



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>&lt;71230989&gt;</b>
<b>Nama Lengkap</b>	<b>&lt;Yohanes Nevan Adventus Wibawa&gt;</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  
2024

## BAGIAN 1: Struktur Kontrol Perulangan (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

Suatu program memiliki kemampuan untuk mengatur jalannya dengan berbagai metode, seperti sekuensial, percabangan, perulangan, atau kombinasi dari ketiganya. Pengaturan ini dikenal sebagai struktur kontrol. Perulangan digunakan ketika program memerlukan:

- Melakukan suatu tindakan secara berulang beberapa kali.
- Melakukan suatu tindakan secara bertahap, di mana setiap tahap memiliki langkah yang sama.
- Mengakses sekumpulan data dalam struktur data seperti List, Tuple, Queue, Stack, dan struktur data lainnya.

Dalam bahasa pemrograman Python, perulangan dapat diimplementasikan menggunakan pernyataan for, while, atau dengan pendekatan rekursif. Pada kesempatan ini, kita akan membahas perulangan menggunakan for dan while.

### Bentuk Perulangan For

Pada Python, perulangan dapat dinyatakan dalam bentuk for dan while. 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:
- range(stop). Digunakan untuk menghasilkan rentang dari 0 sampai stop-1. Misalnya range(6), berarti menghasilkan rentang 0-5.
- range(start, stop, [step]). Digunakan untuk menghasilkan rentang dari start, sampai stop dengan peningkatan sejumlah step. Berikut ini adalah contoh program untuk menampilkan bilangan dari 1 sampai 100 dengan menggunakan for dan range():

```
for i in range(1, 101):  
    print(i)
```

Dalam program tersebut, terdapat perulangan for yang menggunakan fungsi range(), dimulai dari 1 (start) hingga 101 (stop-1) dengan langkah 1 (default step adalah 1). Variabel i digunakan sebagai counter, di mana nilai i akan bertambah secara berurutan sesuai dengan nilai yang dihasilkan oleh fungsi range(). Namun, jika counter tidak diperlukan, perulangan dapat diimplementasikan tanpa menggunakan variabel i, seperti contoh berikut:

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

Program tersebut akan menampilkan teks "Hello World" sebanyak 100 kali tanpa memerlukan nilai dari suatu counter.

Step Negatif, perulangan for berikut ini yang akan menampilkan seluruh bilangan genap dari 2 sampai 100:

```
for i in range(2, 101, 2):  
    print(i)
```

Jika diperlukan untuk menampilkan bilangan genap dari 100 sampai 2, Anda dapat menggunakan fungsi range() dengan langkah (step) yang bersifat negatif. Contoh penggunaannya adalah sebagai berikut:

```
for i in range(100, 1, -2):  
    print(i)
```

Program tersebut akan menampilkan bilangan genap mulai dari 100, 98, 96, 94, ..., sampai 2.

## Bentuk Perulangan While

```
bilangan = 0  
genap = False  
while genap == False:  
    bilangan = int(input("Masukan bilangan genap= "))  
    if bilangan % 2 == 0:  
        genap = True  
print(bilangan, "yang anda masukan adalah bilangan genap")
```

Program tersebut menggunakan perulangan `while` untuk memastikan bahwa pengguna memasukkan bilangan genap. Jika pengguna memasukkan bilangan ganjil, program akan terus meminta input hingga bilangan genap dimasukkan. Setelah itu, program akan mencetak bilangan genap yang dimasukkan oleh pengguna. Perulangan `while` digunakan untuk menjalankan blok kode berulang kali selama kondisi yang ditentukan (`genap == False`) masih terpenuhi.

Perulangan while digunakan untuk terus melakukan eksekusi blok pernyataan selama kondisi yang diberikan masih bernilai True. Pada contoh ini, perulangan while digunakan untuk memastikan bahwa pengguna memasukkan bilangan genap sebelum program melanjutkan ke langkah selanjutnya.

## Penggunaan Break dan Continue

Kontrol pada perulangan dapat dilakukan menggunakan pernyataan `break` dan `continue`. Pada umumnya, `break` digunakan untuk menghentikan perulangan secara keseluruhan, sementara `continue` digunakan untuk melanjutkan ke iterasi berikutnya. Program di bawah ini menunjukkan contoh penggunaan kedua pernyataan tersebut untuk menampilkan bilangan dari 1 sampai 10:

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

Ketika nilai *i* mencapai 5, evaluasi dari boolean expression *i*==5 menjadi True, yang mengakibatkan program menjalankan pernyataan break. Akibatnya, perulangan dihentikan, dan program melanjutkan eksekusi ke baris setelah perulangan, yaitu menampilkan tulisan 'Selesai'. Perbedaan antara break dan continue terletak pada pengaruhnya terhadap perulangan: break: Digunakan untuk menghentikan perulangan secara keseluruhan ketika suatu kondisi terpenuhi. Setelah break dijalankan, program keluar dari perulangan dan melanjutkan eksekusi baris kode setelah perulangan.

```
PS C:\Users\LENOVO\Tugas PrakAlPro\Tugas 5> & "C:/Program Files/Python312/python.exe" "c:/Users/LENOVO/Tugas  
PrakAlPro/Tugas 5/coba.py"  
1  
2  
3  
4  
Selesai
```

continue: Digunakan untuk melanjutkan ke iterasi berikutnya dalam perulangan saat suatu kondisi terpenuhi. Jika continue dijalankan, program akan melanjutkan ke iterasi berikutnya tanpa mengeksekusi kode yang ada di bawahnya dalam iterasi saat itu. Contoh program yang tidak menampilkan angka 6:

```
for i in range(1, 11):  
    if i == 6:  
        continue  
    else:  
        print(i)  
print("Selesai")
```

Output yang dihasilkan dari contoh penggunaan continue dapat dilihat pada Gambar 5.2. Program menampilkan angka dari 1 sampai 10, tetapi melewati angka 6.

```
PS C:\Users\LENOVO\Tugas PrakAlPro\Tugas 5> & "C:/Program Files/Python312/python.exe" "c:/Users/LENOVO/Tugas  
PrakAlPro/Tugas 5/coba.py"  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
Selesai
```

Melewatkan angka 6

## Deret Bilangan

Program untuk menampilkan deret bilangan fibonacci mulai dari 1 sampai batas tertentu yang dimasukkan oleh pengguna! Deret bilangan fibonacci adalah deret bilangan yang tersusun dari penjumlahan dua suku sebelumnya dari deret bilangan tersebut.

Implementasi dari deret fibonacci lebih baik jika dipisahkan dalam bentuk fungsi fibo() yang menerima parameter batas. Langkah-langkah dari implementasi tersebut adalah:

1. Minta nilai batas dari pengguna.
  2. Panggil fungsi fibo() untuk menampilkan deret fibonacci dengan menggunakan perulangan while.
- Program untuk menjawab permasalahan tersebut adalah sebagai berikut:

```
def fibo(batas):
    bil1 =1
    bil2 =2

    if bil1 > bil2:
        print(bil1, end='\t')
        print(bil1, end='\t')
    suku_baru = bil1 + bil2
    while suku_baru < batas:
        print(suku_baru, end='\t')

        bil1 = bil2
        bil2 = suku_baru
        suku_baru = bil1 + bil2
batas = int(input('Masukan batas dari deret fibonacci: '))
fibo(batas)
```

Program yang dapat menampilkan deret bilangan konvergen, yang diawali dari input bilangan dari pengguna. Suku-suku dari deret tersebut berikutnya didapatkan dengan cara berikut ini:

- Jika ganjil, maka kalikan dengan tiga, lalu tambah 1.
- Jika genap, bagi dengan 2. Tampilkan suku-suku dari deret bilangan konvergen tersebut yang berakhir jika mencapai nilai 1

```
def konvergen(start):
    suku = start
    while suku != 1:
        print(suku)
        if suku % 2 == 0:
            suku = suku // 2
        else:
            suku = suku * 3 + 1
start = int(input('Masukan suku pertama dari deret konvergen: '))
konvergen(start)
```

```
PrakAlPro/Tugas 5/konvergen.py"
Masukan suku pertama dari deret konvergen: 12
12
6
3
10
5
16
8
4
2
```

### Penggunaan Break

Buatlah program yang dapat menghitung rata-rata dari sejumlah input yang diberikan oleh pengguna. Program akan terus meminta input dari pengguna, sampai pengguna memasukkan bilangan negatif atau nol. Program kemudian menampilkan rata-rata dari keseluruhan input (abaikan input negatif atau nol).

Untuk menyelesaikan situasi di mana program meminta input pengguna terus-menerus hingga pengguna memasukkan nilai negatif atau nol, digunakan perulangan dalam bentuk `while`. Karena tidak diketahui sebelumnya berapa kali pengguna akan memasukkan input, `while` sangat cocok untuk kasus ini. Untuk menghentikan permintaan input, digunakan pernyataan `break` ketika kondisi input menjadi negatif atau nol. Berikut adalah solusi untuk permasalahan ini:

```
def avarage():
    total = 0
    count = 0
    while True:
        input_user = int(input('Masukan 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 = avarage()
print('Rata-rata: ', hasil)
```

Dan outputnya akan seperti dibawah ini

## 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 5.1

#### A. Source Code

```
def perkalian_dengan_penjumlahan():
    angka1 = int(input("Masukkan angka pertama: "))
    angka2 = int(input("Masukkan angka kedua: "))

    hasil_perkalian = 0
    for _ in range(angka1):
        hasil_perkalian += angka2

    print(f"{angka1} x {angka2} =", end=" ")
    for i in range(angka1):
        print(angka2, end="")
        if i < angka1 - 1:
            print(" + ", end="")
        else:
            print(f" = {hasil_perkalian}")

perkalian_dengan_penjumlahan()
```

#### B. Output

```
perkalian_dengan_penjumlahan()
✓ 3.6s

6 x 5 = 5 + 5 + 5 + 5 + 5 + 5 = 30
```

#### C. Penjelasan

Kode tersebut adalah program Python sederhana yang melakukan perkalian dua bilangan menggunakan operasi penjumlahan berulang. Berikut adalah penjelasan langkah demi langkah: Program meminta pengguna memasukkan dua bilangan secara berturut-turut. Variabel `hasil_perkalian` diinisialisasi dengan nilai 0. Program menggunakan loop `for` untuk menjalankan penjumlahan berulang sebanyak `angka1` kali. Pada setiap iterasi, nilai `angka2` ditambahkan ke `hasil_perkalian`. Setelah hasil perkalian dihitung, program menggunakan loop `for` kedua untuk menampilkan langkah-langkah perkalian. Setiap nilai `angka2` dicetak, dan di antaranya ditambahkan tanda tambah. Loop ini digunakan untuk merepresentasikan operasi penjumlahan yang dilakukan dalam perkalian. Hasil akhir perkalian dan langkah-langkahnya ditampilkan dalam bentuk kalimat,

## SOAL 5.2

### A. Source Code

```
def ganjil(bawah, atas):  
    if bawah < atas:  
        for i in range(bawah, atas + 1):  
            if i % 2 != 0:  
                print(i, end=' ')  
    else:  
        for i in range(bawah, atas - 1, -1):  
            if i % 2 != 0:  
                print(i, end=' ')  
  
bawah = int(input("Masukkan batas bawah: "))  
atas = int(input("Masukkan batas atas: "))  
  
print("Deret bilangan ganjil:")  
ganjil(bawah, atas)
```

### B. Output

```
ganjil(bawah, atas)  
  
✓ 3.1s  
  
Deret bilangan ganjil:  
11 13 15 17 19 21 23 25 27 29
```

### A. Penjelasan

Kode tersebut adalah sebuah fungsi Python yang disebut ganjil untuk mencetak deret bilangan ganjil dalam rentang tertentu. Berikut penjelasan langkah-demi-langkahnya: Fungsi ganjil didefinisikan dengan dua parameter, yaitu bawah dan atas. Fungsi ini bertujuan untuk mencetak deret bilangan ganjil dalam rentang dari bawah hingga atas. Program memeriksa apakah bawah lebih kecil dari atas. Jika ya, itu berarti rentang deret adalah naik (dari yang lebih kecil ke yang lebih besar). Jika tidak, itu berarti rentang deret adalah turun (dari yang lebih besar ke yang lebih kecil). Dalam loop for, program mencetak bilangan ganjil dari bawah hingga atas + 1 jika rentang naik, dan dari bawah hingga atas - 1 dengan langkah mundur jika rentang turun. Bilangan ganjil dikenali dengan sisa pembagian ( $i \% 2 \neq 0$ ). Hasilnya dicetak dalam satu baris dengan menggunakan `end=' '` sehingga bilangan ganjil tercetak dalam satu baris. Pengguna diminta untuk memasukkan dua nilai, yaitu bawah dan atas, dan kemudian fungsi ganjil dipanggil dengan nilai-nilai tersebut.



### Soal 5.3

#### A. Source Code

```
def hitung_ips(jumlah_mata_kuliah):
    total_sks = jumlah_mata_kuliah * 3
    total_bobot = 0

    for i in range(1, jumlah_mata_kuliah + 1):
        nilai = input(f'Nilai MK {i} : ')

        if nilai == 'A':
            bobot = 4
        elif nilai == 'B':
            bobot = 3
        elif nilai == 'C':
            bobot = 2
        elif nilai == 'D':
            bobot = 1
        else:
            print('Nilai tidak valid. Mohon input A, B, C, atau D.')
            return

        total_bobot += bobot * 3

    ips = total_bobot / total_sks
    return ips

jumlah_mata_kuliah = int(input('Berapa jumlah mata kuliah? '))

hasil_ips = hitung_ips(jumlah_mata_kuliah)

print(f'Nilai IPS Anda semester ini: {hasil_ips:.2f}')
```

#### B. Output

```
PS C:\Users\LENOVO\Tugas PrakAlPro\Tugas 5> & "
PrakAlPro/Tugas 5/coba.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
```

### C. Penjelasan

Kode tersebut adalah program Python untuk menghitung Indeks Prestasi Semester (IPS) berdasarkan nilai mata kuliah yang dimasukkan oleh pengguna. Berikut adalah penjelasan langkah-demi-langkahnya: Fungsi `hitung_ips` didefinisikan dengan satu parameter, yaitu `jumlah_mata_kuliah`. Fungsi ini menghitung IPS berdasarkan nilai yang dimasukkan pengguna. Variabel `total_sks` dihitung sebagai perkalian `jumlah_mata_kuliah` dengan 3, karena asumsinya setiap mata kuliah memiliki bobot 3 SKS. Variabel `total_bobot` diinisialisasi dengan nilai 0. Ini akan digunakan untuk mengakumulasi bobot nilai setiap mata kuliah. Program menggunakan loop `for` untuk mengiterasi sebanyak `jumlah_mata_kuliah`. Pada setiap iterasi, pengguna diminta memasukkan nilai mata kuliah (A, B, C, atau D). Untuk setiap nilai, program menentukan bobotnya (4 untuk A, 3 untuk B, 2 untuk C, dan 1 untuk D). Bobot nilai kemudian dikalikan dengan 3 (bobot SKS), dan hasilnya ditambahkan ke `total_bobot`. Jika pengguna memasukkan nilai selain A, B, C, atau D, program memberikan pesan kesalahan dan keluar dari fungsi. Setelah semua nilai dimasukkan, IPS dihitung sebagai hasil pembagian `total_bobot` dengan `total_sks`. Fungsi mengembalikan nilai IPS. Pengguna diminta untuk memasukkan jumlah mata kuliah, dan fungsi `hitung_ips` dipanggil dengan nilai tersebut. Hasil IPS kemudian dicetak dengan dua desimal.

### Link GitHub

Link = <https://github.com/YohanesNevan/Tugas-Laporan-AIPro-5.git>