



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	71251214
Nama Lengkap	Raffiel Gideon Nyolo Nyolo
Minggu ke / Materi	04/Boolean Expression dan Logical Operator

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

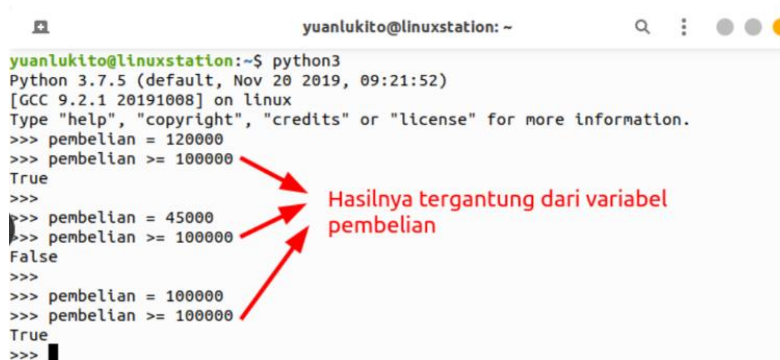
BAGIAN 1: MATERI MINGGU INI (40%)

Boolean Expression dan Logical Operator

Perhatikan kasus Voucher diskon 30% dapat dipakai jika minimum pembelian adalah Rp. 100.000. Minimum pembelian adalah syarat yang harus dipenuhi untuk mendapatkan diskon. Minimum pembelian tersebut dapat dinyatakan dalam Python sebagai berikut:

```
pembelian >= 100000
```

Bentuk tersebut dinamakan boolean expression, karena hasilnya hanya ada dua kemungkinan, yaitu **True** atau **False**, tergantung dari variabel pembelian. masukkan perintah tersebut ke dalam Python mode interaktif, hasilnya dapat dilihat pada Gambar 1.1.



```
yuanlukito@linuxstation: ~  
yuanlukito@linuxstation:~$ python3  
Python 3.7.5 (default, Nov 20 2019, 09:21:52)  
[GCC 9.2.1 20191008] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> pembelian = 120000  
>>> pembelian >= 100000  
True  
>>>  
>>> pembelian = 45000  
>>> pembelian >= 100000  
False  
>>>  
>>> pembelian = 100000  
>>> pembelian >= 100000  
True  
>>>
```

Hasilnya tergantung dari variabel pembelian

Gambar 1.1

Boolean expression dapat disusun menggunakan operator-operator perbandingan. Antara lain:

- $x == y$, artinya apakah x sama dengan y ?
- $x != y$, artinya apakah x tidak sama dengan y ?
- $x > y$, artinya apakah x lebih besar dari y ?
- $x >= y$, artinya apakah x lebih besar atau sama dengan y ?
- $x < y$, artinya apakah x lebih kecil dari y ?
- $x <= y$, artinya apakah x lebih kecil atau sama dengan y ?
- $x \text{ is } y$, artinya apakah x sama dengan y ?
- $x \text{ is not } y$, artinya apakah x tidak sama dengan y ?

Boolean expression harus disusun dengan baik dan memilih operator yang tepat yang sesuai dengan permasalahan. Berikut hal yang harus diperhatikan saat menyusun:

- Bentuk boolean expression hasilnya hanya 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 dapat dilihat pada tabel dibawah. Beberapa boolean expression dapat digabungkan dengan menggunakan logical operator. Logical operator pada Python adalah and, or dan not. Sebagai contoh, misalnya wahana Rollercoaster.

Contoh	Keterangan
Untuk lulus dibutuhkan IPK minimum 2.25	<code>ipk >= 2.25</code>
Golden Button hanya diberikan untuk Youtuber dengan subscriber lebih dari 1 juta	<code>subscriber > 1000000</code>
Nilai ujian Hanna adalah 75 sedangkan Robby mendapatkannilai 75. Apakah nilai keduanya sama?	<code>hanna is robbly</code>
Junaedi memiliki 10 sepatu, Ricky punya 15 sepatu dan Arnold punya 20 sepatu. Apakah gabungan sepatu Junaedi dan Ricky lebih banyak dari sepatu milik Arnold?	<code>junaedi + ricky > arnold</code>

Bentuk-bentuk Percabangan

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

```
if <kondisi>:  
    <lakukan ini>  
    <lakukan ini>  
    ...
```

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

```
if nilai_akhir > 70:  
    print("Anda lulus dan mendapatkan sertifikat kelulusan!")
```

Bentuk alternative conditional adalah bentuk percabangan yang memiliki dua alternatif langkah yang harus dijalankan berdasarkan kondisi tertentu. Secara umum bentuknya adalah sebagai berikut:

```
if <kondisi>:  
    <lakukan ini>  
    ...  
else:  
    <lakukan itu>  
    ...
```

Sebagai contoh, jika nilai akhir > 60, tampilkan tulisan Lulus. Jika tidak, tampilkan tulisan Tidak Lulus. Implementasinya sebagai berikut:

```
if nilai_akhir > 60:  
    print("Lulus")  
else:  
    print("Tidak Lulus")
```

Bentuk chained conditional digunakan jika kemungkinan langkah yang harus dijalankan berikutnya lebih dari dua. Bentuknya secara umum adalah sebagai berikut:

```
if <kondisi 1>:  
    <lakukan A1>  
    ...  
elif <kondisi 2>:  
    <lakukan B1>  
    ...  
elif <kondisi 3>:  
    <lakukan C1>  
    ...  
...  
else:
```

```
<lakukan ...>
```

```
...
```

Sebagai contoh, misalnya sebuah toko pakaian memberi diskon yang besarnya ditentukan oleh nilai pembelian anda. Untuk pembelian di atas Rp. 1.000.000 mendapatkan diskon 30%. Pembelian lebih dari Rp. 500.000 sampai Rp. 1.000.000 mendapatkan diskon 20%. Pembelian di bawah Rp. 100.000 tidak mendapatkan diskon. Kemungkinan diskon yang diberikan ada 4, yaitu: 30%, 20%, 15% dan 0% (tidak diskon). Perhatikan implementasi berikut ini:

```
if pembelian > 1000000:
    diskon = 0.3                # diskon 30%
elif pembelian > 500000 and pembelian <= 1000000:
    diskon = 0.2                # diskon 20%
else:
    diskon = 0                  # tidak ada diskon
```

Pernyataan "Pembelian lebih dari Rp. 500.000 sampai Rp. 1.000.000 mendapatkan diskon 20%" diimplementasikan dalam bentuk gabungan dari dua boolean expression.

Umumnya untuk menyatakan rentang nilai tersebut kita menggunakan logical operator `and`, karena kedua kondisi tersebut harus terpenuhi agar bernilai `True`.

Selain bentuk-bentuk percabangan tersebut, Python juga memiliki sintaks alternatif untuk menuliskan percabangan yang biasa disebut sebagai ternary operator. Sebagai contoh bentuk percabangan berikut ini:

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

Penanganan Kesalahan Input Menggunakan Exception Handling

Dalam menangani input dari pengguna, perlu diperhatikan potensi kesalahan yang mungkin terjadi sehingga program tidak bisa berjalan dengan semestinya. Mari kita perhatikan program yang meminta input tinggi badan pengguna, kemudian program akan menampilkan apakah pengguna termasuk tinggi dibawah rata-rata, tinggi normal dan tinggi di atas rata-rata. Kategori tinggi tersebut mengikuti aturan sebagai berikut:

- Tinggi tidak normal: 147 cm
- Tinggi dibawah rata-rata: 148-164 cm.
- Tinggi normal: 165–168 cm.
- Tinggi diatas rata-rata: > 168 cm.

Program dari kasus tersebut, adalah sebagai berikut:

```
tinggi = int(input("Masukkan tinggi anda: "))
if tinggi <= 147:
    print("Tinggi tidak normal")
elif tinggi >= 148 and tinggi <= 164:
    print("Tinggi dibawah rata-rata")
elif tinggi >= 165 and tinggi <= 168:
    print("Tinggi normal")
else:
    print("Tinggi diatas rata-rata")
```

Gambar 1.2: program yang menunjukkan status tinggi badan seseorang

Jika program tersebut dijalankan beberapa kali dengan input yang berbeda-beda, hasilnya telah sesuai dengan yang diharapkan. Hasil dari program tersebut dapat dilihat pada Gambar 1.3.

```
PS D:\Semester_2\tinggibadan> python -u "d:\Semester_2\tinggibadan\tinggi.py"
Masukkan tinggi anda: 145
Tinggi tidak normal
PS D:\Semester_2\tinggibadan> python -u "d:\Semester_2\tinggibadan\tinggi.py"
Masukkan tinggi anda: 167
Tinggi normal
PS D:\Semester_2\tinggibadan> python -u "d:\Semester_2\tinggibadan\tinggi.py"
Masukkan tinggi anda: 170
Tinggi diatas rata-rata
```

Gambar: 1.3: Output program kategori tinggi badan.

Dikarenakan input dari program tersebut harus sebuah float. Maka jika kita memasukan input yang tidak sesuai, maka output yang keluar adalah error.

Contoh :

```
PS D:\Semester_2\tinggibadan> python -u "d:\Semester_2\tinggibadan\tinggi.py"
Masukkan tinggi anda: seratus enam puluh satu
Traceback (most recent call last):
  File "d:\Semester_2\tinggibadan\tinggi.py", line 1, in <module>
    tinggi = int(input("Masukkan tinggi anda: "))
ValueError: invalid literal for int() with base 10: 'seratus enam puluh satu'
```

Gambar 1.4: input yang tidak sesuai

Bagaimana cara menangani input yang tidak sesuai? Salah satu cara yang dapat digunakan adalah menggunakan try dan except. Penggunaannya dalam kasus kategori tinggi dapat dilihat pada source code program berikut ini:

```
inputuser = input("Masukkan tinggi anda: ")
try:
    tinggi = int(inputuser)
    if tinggi <= 147:
        print("Tinggi tidak normal")
    elif tinggi >= 148 and tinggi <= 164:
        print("Tinggi dibawah rata-rata")
    elif tinggi >= 165 and tinggi <= 168:
        print("Tinggi normal")
    else:
        print("Tinggi diatas rata-rata")
except:
    print ("Anda salah memasukkan input tinggi!!")
```

Gambar: 1.5: penggunaan try except

Jika dijalankan dan diberi input yang tidak sesuai, program tidak mengalami kesalahan seperti sebelumnya. Hasilnya jika dijalankan dapat dilihat pada Gambar 1.6.

```
PS D:\Semester_2\tinggibadan> python -u "d:\Semester_2\tinggibadan\tinggi.py"
Masukkan tinggi anda: seratus enam puluh satu
Anda salah memasukkan input tinggi!!
```

Gambar 1.6: Output program kategori tinggi badan.

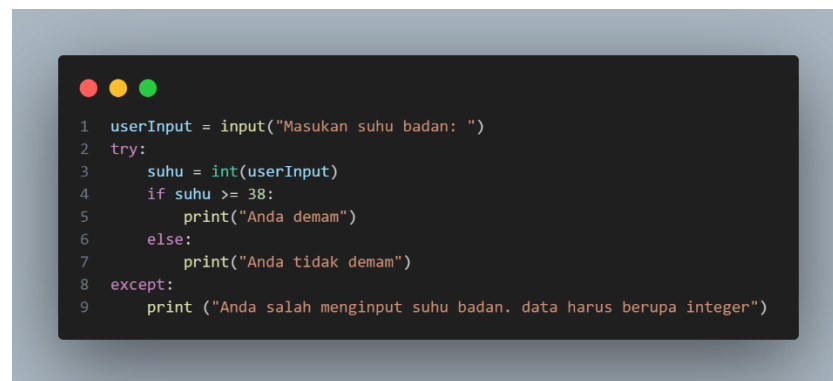
BAGIAN 2: LATIHAN MANDIRI (60%)

Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

Link Github: https://github.com/fieliel-star/71251214_raffiel.git

Catatan: Penulis menggunakan extender yang bernama CodeSnap untuk mengambil *screenshot* pada program.

SOAL 1.1

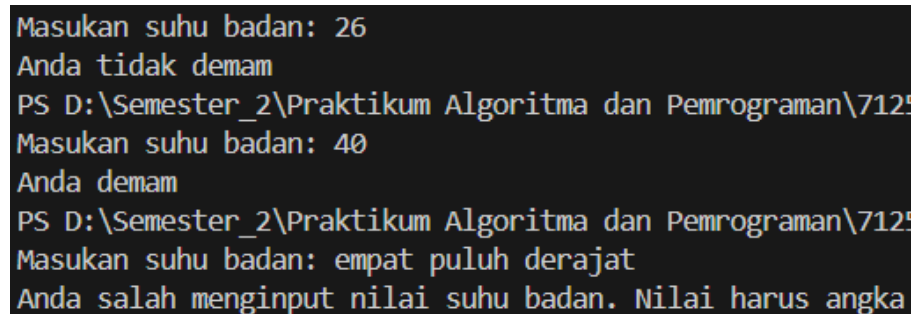


```
1 userInput = input("Masukan suhu badan: ")
2 try:
3     suhu = int(userInput)
4     if suhu >= 38:
5         print("Anda demam")
6     else:
7         print("Anda tidak demam")
8 except:
9     print ("Anda salah menginput suhu badan. data harus berupa integer")
```

Penjelasan:

Program di atas adalah program yang percabangan yang mengecek apakah pengguna yang memasukan suhu mengalami demam atau tidak. Dan pada program tersebut menggunakan **try** dan **except** untuk mengecek apakah terjadi kesalahan saat pengguna memasukan nilai suhu badan yang tidak sesuai ketentuan. Output dari program dapat dilihat di bagian bawah teks ini.

Output:



```
Masukan suhu badan: 26
Anda tidak demam
PS D:\Semester_2\Praktikum Algoritma dan Pemrograman\71251214_raffiel>
Masukan suhu badan: 40
Anda demam
PS D:\Semester_2\Praktikum Algoritma dan Pemrograman\71251214_raffiel>
Masukan suhu badan: empat puluh derajat
Anda salah menginput nilai suhu badan. Nilai harus angka
```


Jika pengguna memasukkan angka kurang dari 38, maka akan mengeluarkan “Anda tidak demam”, tetapi jika pengguna memasuka angka lebih dari 38 maka akan mengeluarkan “Anda demam” dan jika pengguna memasukkan nilai lain selain angka, misalnya string, maka yang akan keluar adalah "Anda salah menginput nilai suhu badan. Nilai harus angka". Dan ini memberitahukan kepada inputer bahwa dia melakukan kesalahan ketika memasukkan nilai.

SOAL 1.2



```
1 userInput = input("Masukan bilangan: ")
2 try:
3     bilangan = int(userInput)
4     if bilangan > 0:
5         print("Positif")
6     elif bilangan < 0:
7         print("Negatif")
8     elif bilangan == 0:
9         print("Nol")
10 except:
11     print ("Anda salah menginput nilai bilangan")
```

Penjelasan:

Program ini meminta pengguna memasukkan sebuah angka, lalu mencoba mengubahnya menjadi bilangan bulat. Jika berhasil, program akan mengecek apakah angka tersebut lebih dari nol (positif), kurang dari nol (negatif), atau sama dengan nol, dan menampilkan hasilnya. Jika pengguna memasukkan sesuatu yang bukan angka, maka bagian except akan menangani kesalahan tersebut dan menampilkan pesan bahwa input yang diberikan salah, sehingga program tidak langsung berhenti karena error. Outputnya bisa dilihat pada gambar dibawah.

Output:

```
Masukan bilangan: 0
Nol
PS D:\Semester_2\Praktikum Algoritma
Masukan bilangan: 1
Positif
PS D:\Semester_2\Praktikum Algoritma
Masukan bilangan: -1
Negatif
PS D:\Semester_2\Praktikum Algoritma
Masukan bilangan: satu
Anda salah menginput nilai bilangan
```

SOAL 1.3

```
1 userInputA = input("Masukkan bilangan pertama: ")
2 userInputB = input("Masukkan bilangan kedua: ")
3 userInputC = input("Masukkan bilangan ketiga: ")
4 try:
5     a = int(userInputA)
6     b = int(userInputB)
7     c = int(userInputC)
8     if a > b and a > c:
9         print("Terbesar: ", a)
10    elif b > a and b > c:
11        print("Terbesar: ", b)
12    elif c > a and c > b:
13        print("Terbesar: ", c)
14 except:
15    print("Anda salah menginput nilai bilangan")
```

Penjelasan:

Program ini meminta pengguna memasukkan tiga bilangan, lalu mencoba mengubah masing-masing input menjadi angka bulat menggunakan `int()` di dalam **try**. Jika konversi berhasil, program akan membandingkan ketiga angka tersebut menggunakan percabangan `if` dan `elif` untuk menentukan mana yang paling besar, kemudian menampilkan nilai terbesar tersebut. Jika pengguna memasukkan data yang bukan angka sehingga terjadi kesalahan saat konversi, bagian `except` akan menangani error tersebut dan menampilkan pesan bahwa input yang diberikan salah. Outputnya bisa dilihat pada gambar dibawah.

Output:

```
Masukkan bilangan pertama: 10
Masukkan bilangan kedua: 20
Masukkan bilangan ketiga: 14
Terbesar: 20
```

```
Masukkan bilangan pertama: dua
Masukkan bilangan kedua: 13
Masukkan bilangan ketiga: 39
Anda salah menginput nilai bilangan
```

SOAL 2

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

Penjelasan:

Program ini meminta pengguna memasukkan sebuah angka, lalu mencoba mengubahnya menjadi bilangan bulat. Jika berhasil, program akan mengecek apakah angka tersebut lebih dari nol (positif), kurang dari nol (negatif), atau sama dengan nol, dan menampilkan hasilnya. Dan kode programnya di tulis dengan **ternary operator**.

SOAL 3

```
1  userInput = input("Masukan bulan (1-12): ")
2  try:
3      bulan = int(userInput)
4      if bulan in (1, 3, 5, 7, 8, 10, 12):
5          print("Jumlah hari:",31,"hari")
6      elif bulan in (4, 6, 9, 11):
7          print("Jumlah hari:",30,"hari")
8      elif bulan == 2:
9          print("Jumlah hari:",29, "hari")
10     else:
11         print("Bulan tidak valid")
12 except:
13     print ("Anda salah menginput nilai bulan. Ulang kembali!!")
```

Penjelasan:

Kode tersebut adalah program yang meminta pengguna memasukkan nomor bulan (1–12), lalu mengonversinya menjadi bilangan bulat di dalam **try** untuk mencegah error. Program kemudian menggunakan percabangan **if-elif-else** untuk menentukan jumlah hari, bulan dengan 31 hari (1, 3, 5, 7, 8, 10, 12), 30 hari (4, 6, 9, 11), dan Februari (2) yang diatur 29 hari. Jika angka di luar 1–12 dimasukkan, akan muncul pesan “Bulan tidak valid”, dan jika input bukan angka, **except** akan menampilkan pesan kesalahan agar pengguna mengulangi input.

Output:

Ini merupakan output yang akan keluar ketika memasukan bulan (1, 3, 5, 7, 8, 10, 12) yang jumlah harinya sama (31 hari).

```
PS D:\Semester_2\Praktikum
gg04\soal3.py"
Masukan bulan (1-12): 1
Jumlah hari: 31 hari
PS D:\Semester_2\Praktikum
gg04\soal3.py"
Masukan bulan (1-12): 5
Jumlah hari: 31 hari
```

Ini merupakan output yang akan keluar ketika memasukan bulan (4, 6, 9, 11) yang jumlah harinya sama (30 hari).

```
PS D:\Semester_2\Praktikum
gg04\soal3.py"
Masukan bulan (1-12): 4
Jumlah hari: 30 hari
PS D:\Semester_2\Praktikum
gg04\soal3.py"
Masukan bulan (1-12): 6
Jumlah hari: 30 hari
```

Ini merupakan output yang akan keluar ketika memasukan bulan 2 yang jumlah harinya hanya 29.

```
Masukan bulan (1-12): 2
Jumlah hari: 29 hari
```

Ini merupakan output yang akan keluar ketika memasukan bulan yang tidak termasuk didalam program dan output ketika salah memasuka nilai bulan

```
PS D:\Semester_2\Praktikum Algoritma dan Pemrograman
gg04\soal3.py"
Masukan bulan (1-12): 13
Bulan tidak valid
PS D:\Semester_2\Praktikum Algoritma dan Pemrograman
gg04\soal3.py"
Masukan bulan (1-12): tiga
Anda salah menginput nilai bulan. Ulang kembali!!
```

SOAL 4

```
1 userInputA = input("Masukkan sisi 1: ")
2 userInputB = input("Masukkan sisi 2: ")
3 userInputC = input("Masukkan sisi 3: ")
4 try:
5     sisi1 = int(userInputA)
6     sisi2 = int(userInputB)
7     sisi3 = int(userInputC)
8     if sisi1 == sisi2 == sisi3 :
9         print ("3 sisi sama")
10    elif sisi1 == sisi2 or sisi2 == sisi3 or sisi3 == sisi1:
11        print ("2 sisi sama")
12    else:
13        print ("Tidak ada yang sama")
14
15 except:
16    print ("Anda salah menginput nilai bilangan")
```

Penjelasan:

Kode tersebut adalah program Python yang meminta pengguna memasukkan tiga nilai sisi, lalu mengubahnya menjadi bilangan bulat di dalam **try** untuk menghindari kesalahan input. Program kemudian menggunakan percabangan if-elif-else untuk mengecek kesamaan nilai: jika ketiga sisi sama maka menampilkan "3 sisi sama", jika hanya dua sisi yang sama maka menampilkan "2 sisi sama", dan jika tidak ada yang sama maka menampilkan "Tidak ada yang sama". Jika input bukan angka, except akan menampilkan pesan kesalahan bahwa pengguna salah menginput nilai bilangan.

Pada percabangan tersebut, **and** digunakan untuk memastikan *semua kondisi bernilai benar sekaligus*. Pada bagian if sisi1 == sisi2 == sisi3, Python mengecek bahwa sisi1 sama dengan sisi2 dan sisi2 sama dengan sisi3, sehingga artinya ketiga sisi memiliki nilai yang sama.

Sedangkan **or** digunakan ketika *cukup salah satu kondisi saja yang benar*. Pada bagian elif sisi1 == sisi2 **or** sisi2 == sisi3 **or** sisi3 == sisi1, program akan menampilkan "2 sisi sama" jika minimal ada satu pasangan sisi yang nilainya sama. Jadi, **and** berarti semua syarat harus terpenuhi, sementara **or** berarti cukup satu syarat saja yang terpenuhi.

Output:

Ini merupakan output yang akan keluar ketika memasukan semua sisi sama.

```
Masukkan sisi 1: 1
Masukkan sisi 2: 1
Masukkan sisi 3: 1
3 sisi sama
```

Ini merupakan output yang akan keluar ketika memasukan dua sisi yang sama. Terdapat 3 kondisi pada bagian ini. Berikut outputnya.

```
ggu04\soal4.py"2\Pr
Masukkan sisi 1: 1
Masukkan sisi 2: 1
Masukkan sisi 3: 2
2 sisi sama
PS D:\Semester_2\Pr
ggu04\soal4.py"
Masukkan sisi 1: 1
Masukkan sisi 2: 2
Masukkan sisi 3: 1
2 sisi sama
PS D:\Semester_2\Pr
ggu04\soal4.py"
Masukkan sisi 1: 2
Masukkan sisi 2: 1
Masukkan sisi 3: 1
2 sisi sama
```

Ini merupakan output yang akan keluar ketika memasukan semua sisi tidak ada nilai yang sama.

```
Masukkan sisi 1: 1
Masukkan sisi 2: 2
Masukkan sisi 3: 3
Tidak ada yang sama
```

Ini merupakan output yang akan keluar ketika memasukan nilai yang tidak sesuai.

```
Masukkan sisi 1: lima
Masukkan sisi 2: 3
Masukkan sisi 3: 4
Anda salah menginput nilai bilangan
```