

MODUL IV

PERULANGAN

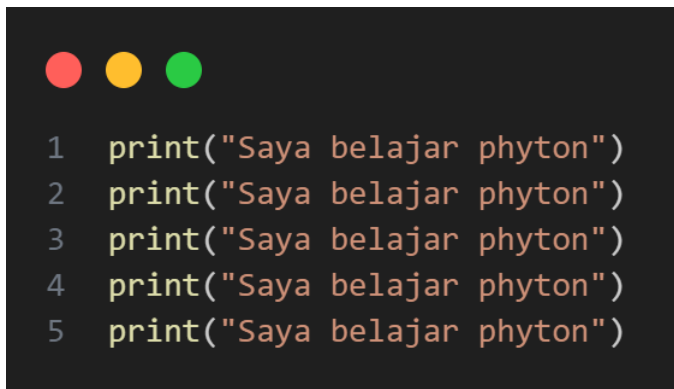
1. TUJUAN

- Memahami struktur perulangan dalam bahasa pemrograman Python
- Mengenal struktur For, While
- Mampu menyelesaikan persoalan tentang perulangan

2. DASAR TEORI

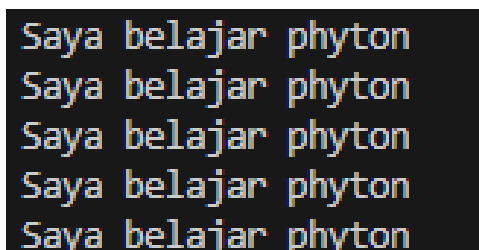
Dalam pembuatan program, terkadang kita harus melakukan pengulangan suatu aksi misalnya untuk melakukan perhitungan berulang dengan menggunakan formula yang sama. Sebagai contoh, misalnya kita ingin membuat program yang dapat menampilkan sebuah teks *"Saya sedang belajar python"* sebanyak 5 kali, maka kita tidak perlu untuk menuliskan 5 buah statement melainkan kita hanya tinggal menempatkan satu buah statement ke dalam suatu struktur pengulangan. Dengan demikian program kita akan lebih efisien.

Sebagai gambaran bagi anda untuk dapat lebih memahami konsep perulangan, coba perhatikan kode program di bawah ini :



```
1 print("Saya belajar phyton")
2 print("Saya belajar phyton")
3 print("Saya belajar phyton")
4 print("Saya belajar phyton")
5 print("Saya belajar phyton")
```

Output:




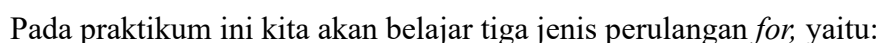
```
Saya belajar phyton
Saya belajar phyton
Saya belajar phyton
Saya belajar phyton
Saya belajar phyton
```

Pada source code diatas sangat tidak efisien karena tidak menggunakan struktur perulangan di dalamnya.

Perbedaan percabangan dan perulangan:

- ### 3. STRUKTUR FOR

Adapun bentuk umum dari pendefinisian struktur for untuk satu statement adalah sebagai berikut :



```
1 for i in range(7):  
2     print("Hallo world!")
```

```
PS C:\Users\Diva> &
Hallo world!
Hallo world!
Hallo world!
Hallo world!
Hallo world!
Hallo world!
Hallo world!
```

Untuk `for` jenis pertama ini, kita masukan banyaknya perulangan yang ingin dilakukan ke dalam `range()`. Nilai variable `i` nantinya akan berubah, dimulai dari 0 hingga bilangan yang dimasukan ke `range()`, dan setiap perulangan bilangan tersebut akan dikurangi satu.

b. Range (*min,max*)

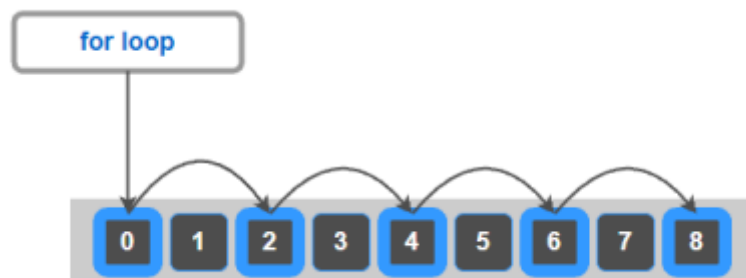
```
1 for i in range(1, 10):
2     print(f"Perulangan ke- {i}")
```

Output:

```
PS C:\Users\Diva> &
Perulangan ke- 1
Perulangan ke- 2
Perulangan ke- 3
Perulangan ke- 4
Perulangan ke- 5
Perulangan ke- 6
Perulangan ke- 7
Perulangan ke- 8
Perulangan ke- 9
```


Pada perulangan ini counter `i` menyimpan nilai pada *range min* (nilai awal) adalah 1 dan *max* (batas akhir) adalah 10. Perlu diingat bahwa batas akhir selalu dikurang 1. Sehingga, hasil perulangan yang dihasilkan adalah 1 sampai 9.

Contoh diatas adalah perulangan `for` tanpa menggunakan **STEP**, berikut adalah ilustrasi jika perulangan `for` menggunakan **STEP**:



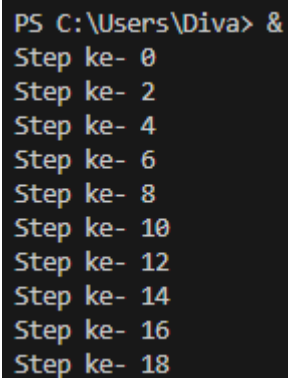
Dengan menggunakan **STEP** kita dapat mengatur berapa pertambahan di setiap perulangan yang dijalankan.

c. Range (*min*, *max*, *step*)



```
1 for i in range(0, 20, 2):
2     print(f"Step ke- {i}")
```

Output:

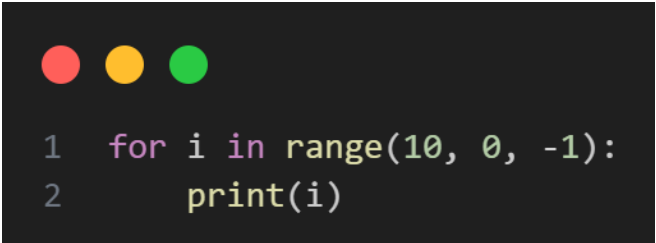


```
PS C:\Users\Diva> &
Step ke- 0
Step ke- 2
Step ke- 4
Step ke- 6
Step ke- 8
Step ke- 10
Step ke- 12
Step ke- 14
Step ke- 16
Step ke- 18
```

Pada struktur perulangan *for* ini counter *i* menyimpan nilai pada range *min* (nilai awal) adalah 0 dan *max* (nilai akhir) adalah 20, lalu pada code tersebut kita menambahkan *step* 2 untuk setiap perulangan.

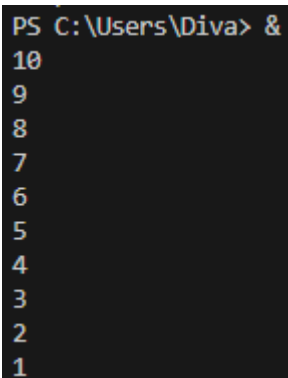
Contoh lainnya kita akan membuat sebuah perulangan menurun/decrement.

Source code:



```
1 for i in range(10, 0, -1):
2     print(i)
```

Output:



```
PS C:\Users\Diva> &
10
9
8
7
6
5
4
3
2
1
```


Perulangan increment:

```
● ● ●  
1 a = 1  
2 b = 10  
3 while a < b:  
4     print("Step ke-", a)  
5     a += 1
```

Output:

```
PS C:\Users\Diva> &  
Step ke- 1  
Step ke- 2  
Step ke- 3  
Step ke- 4  
Step ke- 5  
Step ke- 6  
Step ke- 7  
Step ke- 8  
Step ke- 9
```

Perulangan decrement:

```
● ● ●  
1 a = 10  
2 b = 0  
3 while a > b:  
4     print("Kuota internet anda sisa", a, "GB")  
5     a -= 1
```

Output:

```
PS C:\Users\Diva> & C:/Users/Diva/AppData/  
Kuota internet anda sisa 10 GB  
Kuota internet anda sisa 9 GB  
Kuota internet anda sisa 8 GB  
Kuota internet anda sisa 7 GB  
Kuota internet anda sisa 6 GB  
Kuota internet anda sisa 5 GB  
Kuota internet anda sisa 4 GB  
Kuota internet anda sisa 3 GB  
Kuota internet anda sisa 2 GB  
Kuota internet anda sisa 1 GB
```

5. FUNGSI *BREAK* & *CONTINUE*

Pada python, kita bisa menginterupsi dan juga men-skip suatu iterasi pada perulangan. Terdapat 2 perintah yang bisa kita gunakan, yaitu: ***break*** untuk interupsi (memberhentikan paksa) sebuah perulangan dan ***continue*** untuk menskip ke iterasi selanjutnya.

Contoh ***break*** pada perulangan *for*:

```
1 for i in range(1, 10):
2     print("Ini perulangan ke-", i)
3     if i == 7:
4         print("Perulangan ke-", i, "dihentikan!")
5         break
```

Output:

```
d:/Kuliah/ASPRAK/PEMROG_1/perulangan.py
Ini perulangan ke- 1
Ini perulangan ke- 2
Ini perulangan ke- 3
Ini perulangan ke- 4
Ini perulangan ke- 5
Ini perulangan ke- 6
Ini perulangan ke- 7
Perulangan ke- 7 dihentikan!
```

Contoh ***continue*** pada perulangan *for*:

```
1 for i in range(0, 10):
2     # skip jika i == 5
3     if (i == 5):
4         continue
5
6     print(i)
7
```

Output:

```
PS C:\Users\Diva> &
0
1
2
3
4
6
7
8
9
```

Contoh **break** pada perulangan *while*:

```
1 a = 0
2 while True:
3     print("Step ke-", a, "!")
4     a += 1
5     if a == 7:
6         print("Step ke-", a, "dihentikan!")
7         break
```

Output:

```
d:/Kuliah/ASPRAK/PEMROG_1/perulangan.py
Step ke- 0 !
Step ke- 1 !
Step ke- 2 !
Step ke- 3 !
Step ke- 4 !
Step ke- 5 !
Step ke- 6 !
Step ke- 7 dihentikan!
```

Contoh **continue** pada perulangan *while*:

```
1 angka = ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10']
2
3 # skip jika i adalah bilangan genap
4 # dan i lebih dari 0
5 i = -1
6 while i < len(angka):
7     i += 1
8     if i % 2 == 0 and i > 0:
9         print('skip')
10        continue
11
12 # tidak dieksekusi ketika continue dipanggil
13 print(angka[i])
```


Output:

```
PS C:\Users\Divi> &
1
2
skip
4
skip
6
skip
8
skip
10
skip
```

6. LATIHAN

- a. Buatlah sebuah sistem login sederhana, dengan ketentuan jika user memasukan password yang salah dalam waktu 3x maka tidak dapat melakukan login kembali

DATA	VALUE
username	namakamu
password	terserahkamu

Output:

- Jika username dan password benar

```
PS C:\Users\Divi> & C:/Users/Divi/AppData/
=====SISTEM LOGIN SEDERHANA=====
Masukkan username: divaangelica
Masukkan password: 12345678
Selamat datang divaangelica !
```

- Jika username dan password salah lebih dari 3X

```
PS C:\Users\Divi> & C:/Users/Divi/AppData/Local/Programs/Python/Python310/python.exe
=====SISTEM LOGIN SEDERHANA=====
Masukkan username: diva
Masukkan password: 1234
Coba cek kembali, username atau password salah!
Masukkan username: divaangel
Masukkan password: 12345
Coba cek kembali, username atau password salah!
Masukkan username: dipaaa
Masukkan password: 12345
Coba cek kembali, username atau password salah!
Masukkan username: divaangel
Masukkan password: 12345678
Login gagal! Kesempatan mencoba sudah habis. Silakan coba lagi nanti!
```

- b. Buatlah sebuah program mencari bilangan genap, dimana inputan berasal dari user adalah berupa range maksimal. Sehingga output yang dihasilkan adalah sebagai berikut:

```
PS C:\Users\Diva> & C:/Users/Diva/AppData/Local/Programs/PowerShell/PowerShell.exe
=====MENCARI BILANGAN GENAP=====
Masukkan Bilangan Maksimal: 30
2
4
6
8
10
12
14
16
18
20
22
24
26
28
```

- c. Membuat program dengan menggunakan for untuk menentukan nilai factor dari persekutuan terbesar dari dua buah bilangan bulat. Sebagai contoh kita memasukkan dua buah bilangan bulat yaitu 12 dan 72, maka FPB dari kedua bilangan tersebut adalah 12. Sehingga, output yang dihasilkan adalah:

```
PS C:\Users\Diva> & C:/Users/Diva/AppData/Local/Programs/PowerShell/PowerShell.exe
=====PROGRAM MENCARI FPB=====
Masukan bilangan pertama: 12
Masukan bilangan kedua: 72
Faktor Persekutuan Terbesar = 12
```

7. TUGAS

- 1) Buatlah sebuah program dengan statement perulangan dimana dapat menghitung total nilai dari suatu bilangan yang diinputkan. Dengan tampilan output sebagai berikut:

```
0/python.exe "d:/#1 ITTP/SEMESTER 2/Praktikum Pemrograman 1/Project
===== PROGRAM SEDERHANA MENGHITUNG JUMLAH TOTAL BILANGAN =====
Masukkan bilangan = 10
Total Nilai = 10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 = 55
```

**inputkan bilangan sesuai dengan dua angka terakhir nim kalian, jika yang depannya 0 cukup satu angka terakhir saja. (misal: 02, cukup input 2)*

- 2) Buatlah sebuah program dengan statement perulangan, dimana dapat menghitung hasil pangkat suatu bilangan. Dengan tampilan output sebagai berikut:

```
0/python.exe "d:/#1 ITTP/SEMESTER 2/Praktikum Pemrograman 1/
===== PROGRAM SEDERHANA MENGHITUNG PANGKAT =====
Masukkan bilangan = 36
Masukkan pencacah = 2
Hasil pangkat = 1296
```

**inputkan bilangan dengan 2 angka awal nim + 2 angka terakhir (misal: 21104015, maka $21 + 15 = 36$) dan pencacah inputkan nilai 2 saja.*

- 3) Buatlah sebuah program dengan statement perulangan untuk menentukan KPK dari dua buah bilangan bulat. Sebagai contoh KPK dari 8 dan 12 adalah 24.

```
0/python.exe "d:/#1 ITTP/SEMESTER 2/Praktikum Pemrograman 1,
===== PROGRAM SEDERHANA MENCARI KPK =====
Masukkan bilangan pertama = 8
Masukkan bilangan kedua = 12
KPK = 24
```