

# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024



<b>NIM</b>	<b>71200645</b>
<b>Nama Lengkap</b>	<b>Edith Felicia Putri</b>
<b>Minggu ke / Materi</b>	<b>05 / Struktur Kontrol Perulangan</b>

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  
2023

## 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.

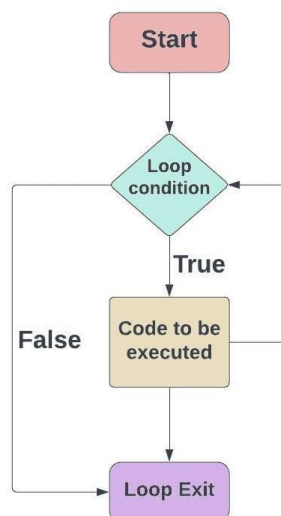
### MATERI 1

#### PERULANGAN

Program dapat berjalan secara sekuensial, percabangan, perulangan, maupun kombinasi dari ketiganya. Pengaturan tersebut biasa disebut sebagai struktur kontrol. Perulangan digunakan apabila dalam program diperlukan untuk:

1. Melakukan suatu hal yang sama beberapa kali.
2. Melakukan suatu hal secara bertahap, di mana setiap tahap sebenarnya memiliki langkah yang sama.
3. Mengakses sekumpulan data dalam suatu struktur data, misalnya: List, Tuple, Queue, Stack dan beberapa struktur data lainnya.
4. Untuk melakukan iterasi dari sequence seperti: list, tuple, dictionary, string, range atau objek yang dapat dilakukan iterasi lainnya.

#### Bentuk Perulangan For



<https://pythonguides.com/for-loop-vs-while-loop-in-python/>

Perulangan for biasanya digunakan pada kondisi:

- Jumlah perulangan sudah diketahui sejak awal. Misalnya akan dilakukan pembacaan data dari 10 file teks. Walaupun setiap file teks memiliki isi yang berbeda, tetapi membaca file

teks secara umum tetap sama. Pembacaan akan dilakukan dari file pertama, kedua, ketiga, dan seterusnya sampai file ke-sepuluh.

- Perulangan terjadi karena operasi yang sama pada suatu rentang data atau rentang nilai. Misalnya dalam mencari jumlah dari 100 bilangan pertama, maka secara berturut-turut dilakukan penjumlahan  $1 + 2 + 3 + \dots$  <berulang-ulang>  $+ 100$ . Berarti dilakukan dalam rentang mulai dari 1 sampai 100.

Perulangan for pada rentang tertentu lebih mudah dilakukan dengan menggunakan bantuan fungsi range(), yang bentuknya sebagai berikut:

1. range(stop). Digunakan untuk menghasilkan rentang dari 0 sampai stop-1. Misalnya range(6), berarti menghasilkan rentang 0-5.
2. range(start, stop, [step]). Digunakan untuk menghasilkan rentang dari start, sampai stop dengan peningkatan sejumlah step

#### SOURCE CODE:

```
for _ in range(1, 101):  
    print(_, "Hello World")
```

variabel \_ (temporary variable) digunakan sebagai counter, dimana nilai \_ akan naik secara berurutan sesuai dengan nilai yang dihasilkan dari fungsi range() tersebut.

#### OUTPUT:

```
92 Hello World  
93 Hello World  
94 Hello World  
95 Hello World  
96 Hello World  
97 Hello World  
98 Hello World  
99 Hello World  
100 Hello World  
PS D:\71200645_PrakAlpro_5>
```

#### SOURCE CODE:

```
kalimat = "Konversi yang sama"  
panjang = len(kalimat)  
for i in range(1, panjang):  
    print(i, kalimat)
```

## OUTPUT:

```
grams/Python/Python311/python.exe d:/71200645_PrakAlpro_5/m
odul.py
1 Konversi yang sama
2 Konversi yang sama
3 Konversi yang sama
4 Konversi yang sama
5 Konversi yang sama
6 Konversi yang sama
7 Konversi yang sama
8 Konversi yang sama
9 Konversi yang sama
10 Konversi yang sama
11 Konversi yang sama
12 Konversi yang sama
13 Konversi yang sama
14 Konversi yang sama
15 Konversi yang sama
16 Konversi yang sama
17 Konversi yang sama
PS D:\71200645_PrakAlpro_5> □
```

## Step Negatif

```
for i in range(8, 101, 5):
    print(i)
```

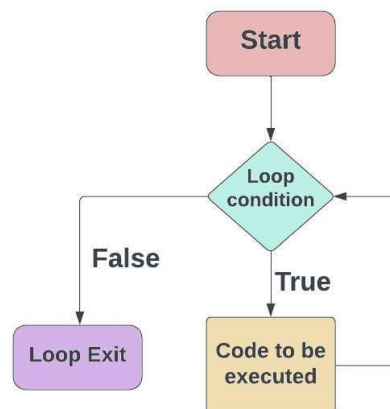
Perulangan dilakukan pada rentang 8 – 100 dengan langkah 5.

Fungsi range() yang menerima step negatif, sebagai berikut:

```
for i in range(50, 1, -2):
    print(i)
```

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

## Bentuk Perulangan While



Bentuk while biasanya digunakan pada kondisi di mana jumlah perulangan belum diketahui sebelumnya.

### SOURCE CODE:

```
ganjil = False
while ganjil == False:
    bilangan = int(input("Masukkan bilangan ganjil: "))
    if bilangan % 2 != 0:
        ganjil = True
print(bilangan, "yang anda masukkan adalah bilangan ganjil")
```

Pengguna awalnya memasukkan bilangan genap, tetapi program terus meminta pengguna memasukkan bilangan ganjil. Program berhenti setelah pengguna memasukkan 7, yang merupakan bilangan ganjil. Perulangan while sesuai digunakan karena tidak diketahui sampai berapa kali pengguna memasukkan bilangan genap yang tidak sesuai dengan permintaan.

### OUTPUT:

```
PS D:\71200645_PrakAlpro_5> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_5/modul.py
Masukkan bilangan ganjil: 2
Masukkan bilangan ganjil: 6
Masukkan bilangan ganjil: 8
Masukkan bilangan ganjil: 7
7 yang anda masukkan adalah bilangan ganjil
PS D:\71200645_PrakAlpro_5> □
```

### SOURCE CODE:

```
nilai = [10, 99, 98, 85, 45, 59, 65, 66, 76, 12, 35, 13, 100, 80, 95]
my_message=""

length = len(nilai)

i = 0
while i < length:
    if nilai[i] == 100:
        my_message = f"There is a 100 at index no: {i}"
    i += 1
print(my_message)
```

### OUTPUT:

```
PS D:\71200645_PrakAlpro_5> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/71200645_PrakAlpro_5/modul.py
There is a 100 at index no: 12
PS D:\71200645_PrakAlpro_5>
```

## MATERI 2

### PENGUNAAN BREAK DAN CONTINUE

Perulangan dapat dikontrol dengan menggunakan break dan continue. Secara umum break digunakan untuk menghentikan perulangan, sedangkan continue digunakan untuk melanjutkan perulangan ke iterasi berikutnya

#### SOURCE CODE:

```
for i in range(4, 24):  
    if i == 19:  
        break  
    else:  
        print(f"Halooo, no.{i}")  
print("Finish!")
```

#### OUTPUT:

```
PS D:\71200645_PrakAlpro_5> & C:/Users/HP/AppData/Local/Programs/  
Python/Python311/python.exe d:/71200645_PrakAlpro_5/modul.py  
Halooo, no.4  
Halooo, no.5  
Halooo, no.6  
Halooo, no.7  
Halooo, no.8  
Halooo, no.9  
Halooo, no.10  
Halooo, no.11  
Halooo, no.12  
Halooo, no.13  
Halooo, no.14  
Halooo, no.15  
Halooo, no.16  
Halooo, no.17  
Halooo, no.18  
Finish!
```

## MATERI 3

### KONVERSI DARI BENTUK FOR MENJADI BENTUK WHILE

Bentuk perulangan for sebagian besar dapat dikonversi menjadi bentuk while. Beberapa hal yang ada di bentuk for dan while adalah sebagai berikut:

1. Harus ada nilai awal, untuk memulai perulangan.
2. Harus ada nilai akhir, untuk mengakhiri perulangan.
3. Harus ada langkah, agar iterasi dari awal bisa terus berjalan sampai mencapai nilai akhir.

## Penggunaan Break

```
def average():
    total = 0
    count = 0
    while True:
        input_user = int(input("Masukkan nilai (nol atau negatif untuk berhenti): "))
        if input_user < 1:
            break
        else:
            total = total + input_user
            count = count + 1
    if count > 0:
        return total / count
    else:
        return 0

hasil = average()
print("Rata-rata: ", hasil)
```

## 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

**Latihan 5.1** Buatlah program yang menerapkan perhitungan perkalian dengan menggunakan penjumlahan. Buatlah fungsi perkalian() dalam program tersebut! Berikut ini adalah beberapa contoh perhitungan yang diharapkan:

- $6 \times 5 = 5 + 5 + 5 + 5 + 5 + 5 = 30$ .
- $7 \times 10 = 10 + 10 + 10 + 10 + 10 + 10 + 10 = 70$ .

Pengerjaan soal menggunakan dua nilai inputan user, lalu dibuat for loop dalam range inputan pertama, karena input\_b akan dibuat sebanyak input\_a.

### SOURCE CODE:

```
def perkalian():
    input_a = int(input("Masukkan nilai a:"))
    input_b = int(input("Masukkan nilai b:"))

    for i in range (input_a):
        if i == 0:
```

```

        print(f"{input_a} x {input_b} = {input_b} +", end=" ")
    elif i != input_a - 1:
        print(f"{input_b}", end=" + ")
    else:
        print(f"{input_b} =", input_a * input_b)

```

perkalian()

## OUTPUT:

```

Latihan 5.1.py"
Masukkan nilai a:6
Masukkan nilai b:5
6 x 5 = 5 + 5 + 5 + 5 + 5 + 5 = 30
PS D:\71200645_PrakAlpro_5> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "d:/71200645_PrakAlpro_5/Latihan 5.1.py"
Masukkan nilai a:7
Masukkan nilai b:10
7 x 10 = 10 + 10 + 10 + 10 + 10 + 10 + 10 = 70
PS D:\71200645_PrakAlpro_5> 

```

## SOAL 2

**Latihan 5.2** Buatlah program yang dapat menampilkan deret bilangan ganjil dari batas bawah dan batas atas yang diberikan oleh pengguna. Jika ternyata batas atas < batas bawah, berarti deret tersebut dimulai dari batas atas, sampai batas bawah (negatif range). Buatlah fungsi ganjil() dalam program tersebut! Berikut ini adalah contoh hasil yang diharapkan:

- bawah = 10, atas = 30. Karena bawah < atas, berarti dari kecil ke besar, maka hasilnya adalah: 11, 13, 15, 17, 19, 21, 23, 25, 27, 29.
- bawah = 97, atas = 82. Karena bawah > atas, berarti dari besar ke kecil, maka hasilnya adalah: 97, 95, 93, 91, 89, 87, 85, 83.

Pengerjaan soal untuk menampilkan urutan bilangan ganjil (angka yang tidak habis dibagi / != 0) dari batas bawah dan batas atas yang diberikan pengguna.

Menggunakan percabangan *if-else* untuk aturan nilai batas bawah dan batas atas yang lebih besar atau lebih kecil di keduanya. Batas bawah yang lebih besar dari batas atas, akan tampil berurutan dari angka terbesar (bawah) ke angka terkecil (atas) dalam range for loop. Begitu juga dengan batas atas yang lebih besar dari batas bawah, akan tampil berurutan dari angka terbesar (atas) ke angka terkecil (bawah). Dilakukan if-else bilangan ganjil di dalam for loop (selama perulangan).

## SOURCE CODE:

```

# Deret bilangan ganjil dari batas bawah dan batas atas pengguna.
# batas atas < batas bawah => Dimulai dari batas atas

def ganjil():
    bawah = int(input("Masukkan bilangan batas bawah: "))
    atas = int(input("Masukkan bilangan batas atas: "))

```



```

    if atas < bawah:
        for i in range(bawah, atas - 1, -1):
            if i % 2 != 0:
                print(i,end="." if atas+1 == i or atas == i else ", ")

    elif atas > bawah:
        for i in range(bawah, atas + 1):
            if i % 2 != 0:
                print(i,end="." if atas-1 == i or atas == i else ", ")

ganjil()

```

## OUTPUT:

```

Latihan 5.2.py"
Masukkan bilangan batas bawah: 10
Masukkan bilangan batas atas: 30
11, 13, 15, 17, 19, 21, 23, 25, 27, 29.
PS D:\71200645_PrakAlpro_5> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe "d:/71200645_PrakAlpro_5/Latihan 5.2.py"
Masukkan bilangan batas bawah: 97
Masukkan bilangan batas atas: 82
97, 95, 93, 91, 89, 87, 85, 83.
PS D:\71200645_PrakAlpro_5>

```

## SOAL 3

**Latihan 5.3** Buatlah sebuah program penghitung nilai Indeks Prestasi Semester (IPS). Input bagi program:

- Jumlah mata kuliah
- Nilai A, B, C, dan D untuk setiap mata kuliah mahasiswa. Diasumsikan sks setiap mata kuliah selalu 3. Kemudian bobot dari masing-masing nilai adalah: A=4, B=3, C=2, D=1.

Output program ialah hasil IPS yang didapatkan. Jalannya program seperti pada Gambar 5.7. Tips: Gunakan kontrol percabangan di dalam perulangan.

Pengerjaan soal dengan inputan jumlah dan nilai abjad oleh pengguna. Karena akan menanyakan nilai berulang kali sebanyak input angka dalam variabel jumlah, maka inputan akan dimasukkan dalam for loop. Untuk menerima inputan pengguna tetap abjad dengan huruf besar digunakan fungsi upper(). Nilai-nilai dari setiap abjad dalam bentuk *if-else* berada di dalam perulangan, untuk mengolah inputan pengguna sekaligus menjumlahkan total nilai. Total SKS adalah jumlah mata kuliah yang diambil dikali 3 (SKS setiap mata kuliah). Lalu nilai IPS didapatkan dengan pembagian total nilai dengan total SKS. Fungsi round() dilakukan untuk mendapat nilai dua angka di belakang koma.

## SOURCE CODE:

```

def penghitung_IPS():
    jumlah = int(input("Berapa jumlah mata kuliah? "))
    total_sks = jumlah * 3

```

```

total_nilai = 0

for i in range(jumlah):
    nilai = input(f"Nilai MK {i+1}: ").upper()
    if nilai == "A":
        total_nilai += 12
    elif nilai == "B":
        total_nilai += 9
    elif nilai == "C":
        total_nilai += 6
    elif nilai == "D":
        total_nilai += 3
    else:
        print("Nilai hanya bisa diinput A, B, C, atau D.")

print("Nilai IPS Anda semester ini: ", round(total_nilai / total_sks,
2))

penghitung_IPS()

```

## OUTPUT:

```

grams/Python/Python311/python.exe "d:/71200645_PrakAlpro_5/
Latihan 5.3.py"
Berapa jumlah mata kuliah? 6
Nilai MK 1: A
Nilai MK 2: B
Nilai MK 3: C
Nilai MK 4: A
Nilai MK 5: D
Nilai MK 6: C
Nilai IPS Anda semester ini: 2.67
PS D:\71200645_PrakAlpro_5>

```

## Link Github:

[https://github.com/EdithFelicia/71200645\\_Guided/tree/main/71200645\\_PrakAlpro\\_5](https://github.com/EdithFelicia/71200645_Guided/tree/main/71200645_PrakAlpro_5)