



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231058
Nama Lengkap	Michael Chandra Mahanaim
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

BAGIAN 1: MATERI MINGGU INI (40%)

MATERI 1: Booeelan and Logic

Booeelan adalah logic True dan False yang digunakan untuk masuk ke percabangan yang ada di dalam kode dengan IF ELSE, WHILE, dan FOR. Beberapa contoh Booeelan 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)
```

Gambar Boolean Expressions

Di kode tersebut ada print statement, Diibaratkan **x = 1** dan **y = 0**, dan karena yang di print adalah boolean expression maka yang di output pasti juga boolean expression. Arti kode dan outputnya adalah:

- "x == y" berarti x sama dengan y
- "x != y" berarti x tidak sama dengan y
- "x > y" berarti x lebih besar dari y
- "x >= y" berarti x lebih besar atau sama dengan y
- "x < y" berarti x lebih kecil dari y
- "x <= y" berarti x lebih kecil atau sama dengan y
- "x is y" berarti x adalah y
- "x is not y" berarti x adalah bukan y

```
False
True
True
True
False
False
False
True
```

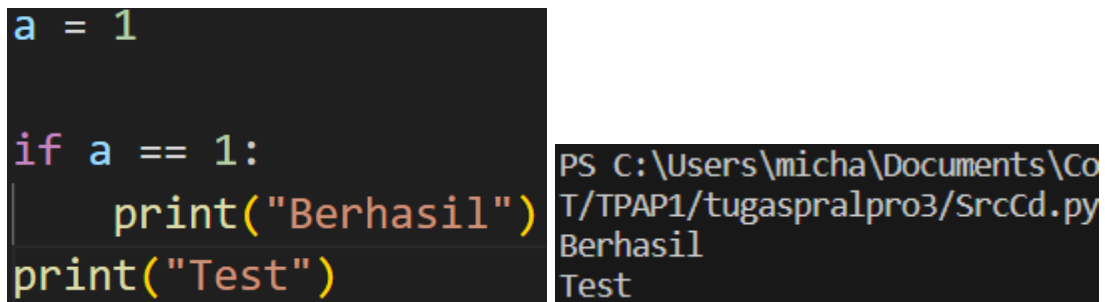
Boolean expression bukan hanya berguna tetapi vital dan sangat penting dalam percabangan, karena percabangan memerlukan True atau False untuk dapat masuk ke kode yang ada di dalam, maka tanpa Booeelan expression maka IF ELSE tidak dapat berfungsi, sama seperti WHILE dan FOR loops juga.

MATERI 2: IF Types

Ada 3 tipe percabangan dalam Python, yaitu:

A. Conditional:

Tipe conditional hanya menggunakan IF dan syarat serta apa yang dilakukan jika syarat terpenuhi (TRUE). Contoh mudah untuk penggunaan dan output tipe ini adalah:



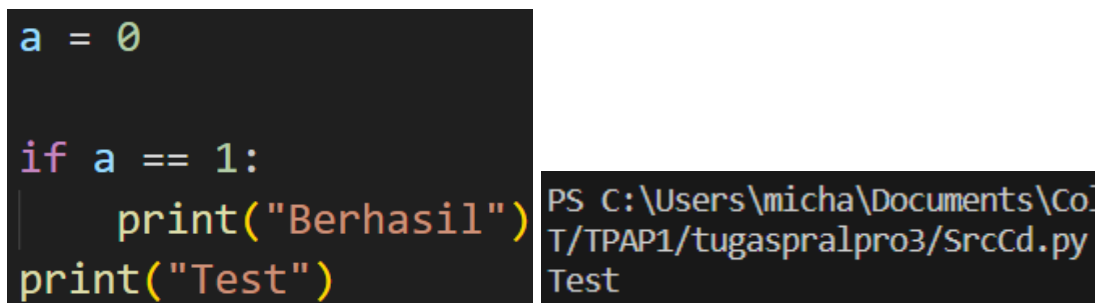
```
a = 1

if a == 1:
    print("Berhasil")
print("Test")
```

PS C:\Users\micha\Documents\Co
T/TPAP1/tugaspralpro3/SrcCd.py
Berhasil
Test

Gambar penggunaan IF conditional dan Outputnya

Disebelah kiri adalah sintaks yang digunakan untuk IF conditional dan disebelah kanan adalah outputnya. Karena `a == 1` terpenuhi maka `print "Berhasil"` akan keluar dalam output. Jika `a == 1` tidak terpenuhi maka outputnya akan terjadi seperti ini:



```
a = 0

if a == 1:
    print("Berhasil")
print("Test")
```

PS C:\Users\micha\Documents\Co
T/TPAP1/tugaspralpro3/SrcCd.py
Test

Gambar penggunaan IF conditional yang tidak terpenuhi dan Outputnya

Karena `a == 1` tidak terpenuhi maka pada output, Berhasil tidak muncul dan sintaks if dilompati dan langsung ke `print "Test"`. IF conditional mempermudah jika output yang dikeluarkan bergantung dengan input user atau kondisi yang diberikan.

B. Alternative Conditional:

Alternative conditional sama seperti normal Conditional tetapi hanya menambahkan sintaks ELSE untuk menangani jika conditional IF tidak terpenuhi. Contohnya seperti di bawah ini:

<pre>t = 2 if t == 0: print("Nice") else: print("What?") print("Hello")</pre>	<pre>t = 5 if t == 0: print("Nice") else: print("What?") print("Hello")</pre>
--	--

Gambar penggunaan Alternative Conditional

Di kode ini variabel t memiliki 2 value, dan IF conditionalnya menggunakan `t == 0`. Karena conditional tidak terpenuhi, maka dengan menggunakan ELSE output yang dikeluarkan dapat dipermudah dan dijadikan 1. Output yang keluar dari kedua kode adalah:

```
PS C:\Users\micha\Documents\Co...
T/TPAP1/tugaspralpro3/SrcCd.py
what?
Hello
```

Gambar output Alternative Conditional dengan ELSE

ELSE mempermudah output yang dapat dikeluarkan jika user memberikan input diluar yang di ekspektasi atau kondisi yang terjadi tidak memenuhi if.

C. Chained Conditional:

Chained conditional menggunakan ELIF (ELSE IF) untuk membuat rantai Conditional yang bertumpuk jika kode menginginkan banyak kondisi tertentu. Contoh bagus untuk penggunaan ELIF ini adalah sebagai berikut:

```
l = 90

if l == 1:
    print("Whuh?")
elif l == 90:
    print("Ah, ok")
elif l > 70:
    print("Hmm...")
else:
    print("Nope")
```

Gambar penggunaan ELIF

$l = 90$, maka dari keempat cabang yang dapat dimasukkan, 2 cabang memenuhi kondisi untuk menjalankan kode selanjutnya. Tetapi dalam percabangan seperti ini, hanya conditional pertama yang terpenuhi yang dimasuki, dan kondisi yang terpenuhi satunya tidak dimasuki karena Percabangan hanya dilakukan 1 kali saja. Output yang keluar akan tampil seperti ini:

```
T/TPAP1/tugaspralpro3/SrcCd.py
Ah, ok
```

Gambar output percabangan ELIF

Kondisi yang terpenuhi yaitu $l == 90$, maka output yang keluar adalah "Ah, ok". kondisi yang satunya $l > 70$ juga terpenuhi, tetapi tidak dieksekusi karena sudah ada kondisi yang terpenuhi dan IF conditional hanya perlu dipenuhi 1 kali saja.

Ternary Operator

Ternary operator adalah penggunaan percabangan yang hanya menggunakan 1 baris dan dapat dikatakan dibaca seperti kalimat sehari-hari dan lebih mudah dibaca. Contoh sederhana adalah:

```
d = 2  
print("Whuh?") if d == 1 else print("Great") if d > 0 else print("Natha")
```

Gambar penggunaan Ternary Operator

Penggunaan seperti ini lebih mudah jika dibaca daripada dianalisis karena hanya dalam 1 baris yang berbentuk kalimat.

MATERI 3: User "Mis-Input" Handling

Dalam penggunaan input(), jika input yang dimasukkan tidak sesuai dengan variabel yang diberikan dalam input, maka kode akan mengeluarkan sistem error seperti berikut.

```
inp = int(input("Angka: "))
```

```
Angka: dua
Traceback (most recent call last):
  File "c:\Users\micha\Documents\College\2SMT\TPAP1\tugaspralpro3\SrcCd.py", line 41, in <module>
    inp = int(input("Angka: "))
          ^^^^^^^^^^^^^^^^^^^
ValueError: invalid literal for int() with base 10: 'dua'
```

Gambar input integer dan error akibat misinput user

Dalam kode di atas, sintaks input meminta angka, tetapi user mengisi angka dengan kata "dua" yang mengakibatkan error variabel yang tidak cocok. Dalam situasi seperti ini, try: dan except: dapat digunakan untuk menangani mis-input ini dan memberi tahu user apa yang salah dengan input mereka. Contoh berikut akan menjelaskan:

```
inp = input("Angka: ")

try:
    inp = int(inp)
    print("Number ", inp)
except:
    print("Please use number")
```

Gambar penggunaan try except untuk menangani misinput

Dengan try, Kode dapat dijalankan dengan normal hingga mencapai error. Jika error ditemui, maka except akan dieksekusi untuk menangani kesalahan input user, seperti gambar di bawah ini:

Angka: enam Please use number	Angka: 7 Number 7
----------------------------------	----------------------

Gambar penanganan misinput dan output input yang benar

Jika user memasukkan "enam", maka konversi input menjadi int akan terhitung error dan mengeluarkan output dari except dan memberi tahu user mengapa input salah. Sebaliknya jika input benar maka kode akan berjalan lancar dan except tidak dieksekusi.

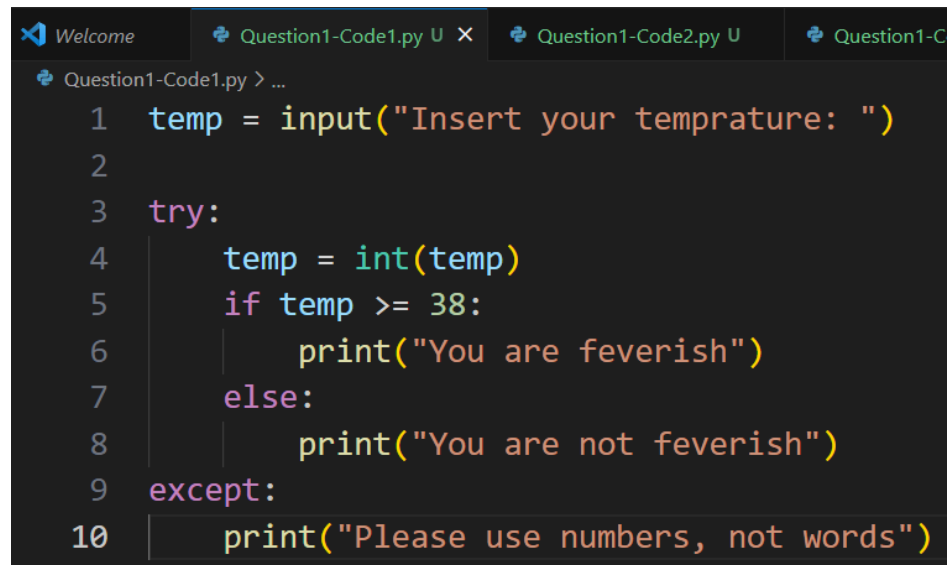
BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

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

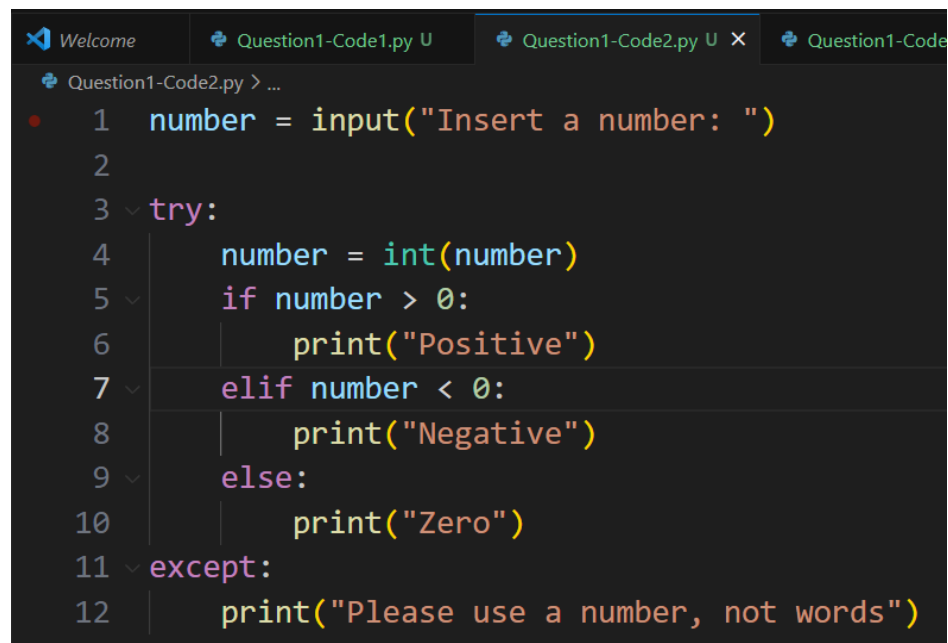
A. Source Code

a. Contoh 3.1



```
1 temp = input("Insert your temprature: ")
2
3 try:
4     temp = int(temp)
5     if temp >= 38:
6         print("You are feverish")
7     else:
8         print("You are not feverish")
9 except:
10    print("Please use numbers, not words")
```

b. Contoh 3.2



```
1 number = input("Insert a number: ")
2
3 try:
4     number = int(number)
5     if number > 0:
6         print("Positive")
7     elif number < 0:
8         print("Negative")
9     else:
10        print("Zero")
11 except:
12    print("Please use a number, not words")
```


c. Contoh 3.2

```
Question1-Code3.py > ...
1 num1 = input("Insert 1st number: ")
2 num2 = input("Insert 2nd number: ")
3 num3 = input("Insert 3rd number: ")
4
5 try:
6     num1 = int(num1)
7     num2 = int(num2)
8     num3 = int(num3)
9     if num1 > num2 and num1 > num3:
10         print("Biggest: ", num1)
11     elif num2 > num1 and num2 > num3:
12         print("Biggest: ", num2)
13     elif num3 > num1 and num3 > num2:
14         print("Biggest: ", num3)
15 except:
16     print("Please use numbers, not words")
```

B. Output Result

- a.

```
Insert your temprature: dua puluh
Please use numbers, not words
```
- b.

```
Insert a number: enam
Please use a number, not words
```
- c.

```
Insert 1st number: dua
Insert 2nd number: tiga
Insert 3rd number: enam
Please use numbers, not words
```

C. Explanation

Ketiga kode ini hanyalah kode sederhana untuk memberi tahu user jika mereka sedang demam, jika angka adalah negatif atau positif atau nol, dan mengetahui mana angka terbesar berdasarkan dengan input user. Dengan menggunakan try() dan except() penanganan kesalahan input pengguna dapat diatasi agar tidak membuat sistem error sekaligus memberi tahu user mengapa sistem tidak dapat memproses input dari user.

SOAL 2

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

A. Source Code

```
Welcome | Question1-Code1.py U | Question1-Code2.py U | Question1-Code3.py U | Question2.py U x | Question3.py U | Question4.py U
Question2.py > ...
1 number = input("Insert a number: ")
2
3 try:
4     number = int(number)
5     print("Positive") if number > 0 else print("Negative") if number < 0 else print("Zero")
6 except:
7     print("Please use a number, not words")
```

B. Output Result

Insert a number: 2 Positive	Insert a number: -20 Negative	Insert a number: 0 Zero
--------------------------------	----------------------------------	----------------------------

C. Explanation

Ternary operator merujuk penggunaan IF ELSE dalam 1 baris atau dalam 1 kalimat yang dapat dibaca secara normal dan lebih mudah dijelaskan daripada penggunaan IF ELSE yang menurun. Dengan menggunakan IF ELSE dengan cara ini kalimat dapat dibaca seperti: print "positif" jika angka lebih dari nol, jika tidak, print "negatif" jika angka kurang dari nol, jika tidak, print "zero".

SOAL 3

Buatlah sebuah program yang dapat menampilkan jumlah hari dalam suatu bulan di tahun 2020.

A. Source Code

```
Welcome | Question1-Code1.py U | Question1-Code2.py U | Question1-Code3.py U | Question2.py U | Question3.py > ...
1  mth = input("Pick a month (1-12): ")
2
3  try:
4      mth = int(mth)
5      if mth > 12:
6          print("This month doesn't exist (Invalid)")
7      elif mth == 4 or mth == 6 or mth == 9 or mth == 11:
8          print("This month has 30 days")
9      elif mth == 2:
10         print("This month has 28 days and 29 days on leap years")
11     else:
12         print("This month has 31 days")
13 except:
14     print("Please use a number, not word(s)")
```

B. Output Result

```
Pick a month (1-12): 24
This month doesn't exist (Invalid)
```

```
Pick a month (1-12): 3   Pick a month (1-12): 4
This month has 31 days   This month has 30 days
```

```
Pick a month (1-12): 2
This month has 28 days and 29 days on leap years
```

```
Pick a month (1-12): tujuh
Please use a number, not word(s)
```

C. Explanation

Kode ini hanyalah kode simpel untuk memberi tahu user jumlah hari yang ada pada bulan tertentu. Dengan try() dan except(), kode dapat memfilter masukan kode yang benar dan memberi tahu user jika inputan mereka salah dan alasan mengapa salah. Dengan percabangan dan logic, kode dapat memberikan output yang benar berdasarkan input user, seperti: Februari bulan kedua mempunyai 28/29 hari, April; Juni; September; dan November memiliki 30 hari, dan bulan lainnya memiliki 31 hari. Jika angka yang dimasukkan bukan termasuk 12 bulan maka kode akan memberikan output bahwa bulan dengan angka tersebut tidak ada.

SOAL 4

Sebuah program meminta pengguna memasukkan ketiga panjang sisi suatu segitiga (berarti pengguna memasukkan tiga bilangan).

A. Source Code

```
Welcome | Question1-Code1.py U | Question1-Code2.py U | Question1-Co
Question4.py > ...
1  s1 = input("Insert side 1: ")
2  s2 = input("Insert side 2: ")
3  s3 = input("Insert side 3: ")
4
5  try:
6      s1 = int(s1)
7      s2 = int(s2)
8      s3 = int(s3)
9
10     if s1 == s2 == s3:
11         print("Three sides are equal")
12     elif s1 == s2 or s1 == s3 or s2 == s3:
13         print("Two sides are the same")
14     else:
15         print("No sides are equal")
16 except:
17     print("Please use numbers, not words")
```

B. Output Result

Insert side 1: 12	Insert side 1: 367
Insert side 2: 12	Insert side 2: 65
Insert side 3: 33	Insert side 3: 432
Two sides are the same	No sides are equal

Insert side 1: 89	Insert side 1: enam puluh
Insert side 2: 89	Insert side 2: tiga puluh
Insert side 3: 89	Insert side 3: dua puluh
Three sides are equal	Please use numbers, not words

C. Explanation

Kode ini menggunakan 3 inputan user dalam bentuk angka dan membandingkan ketiga angka tersebut untuk menemukan berapa angka yang sama dan mengeluarkan output yang benar menggunakan percabangan. Dengan `try()` dan `except()`, kode dapat memfilter input yang salah dan memberi tahu user mengapa input salah. Kode ini hanya menggunakan comparison logic sederhana untuk membandingkan 3 angka inputan user untuk masuk ke percabangan yang benar.