



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71180294
Nama Lengkap	Michael Yoga Setiawan
Minggu ke / Materi	05 /Struktur Kontrol Perulangan

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

BAGIAN 1: MATERI MINGGU INI (40%)

Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

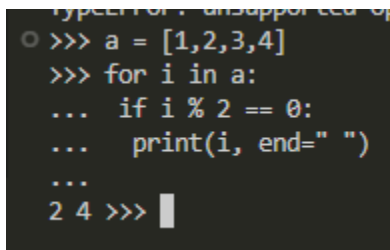
Definisi Perulangan

Perulangan dilakukan saat:

- Melakukan hal yang sama berulang – ulang
- Melakukan operasi secara bertahap dengan tahapan yang sama
- Mengolah data dari struktur data dengan operasi yang sama setiap datanya.

Perulangan For

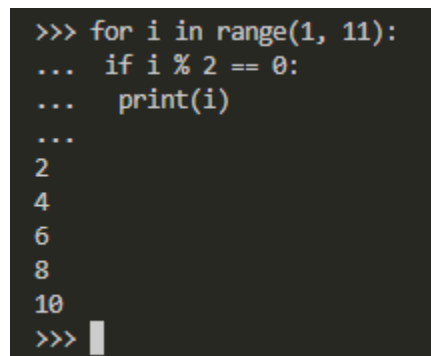
Bentuk perulangan for digunakan jika setiap data dari sebuah struktur data diolah dengan cara yang sama. Misalnya, ada list `a = [1, 2, 3, 4]`. Setiap data yang ada pada struktur tersebut dimaksudkan untuk dicek satu – persatu, lalu jika data merupakan bilangan genap, komputer akan meng-outputkan data. Dalam kasus ini bisa memakai bentuk for:



```
>>> a = [1,2,3,4]
>>> for i in a:
...     if i % 2 == 0:
...         print(i, end=" ")
...
2 4 >>>
```

Dari gambar tersebut, `i` merupakan variabel yang menyimpan data dari list `a`, dan `a` adalah nama list.

Bentuk perulangan for juga dapat menggunakan data dari rentang angka dengan memanfaatkan fungsi `range()`. Misal, kita hendak menampilkan angka genap dari rentang angka 1 — 10, kita bisa menggunakan perulangan for:



```
>>> for i in range(1, 11):
...     if i % 2 == 0:
...         print(i)
...
2
4
6
8
10
>>>
```

Berdasarkan kegunaannya, kita bisa memanfaatkannya untuk melakukan perulangan yang jumlah perulangannya sudah diketahui. Misalnya, kita mau menge-print “Hello world” sebanyak 10 kali, kita bisa memanfaatkan struktur data yang jumlahnya 10, seperti range(10). Karena jumlah data adalah 10, akan dilakukan perulangan sebanyak 10 kali dan dalam setiap perulangan, kita bisa memprogramnya agar menge-print “Hello world” pada setiap perulangan.

```
>>> for i in range(10):  
...   print("Hello, world!")  
...  
Hello, world!  
Hello, world!  
Hello, world!  
Hello, world!  
Hello, world!  
Hello, world!  
Hello, world!  
Hello, world!  
Hello, world!  
>>> █
```

Perulangan While

Konsep dasar dari perulangan while adalah penggunaan proposisi boolean expression: jika proposisi bernilai True, perulangan akan terus dilakukan hingga proposisi bernilai False. Jika nilai proposisi selalu True, perulangan akan dilakukan tak hingga atau biasa disebut infinite loop. Infinite loop seringkali merupakan hal yang perlu dihindari sehingga program harus diprogram sedemikian rupa sehingga proposisi dapat berganti nilai menjadi False, seperti mengganti nilai variabel.

Program di gambar tersebut akan menampilkan nilai 0 — 9. Variabel n awalnya bernilai 0 sehingga $n < 10$ bernilai True. Karena proposisi bernilai True, instruksi di bawah while akan dieksekusikan: sistem akan meng-outputkan nilai n, yaitu 0 dan mengganti nilai n dengan nilai data sebelumnya ditambah 1 sehingga $n = 1$. Kemudian, komputer akan mengecek kembali apakah proposisi masih bernilai True. Karena $n = 1$, $n < 10$ bernilai True, maka perulangan akan dilanjutkan. Perulangan akan terus dilanjutkan hingga $n < 10$ bernilai False, yaitu saat $n = 10$.

```
>>> n = 0
>>> while n < 10:
...     print(n)
...     n+= 1
...
0
1
2
3
4
5
6
7
8
9
>>> 
```

Penggunaan Break dan Continue

Perulangan dapat dikontrol menggunakan break dan continue.

- Break

Fungsi dari break adalah untuk menghentikan perulangan di tengah jalan. Contoh:

```
>>> while (True):
...     n = int(input())
...     if n % 2 == 0:
...         break
...
1
3
5
3
2
>>> 
```

Program tersebut akan meminta pengguna memasukkan input angka. Proposisi selalu bernilai benar, oleh karena itu perulangan akan dilakukan terus-menerus. Jika input angka yang dimasukkan bukan merupakan bilangan genap, maka program akan meminta input pengguna lagi. Jika input pengguna adalah bilangan genap, instruksi break akan dieksekusikan dan perulangan akan berhenti.

- Continue

Continue dipakai untuk mengabaikan perulangan yang sekarang dan berpindah ke perulangan yang selanjutnya. Contoh:

```
>>> for i in range(1, 11):  
...     if i % 2 == 1:  
...         continue  
...     print(i)  
...  
2  
4  
6  
8  
10  
>>>
```

Program tersebut akan meng-outputkan angka yang bernilai genap dalam rentang angka 1 — 10. Jika angka adalah ganjil, instruksi `continue` akan dieksekusikan sehingga perintah `print(i)` diabaikan dan langsung ke perulangan selanjutnya. Jika angka adalah genap, komputer akan meng-outputkan angka genap.

5. Konversi Bentuk For Menjadi While

Jika memanfaatkan fungsi `range()`, semua bentuk `for` bisa diubah menjadi `while`; tidak semua bentuk `while` bisa diubah menjadi `for`.

BAGIAN 2: LATIHAN MANDIRI (60%)

Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

SOAL 1

```
#soal 1

# Fungsi ini akan menerima dua buah parameter: a dan b.
# kedua nya bernilai integer. Cara kerjanya, buat sebuah
# variabel hasil dengan nilai 0 terlebih dulu dan lakukan
# perulangan sebanyak nilai a, tambahkan hasil dengan nilai
# b. jika sudah selesai, fungsi akan mengembalikan hasil.

def perkalian(a, b) :

    hasil = 0

    for _ in range(a):

        hasil += b

    return hasil
```

SOAL 2

Tulis jawaban anda untuk soal nomor 2 di sini. Format untuk soal nomor 3 dan seterusnya juga sama.

```
#soal 2

# Fungsi ini akan menerima dua buah parameter: bawah dan atas.
# Cek dulu apakah bawah lebih kecil dari atas. Jika iya, nilai
# lompatan adalah 1, sedangkan -1 jika sebaliknya. Dengan meman-
# faatkan fungsi range(), masukkan nilai bawah atas dan jump.
# lakukan perulangan dengan for sambil mengeprint hasil angka.

def deret_bilangan_ganjil(bawah, atas):

    jump = 1

    if (bawah > atas):

        jump = -1

    for i in range(bawah, atas, jump):

        if (i % 2 == 1):
```

```
        print(i, end=" ")
    print()
```

SOAL 3

Tulis jawaban anda untuk soal nomor 3 di sini. Format untuk soal nomor 4 dan seterusnya juga sama.

```
#soal 3

# fungsi ini akan meminta pengguna untuk menginputkan
# berapa jumlah mata kuliah. Jika sudah, buat sebuah
# dictionary yang berisi pasangan nilai huruf dan angka.
# Gunakan perulangan untuk menghitung semua nilai angka yang ada,
# kemudian gunakan rumus ips untuk menghitung hasil akhir

def ips():
    n = int(input("Masukkan jumlah mata kuliah: "))
    sks = 3 * n
    sum_nilai = 0
    kamus = {
        "A": 4, "B": 3, "C": 2, "D": 1
    }
    for i in range(n):
        sum_nilai += kamus[(input(f"Nilai MK {i+1}? "))]
    print(f"Nilai IPS Anda semester ini {round(sum_nilai*3/sks, 2)}")
```