



---

# Laporan Praktikum Algoritma & Pemrograman

Semester Genap 2024/2025

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.

NIM	71231046
Nama Lengkap	FREIRE HANAN PUTRA
Minggu ke / Materi	06 / Struktur Kontrol Perulangan

PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA  
YOGYAKARTA  
2025

## BAGIAN 1: MATERI MINGGU INI (40%)

### MATERI 1 Devinisi Perulangan

**Struktur kontrol** adalah pengaturan terhadap jalannya suatu program dapat dilakukan dengan cara sekuensial, percabangan, perulangan, maupun kombinasi dari ketiganya.

Berikut ini beberapa hal mengapa kita membutuhkan **perulangan** :

1. Untuk melakukan hal yang sama berulang kali.
2. Untuk melakukan hal secara bertahap dan disetiap tahap sebenarnya memiliki Langkah yang sama.
3. Untuk mengambil data dari struktur data seperti List, Tuple, dan data lainnya.

Untuk menggunakan **perulangan** kita dapat memakai **for**, **while** atau **rekursif** pada Python.

### MATERI 2 Bentuk Perulangan For

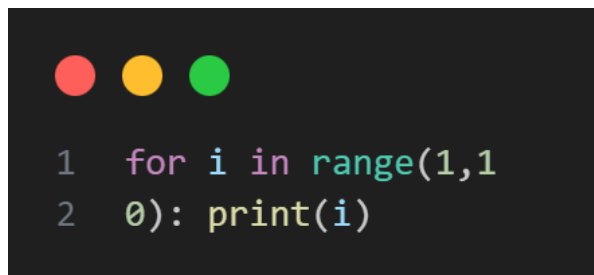
Berikut ini beberapa kondisi dimana kita akan menggunakan **perulangan for** :

1. Kita sudah mengetahui jumlah perulangan sejak awal.
2. Terjadi operasi yang sama pada rentang data atau rentang nilai.

**Perulangan for** dapat kita gunakan dengan bantuan **fungsi range()**, berikut ini beberapa bentuk dari **fungsi range()** :

1. **range(stop)** , menghasilkan rentang dari **0** sampai **stop-1**.
2. **range(start, stop, [step])**, menghasilkan rentang dari **start**, sampai **stop** dengan peningkatan sejumlah **step**.


Agar lebih memahaminya perhatikan contoh berikut:



```
1 for i in range(1,10)
2   print(i)
```

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

Dari contoh diatas terdapat `i` yang berperan sebagai **counter**:




```
1 for i in range(1,10):  
2     print("Hello World")
```

```
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World
```

Step Negative

Agar lebih memahami apa itu step negatif perhatikan contoh berikut ini.



```
1 for i in range(1,11,2):  
2     print(i)
```

```
1
3
5
7
9
```

Lalu bagaimana jika kita ingin menampilkan bilangan ganjil dari 10-1? Pada kasus ini kita menggunakan **step negative**, berikut contohnya.

```
1 for i in range(10,0,-1):
2     if i %2 != 0:
3         print(i)
```

```
D:\Semester 4\PrakAlpro 2425\week6>"C:/Python/Python37/python.exe"
py"
9
7
5
3
1
```

### MATERI 3 Bentuk Perulangan While

Perulangan **while** digunakan jika kita belum mengetahui jumlah perulangan sebelumnya. Agar lebih memahami berikut contohnya.

```
1 bilangan = 0
2 ganjil = False
3 while ganjil == False:
4     bilangan = int(input("Masukan Bilangan: "))
5     if bilangan %2 !=0:
6         ganjil = True
7         print(bilangan, "adalah ganjil")
```

```
D:\Semester 4\PrakAlpro 2425\week6>"C:/
py"
Masukan Bilangan: 3
3 adalah ganjil


D:\Semester 4\PrakAlpro 2425\week6>
```


## MATERI 4 Penggunaan Break dan Continue

Untuk mengontrol perulangan kita dapat menggunakan **break** dan **continue**.

1. **Break** digunakan untuk **menghentikan perulangan**.
2. **Continue** digunakan untuk **melanjutkan perulangan ke iterasi selanjutnya**.

Agar lebih memahaminya perhatikan contoh **break** dan **continue** berikut :

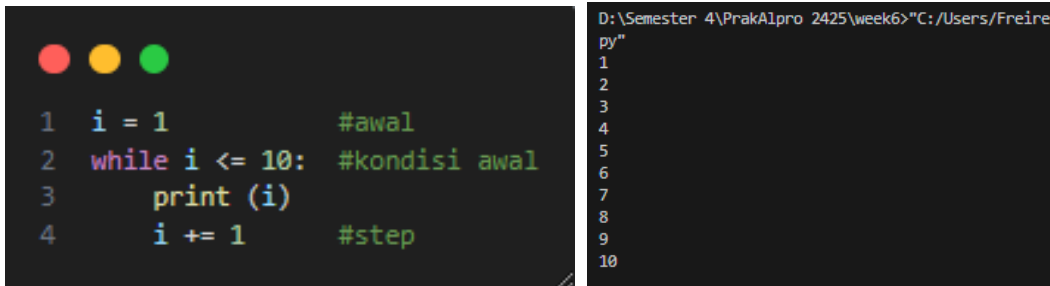
 <pre>1 for i in range(1,11): 2     if i == 9: 3         break 4     else: 5         print(i) 6 print("Selesai")</pre>	<pre>D:\Semester 4 py" 1 2 3 4 5 6 7 8 Selesai</pre>
---	--

 <pre>1 for i in range(1,11): 2     if i == 6: 3         continue 4     else: 5         print(i) 6 print("Selesai")</pre>	<pre>D:\Semester 4 py" 1 2 3 4 5 7 8 9 10 Selesai</pre>
--	---

## MATERI 5 Konveri dari Bentuk For Menjadi Bentuk While

Berikut beberapa hal yang harus diperhatikan agar dapat mengubah bentuk **for** menjadi **while** :

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



The image shows a code editor with a dark background. On the left, there is a Python script with four lines of code, each preceded by a line number (1-4). The code is as follows:

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

On the right, the output of the script is displayed, showing the numbers 1 through 10, each on a new line. The output is preceded by a file path: D:\Semester 4\PrakAlpro 2425\week6>"C:/Users/Freire

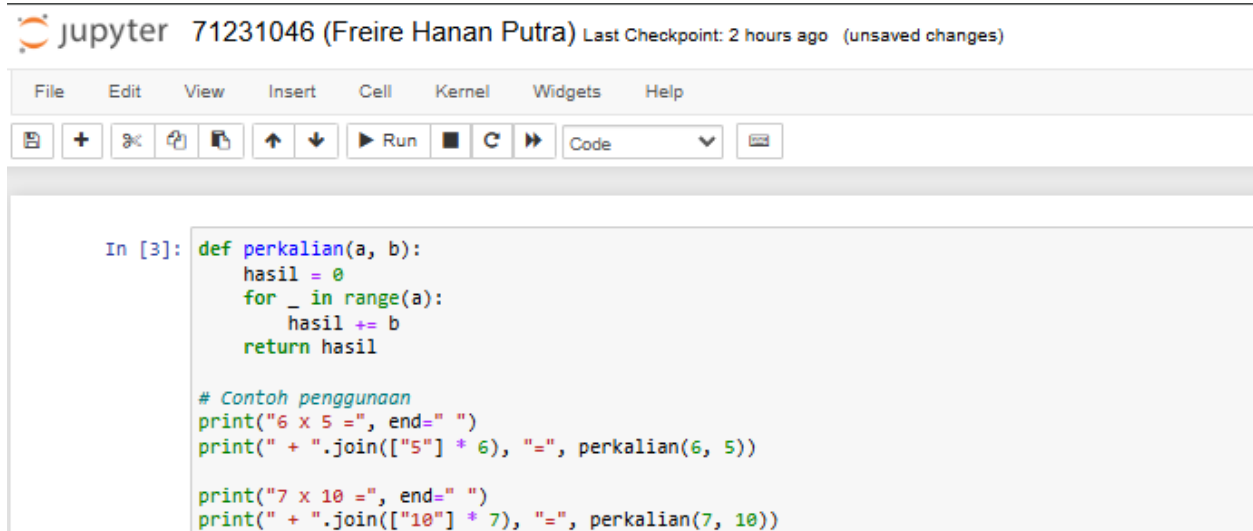
Perulangan tersebut dimulai dari 1, berakhir di 10, dengan langkah (step) sebesar 1. Konversi ke bentuk **while** dapat dilakukan dengan mudah, sehingga menghasilkan perulangan **while** yang memberikan output yang sama.

## BAGIAN 2: LATIHAN MANDIRI (60%)

Link GITHUB: [https://github.com/Freirehnn23/prak\\_alpro\\_week6](https://github.com/Freirehnn23/prak_alpro_week6)

### SOAL 1

Code:



The screenshot shows a Jupyter Notebook interface with the title "71231046 (Freire Hanan Putra) Last Checkpoint: 2 hours ago (unsaved changes)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and other notebook functions. The code cell contains the following Python code:

```
In [3]: def perkalian(a, b):
        hasil = 0
        for _ in range(a):
            hasil += b
        return hasil

        # Contoh penggunaan
        print("6 x 5 =", end=" ")
        print(" + ".join(["5"] * 6), "=", perkalian(6, 5))

        print("7 x 10 =", end=" ")
        print(" + ".join(["10"] * 7), "=", perkalian(7, 10))
```

Penerapan:

```
6 x 5 = 5 + 5 + 5 + 5 + 5 + 5 = 30
7 x 10 = 10 + 10 + 10 + 10 + 10 + 10 + 10 = 70
```

Penjelasan:

**Fungsi perkalian(a, b):**

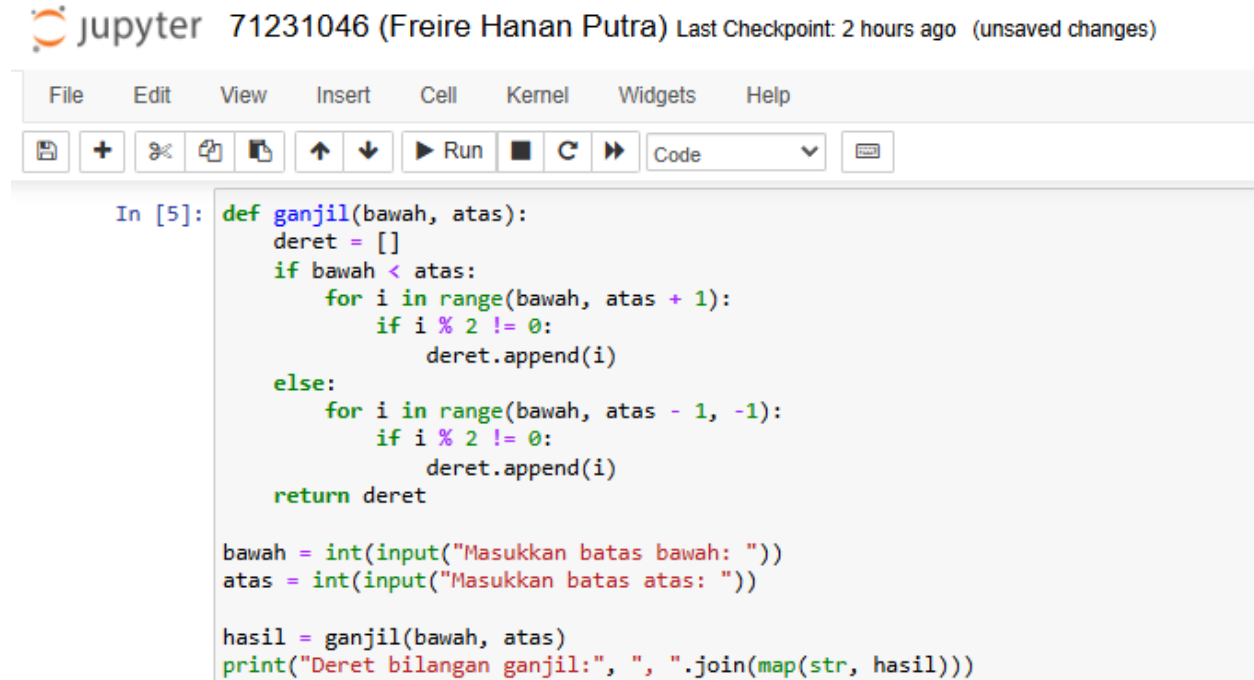
- Menggunakan penjumlahan berulang untuk menghitung perkalian  $a \times b$ .
- Variabel hasil dimulai dari 0 dan ditambah  $b$ , sebanyak  $a$  kali.

**Contoh Penggunaan:**

- `print("6 x 5 =", end=" ")` → Menampilkan operasi perkalian tanpa ganti baris.
- `" + ".join(["5"] * 6)` → Menampilkan proses penjumlahan berulang ( $5 + 5 + 5 + 5 + 5 + 5$ ).
- Menampilkan hasil akhir dengan memanggil `perkalian(6, 5)`.
- Sama untuk  $7 \times 10$ .

## SOAL 2

Code:



Jupyter 71231046 (Freire Hanan Putra) Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Run

```
In [5]: def ganjil(bawah, atas):
        deret = []
        if bawah < atas:
            for i in range(bawah, atas + 1):
                if i % 2 != 0:
                    deret.append(i)
        else:
            for i in range(bawah, atas - 1, -1):
                if i % 2 != 0:
                    deret.append(i)
        return deret

        bawah = int(input("Masukkan batas bawah: "))
        atas = int(input("Masukkan batas atas: "))

        hasil = ganjil(bawah, atas)
        print("Deret bilangan ganjil:", ", ".join(map(str, hasil)))
```

Penerapan:

```
Masukkan batas bawah: 67
Masukkan batas atas: 88
Deret bilangan ganjil: 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87
```

Penjelasan:

**Fungsi ganjil(bawah, atas):**

- Menghasilkan deret bilangan ganjil antara **bawah** dan **atas**.
- Jika **bawah < atas**, iterasi maju.
- Jika **bawah > atas**, iterasi mundur.
- Hanya angka ganjil ( $i \% 2 \neq 0$ ) yang dimasukkan ke dalam **list deret**.

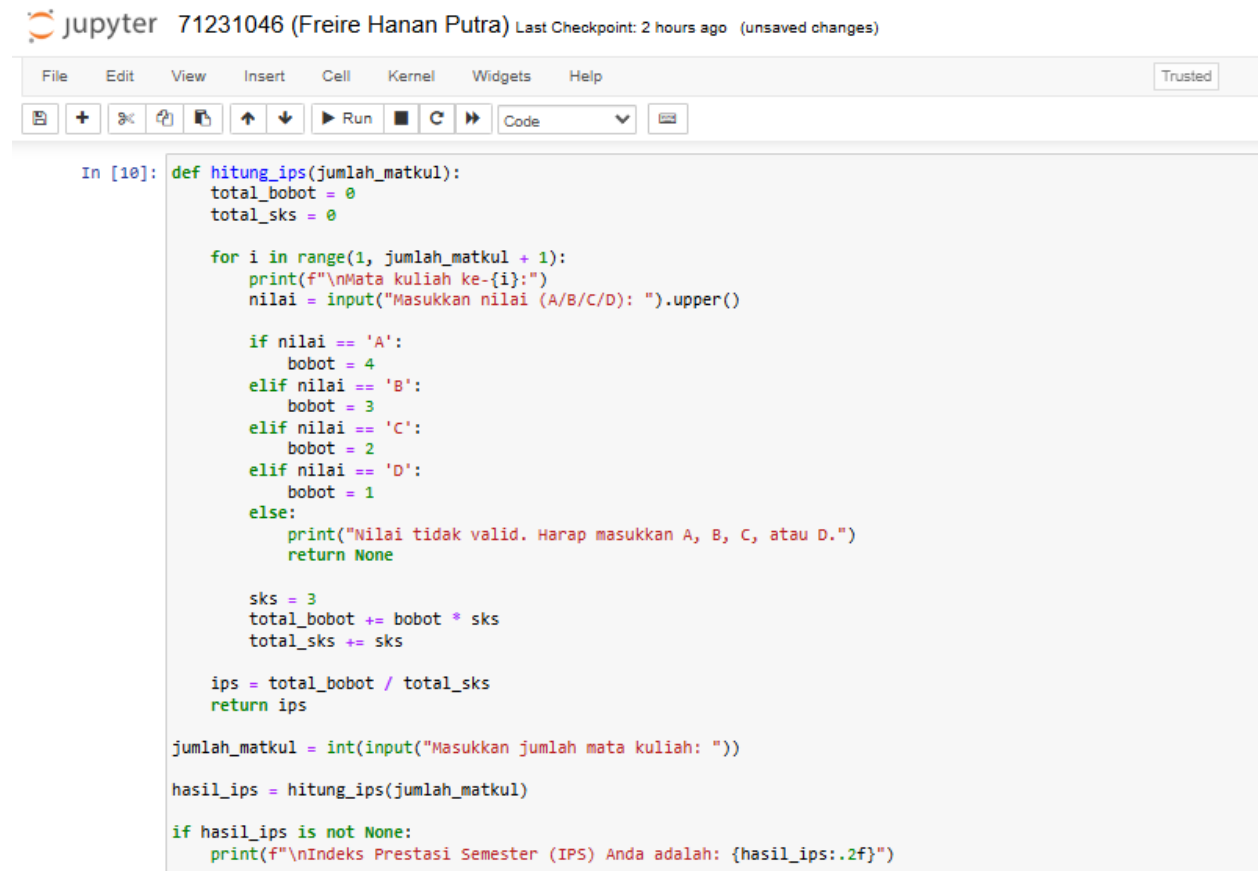
**Program Utama:**

- Meminta input **batas bawah** dan **batas atas** dari pengguna.
- Memanggil fungsi `ganjil(bawah, atas)`.
- Menampilkan hasil dalam format deret angka yang dipisahkan koma.



## SOAL 3

Code:



```
In [10]: def hitung_ips(jumlah_matkul):
total_bobot = 0
total_sks = 0

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

    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. Harap masukkan A, B, C, atau D.")
        return None

    sks = 3
    total_bobot += bobot * sks
    total_sks += sks

ips = total_bobot / total_sks
return ips

jumlah_matkul = int(input("Masukkan jumlah mata kuliah: "))
hasil_ips = hitung_ips(jumlah_matkul)

if hasil_ips is not None:
    print(f"\nIndeks Prestasi Semester (IPS) Anda adalah: {hasil_ips:.2f}")
```

Penerapan:

Masukkan jumlah mata kuliah: 4

Mata kuliah ke-1:

Masukkan nilai (A/B/C/D): A

Mata kuliah ke-2:

Masukkan nilai (A/B/C/D): D

Mata kuliah ke-3:

Masukkan nilai (A/B/C/D): C

Mata kuliah ke-4:

Masukkan nilai (A/B/C/D): A

Indeks Prestasi Semester (IPS) Anda adalah: 2.75

Penjelasan:

Fungsi `hitung_ips(jumlah_matkul):`

- Menghitung **Indeks Prestasi Semester (IPS)** berdasarkan nilai dan SKS.

- **Variabel total\_bobot** menyimpan total nilai bobot dikali SKS.
- **Variabel total\_sks** menyimpan jumlah total SKS.
- Iterasi dilakukan sebanyak **jumlah mata kuliah** yang dimasukkan pengguna.
- Mengonversi nilai **A, B, C, D** menjadi bobot **4, 3, 2, 1**.
- Jika nilai tidak valid, program menampilkan pesan error dan berhenti (return None).
- Setiap mata kuliah dianggap memiliki **3 SKS**.
- IPS dihitung dengan cara  $IPS = \text{total bobot} / \text{total sks}$

**Program Utama:**

- Meminta input jumlah mata kuliah dari pengguna.
- Memanggil fungsi `hitung_ips(jumlah_matkul)`.
- Menampilkan IPS dengan format dua angka desimal.