

# Plagiarism Scan Report

## Summary

Report Generated Date	27 Aug, 2017
Plagiarism Status	<b>100% Unique</b>
Total Words	530
Total Characters	3579
Any Ignore Url Used	

## Content Checked For Plagiarism:

### Python File I/O

Unit input adalah (masukan) suatu data yang berbentuk document bait itu data data foto, data data huruf ataupun data tanggal ke dalam sebuah system yang terkomputerisasi. unit output (keluaran) biasanya digunakan untuk menampilkan data, atau dengan kata lain untuk menangkap data yang telah diinputkan terlebih dahulu dalam sebuah penyimpanan media elektronik , contohnya data yang akan ditampilkan pada layar monitor atau printer. I/O Input/Output Read [IOR] dan untuk tulis I/O Input/Output Write [IOW].

File adalah lokasi bernama pada disk untuk menyimpan informasi terkait. Ini digunakan untuk menyimpan data secara permanen dalam memori non-volatile (misalnya hard disk).

Karena, random access memory (RAM) bersifat volatile sehingga kehilangan datanya saat komputer dimatikan, kita menggunakan file untuk penggunaan data masa depan.

Bila kita ingin membaca dari atau menulis ke file kita perlu membukanya terlebih dahulu. Bila sudah selesai, perlu ditutup, agar sumber yang diikat dengan file tersebut dibebaskan.

Oleh karena itu, dengan Python, sebuah operasi file berlangsung dengan urutan sebagai berikut.

1. Buka file
2. Membaca atau menulis (melakukan operasi)
3. Tutup file tersebut

Membuka sebuah file

Python memiliki built-in function `open ()` untuk membuka file. Fungsi ini mengembalikan objek file, juga disebut handle, karena digunakan untuk membaca atau memodifikasi file yang sesuai.

```
>>> f = open("test.txt") # open file in current directory
>>> f = open("C:/Python33/README.txt") # specifying full path
```

Kita bisa menentukan mode saat membuka file. Dalam mode, kami menentukan apakah kita ingin membaca 'r', menulis 'w' atau menambahkan 'a' ke file. Kita juga menentukan apakah kita ingin membuka file dalam mode teks atau mode biner.

Defaultnya adalah membaca dalam mode teks. Dalam mode ini, kita mendapatkan string saat membaca dari file.

Di sisi lain, mode biner mengembalikan byte dan ini adalah mode yang akan digunakan saat berhadapan dengan file non-teks seperti file gambar atau exe.

'r' Buka file untuk dibaca. (default)

'w' Buka file untuk menulis. Membuat file baru jika tidak ada atau memotong file jika file tersebut ada.

'x' Buka file untuk pembuatan eksklusif. Jika file sudah ada, operasi gagal

'a' Buka untuk menambahkan di akhir file tanpa memotongnya. Membuat file baru jika tidak ada.

't' Buka dalam mode teks. (default)

'b' Buka dalam mode biner.

'+' Buka file untuk mengupdate (membaca dan menulis)

```
= open("test.txt") # equivalent to 'r' or 'rt'
= open("test.txt",'w') # write in text mode
= open("img.bmp",'r+b') # read and write in binary mode
```

Tidak seperti bahasa lain, karakter 'a' tidak menyiratkan angka 97 sampai dikodekan menggunakan ASCII (atau pengkodean setara lainnya).

Apalagi, pengkodean default bergantung pada platform. Di jendela, itu adalah 'cp1252' tapi 'utf-8' di Linux.

Jadi, kita juga tidak harus bergantung pada pengkodean default atau kode kita akan berperilaku berbeda di berbagai platform.

Oleh karena itu, ketika bekerja dengan file dalam mode teks, sangat disarankan untuk menentukan jenis pengkodean.

```
= open("test.txt",mode = 'r',encoding = 'utf-8')
```

Menutup sebuah File

Ketika kita selesai dengan operasi ke file, kita perlu menutupinya dengan benar.

Menutup file akan membebaskan sumber daya yang terkait dengan file dan dilakukan dengan menggunakan metode close ().

Python memiliki pengumpul sampah untuk membersihkan benda yang tidak diinterpretasi tapi, kita tidak boleh bergantung padanya untuk menutup file.

```
= open("test.txt",encoding = 'utf-8')
```

```
# perform file operations  
.close()
```

Report generated by [smallseotools.com](https://smallseotools.com)

SmallSeoTools.com