



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>71230971</b>
<b>Nama Lengkap</b>	<b>James Marvin Santoso</b>
<b>Minggu ke / Materi</b>	<b>08 / Membaca dan Menulis File</b>

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

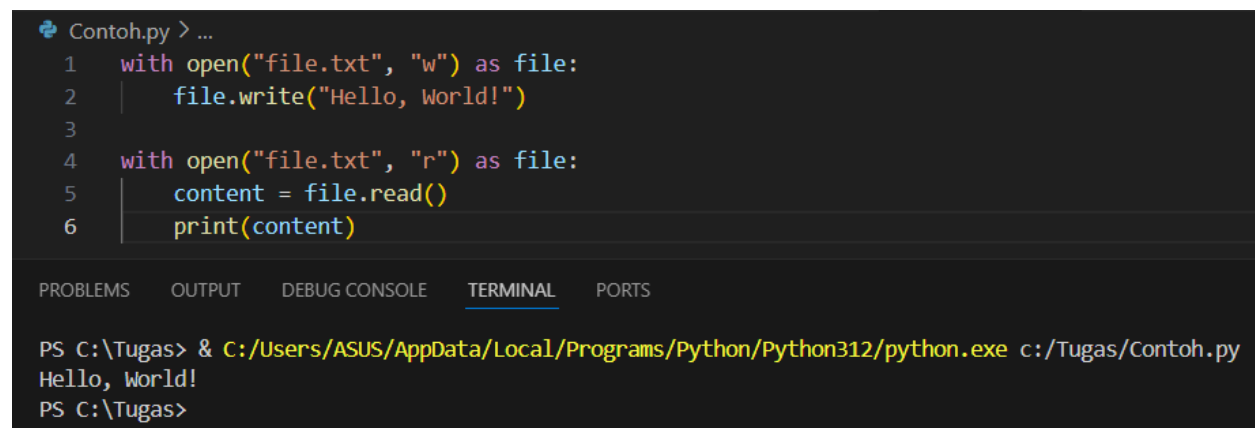
# BAGIAN 1: MATERI MINGGU INI

## Pengantar File

Program menggunakan memory primer untuk menyimpan data sementara. Data dalam memory primer akan hilang saat program dimatikan karena sifatnya yang tidak permanen (volatile). Untuk menyimpan data secara permanen, digunakan secondary memory seperti file. File adalah kumpulan data yang disimpan di secondary memory, seperti hard disk, dan tidak akan hilang saat komputer dimatikan. File memiliki berbagai properti seperti nama, ukuran, lokasi, pemilik, hak akses, dan tanggal akses.

File adalah kumpulan data yang disimpan di dalam memori sekunder, seperti hard drive atau SSD, dan dapat diakses dan dikelola oleh program komputer. Dalam Python, pengelolaan file dilakukan menggunakan fungsi bawaan yang disebut `open()`. Fungsi ini digunakan untuk membuka file dalam mode tertentu, seperti mode membaca (`read`), menulis (`write`), atau menambahkan (`append`).

Berikut contoh singkat penggunaan file dalam python, sebagai berikut :

The image is a screenshot of a Python IDE. The top part shows a code editor with a file named 'Contoh.py'. The code consists of six lines: line 1 opens 'file.txt' in write mode ('w') and assigns it to 'file'; line 2 writes 'Hello, World!' to the file; line 3 is a blank line; line 4 opens 'file.txt' in read mode ('r') and assigns it to 'file'; line 5 reads the content of the file and assigns it to 'content'; line 6 prints the content. Below the code editor, there is a terminal window. The terminal shows the command 'PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe c:/Tugas/Contoh.py' being executed, followed by the output 'Hello, World!' and the prompt 'PS C:\Tugas>'.

```
Contoh.py > ...
1  with open("file.txt", "w") as file:
2      file.write("Hello, World!")
3
4  with open("file.txt", "r") as file:
5      content = file.read()
6      print(content)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe c:/Tugas/Contoh.py  
Hello, World!  
PS C:\Tugas>

Dalam contoh ini, pertama-tama kita membuka file "file.txt" dalam mode menulis ("w") dan menulis string "Hello, World!" ke dalamnya. Kemudian, kita membuka file yang sama dalam mode membaca ("r") dan membaca kontennya menggunakan metode `read()`, lalu mencetaknya. Dengan menggunakan statement `with`, kita memastikan bahwa file ditutup secara otomatis setelah blok `with` selesai dieksekusi.

## Pengaksesan File

Dalam Python, akses file dilakukan menggunakan fungsi bawaan `open()`, yang memungkinkan Anda membuka file dalam berbagai mode. Fungsi `open()` mengembalikan objek file yang dapat digunakan untuk membaca, menulis, atau menambahkan isi ke file. Setelah selesai bekerja dengan file, disarankan untuk menutupnya dengan menggunakan metode `close()` untuk memastikan bahwa sumber daya file dilepaskan dan tidak digunakan oleh program lagi.

- Mode-mode Akses File

- a) **"r" (Read):** Mode default. Membuka file untuk dibaca. Jika file tidak ditemukan, akan terjadi exception.
- b) **"w" (Write):** Mode menulis. Jika file tidak ada, akan dibuat. Jika file sudah ada, isinya akan dihapus.
- c) **"a" (Append):** Mode menambahkan. Membuka file untuk menambahkan isi di akhir file.
- d) **"r+" (Read and Write):** Mode membaca dan menulis. File akan dibuka untuk dibaca dan ditulis. Jika file tidak ditemukan, akan terjadi exception.

Berikut adalah contoh singkat pengaksesan file dalam Python, sebagai berikut :

- a) Membuka File

```
Contoh.py > ...  
1  with open('file.txt', 'r') as file:  
2      content = file.read()  
3      print(content)
```

- b) Menulis ke File

```
Contoh.py > ...  
1  with open('file.txt', 'w') as file:  
2      file.write('Hello, World!')
```

- c) Menambahkan File

```
Contoh.py > ...  
1  with open('file.txt', 'a') as file:  
2      file.write('\nAppending a new line.')
```

d) Membaca Isi File

```
Contoh.py > ...  
1  content = file.read()  
2  print(content)  
3  
4  line = file.readline()  
5  print(line)
```

Pada baris pertama, `content = file.read()` membaca seluruh isi file yang dibuka dan menyimpannya ke dalam variabel `content`, sedangkan `print(content)` akan mencetak isi file tersebut. Pada baris kedua, `line = file.readline()` membaca satu baris teks dari file yang dibuka dan menyimpannya ke dalam variabel `line`, kemudian `print(line)` akan mencetak baris tersebut.

e) Mengedit Isi File

```
Contoh.py > ...  
1  with open("myfile.txt", "r") as file:  
2      content = file.read()  
3      new_content = content.replace("lama", "baru")  
4  
5  with open("myfile.txt", "w") as file:  
6      file.write(new_content)
```

Kode tersebut membuka file "myfile.txt" untuk dibaca, mengganti setiap kemunculan kata "lama" dengan "baru" dalam isi file, dan kemudian menulis kembali isi file yang telah diubah ke "myfile.txt".

f) Menutup File

```
Contoh.py  
1  file.close()
```

`file.close()` adalah sebuah perintah dalam Python yang digunakan untuk menutup file yang sedang dibuka. Ketika sebuah file dibuka dalam mode baca atau tulis, penting untuk menutupnya setelah selesai digunakan agar sumber daya sistem dapat dibebaskan dan file tersebut tidak tetap terbuka dalam memori. Dengan menutup file, kita memastikan bahwa semua data yang tertulis sudah benar-benar tersimpan dan tidak terjadi kebocoran memori.

## Manipulasi File

Manipulasi file dalam Python merupakan aspek penting yang memungkinkan Anda untuk berinteraksi dengan data eksternal, menyimpan dan memuat informasi, serta membangun aplikasi yang lebih kompleks. Python menyediakan berbagai fungsi bawaan dan pustaka pihak ketiga yang mempermudah operasi file, menjadikan prosesnya intuitif dan efisien. Memahami cara membuka, membaca, menulis, mengedit, dan menutup file secara efektif akan membantu Anda membangun aplikasi yang lebih kuat dan dinamis.

Manipulasi file dalam Python melibatkan berbagai operasi untuk membuka, membaca, menulis, mengedit, dan menutup file. Python menyediakan sejumlah fungsi bawaan yang memudahkan penggunaan file, seperti fungsi `open()` untuk membuka file, metode `read()` untuk membaca isi file, metode `write()` untuk menulis ke file, dan sebagainya.

Untuk bisa memanipulasi file maka harus dimulai dari membaca file tersebut terlebih dahulu. Cara membaca file pada Python adalah:

1. Siapkan file
2. Open file
3. Loop setiap baris pada file
4. Close file

## Penyimpanan File

Modul `os` di Python adalah bagian dari pustaka standar dan menyediakan berbagai fungsi untuk berinteraksi dengan sistem operasi serta melakukan operasi file. Ini membuatnya menjadi alat yang sangat penting bagi programmer Python. Modul `os` tidak perlu diinstal secara terpisah dan menyediakan fungsionalitas yang luas serta mudah digunakan, menjadikannya pilihan yang tepat untuk berbagai tugas terkait file dan sistem.

Berikut contoh sederhana penyimpanan file, sebagai berikut :

```
Contoh.py > ...
1  with open("output.txt", "w") as file:
2      file.write("Halo, dunia!\n")
3      file.write("Ini adalah contoh teks.")
4
5  try:
6      with open("output.txt", "x") as file:
7          # Menyimpan teks ke dalam file
8          file.write("Halo, dunia!\n")
9          file.write("Ini adalah contoh teks.")
10
11 except FileExistsError:
12     print("File sudah ada.")
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe c:/Tugas/Contoh.py
File sudah ada.
```

Dalam contoh di atas, kita menggunakan fungsi `open()` untuk membuka file "output.txt" dalam mode menulis ("w") atau mode tambahan menulis ("x"). Jika file sudah ada dalam mode "x", maka akan menimbulkan `FileExistsError`, sedangkan dalam mode "w" akan menghapus isi file tersebut jika sudah ada sebelumnya.

## BAGIAN 2: LATIHAN MANDIRI

### SOAL 8.1

- Source Code :

```
Latihan 8,1.py > ...
1 def bandingkan_file(file1, file2):
2     with open(file1, 'r') as f1, open(file2, 'r') as f2:
3         for i, (line1, line2) in enumerate(zip(f1, f2), start=1):
4             if line1 != line2:
5                 print(f"Perbedaan pada baris {i}:")
6                 print(f"File 1: {line1.strip()}")
7                 print(f"File 2: {line2.strip()}")
8
9         print("Tidak ada perbedaan antara kedua file." if i == 0 else "")
10
11 bandingkan_file('file1.txt', 'file2.txt')
```

- Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe "c:/Tugas/Latihan 8,1.py"
Perbedaan pada baris 1:
File 1: Hallo
File 2: Nama saya Marvin
```

- Penjelasan :

Fungsi `bandingkan\_file` membandingkan isi dua file baris per baris, kemudian menampilkan perbedaan pada baris yang tidak sama beserta isi dari kedua file tersebut, dan mengeluarkan pesan bahwa tidak ada perbedaan jika kedua file identik.

## SOAL 8.2

- Source Code :

```
Latihan 8,2.py > main
1 def load_soal(file_path):
2     with open(file_path, 'r') as file:
3         return [tuple(map(str.strip, line.split("||"))) for line in file]
4
5 def main():
6     nama_file = "soal.txt"
7     soal = load_soal(nama_file)
8
9     for idx, (pertanyaan, jawaban_benar) in enumerate(soal, start=1):
10        print(pertanyaan)
11        jawaban = input("Jawab: ").strip().lower()
12        if jawaban == jawaban_benar.lower():
13            print("Jawaban benar!")
14        else:
15            print("Jawaban salah!")
16        print()
17
18 if __name__ == "__main__":
19     main()
```

- Output :

```
Kota gudeg adalah:
PS C:\Tugas> & C:/Users/ASUS/AppData/Local/Programs/Python/Python312/python.exe "c:/Tugas/Latihan 8,2.py"
1+1 =
Jawab: 2
Jawaban benar!

Bendera Indonesia?
Jawab: Merah Putih
Jawaban benar!

Kota gudeg adalah:
Jawab: Yogyakarta
Jawaban benar!

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

50 * 20 =
Jawab: 1000
Jawaban benar!
```

- Penjelasan :  
Kode ini membaca file teks `soal.txt` yang berisi pertanyaan dan jawaban, kemudian menampilkan pertanyaan, meminta jawaban dari pengguna, dan memberikan respons apakah jawaban tersebut benar atau salah.
- Link GitHub :  
<https://github.com/Jamesmarvins/Tugas-Alpro-8.git>