

Dosen Pengampu: Sevi Nurafni



PRAKTIKUM 6

Seleksi Kondisi dalam Python

Asisten Laboratorium:
Najlia Intani
Ryan Fadhilah Faizal Hakim



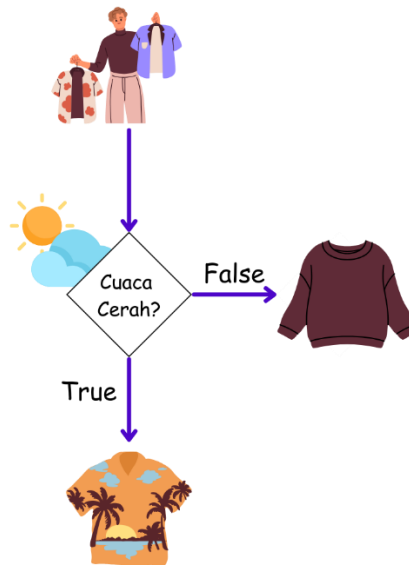
*"Modul ini diperuntukkan bagi Mata Kuliah
Algoritma Program Studi Agribisnis"*

A. PENGANTAR PERCABANGAN

Percabangan atau seleksi kondisi adalah pernyataan dalam Python yang menyediakan pilihan untuk alur kontrol berdasarkan suatu kondisi tertentu. Sederhananya, percabangan memungkinkan program untuk memilih jalur eksekusi yang berbeda tergantung pada apakah suatu kondisi bernilai benar (True) atau salah (False).

Bayangkan sebuah persimpangan jalan. Kendaraan hanya bisa melewati satu jalur pada satu waktu. Pilihan jalur mana yang diambil tergantung pada kondisi, seperti lampu lalu lintas atau petunjuk arah. Percabangan dalam pemrograman bekerja dalam prinsip yang sama. Program mengevaluasi sebuah kondisi, dan berdasarkan hasil evaluasi tersebut, program akan mengeksekusi blok kode yang sesuai.

Perhatikan ilustrasi berikut:



Pada ilustrasi diatas, seseorang ingin memilih baju yang tepat berdasarkan kondisi apakah cuaca di luar cerah atau tidak. Apabila cuaca cerah, maka memilih menggunakan t-shirt. Apabila cuaca tidak cerah, maka memilih menggunakan sweater.

B. SELEKSI KONDISI IF

Pengambilan keputusan (kondisi if) digunakan untuk mengantisipasi kondisi yang terjadi saat jalannya program dan menentukan tindakan apa yang akan diambil sesuai dengan kondisi. Pada python ada beberapa statement/kondisi diantaranya adalah if, else dan elif Kondisi if digunakan untuk mengeksekusi kode jika kondisi bernilai benar True. Jika kondisi bernilai salah False maka statement/kondisi if tidak akan di-eksekusi.

Contoh 1

#Kondisi if adalah kondisi yang akan dieksekusi oleh program jika bernilai benar atau TRUE

```
nilai = 9
#jika kondisi benar/TRUE maka program akan mengeksekusi perintah dibawahnya
if(nilai > 7):
    print("Sembilan Lebih Besar Dari Angka Tujuh") # Kondisi Benar,
    Dieksekusi
#jika kondisi salah/FALSE maka program tidak akan mengeksekusi perintah
dibawahnya
if(nilai > 10):
    print("Sembilan Lebih Besar Dari Angka Sepuluh") # Kondisi Salah, Maka
    tidak tereksekusi
```

Contoh 2

```
grade = float(input("Masukkan Nilai Anda: "))

if grade == 100:
    print("perfect")

if grade == 90:
    print("ok")
    print("keep working hard!")
```

Contoh 3

```
grade = float(input("Masukkan Nilai Anda: "))

if grade == 100:
    print("perfect")

if grade == 90:
    print("ok")
    print("keep working hard!")
```

C. SELEKSI KONDISI IF-ELSE

Pengambilan keputusan (kondisi if else) tidak hanya digunakan untuk menentukan tindakan apa yang akan diambil sesuai dengan kondisi, tetapi juga digunakan untuk menentukan tindakan apa yang akan diambil/dijalankan jika kondisi tidak sesuai. Pada python ada beberapa statement/kondisi diantaranya adalah if, else dan elif. Kondisi if digunakan untuk mengeksekusi kode jika kondisi bernilai benar. Kondisi if else adalah kondisi dimana jika pernyataan benar True maka kode dalam if akan dieksekusi, tetapi jika bernilai salah False maka akan mengeksekusi kode di dalam else.

Contoh 4

```
# Kondisi if else adalah jika kondisi bernilai TRUE maka akan dieksekusi
pada if,
# tetapi jika bernilai FALSE maka akan dieksekusi kode pada else

nilai = 8
# Jika pernyataan pada if bernilai TRUE maka if akan dieksekusi,
# tetapi jika FALSE kode pada else yang akan dieksekusi.
if(nilai > 7):
    print("Selamat Anda Lulus")
else:
    print("Maaf Anda Tidak Lulus")
```

Contoh 5

```
str_input = input('Enter your grade: ')
grade = int(str_input)

if grade == 100:
    print("perfect")
else:
    print("keep it up")
```

Contoh 6

```
Lapar = input("Apakah zaki lapar? ")
if("lapar == ya"):
    print("Zaki makan")
else:
    print("Zaki tidak makan")
```

D. SELEKSI KONDISI IF-ELIF-ELSE

Pengambilan keputusan (kondisi if elif) merupakan lanjutan/percabangan logika dari "kondisi if". Dengan elif kita bisa membuat kode program yang akan menyeleksi beberapa kemungkinan yang bisa terjadi. Hampir sama dengan kondisi "else", bedanya kondisi "elif" bisa banyak dan tidak hanya satu.

Contoh 7

```
#Contoh penggunaan kondisi elif

hari_ini = input("Masukkan hari(gunakan huruf kecil): ")

if(hari_ini == "senin"):
    print("Saya akan kuliah")
elif(hari_ini == "selasa"):
    print("Saya akan kuliah")
elif(hari_ini == "rabu"):
    print("Saya akan kuliah")
elif(hari_ini == "kamis"):
    print("Saya akan kuliah")
elif(hari_ini == "jumat"):
    print("Saya akan kuliah")
elif(hari_ini == "sabtu"):
    print("Saya akan kuliah")
elif(hari_ini == "minggu"):
    print("Saya akan libur")
```

Contoh 8

```
str_input = input('Enter your grade: ')
grade = int(str_input)

if grade == 100:
    print("perfect")
elif grade >= 85:
    print("awesome")
elif grade >= 65:
    print("passed the exam")
else:
    print("below the passing grade")
```

E. SELEKSI KONDISI BERCABANG/NESTED

Seleksi kondisi bisa saja berada di dalam suatu blok seleksi kondisi. Teknik ini biasa disebut dengan seleksi kondisi bercabang atau bersarang.

Di Python, cara penerapannya cukup dengan menuliskan blok seleksi kondisi tersebut. Gunakan indentation yang lebih ke kanan untuk seleksi kondisi terdalam.

Contoh 9

```
str_input = input('Enter your grade: ')
grade = int(str_input)

if grade == 100:
    print("perfect")

elif grade >= 85:
    print("awesome")

elif grade >= 65:
    print("passed the exam")

    if grade <= 70:
        print("but you need to improve it!")
    else:
        print("with ok grade")

else:
    print("below the passing grade")
```

F. SELEKSI KONDISI OPERASI LOGIKA

Keyword and, or, dan not bisa digunakan dalam seleksi kondisi.

Contoh 10

```
grade = int(input('Enter your current grade: '))
prev_grade = int(input('Enter your previous grade: '))

if grade >= 90 and prev_grade >= 65:
    print("awesome")
if grade >= 90 and prev_grade < 65:
    print("awesome. you definitely working hard, right?")
elif grade >= 65:
    print("passed the exam")
else:
    print("below the passing grade")

if (grade >= 65 and not prev_grade >= 65) or (not grade >= 65 and
prev_grade >= 65):
    print("at least you passed one exam. good job!")
```

G. TERNARY OPERATOR

Ternary Operator adalah cara ringkas untuk menulis pernyataan `if...else` dalam satu baris kode (one-liner) di Python. Ini adalah cara yang efisien untuk menetapkan nilai ke variabel berdasarkan suatu kondisi.

Perhatikan perbedaan struktur IF-ELSE Statement dengan Ternary Operator berikut:

IF-ELSE Statement

```
if kondisi :  
    blok_kode_jika_benar  
else :  
    blok_kode_jika_salah
```

Ternary Operator

```
blok_kode_jika_benar if kondisi else blok_kode_jika_salah
```

Ternary operator digunakan saat ingin menetapkan nilai ke variabel berdasarkan kondisi sederhana, ternary operator membuat kode lebih ringkas dan mudah dibaca (dalam kasus yang tidak terlalu kompleks). Jika kondisi sangat kompleks atau bercabang, (seperti `if...elif...else` dengan banyak `elif`), menggunakan ternary operator bisa membuat kode sulit dibaca dan dipahami.

Contoh penggunaan Ternary Operator:

```
umur = 20  
  
status = "Dewasa" if umur >= 18 else "Anak-anak"  
  
print(status)
```


LATIHAN PRAKTIKUM 6

1. Minta user untuk memberikan sebuah bilangan dengan menggunakan perintah input. Jika bilangan tersebut habis dibagi 3 (tiga).
2. Minta user untuk memberikan namanya. Kemudian minta lagi ia untuk memasukkan jenis kelaminnya (pria,wanita). Jika ia memilih pria, cetak tulisan halo bro diikuti dengan namanya, jika ia memilih wanita, cetak tulisan halo sis diikuti dengan Namanya.
3. Buat program kondisi python bebas, dengan syarat input dari user dan memenuhi kondisi if-elif-else.