

# **BAB VI**

## **REPETITION**

### **6.1 Tujuan Praktikum**

- Mahasiswa mampu memahami konsep perulangan/repetition
- Mahasiswa mampu memahami cara mendefinisikan perulangan
- Mahasiswa dapat membuat program sederhana menggunakan perulangan

### **6.2 Alokasi Waktu**

1 x Pertemuan = 50 Menit

### **6.3 Dasar Teori**

#### **6.3.1 Repetition**

Perulangan atau bisa juga disebut dengan looping/iteration/repetition, ada dua jenis perulangan pada python yaitu for dan while.

##### **a) For**

Secara sederhana penggunaan for ditujukan untuk:

- Counted loop (perulangan yang terhitung)
- Biasanya digunakan untuk perulangan yang sudah jelas banyaknya
- Perulangan berdasarkan jumlah elemen

##### **b) While**

Secara sederhana penggunaan while ditujukan untuk:

- Uncounted loop (perulangan yang tak terhitung)
- Biasanya digunakan untuk perulangan yang belum tentu banyaknya

#### **6.3.2 Control Statements**

Control statements adalah sebuah statement yang bisa digunakan untuk mengubah cara eksekusi dalam percabangan dan perulangan dari cara eksekusi normal. Control statement terdiri dari:

- Break Statement
- Continue Statement
- Pass Statement

##### **a) Break Statement**

Break statement adalah sebuah statement yang akan membuat suatu program berhenti atau keluar dari suatu blok program.

b) Continue Statement

Continue statement adalah sebuah statement yang akan membuat suatu program langsung menuju step selanjutnya atau interval berikutnya dan mengabaikan (skip) baris kode di bawahnya.

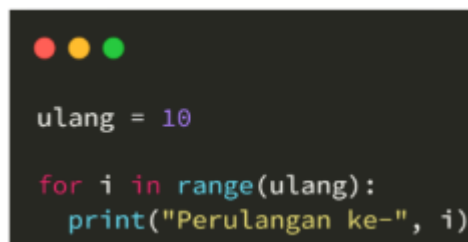
c) Pass Statement

Pass statement adalah sebuah statement yang akan membuat suatu blok program tidak akan dieksekusi.

## 6.4 Latihan Praktikum

### 6.4.1 Latihan 1

Buatlah sebuah file dengan nama **for.py** kemudian ketikkan kode berikut :



```
ulang = 10

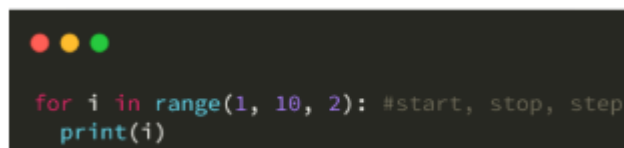
for i in range(ulang):
    print("Perulangan ke-", i)
```

*Gambar 54. Coding repetition for*

Amati hasilnya dan modifikasilah variable ulang !

### 6.4.2 Latihan 2

Buatlah sebuah file dengan nama **for\_range.py** kemudian ketikkan kode berikut:



```
for i in range(1, 10, 2): #start, stop, step
    print(i)
```

*Gambar 55. Coding repetition for untuk range 1*

Amati hasilnya dan modifikasilah fungsi range! Fungsi range() memiliki beberapa opsi yaitu start, stop dan step. Lebih jelasnya ketikkan kode berikut:

### 6.4.3 Latihan 3

Buatlah sebuah file dengan nama **range.py** kemudian ketikkan kode berikut:



```
#menggunakan start dan stop
for i in range(1, 10):
    print(i, end=' ')

print()

#menggunakan start dan stop default
for i in range(10):
    print(i, end=' ')

print()

#menggunakan start, stop dan step bersamaan
for i in range(1, 15, 3):
    print(i, end=' ')

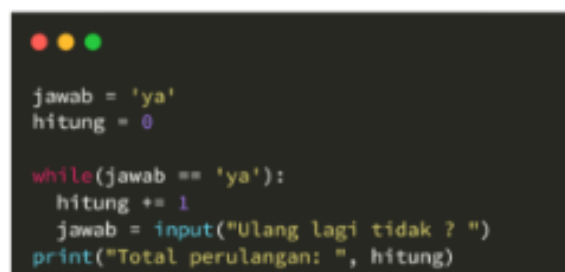
print()
```

Gambar 56. Coding repetition for untuk range 2

- Pada bagian kode `range(1, 10)`, angka 1 menjadi bagian start atau angka awal dan 10 menjadi bagian stop. Urutan yang tercetak pada Shell adalah 1 2 3 4 5 6 7 8 9 sebelum nilai 10 perulangan berhenti atau bisa diartikan  $\text{stop} - 1$  ( $10 - 1$ )
- Pada bagian kode `range(10)`, secara default bagian start akan bernilai 0 dan 10 menjadi bagian stop. Urutan yang tercetak pada Shell adalah 0 1 2 3 4 5 6 7 8 9 sebelum nilai 10 perulangan berhenti atau bisa diartikan  $\text{stop} - 1$  ( $10 - 1$ )
- Pada bagian kode `range(1, 15, 3)`, angka 1 menjadi bagian start atau angka awal, angka 15 menjadi bagian stop dan angka 3 menjadi step. Step di sini berarti setiap satu iterasi akan loncat sebanyak 3 sehingga urutan yang tercetak pada Shell adalah 1 4 7 10 13 sebelum nilai 15 perulangan berhenti atau bisa diartikan  $\text{stop} - 1$  ( $15 - 1$ ) tetapi karena hasil step melebihi batas stop maka nilai terakhir adalah nilai step yang paling akhir.

### 6.4.4 Latihan 4

Buatlah sebuah file dengan nama **while.py** kemudian ketikkan kode berikut:



```
jawab = 'ya'
hitung = 0

while(jawab == 'ya'):
    hitung += 1
    jawab = input("Ulang lagi tidak ? ")
print("Total perulangan: ", hitung)
```

Gambar 57. Coding repetition while

Amati hasilnya dan diskusikan!

#### 6.4.5 Latihan 5

Buatlah sebuah file dengan nama `while_question.py` kemudian ketikkan kode berikut:



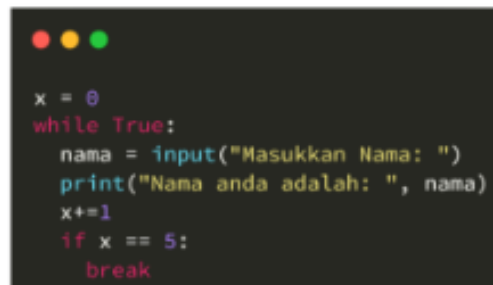
```
while True:
    reply = input('Enter text, [type "stop" to quit]: ')
    if reply == 'stop':
        break
    print(reply.lower())
```

Gambar 58. Coding repetition while dengan if statement

Amati hasilnya dan diskusikan!

#### 6.4.6 Latihan 6

Buatlah sebuah file dengan nama `while_question2.py` kemudian ketikkan kode berikut:



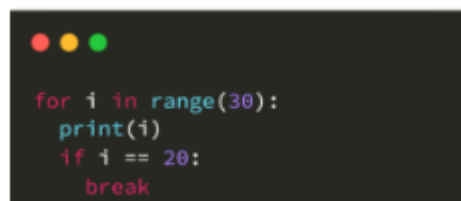
```
x = 0
while True:
    nama = input("Masukkan Nama: ")
    print("Nama anda adalah: ", nama)
    x+=1
    if x == 5:
        break
```

Gambar 59. Coding repetition while dengan if statement dan operator

Amati hasilnya dan diskusikan !

#### 6.4.7 Latihan 7

Buatlah sebuah file dengan nama `break.py` kemudian ketikkan kode berikut:



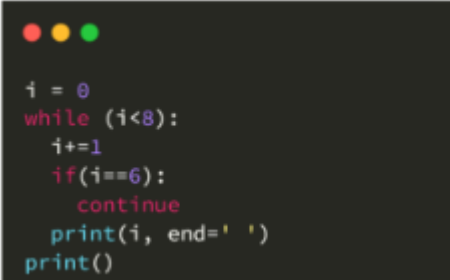
```
for i in range(30):
    print(i)
    if i == 20:
        break
```

Gambar 60. Coding break statement

Ketika nilai perulangan dari variabel `i` sama dengan 20 maka proses perulangan akan dihentikan atau dikeluarkan dari blok program menggunakan `break`.

#### 6.4.8 Latihan 8

Buatlah sebuah file dengan nama `continue.py` kemudian ketikkan kode berikut:



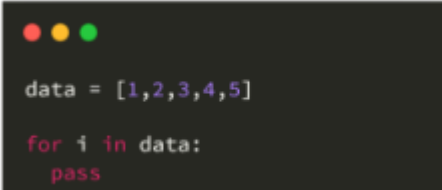
```
i = 0
while (i<8):
    i+=1
    if(i==6):
        continue
    print(i, end=' ')
print()
```

*Gambar 61. Coding continue statement*

Hasil dari kode program diatas adalah 1 2 3 4 5 7 8 bisa dilihat angka 6 tidak ikut ditampilkan karena pada blok program while terdapat statement continue yang berada dalam blok if dengan kondisi ketika nilai dari variabel i sama dengan 6 maka baris berikutnya akan dilewati menuju perulangan selanjutnya.

#### 6.4.9 Latihan 9

Buatlah sebuah file dengan nama `pass.py` kemudian ketikkan kode berikut:



```
data = [1,2,3,4,5]

for i in data:
    pass
```

*Gambar 62. Coding pass statement*

Ketika kode program tersebut dijalankan maka tidak akan menghasilkan output apapun karena pada blok perulangan terdapat pass statement. Untuk lebih mendalami materi perulangan lakukanlah latihan-latihan berikut:

#### 6.4.10 Latihan 10

Buatlah sebuah file dengan nama `for_list.py` kemudian ketikkan kode berikut:



```
data = [1, 2, 'tiga', 4, 'lima', 'enam', 7, 8.2, 9.3, 'sepuluh']
batas = len(data)

for i in range(batas):
    print("data indek ke", i, "adalah", data[i])
```

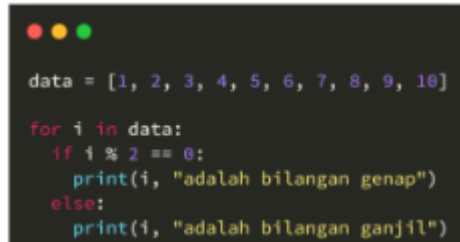
*Gambar 63. Coding repetition for dengan list*

Amati hasilnya dan diskusikan!

Tipe data dari variabel data adalah list, setiap elemen yang ada di dalam list diwakilkan oleh index, index selalu dimulai dari 0. Untuk lebih jelasnya mengenai tipe data list akan di jelaskan pada pertemuan selanjutnya.

#### 6.4.11 Latihan 11

Buatlah sebuah file dengan nama `for_condition.py` kemudian ketikkan kode berikut:



```
data = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

for i in data:
    if i % 2 == 0:
        print(i, "adalah bilangan genap")
    else:
        print(i, "adalah bilangan ganjil")
```

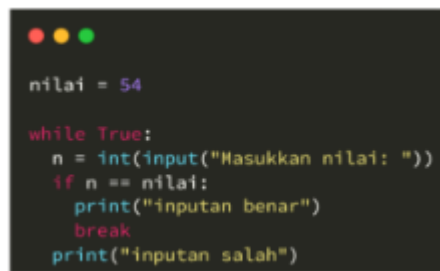
*Gambar 64. Coding repetition for dengan list dan if else statement*

Amati hasilnya dan diskusikan!

Pada kode di atas, di dalam perulangan ada percabangan. Bandingkanlah cara menampilkan data dalam bentuk perulangan antara kode pada Latihan 10 dan Latihan 11.

#### 6.4.12 Latihan 12

Buatlah sebuah file dengan nama `while_tebak_nilai.py` kemudian ketikkan kode berikut:



```
nilai = 54

while True:
    n = int(input("Masukkan nilai: "))
    if n == nilai:
        print("inputan benar")
        break
    print("inputan salah")
```

*Gambar 65. Coding repetition while dengan if dan break statement*

Amati hasilnya dan diskusikan!

Perulangan tidak akan berhenti sampai nilai yang dimaksud diinputkan.

#### 6.4.13 Latihan 13

Buatlah sebuah file dengan nama `while_list.py` kemudian ketikkan kode berikut:

```
lines = list()
testAnswer = input("Press 'y' if you want to enter more lines: ")
while testAnswer == 'y':
    line = input('Next line: ')
    lines.append(line)
    testAnswer = input("Press 'y' if you want to enter more lines: ")

print('Your lines were: ')
for line in lines:
    print(line)
```

Gambar 66. Coding repetition while dan for

Amati hasilnya dan diskusikan!

### 6.5 Tugas Praktikum

1. Buatlah sebuah program kalkulator dengan skenario berikut:
  - Buatlah sebuah kalkulator sederhana untuk menghitung 2 buah bilangan dengan operator(\*,/,+,-).
  - Jika operator (\*) diinputkan maka akan melakukan proses perkalian.
  - Jika operator (/) diinputkan maka akan melakukan proses pembagian.
  - Jika operator (+) diinputkan maka akan melakukan proses penjumlahan.
  - Jika operator (-) diinputkan maka akan melakukan proses pengurangan.
  - Jika operator yang diinputkan selain operator (\*,/,+,-), munculkan pesan operator tidak valid.
  - Jika operator yang diinputkan adalah bilangan 0 maka program diakhiri.
  - Bilangan tidak boleh dibagi dengan 0.
  - Gunakan percabangan dan perulangan
2. Buatlah piramida bintang menggunakan perulangan, boleh memakai for maupun while



Gambar 67. Piramida dari hasil repetition

3. Buatlah sebuah file dengan nama `sum_number_xxxx.py` dengan skenario berikut:
  - Buatlah sebuah list dengan nama `numbers`, isikan list tersebut nilai 7,8,9,10,11,12,13,6,5,4,3,2,1
  - Jumlahkan semua nilai tersebut
  - Gunakan `for`
4. Pengumpulan laporan praktikum dalam bentuk pdf