

Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2021/2022



NIM	71210695
Nama Lengkap	Cahaya Sampebua
Minggu ke / Materi	05 / Struktur Kontrol Perulangan

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
2022

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.

Definisi Perulangan

Perulangan **for** pada python adalah perintah yang digunakan untuk melakukan literasi dari sebuah nilai sequence atau data koleksi pada python seperti **List, Tuple, String** dan lain-lain .

For pada python memiliki perilaku yang berbeda dengan for pada kebanyakan bahasa pemrograman yang lain, karena pada python ia sangat berkaitan dengan data *sequence* atau data kolektif. Mungkin kalau dibandingkan dengan bahasa lain, for pada python lebih dikenal sebagai foreach.

Bentuk Perulangan for

Perulangan dalam bahasa pemrograman berfungsi menyuruh komputer melakukan sesuatu secara berulang-ulang. Terdapat dua jenis perulangan dalam bahasa pemrograman python, yaitu perulangan dengan **for** dan **while**.

Perulangan for

Bentuk umum:

for indeks in range(banyak_perulangan):

 # jalankan source code ini

 # jalankan juga source code ini

Contoh program:

```
File Edit Format Run Options Window Help
#Pertama kita menentukan berapa banyak perulangannya sebanyak 5 kali
mengulang = 5

for i in range(mengulang):
    print(f"Perulangan ke-{i}")
|
```

Variabel `i` berfungsi untuk menampung indeks, dan fungsi `range()` berfungsi untuk membuat list dengan range dari 0-10. Fungsi `str()` berfungsi merubah tipe data integer ke string.

Outputnya:

```
Perulangan ke-0
Perulangan ke-1
Perulangan ke-2
Perulangan ke-3
Perulangan ke-4
```

Bentuk Perulangan While

Perulangan while pada python adalah proses pengulangan suatu blok kode program selama sebuah kondisi terpenuhi. Singkatnya, perulangan while adalah perulangan yang bersifat *indefinite* alias tidak pasti, atau bahkan tidak terbatas. Sebuah blok kode akan dilakukan terus-menerus selama suatu kondisi terpenuhi.

Syntax while:

while <kondisi>:

blok source code yang akan diulang-ulang

Contoh program:

```
File Edit Format Run Options Window Help
i = 1
while i <= 4:
    print('Belajar while PrakAIPro GD!')
    i += 1
```

Variabel `i` sebagai inisiasi, atau dengan sebutan lain variabel counter yang nilainya akan terus dipakai selama program berjalan untuk menentukan jumlah berapa kali loop. Dan pada **while i <= 4**: inilah awal dimulainya perulangan selama nilai `i` kurang dari sama dengan 4. Artinya selama belum terpenuhi kondisi akhirnya, maka loop akan terus berjalan sampai akhir. Ketika nilai terpenuhi, yaitu `i` kurang dari sama dengan delapan maka sebanyak itu perulangan terjadi dan sebanyak itu juga python menjalankan perintah `print('Belajar while PrakAIPro')`.

Outputnya:

```
Belajar while PrakAlPro GD!  
Belajar while PrakAlPro GD!  
Belajar while PrakAlPro GD!  
Belajar while PrakAlPro GD!  
>>>
```

While loop adalah perulangan uncountable atau perulangan yang jumlah proses pengulangannya tidak ditentukan. Ia akan menjalankan baris kode di dalam blok kodenya secara terus menerus selama masih memenuhi ekspresi yang sudah ditentukan sebelumnya, yang berarti ia akan terus mengulang selama kondisi bernilai TRUE.

Berikut adalah contoh penerapan perulangan while loop dalam bahasa Python:

```
print("Program Bilangan Bulat")  
i = 1;  
x = int(input("Masukkan blangan bulat x = "));  
while i <= x:  
    print(i);  
    i=i+1;
```

output program di atas:

```
Program Bilangan Bulat  
Masukkan blangan bulat x = 6  
1  
2  
3  
4  
5  
6  
,
```

Pertama-tama, kita inisialisasi nilai variabel i=1, kemudian dilakukan perulangan dari i=1 sampai nilai i kurang dari atau sama dengan 5. Perulangan while di atas akan di ulang sebanyak 6 kali, mulai dari i = 1, i = 2, i = 3, i = 4, i = 5, hingga i = 6. Ketika nilai variabel counter i sudah mencapai 7, maka kondisi while i <= 6 tidak terpenuhi lagi (FALSE), sehingga perulangan berhenti.

Penggunaan Break dan Continue

a. Break Pada Python

Ketika membuat perulangan, kadang kita ingin keluar dari perulangan sebelum waktunya. Dengan kata lain, **break** adalah perintah khusus yang dipakai untuk memaksa sebuah perulangan berhenti sebelum waktunya. Perintah **break** ini bisa dipakai dalam perulangan **while** maupun perulangan **for** Python.

Contoh penggunaan **break**:

```
File Edit Format Run Options Window
for i in range(1, 15):
    if i == 10:
        break
    print(i)
```

Bisa dilihat bahwa angka yang sebelumnya dicetak 14 kali sekarang hanya menjadi 9 kali saja. Hal ini disebabkan oleh pemanggilan **break** statement ketika nilai **i** sama dengan 10. Ketika **break** dipanggil maka perintah dibawahnya (dalam kasus ini **print**) tidak akan dieksekusi. Bukan hanya itu saja, perulangan juga dianggap selesai.

b. Continue Pada Python

Continue berfungsi untuk lompat ke iterasi selanjutnya tanpa harus mengeksekusi sisa kode yang ada di bawahnya. Perbedaan utama dari **break** dan **continue** adalah jika **break** akan menghentikan perulangan secara total, sedangkan **continue** hanya akan lompat ke iterasi selanjutnya.

Contoh penggunaan **continue**:

```
File Edit Format Run Option:
for i in range(1, 15):
    if i == 10:
        continue
    print(i)
```

Outputnya:

```
1
2
3
4
5
6
7
8
9
11
12
13
14
```

Dari output yang dihasilkan angka **10** tidak dicetak hal ini disebabkan karena pemanggilan perintah **continue**. Jadi ketika kita memasuki kondisi dimana **i** sama dengan **10**, maka perintah **continue** akan dijalankan. Pemanggilan ini akan menyebabkan iterasi berlanjut ke nilai **i** selanjutnya secara tiba-tiba. Oleh karena itu semua perintah dibawahnya (dalam hal ini **print**) tidak akan dieksekusi sehingga angka **10** tidak dicetak.

Konversi dari Bentuk for Menjadi Bentuk while

Perulangan `for` disebut *counted loop* (perulangan yang terhitung), sementara perulangan `while` disebut *uncounted loop* (perulangan yang tak terhitung). Perbedaannya adalah perulangan `for` biasanya digunakan untuk mengulangi kode yang sudah diketahui banyak perulangannya.

Contoh perulangan `for`:

```
File Edit Format Run Options
for i in range(11, 16):
    print(i)
```

Outputnya:

```
11
12
13
14
15
>>
```

Berdasarkan tampilan output dari kedua jenis perulangan `for` dan `while`, maka dapat diidentifikasi bahwa perulangan tersebut dimulai dari 11, berakhir di 15, stepnya adalah 1. Untuk dapat melihat perbedaan penerapan kedua perulangan tersebut maka dibuat contoh hasil program (Output) yang sama. Berikut ini adalah contoh konversi kode program perulangan `for` ke perulangan yang menggunakan perintah `while`.

```
File Edit Format Run
#awal
i = 11

#kondisi akhir
while i <= 15:
    print(i)

#step
    i = i + 1
```

Outputnya:

```
===
11
12
13
14
15
>>>
```

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.

Link GitHub → <https://github.com/cahaya-sampebua/Latihan-Mandiri-Pertemuan-5-ParakAIPro.git>

SOAL 1

```
#Soal 5.1

def perkalian(bil1,bil2):
    print(bil1,"x", bil2, "=", end=' ')
    for i in range(bil1):
        if i == bil1 - 1:
            print(bilang2,"=", bil1 * bil2)
        else:
            print(bilang2,"+", end=' ')

bilang1 = int(input("Masukan bilangan pertama yang ingin dikali = "))
bilang2 = int(input("Masukan bilangan kedua yang ingin dikali = "))

perkalian(bilang1,bilang2)
```

```
Masukan bilangan pertama yang ingin dikali = 6
Masukan bilangan kedua yang ingin dikali = 5
6 x 5 = 5 + 5 + 5 + 5 + 5 + 5 = 30
```

```
Masukan bilangan pertama yang ingin dikali = 7
Masukan bilangan kedua yang ingin dikali = 10
7 x 10 = 10 + 10 + 10 + 10 + 10 + 10 + 10 = 70
```

Penjelasan Source Code :

Langkah pertama :

Pertama kita buat fungsi dengan nama perkalian

```
> def cek_angka (bil1,bil2,bil3):
```

Langkah kedua :

Setelah kita membuat fungsinya kita print bilangan yang akan dikalikan dan supaya angka selanjutnya tetap pada baris yang sama kita menambahkan end=" ".

```
> print(bil1,"x", bil2, "=", end=' ')
```

Langkah ketiga :

Setelah itu kita buat perulangannya kita disini memakai perulangan for dengan fungsi range

```
> for i in range(bil1):
```

Didalam for kita buat suatu kondisi yaitu jika i sama dengan parameter pertama dikurangi 1 maka print variable bilang2 = perkalian antara bilanga1 dan bilanga2 , selain dari itu maka print variable bilang2 +

```
>     if i == bil1 - 1:
```

```
>         print(bilang2,"=", end=' ')
```

```
>         print(bil1 * bil2)
```

```
>     else:
```

```
>         print(bilang2,"+", end=' ')
```

Langkah keempat :

Setelah kita selesai membuat fungsinya kita buat variable inputan untuk user nya dan kita panggil fungsinya

```
> bilang1 = int(input("Masukan bilangan pertama yang ingin dikali = "))
```

```
> bilang2 = int(input("Masukan bilangan kedua yang ingin dikali = "))
```

```
> perkalian(bilang1,bilang2)
```


SOAL 2

#Soal 5.2

```
def ganjil(ats,bwh):
    if bwh > ats:
        while bwh != ats:
            if bwh % 2 == 1:
                if bwh == ats + 2 :
                    print(bwh-2, end='.')
                elif bwh == ats + 3:
                    print(str(bwh)+", "+str(bwh-2), end='.')
                    break
                else:
                    print(bwh, end=',')
            bwh = bwh - 1
        else:
            while bwh != ats:
                if bwh % 2 == 1:
                    if bwh == ats - 2 :
                        print(bwh+2, end='.')
                    elif bwh == ats - 3 :
                        print(str(bwh)+", "+str(bwh+2), end='.')
                        break
                    else:
                        print(bwh, end=',')
                bwh = bwh + 1
            else:
                pass

bawah = int(input("Masukan nilai bawah = "))
atas = int(input("Masukan nilai atas = "))
ganjil(atas,bawah)
```

```
Masukan nilai bawah = 10
Masukan nilai atas = 30
11,13,15,17,19,21,23,25,27,29.
```

```
Masukan nilai bawah = 97
Masukan nilai atas = 82
97,95,93,91,89,87,85,83.
```

Penjelasan Source Code :

Pertama kita buat fungsi dengan nama ganjil

```
> def ganjil(ats,bwh):
```

Setelah kita membuat fungsinya didalam fungsi kita buat suatu perkondisian yaitu jika parameter 2 lebih dari parameter 1 maka akan ada perulangan, perulangan yang kita pakai disini yaitu while, yang dimana kondisi whilenya parameter 2 != parameter 1, didalam preulangannya kita masukan sebuah kondisi didalam kondisi yaitu jika parameter 2 % 2 == 1 maka, kondisi didalamnya jika parameter 2 == parameter 1 + 2 maka print parameter 2 – 2 “.”, jika parameter 2 == parameter 1 + 3 maka print parameter 2 “,” parameter 2 – 2 “.” Dan langsung di break, selain dari itu maka print parameter 2 “.”, dan kita buat variabel lagi seperti kondis while supaya ada whilenya nanti berhenti yaitu parameter 2 = parameter 2 – 1.

```

> if bwh > ats:
>     while bwh != ats:
>         if bwh % 2 == 1:
>             if bwh == ats + 2 :
>                 print(bwh-2, end='.')
>             elif bwh == ats + 3:
>                 print(str(bwh)+", "+str(bwh-2), end='.')
>                 break
>             else:
>                 print(bwh, end=',')
>         bwh = bwh - 1

```

Setelah kita membuat kondisi yang pertama yaitu jika parameter 2 > parameter 1 kita langsung saja pakai else karena hanya ada dua kondisi. didalam kondisi tersebut kita masukan perulangan, perulangan yang kita pakai disini yaitu while, yang dimana kondisi whilenya parameter 2 != parameter 1, didalam preulangannya kita masukan sebuah kondisi didalam kondisi yaitu jika parameter 2 % 2 == 1 maka, kondisi didalamnya jika parameter 2 == parameter 1 - 2 maka print parameter 2 + 2 ".", jika parameter 2 == parameter 1 - 3 maka print parameter 2 ",", parameter 2 + 2 "." Dan langsung di break,

selain dari itu maka print parameter 2 ".", dan kita buat variabel lagi seperti kondisi while supaya ada whilenya nanti berhenti yaitu parameter 2 = parameter 2 + 1.

> else:

> while bwh != ats:

> if bwh % 2 == 1:

> if bwh == ats - 2 :

> print(bwh+2, end='.')

> elif bwh == ats - 3:

> print(str(bwh)+", "+str(bwh+2), end='.')

> break

> else:

> print(bwh, end=',')

> bwh = bwh + 1

Setelah kita membuat kondisinya selanjutnya kita buat variable untuk input usernya dan memanggil fungsinya

> bawah = int(input("Masukan nilai bawah = "))

> atas = int(input("Masukan nilai atas = "))

ganjil(atas,bawah)

SOAL 3

#Soal 5.3

```
print("Program penghitung IPS Mahasiswa")
try:
    sks = int(input("Berapa jumlah mata kuliah? "))
    i = 0
    jumlah = 0
    try:
        while i < sks:
            mk = input("Nilai MK %d: " % (i+1))
            if mk == "A":
                nilai = 4
            elif mk == "B":
                nilai = 3
            elif mk == "C":
                nilai = 2
            elif mk == "D":
                nilai = 1
            else:
                print("Nilai yang anda masukan salah atau tidak ada", end="")
                break
            i = i + 1
            jumlah += nilai
    except:
        print("Nilai yang anda masukan salah", end="")
    rata_rata = jumlah / sks
    print("\nNilai IPS anda semester ini %0.2f" % rata_rata)
except:
    print("Anda salah memasukan input")
```

```
Program penghitung IPS Mahasiswa
Berapa jumlah mata kuliah? 6
Nilai MK 1: A
Nilai MK 2: B
Nilai MK 3: C
Nilai MK 4: A
Nilai MK 5: D
Nilai MK 6: C
```

Penjelasan Source Code :

Pertama kita buat judul dari programnya

```
> print("Program penghitung IPS Mahasiswa")
```

Setelah kita membuat judulnya kita menggunakan **try** dan **except** untuk menangani kesalahan input setelah kita membuatnya kita buat variable untuk inputan user setelah itu kita buat variable untuk kondisi whilenya nanti yaitu `i = 0` dan jumlah dari nilai nanti yaitu `jumlah = 0`

```
> try:
```

```
> sks = int(input("Berapa jumlah mata kuliah? "))
```

```
> i = 0
```

```
> jumlah = 0
```

Setelah itu kita buat lagi **try** dan **except** untuk menangani kesalahan input yang ada didalam whilenya, setelah itu kita buat while dengan kondisi $i < \text{sks}$, didalam while kita buat lagi variable untuk inputan nilai MK user setelah kita membuat inputnya kita buat beberapa kondisi yaitu jika inputan nilai $\text{MK} == \text{A}$ maka nilai = 4, jika inputan nilai $\text{MK} == \text{B}$ maka nilai = 3, jika inputan nilai $\text{MK} == \text{C}$ maka nilai = 2, jika inputan nilai $\text{MK} == \text{D}$ maka nilai = 1, selain dari pada itu maka print nilai yang anda masukan salah dan langsung di break, setelah itu kita buat variable lagi yaitu $i = i + 1$ untuk whilenya berhenti dan kita juga masukan operator penugasan jumlah $+=$ nilai supaya variable jumlah ditambah dengan hasil dari nilai nanti

```
> try:
>     while i < sks:
>         mk = input("Nilai MK %d: " % (i+1))
>         if mk == "A":
>             nilai = 4
>         elif mk == "B":
>             nilai = 3
>         elif mk == "C":
>             nilai = 2
>         elif mk == "D":
>             nilai = 1
>         else:
>             print("Nilai yang anda masukan salah atau tidak ada", end="")
>             break
>         i = i + 1
>         jumlah += nilai
> except:
>     print("Nilai yang anda masukan salah", end="")
```

Setelah kita membuat perulangannya kita membuat variable rata-rata dari jumlah sksnya dan setelah itu kita print hasilnya

```
> rata_rata = jumlah / sks
> print("\nNilai IPS anda semester ini %0.2f" % rata_rata)

> except:
print("Anda salah memasukan input")
```