



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>71230985</b>
<b>Nama Lengkap</b>	<b>TOMAS BECKET</b>
<b>Minggu ke / Materi</b>	<b>03 / 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.

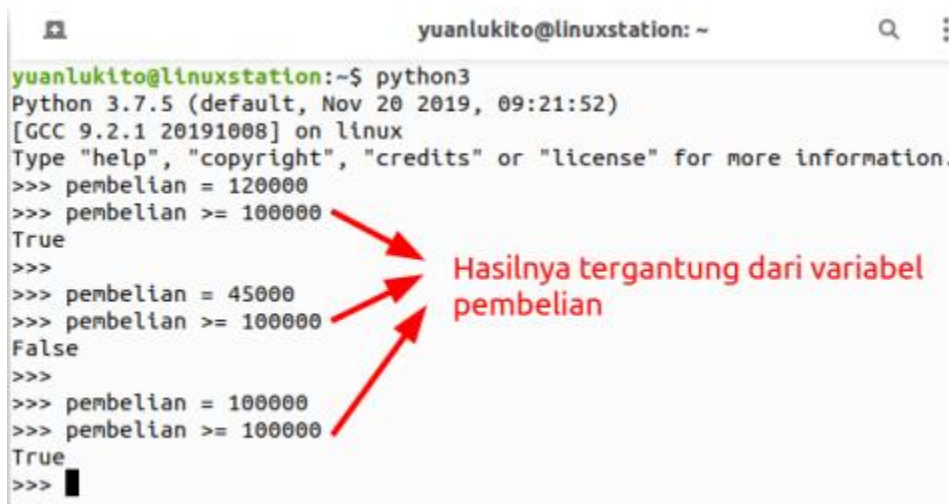
PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA  
YOGYAKARTA  
2024

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

### MATERI 1

#### BOOLEAN EXPRESION DAN LOGICAL OPERATOR

Pada materi minggu ke-3 dijelaskan tentang percabangan dimana pada fungsi percabangan terdapat boolean expression yaitu pernyataan benar atau salah contoh seperti gambar dibawah.



```
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

Source code buku 03 – Percabangan. Pdf

Yaitu source code diatas menunjukkan epression lebih besar sama dengan dan sama dengan jika dijelaskan maka.

Pembelian = 120000. Jika pembelian lebih besar dari sama dengan 100000 maka hasil out yang keluar yaitu True atau benar sesuai dengan fungsi yang dibuat, namun jika pembelian kurang dari sama dengan 100000 seperti contoh kedia yang dimana pembelian yaitu 45000 maka akan menghasilkan output False yaitu 45000 kurang dari 100000

Adapun beberapa operator yang digunakan dalam fungsi percabangan.

x == y	Apakah x sama dengan y?
x != y	Apakah x tidak sama dengan y?
x > y	Apakah x lebih besar dari y?
x >= y	Apakah x lebih besar atau sama dengan y?
x < y	Apakah x lebih kecil dari y?
x <= y	Apakah x lebih kecil atau sama dengan y?
x is y	Apakah x sama dengan y?
x is not y	Apakah x tidak sama dengan y?

Misal pada opertator peratam x == y artinya apakah x sama dengan y. jika x sama dengan y maka pernyataan tersebut True. Kemudian x != y artinya apakah nilai x bukan sama dengan y. jika nilai x benar bukan sama dengan nilai y maka pernyataan tersebut True

Sehingga dari semua operator yang ditunjukkan pada gambar diatas memiliki statement untuk menghasilkan sebuah jawaban boolean untuk menentukan apakah pernyataan pada sebuah source code True atau False

## MATERI 2

Ada pun juga bentuk-bentuk kode program dalam percabangan sebagai contoh

```
if <kondisi>
    <lakukan ini>
    <lakukan ini>
```

*Source code Vscode Salinan buku 03 – Percabangan.pdf*

Dengan kata lain bentuk tersebut yang biasanya digunakan dalam pemrograman jika ingin membuat suatu kondisi yaitu menggunakan if else atau elif.

Contoh seperti source code dibawah ini :

```
nilai_akhir = int(input("Input Nilai Anda = "))

if nilai_akhir > 70:
    print("Anda lulus dan dapat Sertifikasi")
elif nilai_akhir > 60 and nilai_akhir <= 70:
    print("Anda lulus dan Tidak dapat sertifikat")
elif nilai_akhir > 50 and nilai_akhir <= 60:
    print("Anda tidak lulus, sialhkan remedial")
elif nilai_akhir > 40 and nilai_akhir <= 50:
    print("Silahkan mengulang tahun depan")
else:
    print("Anda tidak lulus")
```

*Source code Latihan Vscode*

berikuta lah kondisi yang digunakan untuk memulai perintah yang dilanjutkan

```
if nilai_akhir > 70:
```

*Source code Latihan Vscode*

yaitu jika nilai akhir lebih dari 70 maka lakukan

```
    print("Anda lulus dan dapat Sertifikasi")
```

*Source code Latihan Vscode*

sehingga jika kondisi tersebut benar maka perintah yang akan dijalankan yaitu “Anda lulus dan dapat sertifikat”

Penggunaan elif lah digunakan jika pernyataan pertama salah atau False, jika kondisi yang dimiliki kurang dari sama dengan 70 dan nilai akhirnya lebih dari 60 maka akan melakukan perintah yang sesuai dengan kondisi yang dimiliki.

```
elif nilai_akhir > 60 and nilai_akhir <= 70:
    print("Anda lulus dan Tidak dapat sertifikat")
elif nilai_akhir > 50 and nilai_akhir <= 60:
    print("Anda tidak lulus, sialhkan remedial")
elif nilai_akhir > 40 and nilai_akhir <= 50:
    print("Silahkan mengulang tahun depan")
else:
    print("Anda tidak lulus")
```

*Source code Latihan Vscode*

Ada juga bentuk-bentuk percabangan lainnya seperti ternary operator

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

*Source code Buku 03 - Percabngan*

Pada bentuk ternary operator, kode yang ditulis terlihat pendek dan simple untuk mengurangi line program yang ada, berbeda dengan bentuk program biasa seperti code dibawah ini.

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

*Source code Buku 03 - Percabngan*

### MATERI 3

Penanganan kesalahan Input. Bagaimana jika kondisi yang dimiliki tidak sesuai dengan fungsi yang dimiliki, apakah mungkin kode tersebut tetap berjalan atau error? Disini ditunjukan penanganan kesalahan input menggunakan exception handling yaitu menggunakan try except. Contohnya seperti source code dibawah tersebut.

```
try:
    suhu = float(input("Masukkan suhu tubuh: "))
    if suhu >= 38:
        print("Anda demam")
    else:
        print("Anda tidak demam")
except:
    print("Kesalahan input")
```

*Source code Latihan Vscode*

Dicode ini diberitahu untuk memasukan input atau data untuk mengisi suatu variabel “suhu” sehingga bisa menjalankan perintah dibawah. Sehingga jika mengisi suhu tubuh dengan benar misal suhu tubuh yang dimiliki sekarang yaitu 35 derajat maka akan memunculkan ouput “anda tidak demam”

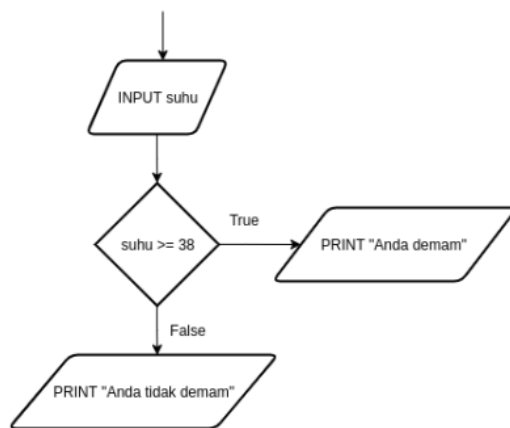
```
Masukkan suhu tubuh: 35
Anda tidak demam
```

*Hasil Ouput Latihan Vscode*

```
Masukkan suhu tubuh: Test
Kesalahan input
```

*Hasil Ouput Latihan Vscode ketika bukan menginput angka atau tidak sesuai*

Sehingga jika ditunjukan menggunakan flowchart maka akan dijeaskan seperti ini



*Source screenshot Flowchart pada buku 03 - Percabangan*

## MATERI 4

Adapun contoh lain dengan menggunakan fungsi percabangan pada Python contoh seperti source dibawah.

```
inputan = input("Masukan usia anda: ")

try:
    usia = int(inputan)
    if usia <= 5 and usia >=1:
        print("Balita")
    elif usia >= 6 and usia <= 11:
        print("Kanak-kanak")
    elif usia >=12 and usia <= 25:
        print("Remaja")
```

```

    elif usia >= 26 and usia <= 45:
        print("Dewasa")
    elif usia > 45:
        print("Lansia")
except:
    print("Anda salah memasukan input")

```

*Source code Latihan Vscod*

Sama hal nya dengan contoh percabangan lain, hanya saja pada source code ini memerintahkan program untuk menentukan umur kita pada sekarang ini.

```

inputan = input("Masukan usia anda: ")

```

*Source code Latihan Vscod*

Dengan menggunakan input pada statement pertama digunakan untuk mendapatkan sebuah nilai untuk dijalankan pada program yaitu usia.

```

usia = int(inputan)
    if usia <= 5 and usia >=1:
        print("Balita")
    elif usia >= 6 and usia <= 11:
        print("Kanak-kanak")
    elif usia >=12 and usia <= 25:
        print("Remaja")
    elif usia >= 26 and usia <= 45:
        print("Dewasa")
    elif usia > 45:
        print("Lansia")

```

*Source code Latihan Vscod*

Kode diatas hanya akan bekerja jika hasil inputan yang dimasukan sesuai dengan salah satu atau menghasilkan true pada salah satu dari 5 percabangan yang ada. Jika usia yang diinputkan merupakan angka 1, 2, 3, 4, 5 maka akan menghasilkan output ("Balita") dst

Namun jika hasil input yang dimasukan tidak benar seperti mengisi String pada kolom input maka

```

except:
    print("Anda salah memasukan input")

```

*Source code Latihan Vscod*

Kode ini lah yang akan berjalan untuk mengeksekusi inputan yang salah.

## BAGIAN 2: LATIHAN MANDIRI (60%)

### SOAL 1

#### 1.1 SOURCE CODE

```
# LATIHAN 3.1 CONTOH 3.1
try:
    suhu = float(input("Masukkan suhu tubuh: "))
    if suhu >= 38:
        print("Anda demam")
    else:
        print("Anda tidak demam")

except:
    print("Kesalahan input")

# LATIHAN 3.1 CONTOH 3.2
try:
    bilangan = float(input("Masukkan suatu bilangan: "))
    if bilangan > 0:
        print("Positif")
    elif bilangan < 0:
        print("Negatif")
    elif bilangan == 0:
        print("Nol")

except:
    print("Kesalahan input")

# LATIHAN 3.1 CONTOH 3.3
try:
    a = int(input("Masukkan bilangan pertama: "))
    b = int(input("Masukkan bilangan kedua: "))
    c = int(input("Masukkan bilangan ketiga: "))

    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("Kesalah Input")
```

## 1.2 PENJELASAN

Sama seperti penjelasan yang dirangkum pada catatan diatas, dijelaskan bahwa penggunaan try except digunakan untuk memunculkan kesalahan input yang diberikan oleh pengguna contoh 1:

```
try:
    suhu = float(input("Masukkan suhu tubuh: "))
    if suhu >= 38:
        print("Anda demam")
    else:
        print("Anda tidak demam")
```

try : artinya coba, atau mengetest hasil input yang diberikan, jika hasil input yang diberikan benar dan sesuai maka langkah selanjutnya yaitu dengan melakukan pengeksekusian dengan menentukan hasil inputan sesuai dengan fungsi yang dimiliki atau mungkin fungsi tersebut tidak benar.

### Arti code:

Jika suhu lebih besar sama dengan 38 maka print “anda demam”, else (atau) print “Anda tidak demam” jika hasil input dibawah 38.

```
except:
    print("Kesalahan input")
```

disinilah except digunakan jika value input tidak sesuai seperti hasil output screenshot dibawah.

Dan penjelasan ini berlaku pada semua Contoh 3.2 dan 3.3

## 1.3 HASIL OUTPUT

- INPUT BENAR

```
Masukkan suhu tubuh: 35
Anda tidak demam
Masukkan suatu bilangan: 5
Positif
Masukkan bilangan pertama: 3
Masukkan bilangan kedua: 4
Masukkan bilangan ketiga: 5
Terbesar: 5
```

- INPUT SALAH

```
Masukkan suhu tubuh: Test
Kesalahan input
Masukkan suatu bilangan: Test
Kesalahan input
Masukkan bilangan pertama: Test
Kesalah Input
```



## SOAL 2

### 2.1 SOURCE CODE

```
# LATIHAN 3.2
bilangan = float(input("Masukkan suatu bilangan: "))
print("Positif") if bilangan > 0 else print("Negatif") if bilangan < 0 else
print("Nol")
```

### 2.2 PENJELASAN

Disoal dijelaskan untuk mengubah contoh code 3.2 menjadi ternary code, dan kode diatas merupakan bentuk berubahnya code yang awalnya seperti code dibawah.

```
bilangan = float(input("Masukkan suatu bilangan: "))
if bilangan > 0:
    print("Positif")
elif bilangan < 0:
    print("Negatif")
elif bilangan == 0:
    print("Nol")
```

menjadi hanya menggunakan 3 line saja pada ternary code. Kedua code tersebut memiliki arti yang sama yaitu menentukan nilai positif dan negatif pada value yang di inputkan dan Nol jika angka inputan nol.

### 2.3 HASIL OUTPUT

```
Masukkan suatu bilangan: 1
Positif
```

## SOAL 3

### 3.1 SOURCE CODE

```
# LATIHAN 3.3
def jumlah_hari():
    try:
        bulan = int(input("Masukkan bulan (1-12): "))
        if bulan < 1 or bulan > 12:
            print("bulan yang diinputkan tidak valid.")
        else:
            if bulan in [1, 3, 5, 7, 8, 10, 12]:
                print("Jumlah hari: 31")
            elif bulan in [4, 6, 9, 11]:
                print("Jumlah hari: 30")
            else:
                print("Jumlah hari: 29 (Tahun kabisat)")
    except:
        print("bulan yang diinputkan tidak valid.")

jumlah_hari()
```

### 3.2 PENJELASAN

```
def jumlah_hari():
    try:
        bulan = int(input("Masukkan bulan (1-12): "))
        if bulan < 1 or bulan > 12:
            print("bulan yang diinputkan tidak valid.")
```

dimulai dengan fungsi def untuk mendefinisikan sebuah fungsi, dimana line 3 dengan mengambil sebuah input angka bulan dari 1 sampai dengan 12 pada tahun 2020 yang dimana merupakan Tahun kabisat. Dan dilanjutkan dengan code :

```
        if bulan < 1 or bulan > 12:
            print("bulan yang diinputkan tidak valid.")
```

dengan statement jika bulan kurang 1 atau bulan lebih besar dari 12 maka akan menghasilkan output "Bulan yang diinputkan tidak valid." Kode awalan ini ditujukan untuk mendapatkan hasil kesalahan input jika user memberikan input yang tidak sesuai dengan apa yang diminta.

```
    else:
        if bulan in [1, 3, 5, 7, 8, 10, 12]:
            print("Jumlah hari: 31")
        elif bulan in [4, 6, 9, 11]:
            print("Jumlah hari: 30")
        else:
            print("Jumlah hari: 29 (Tahun kabisat)")
```

dilanjutkan dengan fungsi lain, dimana jika bulan yang dimasukan sesuai pada data list atau array yang bernilai 1, 3, 5, 7, 8, 10, dan 12 maka akan menunjukan "Jumlah Hari : 31", dan fungsi lainnya jika fungsi pertama bukan akan 1, 3, 5, 7, 8, 10 maka akan dilanjutkan ke fungsi kedua yaitu data array dengan angka 4, 6, 9, 11 yang akan memunculkan "jumlah hari : 30". Fungsi lainnya yaitu jika angka lain selain kedua data tersebut yaitu angka 2 akan memunculkan "Jumlah hari : 29 (Tahun kabisat)"

```
except:
```

```
    print("bulan yang diinputkan tidak valid.")
```

dan except inilah yang akan memunculkan hasil input yang salah juga jika tidak angka yang diberikan, maka string juga bisa terdetect dengan memberikan except pada akhir code.

```
jumlah_hari()
```

dan dilanjutkan fungsi akhir untuk memanggil value jumlah\_hari()

### 3. HASIL OUTPUT

- BULAN BIASA

```
Masukkan bulan (1-12): 7
Jumlah hari: 31
```

- BULAN KABISAT

```
Masukkan bulan (1-12): 2
Jumlah hari: 29 (Tahun kabisat)
```

- INPUT SALAH

```
Masukkan bulan (1-12): Test
bulan yang diinputkan tidak valid.
```

## SOAL 4

### 4.1 SOURCE CODE

```
# LATIHAN 3.4
def sisi_segitiga():
    try:
        a = float(input("Masukan Sisi 1: "))
        b = float(input("Masukan Sisi 2: "))
        c = float(input("Masukan Sisi 3: "))
        if a == b == c:
            print("3 sisi sama")
        elif (a == b and c) or (b == c and a) or (c == a and b):
            print("2 sisi sama")
        elif (a != b and c) or (b != c and a) or (c != a and b):
            print("Tidak ada yang sama")

    except:
        print("Input tidak valid.")

sisi_segitiga()
```

### 4.2 PENJELASAN

Sama seperti Penjelasan Soal 3

```
def sisi_segitiga():
```

digunakan awalan untuk mendefinisikan sebuah fungsi kemudian dilanjut dengan try dan except untuk memberikan kesalahan input. Dan fungsi percabangan

```
    try:
        a = float(input("Masukan Sisi 1: "))
        b = float(input("Masukan Sisi 2: "))
        c = float(input("Masukan Sisi 3: "))
        if a == b == c:
            print("3 sisi sama")
        elif (a == b and c) or (b == c and a) or (c == a and b):
            print("2 sisi sama")
        elif (a != b and c) or (b != c and a) or (c != a and b):
            print("Tidak ada yang sama")
```

variabel a, b, dan c digunakan untuk mempermudah variabel fungsi, menggunakan float jika ada penggunaan input berangka desimal.

```
        if a == b == c:
            print("3 sisi sama")
```

fungsi percabangan pertama dijelaskan jika a sama dengan b dan sama dengan c maka lakukan print ("3 sisi sama") seperti hasil output pada (3 sisi sama).

```
elif (a == b and c) or (b == c and a) or (c == a and b):  
    print("2 sisi sama")
```

fungsi kedua digunakan untuk menentukan 2 sisi yang sama dimana

jika a sama dengan b dan c atau

b sama dengan c dan a atau

c sama dengan a dan b

maka lakukan print("2 sisi sama"), fungsi ini sama seperti logika matematik dimana fungsi dan dan atau memiliki tingkatan sendiri misal terdapat kata DAN maka didahulukan terlebih dahulu dan kemudian dilanjutkan dengan ATAU.

```
elif (a != b and c) or (b != c and a) or (c != a and b):  
    print("Tidak ada yang sama")
```

dan fungsi ketiga akan muncul jika nilai a, b, dan c tidak memiliki angka yang sama dimana menggunakan operator != dimana jika a bukan sama dengan b dan c, atau b bukan sama dengan c dan a, atau c bukan sama dengan a dan b, maka akan memunculkan "Tidak ada yang sama" pada hasil outputan.

#### 4.3 HASIL OUTPUT

##### TIDAK SAMA

```
Masukan Sisi 1: 1  
Masukan Sisi 2: 2  
Masukan Sisi 3: 3  
Tidak ada yang sama
```

##### 2 SAMA

```
Masukan Sisi 1: 1  
Masukan Sisi 2: 1  
Masukan Sisi 3: 2  
2 sisi sama
```

##### 3 SAMA

```
Masukan Sisi 1: 1  
Masukan Sisi 2: 1  
Masukan Sisi 3: 1  
3 sisi sama
```

##### INPUT SALAH

```
Masukan Sisi 1: Test  
Input tidak valid.
```