

## TEXT FILE

### 1. Penjelasan pengertian Text file

= Text file adalah jenis file yang digunakan untuk menyimpan data dalam bentuk teks biasa. ini adalah format paling umum digunakan untuk menyimpan dan membaca data dalam berbagai aplikasi. file text dapat berisi karakter, kata, kalimat atau paragraf yang dapat dibaca manusia.

### 2. Penjelasan perbedaan method read() dan open() dan contoh syntaxnya.

= metode read() dan open adalah dua fungsi yang digunakan dalam pemrograman untuk membaca dan membuka file text.

- Open() adalah fungsi yang digunakan untuk membuka file. ~~metode ini~~ mode ini dapat berupa 'r' untuk membaca file, 'w' untuk menulis file, atau 'a' untuk menambahkan konten ke file yang sudah ada. contoh syntaxnya sebagai berikut :

Python

```
File = open('nama_file.txt', 'r')
```

- read() adalah method yang digunakan untuk ~~untuk~~ membaca isi file yang telah dibuka.

Jika argumen tidak diberikan, maka seluruh isi file akan dibaca.



contoh syntaxnya sebagai berikut:

Python

```
content = file.read()
```

3. mode 'w' dan mode 'a' adalah dua mode yang berbeda yang dapat digunakan saat membuka file dengan method 'open()'.

- mode 'w' (write) digunakan untuk menulis konten baru ke dalam file. Jika file sudah ada, maka konten yang ada akan dihapus dengan ~~konten~~ konten baru. Jika file tidak ada maka file baru akan dibuat.
- mode 'a' (append) digunakan untuk menambahkan konten baru ke dalam file yang sudah ada. Jika file tidak ada maka file baru akan dibuat.

## EXCEPTION

1. penjelasan tentang exception.

= Exception adalah kondisi yang terjadi saat program mengalami kesalahan atau situasi yang tidak terduga saat dieksekusi. Ketika exception terjadi, program biasanya akan berhenti dan memberikan pesan error kepada pengguna. Exception memungkinkan program untuk mengatasi kesalahan dan mengambil tindakan yang sesuai.



2. Jenis-jenis untuk menangani exception error dan contoh syntaxnya.

= 'try-except' digunakan untuk menangani exception dengan menjalankan blok kode dalam 'try' dan menangkap exception yang mungkin terjadi dalam blok except. contoh syntaxnya sebagai berikut:

try:

# Blok kode yang mungkin menimbulkan exception

hasil = 10/0

except zerodivisionerror:

# Blok kode yang akan dieksekusi jika zerodivision terjadi

print('Tidak dapat membagi angka dengan nol')

- 'try-except-else' digunakan untuk menangani exception dengan menjalankan blok kode dalam 'try' dan menangkap exception yang mungkin terjadi dalam blok 'except'. contoh syntaxnya sebagai berikut:

try:

# Blok kode yang mungkin menimbulkan exception

hasil = 10/2

Exception zerodivisionerror

# Blok kode yang akan dieksekusi jika tidak ada



zerodivision terjadi

```
print('Tidak dapat membagi angka dengan nol')
```

else

\* Blok kode yg tidak akan dieksekusi jika tidak ada ekseepthn yang terjadi.

```
print('Hasil pembagian adalah:', hasil)
```

- 'try-except-finally' digunakan untuk menangani exception dengan menjalankan blok kode dalam 'try' dan menangkap exception yang mungkin terjadi dalam blok 'except'. Blok kode dalam 'finally' akan selalu dieksekusi, baik ada exception atau tidak. contoh :

try :

\* Blok kode yang mungkin menimbulkan exception  
hasil = 10/0

exception zero division error :

\* Blok kode yang akan dieksekusi jika zerodivision error terjadi.

```
print('Tidak dapat membagi angka dengan nol')
```

finally :

\* Blok kode<sup>yg</sup> akan selalu dieksekusi

```
print('program selesai')
```

- 'raise' digunakan untuk melempar exception secara manual. ini memungkinkan ~~anda~~ anda untuk membuat dan melempar exception sesuai dengan kebutuhan ~~Anda~~ Anda. contoh syntaxnya :

try:

```
umur = int(input('masukkan umur anda: '))
```

```
if umur < 0:
```

```
    raise ValueError('umur tidak boleh negatif.')
```

```
except ValueError as e:
```

```
    print(e)
```

Itulah beberapa jenis penanganan exception yang umum digunakan.