

[MODUL 12]

Penanganan Eksepsi

A. Apa itu Eksepsi?

Secara umum , eksepsi adalah suatu kesalahan yang terjadi pada saat proses eksekusi program, dan akan mengacaukan alur dari perintah-perintah normal yang terdapat di dalam program. Dalam python, kesalahan-kesalahan tersebut diidentifikasi dengan nama-nama tertentu dan direpresentasikan sebagai objek. Sederhananya, eksepsi adalah suatu objek yang merepresentasikan kesalahan. Berikut adalah contohnya:

```
>>> Print 'halo'
File "<stdin>", line 1
Print 'halo'
      ^
SyntaxError: invalid syntax
>>> while True print 'Hello world'
File "<stdin>", line 1
while True print 'Hello world'
      ^
SyntaxError: invalid syntax
```

SyntaxError inilah yang disebut sebagai eksepsi

B. Membangkitkan Eksepsi dengan **raise**

Suatu eksepsi tertentu dapat kita bangkitkan atau panggil secara paksa menggunakan perintah `raise`, meskipun sebenarnya tidak ada kejadian yang menyebabkan jenis kesalahan tersebut. Silakan buka laman berikut: https://www.w3schools.com/python/gloss_python_raise.asp#:~:text=As%20a%20Python%20developer%20you,exception%2C%20use%20the%20raise%20keyword.

Akses tautan tersebut agar lebih mudah memahami maksud dan tujuan penggunaan perintah `raise`.

C. Menangani Eksepsi dengan Blok `try...except`

Bentuk umum penggunaan blok `try...except` dalam python adalah:

```
try:
    #kode
except TipeEksepsi:
    #penanganan kesalahan
```

Sebagai contoh, perhatikan perbedaan kedua gambar berikut!

Gambar 1

```
#The try block will generate an error, because x is not defined:
try:
    print(x)
except:
    print("An exception occurred")
```

An exception occurred

Gambar 2

```
#The try block will generate an error, because x is not defined:
try:
    x = 'teknik'
    print(x)
except:
    print("An exception occurred")
```

teknik

Pada gambar 1, variabel `x` belum didefinisikan sehingga saat perintah `print` dieksekusi tidak ada yang bisa diprint dari variabel `x` dan muncul error yang berbunyi seperti pada gambar. Sedangkan untuk gambar 2 perintah `print` dapat dieksekusi karena variabel `x` sudah diinisiasi sebelumnya. Pada beberapa kasus, digunakan perintah `exit()` untuk keluar/menghentikan program seperti dalam penanganan eksepsi pada blok *looping*.

D. Klausula else dalam Blok try...except

Bentuk umum penggunaan blok try...except dalam python adalah:

```
try:
    #kode
except TipeEksepsi:
    #penanganan kesalahan
else:
    #kode selanjutnya jika tidak terjadi error
```

Contohnya seperti ini:

```
try:
    teks = raw_input('Ketikkan sesuatu: ')
except EOFError:
    print '\nKenapa sudah EOF?'
except KeyboardInterrupt:
    print '\nAnda membatalkan operasi'
else:
    print 'Anda mengetikkan "%s"' % teks
```

E. Menangani Banyak Eksepsi

Bentuk umum penggunaan blok try...except lebih dari satu dalam python adalah:

```
try:
    #kode
except TipeEksepsi1:
    #penanganan kesalahan1
except TipeEksepsi2:
    #penanganan kesalahan2
except TipeEksepsi3:
    #penanganan kesalahan3
else:
    #kode selanjutnya jika tidak terjadi error
```

Contohnya seperti ini:

```
try:
    teks = raw_input('Ketikkan sesuatu: ')
except EOFError:
    print '\nKenapa sudah EOF?'
except KeyboardInterrupt:
    print '\nAnda membatalkan operasi'
else:
    print 'Anda mengetikkan "%s"' % teks
```

F. Blok `try...finally`

Ketika kita membaca file dari program, kita perlu memastikan file akan ditutup baik ada eksepsi maupun tidak. Dalam hal ini kita bisa menggunakan blok `finally` pada blok `try`. Blok ini tidak digunakan untuk menangani atau menangkap eksepsi, melainkan untuk memastikan bahwa kode akan selalu dilaksanakan baik ada kesalahan ataupun tidak.

Bentuk umum penggunaan blok `try...finally` dalam python adalah:

```
try:
    #kode
finally:
    #kode ini akan selalu dieksekusi
```

Contohnya adalah sebagai berikut:

```
import time

try:
    f = open('coba.txt')
    while True:
        baris = f.readline()
        if len(baris) == 0:
            # EOF
            break
        print baris,
        time.sleep(2) # delay 2 detik
except KeyboardInterrupt:
    print '\nAnda membatalkan operasi'
finally:
    f.close()
    print '\nfile ditutup.'
```

G. Daftar Tipe Eksepsi dalam Python

Eksepsi	Penyebab Error
AssertionError	Muncul pada saat pernyataan assert gagal
AttributeError	Muncul pada saat penugasan terhadap attribute atau referensi gagal
EOFError	Muncul saat fungsi <code>input()</code> mendapatkan kondisi akhir file (end-of-file)

FloatingPointError	Muncul saat operasi terhadap bilangan float gagal
GeneratorExit	Muncul saat metode close() generator dipanggil
ImportError	Muncul saat modul yang hendak diimpor tidak ditemukan
IndexError	Muncul saat indeks dari sequence berada di luar range
KeyError	Muncul saat suatu key tidak ditemukan di dalam dictionary
KeyboardInterrupt	Muncul saat user menekan tombol interupsi (Ctrl + C)
MemoryError	Muncul saat operasi kehabisan memori
NameError	Muncul saat variabel tidak ditemukan
NotImplementedError	Muncul oleh metode abstrak
OSError	Muncul saat sistem operasi bersangkutan mengalami error
OverflowError	Muncul saat hasil operasi perhitungan terlalu besar untuk direpresentasikan
ReferenceError	Muncul saat weak reference digunakan untuk mengakses referensi sampah program
RuntimeError	Muncul saat error yang terjadi di luar semua kategori eksepsi lain
StopIteration	Muncul oleh fungsi next() untuk menunjukkan bahwa tidak ada lagi item yang tersisa pada iterator

SyntaxError	Muncul oleh parser saat terjadi kesalahan sintaks
IndentationError	Muncul saat ada indentasi yang salah
TabError	Muncul saat indentasi memiliki jumlah spasi atau tab yang tidak konsisten
SystemError	Muncul saat interpreter mendeteksi kesalahan internal
SystemExit	Muncul oleh fungsi sys.exit()
TypeError	Muncul saat melakukan operasi pada tipe data yang tidak sesuai
UnboundLocalError	Muncul saat referensi dibuat untuk variabel lokal dari fungsi, tapi tidak ada nilainya.
UnicodeError	Muncul saat terjadi kesalahan berkenaan dengan encoding dan decoding unicode
UnicodeEncodeError	Muncul saat terjadi kesalahan pada proses encoding
UnicodeDecodeError	Muncul saat terjadi kesalahan pada proses decoding
UnicodeTranslateError	Muncul saat terjadi kesalahan berkenaan dengan penerjemahan unicode
ValueError	Muncul saat fungsi menerima argumen yang tipe datanya salah
ZeroDivisionError	Muncul saat terjadi operasi pembagian bilangan dengan nol

Tambahan:

Silakan kunjungi laman resmi python untuk belajar lebih dalam tentang penanganan eksepsi pada tautan berikut:

<https://docs.python.org/3/tutorial/errors.html>

Sumber:

- Mudah Belajar Python Edisi Revisi (Budi Raharjo)
- sakti.github.io
- w3schools.com