

LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN
KE - 12
PYTHON TRY EXCEPT



DISUSUN OLEH :

Restu Wibisono

2340506061

JURUSAN TEKNOLOGI INORMASI
FAKULTAS TEKNIK
UNIVERSITAS TIDAR

2023

LAPORAN
SESUAI NAMA MATA KULIAH PRAKTIKUM



Diisi Mahasiswa Praktikan										
Nama Praktikan	Restu Wibisono									
NPM	2340506061									
Rombel	03									
Judul Praktikum	Python Try Except									
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><td></td></tr></table>									
Catatan										

PENGESAHAN		NILAI
Diperiksa oleh :	Disahkan oleh :	
Asisten Praktikum	Dosen Pengampu	
Ludfi Arfiani	Imam Adi Nata	

PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS TIDAR
2023

I. Tujuan Praktikum

Adapun tujuan praktikum ini sebagai berikut :

1. Mengetahui konsep penanganan error dalam pemrograman. Praktikum ini akan belajar bagaimana mengidentifikasi dan juga menangani kesalahan yang mungkin terjadi saat menjalankan program.
2. Dengan menggunakan blok try-except, praktikum akan belajar cara bagaimana memperbaiki yang mungkin terjadi didalam program. Hal ini membantu mencegah program berhenti secara tiba-tiba ketika terjadi kesalahan, sehingga memungkinkan program dapat berjalan dengan lebih lancar, dan dapat memberikan pesan kesalahan yang lebih informatif kepada pengguna.
3. Memahami cara menulis kode yang lebih kuat yang menangani kesalahan dengan baik. Dengan menggunakan blok try-except, dapat mengantisipasi dan bereaksi terhadap kesalahan yang akan terjadi, sehingga meningkatkan keandalan program yang dibuat.

II. Dasar Teori

Dalam latihan ini, akan menggunakan blok Try untuk menempatkan kode yang dapat menyebabkan pengecualian. Jika pengecualian terjadi di blok Coba, kontrol segera diteruskan ke blok Kecuali yang sesuai. Blok kecuali digunakan untuk menangkap pengecualian yang terjadi dan menentukan tindakan yang akan diambil. Jika pengecualian terjadi di blok percobaan, kontrol segera diteruskan ke blok pengecualian yang sesuai. Blok kecuali digunakan untuk menangkap pengecualian yang terjadi dan menentukan tindakan yang akan diambil.

Dalam blok except, dapat menentukan jenis pengecualian yang ingin ditangkap, atau menggunakan klausa except tanpa argumen untuk menangkap semua jenis pengecualian. Misalnya, jika memiliki kode yang mencoba untuk membuka dan membaca file, dapat menempatkan kode ini di dalam blok try. Jika file tidak ditemukan atau terjadi kesalahan lain saat membaca file, pengecualian akan terjadi.

Dalam blok except, dapat menangkap pengecualian yang terjadi, mencetak pesan kesalahan yang sesuai, dan melakukan tindakan yang diperlukan, seperti menutup file atau memberikan pesan. Selain itu, juga dapat menggunakan klausa else setelah blok except. Klausa else digunakan

untuk mengeksekusi kode tambahan setelah blok try tanpa pengecualian. Artinya, jika tidak ada pengecualian yang terjadi di blok try, kode dalam klausa else akan dieksekusi. Klausa else sering digunakan untuk mengeksekusi kode yang bergantung pada keberhasilan blok try.

Tipe pengecualian yang umum adalah ValueError. ValueError terjadi ketika tipe data yang ditentukan tidak cocok dengan tipe data yang diharapkan. Misalnya, jika mencoba mengonversi string menjadi bilangan bulat, tetapi string tidak dapat dikonversi menjadi angka, ValueError akan terjadi.

Selain menangani pengecualian, praktikum ini juga dapat melibatkan validasi input. Validasi input adalah proses memeriksa apakah input yang diberikan memenuhi persyaratan yang ditetapkan. Misalnya, jika untuk memasukkan sebuah nilai, dapat menggunakan try-except untuk menangani pengecualian jika input yang diberikan tidak sesuai.

Sebagai contoh, jika ingin memastikan bahwa input tidak boleh kosong, dapat menggunakan try-except untuk menangani pengecualian yang terjadi jika input kosong. Dalam blok try, dapat untuk memasukkan input, dan jika input kosong, dapat memunculkan pengecualian khusus, seperti ValueError, dan menampilkan pesan kesalahan yang sesuai.

III. Metode Praktikum

A. Alat dan bahan

1. Komputer: Pastikan memiliki komputer yang dapat menjalankan Python. Komputer harus memiliki sistem operasi yang kompatibel dengan Python dan spesifikasi yang memadai untuk menjalankan aplikasi Python dengan lancar.
2. Python: Pastikan Python terinstal di komputer. Bisa didapatkan dengan mengunduh Python versi terbaru dari situs resmi Python dan mengikuti petunjuk instalasi terlampir.
3. Editor teks atau IDE: Pilih editor teks atau IDE (lingkungan pengembangan terintegrasi) yang sesuai dengan preferensi. Beberapa opsi populer termasuk PyCharm, Visual Studio Code, Atom, Sublime Text, atau IDLE (bawaan Python). Pastikan memiliki editor teks atau IDE yang mendukung Python dan diinstal di komputer.
4. Modul Latihan: Jika ada modul yang berkaitan dengan praktik pemrograman Python, baca dan pahami materi tersebut sebelum memulai praktik.
5. Koneksi internet (opsional): Jika ingin mencari lebih banyak materi referensi, mengunduh lebih banyak paket atau modul Python, atau mengakses dokumentasi online, pastikan memiliki koneksi Internet yang stabil.

B. Langkah kerja

1. Persiapan:
 - Pastikan telah menginstal Jupyter Notebook di komputer. Notebook Jupyter dapat diinstal menggunakan Anaconda atau pip.
 - Pastikan juga Python diinstal di komputer.
 - Buka Jupyter Notebook dengan menjalankan "jupyter notebook" di terminal atau baris perintah.

2. Buat buku catatan baru:
 - Saat buku catatan Jupyter terbuka di browser, klik tombol "Baru" untuk membuat buku catatan baru.
 - Pilih kernel yang sesuai yaitu. Python 2 atau Python 3.
3. Tulis kodenya:
 - Di buku catatan baru, akan melihat sel kosong yang bisa diisi dengan kode Python.
 - Tulis kodenya sesuai petunjuk pada modul latihan.
4. Jalankan kode:
 - Untuk menjalankan kode di dalam sel, klik tombol "Run" atau tekan "Shift+Enter" pada keyboard.
 - Hasil kode ditampilkan di bawah sel.
5. Periksa hasilnya dan perbaiki:
 - Perhatikan hasil yang ditampilkan setelah menjalankan kode.
 - Jika ada error atau kekeliruan, perbaiki kode yang salah tersebut.
6. Penyusunan Laporan:
 - Di akhir latihan, buat laporan yang memuat tujuan latihan, langkah-langkah yang dilakukan, hasil observasi, analisis dan kesimpulan yang diambil

IV. Hasil dan Analisis

A. Weekly Exercise 1

```
# Write your code below and press Shift+Enter to execute
import matematika as mtk

selesai = False
while not selesai:
    try:
        pilih = input("Bangun datar apa yang ingin dihitung (lingkaran, persegi, segitiga): ")
        if pilih == 'lingkaran':
            r = input("Masukkan jari-jari: ")
            if r == "":
                raise ValueError("Tidak boleh kosong")
            if not r.isdigit():
                raise ValueError("Hanya boleh masukkan angka")
            r = int(r)
            lingkaran = mtk.lingkaran(r)
            print("Luas lingkaran dengan jari-jari", r, "adalah", lingkaran)
        elif pilih == 'persegi':
            s = input("Masukkan panjang sisi: ")
            if s == "":
                raise ValueError("Tidak boleh kosong")
            if not s.isdigit():
                raise ValueError("Hanya boleh masukkan angka")
            s = int(s)
            persegi = mtk.persegi(s)
            print("Luas persegi dengan panjang sisi", s, "adalah", persegi)
        elif pilih == 'segitiga':
            a = input("Masukkan lebar alas: ")
            t = input("Masukkan tinggi: ")
            if a == "" or t == "":
                raise ValueError("Tidak boleh kosong")
            if not a.isdigit() or not t.isdigit():
                raise ValueError("Hanya boleh masukkan angka")
            a = int(a)
            t = int(t)
            segitiga = mtk.segitiga(a, t)
            print("Luas segitiga dengan alas", a, "dan tinggi", t, "adalah", segitiga)
        else:
            print("Pilihlah lingkaran, persegi, segitiga")
        selesai = input("selesai") == 'selesai'
    except ValueError as e:
        print(e)
    except:
        print("Terjadi kesalahan")
```

(Gambar 1.1)

1. Program akan mengimpor modul matematika dengan alias mtk. Modul ini harus berisi fungsi untuk menghitung luas lingkaran, persegi, dan juga segitiga.
2. Variabel yang selesai diinisialisasi sebagai False. Ini akan digunakan untuk mengontrol perulangan (while) sehingga program ini akan terus berjalan selama selesai bernilai False.
3. Perulangan ini akan terus berjalan selama selesai adalah False.
4. Blok try dimulai, yang berarti program akan mencoba menjalankan kode ada di dalamnya. Namun jika terjadi kesalahan (error), program akan melompat ke blok except yang berguna untuk menangani kesalahan tersebut.

5. Program meminta pengguna memasukkan jenis bangun datar yang ingin dihitung (lingkaran, persegi, atau segitiga) dan akan menyimpannya dalam variabel pilih.
6. Jika pilihan pengguna adalah lingkaran, program akan meminta memasukkan jari-jari, memvalidasi masukkan, menghitung luas, dan mencetak hasilnya. Langkah-langkah serupa diikuti untuk persegi dan segitiga.
7. Program meminta pengguna untuk memasukkan 'selesai'. Jika pengguna memasukkan 'selesai', variabel selesai akan menjadi True, dan perulangan akan berakhir. Jika tidak, perulangan akan terus berlanjut.
8. Jika terjadi ValueError (misalnya, karena input tidak valid), program akan mencetak pesan kesalahan yang sesuai. Blok except terakhir menangani kesalahan umum dan mencetak pesan kesalahan generik.

B. Weekly Exercise 2

```
# Write your code below and press Shift+Enter to execute
def biaya(lama, harga, member):
    if member == "y":
        potongan = 20/100
    else:
        potongan = 0
    total = harga * lama * (1 - potongan)
    return total

try:
    nama = input("Masukkan nama:")
    if nama == "":
        raise ValueError("Harus memasukkan data diri")
    alamat = input("Masukkan alamat:")
    if alamat == "":
        raise ValueError("Harus memasukkan data diri")
    print("Apakah kamu termasuk member? (y/n):")
    member = input("Apakah kamu member (y/n)").lower()
    while member not in ["y", "n"]:
        print("Ulangi. Harap masukkan 'y' atau 'n'.")
        member = input().lower()

    print("Pilihan jenis kamar:")
    print("1. Suite (Rp 1.000.000/hari)")
    print("2. Superior (Rp 600.000/hari)")
    print("3. Deluxe (Rp 400.000/hari)")

    try:
        jenis = int(input("Masukkan jenis kamar yang akan dipilih (1 / 2 / 3):"))
    except ValueError:
        raise ValueError("Hanya bisa memasukkan angka")

    while jenis not in [1, 2, 3]:
        print("Pilihan tidak valid. Harap masukkan 1, 2, atau 3.")
        try:
            jenis = int(input())
        except ValueError:
            raise ValueError("Hanya bisa memasukkan angka atau Harap pilih kamar terlebih dahulu.")

    if jenis == 1:
        harga = 1000000
    elif jenis == 2:
        harga = 600000
    else:
        harga = 400000

    try:
        lama = int(input("Lama tamu menginap (dalam hari):"))
    except ValueError:
        raise ValueError("Hanya bisa memasukkan angka atau Harap memasukkan lama hari menginap.")

    total = biaya(lama, harga, member)

    print("Info tamu:")
    print("Nama:", nama)
    print("Alamat:", alamat)

    if member == "y":
        print("Member Hotel: Member")
    else:
        print("Member Hotel: Bukan Member")

    print("Tamu", nama, "beralamat", alamat, "memilih kamar tipe nomor", jenis, ", menginap", lama,
          "hari dengan total biaya sebesar Rp", total)

except ValueError as e:
    print(e)
```

(Gambar 2.1)

1. Program ini akan menghitung total biaya yang dibutuhkan berdasarkan lama menginap, harga kamar, dan status keanggotaan member.
2. Jika pengunjung adalah member ("y"), maka ada potongan harga sebesar 20%.
3. Total biaya dihitung adalah dengan mengalikan harga kamar, lama menginap, dan potongan.
4. Program akan meminta pengguna untuk memasukkan nama dan alamat. Jika salah satu atau keduanya kosong, maka program akan menyebabkan ValueError dengan pesan yang sesuai.
5. Program akan meminta pengguna untuk memasukkan status keanggotaan member (y/n) dan memvalidasi input. Pengguna akan dimintai untuk memasukkan ulang jika input tidak valid.
6. Program menampilkan pilihan berupa jenis kamar dan meminta pengguna untuk memasukkan pilihan. Program mencoba mengonversi input menjadi integer, dan jika gagal, maka akan menyebabkan ValueError.
7. Program akan melakukan validasi bahwa pilihan jenis kamar harus berada di antara 1, 2, atau 3. Pengguna akan diminta untuk memasukkan ulang jika input tidak valid.
8. Program akan menentukan harga kamar berdasarkan pilihan jenis kamar yang sesuai dimasukkan pengguna.
9. Program akan meminta pengguna untuk memasukkan hari lama menginap dan mencoba mengonversi input menjadi integer. Jika gagal, program akan menyebabkan ValueError.
10. Program memanggil fungsi biaya dengan parameter lama, harga kamar, dan status keanggotaan member.
11. Program nantinya menampilkan informasi tamu, termasuk nama, alamat, status keanggotaan member, pilihan jenis kamar, lama menginap, dan total biaya.
12. Blok ini menangani kesalahan ValueError yang akan mungkin terjadi selama eksekusi program dan mencetak akan pesan kesalahan

yang sesuai. Misalnya, jika pengguna tidak memasukkan angka saat diminta untuk lama menginap, program akan menangkap kesalahan tersebut dan mencetak pesan kesalahan.

V. Kesimpulan

Dalam latihan ini, akan mempelajari cara menggunakan mekanisme penanganan kesalahan Python, blok try-excel. Blok try digunakan untuk menandai bagian kode yang mungkin menyebabkan kesalahan, dan blok Kecuali digunakan untuk menangkap dan menangani kesalahan yang terjadi. yang akan memahami sintaks blok try-excel, termasuk cara mengenali dan menangani berbagai jenis kesalahan yang mungkin terjadi. Kesalahan pembagian dengan nol (ZeroDivisionError), kesalahan format (ValueError), atau kesalahan indeks tidak valid (IndexError).

Selain itu, lab ini juga mendemonstrasikan penggunaan blok terakhir, yang dapat digunakan untuk mengeksekusi kode yang perlu dieksekusi, terlepas dari apakah terjadi kesalahan atau tidak.

Dalam ini kan memahami cara menggunakan blok "akhirnya" untuk membersihkan sumber daya atau melakukan tindakan akhir sebelum program keluar. Memahami konsep dan penggunaan blok try-excel dapat membantu menulis kode yang lebih kuat dan menangani kesalahan dengan lebih baik. Ini membantu meningkatkan keandalan dan stabilitas program yang dibuat. Ini memberikan kemampuan untuk mendeteksi dan memperbaiki kesalahan yang terjadi dalam suatu program, sehingga meningkatkan kualitas dan kinerja program secara keseluruhan.

VI. Referensi

Cahyanto, T. A. 2017. Praktikum Mata Kuliah Keamanan Komputer.

Herho, S. H. S. 2018. Tutorial Pemrograman Python 2 Untuk Pemula.

Huda, A., Kom, S., Kom, M., & Ardi, N. 2020. Dasar-Dasar Pemrograman Berbasis Python. UNP PRESS.

ST Wadi, H. 2018. Pemrograman Python: Untuk Pelajar & Mahasiswa (Vol. 1). TR Publisher.

Suryawan, F.*et-al*, 2020. Modul Praktikum Algoritma & Struktur Data Versi 4.3. Muhammadiyah University Press.