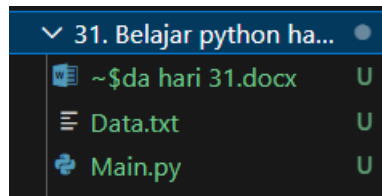


Pada hari 31, saya belajar bagaimana cara membaca, membuat, mengexport file external (.txt) di python. Semisal kita punya file txt yang isinya seperti di bawah

data.txt

```
Acep Nurmagedov
Ojos Surojos
Adi Mangento
```

Dan struktur filenya seperti dibawah (file txt setara dengan file .py-nya)



Nah jadi cara menaruh sebuah data (txt) ke dalam variabel caranya seperti dibawah

```
File = open("data.txt", mode = "r")
#mode ini menentukan data ini sebagai apa? Apakah read atau write
#kalo gk mau error mending di open(...) itu masukin filepathnya aja sekalian
#nah cara printnya
print(file.read())
```

#outputnya seperti dibawah

#pastikan saat memasukkan data.txtnya , directorynya sudah sesuai jgn kyk dibawah

```
PS C:\Users\bgusr\Belajar-Python-kelas-terbuka> & C:\Users\bgusr\anaconda3\envs\cv_env\python.exe "c:/Users/bgusr/Belajar-Python-kelas-terbuka/31. Belajar python hari 31/Main
.py"
Traceback (most recent call last):
  File "c:\Users\bgusr\Belajar-Python-kelas-terbuka\31. Belajar python hari 31\Main.py", line 1, in <module>
    file = open("data.txt", mode="r")
          ^^^^^^^^^^^^^^^^^^^^^^^^^^
FileNotFoundError: [Errno 2] No such file or directory: 'data.txt'
```

#tapi seperti dibawah ini

```
PS C:\Users\bgusr\Belajar-Python-kelas-terbuka> cd ".\31. Belajar python hari 31\"
PS C:\Users\bgusr\Belajar-Python-kelas-terbuka\31. Belajar python hari 31> & C:\Users\bgusr\anaconda3\envs\cv_env\python.exe "c:/Users/bgusr/Belajar-Python-kelas-terbuka/31.
Belajar python hari 31/Main.py"
Acep Nurmagedov
Ojos Surojos
Adi Mangento
```

Nah cara untuk mengetahui data bisa di read atau tidak; atau di write atau tidak menggunakan .readable() atau .writeable() , contohnya:

```
print(f'Status Read: {file.readable()}')
print(f'Status Write: {file.writable()}')
```

#output

```
PS C:\Users\bgusr\Belajar-Python-kelas-terbuka> cd ".\31. Belajar python hari 31\"
PS C:\Users\bgusr\Belajar-Python-kelas-terbuka\31. Belajar python hari 31> & C:\Users\bgusr\anaconda3\envs\cv_env\python.exe "c:/Users/bgusr/Belajar-Python-kelas-terbuka/31.
Belajar python hari 31/Main.py"
Acep Nurmagedov
Ojos Surojos
Adi Mangento

Status Read: True
Status Write: False
```

Adapun cara print yang lain, salah satunya .readline(), bacanya perbaris

```

file = open("data.txt", mode="r")
#mode ini menentukan data ini sebagai apa? Apakah read atau write
#nah cara printnya
print(file.read())
print(f"Status Read: {file.readable()}")
print(f"Status Write: {file.writable()}")

print(file.readline())

```

Bisa dilihat hasil eksekusi print(file.readline()) tidak muncul itu dikarenakan sebelumnya sudah dibaca di print(file.read())

```

PS C:\Users\bgusr\Belajar-Python-kelas-terbuka\31. Belajar python hari 31> & C:/U
Belajar python hari 31/Main.py"
Acep Nurmagedov
Ojos Surojos
Adi Mangento

Status Read: True
Status Write: False

PS C:\Users\bgusr\Belajar-Python-kelas-terbuka\31. Belajar python hari 31> 

```

Contoh benarnya

```

file = open("data.txt", mode="r")
#mode ini menentukan data ini sebagai apa? Apakah read atau write
#nah cara printnya
#print(file.read())
print(f"Status Read: {file.readable()}")
print(f"Status Write: {file.writable()}")

print(file.readline()) #baris pertama
print(file.readline()) #baris kedua

```

Hasilnya

```

PS C:\Users\bgusr\Belajar-Python-kelas-terbuka\31. Belajar python hari 31> & C:/U
Belajar python hari 31/Main.py"
Status Read: True
Status Write: False
Acep Nurmagedov

Ojos Surojos

PS C:\Users\bgusr\Belajar-Python-kelas-terbuka\31. Belajar python hari 31> 

```

Adapun cara read yang lain menggunakan .readlines(), yang mana membaca semua baris sebagai list, contohnya:

```

file = open("data.txt", mode="r")
#mode ini menentukan data ini sebagai apa? Apakah read atau write
#nah cara printnya
#print(file.read())
print(f"Status Read: {file.readable()}")
print(f"Status Write: {file.writable()}")

#print(file.readline()) #baris pertama
#print(file.readline()) #baris ketiga

```

```
print(file.readlines())
```

Hasilnya

```
Belajar python hari 31/Main.py"
Status Read: True
Status Write: False
['Acep Nurmagedov\n', 'Ojos Surojos\n', 'Adi Mangento\n']
PS C:\Users\bgusr\Belajar-Python-kelas-terbuka\31. Belajar python hari 31> █
```

Bisa dilihat dihasilnya ada \n (enter), nah itu alasan mengapa pada saat menggunakan .readline() hasilnya ada enter. Nah cara agar agar gk ada enternya di .readline() dibikin ada end kyk dibawah print(file.readline(), end="")

karena defaultnya readline itu adalah \n

Adapun jgn lupa setiap menggunakan fungsi .open()

Wajib diakhiri dengan .close, kalau tidak itu akan bisa menyebabkan memori akan terus terpakai ataupun resikonya file txt bisa corrupt. Contoh penggunaannya seperti dibawah:

```
file = open("data.txt", mode="r")
#mode ini menentukan data ini sebagai apa? Apakah read atau write
#nah cara printnya
#print(file.read())
print(f"Status Read: {file.readable()}")
print(f"Status Write: {file.writable()}")

print(file.readline(), end="") #baris pertama
print(file.readline()) #baris ketiga

#print(file.readlines())
file.close() #Menutup file
print(file.closed) #Mengecek apakah sudah tertutup
```

Hasilnya True

```
PS C:\Users\bgusr\Belajar-Python-kelas-terbuka\31. Belajar python hari 31> & C:/U
Belajar python hari 31/Main.py"
Status Read: True
Status Write: False
Acep Nurmagedov
Ojos Surojos

True
PS C:\Users\bgusr\Belajar-Python-kelas-terbuka\31. Belajar python hari 31> █
```

Nah agar lebih mudah dan langsung tertutup tidak perlu menggunakan file.close caranya menggunakan with, contohnya:

```
file = open("data.txt", mode="r")
#mode ini menentukan data ini sebagai apa? Apakah read atau write
#nah cara printnya
#print(file.read())
print(f"Status Read: {file.readable()}")
print(f"Status Write: {file.writable()}")

print(file.readline(), end="") #baris pertama
print(file.readline()) #baris ketiga

#print(file.readlines())
file.close() #Menutup file
```

```

print(f"file ditutup : {file.closed}") #Mengecek apakah sudah tertutup

print('Membaca dengan with'.center(50, "="))
with open("data.txt", mode="r") as file: #artinya dengan saat ini membuka file data.txt sebagai file,
    apa yang akan kita lakukan (setelah tanda:)
    content = file.readline()
    print(content)

print(f"file ditutup : {file.closed}")

```

Hasilnya

```

PS C:\Users\bgusr\Belajar-Python-kelas-terbuka\31. Belajar
Belajar python hari 31/Main.py"
Status Read: True
Status Write: False
Acep Nurmagedov
Djos Surojos

file ditutup : True
=====Membaca dengan with=====
Acep Nurmagedov

file ditutup : True

```

Nah dalam open() modenya tidak hanya ada r saja tapia da yang lain berikut mode dan perbandingannya:

 **TABEL PERBANDINGAN:**

Mode	Jika File Tidak Ada	Jika File Sudah Ada
"r" (read)	✗ Error	✓ Baca file
"w" (write)	✓ Buat baru	✗ Overwrite
"a" (append)	✓ Buat baru	✓ Tambah di akhir
"x" (exclusive)	✓ Buat baru	✗ Error
"r+" (read+)	✗ Error	✓ Baca/tulis
"w+" (write+)	✓ Buat baru	✗ Overwrite
"a+" (append+)	✓ Buat baru	✓ Baca/tambah

r ini buat baca aja, kalo w ini dia bikin file baru dan juga bisa membuat isi dari file tersebut, contoh scriptnya:

```

# mode writ
# dia akan membuat file baru jika tidak ada,
# lalu akan menimpa atau overwrite isinya

with open("data_1.txt",'w',encoding="utf-8") as file:
    file.write("jon si jhonny")

with open("data_1.txt",'w',encoding="utf-8") as file:

```

```
file.write("ucup surucup")
```

```
with open("data_1.txt",'w',encoding="utf-8") as file:  
    file.write("overwrite")
```

Nah kalo write ini hasil isi file txtnya bukan perbaris kyk dibawahj:

```
jon si Jhonny  
ucup surucup  
overwrite  
# tapi ouputnya hanya  
overwrite
```

karena kalo masih dalam mode w ini hanya overwrite, cuman ngeganti isinya aja

nah kalo mau nambahin kyk:

```
jon si Jhonny  
ucup surucup  
overwrite
```

itu pake mode append (a). contoh scriptnya:

```
with open("data_2.txt",'w',encoding="utf-8") as file:  
    file.write("jon si jhonny \n")  
  
with open("data_2.txt",'a',encoding="utf-8") as file:  
    file.write("ucup surucup\n")  
  
with open("data_2.txt",'a',encoding="utf-8") as file:  
    file.write("tambah lagi dengan append\n")
```

Itu outputnya

```
jon si Jhonny  
ucup surucup  
tambah lagi dengan append
```

mode append itu untuk **menambahkan data di akhir file** tanpa menghapus data yang sudah ada.

Kalo mode r+(read + write) itu dia kayak semacam nimpa isi datanya sesuai dengan Panjang datanya, contoh scriptnya:

```
with open("data_3.txt",'w',encoding="utf-8") as file:  
    file.write("data ke 3\n")  
  
with open("data_3.txt",'r+',encoding="utf-8") as file:  
    file.write("baris-1 \n")  
    file.write("baris-2 \n")  
    file.write("baris-3 \n")  
  
with open("data_3.txt",'r+',encoding="utf-8") as file:  
    data = file.read()  
    print(data)  
  
with open("data_3.txt",'r+',encoding="utf-8") as file:  
    file.write("otong") # menimpa isi text sesuai dengan panjang data
```

hasil txt-nya:

```
otong-1  
baris-2  
baris-3
```