteraksi dengan file dan dapat memastikan bahwa file ditutup secara otomatis setelah semua operasi selesai dilakukan, bahkan jika terjadi kesalahan di tengah proses.

Cara kerja with open() adalah dengan menggunakan blok with, diikuti oleh open() untuk membuka file. Dalam blok tersebut, dapat melakukan operasi membaca atau menulis ke file tanpa perlu khawatir tentang menutup file secara manual seperti pada materi 3 yang dimana menggunakan Fungsi handle.close, setelah blok with selesai dieksekusi, baik karena selesai atau karena terjadi kesalahan, file akan ditutup secara otomatis.

Contoh seperti Source code minggu 9 yang diberikan pada Asisten Dosen dimana fungsi tersebut hampir mirip seperti membuka dan membaca file pertemuan 9 namun ingin mengubah kata ‘Hello’ menjadi ‘Hi’

```
20 with open('pertemuan9.txt', 'r') as file:
21     ganti_kata = file.read().replace('Hello', "Hi")
22     with open('pertemuan9.txt', 'w') as file:
23         file.write(ganti_kata)
```

Dengan cara menuliskan with open('pertemuan9.txt', 'w') as file:. Dan pada blok with open() kedua dibuka untuk menulis ke file yang sama, 'pertemuan9.txt', dalam mode menulis ('w'). dan juga menggunakan nama variabel file. file.write(ganti_kata): Dalam blok with kedua, teks yang telah diubah (ganti_kata) dengan kata-kata yang diganti ditulis kembali ke dalam file 'pertemuan9.txt' menggunakan metode .write().

MATERI 5

Adapun contoh lainnya dimana kode dibawah akan menunjukan dan membaca file mbox-short.txt yang berawalan dengan kata Received dengan batas 10x sehingga tidak memprint semua yang ada kata Received pada kalimat awal dengan menambah fungsi Count += 1 untuk memenuhi kondisi if count <= 10.

```
26  # Membuka File
27  with open('mbox-short.txt', 'r') as file:
28      count = 1
29      for line in file:
30          if line.startswith('Received'):
31              if count <= 10:
32                  print(f'{count} {line}')
33                  count += 1
```

Sehingga output yang diberikan akan seperti berikut :

```
1 Received: from murder (mail.umich.edu [141.211.14.90])

2 Received: from murder ([unix socket])

3 Received: from holes.mr.itd.umich.edu (holes.mr.itd.umich.edu [141.211.14.79])

4 Received: FROM paploo.uhi.ac.uk (app1.prod.collab.uhi.ac.uk [194.35.219.184])

5 Received: from paploo.uhi.ac.uk (localhost [127.0.0.1])

6 Received: from prod.collab.uhi.ac.uk ([194.35.219.182])

7 Received: from nakamura.uits.iupui.edu (nakamura.uits.iupui.edu [134.68.220.122])

8 Received: from nakamura.uits.iupui.edu (localhost [127.0.0.1])

9 Received: (from apache@localhost)

10 Received: from murder (mail.umich.edu [141.211.14.97])
```

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

1.1 SOURCE CODE

```python
# LATIHAN 8.1

with open('SOAL-1/teks1.txt', 'r') as file1, open('SOAL-1/teks2.txt', 'r') as file2:
    lines1 = file1.readlines()
    lines2 = file2.readlines()

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

    for line in range(max_lines):
        line1 = lines1[line].strip() if line < len(lines1) else ''
        line2 = lines2[line].strip() if line < len(lines2) else ''

        if line1 != line2:
            print(f'Perbedaan baris {line + 1}:')
            print(f'File 1: {line1}')
            print(f'File 2: {line2}')
            print()
```

SOAL-1 > teks1.txt

```
1  Saya lapar
2  Saya makan
3  Saya kenyang
4  Saya tidak lapar
```

SOAL-1 > teks2.txt

```
1  Saya kenyang
2  Saya makan
3  Saya tidak kenyang
4  Saya lapar
```

1.2 PENJELASAN

```
with open('SOAL-1/teks1.txt', 'r') as file1, open('SOAL-1/teks2.txt', 'r') as file2:
```

Kode tersebut membuka dua file teks, teks1.txt dan teks2.txt, dan membandingkan setiap baris dari keduanya. Jika ada perbedaan pada sebuah baris, kode tersebut mencetak baris tersebut beserta isi dari kedua file di baris yang bersesuaian. Setelah membandingkan semua baris, kedua file akan ditutup secara otomatis. Dimana with open('SOAL-1/teks1.txt', 'r') as file1, open('SOAL-1/teks2.txt', 'r') as file2:: Blok with open() akan membuka kedua file teks untuk dibaca ('r'). File pertama dibuka dengan nama file1 dan file kedua dibuka dengan nama file2.

```
lines1 = file1.readlines()
lines2 = file2.readlines()
```

Dan dilanjutkan dengan fungsi file1/2.readlines yang digunakan untuk membaca seluruh konten dari file pertama juga kedua dan menyimpannya dalam bentuk daftar baris dalam variabel lines1 dan lines2.

```
max_lines = max(len(lines1), len(lines2))
```

max_lines = max(len(lines1), len(lines2)) yang dimana variabel max_lines diinisialisasi dengan panjang maksimum dari kedua daftar baris lines1 dan lines2, yang akan digunakan sebagai batas iterasi untuk memastikan bahwa kita akan membandingkan setiap baris dari kedua file.

```
for line in range(max_lines):
    line1 = lines1[line].strip() if line < len(lines1) else ''
    line2 = lines2[line].strip() if line < len(lines2) else ''
```

Pada fungsi ini lah Iterasi akan dilakukan sejumlah maksimum baris yang ada di kedua file.

Dan Variabel line1 akan diisi dengan baris ke-line dari file pertama yang sudah di-strip (menghilangkan spasi di awal hingga akhir baris) ketika line kurang dari panjang lines1, namu jika tidak maka akan diisi dengan string kosong. Dan line2 sama halnya seperti line satu untuk menghilangkan sebuah spasi menggunakan fungsi .strip().

```
if line1 != line2:
    print(f'Perbedaan baris {line + 1}:')
    print(f'File 1: {line1}')
    print(f'File 2: {line2}')
    print()
```

Dan fungsi terakhir Dilakukan pengecekan apakah baris dari kedua file tersebut tidak sama., jika tidak sama maka akan memprint hasil output yang akan diberikan.

1.3 HASIL OUTPUT

```
Perbedaan baris 1:
File 1: Saya lapar
File 2: Saya kenyang

Perbedaan baris 3:
File 1: Saya kenyang
File 2: Saya tidak kenyang

Perbedaan baris 4:
File 1: Saya tidak lapar
File 2: Saya lapar
```

SOAL 2

2.1 SOURCE CODE

```python
# LATIHAN 8.2

pertanyaan = []

with open('SOAL-2/soal.txt', 'r') as file:
    for line in file:
        soal, jawab = line.strip().split('||')
        pertanyaan.append((soal.strip(), jawab.strip()))

total_pertanyaan = len(pertanyaan)

for i in range(total_pertanyaan):
    soal, jawab = pertanyaan[i]
    print(f"{soal}")
    jawab_user = input("Jawab: ").strip()

    if jawab_user.lower() == jawab.lower():
        print("Jawaban benar!")
    else:
        print("Jawaban salah")
```

SOAL-2 > soal.txt

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

2.2 PENJELASAN

Kode diatas bertujuan untuk memuat kumpulan pertanyaan dan jawaban dari sebuah file teks pada soal.txt yang diaman akan mengajukan setiap pertanyaan kepada pengguna, dan memeriksa apakah jawaban yang diberikan oleh pengguna sesuai dengan jawaban yang benar.

```
pertanyaan = []
```

pertanyaan = []: Membuat sebuah list kosong yang akan digunakan untuk menyimpan pasangan pertanyaan dan jawaban dari file teks.

```
with open('SOAL-2/soal.txt', 'r') as file:
    for line in file:
        soal, jawab = line.strip().split('||')
        pertanyaan.append((soal.strip(), jawab.strip()))
```

with open('SOAL-2/soal.txt', 'r') as file: akan Membuka file teks 'soal.txt' untuk dibaca dan menyimpan objek file dalam variabel file. Dan for line in file: akan Mengiterasi setiap baris dalam file.

soal, jawab = line.strip().split('||'): Memisahkan baris menjadi dua bagian berdasarkan pemisah '||', yaitu bagian pertanyaan dan jawaban, kemudian menghapus spasi di awal dan akhir baris menggunakan .strip(). Dan pertanyaan.append((soal.strip(), jawab.strip())): akan menambahkan pasangan pertanyaan dan jawaban yang telah dipisahkan ke dalam list pertanyaan.

```
total_pertanyaan = len(pertanyaan)
```

total_pertanyaan = len(pertanyaan): Menghitung jumlah total pertanyaan yang telah dimuat dari file.

```
for i in range(total_pertanyaan):
    soal, jawab = pertanyaan[i]
    print(f"{soal}")
    jawab_user = input("Jawab: ").strip()
```

Dan penggunaan for i in range(total_pertanyaan): akan melakukan Iterasi melalui setiap pertanyaan. Yang dimana soal, jawab = pertanyaan[i]: mendapatkan pasangan pertanyaan dan jawaban dari list pertanyaan. Dan menuliskan print(f"{soal}"): untuk mencetak pertanyaan ke layar.

Fungsi jawab_user = input("Jawab: ").strip() akan meminta pengguna untuk memberikan jawaban melalui input yang dimasukkan.

```
if jawab_user.lower() == jawab.lower():
    print("Jawaban benar!")
else:
    print("Jawaban salah")
```

if jawab_user.lower() == jawab.lower():: akan memeriksa apakah jawaban pengguna (yang sudah dikonversi menjadi huruf kecil) sama dengan jawaban yang benar (juga dikonversi menjadi huruf kecil).

print("Jawaban benar!") atau print("Jawaban salah"): akan mencetak pesan ke layar tergantung dari hasil perbandingan jawaban.

Dan dengan proses kode tersebut akan mengajukan pertanyaa yang sudah tersedia pada teks soal dan akan memeriksa jawaban yang akan diberikan oleh pengguna, jika benar maka ‘jawaban benar!’, jika salah maka ‘jawaban salah’

2.3 HASIL OUPUT

```
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
Jawaban benar!
```