|  |  |
| --- | --- |
| A picture containing text, clipart  Description automatically generated | **Laporan Praktikum Algoritma dan Pemrograman**  Semester Genap 2023/2024 |

|  |  |
| --- | --- |
| **NIM** | **71220895** |
| **Nama Lengkap** | **Drestanta Dipta Jalu Prakasya** |
| **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**

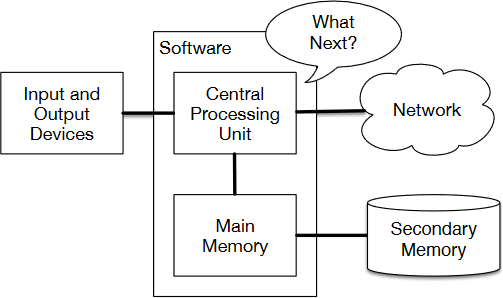
# BAGIAN 1: MATERI MINGGU INI (40%)

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

Program yang berjalan di komputer memerlukan RAM, atau random access memory. Semua data yang digunakan oleh program disimpan di memori, dan ketika program dijalankan dan diaktifkan, semua data yang disimpan di memori juga ikut hilang. Data yang disimpan dalam memori tidak disimpan secara permanen; itu tidak stabil. Oleh karena itu, program yang hanya menggunakan memori primer tidak dapat menyimpan data setelah program tersebut dimanipulasi.

Untuk menyimpan data secara permanen dan memudahkan akses ketika program selesai, memori sekunder seperti harddisk, SSD, atau media penyimpanan lainnya harus digunakan. Bahkan setelah suatu program diotomatisasi, data yang telah disimpan di memori sekunder akan tetap ada, sehingga memungkinkan untuk menyimpan dan mengambil data secara berurutan di antara sesi program yang berbeda. Secondary memory dapat dilihat di bawah.



## MATERI 2

Manipulasi file Python melibatkan beberapa bahasa dasbor:

1. Siapkan file dan path yang akan diakses.

2. Buka file menggunakan fungsi open().

3. Jalankan tindakan yang diinginkan pada file tersebut di atas, seperti membaca atau menulis isinya.

4. Tutup file setelah menggunakan close() untuk mengakhiri proses.

Contoh program Python untuk mengakses file dapat dijelaskan sebagai berikut:

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

print(handle)

dan hasilnya:

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

Hasilnya menunjukkan nama file, mode (r untuk membaca), dan pengkodean (UTF-8) yang digunakan sistem operasi Python. Jika nama file tidak dimasukkan, pesan kesalahan seperti ini akan muncul:

Traceback (most recent call last):

File "c:\Users\User\Downloads\Laprak 8\Laprak 8.py", line 26, in <module>

handle = open('tidak-ada.txt')

^^^^^^^^^^^^^^^^^^^^^

FileNotFoundError: [Errno 2] No such file or directory: 'tidak-ada.txt'

Dalam file teks, karakter sering kali dipisahkan dengan spasi. Umumnya proses pembacaan suatu file teks menggunakan model pembacaan setiap karakter hingga End of File (EOF).

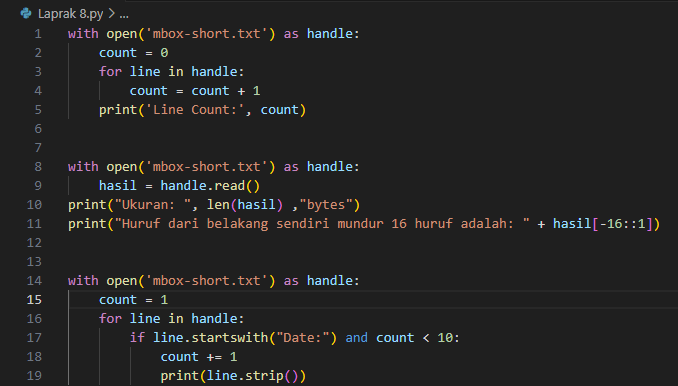
## MATERI 3

Untuk memanipulasi file, langkah pertama adalah membuka file. Dengan Python, langkah-langkahnya adalah membuat file, menjalankan loop pada setiap baris, lalu memilih file. Hal ini diperlukan untuk mencegah kesalahan pembuatan file berukuran besar yang dapat menyebabkan kegagalan fungsi sistem. Dengan menggunakan fungsi len() dari string yang diperoleh dari file tersebut, kita bisa mendapatkan ukuran file dalam byte.

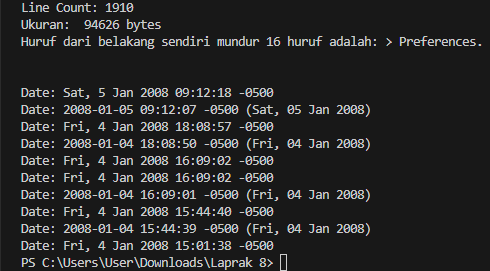
Namun, penting untuk diingat bahwa penggunaan fungsi read() pada file mungkin meningkatkan ukuran memori, sehingga penggunaan teknik perulangan menjadi lebih efektif. Selama proses perulangan, kita juga dapat memanipulasi file, misalnya mencetak hanya bagian string saat ini, atau mencetak karakter yang dimulai dengan "Tanggal:". Untuk menampilkan baris kosong yang dihasilkan dari baris baru, Anda dapat menggunakan fungsi rstrip() pada baris yang dimaksud atau menggunakan fungsi print tanpa baris baru.

Kode Python untuk membaca file, menampilkan ukurannya dalam byte, dan menampilkan baris yang diakhiri dengan "Tanggal:" tanpa awalan nol:

Input:



Output:



## MATERI 4

Penentuan ukuran file dengan Python dapat dilakukan dengan membuat file menggunakan metode open() dan mode tulis ('w'). Setelah itu, sebuah string dapat ditulis ke file menggunakan fungsi tulis (<string>). Harap jangan membuka file setelah selesai dengan menggunakan close().

Contoh lengkap adalah sebagai berikut:

handle = open('output.txt', 'w')

tulisan = "teks ini akan dituliskan ke file\n "

handle.write(tulisan)

handle.close()

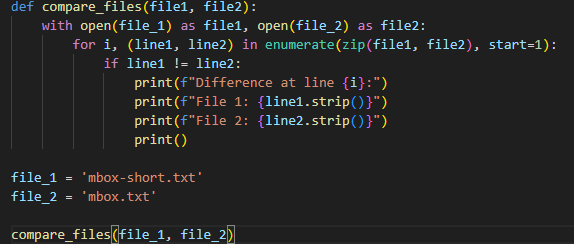
# BAGIAN 2: LATIHAN MANDIRI (60%)

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

Python yang membandingkan dua file teks, mbox-short.txt dan mbox.txt, dan menampilkan perbedaan antara keduanya per baris jika ada:

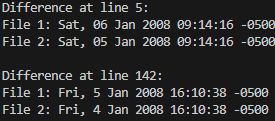
Input:



Output jika tidak ada:



Output jika ada:

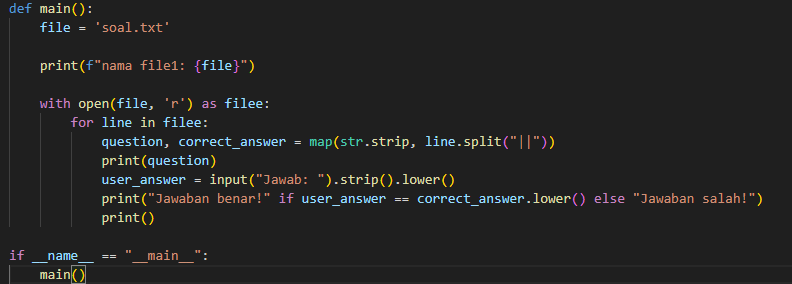


Fungsi ini menggunakan dua jalur sebagai argumen, membuat file yang dimaksud, dan membandingkan setiap baris dari kedua file. Jika terdapat perbedaan pada salah satu field, maka aplikasi akan mengenali nama field dan isi field dari kedua file tersebut.

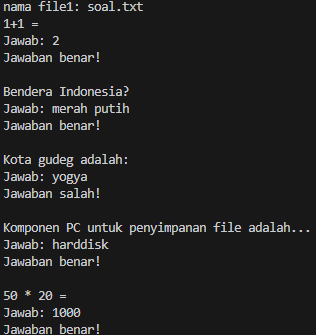
## SOAL 2

Python dapat digunakan untuk mengekstrak data dari file teks bernama soal.txt, meminta masukan pengguna, dan mengekstrak masukan pengguna beserta apakah masukan tersebut benar atau salah:

Input:



Output:



Program ini membaca file bernama soal.txt dan mengekstrak setiap baris secara diam-diam. Selanjutnya, tanggapi setiap pertanyaan, minta masukan dari pengguna, dan tunjukkan apakah masukan tersebut akurat atau tidak.