



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230977
Nama Lengkap	MICHAEL HOSEA
Minggu ke / Materi	03 / Struktur Kontrol Percabangan

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.

PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA  
YOGYAKARTA  
2024

## BAGIAN 1: MATERI MINGGU INI (40%)

### MATERI 1 : Boolean and Logic

*Boolean* adalah sebuah tipe data yang memang hanya memiliki dua nilai, yaitu benar atau salah (*true or false*). Dalam bahasa pemrograman nilai *true* atau benar dapat digantikan atau dialihkan menjadi 1, sedangkan nilai *false* atau salah dialihkan menjadi 0. Boolean sering digunakan

untuk masuk ke percabangan yang ada di dalam kode dengan IF ELSE, WHILE, dan FOR. Beberapa contoh Boolean expression adalah:

```
print(x == y)
print(x != y)
print(x > y)
print(x < y)
print(x >= y)
print(x <= y)
print(x is y)
print(x is not y)
```

Dalam kode tersebut, terdapat pernyataan cetak (print statement). Dengan asumsi  $x = 1$  dan  $y = 0$ , dan karena yang dicetak adalah ekspresi boolean, hasil cetaknya juga pasti berupa ekspresi boolean. Berikut adalah penjelasan arti kode dan keluarannya:

- " $x == y$ " memeriksa apakah nilai  $x$  sama dengan nilai  $y$ .
- " $x != y$ " memeriksa apakah nilai  $x$  tidak sama dengan nilai  $y$ .
- " $x > y$ " memeriksa apakah nilai  $x$  lebih besar dari nilai  $y$ .
- " $x >= y$ " memeriksa apakah nilai  $x$  lebih besar atau sama dengan nilai  $y$ .
- " $x < y$ " memeriksa apakah nilai  $x$  lebih kecil dari nilai  $y$ .
- " $x <= y$ " memeriksa apakah nilai  $x$  lebih kecil atau sama dengan nilai  $y$ .
- " $x \text{ is } y$ " memeriksa apakah  $x$  adalah objek yang sama dengan  $y$ .
- " $x \text{ is not } y$ " memeriksa apakah  $x$  bukan objek yang sama dengan  $y$ .

Dengan  $x = 1$  dan  $y = 0$ , hasil ekspresi boolean dari setiap perbandingan adalah sebagai berikut:

- " $x == y$ " akan menghasilkan False.
- " $x != y$ " akan menghasilkan True.
- " $x > y$ " akan menghasilkan True.
- " $x >= y$ " akan menghasilkan True.
- " $x < y$ " akan menghasilkan False.
- " $x <= y$ " akan menghasilkan False.
- " $x \text{ is } y$ " akan menghasilkan False.

- "x is not y" akan menghasilkan True.

## MATERI 2 : Bentuk – bentuk percabangan

Percabangan pada Python secara umum ada tiga bentuk, yaitu: conditional, alternative dan chained conditional. Bentuk conditional secara umum dinyatakan dalam kode program sebagai berikut:

```
if <kondisi>:  
    <pernyataan>
```

Sebagai contoh, jika nilai akhir > 70 maka akan lulus. Kode programnya sebagai berikut:

```
if nilai_akhir > 80:  
    print("Selamat, Anda Lulus!")
```

Bentuk alternative conditional adalah bentuk percabangan yang memiliki dua alternatif langkah yang harus dijalankan berdasarkan kondisi tertentu. Contohnya seperti ini dengan code lanjutan atas:

```
if nilai_akhir > 80:  
    print("Selamat, Anda Lulus!")  
else:  
    print("Maaf, Anda Tidak Lulus.")
```

Sebagai contoh, jika nilai akhir > 80, tampilkan tulisan Lulus. Jika tidak, tampilkan tulisan Tidak Lulus. Pada contoh ini, ada dua kemungkinan tulisan yang muncul, yaitu Lulus atau Tidak Lulus.

Bentuk chained conditional digunakan jika kemungkinan langkah yang harus dijalankan berikutnya lebih dari dua.

```
if nilai_akhir >= 80:  
    print("A")  
elif nilai_akhir >= 75:  
    print("A-")  
elif nilai_akhir >= 70:  
    print("B+")  
elif nilai_akhir >= 65:  
    print("B")  
elif nilai_akhir >= 60:  
    print("B-")  
elif nilai_akhir >= 55:  
    print("C+")  
elif nilai_akhir >= 50:  
    print("C")  
else:  
    print("E")
```

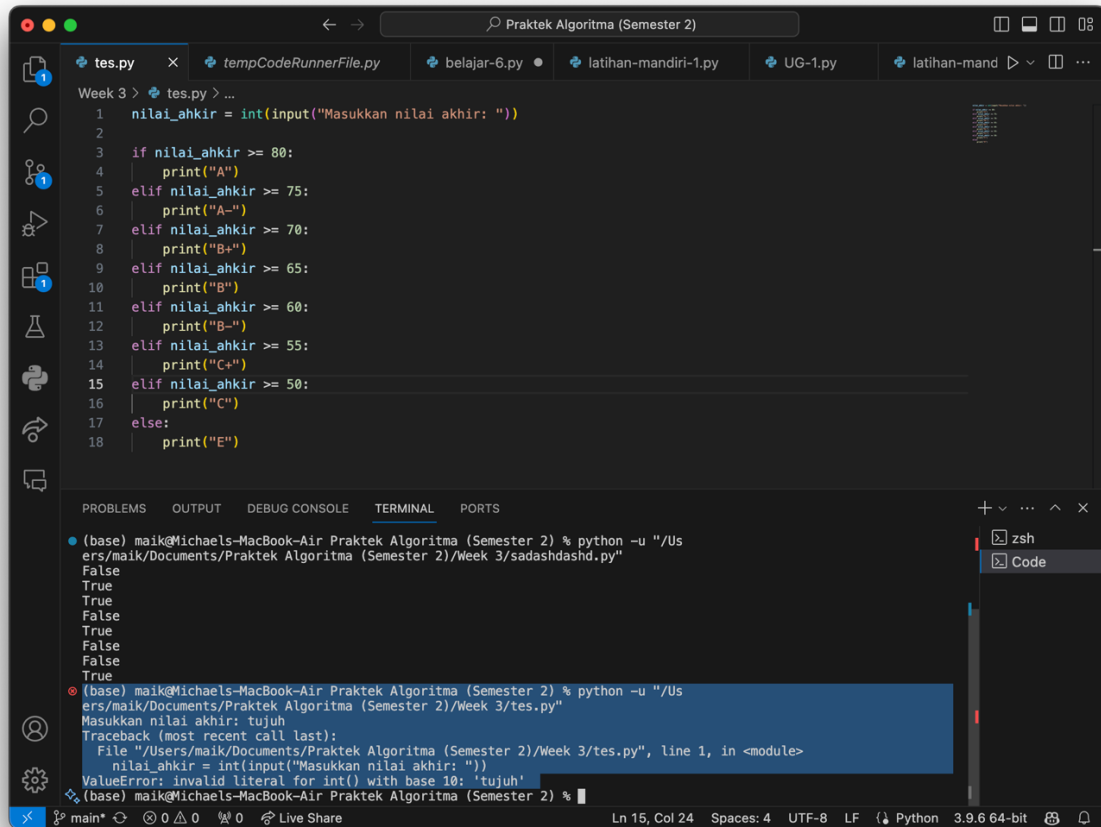
Pada kasus ini, Jika nilai numerik (nilai\_akhir) lebih besar atau sama dengan 80, maka program akan mencetak "A". Jika tidak, maka akan diuji kondisi berikutnya, yaitu jika nilai\_akhir lebih besar atau sama dengan 75, program akan mencetak "A-". Proses ini akan terus dilakukan untuk setiap kondisi yang tercantum. Jika tidak ada kondisi yang terpenuhi, program akan mencetak "E". Dengan demikian, code ini memberikan penilaian huruf berdasarkan kisaran nilai numerik yang diberikan dalam variabel nilai\_akhir.

### Tenary Operator

```
bilangan = int(input("Masukkan suatu bilangan: "))  
result = "Positif" if bilangan > 0 else ("Negatif" if bilangan < 0 else "Nol")  
print(result)
```

Code ini merupakan contoh penggunaan ekspresi kondisional (ternary operator) dalam bahasa pemrograman Python. Program ini meminta pengguna untuk memasukkan suatu bilangan, kemudian akan mengecek kondisi bilangan tersebut. Jika bilangan tersebut lebih besar dari 0, maka variabel result akan berisi string "Positif". Jika tidak, program akan mengecek apakah bilangan tersebut kurang dari 0. Jika ya, variabel result akan berisi string "Negatif". Jika kedua kondisi sebelumnya tidak terpenuhi, artinya bilangan tersebut sama dengan 0, maka variabel result akan berisi string "Nol". Akhirnya, hasil dari variabel result akan dicetak. Dengan demikian, program ini memberikan output berupa status bilangan yang dimasukkan, apakah positif, negatif, atau nol.

## Materi 3 : Kesalahan user Ketika menginput

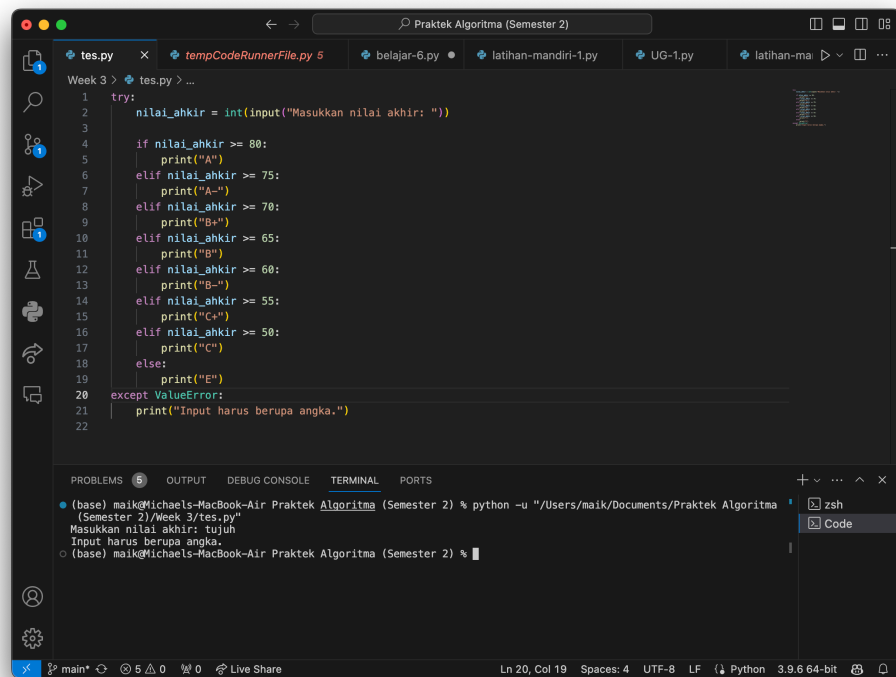


```
Week 3 > tes.py > ...
1  nilai_akhir = int(input("Masukkan nilai akhir: "))
2
3  if nilai_akhir >= 80:
4      print("A")
5  elif nilai_akhir >= 75:
6      print("A-")
7  elif nilai_akhir >= 70:
8      print("B+")
9  elif nilai_akhir >= 65:
10     print("B")
11  elif nilai_akhir >= 60:
12     print("B-")
13  elif nilai_akhir >= 55:
14     print("C+")
15  elif nilai_akhir >= 50:
16     print("C")
17  else:
18     print("E")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
• (base) maik@Michaels-MacBook-Air Praktek Algoritma (Semester 2) % python -u "/Users/maik/Documents/Praktek Algoritma (Semester 2)/Week 3/sadashdashd.py"
False
True
True
False
True
False
True
False
True
• (base) maik@Michaels-MacBook-Air Praktek Algoritma (Semester 2) % python -u "/Users/maik/Documents/Praktek Algoritma (Semester 2)/Week 3/tes.py"
Masukkan nilai akhir: tujuh
Traceback (most recent call last):
  File "/Users/maik/Documents/Praktek Algoritma (Semester 2)/Week 3/tes.py", line 1, in <module>
    nilai_akhir = int(input("Masukkan nilai akhir: "))
ValueError: invalid literal for int() with base 10: 'tujuh'
(base) maik@Michaels-MacBook-Air Praktek Algoritma (Semester 2) %
```

Gambar 1 : Kesalahan memasukkan input

Code diatas merupakan salah satu contoh kesalahan yang terjadi kepada user ketika proses penginputan. Pada code diatas perintah input diminta untuk memasukkan angka, namun jika pengguna memasukkan kata "tujuh", yang bukan merupakan angka, akan menyebabkan kesalahan karena variabel yang tidak sesuai. Dalam situasi ini, **try** dan **except** dapat digunakan untuk menangani kesalahan input dan memberitahu pengguna apa yang salah dengan input mereka. Sebagai contoh:



```
1 try:
2     nilai_akhir = int(input("Masukkan nilai akhir: "))
3
4     if nilai_akhir >= 80:
5         print("A")
6     elif nilai_akhir >= 75:
7         print("A-")
8     elif nilai_akhir >= 70:
9         print("B+")
10    elif nilai_akhir >= 65:
11        print("B")
12    elif nilai_akhir >= 60:
13        print("B-")
14    elif nilai_akhir >= 55:
15        print("C+")
16    elif nilai_akhir >= 50:
17        print("C")
18    else:
19        print("E")
20 except ValueError:
21     print("Input harus berupa angka.")
22
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS

(base) maik@Michaels-MacBook-Air Praktek Algoritma (Semester 2) % python -u "/Users/maik/Documents/Praktek Algoritma (Semester 2)/Week 3/tes.py"  
Masukkan nilai akhir: tujuh  
Input harus berupa angka.  
(base) maik@Michaels-MacBook-Air Praktek Algoritma (Semester 2) %

Gambar 2 : Perbaikan agar tidak error.

Kode di atas menggunakan blok **try** dan **except** untuk menangani kesalahan saat pengguna memasukkan input yang bukan angka saat diminta untuk memasukkan nilai akhir. Jika pengguna memasukkan teks yang tidak dapat diubah menjadi bilangan bulat, seperti "tujuh" atau "tiga belas", maka akan terjadi error **ValueError**. Dalam blok **except**, program mencetak pesan "Input harus berupa angka.", memberi tahu pengguna tentang jenis input yang diharapkan. Dengan demikian, pengguna diberi petunjuk untuk memasukkan input yang sesuai dengan yang diharapkan oleh program.

## BAGIAN 2: LATIHAN MANDIRI (60%)

### SOAL 1

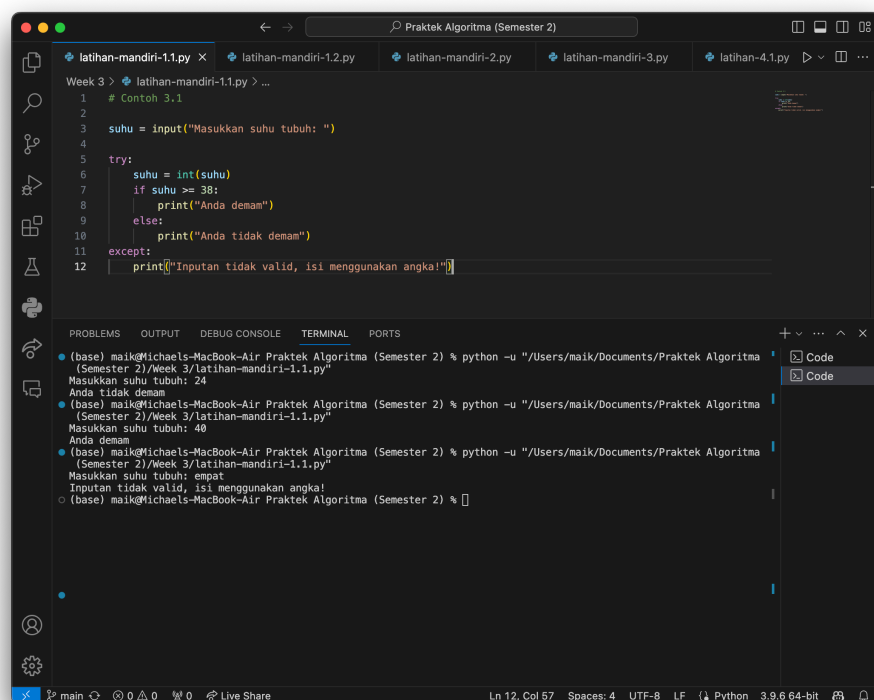
Implementasikan penanganan kesalahan input pengguna dari program-program pada Contoh 3.1, 3.2 dan 3.3.

#### A. Source Code dan Output

##### ▪ Contoh 3.1

```
# Contoh 3.1
suhu = int(input("Masukkan suhu tubuh: "))
try:
    if suhu >= 38:
        print("Anda demam")
    else:
        print("Anda tidak demam")
except:
    print("Inputan tidak valid, isi menggunakan angka!")
```

Berikut source code dan output ketika sudah diinputkan



Gambar 3 : Penyelesaian Source Code Contoh 3.1

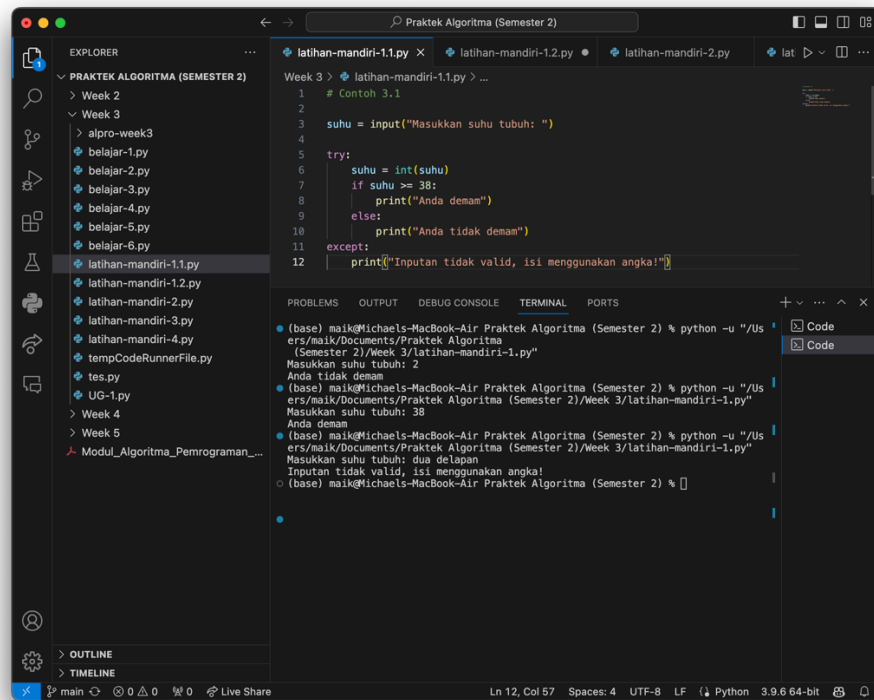
- Contoh 3.2

```
# Contoh 3.2

bilangan = input("Masukkan suatu bilangan: ")

try:
    bilangan = int(bilangan)
    if bilangan > 0:
        print("Positif")
    elif bilangan < 0:
        print("Negatif")
    elif bilangan == 0:
        print("Nol")
except:
    print("Inputan tidak valid, isi menggunakan angka!")
```

Berikut source code dan output ketika sudah diinputkan



Gambar 4 : Penyelesaian Source Code Contoh 3.2



- Contoh 3.3

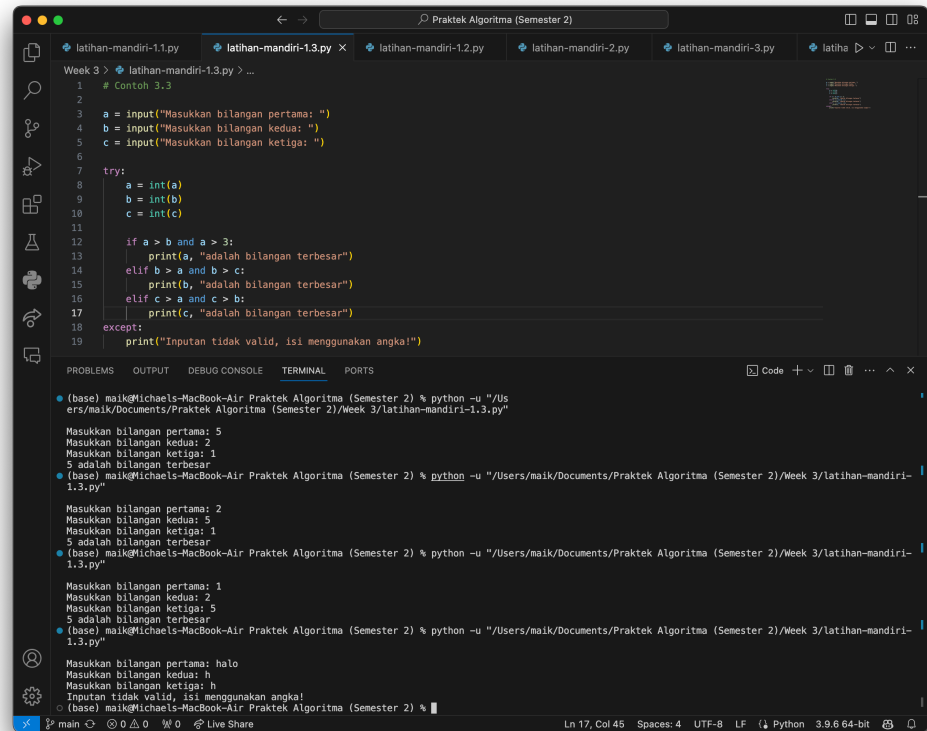
```
# Contoh 3.3

a = input("Masukkan bilangan pertama: ")
b = input("Masukkan bilangan kedua: ")
c = input("Masukkan bilangan ketiga: ")

try:
    a = int(a)
    b = int(b)
    c = int(c)

    if a > b and a > c:
        print(a, "adalah bilangan terbesar")
    elif b > a and b > c:
        print(b, "adalah bilangan terbesar")
    elif c > a and c > b:
        print(c, "adalah bilangan terbesar")
except:
    print("Inputan tidak valid, isi menggunakan angka!")
```

Berikut source code dan output ketika sudah diinputkan



Gambar 5 : Penyelesaian Source Code Contoh 3.3

## B. Penjelasan

Dari ketiga Contoh diatas, kemungkinan kesalahan pengguna kadang-kadang mungkin salah memasukkan input karena kesalahan dalam tipe data atau kesalahan pengetikan. Namun, kita dapat mengatasi masalah ini dengan menggunakan blok **try** dan **except** dalam code. Dengan cara ini, jika terjadi kesalahan saat mencoba memproses input pengguna, program tidak akan berhenti secara paksa, melainkan akan menangani kesalahan tersebut dengan memberi tahu pengguna tentang masalah yang terjadi, sehingga mereka dapat memperbaikinya.

## SOAL 2

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

### A. Source Code dan Output

Source Code :

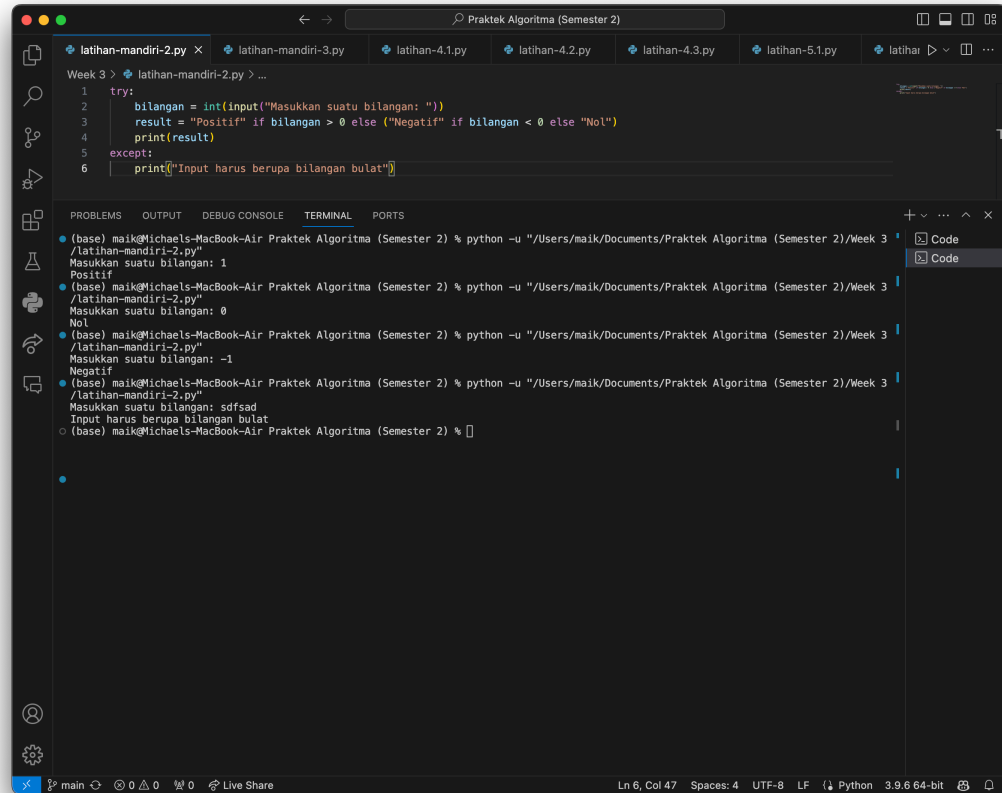
```

try:
    bilangan = int(input("Masukkan suatu bilangan: "))
    result = "Positif" if bilangan > 0 else ("Negatif" if bilangan < 0 else "Nol")
    print(result)
except:

```

```
print("Input harus berupa bilangan bulat")
```

Berikut source code dan output ketika sudah diinputkan dengan beberapa skenario :



The screenshot shows a code editor with a file named 'latihan-mandiri-2.py'. The code is a Python script that prompts the user to enter a number and then checks if it is positive, negative, or zero using a ternary operator. The terminal output shows three test cases: a positive number (1), zero (0), and a negative number (-1), each resulting in the correct classification. A fourth test case with an invalid input 'sdfsad' results in an error message.

```
1 try:
2     bilangan = int(input("Masukkan suatu bilangan: "))
3     result = "Positif" if bilangan > 0 else ("Negatif" if bilangan < 0 else "Nol")
4     print(result)
5 except:
6     print("Input harus berupa bilangan bulat")
```

Terminal Output:

```
(base) maik@Michaels-MacBook-Air Praktek Algoritma (Semester 2) % python -u "/Users/maik/Documents/Praktek Algoritma (Semester 2)/Week 3 /latihan-mandiri-2.py"
Masukkan suatu bilangan: 1
Positif
(base) maik@Michaels-MacBook-Air Praktek Algoritma (Semester 2) % python -u "/Users/maik/Documents/Praktek Algoritma (Semester 2)/Week 3 /latihan-mandiri-2.py"
Masukkan suatu bilangan: 0
Nol
(base) maik@Michaels-MacBook-Air Praktek Algoritma (Semester 2) % python -u "/Users/maik/Documents/Praktek Algoritma (Semester 2)/Week 3 /latihan-mandiri-2.py"
Masukkan suatu bilangan: -1
Negatif
(base) maik@Michaels-MacBook-Air Praktek Algoritma (Semester 2) % python -u "/Users/maik/Documents/Praktek Algoritma (Semester 2)/Week 3 /latihan-mandiri-2.py"
Masukkan suatu bilangan: sdfsad
Input harus berupa bilangan bulat
(base) maik@Michaels-MacBook-Air Praktek Algoritma (Semester 2) %
```

Gambar 6 : Penyelesaian Source Code Latihan mandiri 2

## B. Penjelasan

Ternary operator adalah cara yang lebih ringkas untuk menyatakan pernyataan IF ELSE dalam satu baris atau satu kalimat. Ini memungkinkan kita untuk menyajikan kondisi dan tindakan yang akan diambil dalam bentuk yang lebih mudah dibaca. Misalnya, kita dapat menggunakan ternary operator untuk mengecek apakah sebuah angka lebih besar dari nol, jika ya, kita mencetak "positif", jika tidak, kita cek apakah angka kurang dari nol, jika ya, kita mencetak "negatif", dan jika tidak, kita mencetak "nol". Dengan menggunakan ternary operator, kita dapat menyampaikan logika ini dalam satu kalimat yang lebih ringkas dan mudah dipahami.

## SOAL 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.

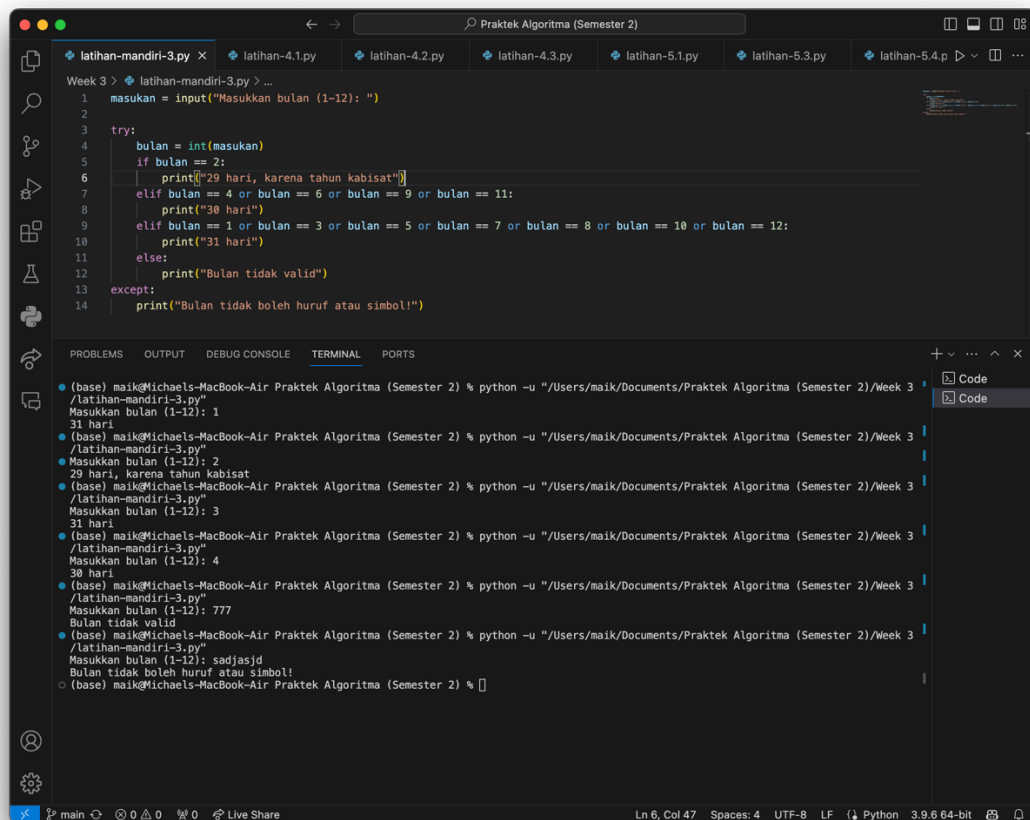
### A. Source Code dan Output

Source Code:

```
masukan = input("Masukkan bulan (1-12): ")

try:
    bulan = int(masukan)
    if bulan == 2:
        print("29 hari, karena tahun kabisat")
    elif bulan == 4 or bulan == 6 or bulan == 9 or bulan == 11:
        print("30 hari")
    elif bulan == 1 or bulan == 3 or bulan == 5 or bulan == 7 or bulan == 8 or bulan == 10 or bulan == 12:
        print("31 hari")
    else:
        print("Bulan tidak valid")
except:
    print("Bulan tidak boleh huruf atau simbol!")
```

Berikut source code dan output ketika sudah diinputkan dengan beberapa skenario :



The screenshot shows a code editor with the following source code:

```
1 masukan = input("Masukkan bulan (1-12): ")
2
3 try:
4     bulan = int(masukan)
5     if bulan == 2:
6         print("29 hari, karena tahun kabisat")
7     elif bulan == 4 or bulan == 6 or bulan == 9 or bulan == 11:
8         print("30 hari")
9     elif bulan == 1 or bulan == 3 or bulan == 5 or bulan == 7 or bulan == 8 or bulan == 10 or bulan == 12:
10        print("31 hari")
11    else:
12        print("Bulan tidak valid")
13 except:
14    print("Bulan tidak boleh huruf atau simbol!")
```

The terminal output shows the following results for different inputs:

- Input: 1 → Output: 31 hari
- Input: 2 → Output: 29 hari, karena tahun kabisat
- Input: 3 → Output: 31 hari
- Input: 4 → Output: 30 hari
- Input: 777 → Output: Bulan tidak valid
- Input: sadjasjd → Output: Bulan tidak boleh huruf atau simbol!

Gambar 7 : Penyelesaian Source Code Latihan mandiri 3

## B. Penjelasan

Code ini merupakan sebuah program sederhana dalam bahasa Python yang meminta pengguna untuk memasukkan nomor bulan (antara 1 hingga 12) dan mencetak jumlah hari dalam bulan tersebut. Ketika memasukkan input bulan adalah Februari (2), program mencetak bahwa bulan tersebut memiliki 29 hari jika tahunnya adalah tahun kabisat. Jika bulan adalah April (4), Juni (6), September (9), atau November (11), program mencetak bahwa bulan tersebut memiliki 30 hari, jika bulan adalah Januari (1), Maret (3), Mei (5), Juli (7), Agustus (8), Oktober (10), atau Desember (12), program mencetak bahwa bulan tersebut memiliki 31 hari dan jika nomor bulan tidak sesuai dengan kriteria di atas, program mencetak bahwa bulan tersebut tidak valid.

Code ini juga menggunakan **try** dan **except** agar jika ketika user salah menulis atau memasukkan inputan program akan menangkap kesalahan tersebut dengan blok **except** dan mencetak pesan yang sesuai.

## Soal 4

Sebuah program meminta pengguna memasukkan ketiga Panjang sisi suatu segitiga (berarti pengguna memasukkan tiga bilangan). Jika ketiga sisi segitiga tersebut semuanya sama, tampilkan 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".

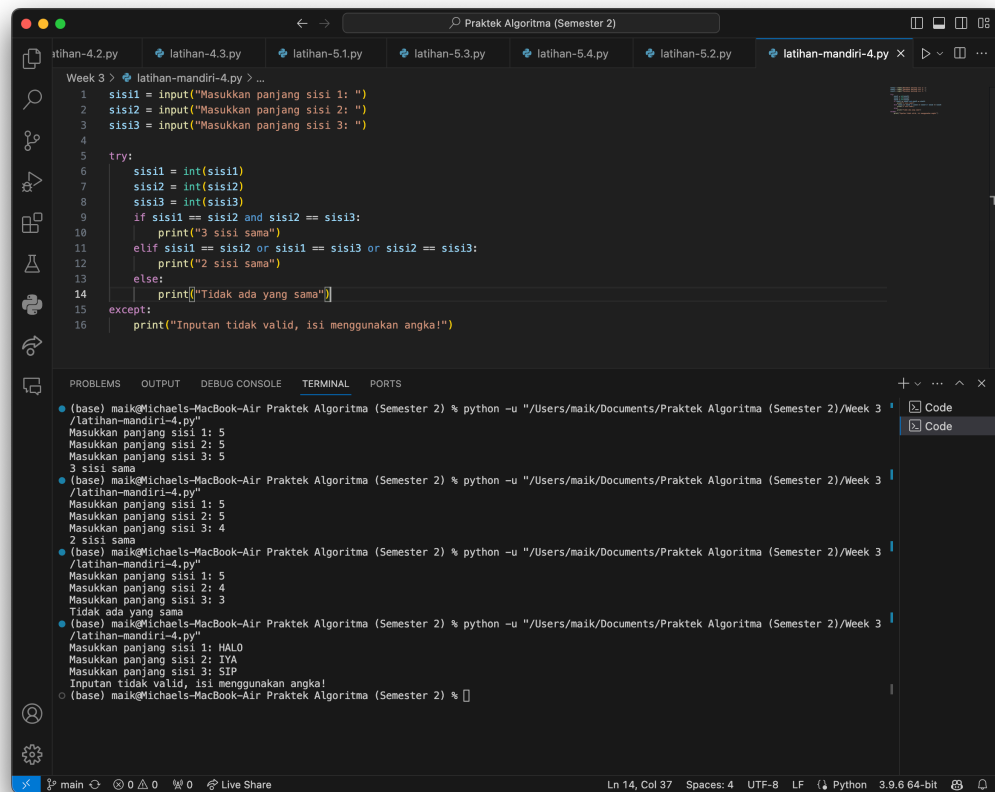
## A. Source Code dan Output

Source Code :

```
sisil = input("Masukkan panjang sisi 1: ")
sisi2 = input("Masukkan panjang sisi 2: ")
sisi3 = input("Masukkan panjang sisi 3: ")

try:
    sisi1 = int(sisi1)
    sisi2 = int(sisi2)
    sisi3 = int(sisi3)
    if sisi1 == sisi2 and sisi2 == sisi3:
        print("3 sisi sama")
    elif sisi1 == sisi2 or sisi1 == sisi3 or sisi2 == sisi3:
        print("2 sisi sama")
    else:
        print("Tidak ada yang sama")
except:
    print("Inputan tidak valid, isi menggunakan angka!")
```

Berikut source code dan output ketika sudah diinputkan dengan beberapa skenario :



```
Week 3 > latihan-mandiri-4.py > ...
1 sisi1 = input("Masukkan panjang sisi 1: ")
2 sisi2 = input("Masukkan panjang sisi 2: ")
3 sisi3 = input("Masukkan panjang sisi 3: ")
4
5 try:
6     sisi1 = int(sisi1)
7     sisi2 = int(sisi2)
8     sisi3 = int(sisi3)
9     if sisi1 == sisi2 and sisi2 == sisi3:
10        print("3 sisi sama")
11    elif sisi1 == sisi2 or sisi1 == sisi3 or sisi2 == sisi3:
12        print("2 sisi sama")
13    else:
14        print("Tidak ada yang sama")
15 except:
16    print("Inputan tidak valid, isi menggunakan angka!")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
• (base) maik@Michaels-MacBook-Air Praktek Algoritma (Semester 2) % python -u "/Users/maik/Documents/Praktek Algoritma (Semester 2)/Week 3 /latihan-mandiri-4.py"
Masukkan panjang sisi 1: 5
Masukkan panjang sisi 2: 5
Masukkan panjang sisi 3: 5
3 sisi sama
• (base) maik@Michaels-MacBook-Air Praktek Algoritma (Semester 2) % python -u "/Users/maik/Documents/Praktek Algoritma (Semester 2)/Week 3 /latihan-mandiri-4.py"
Masukkan panjang sisi 1: 5
Masukkan panjang sisi 2: 5
Masukkan panjang sisi 3: 4
2 sisi sama
• (base) maik@Michaels-MacBook-Air Praktek Algoritma (Semester 2) % python -u "/Users/maik/Documents/Praktek Algoritma (Semester 2)/Week 3 /latihan-mandiri-4.py"
Masukkan panjang sisi 1: 5
Masukkan panjang sisi 2: 4
Masukkan panjang sisi 3: 3
Tidak ada yang sama
• (base) maik@Michaels-MacBook-Air Praktek Algoritma (Semester 2) % python -u "/Users/maik/Documents/Praktek Algoritma (Semester 2)/Week 3 /latihan-mandiri-4.py"
Masukkan panjang sisi 1: HALO
Masukkan panjang sisi 2: IYA
Masukkan panjang sisi 3: SIP
Inputan tidak valid, isi menggunakan angka!
• (base) maik@Michaels-MacBook-Air Praktek Algoritma (Semester 2) %
```

Gambar 8 : Penyelesaian Source Code Latihan mandiri 4

## B. Penjelasan

Code ini adalah punya tujuan agar pengguna untuk memasukkan panjang sisi dari sebuah segitiga dan kemudian menentukan jenis segitiga berdasarkan panjang sisinya. Ketika panjang sisi-sisinya sama ( $sisi1 == sisi2 == sisi3$ ), maka program akan mencetak bahwa segitiga tersebut memiliki "3 sisi sama". Jika dua panjang sisi yang sama ditemukan ( $sisi1 == sisi2$  atau  $sisi1 == sisi3$  atau  $sisi2 == sisi3$ ), maka program akan mencetak bahwa segitiga tersebut memiliki "2 sisi sama". Jika tidak ada panjang sisi yang sama, maka program akan mencetak bahwa "Tidak ada yang sama".

Code ini juga menggunakan **try** dan **except** seperti Latihan yang lainnya agar jika ketika user salah menulis atau memasukkan inputan program akan menangkap kesalahan tersebut dengan blok **except** dan mencetak pesan yang sesuai.