



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	<71231038>
Nama Lengkap	<AMELIA AGUSTIN>
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%)

Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

### MATERI 1

#### 3.3.1 Boolean Expression dan Logical Operator

Boolean merupakan ekspresi yang dimana dapat menghasilkan TRUE dan FALSE

```
apel > 5
jeruk <= 10
```

sedangkan untuk logical operator itu sendiri merupakan simbol logika yang dimana memiliki makna DAN, ATAU, TIDAK.

```
apel > 5
jeruk <= 10
apel > 5 DAN jeruk <= 10
```

artinya - apel harus lebih dari 5 dan jeruk harus kurang dari sama dengan 10 supaya persamaannya TRUE.

Jadi untuk logical operator seperti DAN dan ATAU itu memiliki cara yang mudah untuk menggabungkan beberapa kondisi dalam satu persamaan yang besar. Program ini sangat berguna untuk membuat keputusan sendiri.

Boolean expression dapat disusun menggunakan operator-operator perbandingan seperti pada Tabel di bawah ini:

Operator	Nama	Contoh	Hasil
==	Sama dengan	$Sx == Sy$	Jika $Sx$ sama dengan $Sy$ , Benar.
===	Identik	$Sx === Sy$	Jika $Sx$ sama dengan $Sy$ , dan memiliki jenis yang sama, Benar.
!=	Tidak sama	$Sx != Sy$	Jika $Sx$ tidak sama dengan $Sy$ , Benar.
<>	Tidak sama	$Sx <> Sy$	Jika $Sx$ tidak sama dengan $Sy$ , Benar.
!==	Tidak identik	$Sx !== Sy$	Jika $Sx$ tidak sama dengan $Sy$ , atau bukan jenis yang sama, Benar.
>	Lebih besar dari	$Sx > Sy$	Jika $Sx$ lebih besar dari $Sy$ , Benar.
<	Lebih kecil dari	$Sx < Sy$	Jika $Sx$ lebih kecil dari $Sy$ , Benar.
>=	Lebih besar dari atau sama dengan	$Sx >= Sy$	Jika $Sx$ lebih besar dari atau sama dengan $Sy$ , Benar.
<=	Kurang dari atau sama dengan	$Sx <= Sy$	Jika $Sx$ kurang dari atau sama dengan $Sy$ , Benar.

Disaat akan membuat persamaan yang dimana nilainya hanya TRUE and FALSE, inilah yang harus diperhatikan sebagai berikut:

1. Hasil persamaan pasti selalu TRUE atau FALSE.
2. Mencari kata seperti minimal, maksimal, tidak lebih dari, tidak kurang dari.

3. Bandingkan nilai variabel yang tepat sesuai dengan masalahnya.
4. Gabungkan pemakaian operator DAN, ATAU, TIDAK.

Beberapa contoh permasalahan dan persamaanya:

```
# DAN (and)
stok_apel = 5
stok_jeruk = 10
syarat1 = (stok_apel > 3)
syarat2 = (stok_jeruk < 20)
print(syarat1 and syarat2) # True

# ATAU (or)
harga_1 = 5000
harga_2 = 15000
murah1 = (harga_1 < 10000)
murah2 = (harga_2 < 10000)
print(murah1 or murah2) # True

# TIDAK (not)
nilaiA = 70
nilaiB = 60
lolosA = (nilaiA >= 75)
lolosB = (nilaiB >= 75)
print(not lolosB) # True
```

## MATERI 2

### 3.3.2 Bentuk-bentuk Percabangan

Berikut merupakan bentuk-bentuk percabangan dalam python:

1. Percabangan IF merupakan bentuk percabangan yang paling mendasar. Berikut adalah syntax-nya:

```
if kondisi:
    # blok kode yang dieksekusi jika kondisi bernilai True
```

Jika kondisi bernilai TRUE, maka kode setelah if akan dieksekusi. Kondisi ini biasanya berupa boolean expression atau perbandingan seperti  $x > 5$ .

2. Percabangan IF-ELSE

If-else ini berfungsi untuk mengeksekusi dua kemungkinan kode namun tergantung kondisi.

```
if kondisi:
    # kode 1, jika kondisi True
else:
    # kode 2, jika kondisi False
```

Jadi untuk program ini akan memeriksa kondisi, lalu akan menjalankan kode 1 jika TRUE, atau bahkan kode 2 jika FALSE.

### 3. percabangan if-elif-else

Bentuk program ini untuk memeriksa beberapa kemungkinan kondisi

```
if kondisi1:
    #kode1
elif kondisi2:
    #kode2
else:
    #kode3
```

Bentuk program ini akan memeriksa kondisi 1, kondisi 2, dan seterusnya hingga munculnya TRUE. Jika tidak muncul adanya TRUE maka akan masuk ke else.

### 4. Ternary Operator

Bentuk percabangan ini sama saja dengan percabagana kode if-else namun hanya dalam satu baris saja.

Contoh bentuknya:

```
hasil = nilai_jika_benar if kondisi benar else nilai_jika_salah
penerapannya:
```

```
angka = 5
text = "Positif" if angka > 0 else "Negatif"
```

sama saja dengan kode seperti di bawah ini:

```
if angka > 0:
    text = "Positif"
else:
    text = "Negatif"
```

Dalam bentuk program ternary operator ini memiliki kelebihan dan kekurangan,berikut:

Kelebihan

1. Lebih pendek dan simple
2. Kurang dari 2 baris kode

Kekurangan

1. Sulit dibaca jika terlalu banyak atau panjang.
2. Hanya memiliki dua kemungkinan (TRUE atau FALSE)
3. Lebih disarankan menggunakan if-else untuk 2 kemungkinan

### 3.3.3 Penanganan Kesalahan Input Menggunakan Exception Handlin

Exception Handling ini digunakan untuk mengatasi kesalahan yang akan mungkin terjadi saat program dijalankan dalam python. Berikut ini merupakan konsep dasar yang terkait dengan exception handling:

1. Exceptions merupakan moment yang dimana akan terjadi selama proses eksekusi program yang akan menghentikan alur normal program. Misalnya saat user akan memasukkan input yang salah.

```
try:
    x = int(input("Please enter a number: "))
    print(x)
except ValueError:
    print("Oops! That was not a valid number. Try again...")
```

2. Catching Exception merupakan penangkapan yang mungkin terjadi dengan menggunakan blok try dan except

```
try:
    # kode yang mungkin menimbulkan exceptions
except:
    # kode untuk ditampilkan jika terjadi exceptio
```

3. Raising Exceptions secara otomatis memunculkan exceptions dengan statement raise, bisa digunakan untuk melempar exceptions saat kondisi eror yang terdeteksi.

```
def calculate_age(year_born):
    if year_born > 2024:
        raise ValueError('Year of birth cannot be in the future')
    age = 2024 - year_born
    return age

try:
    age = calculate_age(2034)
    print(age)
except ValueError as e:
    print(e)
```

4. Finally biasanya digunakan untuk cleanup resources.

```
try:
    file = open("data.txt", "r")
    data = file.read()
    num = int(data)
    print(num)
except OSError as err:
    print("OS error: {0}".format(err))
except:
    print("Unexpected error:", sys.exc_info()[0])
else:
```

```
print("No exceptions raised.")
finally:
    file.close()
print("File closed")
```

## 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.

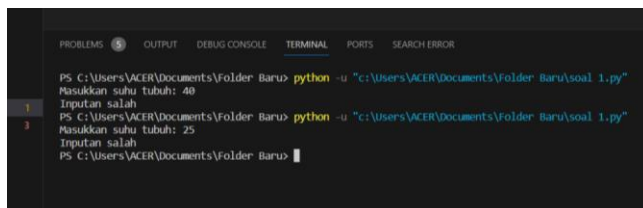
### SOAL 1

#### Contoh 3.1

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

try:
    suhu = int(tulis)
    if suhu >= 38 and suhu <= 60:
        print("Anda demam")
    elif suhu >= 35 and suhu <= 37:
        print("Anda tidak demam")

except:
    print("Inputan salah")
```



```
PS C:\Users\VACER\Documents\Folder Baru> python -u "c:\Users\VACER\Documents\Folder Baru\soal 1.py"
Masukkan suhu tubuh: 40
Anda demam
PS C:\Users\VACER\Documents\Folder Baru> python -u "c:\Users\VACER\Documents\Folder Baru\soal 1.py"
Masukkan suhu tubuh: 25
Anda tidak demam
PS C:\Users\VACER\Documents\Folder Baru> python -u "c:\Users\VACER\Documents\Folder Baru\soal 1.py"
Masukkan suhu tubuh: Inputan salah
Inputan salah
PS C:\Users\VACER\Documents\Folder Baru>
```

penjelasannya

masukkan input dalam masukkan suhu tubuh, jika suhu lebih dari 38 dan suhu kurang dari 60 maka itu akan termasuk dalam kategori demam, namun jika memasukkan bilangan suhu lebih dari 35 dan suhu kurang dari 37 maka anda tidak demam.

### contoh 3.2

```
contoh1 = input("Masukkan suatu bilangan: ")

try:
    bilangan = int(tulis1)
    if bilangan > 0:
        print("Positif")
    elif bilangan < 0:
        print("Negatif")
    elif bilangan == 0:
        print("Nol")
except:
    print("salah woiii")
```

```
inputan soal
PS C:\Users\ACER\Documents\Folder Baru> python -u "c:\Users\ACER\Documents\Folder Baru\soal 1.py"
Masukkan suatu bilangan: 10
Positif
PS C:\Users\ACER\Documents\Folder Baru> python -u "c:\Users\ACER\Documents\Folder Baru\soal 1.py"
Masukkan suatu bilangan: -3
Negatif
PS C:\Users\ACER\Documents\Folder Baru> python -u "c:\Users\ACER\Documents\Folder Baru\soal 1.py"
Masukkan suatu bilangan: 0
Nol
PS C:\Users\ACER\Documents\Folder Baru> []
```

### penjelasannya

masukkan input suatu bilangan lalu masukkan bilangan ke terminal jika bilangan tersebut lebih dari 0 maka termasuk bilangan positif sedangkan jika bilangan kurang dari 0 maka termasuk bilangan negatif namun jika bilangan tersebut sama-sama bernilai 0 maka akan tertulis "Nol".

### contoh 3.3

```
tulis_a = input("Masukkan bilangan pertama: ")
tulis_b = input("Masukkan bilangan kedua: ")
tulis_c = input("Masukkan bilangan ketiga: ")

try:
    a = int(tulis_a)
    b = int(tulis_b)
    c = int(tulis_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("SALAH!")
```

```
PS C:\Users\ACER\Documents\Folder Baru> python -u "c:\Users\ACER\Documents\Folder Baru\soal 1.py"
Masukkan suatu bilangan: 0
Hal
PS C:\Users\ACER\Documents\Folder Baru> python -u "c:\Users\ACER\Documents\Folder Baru\soal 1.py"
Masukkan bilangan pertama: 40
Masukkan bilangan kedua: 45
Masukkan bilangan ketiga: 50
Terbesar: 50
PS C:\Users\ACER\Documents\Folder Baru>
```

penjelasannya

masukkan inputan terlebih dahulu untuk bilangan a, b, c atau bilangan 1, 2, 3. Lalu masukkan bilangan dalam tipe integer jika a lebih dari b dan a lebih dari c maka a merupakan bilangan terbesar, jika b lebih dari a dan b lebih besar dari c maka bilangan b merupakan bilangan terbesar, jika bilangan c lebih dari a dan c lebih dari b maka bilangan c merupakan bilangan terbesar.

## SOAL 2

```
bilangan = float(input("Masukkan sebuah bilangan: "))

print(bilangan, "adalah bilangan", "Negatif" if bilangan < 0 else "Positif")
```

```
PS C:\Users\ACER\Documents\Folder Baru> python -u "c:\Users\ACER\Documents\Folder Baru\latian2.py"
Masukkan sebuah bilangan: 2
2.0 adalah bilangan Positif
PS C:\Users\ACER\Documents\Folder Baru> python -u "c:\Users\ACER\Documents\Folder Baru\latian2.py"
Masukkan sebuah bilangan: -7
-7.0 adalah bilangan Negatif
PS C:\Users\ACER\Documents\Folder Baru>
```

Penjelasannya

Masukkan input bilangan dan ubah ke tipe data float lalu akan memproses cetak nilai bilangan setelah itu gunakan ternary operator untuk percabangan positif dan negatif, jika kondisi bilangan tersebut kurang dari  $> 0$  maka akan mencetak bilangan positif atau FALSE namun sebaliknya jika  $bilangan < 0$  maka akan mencetak bilangan negatif atau TRUE.

## SOAL 3

```
try:
```



```

bulan = int(input("Masukkan bulan (1-12): "))

if bulan == 1 or bulan == 3 or bulan == 5 or bulan == 7 or bulan == 8 or bulan
== 10 or bulan == 12:
    hari = 31
elif bulan == 4 or bulan == 6 or bulan == 9 or bulan == 11:
    hari = 30
elif bulan == 2:
    hari = 29 # tahun kabisat
else:
    raise ValueError('Bulan tidak valid')

print(f"Jumlah hari: {hari}")

except ValueError as err:
    print("Bulan yang anda masukkan tidak valid. ", err)

```

```

PS C:\Users\ACER\Documents\Folder Baru> python -u "c:\Users\ACER\Documents\F
older Baru\latian2.py"
Masukkan bulan (1-12): 8
Jumlah hari: 31
PS C:\Users\ACER\Documents\Folder Baru> python -u "c:\Users\ACER\Documents\F
older Baru\latian2.py"
Masukkan bulan (1-12): 13
Bulan yang anda masukkan tidak valid. Bulan tidak valid
PS C:\Users\ACER\Documents\Folder Baru>

```

## Penjelasan

Masukkan input bulan lalu konversi input ke integer lalu tangani kasus masing-masing bulan dengan assign jumlah hari ke variabel hari. Jika input tidak sesuai 1-12 maka raise akan mengalami value eror, lalu tangkap bagian exceptionnya tampilkan pesan bhawa input bulan tidak valid. Lalu tampilkan jumlah hari jika harinya tidak terjadi exception.

## SOAL 4

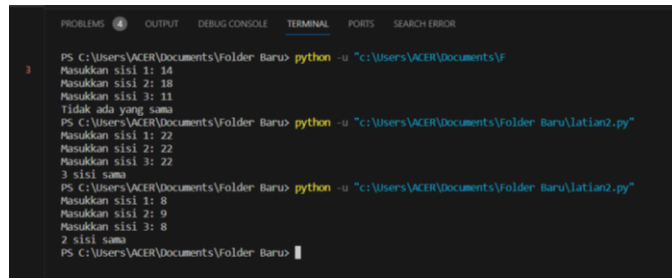
```

try:
    sisi1 = float(input("Masukkan sisi 1: "))
    sisi2 = float(input("Masukkan sisi 2: "))
    sisi3 = float(input("Masukkan sisi 3: "))

    if sisi1 == 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 ValueError:  
    print("Input harus berupa bilangan")
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR  
3 PS C:\Users\ACER\Documents\Folder Baru> python -u "c:\Users\ACER\Documents\F  
Masukkan sisi 1: 14  
Masukkan sisi 2: 18  
Masukkan sisi 3: 11  
Tidak ada yang sama  
PS C:\Users\ACER\Documents\Folder Baru> python -u "c:\Users\ACER\Documents\Folder Baru\latian2.py"  
Masukkan sisi 1: 22  
Masukkan sisi 2: 22  
Masukkan sisi 3: 22  
3 sisi sama  
PS C:\Users\ACER\Documents\Folder Baru> python -u "c:\Users\ACER\Documents\Folder Baru\latian2.py"  
Masukkan sisi 1: 8  
Masukkan sisi 2: 9  
Masukkan sisi 3: 8  
2 sisi sama  
PS C:\Users\ACER\Documents\Folder Baru> |
```

## Penjelasan

Masukkan input untuk ketiga sisi segitiga lalu konversi input ke tipe float, kemudian bandingkan nilai tiap sisi untuk mengecek apakah ada yang sama, tampilkan pesan sesuai hasil perbandingan, lalu tangani exception ValueError jika input bukan angka.