

# The Document

## Lab 4 - Dasar-Dasar Pemrograman 1 Gasal 2020/2021

Catatan: **Revisi ditandai dengan highlight berwarna kuning.**



Sumber: <https://spongebob.fandom.com/>

Mang Oleh adalah seorang pekerja kantoran. Pada suatu hari, Mang Oleh mendapatkan sebuah pekerjaan unik dari bos-nya. Mang Oleh diberikan banyak dokumen untuk dikerjakan. Pada setiap dokumen terdapat banyak baris dengan jumlah karakter yang berbeda-beda pada setiap barisnya. Bos memintanya untuk **menghitung jumlah baris, karakter, serta rata-rata jumlah karakter tiap baris** yang ada di dokumen tersebut.

Karena hari sudah menjelang sore, Mang Oleh ingin menggunakan sebuah program untuk membantu menyelesaikan pekerjaan tersebut sebelum larut malam. Program tersebut tentunya harus dapat menyelesaikan tugas yang diminta oleh Bos. Tetapi karena tidak mengerti *programming*, Mang Oleh meminta kamu –odading

kesayangannya— untuk membuat program tersebut. Kamu dapat membuat program menggunakan bahasa Python dan dengan materi **Files I/O & Exception**.

## Pengantar Files I/O & Exception

Pada Python, selain menggunakan console untuk menerima input user, kita dapat **membaca input dari sebuah file** dan **menuliskan outputnya ke sebuah file** pula.

Terdapat beberapa fungsi yang dapat digunakan untuk membaca dan mencetak teks ke dalam sebuah file.

### 1. Membuka dan menutup file

- **open()** dan **.close()**

Untuk membuka isi dari sebuah file, kita dapat menggunakan perintah:

```
in_file = open("nama_file", "r")
```

Perintah `open("nama_file", "r")` akan membuka file dengan nama `nama_file` dan menyimpannya pada variabel `in_file` untuk dapat kita gunakan.

Argumen ke-2 dari `open()` adalah mode yang kita gunakan pada saat mengakses sebuah file.

- "r" digunakan jika kita ingin membaca
- "w" jika ingin menulis, dan
- beberapa mode lainnya yang dapat kamu eksplorasi sendiri :D

Setelah selesai menggunakan suatu file, maka sebaiknya kita menutupnya supaya tidak terjadi hal-hal yang tidak diinginkan, dengan perintah:

```
in_file.close()
```

- **with open() as :**

Untuk membuat proses buka dan tutup file terjadi secara otomatis, Python menyediakan perintah sebagai berikut:

```
with open("nama_file", "r") as in_file:
```

```
# in_file hanya bisa diakses di dalam blok ini  
# in_file sudah tertutup dan tidak dapat diakses lagi
```

sehingga kita tidak perlu memanggil perintah `in_file.close()` lagi.

## 2. Membaca isi dari sebuah file

- **`in_file.read()`**

Perintah ini akan mengembalikan string berupa isi suatu file dari awal sampai akhir, **termasuk karakter `\n` (*newline*)** yang digunakan untuk memisahkan antar baris dalam file.

```
isi_file = in_file.read()
```

- **`in_file.readlines()`**

Perintah ini akan mengembalikan list yang berisi string setiap baris, **termasuk karakter `\n` (*newline*)** pada akhir string setiap baris. Dengan begitu, kita bisa menggunakan for-loop pada variabel `isi_file` untuk mengaksesnya baris-perbaris.

```
isi_file = in_file.readlines()
```

- **Dengan for**

Selain menggunakan for pada `in_file.readlines()`, kita juga bisa langsung melakukan looping ada `in_file`, seperti:

```
for line in in_file:  
    # line akan berisi tiap baris di in_file (termasuk  
    \n)
```

## 3. Menulis text ke sebuah file

Untuk menulis string pada sebuah output file terpisah. Perintahnya:

```
out_file.write("suatu_nilai")
```

Atau, kita bisa menggunakan print dengan tambahan argumen, seperti:

```
print("suatu_nilai", file=out_file)
```

## Exception

Python memiliki suatu exception yang akan di-raise ketika terdapat error yang terjadi pada program. Exception tersebut dapat di handle menggunakan try, except, else dan finally statement

### try-except-else-finally

```
try :  
    # Baris-baris kode yang berpotensi menghasilkan Exception  
    raise Exception("Ini Pesan Exception")  
except Exception as e:  
    print("Terdapat Exception dengan pesan", e)  
else : # Opsional  
    print("Program berjalan tanpa exception")  
finally : # Opsional  
    print("Dijalankan baik dengan maupun tanpa exception")
```

Selain error yang diakibatkan oleh program, terdapat juga cara yang dapat digunakan untuk menghasilkan sebuah sinyal error dengan sengaja yaitu dengan melakukan raise [JenisException] ( jenis exception dapat berupa : Exception, TypeError, ValueError, etc ).

### raise Exception()

```
raise Exception("Ini Pesan Exception")  
# Akan di-raise Exception dengan pesan "Ini Pesan Exception"
```

Beberapa jenis exception yang terdapat di Python adalah

1. **AssertionError**: Dipanggil ketika tidak memenuhi statement assert (statement assert bernilai False)
2. **TypeError**: Dipanggil ketika tipe data tidak sesuai
3. **ValueError**: Dipanggil ketika value / nilai data tidak sesuai

4. **ZeroDivisionError**: Dipanggil ketika terdapat pembagian dengan angka nol
5. **NameError**: Dipanggil ketika suatu nama variabel tidak dikenal / belum diinisialisasi

## Task 1: Average and Character Counter (File: lab04.py)

Menggunakan template yang sudah diberikan dibawah, hitung banyaknya karakter pada setiap baris dan hitung rata-rata jumlah karakter per baris. Karakter yang harus dihitung **termasuk spasi, tab, dan special character lainnya selain karakter newline (\n).**

Hal yang dijalankan oleh program :

1. Menerima input berupa nama file input, lalu **membaca isi dari file tersebut.**
2. **Hitunglah Jumlah Minimal, Maximal, dan Range dari jumlah karakter pada setiap baris di textfile.**
3. Kamu diminta untuk meng-*handle exception* yang mungkin terjadi, seperti:
  - a. Error ketika nama file input dan file output sama, gunakan **assert**
  - b. Error ketika file input tidak ditemukan
4. **Menulis output program** ke dalam sebuah **file yang sama dengan input.**  
**Output harus meng-*append* dalam berkas yang sama dengan input.**

**Format yang di gunakan adalah sebagai berikut:**

```
#####  
Min: {xx} karakter  
Max: {xx} karakter  
Range: {xx} karakter
```

**Bila tidak ada baris pada berkas input, maka akan dituliskan **NULL****

```
#####  
NULL
```

Silakan **mengunduh template dan contoh file input** (template lab04.py, in0.txt, in1.txt, in2.txt) untuk Task 1 melalui [tautan ini](#).

**Pastikan file input dan programmu berada pada folder yang sama.**

### Contoh interaksi 1:

#### Input Console:

Masukkan nama file input: `in0.txt`

#### Expected console output:

Output berhasil ditulis pada `in0.txt`  
Program selesai. Tekan enter untuk keluar...

#### Expected isi file `in0.txt` setelah program dijalankan:

```
kata orang
aku berbeda
tapi tak apa
#####
Min: 10 karakter
Max: 12 karakter
Range:2 karakter
```

### Contoh interaksi 2:

#### Input Console:

Masukkan nama file input: `in1.txt`

#### Expected console output:

Output berhasil ditulis pada `in1.txt`  
Program selesai. Tekan enter untuk keluar...

#### Expected isi file `in1.txt` setelah program dijalankan:

```
cLoieqjIttXpfKGQiSyJQqnPeLkSCemUv
ZvveIjzUSMYBIBvvqlEXnCjQWnsixlQydezEaiRp
DoANONqskcebgJidtTyfQlkNeYFembMkiUc
EzkEWsldTutXGYVwaw
bDjJDTkGrfhPkieuECTXdEGLEwsczqGndxEOM
DVuYlGobEWuhQpMBLvzi
RUjcUthYuW
QqCTqwQYIKQBwLkacNrJAOLlyKnMCIEyBn
bvJVCpDtUwBKNVFKeUOXMqLGyludLuvFKDUuTRGHcEYUEkbpAJQrWx
sEmSSShEnDodSRF
CixpaovbDEFqZKLKhKXyRHTanZCsMM
tAeJoguaBqUUguOZmCup
meNoWlgYcwZMGwqyYWDbaZdrloSMMLROoEXQrFDL
tCaBwaAvUDHYtFlURAzCgJELdNeWXDNrNFgCjuTzpYnjTbEkdOkFjcrvCunZRqHA
zdMihXZVmT
```

```
tnvfKMVTEisFMdSamXCy
nPxJYZuVmyHoMjLyBcDc
QHovmeROqplEqnjDWOKBKdxlAkxheNJagGDCzjmmiQMzlbPOgYebL
ecFjTVRjxRQPtjUyKhhziIzjdcleUKAfyjFPTWff
keiCjqkoiMoVrGEkwgaiDapURxYUprvinJMmr
qCZhVIUzVZkxLWDoyFis
jtkSRKRCFGnZXvu
kvjhncOgVpuD
AvQIFDyTbVRnVbxcPFfI
dNUUQoEfmqVkfVtIshpn
MlQYlCZ
XASeSACoq
ffGYOmUIHpnEtnoWn
fjdvizYkoHEqZSwsao
vLV
SisoRDHeslAhWQVPwIic
#####
Min: 3 karakter
Max: 74 karakter
Range: 71 karakter
```

### Contoh interaksi 3:

#### Input Console:

Masukkan nama file input: `in2.txt`

#### Expected console output:

Output berhasil ditulis pada `in2.txt`  
Program selesai. Tekan enter untuk keluar...

#### Expected isi file `in2.txt` setelah program dijalankan:

```
BRhNWmOVEkEvkU
DwhhJfvUd
rTJZUHALgZEPvFGIAAJYxtbVcqcKUteZVn
zPWxAbtPOexZITpzUQTn
xeAGXuYuQUDLFWWLtN
FsIWTTDSIEXwIueFFjIRXEAGCpuQrgahO
lzaNDDRipfohkALungYAipuOhbWIpkmsSqvxkjRYNxCADTUKzS
BYeFiJtrtUlyo
TWgKUakYCiOjpsm
gpooTTkWPQAQmOJelfHt
aQLi
UIZuBbvOBmaMxUjFfliAATubKNDXeyjGntBnefA
vFXGjHdPoyWmQdfUekjhqnLYLoQtS
SghjMseTmeLajxFgSjXPptOejMaITmDINywQpDHF
cBzKULAKcZZlFNUJzDwN
eblxPGOtYYpYNhps
```



ORmSFTqyac

#####

Min: 4 karakter

Max: 50 karakter

Range: 46 karakter

#### Contoh interaksi 4:

##### Input Console:

Masukkan nama file input: `some_file_that_doesnt_exist.txt`

##### Expected Console Output:

File tidak ditemukan :(  
Program selesai. Tekan enter untuk keluar...

## DISCLAIMER:

- Input file dijamin mengandung newline di akhir file.
- Kamu cukup mengumpulkan **program Python kamu (lab04.py)** dan **tidak perlu** mengumpulkan file output yang akan di-*generate* oleh program Python kamu.
- Pastikan penamaan file sesuai dengan ketentuan:

**[Nomer Kelompok]\_[Nama]\_[NPM]\_lab4.zip** yang berisi file Python lab04.py (Contoh: **03\_DarrenNgoh\_1906350723\_lab4.zip**)

### Challenge : Word Counter (File:challenge.py) (Opsional)

Hitunglah jumlah kata yang ada dalam file tersebut. Ingat bahwa **kata dipisahkan oleh spasi**.

**\* Contoh menggunakan input file in0.txt yang telah disediakan**

#### Input Console:

```
Masukkan nama file input: in0.txt  
Masukkan nama file output: out0.txt
```

#### Expected console output:

```
Output berhasil ditulis pada out0.txt  
Program selesai. Tekan enter untuk keluar...
```

#### Expected isi file out0.txt:

```
Terdapat 3 baris, 7 kata, dan 33 karakter pada file in0.txt  
Rata-rata baris mengandung 2.33 kata dan 11.00 karakter
```