

Modul 3

Alur Kontrol

Di dalam program yang kita lihat hingga saat ini, selalu saja urutan statemen yang dijalankan oleh Python berurutan dari atas ke bawah. Bagaimana jika Anda ingin mengubah alur kerjanya? Sebagai contoh Anda ingin program untuk mengambil keputusan dan bertindak secara berbeda tergantung pada kondisi yang ada. Sebagai contoh, misalnya mencetak 'Selamat Pagi' atau 'Selamat Sore' tergantung waktu yang ada saat itu?

Hal ini dapat dilakukan lewat statemen alur kontrol. Ada tiga macam statemen alur kontrol di Python - `if`, `for` dan `while`.

Statemen If

Statemen `if` digunakan untuk mengecek kondisi: jika kondisi `if` bernilai benar, maka kita akan menjalankan satu blok statemen (disebut `if-block`), jika tidak akan diteruskan dengan statemen `else` kita gunakan untuk memproses blok statemen yang lain (dinamakan `else-block`). Bagian `else` tersebut sifatnya tidak wajib atau opsional.

Kita dapat menambahkan kondisi dalam `else-block` menggunakan `elif`.

```
# lat9.py
1
2 nomor_acak = 7
3 print ('tebak nomor acak dari 1 - 10')
4
5 # ``raw_input`` digunakan untuk mendapatkan input dari
6 pengguna
7 # ``int`` digunakan untuk konversi tipe data ``str`` ke
8 ``int``
9 tebakkan = int(raw_input('Tebakan anda (bil bulat): '))
10
11 if tebakkan == nomor_acak:
12     print('Selamat! tebakkan anda benar')
13     print('tapi tidak ada hadiah untuk anda :( ' )
14 elif tebakkan < nomor_acak:
15     print('tebakkan anda terlalu kecil')
16 else:
17     print('tebakkan anda terlalu besar')
18
19 print('selesai')
```

Baris 11-12 adalah `if-block`, baris 14 adalah `elif-block`, dan baris 16 adalah `else-block`.

Bagaimana program ini bekerja?

Program ini akan meminta inputan tebakkan dari pengguna berupa bilangan. Untuk mendapatkan inputan ini kita gunakan fungsi `raw_input`. Keluaran dari fungsi ini adalah string yang diinputkan oleh user, oleh karena itu kita harus melakukan konversi ke tipe data `int`. Untuk konversi ini kita gunakan fungsi `int`. Hasil dari inputan pengguna yang sudah dikonversi disimpan dalam variabel `tebakkan`.

Sebelumnya program telah menentukan bilangan acak yang disimpan dalam variabel `nomor_acak`. Setelah mendapatkan input dari pengguna, program masuk kedalam alur kontrol `if`. Jika tebakan dan nomor acak sama maka tampilkan pesan berhasil, jika tebakan kurang dari nomor acak maka tampilkan pesan tebakan terlalu kecil, dan terakhir berarti tebakan terlalu besar.

Statemen While

Statemen `while` merupakan statemen untuk perulangan, `block` kode akan dijalankan terus menerus selama kondisi benar. Statemen `while` dapat mempunyai bagian `else` (opsional).

```
# lat10.py
# acak looping

nomor_acak = 77
berjalan = True

print('tebak nomor acak dari 1 - 100')

while berjalan:
    tebakan = int(raw_input('Tebakan anda (bil bulat): '))

    if tebakan == nomor_acak:
        print('Selamat! tebakan anda benar')
        print('tapi tidak ada hadiah untuk anda :(')
        berjalan = False
    elif tebakan < nomor_acak:
        print('tebakan anda terlalu kecil')
    else:
        print('tebakan anda terlalu besar')
else:
    print('selesai')
```

Perulangan diatas berhenti jika `berjalan`(kondisi) bernilai `False`. `True` dan `False` merupakan obyek bertipe boolean, dan nilai `True` sama dengan nilai 1, nilai `False` sama dengan nilai 0.

```
>>> True == 1
True
>>> False == 0
True
```

Obyek dapat dinilai atau dikonversi ke nilai boolean

```
>>> bool('nama')
True
>>> bool('')
False
>>> bool(0)
False
>>> bool(-5)
True
```

Perulangan For (For Loop)

Statemen perulangan `for ... in ...` merupakan statemen perulangan selain `while`. Statemen ini melakukan *iterasi* dari rangkaian obyek, berjalan melalui tiap item yang ada pada rangkaian / sequence. Apa itu rangkaian / *sequence*? rangkaian yaitu koleksi item yang terurut.

```
# lat11.py

for i in range(1, 6):
    print(i)
else:
    print('Perulangan sudah selesai')
```

Program ini akan mencetak rangkaian / sequence bilangan, dari 1 sampai 5. Kita membuat rangkaian bilangan ini menggunakan fungsi *builtin* `range`. Apa yang kita lakukan yaitu memanggil fungsi `range` dengan dua parameter, `range` akan mengembalikan rangkaian bilangan dari parameter pertama sampai batas parameter kedua (eksklusif). Sebagai contoh `range(1, 6)` menghasilkan rangkaian `[1, 2, 3, 4, 5]`.

Jika kita memanggil `range` dengan parameter ketiga, yaitu parameter jumlah langkah. Contoh `range(1, 6, 2)` mengembalikan rangkaian `[1, 3, 5]`.

Bagian `else` adalah opsional dan akan selalu dijalankan kecuali jika ada statemen `break`.

Statemen Break

Statemen `break` digunakan untuk keluar dari perulangan, misalnya keluar dari perulangan walaupun kondisi perulangan masih `True` atau rangkaian / sequence belum diiterasi seluruhnya.

```
# lat12.py

while True:
    data = raw_input('Masukkan sesuatu : ')
    if data == 'keluar':
        break
    print('Inputan pengguna "%s"' % data)
print('Selesai')
```

Program ini akan terus meminta inputan pengguna dan akan berhenti ketika pengguna menginputkan `keluar`.

```
# lat13.py

for i in range(1, 11):
    print i
    if i == 5:
        break
else:
    print('Tidak dijalankan karena break')
```

Bagian `else` tidak akan dijalankan karena perulangan tidak berhenti secara normal.

Statemen Continue

Statemen `continue` digunakan untuk melewati statemen yang ada dalam blok perulangan dan *continue* / melanjutkan ke iterasi berikutnya.

```
# lat14.py

for i in range(1, 11):
    if i % 2 == 0:
        # skip bilangan genap
        continue
    print(i)
```

Latihan

1. Buatlah program sederhana untuk menentukan sebuah bilangan yang diinput bilangan ganjil atau genap.
2. Inputkan sebuah bilangan dengan menggunakan perintah input. Jika bilangan tersebut habis dibagi 3 (tiga), tampilkan tulisan di layar : **12 habis dibagi 3** jika tidak habis tuliskan : **tidak habis dibagi 3**.
3. Jumlah 4 buah bilangan kuadrat pertama adalah 30 ($30=1+4+9+16$), buatlah program untuk menjumlahkan bilangan kuadrat pertama yang ditentukan berdasarkan input dari pengguna.
4. Buat program untuk menentukan bobot untuk masing-masing penilaian, misalnya nilai presensi, tugas, UTS, dan UAS. Kemudian, hitung nilai akhir berdasarkan pembobotan nilai. Lalu, tentukan grade dan status kelulusan. Rumus perhitungan = (10% presensi + 30% tugas + 30% UTS + 30% * UAS).