

a. ImportError

ImportError terjadi ketika sebuah program mencoba untuk mengimport modul atau paket yang tidak ditemukan atau tidak dapat diakses. Hal ini dapat terjadi karena modul atau paket tersebut tidak ada dalam sistem, atau karena tidak ada dalam PYTHONPATH (lokasi dimana Python mencari modul dan paket) atau karena terdapat kesalahan pada nama modul atau paket yang diimport.

```
In [1]: import module
```

```
-----  
ModuleNotFoundError                                Traceback (most recent call last)  
Input In [1], in <cell line: 1>()  
----> 1 import module  
  
ModuleNotFoundError: No module named 'module'
```

b. IOError

IOError terjadi ketika sebuah program mengalami masalah saat melakukan operasi input/output seperti membaca atau menulis file. Hal ini dapat terjadi karena file yang akan dibaca atau ditulis tidak ditemukan, tidak dapat diakses karena hak akses yang tidak cukup, atau karena masalah lainnya seperti masalah jaringan atau masalah perangkat keras.

```
In [2]: data = open('file.txt', 'r')
```

```
-----  
FileNotFoundError                                Traceback (most recent call last)  
Input In [2], in <cell line: 1>()  
----> 1 data = open('file.txt', 'r')  
  
FileNotFoundError: [Errno 2] No such file or directory: 'file.txt'
```

c. NameError

NameError terjadi ketika sebuah variabel atau fungsi yang tidak didefinisikan diakses oleh program. Hal ini dapat terjadi karena kesalahan ketik atau karena sebuah variabel atau fungsi belum didefinisikan sebelum digunakan dalam program.

```
In [3]: data = ("helo python")  
print(dats)
```

```
-----  
NameError                                         Traceback (most recent call last)  
Input In [3], in <cell line: 2>()  
      1 data = ("helo python")  
----> 2 print(dats)  
  
NameError: name 'dats' is not defined
```

d. ZeroDivisionError

ZeroDivisionError terjadi ketika sebuah bilangan dibagi dengan nol. Hal ini tidak dapat dilakukan dalam matematika, dan akan menyebabkan error dalam program.

```
In [4]: a = 5/0
```

```
-----  
ZeroDivisionError                                Traceback (most recent call last)  
Input In [4], in <cell line: 1>()  
----> 1 a = 5/0  
  
ZeroDivisionError: division by zero
```

2

a Exception Handling

Exception Handling adalah suatu proses menangani error yang mungkin terjadi dalam sebuah program. Hal ini dilakukan dengan menggunakan perintah try dan except yang digunakan untuk mengeksekusi kode yang mungkin menyebabkan error dan memberikan aksi yang sesuai jika terjadi error.

```
In [5]: try:
        a = 5/0
    except:
        print("Tidak bisa melakukan pembagian")
```

Tidak bisa melakukan pembagian

b Menangkap Exception

Menangkap exception berarti mencoba mendeteksi error-error tertentu yang mungkin terjadi dalam suatu kode program. Sebagai contoh, kita ingin mendeteksi error apabila kita ingin mengakses suatu karakter pada string dengan index yang tidak ada. Maka kita dapat menambahkan exception IndexError setelah except.

```
In [6]: try:
        a = input("Masukan text: ")
        print(a[100])
    except IndexError:
        print(f"Tidak terdapat karakter dengan index ke 100 '{a}'")
```

Masukan text: data

Tidak terdapat karakter dengan index ke 100 'data'

```
In [ ]:
```