

# MODUL 10

## FILE dan DIREKTORI

### PRAKTIKUM PROGRAMAMA KOMPUTER

#### 2021

---

#### A. PENDAHULUAN

Baca dan tulis file adalah teknik dasar yang harus dipahami dalam pemrograman Python, karena banyak digunakan untuk pengolahan dan pemerosesan file. Paham cara membaca dan menulis file dengan Python akan membuatmu mampu membuat aplikasi yang bisa mengambil dan menyimpan data ke file. Selain itu juga, kamu akan lebih mudah memahami beberapa materi Python selanjutnya, seperti baca dan *parsing* file JSON, XML, CSV, XLS, dan sebagainya.

Ada banyak sekali tipe file pada komputer...seperti, dokumen, video, gambar, audio, arsip, dll.

Pada Python, file hanya dikelompokkan menjadi dua tipe:

1. **File Teks:** File yang berisi teks. Setiap baris teks memiliki EOL (*End of Line*).

Contoh: TXT, MD, CSV, JSON, dsb.

2. **File Binary:** File yang bukan teks, hanya bisa diproses oleh program tertentu yang memahami strukturnya.

Contoh: EXE, JPG, MKV, M4A, 3GP, dsb.

Pada tutorial ini, kita akan hanya belajar cara membaca dan menulis file teks saja.

Untuk file *binary*, mungkin nanti di kesempatan yang lain.

#### B. FILE DI PYTHON

Python sudah menyediakan fungsi Open () untuk membaca dan menulis file, Fungsi tersebut memiliki 2 parameter, yaitu nama file dan mode.



Resources: [www.petanikode.com](http://www.petanikode.com)

Objek file adalah variabel objek yang menampung isi file. Kita bisa melakukan pemrosesan file berkatnya.

Nama file bisa kita isi langsung apabila file-nya terletak dalam satu direktori dengan skrip python. Namun, apabila terletak di direktori yang berbeda, maka kita harus memberikan alamat *path* file-nya.

```
obj_file = open("/path/ke/file.txt", "r")
```

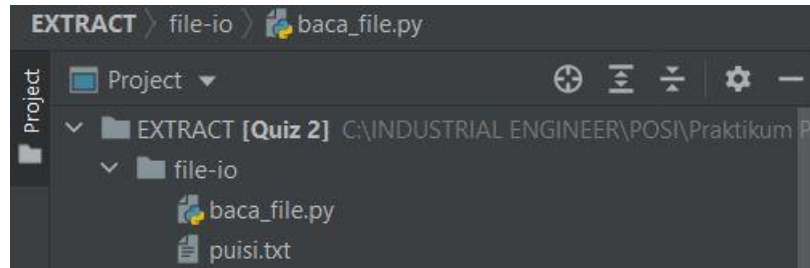
Kemudian untuk parameter *mode* yang fungsinya untuk menentukan hak akses terhadap file.

Ada beberapa mode yang tersedia:

Mode	Keterangan
"r"	hanya baca saja
"w"	akses untuk menulis file, jika file sudah ada, maka file akan di replace dan diganti dengan yang baru ditulis
"a"	digunakan untuk <i>append</i> atau menambah data ke file, artinya jika sudah ada data dalam file, maka akan ditambahkan dan tidak di-replace
"r+"	digunakan untuk membaca sekaligus menulis data ke file

- Praktik

Sekarang silahkan buat direktori bernama file-io, lalu di dalamnya buat file puisi.txt dan baca\_file.py.



Setelah itu bukalah file puisi.txt, dan isi dengan teks berikut:

```
Engkau bahkan bukan lagi variabel bagiku  
Engkau adalah konstanta abadi yang tak tergantikan...
```

1. Membaca File per Baris

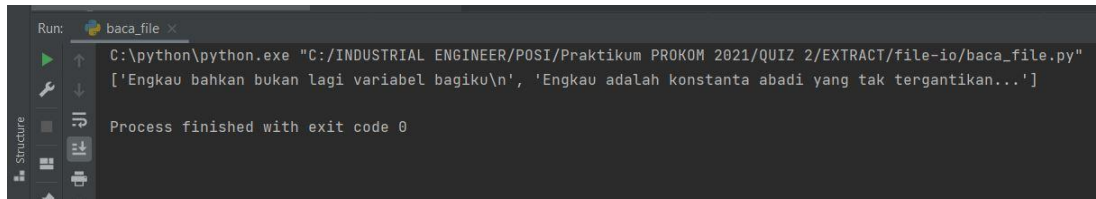
Selanjutnya kita akan mulai menulis kode programnya.

Silahkan buka file baca\_file.py kemudian ketik kode berikut.

```
# buka file  
file_puisi = open("puisi.txt", "r")  
  
# baca isi file  
Print(file_puisi.readlines())  
  
# tutup file  
file_puisi.close()
```

Pertama kita membuka file dengan fungsi `open ()`, selanjutnya mencoba membaca isinya per baris dengan *method* `readlines ()`, dan terakhir menutup file dengan `close ()` agar dihapus dalam memori.

Saat dieksekusi, kode di atas akan menghasilkan output berupa *list*, karena kita menggunakan *method* `readlines ()`.



```
Run: baca_file x
C:\python\python.exe "C:/INDUSTRIAL ENGINEER/POSI/Praktikum PROKOM 2021/QUIZ 2/EXTRACT/file-io/baca_file.py"
['Engkau bahkan bukan lagi variabel bagiku\n', 'Engkau adalah konstanta abadi yang tak tergantikan...']
Process finished with exit code 0
```

## 2. Membaca Semua teks dalam file

Kalau tadi kita membaca isi file per baris, sekarang kita coba baca semua teks menggunakan method `read()`.

Silahkan ikuti kode berikut:

```
# buka file
file_puisi = open("puisi.txt", "r")

# baca isi file
puisi = file_puisi.read()

# cetak isi file
print puisi

# tutup file
file_puisi.close()
```

Hasilnya ialah sebagai berikut:



```
petanikode@imajinasi ~/Desktop/py/file-io
petanikode@imajinasi ~/Desktop/py/file-io $ python baca_file2.py
Engkau bahkan bukan lagi variabel bagiku
Engkau adalah konstanta abadi yang tak tergantikan...
Tak ada yang bisa memisahkan kita

petanikode@imajinasi ~/Desktop/py/file-io $
```

“Apa bedanya method `read()` dengan `readlines()`?”

Bedanya:

Method `read()` membaca seluruh teks dan akan mengembalikan nilai string. Sedangkan `readlines()` membaca isi file per baris dan akan mengembalikan

nilai berupa *list*. Perlu diketahui, method `read ()` dan `readlines ()` hanya sekali pakai. Artinya, hanya dieksekusi sekali saja.

Eksekusi pertama, dia akan mengembalikan nilai berdasarkan isi filenya.

Eksekusi kedua, dia akan mengembalikan nilai kosong.

### 3. Cara Menulis File di Python

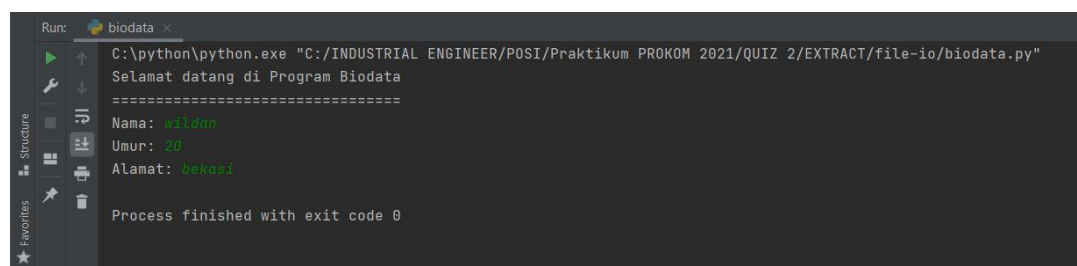
Seperti yang sudah kita ketahui, ada tiga mode yang digunakan bila ingin menulis file, yaitu: "w", "a", dan "r+".

#### Menulis dengan mode “w”

Silahkan buat program baru, kemudian ikuti kodenya seperti ini:

```
1 print("Selamat datang di Program Biodata")
2 print("=====")
3
4 # Ambil input dari user
5 nama = input("Nama: ")
6 umur = input("Umur: ")
7 alamat = input("Alamat: ")
8
9 # format teks
10 teks = "Nama: {}\nUmur: {}\nAlamat: {}".format(nama, umur, alamat)
11
12 # buka file untuk ditulis
13 file_bio = open("biodata.txt", "w")
14
15 # tulis teks ke file
16 file_bio.write(teks)
17
18 # tutup file
19 file_bio.close()
```

Setelah itu, coba eksekusi programnya.



```
Run: biodata x
C:\python\python.exe "C:/INDUSTRIAL ENGINEER/POSI/Praktikum PROKOM 2021/QUIZ 2/EXTRACT/file-io/biodata.py"
Selamat datang di Program Biodata
=====
Nama: wildan
Umur: 20
Alamat: bekasi
Process finished with exit code 0
```

Maka sekarang kita punya file baru bernama biodata.txt, file tersebut akan terbuat secara otomatis.

## Menyisipkan data ke File

Apabila kita tidak ingin menulis ulang atau menindih file yang sudah ada, kita bisa menggunakan mode "a" (*append*) untuk menulisnya.

```
1 print_("Selamat datang di Program Biodata")
2 print_("=====")
3
4 # Ambil input dari user
5 nama = input("Nama: ")
6 umur = input("Umur: ")
7 alamat = input("Alamat: ")
8
9 # format teks
10 teks = "Nama: {}\nUmur: {}\nAlamat: {}".format(nama, umur, alamat)
11
12 # buka file untuk ditulis
13 file_bio = open("biodata.txt", "a")
14
15 # tulis teks ke file
16 file_bio.write(teks)
17
18 # tutup file
19 file_bio.close()
```

Setelah itu eksekusi dengan biodata yang berbeda:

```
C:\python\python.exe "C:/INDUSTRIAL ENGINEER/POSI/Praktikum PROKOM 2021/QUIZ 2/EXTRACT/file-i
Selamat datang di Program Biodata
=====
Nama: adit
Umur: 20
Alamat: tangerang

Process finished with exit code 0
```

Setelahnya dapat dilihat di bagian biodata.txt

```
Hotel.py x baca_file.py x bioda
1 Nama: wildan
2 Umur: 20
3 Alamat: bekasi
4
5 Nama: adit
6 Umur: 20
7 Alamat: tangerang
```

## C. DIREKTORI PADA PYTHON

### 1. Membuat Direktori

Dalam python, pembuatan direktori baru dilakukan dengan fungsi `mkdir()`.

Bentuk umum penggunaan:

`Os.mkdir>NamaDirektori)`

Berikut ialah contoh yang menunjukkan penggunaan fungsi `mkdir()`

```
1      #Nama_File: mkdir.py
2
3      import os
4
5      def main():
6          os.mkdir("unit")
7
8      ifname = "main"
9
10     main()
```

Selanjutnya coba cek akan ada direktori baru bernama “unit” didalam direktori anda.

### 2. Mengubah Nama Direktori

Untuk mengubah nama direktori, kita perlu menggunakan fungsi `rename()` sama seperti pada saat kita mengubah nama file.

Bentuk umum penggunaan:

`Os.rename>NamaDirektori,>NamaDirektoriBaru)`

Berikut ialah contoh yang menunjukkan penggunaan fungsi `rename()`

```
1      #Nama_File: mkdir.py
2
3      import os
4
5      def main():
6          os.rename("unit", "modul")
7
8      ifname = "main"
9
10     main()
```

Selanjutnya coba cek file direktori unit berubah menjadi “modul” .

### 3. Menghapus Direktori

Untuk mengubah nama direktori, kita perlu menggunakan fungsi `rmdir()`

Bentuk umum penggunaan:

`Os.rmdir>NamaDirektori)`

Contohnya ialah sebagai berikut:

```
1  #Nama_File: mkdir.py
2
3  import os
4
5  def main():
6      os.rmdir("modul")
7
8  if __name__ == "__main__":
9
10     main()
```

Dan direktori “Main” sudah tidak ada lagi ketika anda menjalankan program tersebut

---

### TUGAS!

1. Buatlah Direktori menggunakan fungsi `mkdir()`
2. Buatlah suatu program Biodata diri (gunakanlah mode “w” seperti modul diatas) = Output berupa file.txt
3. Gunakanlah mode “a”/ append untuk menambahkan 1 biodata teman anda
4. Rename file direktori menggunakan fungsi `rename()` {rename dengan nama\_nim anda}

#### Resources

Raharjo, B. (2019). *Mudah belajar Python Untuk Aplikasi Desktop dan Web*. Bandung: Informatika.

*Tutorial Git #2: Cara Membuat Repositori Baru dalam Proyek*. (2021, April 19).

Retrieved from [www.petanikode.com](https://www.petanikode.com/git-init/): <https://www.petanikode.com/git-init/>



