



# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

<b>NIM</b>	<b>7120876</b>
<b>Nama Lengkap</b>	<b>DEVINA ELISSE PUTRI</b>
<b>Minggu ke / Materi</b>	<b>05 / 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: 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

Jalannya program bisa diatur secara sekuensial, percabangan, perulangan, atau kombinasi ketiganya, yang disebut struktur kontrol. Perulangan digunakan untuk melakukan hal-hal seperti: melakukan suatu tindakan beberapa kali, melakukan tindakan secara bertahap dengan langkah-langkah yang sama, atau mengakses data dalam struktur data seperti List, Tuple, Queue, Stack, dan lainnya. Di Python, perulangan bisa dilakukan menggunakan for, while, atau secara rekursif. Pada kesempatan ini, kita akan fokus pada perulangan for dan while.

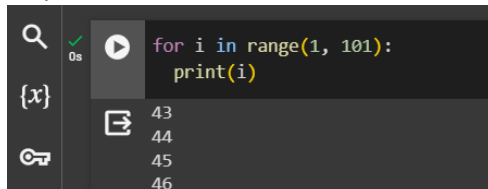
### MATERI 2

Pada Python, perulangan for biasanya digunakan ketika:

- Jumlah perulangan sudah diketahui sejak awal, misalnya untuk membaca data dari 10 file teks.
- Perulangan terjadi karena operasi yang sama pada rentang data atau nilai tertentu, misalnya dalam mencari jumlah dari 100 bilangan pertama.

Perulangan for pada rentang tertentu lebih mudah dilakukan dengan menggunakan fungsi range(), yang memiliki bentuk:

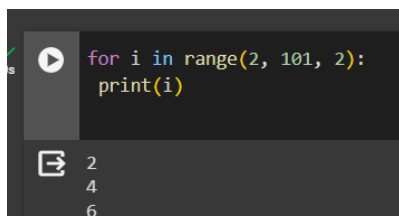
- range(stop): Menghasilkan rentang dari 0 sampai stop-1.
- range(start, stop, [step]): Menghasilkan rentang dari start sampai stop dengan peningkatan sejumlah step.



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

43  
44  
45  
46

Pada program tersebut, terdapat perulangan for dengan menggunakan range() mulai dari 1 hingga 101 dengan langkah 1. Variabel i digunakan sebagai counter yang nilainya naik secara berurutan sesuai dengan nilai dari fungsi range().



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

2  
4  
6

Perulangan dilakukan pada rentang 2-100 dengan langkah 2, menghasilkan rentang 2, 4, 6, 8, 10, 12, ..., 100. Jika perlu menampilkan bilangan genap dari 100 sampai 2, fungsi range() dapat menerima step negatif.

## MATERI 3

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

```
Masukkan bilangan genap: 13
Masukkan bilangan genap: 14
14 yang anda masukkan adalah bilangan genap
```

Program berhenti setelah pengguna memasukkan bilangan genap , yang sebelumnya memasukkan bilangan ganjil. Kasus ini cocok untuk menggunakan perulangan while, karena tidak diketahui berapa kali pengguna akan memasukkan bilangan ganjil yang tidak sesuai permintaan.

## MATERI 4

Perulangan dapat dikontrol dengan break untuk menghentikan perulangan dan continue untuk melanjutkan ke iterasi berikutnya. Berikut adalah program untuk menampilkan bilangan dari 1 sampai 10:

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

```
1
2
3
4
Selesai
```

```
[9] for i in range(1, 11):
    print(i)
print('Selesai')
```

```
1
2
3
4
5
6
7
8
9
10
Selesai
```

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

```
1
2
3
4
5
7
8
9
10
Selesai
```

## MATERI 5

Perulangan for sebagian besar dapat dikonversi menjadi perulangan while dengan syarat berikut:

- Memiliki nilai awal untuk memulai perulangan.
- Memiliki nilai akhir untuk menghentikan perulangan.
- Memiliki langkah agar iterasi nilai awal dapat terus berjalan hingga mencapai nilai akhir.

```
i = 1 #awal
while i <= 10: #kondisi akhir
    print(i)
    i = i + 1 #step
```

1  
...  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1

## KEGIATAN PRAKTIKUM 1

```
def fibo(batas):
    bil1 = 1
    bil2 = 1
    # tampilkan dua suku fibonacci pertama
    if bil1 < batas:
        print(bil1, end='\t')
        print(bil2, end='\t')
    # suku-suku berikutnya dari bil1 + bil2
    suku_baru = bil1 + bil2
    while suku_baru < batas:
        print(suku_baru, end='\t')
    # geser bil1 dan bil2
    bil1 = bil2
    bil2 = suku_baru
    # hitung lagi suku berikutnya
    suku_baru = bil1 + bil2
    # program utama
    batas = int(input('Masukkan batas dari deret fibonacci: '))
    fibo(batas)
```

.. Masukkan batas dari deret fibonacci: 12  
1      1      2      2      2      2      2      2

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

Masukkan suku pertama dari deret konvergen: 12
12
6
3
10
5
16
8
4
2

def average():
    total = 0
    count = 0
    while True:
        input_user = int(input('Masukkan nilai (nol atau negatif untuk berhenti): '))
        if input_user < 1: # negatif atau nol
            break
        else:
            total = total + input_user
            count = count + 1
    if count > 0:
        return total / count
    else:
        return 0
    # bagian utama program
    hasil = average()
    print('Rata-rata: ', hasil)

Masukkan nilai (nol atau negatif untuk berhenti): 18
Masukkan nilai (nol atau negatif untuk berhenti): 12
Masukkan nilai (nol atau negatif untuk berhenti): 30
Masukkan nilai (nol atau negatif untuk berhenti): -3
Rata-rata: 20.0
```

## 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

```
def perkalian(a, b):  
    hasil = 0  
    for _ in range(b):  
        hasil += a  
    return hasil  
  
# Contoh penggunaan fungsi perkalian()  
angka1 = 6  
angka2 = 5  
hasil1 = perkalian(angka1, angka2)  
print(f"{angka1} x {angka2} = {hasil1}")  
  
angka3 = 7  
angka4 = 10  
hasil2 = perkalian(angka3, angka4)  
print(f"{angka3} x {angka4} = {hasil2}")
```

6 x 5 = 30  
7 x 10 = 70

Program di atas mendefinisikan sebuah fungsi bernama `perkalian(a, b)` yang menghitung hasil perkalian dari dua bilangan `a` dan `b` dengan menggunakan penjumlahan. Variabel `hasil` diinisialisasi dengan nilai 0. Selanjutnya, dilakukan perulangan sebanyak `b` kali menggunakan fungsi `range(b)`, di mana setiap iterasi akan menambahkan nilai `a` ke dalam variabel `hasil`. Setelah perulangan selesai, nilai `hasil` yang merupakan hasil perkalian dikembalikan menggunakan pernyataan `return`.

Pada bagian berikutnya, dilakukan dua contoh penggunaan fungsi `perkalian()`. Pertama, `angka1` dan `angka2` diinisialisasi dengan nilai 6 dan 5, dan hasil perkaliannya dicetak. Kedua, `angka3` dan `angka4` diinisialisasi dengan nilai 7 dan 10, dan hasil perkaliannya juga dicetak.

Hasil dari kedua perkalian dicetak ke layar dalam format yang sesuai dengan contoh yang diberikan.

## SOAL 2

```
def ganjil(bawah, atas):
    deret_ganjil = []
    if bawah < atas:
        for i in range(bawah + 1, atas):
            if i % 2 != 0:
                deret_ganjil.append(i)
    else:
        for i in range(bawah - 1, atas, -1):
            if i % 2 != 0:
                deret_ganjil.append(i)
    return deret_ganjil

# Input dari pengguna
bawah = int(input("Masukkan batas bawah: "))
atas = int(input("Masukkan batas atas: "))

# Panggil fungsi ganjil()
hasil = ganjil(bawah, atas)

# Cetak hasil
if bawah < atas:
    print(f"Deret ganjil dari {bawah} sampai {atas}: {hasil}")
else:
    print(f"Deret ganjil dari {atas} sampai {bawah}: {hasil}")
```

Masukkan batas bawah: 100  
Masukkan batas atas: 20  
Deret ganjil dari 20 sampai 100: [99, 97, 95, 93, 91, 89, 87, 85, 83, 81, 79, 77, 75, 73, 71, 69, 67, 65, 63, 61, 59, 57, 55, 53, 51, 49, 47, 45, 43, 41, 39, 37, 35, 33, 31, 29, 27, 25, 23, 21, 19, 17, 15, 13, 11, 9, 7, 5, 3, 1]

### 1. Fungsi ganjil(bawah, atas):

- Fungsi ini menerima dua parameter, yaitu bawah dan atas, yang merupakan batas bawah dan batas atas dari deret ganjil yang akan dihasilkan.
- Sebuah list kosong `deret_ganjil` dibuat untuk menyimpan deret bilangan ganjil.
- Jika bawah kurang dari atas, maka dilakukan perulangan `for` dari bawah + 1 sampai atas - 1.
- Di dalam perulangan, jika nilai `i` adalah bilangan ganjil (diperiksa dengan `i % 2 != 0`), maka nilai `i` akan ditambahkan ke dalam list `deret_ganjil`.
- Jika bawah lebih besar dari atas, maka dilakukan perulangan `for` dari bawah - 1 sampai atas + 1 dengan langkah mundur -1.
- Di dalam perulangan, juga diperiksa apakah nilai `i` adalah bilangan ganjil, dan jika ya, nilai `i` ditambahkan ke dalam list `deret_ganjil`.
- Setelah perulangan selesai, fungsi mengembalikan list `deret_ganjil` yang berisi deret bilangan ganjil.

### 2. Input dari pengguna:

- Pengguna diminta untuk memasukkan dua nilai, yaitu bawah dan atas, yang akan menjadi batas bawah dan batas atas deret ganjil yang dihasilkan.

### 3. Panggil fungsi ganjil():

- Fungsi `ganjil()` kemudian dipanggil dengan dua nilai yang dimasukkan pengguna (bawah dan atas).

### 4. Cetak hasil:

- Hasil dari fungsi `ganjil()` akan disimpan dalam variabel `hasil`.
- Kemudian, hasil akan dicetak ke layar. Jika bawah kurang dari atas, maka deret ganjil dari bawah sampai atas akan dicetak. Jika bawah lebih besar dari atas, maka deret ganjil dari atas sampai bawah akan dicetak.

### SOAL 3

```
def hitung_ips(jumlah_matkul):
    total_sks = 0
    total_nilai = 0

    for i in range(1, jumlah_matkul+1):
        print(f"Masukkan nilai mata kuliah ke-{i}:")
        nilai = input("Nilai (A/B/C/D): ")

        sks = 3 # diasumsikan setiap mata kuliah memiliki sks 3

        if nilai.upper() == "A":
            bobot = 4
        elif nilai.upper() == "B":
            bobot = 3
        elif nilai.upper() == "C":
            bobot = 2
        elif nilai.upper() == "D":
            bobot = 1
        else:
            print("Input nilai tidak valid. Harap masukkan nilai A, B, C, atau D.")
            continue

        total_sks += sks
        total_nilai += sks * bobot

    ips = total_nilai / total_sks
    return ips

# Input jumlah mata kuliah
jumlah_matkul = int(input("Masukkan jumlah mata kuliah: "))

# Panggil fungsi hitung_ips
hasil_ips = hitung_ips(jumlah_matkul)

# Cetak hasil IPS
print(f"Indeks Prestasi Semester (IPS): {hasil_ips:.2f}")
```

Masukkan jumlah mata kuliah: 7  
Masukkan nilai mata kuliah ke-1: a  
Masukkan nilai mata kuliah ke-2: b  
Masukkan nilai mata kuliah ke-3: c  
Masukkan nilai mata kuliah ke-4: d  
Masukkan nilai mata kuliah ke-5: a  
Masukkan nilai mata kuliah ke-6: b  
Masukkan nilai mata kuliah ke-7: a  
Indeks Prestasi Semester (IPS): 3.00

#### 1. Fungsi hitung\_ips(jumlah\_matkul):

- Fungsi ini menerima satu parameter, jumlah\_matkul, yang merupakan jumlah mata kuliah.
- Variabel total\_sks dan total\_nilai diinisialisasi dengan nilai 0 untuk menyimpan total sks dan total nilai dari semua mata kuliah.
- Program menggunakan perulangan for untuk meminta input nilai (A/B/C/D) untuk setiap mata kuliah.
- Setiap mata kuliah diasumsikan memiliki 3 sks (sks = 3).
- Bobot nilai bobot diatur sesuai dengan input nilai: A=4, B=3, C=2, D=1.
- Total sks dan total nilai dari setiap mata kuliah diakumulasi ke dalam variabel total\_sks dan total\_nilai.
- IPS dihitung dengan rumus  $\text{total\_nilai} / \text{total\_sks}$  dan di-return sebagai hasil.

#### 2. Input dari pengguna:

- Pengguna diminta untuk memasukkan jumlah mata kuliah.

#### 3. Panggil fungsi hitung\_ips():

- Fungsi hitung\_ips() dipanggil dengan parameter jumlah\_matkul yang dimasukkan oleh pengguna.

#### 4. Cetak hasil IPS:

- Hasil dari fungsi hitung\_ips() disimpan dalam variabel hasil\_ips.
- IPS dicetak ke layar dengan format dua angka desimal.



