Laporan Praktikum Algoritma & Pemrograman

Semester Genap 2025/2026

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

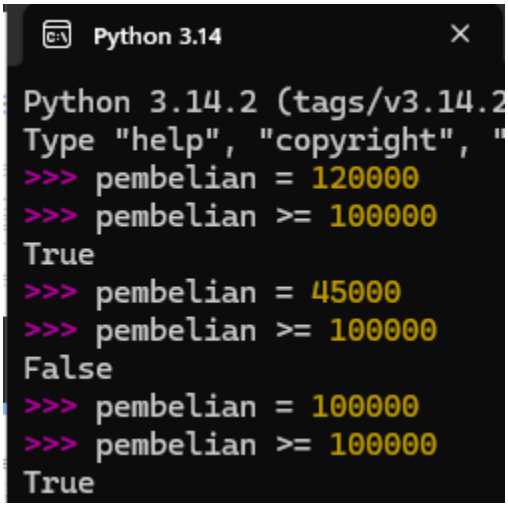
NIM	**71251182**
Nama Lengkap	**Michael Dylan**
Minggu ke / Materi	**04 / Struktur Kontrol Percabangan**

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2026

BAGIAN 1: MATERI MINGGU 1 (40%)

4.3.1 Boolean Expression dan Logical Operator

Ada voucher diskon 30% tapi cuma bisa dipakai kalau belanja minimal Rp100.000. Jadi kalau mau dapet diskon, total belanja harus ≥ 100000. Di Python bisa ditulis: pembelian >= 100000 Artinya, kalau nilai pembelian lebih dari atau sama dengan 100000, baru diskonnya bisa dipakai. Bentuk tersebut dinamakan boolean expression, karena hasilnya hanya tedapat dua kemungkinan, yaitu True atau False, tergantung variabel pembelian. Jika memasukkan perintah tersebut ke dalam Python Interactive hasilnya bisa dilihat pada Gambar 4.1.



Gambar 4.1: Hasil tergantung dari variable pembelian. Sumber Python Interactive

Boolean expression disusun menggunakan operato-operator perbandingan seperti Tabel 4.1.

OPERATOR	KETARANGAN
x == y	Apakah x sama dengan y?
x != y	Apakah x tidak sama dengan y?
x > y	Apakah x lebih besar dari y?
x >= y	Apakah x lebih besar atau sama dengan y?
x < y	Apakah x lebih kecil dari y?
x <= y	Apakah x lebih kecil atau sama dengan y?
x is y	Apakah x sama dengan y?

x is not y	Apakah x tidak sama dengan y?

Tabel 4.1: Operator-operator perbandingan (comparison).

Kita harus bisa menyusun bentuk boolean expression dan memilih operator yang tepat sesuai permasalahan yang dihadapi. Beberapa hal perlu diperhatikan saat menyusun bentuk boolean expression:

• Bentuk boolean expression pasti hasilnya hanya ada dua kemungkinan, yaitu True atau False.

• Perhatikan kata-kata khusus seperti minimum, maksimum, tidak lebih dari, tidak kurang dari, tidak sama, tidak berbeda.

• Perhatikan dengan seksama dan tentukan variabel yang perlu dibandingkan dengan benar sesuai dengan permasalahan.

Beberapa contoh permasalahan dan bentuk boolean expression-nya bisa dilihat di Tabel 4.2.

CONTOH MASALAH	**BOOLEAN EXPRESSION**
Untuk lulus dibutuhkan IPK minimum 2.25	ipk >= 2.25
Golden Button hanya diberikan untuk Youtuber dengan subscriber lebih dari 1 juta	subscriber > 1000000
Pengendara dengan kecepatan lebih dari 90 km/jam akan mendapatkan tilang	kecepatan > 90
Wahana Rollercoaster hanya bisa dinaiki oleh mereka yang tinggi badannya lebih dari 110 cm	tinggi > 110
Nilai ujian Hanna adalah 75 sedangkan Robby mendapat-kan nilai 75. Apakah nilai keduanya sama?	hanna is robby
Junaedi memiliki 10 sepatu, Ricky punya 15 sepatu dan Ar-nold punya 20 sepatu. Apakah gabungan sepatu Junaedi dan Ricky lebih banyak dari sepatu milik Arnold?	junaedi + ricky > arnold

Tabel 4.2: Operator-operator perbandingan (comparison).

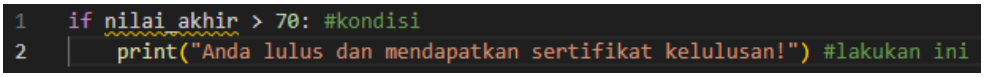
Beberapa boolean expression bisa digabungkan dengan menggunakan logical operator. Logical operator pada Python adalah and, or dan not. Sebagai contoh, misalnya wahana Rollercoaster. hanya dapat dinaiki oleh penumpang dengan usia minimal 10 tahun dan tinggi badan minimal 110 cm. Kedua persyaratan tersebut dapat dinyatakan dalam bentuk sebagai berikut: **usia >= 10 and tinggi >= 110**

Diskon diberikan kepada member atau jumlah pembelian lebih dari Rp. 500.000:

member == true or pembelian > 500000

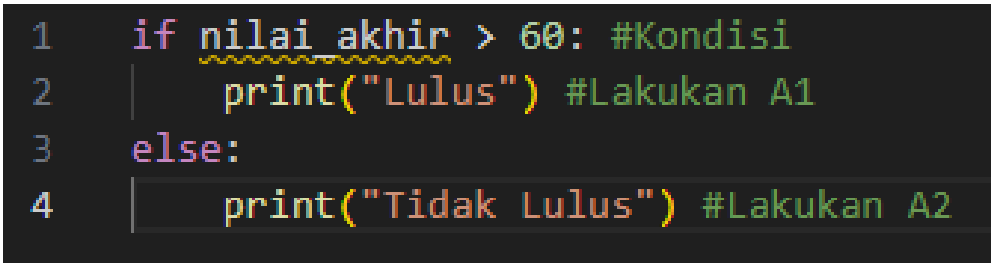
4.3.2 Bentuk-bentuk Percabangan

Percabangan Python secara umum ada tiga bentuk, yaitu: conditional, alternative dan chained conditional. Bentuk conditional secara umum dinyatakan dalam kode program. Sebagai contoh, jika nilai akhir > 70 maka akan mendapatkan sertifikat kelulusan seperti di Gambar 4.2.



Gambar 4.2: Bentuk conditional dalam kode program. Sumber Python Vscode

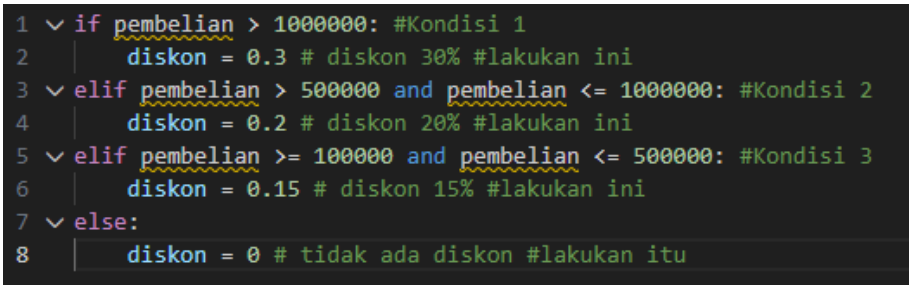
Bentuk alternative conditional adalah bentuk percabangan yang memiliki dua alternatif langkah yang dijalankan berdasarkan kondisi tertentu. Sebagai contoh, jika nilai akhir > 60, tampilkan tulisan Lulus. Jika tidak, tampilkan tulisan Tidak Lulus. Pada contoh ini, ada dua kemungkinan tulisan yang muncul, yaitu Lulus atau Tidak Lulus seperti Gambar 4.3.



Gambar 4.3: Bentuk percabangan berdasarkan kondisi tertentu. Sumber Python Vscode

Chained conditional dipakai kalau kemungkinan langkahnya lebih dari dua. Contohnya di toko pakaian, besar diskon tergantung total belanja. Kalau belanja di atas Rp1.000.000 dapat diskon 30%. Kalau lebih dari Rp500.000 sampai Rp1.000.000 dapat 20%. Kalau Rp100.000 sampai

Rp500.000 dapat 15%. Dan kalau di bawah Rp100.000 tidak dapat diskon. Jadi ada 4 kemungkinan diskon: 30%, 20%, 15%, atau 0% seperti Gambar 4.4.



Gambar 4.4: Bentuk percabangan lebih dari dua. Sumber Python Vscode

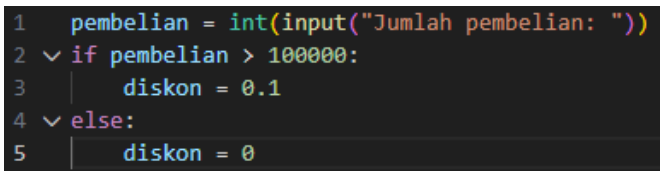
Pernyataan "Pembelian lebih dari Rp. 500.000 sampai Rp. 1.000.000 mendapatkan diskon 20%" diimplementasika bentuk gabungan dari dua boolean expression, yaitu:

```
pembelian > 500000 and pembelian <= 1000000
```

Untuk menyatakan rentang nilai biasanya pakai operator logika and karena dua kondisi harus terpenuhi supaya bernilai True. Selain itu, Python juga punya cara percabangan yang lebih sing-kat, yaitu ternary operator seperti di bawah ini:

```
pembelian = int(input("Jumlah pembelian: "))
diskon = 0.1 if pembelian > 100000 else 0
```

Bentuk ternary tersebut merupakan bentuk lain dari if-else berikut ini:



Gambar 4.5: Bentuk tenary. Sumber Python Vscode

4.3.3 Penanganan Kesalahan Input Menggunakan Exception Handling

Saat menerima input dari pengguna, kita harus siap denga kesalahan supaya program tetap berjalan dengan benar. Contohnya program meminta usia seperti di Gambar 4.6. Kalau program dijalankan dengan berbagai input, hasilnya sudah sesuai harapan seperti di Gambar 4.7. Lalu menentukan kategori umur:

- Balita (0–5 tahun)

- Kanak-kanak (6–11 tahun)

- Remaja (12–25 tahun)

- Dewasa (26–45 tahun)

- Lansia (di atas 45 tahun).

```python
usia = int(input("Masukkan usia anda: ")) #bagian program meminta pengguna memasukkan usianya
if usia <= 5:
    print("Balita")
elif usia >= 6 and usia <= 11:
    print("Kanak-kanak")
elif usia >=12 and usia <= 25:
    print("Remaja")
elif usia >= 26 and usia <= 45:
    print("Dewasa")
elif usia > 45:
    print("Lansia")
```

Gambar 4.6: Percabangan program meminta usia. Sumber Python Vscode

```
PS D:\GitHub\Training Dylan> & C:/Users/merro/AppData/Local/Python/pythoncore-3.14-64/python.exe "d:/GitHub/Training Dylan/gagagugu.py"
Masukkan usia anda: 3
Balita
PS D:\GitHub\Training Dylan> & C:/Users/merro/AppData/Local/Python/pythoncore-3.14-64/python.exe "d:/GitHub/Training Dylan/gagagugu.py"
Masukkan usia anda: 7
Kanak-kanak
PS D:\GitHub\Training Dylan> & C:/Users/merro/AppData/Local/Python/pythoncore-3.14-64/python.exe "d:/GitHub/Training Dylan/gagagugu.py"
Masukkan usia anda: 14
Remaja
PS D:\GitHub\Training Dylan> & C:/Users/merro/AppData/Local/Python/pythoncore-3.14-64/python.exe "d:/GitHub/Training Dylan/gagagugu.py"
Masukkan usia anda: 25
Remaja
PS D:\GitHub\Training Dylan> & C:/Users/merro/AppData/Local/Python/pythoncore-3.14-64/python.exe "d:/GitHub/Training Dylan/gagagugu.py"
Masukkan usia anda: 50
Lansia
```

Gambar 4.7: Hasil dari program meminta usia. Sumber Python Vscode

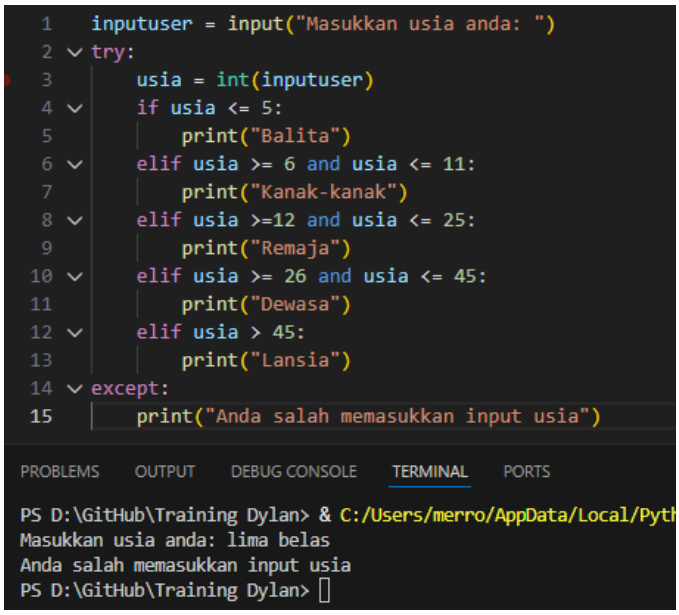
Fungsi input() dipakai untuk membaca masukan dari pengguna, tapi hasilnya berupa string. Karena usia harus berupa angka, maka perlu diubah dulu jadi bilangan bulat dengan int(). Program akan berjalan normal kalau inputnya sesuai, tapi kalau pengguna memasukkan input yang salah atau tidak sesuai, maka bisa terjadi error seperti pada Gambar 4.8.

```
PS D:\GitHub\Training Dylan> & C:/Users/merro/AppData/Local/Python/pythoncore-3.14-64/python.exe "
Masukkan usia anda: lima belas
Traceback (most recent call last):
  File "d:\GitHub\Training Dylan\gagagugu.py", line 1, in <module>
    usia = int(input("Masukkan usia anda: ")) #bagian program meminta pengguna memasukkan usianya
ValueError: invalid literal for int() with base 10: 'lima belas'
```

Gambar 4.8: Kesalahan input. Sumber Python Vscode

Untuk menangani input yang tidak sesuai, bisa pakai try dan except. Jadi kalau pengguna memasukkan data yang salah, program tidak langsung error, tapi bisa menangani kesalahan

tersebut. Pada kasus kategori usia, dengan try dan except, program tetap berjalan normal meskipun inputnya tidak sesuai, seperti yang terlihat pada Gambar 4.9.



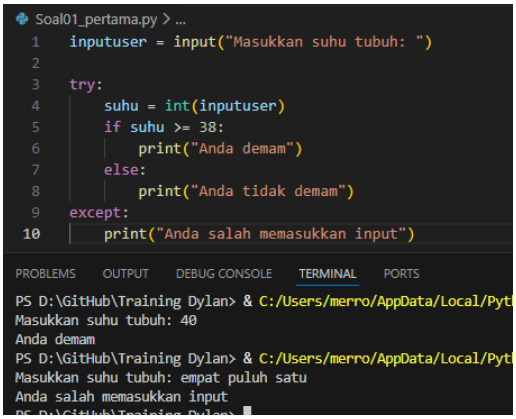
Gambar 4.9: Hasil program dengan menggunakan try dan except. Sumber Python Vscode

BAGIAN 2: LATIHAN MANDIRI (60%)

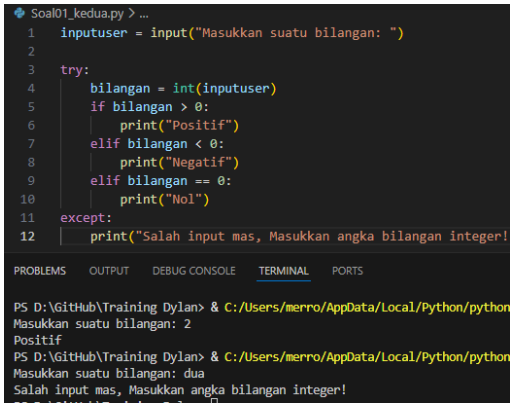
Link Github = https://github.com/MichaelDylan-ti/71251182_michaeldylan/tree/main

LATIHAN 4.1

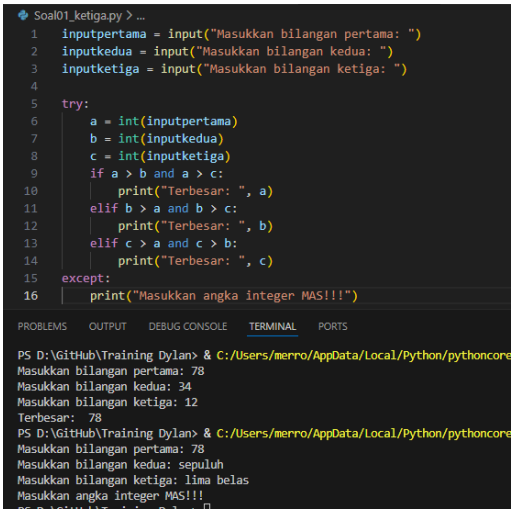
Implementasikan penanganan kesalahan input pengguna dari program-program pada Contoh 3.1, 3.2 dan 3.3.



Gambar 4.10: Program pertama menggunakan try dan except. Sumber Python Vscode



Gambar 4.11: Program kedua menggunakan try dan except. Sumber Python Vscode

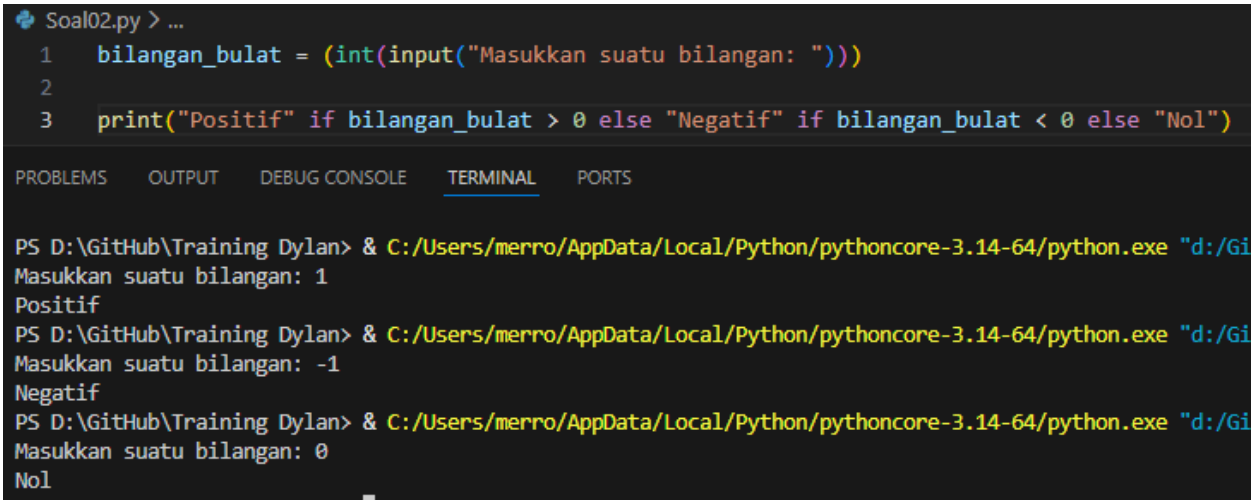


Gambar 4.12: Program ketiga menggunakan try dan except. Sumber Python Vscode

Dari Gambar 4.10, Gambar 4.11, Gambar 4.12 merupakan program yang menggunakan try dan except supaya saat user salah menginput akan diberitahu akan kesalahannya sehingga user bisa menginput ulang dan menghasilkan yang diinginkan dari program percabangan.

LATIHAN 4.2

Implementasikan percabangan pada Contoh 3.2 (Positif-Negatif) menggunakan ternary operator.



Gambar 4.13: Program percabangan menggunakan tenary operator. Sumber Python Vscode

Pada Gambar 4.13 merupakan program percabangan yang menggunakan tenary operator yang dimana terdapat variabel bilangan_bulat yang menyimpan input dengan integar dengan kalimat “Masukkan suatu bilangan: “ lalu di bawahnya ada program percebangan menggunakan tenary operator dengan mengeprint Positif jika bilangan_bulat lebih dari 0 jika tidak print “Negatif” lalu jika bilangan_bulat di bawah maka print “Nol”

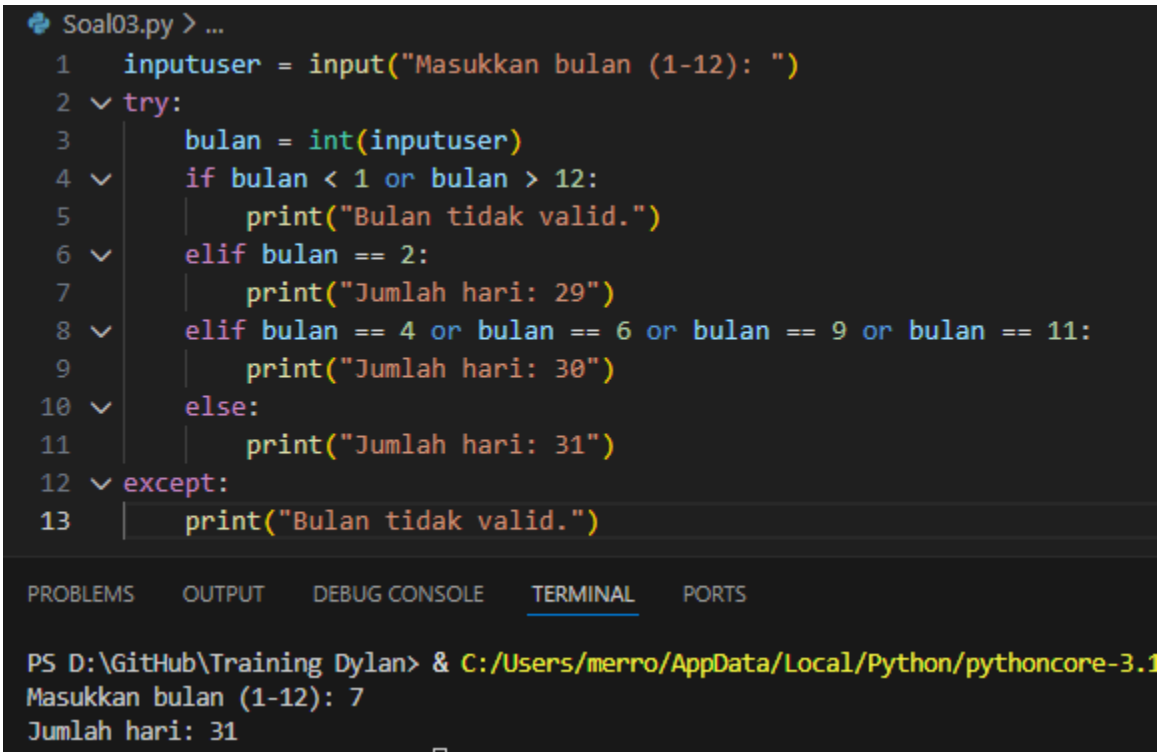
LATIHAN 4.3

Buatlah sebuah program yang dapat menampilkan jumlah hari dalam suatu bulan di tahun 2020. Program meminta pengguna memasukkan nomor bulan (1-12), kemudian program akan menampilkan jumlah hari pada bulan tersebut. Sebagai contoh, perhatikan input dan output berikut ini:

Masukkan bulan (1-12): 7

Jumlah hari: 31

Lengkapi program tersebut dengan penanganan kesalahan jika pengguna memasukkan bulan yang salah. Penanganan kesalahan dalam bentuk memunculkan pesan bahwa bulan yang diinput-kan oleh pengguna tersebut tidak valid.



Gambar 4.14: Program percabangan menggunakan try dan except. Sumber Python Vscode

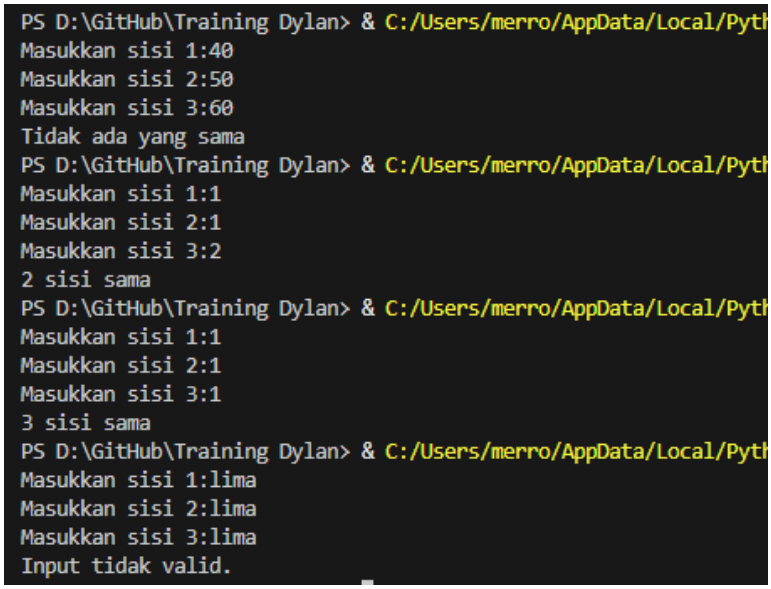
Program ini meminta nomor bulan lalu mengubah input jadi angka. Dipakai try-except supaya kalau input bukan angka, program tidak error dan langsung menampilkan “Bulan tidak valid.” Kalau angka di luar 1–12 juga dianggap tidak valid. Karena tahun 2020 itu tahun kabisat, Februari punya 29 hari, beberapa bulan punya 30 hari, dan sisanya 31 hari.

LATIHAN 4.4

Sebuah program meminta pengguna memasukkan ketiga panjang sisi suatu segitiga (berarti pengguna memasukkan tiga bilangan). Jika ketiga sisi segitiga tersebut semuanya sama, tampil-kan pesan: "3 sisi sama". Jika hanya ada dua sisi yang sama panjang, tampilkan pesan "2 sisi sama". Jika tidak ada yang sama maka tampilkan pesan: "Tidak ada yang sama". Sebagai contoh,

perhatikan input dan output berikut ini: Masukkan sisi 1: 14 Masukkan sisi 2: 18 Masukkan sisi 3: 11 Tidak ada yang sama Masukkan sisi 1: 22 Masukkan sisi 2: 22 Masukkan sisi 3: 22 3 sisi sama Masukkan sisi 1: 8 Masukkan sisi 2: 9 Masukkan sisi 3: 8 2 sisi sama Lengkapi program tersebut dengan penanganan kesalahan jika pengguna memasukkan input yang tidak valid.

Gambar 4.15: Input dari program percabangan try dan except. Sumber Python Vscode



Gambar 4.16: Ouput dari program percabangan try dan except. Sumber Python Vscode

Berdasarkan Gambar 4.15 dan Gambar 4.16, Program ini minta tiga panjang sisi lalu diubah jadi angka. Dipakai try-except supaya kalau input bukan angka, program tidak error dan langsung menampilkan “Input tidak valid.” Kalau ada sisi yang kurang dari atau sama dengan nol juga

dianggap tidak valid. Setelah itu dicek, kalau tiga-tiganya sama tampil “3 sisi sama”, kalau cuma dua yang sama tampil “2 sisi sama”, dan kalau semuanya beda tampil “Tidak ada yang sama”.