

LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN KE 10

TRY EXCEPT IN PYTHON



DISUSUN OLEH :

Oktario Mufti Yudha

2320506044

JURUSAN TEKNOLOGI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS TIDAR

2023

LAPORAN

ALGORITMA PEMROGRAMAN DAN STRUKTUR DATA



Diisi Mahasiswa Praktikan									
Nama Praktikan	Oktario Mufti Yudha								
NPM	2320506044								
Rombel	4								
Judul Praktikum	Try Except in Python								
Tanggal Praktikum	15 November 2023								
Diisi Asisten Praktikum									
Tanggal Pengumpulan	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>								
Catatan									

PENGESAHAN		NILAI
Diperiksa oleh :	Disahkan oleh :	
Asisten Praktikum	Dosen Pengampu	
(Kurnadi)	(Imam Adi Nata M.kom)	

BAB I

TUJUAN

1. Mahasiswa mampu memahami dan mengetahui apa itu error handling pada python
2. Mahasiswa dapat menerapkan try exception sesuai dengan kebutuhan
3. Mahasiswa dapat menggunakan sumber daya yang telah tersedia untuk memperluas kemampuan Python secara signifikan

BAB II

DASAR TEORI

Try dan except adalah bagian dari mekanisme penanganan kesalahan atau exceptions dalam Python. Tujuan utamanya adalah mengantisipasi dan menangani kesalahan yang mungkin terjadi saat menjalankan program.

1. Blok try

Blok try digunakan untuk melindungi potongan kode yang mungkin menimbulkan kesalahan saat dijalankan. Kode yang ditempatkan di dalam blok try adalah yang ingin dijalankan, dan di mana kita ingin menangani kesalahan.

2. Blok except:

Blok except mengikuti blok try dan menangkap kesalahan yang terjadi di dalamnya. Ketika kesalahan terjadi di dalam blok try, Python mencocokkan kesalahan tersebut dengan blok except yang cocok, dan menjalankan kode di dalam blok except yang bersesuaian.

3. Penanganan Beragam Kesalahan:

Dapat menggunakan beberapa blok except dengan jenis kesalahan yang berbeda untuk menangani situasi kesalahan yang berbeda pula. Contohnya, `ZeroDivisionError` untuk kesalahan pembagian dengan angka nol, dan `ValueError` untuk kesalahan konversi tipe data yang tidak tepat.

4. Strategi Penggunaan:

Idealnya, try dan except digunakan untuk menangani kesalahan yang memang sudah bisa diperkirakan akan terjadi. Terlalu banyak penggunaan try dan except untuk menangkap kesalahan secara umum bisa menyulitkan untuk memahami kesalahan yang sebenarnya terjadi.

BAB III

ALAT DAN BAHAN

1. Laptop
2. Visual Studio Code
3. Extension Python
4. Extension Jupyter

BAB IV

LANGKAH KERJA

1. Siapkan alat dan bahan yang akan digunakan ketika praktikum.
2. Melakukan percobaan mengenai try except seperti berikut:

```
try:  
    print(n)  
except:  
    print("n is not defined")
```

Gambar 4.1 Percobaan

3. Mulai mengerjakan weekly exercise 1 dengan soal sebagai berikut:

Weekly Exercise 1

Pada program perhitungan bangun datar tugas sebelumnya, berikan *Exception Handling* :

- Jika user lupa memasukkan nilai atau kosong maka akan muncul keterangan "tidak boleh kosong"
- Jika karakter yang diinputkan user bukan integer maka akan muncul error "hanya boleh memasukkan angka"

Gambar 4.2 Weekly Exercise 1

4. Membuka visual studio code dan membuat file dengan nama matematika.py sebagai module yang nantinya kita gunakan di main program
5. Mengisi file matematika.py dengan code sebagai berikut :

```

1  phi = 3.14
2
3  def luas_lingkaran(r):
4      return phi * r * r
5      Oktario Mufti Yudha, 7 days ago • praktikum 11
6  def luas_persegi(s):
7      return s * s
8
9  def luas_segitiga(a, t):
10     return a * t / 2

```

Gambar 4.3 File matematika.py

6. Mengerjakan weekly exercise 2 dengan soal sebagai berikut:

Weekly Exercise 2

Pada program yang anda buat saat UTS berikan *Exception Handling* lalu kumpulkan file di elita!

Gambar 4.4 Weekly Exercise 2

BAB V

HASIL DAN ANALISIS

A. Percobaan

```

try:
    print(n)
except:
    print("n is not defined")

```

Gambar 5.1 Code Percobaan

1. Try pada baris pertama bermaksud seperti coba jalankan code yang ada dalam try jika code berhasil di jalanka maka code di dalam block except tidak akan di jalankan.
2. Jika pada block code try terdapat error, maka error tersebut akan di lemparkan atau di tangani oleh block code except.
3. Seperti pada percobaan diatas, pada block try berisi syntax untuk menampilkan variable n, sedangkan variable n belum di devinisikan. Ini akan menghasilkan

error. Karena ada error pada bagian try maka akan masuk ke dalam block except.

B. Weekly Exercise 1

```
# Weekly Exercise 1
import matematika as mtk

print('weekly exercise 1')

selesai = False
while selesai == False:
    pilihan = input('persegi/lingkaran/segitiga ? ')

    if pilihan == 'persegi':
        sisi = input('masukkan nilai: ')
        try:
            sisi = int(sisi)
            print(f' 1. ini luas persegi mu: {mtk.luas_persegi(sisi)}')
        except:
            if sisi != '' and not type(sisi) is int:
                raise ValueError("Nilai harus angka")
            elif sisi == '':
                print("Nilai tidak boleh kosong")

    elif pilihan == 'lingkaran':
        radius = input('masukkan nilai: ')
        try:
            radius = int(radius)
            print(f' 1. ini luas lingkaran mu: {mtk.luas_lingkaran(radius)}')
        except:
            if radius != '' and not type(radius) is int:
                raise ValueError("Nilai harus angka")
            elif radius == '':
                print("Nilai tidak boleh kosong")

    elif pilihan == 'segitiga':
        alas = input('masukkan nilai alas: ')
        tinggi = input('masukkan nilai tinggi: ')
        try:
            alas = int(alas)
            tinggi = int(tinggi)
            print(f' 1. ini luas segitiga mu: {mtk.luas_segitiga(alas, tinggi)}')
        except:
            if alas != '' or tinggi != '' and not type(alas) is int or not type(tinggi) is int:
                raise ValueError("Nilai harus angka")
            elif alas == '' or tinggi == '':
                print("Nilai tidak boleh kosong")

    selesai = True if input('selesai apa belum?(selesai/belum)' ) == 'selesai' else False

✓ 20.1s
weekly exercise 1
1. ini luas persegi mu: 144
```

Gambar 5.2 Weekly Exercise 1

1. Pada baris pertama ini mengimpor modul matematika dan memberikan alias mtk agar mudah dipanggil nantinya. Modul ini berisi fungsi-fungsi matematika seperti luas_persegi, luas_lingkaran, dan luas_segitiga.
2. Membuat variabel selesai yang awalnya diatur sebagai False. Kemudian, melakukan looping dengan while selama variabel selesai bernilai False.
3. Baris pertama dalam while berisi berfungsi untuk menerima input dari pengguna untuk memilih antara 'persegi', 'lingkaran', atau 'segitiga'.
4. Jika pilihan adalah 'persegi', program meminta pengguna untuk memasukkan nilai sisi. Kemudian, mencoba mengubah input ke tipe data integer. Jika berhasil, menghitung luas persegi dengan fungsi mtk.luas_persegi(sisi). Jika gagal (misalnya, input bukan angka), menangani kesalahan dengan memberikan pesan yang sesuai.

5. Jika pilihan adalah 'lingkaran', program meminta pengguna untuk memasukkan nilai radius. Kemudian, mencoba mengubah input ke tipe data integer. Jika berhasil, menghitung luas lingkaran dengan fungsi `mtk.luas_lingkaran(radius)`. Jika gagal (misalnya, input bukan angka), menangani kesalahan dengan memberikan pesan yang sesuai.
6. Jika pilihan adalah 'segitiga', program meminta pengguna untuk memasukkan nilai alas dan tinggi. Kemudian, mencoba mengubah input ke tipe data integer. Jika berhasil, menghitung luas segitiga dengan fungsi `mtk.luas_segitiga(alas, tinggi)`. Jika gagal (misalnya, input bukan angka), menangani kesalahan dengan memberikan pesan yang sesuai.
7. Kemudian pada baris terakhir bertujuan untuk menentukan apakah program sudah selesai atau belum. Jika pengguna memasukkan 'selesai', variabel selesai diatur menjadi True, dan program keluar dari loop while. Jika tidak, program akan tetap meminta input dari pengguna dan tetap berjalan di dalam loop.

C. Weekly Exercise 2

```

pasien = input("Masukkan nama pasien: ")
alamat = input("Masukkan alamat pasien: ")
bpjs = input("Apakah memiliki BPJS? (y/n): ")
lamaInap = int(input("Lama inap: "))
kelasKamar = input("Kelas kamar: ")

# jika punya BPJS, maka dapat diskon 15%
try:
    if bpjs == 'y':
        bpjs = 'memiliki'
        if kelasKamar == '1':
            harga = 800000
        elif kelasKamar == '2':
            harga = 500000
        elif kelasKamar == '3':
            harga = 200000
        else:
            print("Kelas kamar tidak tersedia")
            exit()

        diskon = harga * 15 / 100
        totalHarga = harga - diskon

    # jika tidak punya BPJS, maka tidak dapat diskon
    elif bpjs == 'n':
        bpjs = 'tidak memiliki'
        if kelasKamar == '1':
            harga = 800000
        elif kelasKamar == '2':
            harga = 500000
        elif kelasKamar == '3':
            harga = 200000
        else:
            print("Kelas kamar tidak tersedia")
            exit()

        totalHarga = harga

    # jika input BPJS tidak y/n, maka program akan berhenti
    else:
        print("BPJS hanya bisa diisi dengan y/n")
        exit()

    # total harga dikalikan dengan lama inap
    totalHarga *= lamaInap
    print(f"Total harga yang harus dibayar pasien yang {bpjs} BPJS dengan kamar kelas {kelasKamar} selama {lamaInap} hari adalah: Rp{totalHarga}")
except:
    raise TypeError('hanya bisa memilih y/n')

```

Gambar 5.3 Code Weekly Exercise 2

1. Pada code di atas saya menambahkan error handling dari code yang sudah pernah saya buat sebelumnya.

2. Saya membungkus code yang sudah saya tulis dengan block try. Tujuannya agar Ketika terjadi error terutama Ketika user meng input selain dari 'y' atau 'n' pada bagian bpjs maka tidak akan langsung menampilkan error melainkan akan di tangkap oleh block code except
3. Dimana block code except akan mengembalikan TypeError yang berisi 'hanya bisa memilih 'y' atau 'n' '

BAB VI

KESIMPULAN

Padapraktikum error handling dalam Python, saya mempelajari pentingnya penanganan kesalahan dalam sebuah program. Melalui penggunaan blok try, saya melindungi bagian kode yang rentan terjadi kesalahan saat dijalankan. Dalam blok except, saya menangkap kesalahan yang terdeteksi, memberikan pesan yang jelas kepada user untuk memahami alasan terjadinya error. Selain itu, praktikum ini juga menunjukkan bagaimana penggunaan try dan except membantu menjaga kelancaran aliran program, mencegah program dari berhenti tiba-tiba akibat kesalahan, serta mengendalikan validasi input untuk memastikan nilai yang dimasukkan oleh user sesuai dengan yang diharapkan. Kemampuan untuk menangani kesalahan dengan baik merupakan aspek penting dalam pengembangan perangkat lunak yang dapat meningkatkan kinerja program secara keseluruhan.

BAB VII

DAFTAR PUSTAKA

1. *Alfian Ma'arif(2020).Buku Ajar Pemrograman Lanjut Bahasa Pemrograman Python: Universitas Ahmad Dahlan*
2. *Codepolitan.com, “Mengenal Statement Try Except di Python”, 13 Agustus 2016, <
<https://www.codepolitan.com/blog/mengenal-statement-try-except-di-python/>> [21
November 2023].*