

Python Fundamental

Ch.6 : Control Structure

Instruktur : Ahmad Rio Adriansyah
Nurul Fikri Komputer

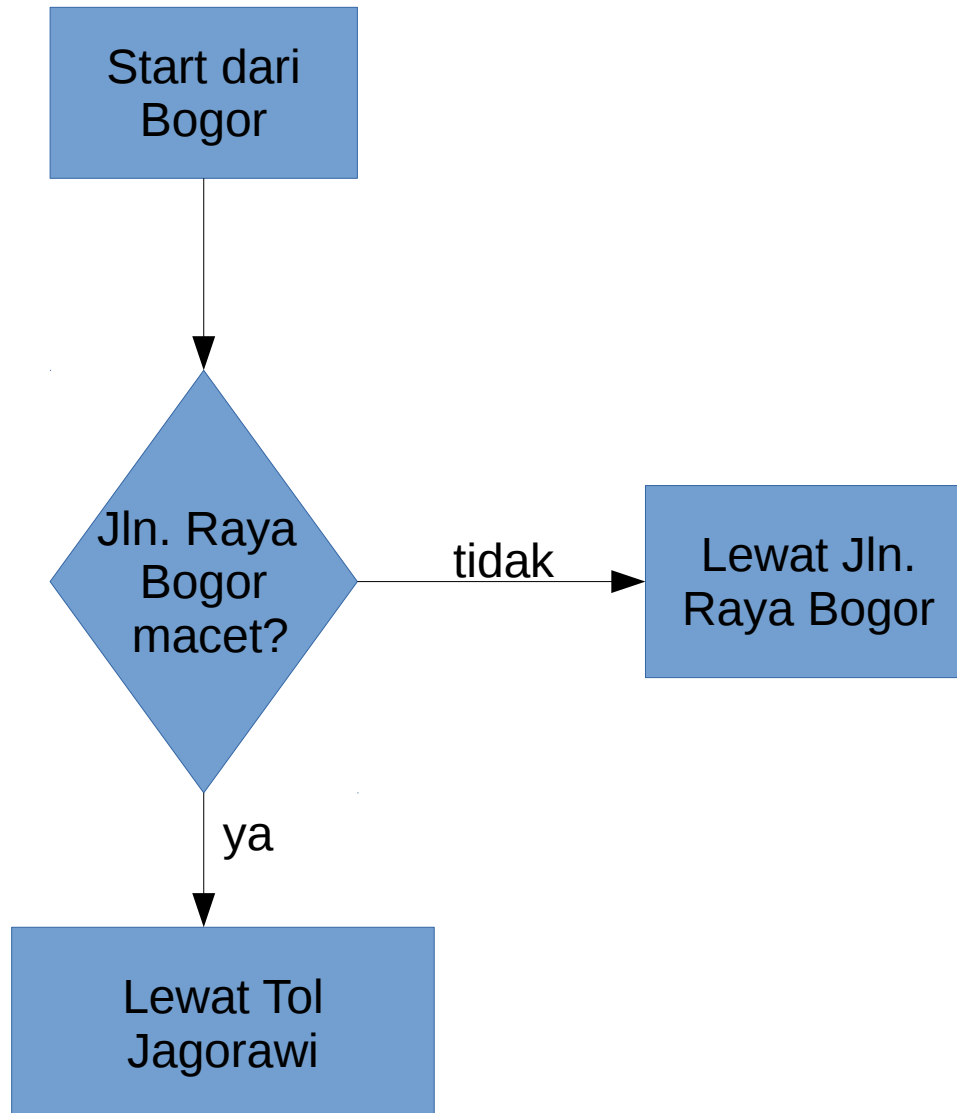
Struktur Kontrol

- Ini adalah bagian inti dari pemrograman.
Tentang bagaimana kita mengatur perilaku program yang kita buat
- Ada 2 hal yang akan dibahas :
 - Percabangan
 - Perulangan

Percabangan

- Pada saat kita ingin melakukan suatu hal yang berbeda dalam **kondisi** berbeda, kita menyusun struktur program kita dalam bentuk bercabang
- Contoh : Berangkat dari Bogor, mau ke Lenteng Agung. Lewat jalur mana tergantung kondisi jalan. Jika Jalan Raya Bogor macet, maka lewat tol Jagorawi. Jika tidak, ya lewat Jalan Raya Bogor saja.

Percabangan



- Kondisi : Jln. Raya Bogor macet
- Jika terpenuhi?
 - Lewat tol Jagorawi
- Jika tidak?
 - Lewat Jln. Raya Bogor

Percabangan

- Dalam python, percabangan dituliskan dengan statement **if-else**

```
>>> if <kondisi> :
```

```
...     <blok kode jika kondisi terpenuhi>
```

```
... else :
```

```
...     <blok kode jika kondisi tidak terpenuhi>
```

```
...
```

```
>>>
```

Contoh

- Kita mau mengetes apakah suatu bilangan n adalah kelipatan 3 atau tidak (ingat operator modulo pada sesi sebelumnya)

```
>>> n = 1437
```

```
>>> if n%3 == 0 :
```

```
...     print("%d kelipatan 3" % n)
```

```
...     else :
```

```
...     print("%d bukan kelipatan 3" % n)
```

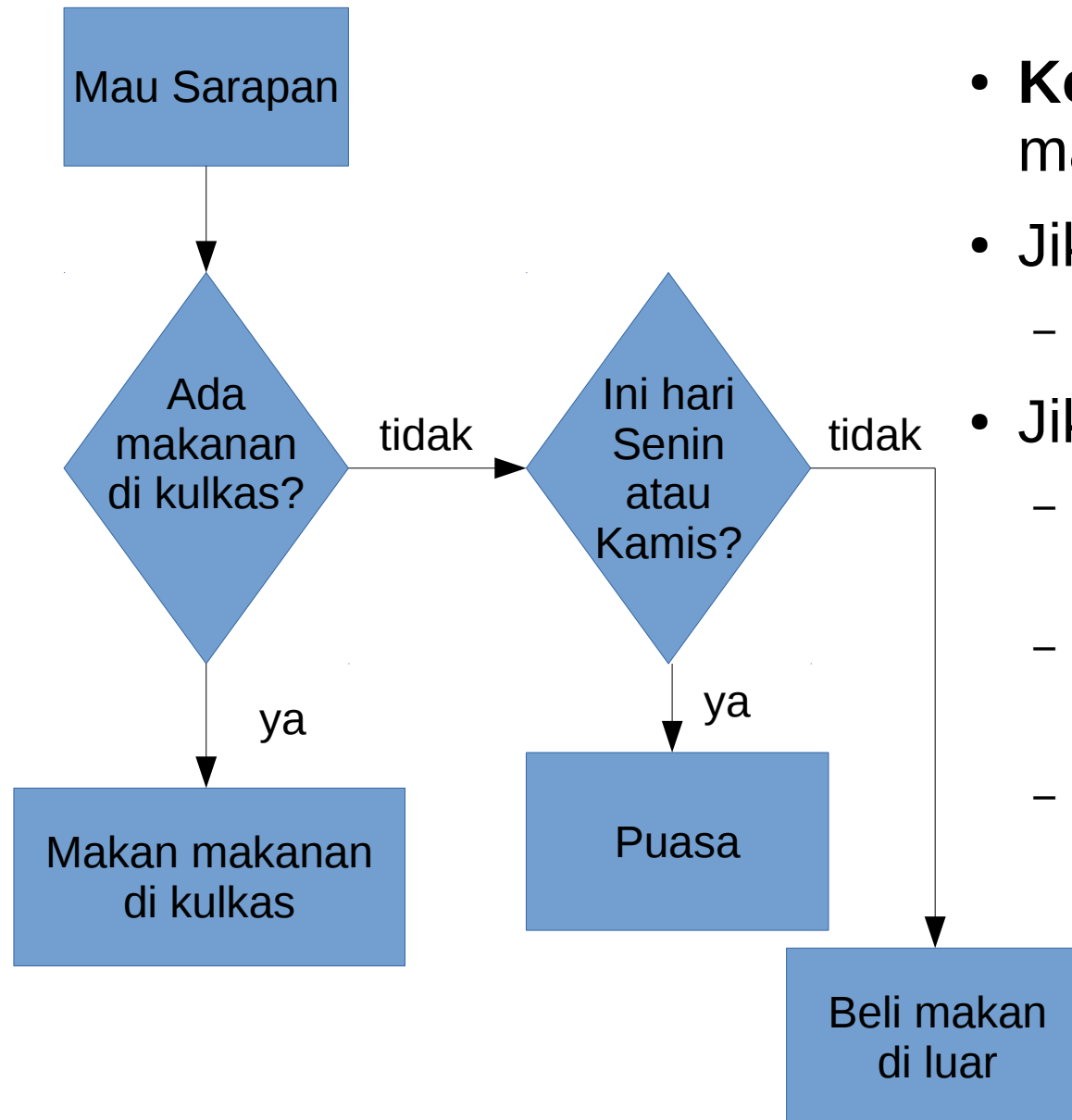
Latihan

- Ambil inputan dari user dengan statement **input()**
- Periksa apakah input dari user berupa bilangan atau bukan. Gunakan fungsi **.isdigit()** pada variabel inputannya
- Apabila inputan itu adalah bilangan, maka hitung dan tuliskan “input user adalah sebuah bilangan. Apabila bilangan tersebut dikali 2 jadi xxx”
- Apabila bukan bilangan (anggap sebagai string), maka tuliskan “input user adalah sebuah string. Apabila dituliskan 5 kali jadi xxx”

If-Elif-Else

- Jika kondisinya lebih dari satu, bisa dituliskan dengan operator boolean (and atau or), bisa juga dibuat bertingkat dengan if-elif-else
- Contoh : Mau sarapan. Kalau ada makanan di kulkas, maka makan yang ada di kulkas. Kalau gak ada, maka kita lihat harinya dulu. Kalau hari Senin atau Kamis, puasa sunah aja. Kalau hari lain, baru beli makan di luar.

If-Elif-Else



- **Kondisi 1:** ada makanan di kulkas?
- Jika terpenuhi?
 - makan
- Jika tidak?
 - Cek **kondisi 2:** hari ini Senin atau Kamis?
 - Jika terpenuhi
 - Puasa
 - Jika tidak
 - Beli makanan di luar

If-Elif-Else

```
>>> if <ada makanan di kulkas> :  
...     <ambil makanan tersebut>  
...     <makan>  
... elif <hari ini Senin> or <hari ini Kamis> :  
...     <gak jadi makan>  
...     <puasa aja>  
... else :  
...     <keluar, beli makanan>  
...     <makan>  
>>>
```

If-Elif-Else

- Atau dibuat if-else di dalam if-else (perhatikan tingkat indentasinya)

```
>>> if <ada makanan di kulkas> :
```

```
...     <ambil makanan tersebut>
```

```
...     <makan>
```

```
... else :
```

```
...     if <hari ini Senin> or <hari ini Kamis> :
```

```
...         <gak jadi makan>
```

```
...         <puasa aja>
```

```
...     else :
```

```
...         <keluar, beli makanan>
```

```
...         <makan>
```

```
>>>
```

Latihan

- Ambil input 3 buah bilangan a,b, dan c dari user
- Tampilkan bilangan yang terbesarnya* dengan membandingkan a dan b, lalu bandingkan yang lebih besar dengan c
- Ketiga bilangan tersebut adalah panjang sisi-sisi segitiga, coba tampilkan jenis segitiga apa yang diinput oleh user : segitiga sama sisi, sama kaki, siku-siku, segitiga dengan sudut tumpul, atau segitiga yang ketiga sudutnya lancip.
- Untuk menghitung lancip atau tumpul segitiganya, dapat dihitung dari kuadrat sisi-sisinya.

* menghitung maximal dari bilangan sebenarnya dapat digunakan fungsi **max()** tetapi untuk latihan if-else jangan dulu

```
>>> a = 5
```

```
>>> b = 11
```

```
>>> c = 7
```

```
>>> max(a,b,c)
```

```
11
```

Latihan

- Buat dua buah list :
 - list pertama berisi nama-nama buah
 - list kedua berisi nama-nama hewan
- Minta inputan dari user
- Periksa apakah masukan user tersebut berupa buah atau hewan
- Beritahukan hasilnya kepada user

Solusi

```
>>> buah = open("buah.txt", "r").read().split("\n")
>>> hewan = open("hewan.txt", "r").read().split("\n")
>>> barang = input("Masukkan suatu nama")
>>> if barang in buah:
...     print("%s adalah buah" %barang)
...     elif barang in hewan:
...         print("%s adalah hewan" %barang)
...     else:
...         print("%s bukan buah maupun hewan" %barang)
```

Switch Case

- Pada bahasa pemrograman lain, biasanya ada statement lain yang dapat digunakan untuk percabangan selain **if-else**, yaitu **switch-case**
- Pada python, statement switch-case **tidak ada**
- Tetapi kita dapat memanfaatkan dictionary untuk mendapatkan hasil yang sama

```
>>> cases = { 1 : "satu", 2 : "dua", 3 : "tiga"}
```

```
>>> result = cases.get(key, "default")
```


Perulangan

- Untuk mengulangi eksekusi dari instruksi pada python dapat digunakan
 - For
 - While
- For digunakan apabila kita tahu batas perulangannya
- While digunakan apabila kita tidak tahu batas perulangan, tetapi kita tahu syaratnya

Perulangan

- Contoh : apabila kita punya toko, dan untuk setiap pelanggan yang datang kita melakukan hal yang sama : greeting, order taking, serving, payment, sign off
- Jika kita tahu bahwa pelanggannya ada 10 orang, maka kita gunakan **for**. For <10 pelanggan>, kita greeting, order taking, serving, payment, sign off.
- Jika kita tidak tahu ada berapa pelanggan, tapi kita tahu bahwa pelayanan cuma sampe jam 5 sore, atau sampai barangnya habis, kita gunakan **while**. While <barang ada>, kita greeting, order taking, serving, payment, sign off.

Contoh

- Ada 5 user :
 - 1, Ayu, P, 18 tahun, Jeruk
 - 2, Budi, L, 21 tahun, Alpukat
 - 3, Cecep, L, 18 tahun, Durian
 - 4, Dona, P, 20 tahun, Melon
 - 5, Emily, P, 18 tahun, Pir
- Buat list berisi nama tersebut dan dictionary berisi kelengkapannya
- Tampilkan nama, usia, dan buah favorit masing masing user tersebut

Contoh

```
user = ["Ayu", "Budi", "Cecep", "Dona", "Emily"]
detail = {
    "Ayu" : {"id":1, "kelamin":"P", "usia":18, "favorit":"Jeruk"},
    "Budi" : {"id":2, "kelamin":"L", "usia":21, "favorit":"Alpukat"},
    "Cecep" : {"id":3, "kelamin":"L", "usia":18, "favorit":"Durian"},
    "Dona" : {"id":4, "kelamin":"P", "usia":20, "favorit":"Melon"},
    "Emily" : {"id":5, "kelamin":"P", "usia":18, "favorit":"Pir"},
}

for i in user:
    usia = detail[i]["usia"]
    fav = detail[i]["favorit"]
    print("%s, usianya %d, suka %s"%(i,usia,fav))
```

- Di sini digunakan **for** karena kita tahu yang akan ditampilkan hanya 5 orang tersebut (yang ada di dalam list)

Enumerate

- Fungsi enumerate akan menambahkan angka berurut pada sequence yang diberikan, dan disimpan sebagai tuple

```
>>> for i in enumerate(user):
```

```
...     print(i)
```

```
...     #print("%d \t %s" %i)
```

```
...
```

```
>>>
```

For Loop

- Variabel `i` pada contoh sebelumnya disebut sebagai **iterator**. Tidak harus menggunakan huruf `i`, bisa juga dengan variabel lain yang lebih representatif.
- Iterator memungkinkan programmer untuk menjelajahi (traverse) sebuah sequence, baik itu list, string, atau yang lainnya.
- Bagian setelah keyword **in** pada for loop dapat berupa sequence apapun, tidak hanya list.

Contoh

```
>>> buah = "alpukat"
>>> for huruf in buah:
...     print(huruf)
...
>>>
```

- Sequence berupa string

Contoh

```
>>> for i in range(10):
```

```
...     print("*" * i)
```

```
...
```

```
>>> for i in range(10):
```

```
...     print(" "*(10-i) + "*" * i)
```

```
...
```

```
>>> for i in range(10):
```

```
...     print(" "*(10-i) + "*" * (2*i-1))
```

```
...
```

- Sequence berupa range

Latihan

- Buat bentuk wajik (belah ketupat) dari simbol * seperti pada contoh sebelumnya
- Latihan ini perlu memanfaatkan percabangan dan perulangan

While Loop

- Perintah : Tuliskan semua bilangan kuadrat yang lebih kecil dari 1000!
- Berapa bilangan yang terakhir? Entah...
- Kita tidak bisa menggunakan for untuk kasus ini
- Tapi bisa menggunakan while
 - Tentukan counter awalnya (bilangan asli mulai dari 1)
 - While (kuadratnya kurang dari 1000), print, tambah 1 ke counter-nya
- Komputer akan melakukan looping selama syaratnya masih terpenuhi

While Loop

```
>>> i = 1
>>> while (i**2 <1000):
...     print(i, i**2)
...     i = i + 1
...
>>>
```

- Jangan lupa diubah counter/iterasinya setiap iterasi dan pastikan ada escape condition. Jika tidak bisa terjadi **infinite loop**.

Kondisi Break, Continue, dan Pass

- Ada beberapa pernyataan pada python yang dapat digunakan untuk mengatur perilaku program pada perulangan
 - Break : mengakibatkan program keluar dari for loop atau while loop
 - Continue : mengakibatkan program menghentikan eksekusi blok kode di iterasi tersebut dan melakukan blok kode pada iterasi selanjutnya
 - Pass : program tidak melakukan apapun pada baris ini

Contoh

- Bandingkan perilaku 3 kondisi di samping!
 - Break membuat statement di bawahnya pada $i=5$ serta iterasi $i=6$ dan seterusnya tidak dilakukan
 - Continue membuat statement di bawahnya pada $i=5$ tidak dilakukan, tapi iterasi dilanjutkan
 - Pass tidak mempengaruhi apa-apa. Dilakukan apabila harus menulis suatu pernyataan tapi program tidak perlu melakukan aksi apapun. Kalau **pass** ada di akhir loop, perilakunya akan sama seperti **continue**

```
|  
for i in range(10):  
    print(i)  
    if i == 5:  
        break  
    print("bawahnya", i)
```

```
for i in range(10):  
    print(i)  
    if i == 5:  
        continue  
    print("bawahnya", i)
```

```
for i in range(10):  
    print(i)  
    if i == 5:  
        pass  
    print("bawahnya", i)
```

Latihan

- Siapkan sebuah list untuk menampung masukan user
- Selama user belum memilih untuk tidak melanjutkan, maka ambil masukan user
- Setelah user selesai menginput, tampilkan string-string yang dimasukkan oleh user dalam beberapa baris, 3 string tiap barisnya