

**LAPORAN PRAKTIKUM
PEMROGRAMAN 1
MODUL 4**



Oleh:

Fauzan Wahyu Mubarak

2211104027

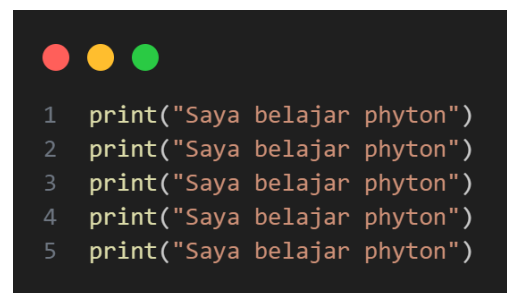
SE06A

**PRODI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2023**

I. DASAR TEORI

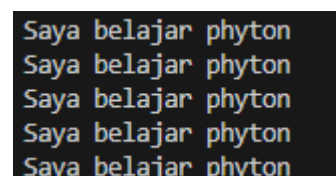
Dalam pembuatan program, terkadang kita harus melakukan pengulangan suatu aksi misalnya untuk melakukan perhitungan berulang dengan menggunakan formula yang sama. Sebagai contoh, misalnya kita ingin membuat program yang dapat menampilkan sebuah teks “*Saya sedang belajar python*” sebanyak 5 kali, maka 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,

cobaperhatikan 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
```

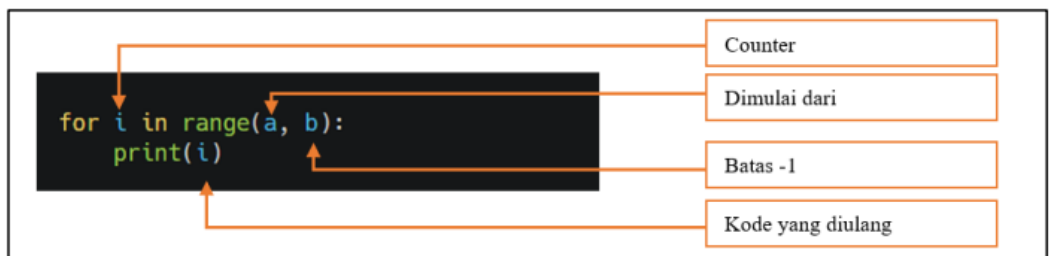
Pada source code diatas sangat tidak efisien karena tidak menggunakan struktur perulangan di dalamnya. Perulangan dalam dunia pemrograman adalah baris kode atau instruksi yang dieksekusi oleh komputer secara berulang-ulang sampai suatu kondisi tertentu terpenuhi.

Perbedaan percabangan dan perulangan:

- Kalau percabangan, blok kode yang memenuhi kondisi tertentu hanya akan dieksekusi satu kali saja.
- Sedangkan perulangan, ia akan dilakukan seterusnya berulang-ulang dengan jumlah tertentu atau selama kondisi tertentu terpenuhi

1. STRUKTUR FOR

For pada python lebih dikenal sebagai foreach. Struktur for ini digunakan untuk menuliskan jenis perulangan yang banyaknya sudah pasti atau telah diketahui sebelumnya. Oleh karena itu, disini kita harus melakukan inisialisasi nilai untuk kondisi awal pengulangan dan juga harus menuliskan kondisi untuk mengentikan proses pengulangan. Adapun bentuk umum dari pendefinisian struktur for untuk satu statement adalah sebagai berikut :



Pada praktikum ini kita akan belajar tiga jenis perulangan *for*, yaitu:

a. Range (*max*)

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

Output:

```
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}")

```

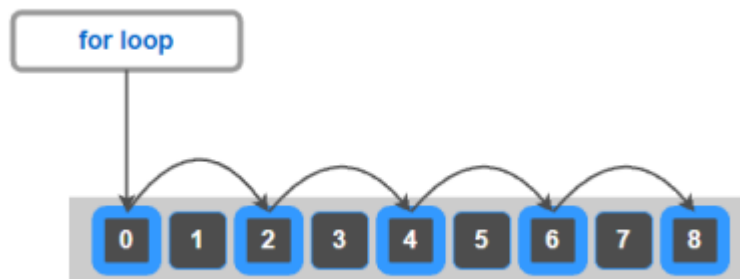
Output:

```

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)
```

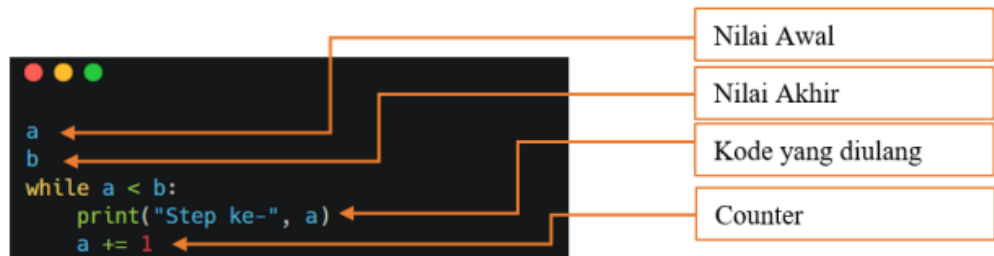
Output:

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

2. STRUKTUR WHILE

Pada struktur pengulangan *while* kondisi akan diperiksa di bagian awal. Hal ini tentu menyebabkan kemungkinan bahwa apabila ternyata kondisi yang kita definisikan tidak terpenuhi (bernilai salah), maka proses pengulangan pun tidak akan pernah dilakukan.

Bentuk umum dari struktur while :



Sebagai pembandingan dengan struktur pengulangan for, maka disini dituliskan kembaliprogram yang akan menampilkan teks “Hallo World!” dengan menggunakan struktur while.

```
1 i = 0
2 while i <= 7:
3     print("Hallo World!")
4     i += 1
```

Output:

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

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
```

3. 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\Diva> &
1
2
skip
4
skip
6
skip
8
skip
10
skip
```

II. GUIDED

- 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

```
ster 2/Praktikum Pemograman 1/Prakti
Masukkan Username fauzanwahyum
Masukkan Password 210310
Sugeng Rawuh fauzanwahyum !
PS D:\Tugas ITTP\Semester 2\Praktiku
```

- Jika username dan password salah lebih dari 3X

```
Pemograman 1/Praktikum 4/Latihan_login.py"
Masukkan Username fauzan
Masukkan Password 1234
Login raiso, Dang dijaal maneh
Masukkan Username fauzanwm
Masukkan Password 123456
Login raiso, Dang dijaal maneh
Masukkan Username fauzanwk
Masukkan Password 12456
Login raiso, Dang dijaal maneh
Masukkan Username fauzanwk
Masukkan Password 1456
Jajalen Sandi karo Password mungkin enek seng salah!
PS D:\Tugas ITTP\Semester 2\Praktikum Pemograman 1\Prakt
```

- B. Buatlah sebuah program mencari bilangan genap, dimana inputan berasal dari user adalah berupa range maksimal. Sehingga output

yang dihasilkan adalah sebagai berikut:

```
Masukkan Bilangan: 30
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
Step Ke- 20
Step Ke- 22
Step Ke- 24
Step Ke- 26
Step Ke- 28
PS D:\Tugas ITTP\Semester 2\Praktikum
```

- 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:

```
Penugasan 1/Praktikum 4/Latihan_FPB.
-----FPB-----
Masukkan Angka Pertama: 12
Masukkan Angka Kedua: 72
FPB = 12
PS D:\Tugas ITTP\Semester 2\Praktikum
```

III. UNGUIDED

TASK 1

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

- Source Code :

```

Task1.py > ...
1 print("-----Program Menghitung Sederhana-----") "Menghitung
2 bil = int(input("Masukkan Angka: ")) "Masukkan": Unknown word.
3 jmlh = 0 "jmlh": Unknown word.
4 smdgn = "" "smdgn": Unknown word.
5
6 while bil > 0:
7     jmlh += bil "jmlh": Unknown word.
8     smdgn += str(bil) "smdgn": Unknown word.
9     if bil != 1:
10         smdgn += " + " "smdgn": Unknown word.
11     bil -= 1
12
13 print("Hasil: ", smdgn, "=", jmlh) "Hasil": Unknown word.
14

```

- Output :

```

PS D:\Tugas ITPP\Semester 2\Praktikum Pemograman 1\Praktikum 4> & C:/Users/hp/AppData/Local/Microsoft/windows/Apps/python3.11.exe "d:/Tugas ITPP/Semester 2\Praktikum Pemograman 1\Praktikum 4\Task1.py"
-----Program Menghitung Sederhana-----
Masukkan Angka: 27
Hasil: 27 + 26 + 25 + 24 + 23 + 22 + 21 + 20 + 19 + 18 + 17 + 16 + 15 + 14 + 13 + 12 + 11 + 10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 = 378
PS D:\Tugas ITPP\Semester 2\Praktikum Pemograman 1\Praktikum 4>

```

TASK 2

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

- Source Code :

```

Task2.py > ...
1 print("-----Program Menghitung Pangkat-----") "M
2
3 b = int(input("Masukkan bilangan: ")) "Masukkan": Unknown word.
4 c = int(input("Masukkan pencacah: ")) "Masukkan": Unknown word.
5 hp = 1
6
7 for i in range(c):
8     hp *= b
9
10 print("Hasil pangkat adalah =", hp) "Hasil": Unknown word.
11

```

- Output :

```

PS D:\Tugas ITPP\Semester 2\Praktikum Pemograman 1\Praktikum 4> & C:/Users/hp/AppData/Local/Microsoft/windows/Apps/python3.11.exe "d:/Tugas ITPP/Semester 2\Praktikum Pemograman 1\Praktikum 4\Task2.py"
-----Program Menghitung Pangkat-----
Masukkan bilangan: 49
Masukkan pencacah: 2
Hasil pangkat adalah = 2401
PS D:\Tugas ITPP\Semester 2\Praktikum Pemograman 1\Praktikum 4>

```

TASK 3

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

- Source Code :

```
Task3.py > ...
3 print("-----Program Menghitung KPK-----")
4 bil1 = int(input("Masukkan bilangan pertama: "))
5 bil2 = int(input("Masukkan bilangan kedua: "))
6
7 i = 1
8 while True:
9     if i % bil1 == 0 and i % bil2 == 0:
10         kpk = i
11         break
12     i += 1
13
14 print("KPK dari", bil1, "dan", bil2, "adalah", kpk)
15
```

- Output :

```
PS D:\Tugas ITTP\Semester 2\Praktikum Pemograman 1\Praktikum 4> & C:/Use
um Pemograman 1/Praktikum 4/Task3.py
-----PROGRAM MENENTUKAN KPK-----
Masukkan bilangan pertama: 8
Masukkan bilangan kedua: 12
KPK dari 8 dan 12 adalah 24
PS D:\Tugas ITTP\Semester 2\Praktikum Pemograman 1\Praktikum 4> |
```