

Tugas Sesi 13

Mochammad Fauzi Dwi Rusdiansyah
20220040036 - TI22F

1. Kapan pengecualian bawaan berikut dimunculkan? Berikan contoh untuk mendukung jawaban Anda. (score = 70)
 - A. ImportError
 - B. IOError
 - C. NameError
 - D. ZeroDivisionError
2. Jelaskan hal berikut: (score = 30)
 - A. Exception Handling
 - B. Menangkap exception

1. Pengecualiaan Bawaan

ImportError

a. ImportError muncul ketika kita mencoba untuk mengimport modul yang tidak ada atau tidak dapat diakses. Contoh kasus coding: Dalam kasus ini, kita akan mencoba untuk mengimport fungsi 'error' dalam modul 'module' yang didalam nya tidak terdapat fungsi tersebut sehingga menyebabkan error.

In [37]:

```
from module import error
kalimat = 'Mencari Total Huruf Besar (UPPERCASE) Dalam Sebuah Kalimat' #menggunakan coding sesi sebelumnya
print(kalimat)
print(uppercase.countUppercase(kalimat))
print(uppercase.countUppercase2(kalimat))
```

```
-----
ImportError                                Traceback (most recent call last)
Cell In [37], line 1
----> 1 from module import error
      2 kalimat = 'Mencari Total Huruf Besar (UPPERCASE) Dalam Sebuah Kalimat' #menggunakan coding sesi sebelumnya
      3 print(kalimat)

ImportError: cannot import name 'error' from 'module' (c:\Users\Ezi\Tugas-Dasar-Pemrograman\Exception Handling\module.py)
```

Dalam kode tersebut, output merupakan error yang terdapat pada baris 2 yang merupakan ImportError. Hal itu karena tidak dapat mengimport fungsi 'error' pada modul 'module' karena fungsi tersebut memang tidak ada.

IOError

b. IOError muncul ketika terjadi masalah dengan input/output, seperti mencoba membuka file yang tidak ada. Contoh kasus coding: Kita akan mencoba memanggil/membuka file 'IOError.txt' dengan mode 'r' (read).

In [34]:

```
file = open("IOError.txt", "r")
```

```
-----
FileNotFoundError                          Traceback (most recent call last)
Cell In [34], line 1
----> 1 with open("dokumen.txt", "r") as dok:
      2     print(dok.read())

FileNotFoundError: [Errno 2] No such file or directory: 'dokumen.txt'
```

Kode diatas menyebabkan output error yang menandakan bahwa file yang akan kita buka tidak ada.

NameError

c. NameError muncul ketika kita mencoba untuk mengakses variabel atau fungsi yang tidak didefinisikan. Contoh kasus coding: Kita akan memanggil variable 'phi' pada sebuah operasi mencari luas lingkaran.

In [35]:

```
def countRadius(r):
    radius = phi * r * r
    return radius
```

```
countRadius(10)
```

NameError Traceback (most recent call last)

Cell In [35], line 5
2 radius = phi * r * r
3 return radius
----> 5 countRadius(10)

Cell In [35], line 2, in countRadius(r)
1 def countRadius(r):
----> 2 radius = phi * r * r
3 return radius

NameError: name 'phi' is not defined

Pada kasus diatas, terjadi NameError pada variable 'phi' karena variable tersebut tidak ada dalam codingan atau belum didefinisikan.

ZeroDivisionError

d. ZeroDivisionError muncul ketika kita mencoba untuk membagi dengan nol. Contoh kasus coding: Kita akan menentukan 5 buah kue yang akan dibagikan kepada 0 orang.

In [36]:

```
def cakeDiv(pp1):
    cake = 5
    result = cake/pp1
    return result
cakeDiv(0)
```

ZeroDivisionError Traceback (most recent call last)

Cell In [36], line 5
3 result = cake/pp1
4 return result
----> 5 cakeDiv(0)

Cell In [36], line 3, in cakeDiv(pp1)
1 def cakeDiv(pp1):
2 cake = 5
----> 3 result = cake/pp1
4 return result

ZeroDivisionError: division by zero

Kode diatas menunjukan error ketika 5 kue dibagi dengan 0 orang.

2. Penjelasan

Exception Handling

Exception handling adalah mekanisme di dalam bahasa pemrograman yang digunakan untuk menangani kesalahan yang mungkin terjadi saat program dijalankan. Ketika kesalahan terjadi, program akan mengeluarkan sebuah exception, yang kemudian dapat ditangani oleh blok kode yang sesuai.

Contoh kode:

In [38]:

```
try:
    angka = 10
    pembagi = 0

    result = angka/pembagi

    print(result)
except:
    print("Error: Pembagi tidak dapat bernilai 0.")
```

Error: Pembagi tidak dapat bernilai 0.

Dalam contoh, kita mencoba membagi angka dengan 0. Di sini, kode ini menghasilkan pengecualian/Exception.

Untuk menangani pengecualian, kita menempatkan kode, `result = angka/pembagi` di dalam blok `try`. Sekarang ketika pengecualian terjadi, sisa kode di dalam blok `try` akan dilewati. Blok `except` menangkap pengecualian dan pernyataan di dalam blok `except` dieksekusi.

Jika tidak ada pernyataan di blok `try` yang menghasilkan pengecualian, blok `except` yang dilewati.

Menangkap Exception / Catching Exeption

Menangkap exception dilakukan dengan menggunakan pernyataan `try-except`. Pernyataan `try` digunakan untuk mengeksekusi kode yang mungkin menyebabkan exception, sedangkan pernyataan `except` digunakan untuk menangani exception yang muncul.

Contoh kode:

In [40]:

```
try:
    even = [2,4,6,8]
    print(even[5])

except ZeroDivisionError:
    print("Denominator cannot be 0.")
except IndexError:
    print("Index Out of Bound.")
```

Index Out of Bound.

Dalam contoh ini, kami telah membuat daftar bernama `even_numbers`. Karena indeks daftar dimulai dari 0, elemen terakhir daftar berada di indeks 3.

Di sini, kami mencoba mengakses nilai ke indeks 5. Oleh karena itu, pengecualian `IndexError` terjadi. Ketika pengecualian `IndexError` terjadi di blok `try`, Pengecualian `ZeroDivisionError` dilewati dan kode di dalam pengecualian `IndexError` dijalankan.