



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>71231052</b>
<b>Nama Lengkap</b>	<b>GABRIEL SACHIO ATMADJAJA</b>
<b>Minggu ke / Materi</b>	<b>03 / Struktur Kontrol Percabangan</b>

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%)

Sumber Materi: Modul Pratikum Algoritma dan Pemrograman

### MATERI 3

Bentuk boolean expression hanya menghasilkan dua kemungkinan yaitu True atau False. Boolean expression dapat disusun dengan operator perbandingan:

Operator	Keterangan
<code>x == y</code>	Apakah x sama dengan y?
<code>x != y</code>	Apakah x tidak sama dengan y?
<code>x &gt; y</code>	Apakah x lebih besar dari y?
<code>x &gt;= y</code>	Apakah x lebih besar atau sama dengan y?
<code>x &lt; y</code>	Apakah x lebih kecil dari y?
<code>x &lt;= y</code>	Apakah x lebih kecil atau sama dengan y?
<code>x is y</code>	Apakah x sama dengan y?
<code>x is not y</code>	Apakah x tidak sama dengan y?

Gambar 1.1: Operator perbandingan

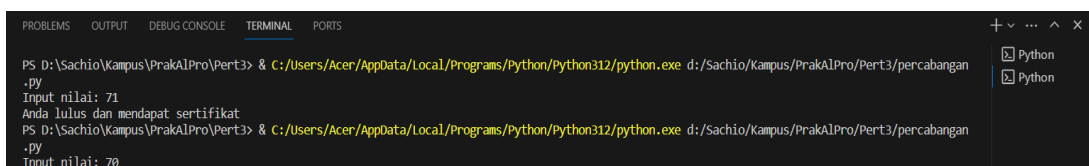
Boolean expression juga dapat digabungkan dengan logical operator yaitu **and**, **or**, dan **not**.

Percabangan pada Python ada tiga macam, yaitu conditional, alternative, dan chained conditional. Contohnya:



```
percabangan.py X
percabangan.py > ...
1  nilai_akhir = int(input("Input nilai: "))
2
3  if nilai_akhir > 70:
4      print("Anda lulus dan mendapat sertifikat")
5
```

Gambar 1.2: Contoh program percabangan conditional

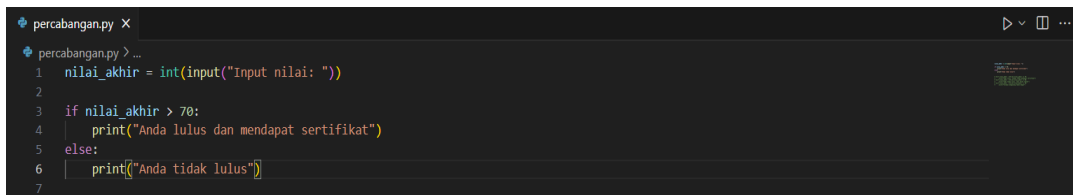


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Sachio\Kampus\PrakAlPro\Pert3> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d:/Sachio/Kampus/PrakAlPro/Pert3/percabangan
.py
Input nilai: 71
Anda lulus dan mendapat sertifikat
PS D:\Sachio\Kampus\PrakAlPro\Pert3> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d:/Sachio/Kampus/PrakAlPro/Pert3/percabangan
.py
Input nilai: 70
```

Gambar 1.3: Contoh output dari program di gambar 1.2

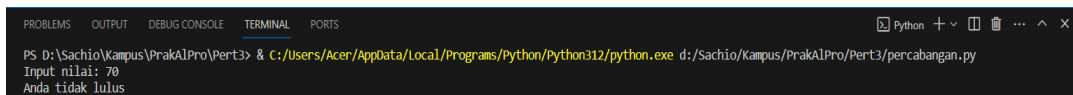
Bentuk percabangan conditional hanya menggunakan perintah `if`. Maksud dari program itu adalah jika (perintah `if`) `nilai_akhir` yang dimasukkan pengguna lebih dari 70, maka akan menampilkan output “Anda lulus dan mendapat sertifikat.” Perhatikan juga di gambar 1.3, jika user mencoba menginputkan `nilai_akhir` 70, maka tidak akan menampilkan apapun karena tidak ada statement terkait itu.

Contoh bentuk percabangan alternative:



```
percabangan.py X
percabangan.py > ...
1  nilai_akhir = int(input("Input nilai: "))
2
3  if nilai_akhir > 70:
4      print("Anda lulus dan mendapat sertifikat")
5  else:
6      print("Anda tidak lulus")
7
```

Gambar 1.4: Contoh program percabangan alternative

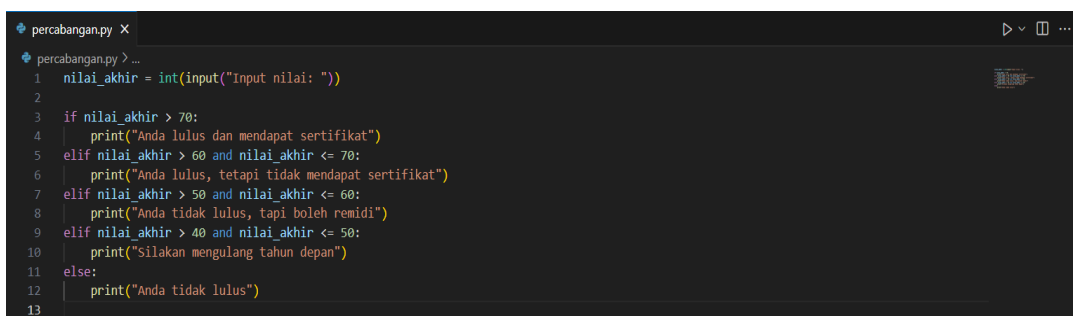


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Sachio\Kampus\PrakAlPro\Pert3> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d:/Sachio/Kampus/PrakAlPro/Pert3/percabangan.py
Input nilai: 70
Anda tidak lulus
```

Gambar 1.5: Contoh output dari program di gambar 1.4

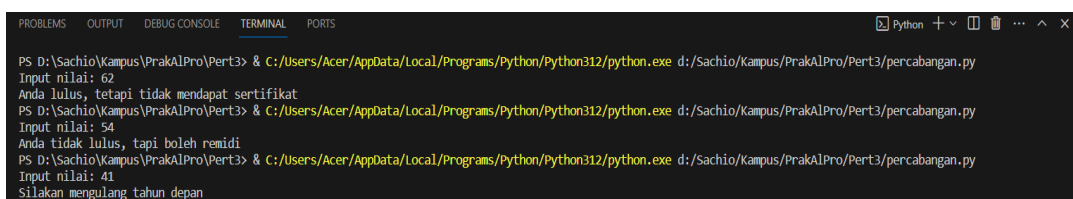
Pada percabangan alternative terdapat perintah tambahan yaitu else. Jadi, jika user meninputkan nilai\_akhir selain lebih dari 70, maka akan menampilkan output “Anda tidak lulus.”

Contoh bentuk percabangan chained conditional:



```
percabangan.py X
percabangan.py > ...
1  nilai_akhir = int(input("Input nilai: "))
2
3  if nilai_akhir > 70:
4      print("Anda lulus dan mendapat sertifikat")
5  elif nilai_akhir > 60 and nilai_akhir <= 70:
6      print("Anda lulus, tetapi tidak mendapat sertifikat")
7  elif nilai_akhir > 50 and nilai_akhir <= 60:
8      print("Anda tidak lulus, tapi boleh remidi")
9  elif nilai_akhir > 40 and nilai_akhir <= 50:
10     print("Silakan mengulang tahun depan")
11 else:
12     print("Anda tidak lulus")
13
```

Gambar 1.6: Contoh kode percabangan bentuk chained conditional



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Sachio\Kampus\PrakAlPro\Pert3> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d:/Sachio/Kampus/PrakAlPro/Pert3/percabangan.py
Input nilai: 62
Anda lulus, tetapi tidak mendapat sertifikat
PS D:\Sachio\Kampus\PrakAlPro\Pert3> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d:/Sachio/Kampus/PrakAlPro/Pert3/percabangan.py
Input nilai: 54
Anda tidak lulus, tapi boleh remidi
PS D:\Sachio\Kampus\PrakAlPro\Pert3> & C:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d:/Sachio/Kampus/PrakAlPro/Pert3/percabangan.py
Input nilai: 41
Silakan mengulang tahun depan
```

Gambar 1.7: Contoh output dari program di gambar 1.6

Dalam bentuk chained conditional, terdapat tambahan perintah yaitu elif. Perintah elif mirip dengan perintah if, sederhananya ini saling berhubungan satu sama lain. Jadi, alasan kenapa tidak menggunakan perintah if pada kondisi kedua, ketiga, dan keempat, adalah karena program tersebut ingin kondisinya saling berhubungan, jika lebih dari 60 tampilkan itu, jika lebih dari 60 dan kurang dari sama dengan 70, maka tampilkan ini. Menggunakan perintah if akan membuat percabangan baru, sedangkan perintah elif membuat percabangan dari perintah if yang ada.


Lalu dalam percabangan Python juga terdapat ternary operator, intinya membuat percabangan satu baris di text editor. Contoh:




```
ternary.py > ...
1 pembelian = int(input("Jumlah pembelian: "))
2
3 # BENTUK TERNARY
4 diskon = 0.1 if pembelian > 100000 else 0
5
```

Gambar 1.8: Contoh ternary operator

```
3 # BENTUK BIASA
4 pembelian = int(input("Jumlah pembelian: "))
5 if pembelian > 100000:
6     diskon = 0.1
7 else:
8     diskon = 0
9
```

Gambar 1.9: Contoh bentuk biasa dari ternary operator di gambar 1.8



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python +    X

```
PS D:\Sachio\Kampus\PrakAlPro\Pert3> & C:\Users\Acer\AppData\Local\Programs\Python\Python312\python.exe d:/Sachio/Kampus/PrakAlPro/Pert3/ternary.py
Jumlah pembelian: 125000
0.1
```


Gambar 1.10: Kedua program menghasilkan output yang sama

Perhatikan di gambar 1.10, kedua program menghasilkan output yang sama, maka dapat dikatakan juga kedua program memang sama hanya bentuk penulisannya saja yang berbeda.

Hal lain yang perlu diperhatikan adalah kemungkinan user salah dalam menginputkan. Misal saja, inputnya akan meminta dalam bentuk integer, tetapi user menginputkannya dalam bentuk string. Maka, akan terjadi error. Error tersebut tidak akan ditampilkan ke user. Contoh:

```
19 usia = int(input("Masukkan usia anda: "))
20
21 if usia <= 5:
22     print("Balita")
23 elif usia >= 6 and usia <= 11:
24     print("Kanak-kanak")
25 elif usia >12 and usia <= 25:
26     print("Remaja")
27 elif usia >= 26 and usia <= 45:
28     print("Dewasa")
29 elif usia > 45:
30     print("Lansia")
31
```

Gambar 1.11: Contoh program usia



The screenshot shows a terminal window with the following content:

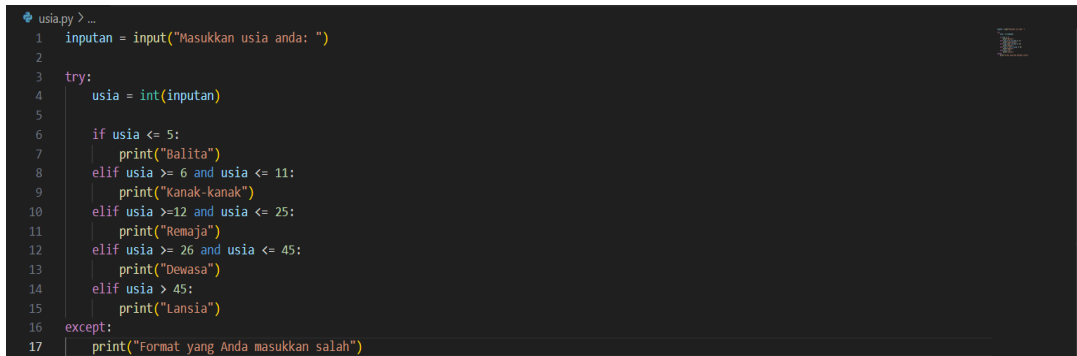
```

PS D:\Sachio\Kampus\PrakAlPro\Pert3> & C:\Users\Acer\AppData\Local\Programs\Python\Python312\python.exe d:/Sachio/Kampus/PrakAlPro/Pert3/usia.py
Masukkan usia anda: dua puluh
Traceback (most recent call last):
  File "d:\Sachio\Kampus\PrakAlPro\Pert3\usia.py", line 19, in <module>
    usia = int(input("Masukkan usia anda: "))
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
ValueError: invalid literal for int() with base 10: 'dua puluh'

```

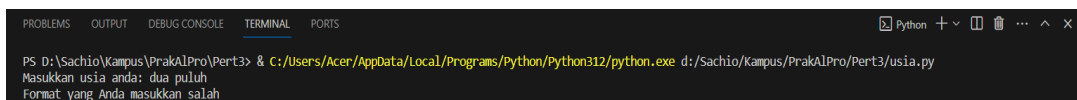
Gambar 1.12: Contoh jika user salah menginputkan

Oleh karena itu, kita bisa membuat program yang menangani kesalahan input dari pengguna. Caranya menggunakan perintah **try** dan **except**. Lebih jelasnya lihat program berikut:



```
1 inputan = input("Masukkan usia anda: ")
2
3 try:
4     usia = int(inputan)
5
6     if usia <= 5:
7         print("Balita")
8     elif usia >= 6 and usia <= 11:
9         print("Kanak-kanak")
10    elif usia >=12 and usia <= 25:
11        print("Remaja")
12    elif usia >= 26 and usia <= 45:
13        print("Dewasa")
14    elif usia > 45:
15        print("Lansia")
16 except:
17     print("Format yang Anda masukkan salah")
```

Gambar 1.13: Penggunaan perintah try dan except



```
PS D:\Sachio\Kampus\PrakAlPro\Pert3> & c:/Users/Acer/AppData/Local/Programs/Python/Python312/python.exe d:/Sachio/Kampus/PrakAlPro/Pert3/usia.py
Masukkan usia anda: dua puluh
Format yang Anda masukkan salah
```

Gambar 1.14: Error akan ditampilkan ke user

Dalam gambar 1.13, variable `inputan` digunakan untuk user menginput, lalu dalam perintah `try` terdapat variable `usia` untuk mengkonversi default ke integer dan terdapat percabangannya. Lalu dalam perintah `except`, akan menampilkan “Format yang Anda masukkan salah.” Jadi, dengan itu user akan dapat mengetahui letak errornya (gambar 1.14) semisal user salah dalam menginputkan data.

## BAGIAN 2: LATIHAN MANDIRI (60%)

Source Code: <https://github.com/gabrielsachioa/PrakAIPro3.git>

### SOAL 1

Seperti yang sudah dijelaskan di materi pada Bagian 1, untuk menangani masalah input dari pengguna, bisa dilakukan dengan menggunakan perintah **try** dan **except**. Pada perintah **try** kita isikan input-an serta kode logikanya (dalam hal ini kode percabangan) dan untuk perintah **except**, ketikkan pesan error yang nantinya ingin ditampilkan bila user salah menginput. Perhatikan juga pada Gambar 2.6, bila user salah menginput, entah itu pada variable a atau b atau c atau seluruh variabelnya, maka pesan output yang ditampilkan hanya akan menampilkan "Format bilangan a, b , atau c salah". Jadi, tidak memberi tahu secara pasti, variable mana yang salah diinputkan.

Berikut implementasi penanganan masalah input pada contoh 3.1 (*pada modul*):

```
input_suhu = input("Masukkan suhu tubuh: ")

try:
    suhu = int(input_suhu)

    if suhu >= 38:
        print("Anda demam")
    else:
        print("Anda tidak demam")

except:
    print("format yang Anda masukkan salah")
```

Python

Gambar 2.1: Implementasi penanganan masalah input

```
... format yang Anda masukkan salah
```

Gambar 2.2: Contoh output bila user salah menginputkan datanya

Berikut implementasi penanganan masalah input pada contoh 3.2 (*pada modul*):

```
input_bilangan = input("Masukkan suatu bilangan: ")

try:
    bilangan = int(input_bilangan)

    if bilangan == 0:
        print("Nol")
    elif bilangan > 0:
        print("Positif")
    elif bilangan < 0:
        print("Negatif")

except:
    print("Format yang Anda masukkan salah")
```

Python

Gambar 2.3: Implementasi penanganan masalah input

```
... Format yang Anda masukkan salah
```

Gambar 2.4: Contoh output bila user salah menginputkan data

Berikut implementasi penanganan masalah input pada contoh 3.3 (*pada modul*):

```
input_a = input("Masukkan bilangan pertama: ")
input_b = input("Masukkan bilangan kedua: ")
input_c = input("Masukkan bilangan ketiga: ")

try:
    a = int(input_a)
    b = int(input_b)
    c = int(input_c)

    if a > b and a > c:
        print("Terbesar: ", a)
    elif b > a and b > c:
        print("Terbesar: ", b)
    elif c > a and c > b:
        print("Terbesar: ", c)
except:
    print("Format bilangan a, b, atau c salah")
```

Gambar 2.5: Implementasi penanganan masalah input

```
... Format bilangan a, b, atau c salah
```

Gambar 2.6: Contoh output saat user salah menginputkan data

## SOAL 2

Bentuk normal dari Contoh 3.2 pada modul:

```
5  bilangan = int(input("Masukkan suatu bilangan: "))
6
7  if bilangan > 0:
8      print("Positif")
9  elif bilangan < 0:
10     print("Negatif")
11 elif bilangan == 0:
12     print("Nol")
```

Gambar 2.7: Kode yang sama dari contoh 3.2 (modul)

Bentuk ternary operator dari gambar 2.7 dapat dilakukan dengan mem-`print("Positif")` bila (`if`) bilangan (variable input-an) lebih besar dari 0. Lalu pada perintah `else`, buat lagi perintah `if` untuk mem-`print("Negatif")` bila bilangan lebih kecil dari 0 dan `else` untuk mem-`print("Nol")`. Berikut bentuk ternary operator dari Gambar 2.7:

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

Gambar 2.8: Contoh ternary operator

Bila kode pada Gambar 2.7 dijalankan, maka akan menghasilkan output yang sama seperti bentuk normalnya. Misal, jika saya inputkan dengan angka 0, maka akan muncul tulisan Nol.

```
... Nol
```

Gambar 2.9: Contoh output

### SOAL 3

Pertama, saya membuat variable `input_bulan` untuk user meng-input bulan yang dipilih pada tahun 2020. Lalu, permasalahan penanganan input bisa ditangani dengan perintah `try` dan `except`. Pada perintah `try`, saya isikan variable **bulan** untuk konversi default perintah input dan saya juga membuat variable **tahun** yang isinya 2020. Lalu, bulan dengan jumlah 31 hari itu terdapat di bulan ke 1, 3, 5, 7, 8, 10, dan 12, maka saya membuat variable **bulan\_hari31** untuk menampung nilai itu. Lalu bulan dengan jumlah 30 hari itu terdapat di bulan ke 4, 6, 9, dan 11, maka saya membuat variable **bulan\_hari30** untuk menampung nilainya. Lalu semisal user meng-input bulan ke-2 pada tahun 2020, saya ingin programnya mengecek sendiri apakah itu termasuk tahun kabisat atau tidak. Jika itu termasuk tahun kabisat, maka jumlah hari pada bulan ke-2 adalah 29 hari. Jika tidak, maka jumlahnya 28 hari. Jadi, cara tahu itu kabisat atau tidak adalah bila tahun habis dibagi 400, itu termasuk kabisat, lalu bila tahun habis dibagi 4 dan tidak habis dibagi 100 itu juga termasuk kabisat, selain dari itu bukan kabisat. Setelah mendapat logikanya, tinggal buat percabangannya dan perhatikan juga pada program, terdapat perintah `in` untuk mengecek apakah bulan yang diinput user terdapat di variable `bulan_hari31` atau `bulan_hari30`, jika ya jalankan sesuai program. Perintah `else`, saya isikan kode untuk menampilkan “Bulan yang Anda masukkan salah”, semisal user benar dalam menginput format, tetapi input bulannya di luar dari pilihan yang ada dan semestinya (contoh: 13, 14, dan sebagainya). Dan untuk perintah `except`, saya isikan kode untuk menampilkan pesan error “Format yang Anda masukkan salah” karena user salah menginput format (contohnya dalam string atau sebagainya).

Berikut kode programnya:

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

try:
    bulan = int(input_bulan)
    tahun = 2020

    bulan_hari31 = 1, 3, 5, 7, 8, 10, 12
    bulan_hari30 = 4, 6, 9, 11

    if bulan in bulan_hari31:
        print("Jumlah hari: 31")
    elif bulan == 2:
        if (tahun % 400 == 0 or (tahun % 4 == 0 and not tahun % 100 == 0)):
            print("Jumlah hari: 29")
        else:
            print("Jumlah hari: 28")
    elif bulan in bulan_hari30:
        print("Jumlah hari: 30")
    else:
        print("Bulan yang Anda masukkan salah")

except:
    print("Format yang Anda masukkan salah")
```

Gambar 2.9: Contoh program

```
... Jumlah hari: 31
```

Gambar 2.10: Output semisal input-nya terdapat di variable `bulan_hari31`

```
... Jumlah hari: 29
```

Gambar 2.11: Output semisal input-nya 2



```
... Jumlah hari: 30
```

Gambar 2.12: Output semisal input-nya terdapat di variable bulan\_hari30

```
... Bulan yang Anda masukkan salah
```

Gambar 2.13: Output bila user meng-input-kan di luar dari pilihan bulan yang ada

```
... Format yang Anda masukkan salah
```

Gambar 2.14: Penanganan masalah input

#### SOAL 4

Penanganan masalah menggunakan perintah try dan except. Saya membuat tiga inputan dan juga membuat variable sisi\_1, sisi\_2, dan sisi\_3 untuk membuat default input menjadi int. Lalu, cara menampilkan 3 sisi sama adalah dengan menggunakan operator == untuk membandingkan nilai dan tipe dari ketiga variable yang nanti diinput user. Lalu untuk menampilkan 2 sisi sama adalah dengan membuat seluruh kemungkinan 2 bilangan sama dari 3 variable ( sisi 1 dengan sisi 2, sisi 2 dengan sisi 3, dan sisi 1 dengan sisi 3), serta saya menggunakan operator logika yaitu **or** untuk menghubungkan kemungkinan itu. Jika kondisi terpenuhi dari 3 kemungkinan yang ada, maka jalankan perintah print("2 sisi sama"). Lalu, selain dari itu atau perintah else, maka print("tidak ada yang sama").

Berikut kode programnya:

```
input_sisi1 = input("Masukkan sisi 1: ")
input_sisi2 = input("Masukkan sisi 2: ")
input_sisi3 = input("Masukkan sisi 3: ")

try:
    sisi_1 = int(input_sisi1)
    sisi_2 = int(input_sisi2)
    sisi_3 = int(input_sisi3)

    if(sisi_1 == sisi_2 == sisi_3):
        print("3 sisi sama")
    elif(sisi_1 == sisi_2 != sisi_3) or (sisi_1 != sisi_2 == sisi_3) or (sisi_1 == sisi_3 != sisi_2):
        print("2 sisi sama")
    else:
        print("Tidak ada yang sama")

except:
    print("Format sisi 1, 2, atau 3 salah")
```

Python

Gambar 2.15: Contoh program

```
... 2 sisi sama
```

Gambar 2.16: Contoh output bila user meng-input-kan dua bilangan yang sama