



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71200609
Nama Lengkap	Airell Aristo Subagia
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

Struktur Kontrol Percabangan

1. Boolean Expression and Logical Operator

Apa itu Boolean Expression ? merupakan sebuah ekspresi matematika yang menghasilkan nilai benar (True) atau salah (False). Boolean Expression terdiri dari beberapa operator logika, yaitu : or, and, not. Ada juga beberapa operator lain yang digunakan seperti pada tabel dibawah ini :

Expression	Simbol	Arti
$X > Y$	$>$	X lebih besar dari Y
$X < Y$	$<$	X kurang dari Y
$X == Y$	$==$	X sama dengan Y
$X != Y$	$!=$	X tidak sama dengan Y

Biasanya pada Boolean Expression digunakan pada operasi logika if-else ataupun looping.

Contoh penggunaan :

```
x = 5
y = 4

print(x>y)
print(x<y)
print(x!=y)
print(x==y)
```

Output :

```
True
False
True
False
```

Penjelasan Singkat :

Pada fungsi print yang pertama disana diperlihatkan ekspresi logika, yaitu $x > y$. Yang dimana berarti dilihat dari variable yang telah dibuat diatasnya yaitu $x = 5$ dan $y = 4$, berarti x itu lebih besar dari y (5 lebih besar dari 4) dan akan menghasilkan nilai true.

2. Bentuk-Bentuk Percabangan

Pada bahasa pemrograman python, ada 4 bentuk percabangan yang umum digunakan, seperti :

1. Percabangan If

```
angka = 2  
  
if angka % 2 == 0 :  
    print("Genap")
```

Ini adalah bentuk pertama yang dimana, if akan mengeluarkan hasilnya jika conditional yang didapat bernilai True.

Outputnya :

Genap

Didapatkan genap karena angka 2 itu habis dibagi dengan 2 dan tidak bersisa nilai apapun.

2. Percabangan If-Else

```
angka = 2  
  
if angka % 2 == 0 :  
    print("Genap")  
else :  
    print("Ganjil")
```

Ini adalah bentuk kedua yang dimana, terdapat dua conditional yang bisa dihasilkan yaitu. Jika if conditional menghasilkan nilai True, maka program akan mengeluarkan output berupa "Genap". Sebaliknya jika conditional menghasilkan nilai False, maka program akan mengeluarkan output berupa "Ganjil"

3. Percabangan If-Elif-Else

```
huruf = "A"

if huruf == "A" :
    print("Ini Huruf A")
elif huruf == "B":
    print("Ini Huruf B")
else :
    print("Ini bukan huruf A ataupun B")
```

Ini adalah bentuk ketiga yang dimana, digunakan untuk mengevaluasi beberapa kondisi secara berurutan.

4. Percabangan Bersarang (Nested If)

```
item = "apel"
itemDibeli = 3

if item == "apel" :
    if itemDibeli > 2 :
        print("Selamat Anda Mendapatkan diskon sebesar 15%")
else :
    print("Belum ada diskon untuk saat ini")
```

Dan ini adalah bentuk yang terakhir dimana, pada bentuk ini dimungkinkan untuk menguji beberapa kondisi secara bertingkat. Seperti pada contoh diatas, if pertama akan dieksekusi jika item sama dengan apel kemudian akan dilakukan pengecekan ulang oleh if kedua jika itemDibeli lebih dari 2.

3. Error Handling Dalam Kesalahan Input Pengguna

Merupakan proses yang dilakukan untuk mendeteksi, menangani dan memulihkan kesalahan atau exception yang dapat terjadi selama program dieksekusi. Salah satu hal yang dapat dilakukan adalah menangani kesalahan input oleh User, yang dimana jika hal ini terjadi maka program tidak akan berjalan sebagaimana mestinya atau error.

Contoh tanpa error handling :

```
itemDibeli = int(input("Masukkan Jumlah Item Yang Anda beli : "))

if itemDibeli >= 1 :
    if itemDibeli >= 1 and itemDibeli < 5:
```

```

        print("Selamat Anda Mendapatkan diskon sebesar 15%")
    elif itemDibeli >= 5 :
        print("Selamat Anda Mendapatkan diskon sebesar 30%")
    else :
        print("Selamat Anda Mendapatkan diskon sebesar 99%")
else :
    print("Anda Tidak Membeli Apapun")

```

Pada contoh diatas dapat dilihat bahwa input yang diterima harus berupa integer. Jika user memasukkan input misalnya berupa string ataupun float maka akan muncul error exception, seperti pada gambar dibawah ini :

- Jika berupa String

```

itemDibeli = int(input("Masukkan Jumlah Item Yang Anda beli : "))
ValueError: invalid literal for int() with base 10: 'satu'

```

- Jika berupa Float

```

itemDibeli = int(input("Masukkan Jumlah Item Yang Anda beli : "))
ValueError: invalid literal for int() with base 10: '1.0'

```

Maka dari itu untuk menangani situasi diatas maka akan dilakukan :

```

try :
    itemDibeli = int(input("Masukkan Jumlah Item Yang Anda beli : "))

    if itemDibeli >= 1 :
        if itemDibeli >= 1 and itemDibeli < 5:
            print("Selamat Anda Mendapatkan diskon sebesar 15%")
        elif itemDibeli >= 5 :
            print("Selamat Anda Mendapatkan diskon sebesar 30%")
        else :
            print("Selamat Anda Mendapatkan diskon sebesar 99%")
    else :
        print("Anda Tidak Membeli Apapun")
except :
    print("Input Invalid")

```

setelah ditambahkan try and except, maka jika terdapat inputan yang bermasalah tidak mengeluarkan error output pada terminal melainkan mengeluarkan pesan yang sudah di set pada blok except.

Contoh outputnya :

```
Masukkan Jumlah Item Yang Anda beli : satu  
Input Invalid
```

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

#Latihan 3.1

```
try :
    suhu = int(input("Masukkan suhu tubuh: "))
    if suhu >= 38:
        print("Anda demam")
    else :
        print("Anda Tidak Demam")
except :
    print("Invalid Input")
```

#Latihan 3.2

```
try :
    bilangan = int(input("Masukkan suatu bilangan: "))
except:
    print("Invalid Input")
else :
    if bilangan > 0:
        print("Positif")
    elif bilangan < 0:
        print("Negatif")
    elif bilangan == 0:
        print("Nol")
```

#Latihan 3.3

input a, b dan c

```
try :
    a = int(input("Masukkan bilangan pertama: "))
    b = int(input("Masukkan bilangan kedua: "))
    c = int(input("Masukkan bilangan ketiga: "))
except :
    print("Invalid Input")
else :
    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)
```

Outputnya :

- Latihan 3.1

```
Masukkan suhu tubuh: satu
Invalid Input
```

- Latihan 3.2

```
Masukkan suatu bilangan: satu
Invalid Input
```

- Latihan 3.3

```
Masukkan bilangan pertama: satu
Invalid Input
```

Penjelasan :

- Latihan 3.1

Pada latihan 3.1 untuk menangani error input user dapat digunakan blok **try except**. Blok ini berfungsi untuk menangani jika user menginputkan value yang bukan berupa integer maka akan dijalankan hasil yang ada di blok except.

- Latihan 3.2 dan Latihan 3.3

Pada latihan 3.2 dan 3.3, saya gunakan fungsi lain untuk menangani error input user yaitu **try except else**. Blok **try except** masih sama akan tetapi disini ditambahkan blok **else**, yang akan berjalan jika blok **try** tidak mengeluarkan exception.

SOAL 2

```
bilangan = int(input("Masukkan Sebuah Bilangan: "))
cek = (
    "Positif" if bilangan >= 1 else
    ("No1" if bilangan == 0 else
    ("Negatif" if bilangan < 1 else "Invalid Input"))
print(cek)
```

Outputnya :

```
Masukkan Sebuah Bilangan: 1
Positif
```

```
Masukkan Sebuah Bilangan: 0
No1
```



```
Masukkan Sebuah Bilangan: -1
Negatif
```

Penjelasan :

1. Sama dengan pembuatan if statement pada umumnya, akan tetapi untuk melakukan ternary operator akan dilakukan didalam sebuah variable. Jadi jika ingin mendapatkan hasilnya harus di print.
2. Kemudian untuk membuat ternary menjadi bentuk **if-elif-else**. Akan dilakukan syntax yang dimulai dengan outputnya terlebih dahulu sebagai contoh :

```
"Positif" if bilangan >= 1 else
```

Kemudian setelah itu fungsi logikanya, setelah itu dibuat else untuk masuk ke elif berikutnya dengan cara

```
("No!" if bilangan == 0 else
("Negatif" if bilangan < 1 else "Invalid Input"))
```

3. Setelah itu untuk mendapatkan hasilnya akan digunakan fungsi **print(cek)**.

SOAL 3

```
try :
    bulan = int(input("Masukkan Bulan (1-12) : "))
    jumlahHari = 0
    check = 1

    if bulan == 1 :
        jumlahHari += 31
    elif bulan == 2 :
        jumlahHari += 29
    elif bulan == 3 :
        jumlahHari += 31
    elif bulan == 4 :
        jumlahHari += 30
    elif bulan == 5 :
        jumlahHari += 31
    elif bulan == 6 :
        jumlahHari += 30
    elif bulan == 7 :
        jumlahHari += 31
    elif bulan == 8 :
        jumlahHari += 31
    elif bulan == 9 :
        jumlahHari += 30
    elif bulan == 10 :
```

```

        jumlahHari += 31
    elif bulan == 11 :
        jumlahHari += 30
    elif bulan == 12 :
        jumlahHari += 31
    else :
        check = 0

    if check != 0 :
        print("Jumlah Hari :", jumlahHari)
    else :
        print("Masukkan Bulan Dari Range 1-12 saja")

except :
    print("Invalid Input")

```

Output :

```

Masukkan Bulan (1-12) : 1
Jumlah Hari : 31

```

```

Masukkan Bulan (1-12) : cel
Invalid Input

```

```

Masukkan Bulan (1-12) : 0
Masukkan Bulan Dari Range 1-12 saja

```

Penjelasan :

1. Pertama tama, dibuat dulu error handlingnya berupa **try except** untuk menangani error input oleh user.
2. Setelah itu dibuat variable untuk mengambil input user yang dinamakan **bulan**.
3. Kemudian dibuat 2 variable tambahan yang bernama **jumlahHari**, yang dimana digunakan untuk menyimpan jumlah hari dari bulan nantinya dan **check** yang digunakan untuk check input user apakah rangnya berada di range 1-12.
4. Setelah itu dibuat **if-elif-else** dengan isinya bulan 1 sampai 3 dengan kode yang akan dieksekusi yaitu berisi hari dari bulan-bulan tersebut.
5. Kemudian **if-else** statement yang terakhir digunakan untuk checking apakah input user berada di range 1-12 atau tidak jika iya maka print jumlah harinya jika tidak print error messagenya.

SOAL 4

```
try :
    sisi1 = int(input("Masukkan Sisi 1 : "))
    sisi2 = int(input("Masukkan Sisi 2 : "))
    sisi3 = int(input("Masukkan Sisi 3 : "))

# 1 1 1
    if sisi1 == sisi2 and sisi1 == sisi3 :
        print("3 sisi sama")
    elif sisi1 == sisi2 and sisi1 != sisi3 :
        print("2 sisi sama")
    elif sisi1 == sisi3 and sisi1 != sisi2 :
        print("2 sisi sama")
    elif sisi2 == sisi3 and sisi2 != sisi1 :
        print("2 sisi sama")
    else :
        print("Tidak ada sisi yang sama")
except :
    print("Inputan tidak valid ")
```

Outputnya :

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

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

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

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

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

Penjelasan :

1. Pertama tama, dibuat dulu error handlingnya berupa **try except** untuk menangani error input oleh user.
2. Kemudian dibuat 3 variable yang digunakan untuk menyimpan ketiga sisi-sisinya dengan nama **sisi1,sisi2,sisi3**.
3. Setelah itu, dibuat logikanya yang dimana :
 - Jika ingin mendapatkan ketiga sisi yang sama jadinya dibuat **sisi1** sama dengan **sisi2** dan **sisi1** sama dengan **sisi3**
 - Setelah itu untuk mendapatkan 2 sisi yang sama terdapat 3 kondisi yaitu **sisi1 sama dengan sisi2, sisi1 sama dengan sisi3, sisi2 sama dengan sisi3** .
 - Setelah itu untuk mendapatkan hasil tidak ada sisi yang sama cukup menggunakan else saja. Hal ini dikarenakan semua kondisi sudah terpenuhi.

Link Github :