

# Modul Pemrograman Lanjut

Sevi Nurafni

**Catatan**

1. Modul ini dirancang untuk dapat menjadi pegangan mata kuliah Pemrograman Lanjut
2. Anda dapat membuka modul ini saat latihan praktikum.
3. Anda sangat disarankan untuk mencoba menjalankan semua program modul ini di komputer Anda, supaya Anda dapat mengetahui keluaran dari program yang ada.
4. Anda sangat disarankan untuk bereksperimen dari program-program yang ada di modul ini supaya Anda mendapat gambaran lebih jelas mengenai apa yang program Anda lakukan.
5. Anda sangat disarankan membaca tutorial dari tempat lain dan mengeksplor sendiri bahasa yang Anda gunakan.

## 1. PERULANGAN (LOOP)

Pada pemrograman, sering kali dibutuhkan pemrosesan berulang-ulang untuk mencapai suatu hasil tertentu. Apabila pengulangan ini dilakukan secara manual ukuran file program akan menjadi terlalu besar. Contoh sederhana adalah jika kita ingin menuliskan “Hello World” di layar sebanyak 1000 kali, maka akan dibutuhkan paling tidak 1000 baris perintah. Menggunakan sintaks pengulangan, persoalan tersebut dapat diselesaikan hanya menggunakan beberapa baris program.

### 1.1. FOR LOOP

- Bentuk looping yang kedua adalah bentuk For. Bentuk ini umumnya digunakan untuk pengulangan yang sudah diketahui jumlahnya (*definite loop*). *Definite iteration* adalah sebuah proses iterasi atau perulangan ketika jumlah pengulangannya ditentukan secara eksplisit sebelumnya. Namun, for loop juga dapat dibuat menggunakan while-do loop.
- Struktur program FOR LOOP pada umumnya adalah:



```
for <var> in <iterable>:  
    <statement(s)>
```

- [<iterable>](#) merupakan segala *object* dalam Python yang dapat diiterasi seperti list, tuple, hingga string. Ada pula `<var>` merupakan variabel yang akan mengambil elemen berikutnya dari `<iterable>` setiap kali iterasi berjalan.
- Berikut adalah program yang menerima  $a$  dan  $b$  dan menuliskan  $a, a + 1, a + 2, \dots, b - 1, b$ .

```
a = int(input())  
b = int(input())  
  
for i in range(a,b+1):  
    print(i)
```

- Perhatikan bahwa:
  - a. Sintaks for loop adalah **for variable in range(value):**
  - b. Perulangan body for loop akan dilakukan sebanyak jumlah range yang diberikan.
  - c. Body for loop adalah bagian kode yang indentasi ke dalam body for loop.
  - d. Nilai variable akan berubah sesuai dengan nilai selanjutnya dalam range setiap perulangan.
- **Range** dapat digunakan dengan satu, dua, atau tiga parameter. Struktur dari Range adalah sebagai berikut:



Berikut adalah penjelasan detail terkait fungsi "range()".

- ✓ **"Start"** merupakan nilai awal dari urutan bilangan yang bersifat opsional, jika Anda tidak memasukkannya, nilai awal akan dianggap 0.
- ✓ **"Stop"** merupakan nilai batas yang wajib dimasukkan. Urutan akan berhenti sebelum mencapai nilai "stop" (eksklusif).
- ✓ **"Step"** merupakan nilai penambahan antara setiap dua bilangan dalam urutan yang bersifat opsional. Jika nilai tersebut tidak diberikan, secara default nilai yang dimasukkan adalah 1

\*) Untuk lebih jelasnya, perhatikan potongan kode berikut (coba terapkan range pada program FOR LOOP sebagai 'Eksporasi Pribadi di Kelas').

<code>range(10)</code>	<code>#</code>	<code>0</code>	<code>1</code>	<code>2, 3, 4, 5, 6, 7, 8, 9</code>
<code>range(2, 5)</code>	<code>#</code>	<code>2</code>	<code>3</code>	<code>4</code>
<code>range(7, 2, -1)</code>	<code>#</code>	<code>7</code>	<code>6</code>	<code>5, 4, 3</code>
<code>range(2, 8, 2)</code>	<code>#</code>	<code>2</code>	<code>4</code>	<code>6</code>

```
for i in range (1000):
    print(" Hello World")
```

## 1.2. WHILE LOOP

- While termasuk sintaks dalam Python yang bersifat indefinite iteration. *Indefinite iteration* adalah sebuah proses iterasi yang akan berhenti ketika memenuhi kondisi tertentu.
- Salah satu sintaks yang looping / pengulangan yang sering digunakan adalah sintaks While-Do. Program akan mengecek sebuah kondisi (kondisi pada akhirnya meminta output berupa TRUE atau FALSE, karena pada kondisi kita menggunakan Operator Logika atau Boolean), yang diberikan terlebih

dahulu sebelum menjalankan statement yang ada di dalamnya.

- Struktur program WHILE LOOP adalah:

```
while kondisi:  
    # blok pernyataan yang akan diulang selama kondisi bernilai True
```

- Berikut adalah program yang menerima  $a$  dan  $b$  dan menuliskan  $a, a + 1, a + 2, \dots, b - 1, b$ .

```
a = int(input())  
b = int(input())  
  
i = a  
while (i <= b):  
    print(i)  
    i += 1
```

- Perhatikan bahwa:

- a. Sintaks while loop adalah **while (kondisi / Boolean)**:
- b. Perulangan body while loop akan terus dilakukan selama kondisi terpenuhi.
- c. Body while adalah bagian kode yang indentasi ke dalam body while loop.

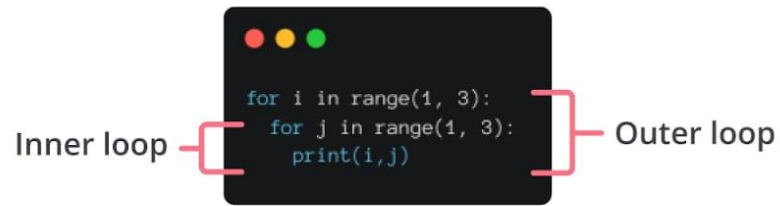
### 1.3. Perulangan Bersarang (FOR NESTED LOOP)

- Perhatikan pula, perulangan dapat dilakukan di dalam perulangan. Sebagai contoh, berikut adalah program untuk membuat pola persegi.

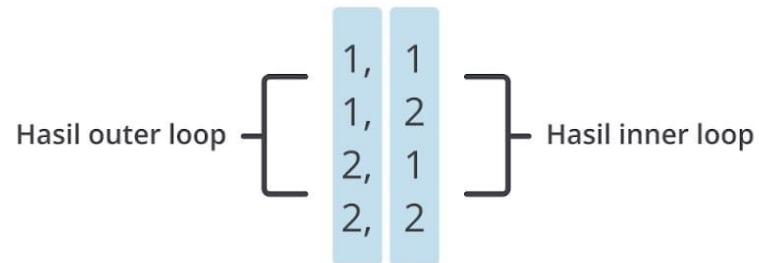
```
n = int(input())  
for i in range(n):  
    for j in range(n):  
        print("*", end=" ") # Dieksekusi sebanyak n * n kali print()  
                           # Dieksekusi sebanyak n kali  
    # Perhatikan indentasi di dalam perulangan.
```

- Anda dapat asumsikan bahwa ada dua perulangan, yakni "perulangan luar" dan "perulangan dalam". Program akan melakukan "perulangan luar" terlebih dahulu, lalu akan melakukan "perulangan dalam". "variabel\_luar" akan mengambil nilai dari "iterable\_luar", sedangkan "variabel\_dalam" akan mengambil nilai dari "iterable\_dalam".
- Hal ini dapat dikembangkan bahwa body dari perulangan bersarang dapat berupa potongan kode lainnya. Kita dapat menggunakan I/O, operasi, dan perulangan lainnya dalam body loop. Kita dapat juga menggunakan while loop di dalam for loop dan sebaliknya.

- Struktur program FOR NESTED LOOP :



### OUTPUT



## 1.4. Latihan Modul 2

### 1.4.1. Kelipatan 10 terkecil

Buatlah program yang menerima bilangan N dan menuliskan bilangan  $10^x$  terkecil yang lebih dari N.

Masukkan N: 5  
10

Contoh 1

Masukkan N: 1234  
10000

Contoh 2

Loop manakah yang lebih baik digunakan untuk soal tersebut?

### 1.4.2. Segitiga Angka

Buatlah program yang menerima bilangan N dan membuat pola segitiga angka dengan dasar N seperti contoh.

Masukkan N: 5  
1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5

Contoh 1

Loop manakah yang lebih baik digunakan untuk soal tersebut?

Dari kedua contoh tersebut, bagaimanakah perbedaan karakteristik kedua loop?