



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71230976
Nama Lengkap	Galih Pramana Chandra Prasetya
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 dalam dunia pemrograman adalah baris kode atau instruksi yang dieksekusi oleh komputer secara berulang-ulang sampai suatu kondisi tertentu terpenuhi. Pada Python, perulangan digunakan untuk mengakses data-data dalam sebuah struktur data seperti List, Tuple, Queue, Stack, dan lainnya. Pada Python, perulangan dapat dilakukan dengan menggunakan **for**, **while** maupun dengan cara **rekursif**.

No.	Tipe Perulangan	Penjelasan
1	For	Mengulangi blok kode berdasarkan jumlah elemen dalam suatu koleksi atau range yang ditentukan
2	While	Perulangan yang bersifat indefinite alias tidak pasti, atau bahkan tidak terbatas.
3	Rekursif	fungsi rekursif merupakan sebuah metode perulangan yang bersifat non-iterasi. Memanggil dirinya sendiri sehingga menimbulkan efek perulangan.

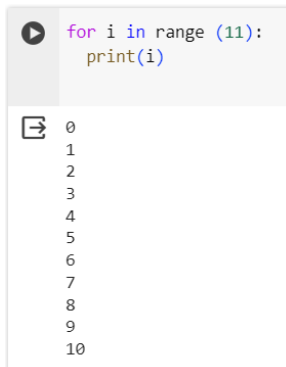
Di atas adalah table penjelasan untuk **for**, **while**, dan **rekursif**. Untuk pertemuan ini, hanya akan dibahas mengenai perulangan for dan while.

Bentuk Perulangan for

Perulangan **for** biasanya digunakan pada kondisi:

- Jumlah perulangan sudah diketahui sejak awal.
- Perulangan terjadi karena operasi yang sama pada suatu rentang data atau rentang nilai.

```
for i in range (11):  
    print(i)
```



Nilai default untuk Start adalah 0 dan untuk step memiliki nilai default 1.

Perulangan **for** dapat dibatasi dengan menggunakan fungsi `range(start,stop,step)`, fungsi digunakan untuk menghasilkan rentang dari start, sampai stop dengan peningkatan sejumlah step.

```
▶ for i in range (2,16,2):  
    print(i)
```

```
⇒ 2  
   4  
   6  
   8  
  10  
  12  
  14
```

Untuk code di atas, nilai **start** nya adalah 2, jadi i dimulai dari angka 2, untuk **stop** nya adalah 16, jadi i akan berhenti sebelum 16, sedangkan untuk **step** nya adalah 2, untuk penjumlahan setiap angka nya adalah 2 (2,4,6,...,14).

Berikut contoh program Python untuk mencari bilangan genap dalam rentang 0 sampai 50 menggunakan perulangan for.

```
▶ for i in range (0,50):  
    if i % 2 == 0:  
        print(i, end=" ")
```

96 98

```
⇒ 0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48
```

Step negatif

Contoh penggunaan step negatif,

```
▶ for i in range(50, 1,-2):  
    print(i, end=" ")
```

```
50 48 46 44 42 40 38 36 34 32 30 28 26 24 22 20 18 16 14 12 10 8 6 4 2
```

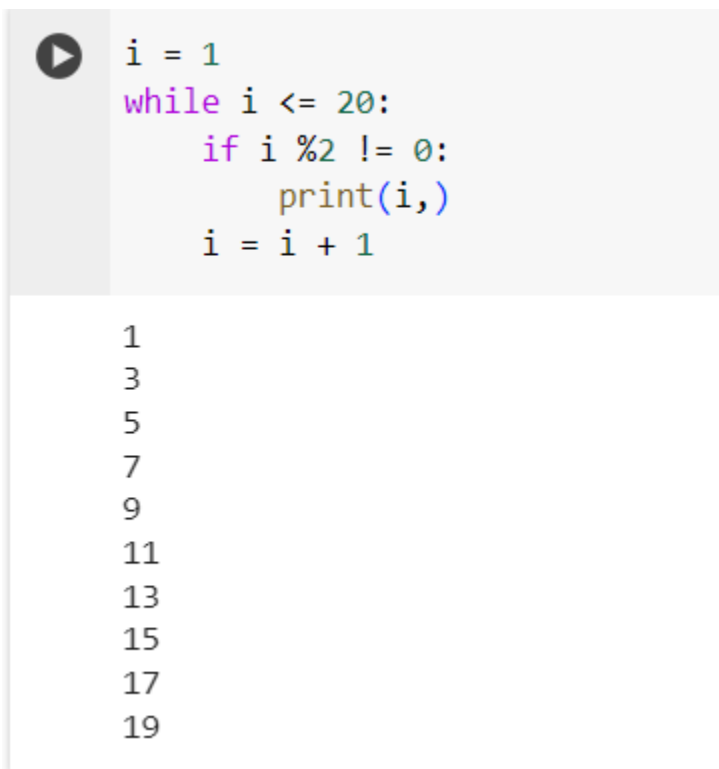
Program tersebut akan menampilkan bilangan genap mulai dari 50, 48, 46, 44, ..., sampai 2.

Bentuk Perulangan While

Perulangan while digunakan ketika kita tidak mengetahui kapan perulangan harus berhenti. .
Bentuk perulangan while secara umum adalah sebagai berikut:

```
<start>
while <stop>:
    operation
    operation
<step (optional)>      (diambil dari modul 5)
```

Contoh penggunaan **while** pada program untuk mencari bilangan ganjil antara 1 sampai 20.



```
i = 1
while i <= 20:
    if i % 2 != 0:
        print(i,)
    i = i + 1
```

1
3
5
7
9
11
13
15
17
19

Ketika nilai variable sudah kurang dari atau sama dengan 20, kode akan berhenti.

Penggunaan Break dan Continue

Perulangan dapat diatur dengan menggunakan **break** dan **continue**. Secara umum **break** digunakan untuk menghentikan perulangan Ketika kondisi sudah terpenuhi, sedangkan **continue** bertujuan untuk mengabaikan atau melewati instruksi perulangan ketika kondisi tertentu terpenuhi. Setelah itu, program akan melanjutkan perulangan hingga mencapai batas yang ditetapkan

1. Contoh penggunaan **break**,

```
▶ for i in range(1, 11):  
    if i == 5:  
        break  
    else:  
        print(i)  
print('Selesai')
```

```
⇒ 1  
   2  
   3  
   4  
   Selesai
```

2. Contoh penggunaan **continue**,

```
▶ for i in range(1, 11):  
    if i == 5:  
        continue  
    else:  
        print(i)  
print('Selesai')
```

```
⇒ 1  
   2  
   3  
   4  
   6  
   7  
   8  
   9  
  10  
   Selesai
```

Pada program yang menggunakan **break**, program berhenti saat nilai variable sudah mencapai 5, sedangkan pada program yang menggunakan **continue**, program akan melewati nilai 5 dan tetap melanjutkan program sampai batas akhir.

Konversi dari Bentuk for Menjadi Bentuk while

Bentuk perulangan for sebagian besar dapat dikonversi menjadi bentuk while. Ada beberapa hal yang harus diperhatikan saat konversi bentuk for menjadi bentuk while:

- Harus ada nilai awal untuk memulai perulangan.
- Harus ada nilai akhir untuk mengakhiri perulangan.
- Harus ada Langkah, agar iterasi dari nilai awal bisa terus berjalan sampai mencapai nilai akhir

Contoh konversi bentuk for menjadi bentuk while,

```
▶ for i in range (1,12):  
    print(i, end=" ")
```

```
➞ 1 2 3 4 5 6 7 8 9 10 11
```

Dari contoh tersebut, dapat diketahui bahwa perulangan dimulai dari 1 dan berakhir pada 12. Proses konversi dapat dilakukan dengan membuat variabel dan kemudian memasukkannya ke dalam perulangan **while**, seperti yang terlihat pada contoh berikut:

```
▶ i = 1  
while i <= 12:  
    print(i, end=" ")  
    i = i+1
```

```
➞ 1 2 3 4 5 6 7 8 9 10 11 12
```

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

Source code:

```
▶ #Latihan Mandiri 5.1

def perkalian(bil1, bil2):
    hasil = 0
    print(f"{bil2} x {bil1} = ", end="")
    for i in range(bil2):
        hasil += bil1
        if i < bil2 - 1:
            print(f"{bil1} + ", end="")
        else:
            print(f"{bil1} = ", end="")
    return hasil

bil1 = int(input("Masukkan bilangan yang dikali : "))
bil2 = int(input("Masukkan bilangan perkalian : "))
print(perkalian(bil1, bil2))
```

Output:

```
➞ Masukkan bilangan yang dikali : 6
   Masukkan bilangan perkalian : 5
   5 x 6 = 6 + 6 + 6 + 6 + 6 = 30
```

```
➞ Masukkan bilangan yang dikali : 7
   Masukkan bilangan perkalian : 10
   10 x 7 = 7 + 7 + 7 + 7 + 7 + 7 + 7 + 7 + 7 + 7 = 70
```

Penjelasan: Kode tersebut adalah program untuk melakukan operasi perkalian dengan menggunakan iterasi.

- Fungsi **perkalian** menerima dua parameter, **bil1** dan **bil2**.
- Mencetak operasi perkalian ($\text{bil2} \times \text{bil1}$).
- Loop sebanyak **bil2** kali, menambahkan **bil1** ke hasil pada setiap iterasi.
- Selama iterasi, pesan output mencetak langkah-langkah perkalian.
- Hasil perkalian dikembalikan dari fungsi.
- Pengguna diminta untuk memasukkan dua bilangan.
- Hasil perkalian dicetak setelah memanggil fungsi dengan bilangan yang dimasukkan.

SOAL 2

Source code:

```
#Latihan Mandiri 5.2

def ganjil(bil1, bil2):
    if bil2 > bil1 :
        while bil2 != bil1 :
            if bil2 % 2 == 1 :
                print(bil2, end=",")
            bil2 = bil2 - 1
        else :
            while bil2 != bil1 :
                bil2 = bil2 + 1
                if bil2 % 2 == 1 :
                    print(bil2, end=",")

bawah = int(input("Masukkan nilai bawah : "))
atas = int(input("Masukkan nilai atas : "))
ganjil(atas, bawah)
```

Output:

11,13,15,17,19,21,23,25,27,29,



```
Masukkan nilai bawah : 97
Masukkan nilai atas : 82
97,95,93,91,89,87,85,83,
```

Penjelasan: Kode ini adalah program untuk menampilkan bilangan ganjil dalam rentang tertentu, dengan batas bawah dan batas atas yang diinput oleh pengguna.

- **Fungsi ganjil(bil1, bil2):**
Mendefinisikan fungsi ganjil dengan dua parameter, yaitu **bil1** dan **bil2**.
- **Pengecekan kondisi:**
Jika **bil2** lebih besar dari **bil1**, program akan menjalankan perulangan untuk mencetak bilangan ganjil dari **bil2** hingga **bil1**.
- **Perulangan while:**
Jika **bil2** tidak sama dengan **bil1**, periksa apakah **bil2** adalah bilangan ganjil. Jika iya, cetak nilai **bil2** dengan menambahkan tanda koma setelahnya.
- **Input dari Pengguna:**
Pengguna untuk memasukkan dua nilai, yaitu **bawah** dan **atas**.
- **Panggilan Fungsi:**
Memanggil fungsi ganjil dengan dua nilai yang dimasukkan oleh pengguna (**atas** dan **bawah**).

SOAL 3

Source Code:

```
#Latihan Mandiri 5.3

def ips(jumlah_matkul):
    total_sks = 0
    total_bobot = 0

    for i in range(jumlah_matkul):
        nilai = input(f"Nilai MK{i + 1}: ")

        if nilai == 'A':
            bobot = 4
        elif nilai == 'B':
            bobot = 3
        elif nilai == 'C':
            bobot = 2
        elif nilai == 'D':
            bobot = 1

        total_sks += 3
        total_bobot += bobot * 3

    ips = total_bobot / total_sks
    return ips

jumlah_matkul = int(input("Masukkan jumlah mata kuliah: "))
hasil = ips(jumlah_matkul)

print(f"Nilai IPS anda semester ini {0.2f}% hasil")
```

Output:

```
➞ Masukkan jumlah mata kuliah: 6
Nilai MK1: A
Nilai MK2: B
Nilai MK3: C
Nilai MK4: A
Nilai MK5: D
Nilai MK6: C
Nilai IPS anda semester ini 2.67
```

Penjelasan:

- **Fungsi ips(jumlah_matkul):**
Mendefinisikan fungsi **ips** dengan parameter **jumlah_matkul** yang merupakan jumlah mata kuliah yang akan dihitung IPS-nya. Program ini menggunakan loop **for** untuk iterasi sebanyak **jumlah_matkul**. Kemudian program meminta pengguna untuk menginput nilai setiap mata

kuliah(A,B,C,D). Percabangan **if-elif** digunakan untuk menentukan bobot nilai berdasarkan input pengguna. Kemudian menambahkan sks (diasumsikan selalu 3) dan bobot nilai ke total sks dan total bobot.

- **Input Jumlah Mata Kuliah:**
Meminta pengguna memasukkan jumlah mata kuliah.
- **Pemanggilan Fungsi:**
Memanggil fungsi ips dengan jumlah mata kuliah yang dimasukkan oleh pengguna.

Link Github:

<https://github.com/GalihPramana/Praktikum-Alpro-71230976.git>