



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71220895
Nama Lengkap	Drestanta Dipta Jalu Prakasya
Minggu ke / Materi	09 / 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

Pada pemograman Python, list adalah rangkaian nilai yang dapat diakses dengan satu nama. String dan list berbeda, karena string adalah rangkaian karakter. List terdiri dari karakter, integer, float, dan tipe data lainnya. Sementara string dan list tidak dapat diubah, list dapat diubah secara langsung. Dalam list, dua string yang memiliki isi yang sama menunjuk pada objek yang sama, sedangkan dalam list, dua string yang memiliki isi yang sama menunjuk pada objek yang berbeda.

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:

```
Laprak 8.py > ...
1  with open('mbox-short.txt') as handle:
2      count = 0
3      for line in handle:
4          count = count + 1
5      print('Line Count:', count)
6
7
8  with open('mbox-short.txt') as handle:
9      hasil = handle.read()
10     print("Ukuran: ", len(hasil), "bytes")
11     print("Huruf dari belakang sendiri mundur 16 huruf adalah: " + hasil[-16::1])
12
13
14  with open('mbox-short.txt') as handle:
15      count = 1
16      for line in handle:
17          if line.startswith("Date:") and count < 10:
18              count += 1
19              print(line.rstrip())
```

Output:

```
Line Count: 1910
Ukuran: 94626 bytes
Huruf dari belakang sendiri mundur 16 huruf adalah: > Preferences.

Date: Sat, 5 Jan 2008 09:12:18 -0500
Date: 2008-01-05 09:12:07 -0500 (Sat, 05 Jan 2008)
Date: Fri, 4 Jan 2008 18:08:57 -0500
Date: 2008-01-04 18:08:50 -0500 (Fri, 04 Jan 2008)
Date: Fri, 4 Jan 2008 16:09:02 -0500
Date: Fri, 4 Jan 2008 16:09:02 -0500
Date: 2008-01-04 16:09:01 -0500 (Fri, 04 Jan 2008)
Date: Fri, 4 Jan 2008 15:44:40 -0500
Date: 2008-01-04 15:44:39 -0500 (Fri, 04 Jan 2008)
Date: Fri, 4 Jan 2008 15:01:38 -0500
PS C:\Users\User\Downloads\Laprak 8> █
```

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:

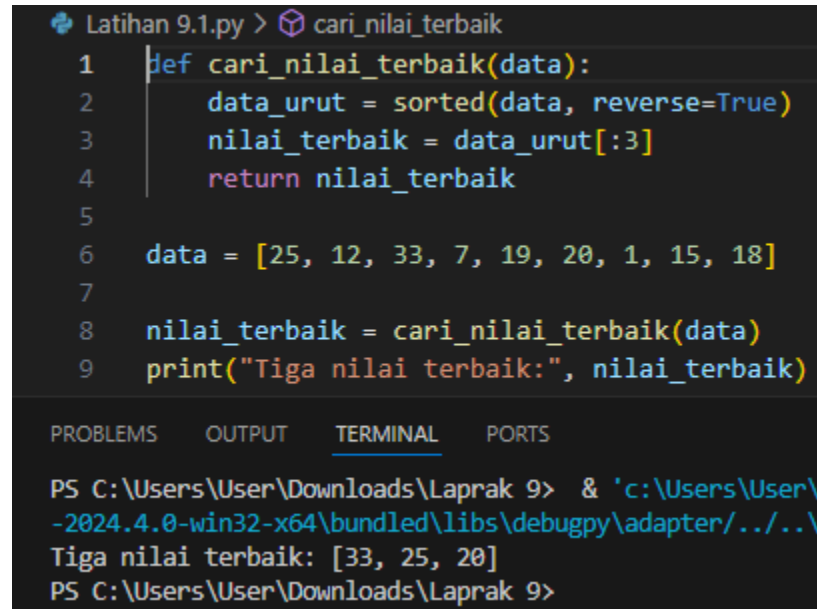
```
handle = open('output.txt', 'w')
tulisan = "teks ini akan dituliskan ke file\n "
handle.write(tulisan)
handle.close()
```

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

Dalam contoh di atas, fungsi cari_nilai_terbaik() mengumpulkan daftar data, mengurutkannya secara menurun, dan mengambil tiga nilai tertinggi. Selanjutnya, kita mencetak nilai terbaik yang ditemukan. Anda memiliki kemampuan untuk mengubah data dengan list angka apa pun yang Anda inginkan.



```
Latihan 9.1.py > cari_nilai_terbaik
1 def cari_nilai_terbaik(data):
2     data_urut = sorted(data, reverse=True)
3     nilai_terbaik = data_urut[:3]
4     return nilai_terbaik
5
6 data = [25, 12, 33, 7, 19, 20, 1, 15, 18]
7
8 nilai_terbaik = cari_nilai_terbaik(data)
9 print("Tiga nilai terbaik:", nilai_terbaik)
```

PROBLEMS OUTPUT TERMINAL PORTS

```
PS C:\Users\User\Downloads\Laprak 9> & 'c:\Users\User\
-2024.4.0-win32-x64\bundled\libs\debugpy\adapter/../../\
Tiga nilai terbaik: [33, 25, 20]
PS C:\Users\User\Downloads\Laprak 9>
```

SOAL 2

```
Latihan 9.2.py > ...
1  numbers = []
2
3  while True:
4      input_number = input("Masukkan angka (ketik 'done' untuk selesai): ")
5      if input_number.lower() == 'done':
6          break
7      try:
8          numbers.append(float(input_number))
9      except ValueError:
10         print("Masukan tidak valid.")
11
12  if numbers:
13      print("Nilai maksimum:", max(numbers))
14      print("Nilai minimum:", min(numbers))
15  else:
16      print("Tidak ada angka yang dimasukkan.")
```

PROBLEMS OUTPUT TERMINAL PORTS

```
PS C:\Users\User\Downloads\Laprak 9> & 'c:\Users\User\AppData\Local\Programs\Python\Python312\python.exe' -2024.4.0-win32-x64\bundled\libs\debugpy\adapter\..\..\debugpy\launcher '52753' '--' 'c:\Users\User\Downloads\Laprak 9.py'
Masukkan angka (ketik 'done' untuk selesai): 11
Masukkan angka (ketik 'done' untuk selesai): 22
Masukkan angka (ketik 'done' untuk selesai): 33
Masukkan angka (ketik 'done' untuk selesai): 44
Masukkan angka (ketik 'done' untuk selesai): 55
Masukkan angka (ketik 'done' untuk selesai): e
Masukan tidak valid.
Masukkan angka (ketik 'done' untuk selesai): done
Nilai maksimum: 55.0
Nilai minimum: 11.0
PS C:\Users\User\Downloads\Laprak 9> |
```

Dalam contoh di atas, fungsi `cari_nilai_terbaik()` mengumpulkan daftar data, mengurutkannya secara menurun, dan mengambil tiga nilai tertinggi. Selanjutnya Ketika user mengetik 'done' maka program selesai, program mencetak nilai terbaik yang ditemukan. User memiliki kemampuan untuk mengubah data dengan list angka apa pun yang user inginkan.

SOAL 3

```
Latihan 9.3.py > ...
1  with open('file.txt', 'r') as file:
2      lines = file.readlines()
3
4  unique_words = []
5  print("====Kata Unik pada Berita====")
6  for line in lines:
7      words = line.split()
8      for word in words:
9          if word not in unique_words:
10             unique_words.append(word)
11
12  print(unique_words)
```

PROBLEMS OUTPUT **TERMINAL** PORTS Python Debug Console + - [] ... ^ X

```
PS C:\Users\User\Downloads\Laparak 9> & 'c:\Users\User\AppData\Local\Programs\Python\Python311\python.exe' 'c:\Users\User\.vscode\extensions\ms-python.debugpy-2024.4.0-win32-x64\bundled\libs\debugpy\adapter\..\..\debugpy\launcher' '53011' '--' 'c:\Users\User\Downloads\Laparak 9\Latihan 9.3.py'
====Kata Unik pada Berita====
['BERITA', 'di', 'media-media', 'online', 'saat', 'ini', 'sedikit', 'unik.', 'Suatu', 'siang', 'ketika', 'sedang', 'iseng', 'melakukan', 'scrolling', 'media', 'online', 'saya', 'menemukan', 'judul', 'berita', 'yang', 'menarik', 'perhatian', 'yaitu', 'tentang', 'uji', 'terbang', 'pesawat', 'KF-21', 'Boramae.', 'adalah', 'proyek', 'tempur', 'hasil', 'kolaborasi', 'pemerintah', 'Indonesia', 'dengan', 'Korea', 'Selatan.', 'Alangkah', 'terkejutnya', 'tersebut', 'menyebutkan', 'bahwa', 'kecepatan', 'mampu', 'menembus', '22', 'ribu', 'kilometer/jam.', 'Sontak', 'mengernyitkan', 'dahi', 'dan', 'bertanya', 'dalam', 'hati:', 'apa', 'mungkin', 'itu', 'bisa', '11', 'kali', 'lebih', 'cepat', 'dari', 'F-16', 'atau', 'pun', 'pada', 'umumnya?', 'Mesin', 'macam', 'digunakan', 'itu?', 'Setelah', 'menelaah', 'jauh', 'rata-rata', 'ternyata', 'memang', 'hanya', 'sekitar', '2.000', 'km/jam.', 'Begitu', 'mengetahui', 'adanya', 'kekeliruan', 'artikel', 'itu', 'merupakan', 'seorang', 'pewira', 'Tentara', 'Nasional', '(TNI)', 'Angkatan', 'Udara', '(AU)', 'tertawa', 'sekaligus', 'terheran-heran.']
PS C:\Users\User\Downloads\Laparak 9>
```

Kode tersebut membuka file.txt, membaca setiap barisnya, membagi setiap baris menjadi kata-kata terpisah, dan mencatat setiap kata yang berbeda di dalam file. Setelah itu, daftar kata-kata yang unik dicetak. Versi yang lebih sederhana ini menggunakan metode extend() untuk menambah kata-kata dari setiap baris langsung ke dalam list unique_words, dan kemudian menggunakan fungsi set() untuk menghilangkan duplikat kata sebelum mencetaknya.