



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231058
Nama Lengkap	Michael Chandra Mahanaim
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.

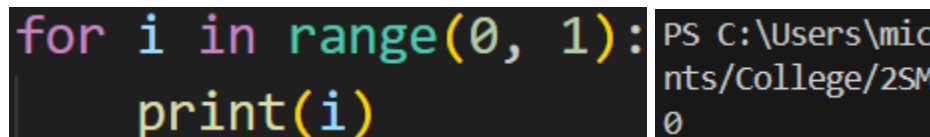
PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA

BAGIAN 1: MATERI MINGGU INI (40%)

MATERI 1: For loop

For loop menggunakan `for i in range()`: untuk mengulangi sintaks yang ada dalam loopnya sebanyak kondisi yang ditetapkan pada range, dimana range memiliki komponen (start, stop, step).

- Start: Permulaan start / variabel integer awal untuk perhitungan. Disimpan dalam `i` jika sintaks `for` dalam bentuk: `[for i in range(1)]`

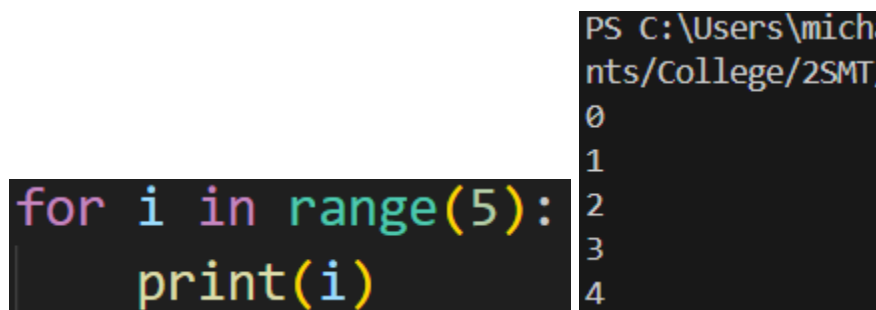


```
for i in range(0, 1):  
    print(i)
```

PS C:\Users\mic
nts/College/2SM
0

Gambar contoh penggunaan for loop dimana start disimpan di `i`

- Stop: Akhir dari loop, jika sintaks dalam bentuk `[for i in range(5)]` maka start secara default akan mulai dari 0 dan 5 akan dijadikan stop point nya.

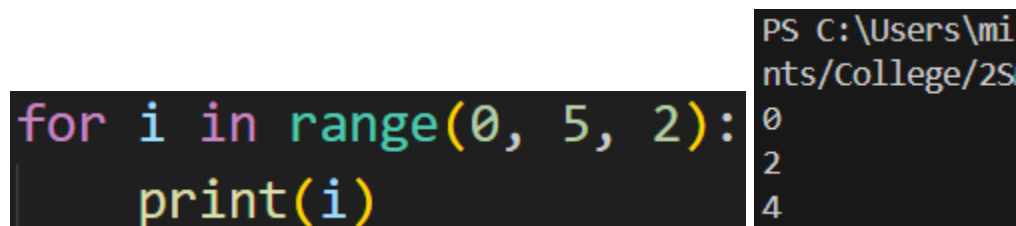


```
for i in range(5):  
    print(i)
```

PS C:\Users\micha
nts/College/2SMT/
0
1
2
3
4

Gambar contoh penggunaan stop sebagai default dalam `range()`

- Step: Seberapa banyak start dapat menuju ke stop, secara default adalah +1, jika sintaks dalam bentuk `[for i in range(0, 5, 2)]` maka start akan +2 setiap kali loop selesai.



```
for i in range(0, 5, 2):  
    print(i)
```

PS C:\Users\mi
nts/College/2S
0
2
4

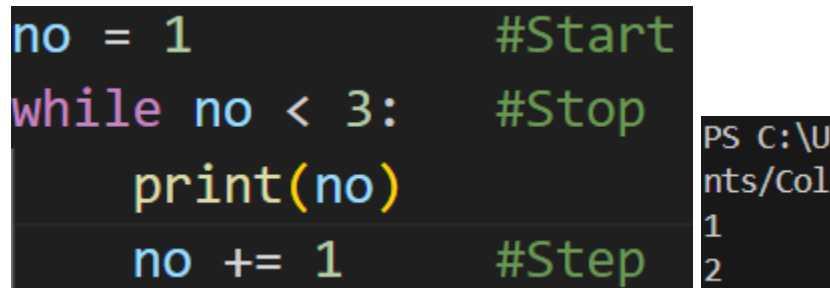
Gambar contoh penggunaan step untuk kondisi yang lebih kompleks

For loop biasa digunakan untuk kode yang sudah diketahui hasil akhirnya / hasil akhirnya dapat diprediksi. Biasanya hasil yang menggunakan angka. For loop tidak bisa digunakan untuk loop yang menggunakan kondisi boolean atau logical. Loop seperti itu hanya bisa digunakan dalam While loop.

MATERI 2: While loop

While loop hampir sama seperti for loop, hanya memerlukan sedikit konfigurasi agar dapat memenuhi kondisi untuk start, stop dan step. While loop juga mempunyai keuntungan yang dapat menggunakan expression dan logical condition sebagai penanda untuk stop.

Contoh dengan expression:



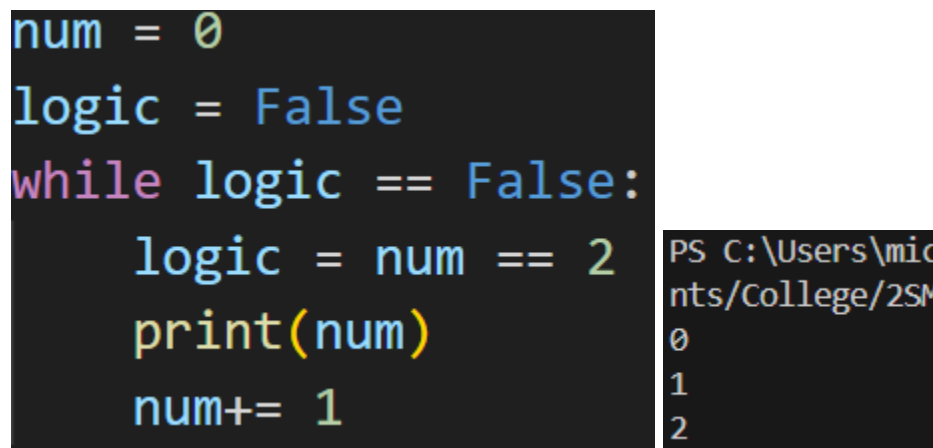
```
no = 1           #Start
while no < 3:    #Stop
    print(no)
    no += 1      #Step
```

PS C:\Us
nts/Col
1
2

Gambar penggunaan dan output while loop menggunakan expression

Dari gambar diatas, pemberian start, stop, dan step menggunakan while loop agak berbeda. Start dimulai diluar loop untuk base start awal, stop digunakan dalam conditional loop dari while loop (dalam expresi maka selama no kurang dari 3 maka loop akan berjalan terus), dan Step menggunakan no += 1. Step jangan sampai lupa karena jika tidak ada maka akan menjadi infinite loop dan tidak akan berhenti.

Contoh dengan logical:



```
num = 0
logic = False
while logic == False:
    logic = num == 2
    print(num)
    num += 1
```

PS C:\Users\mic
nts/College/2SM
0
1
2

Gambar while loop dan output menggunakan logical

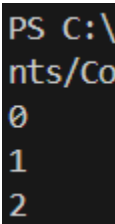
Kode ini menggunakan cara yang sama seperti expression untuk mengeluarkan output, bedanya hanya dalam praktisnya, kode ini menggunakan logical sebagai stopnya, dimana hanya akan berhenti jika kondisi variabel "logic" sudah menjadi True dan dapat keluar dari while loop.

While loop hanya bekerja jika kondisi yang di set menghasilkan True (dari 2 contoh di atas), dan hanya akan berhenti jika kondisi yang di set menghasilkan False. While loop biasa digunakan untuk hasil yang belum diketahui atau untuk ekspresi tertentu.

MATERI 3: Break and Continue

Break adalah sintaks yang digunakan dalam loop untuk memotong loop lebih awal sebelum sampai ke tujuan stopnya, sedangkan Continue digunakan untuk melompati sebuah langkah dari sebuah loop. Contoh penggunaan untuk masing-masing loop ada di bawah ini:

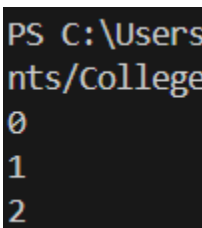
```
i = 0
while i < 5:
    print(i)
    i += 1
    if i == 3:
        break
```



```
PS C:\Users\michael\Documents\College\2SMT\Python>
0
1
2
```

Gambar penggunaan dan output break dalam while loop

```
for i in range(5):
    print(i)
    if i == 3:
        break
```



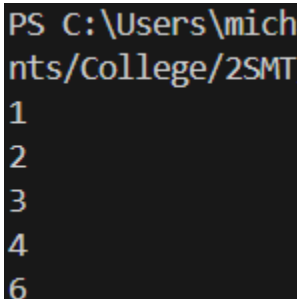
```
PS C:\Users\michael\Documents\College\2SMT\Python>
0
1
2
```

Gambar penggunaan dan output break dalam for loop

Kedua kode di atas fungsinya sama, hanya penggunaan dalam while dan for loopnya yang berbeda sedikit. Loop akan langsung berhenti saat loop ketiga karena ada perintah if dimana jika $i == 3$ maka break. Kode ini sangat berguna untuk menghentikan loop jika menginginkan batas maksimum melakukan sesuatu dalam kode atau berdasarkan input user.

Selanjutnya adalah penggunaan dalam continue, contohnya untuk masing-masing loop sebagai berikut:

```
n = 1
while n < 7:
    if n == 5:
        n += 1
        continue
    print(n)
    n += 1
```



```
PS C:\Users\michael\Documents\College\2SMT\Python>
1
2
3
4
6
```

Gambar penggunaan dan output continue dalam while loop

```
for i in range(1,7):
    if i == 5:
        continue
    print(i)
```

```
PS C:\Users\mich
nts/College/2SMT
1
2
3
4
6
```

Gambar penggunaan dan output continue dalam for loop

Sama seperti saat break, kedua kode diatas memiliki fungsi yang sama, hanya perbedaan ada pada penulisan sintaks dalam while loop untuk menghindari infinite loop. Dalam while loop `n += 1` ditulis dua kali agar saat kondisi if terpenuhi tidak terjadi infinite loop karena step yang diluar if di skip oleh continue. Sedangkan dalam for tidak perlu terlalu rumit karena start, step, dan stop sudah dikalkulasi oleh `range()`.

MATERI 4: While-For Conversion

Dari contoh dalam materi 3, diketahui bahwa semua for loop bisa diubah menjadi while loop, tetapi tidak semua while loop bisa diubah menjadi for loop. Karena while loop dapat bermain dengan expression dan logic, maka while dapat digunakan lebih fleksibel dalam situasi tertentu, tetapi lebih rumit dan jika tidak hati-hati dapat menghasilkan infinite loop. Sebaliknya, for loop lebih streamlined dan dapat dibilang lebih mudah dibandingkan while loop jika hanya bermain angka. Beberapa contoh dari konversi for loop dan while loop adalah:

```
for i in range(0, 11):
    print(i)
```

```
n = 0
while n < 11:
    print(n)
    n += 1
```

Gambar perbedaan / konversi for loop dan while loop

Dalam for loop terlihat jelas dalam `range()` start, stop, dan step dengan kondisi yang diinginkan hanya membutuhkan 1 line saja, sedangkan while loop memerlukan 3 line untuk start, stop, dan step. Tergantung pengalaman programmer, kedua ini adalah cara yang valid untuk melakukan sebuah loop.

Di sisi lainnya ada juga while loop yang tidak bisa di konversi menjadi for loop, contohnya:

```
num = 0
logic = False
while logic == False:
    logic = num == 2
    print(num)
    num+= 1
```

Contoh while loop yang tidak bisa dikonversi ke for loop

Gambar di atas diambil dari gambar materi 2 untuk contoh while loop logical. Karena for loop spesialisasinya dalam angka / indeks, maka untuk logical comparison seperti ini tidak bisa. Contoh lain juga berupa string comparison, seperti di bawah ini:

```
inp = input("Letter: ")
while inp == "A":
    inp = input("Letter: ")
```

Gambar contoh while loop yang tidak bisa dikonversi ke for loop

Dalam kode di atas akan meminta input dari user berupa sebuah huruf, jika huruf itu A, maka while loop akan diaktifkan dan meminta terus sebuah huruf dari user. Jika user menginput A lagi maka while loop akan terus berjalan, sebaliknya jika input user bukan A maka while loop akan berhenti.

Ada juga 1 keuntungan dari While loop, yaitu dengan multiple expression, dimana ada 2 persyaratan yang harus dipenuhi agar while loop terus berjalan. Contoh sebagai berikut:

```
m = 0
n = 0
while (n > -4) and (m < 5):
    m += 1
    n -= 1
    print(m, n)
```

PS C:\Users
nts/College
1 -1
2 -2
3 -3
4 -4

Gambar multiple expression dalam while loop dan outputnya

Kode ini menggunakan 2 kondisi serta logical "and" sebagai conditional stopnya (and adalah jika salah satu dari kedua kondisi tidak terpenuhi maka akan mengeluarkan False). Dari contoh diatas variabel m seharusnya dapat melakukan loop 1 kali lagi, tetapi karena variabel n sudah tidak memenuhi kondisi maka kondisi dari "and" akan mengeluarkan False yang menghentikan while loop.

BAGIAN 2: LATIHAN MANDIRI (60%)

SOAL 1

A. Source Code

```
Welcome | Question1.py U x | Question2.py U | Question3.py U | S
Question1.py > mult
1 def mult(n1, n2):
2     count = 0
3     r = 0
4     print(f"{n1} x {n2} =", end="")
5     while count < n1:
6         r += n2
7         count += 1
8         print(f" {n2}", end="")
9         if count == n1:
10             continue
11         else:
12             print(" +", end="")
13     print(f" = {r}")
14 n1 = input("Number: ")
15 n2 = input("Number: ")
16
17 try:
18     n1 = int(n1)
19     n2 = int(n2)
20 except:
21     print("Please use numbers, not words")
22     exit()
23
24 mult(n1, n2)
```

B. Output Result

```
PS C:\Users\micha\Documents\College\2SMT\TPAP1\tugaspralp
nts\College\2SMT\TPAP1\tugaspralpro5\Question1.py
Number: 6
Number: 55
6 x 55 = 55 + 55 + 55 + 55 + 55 + 55 = 330
PS C:\Users\micha\Documents\College\2SMT\TPAP1\tugaspralp
nts\College\2SMT\TPAP1\tugaspralpro5\Question1.py
Number: 7
Number: 213
7 x 213 = 213 + 213 + 213 + 213 + 213 + 213 + 213 = 1491
```


C. Explanation

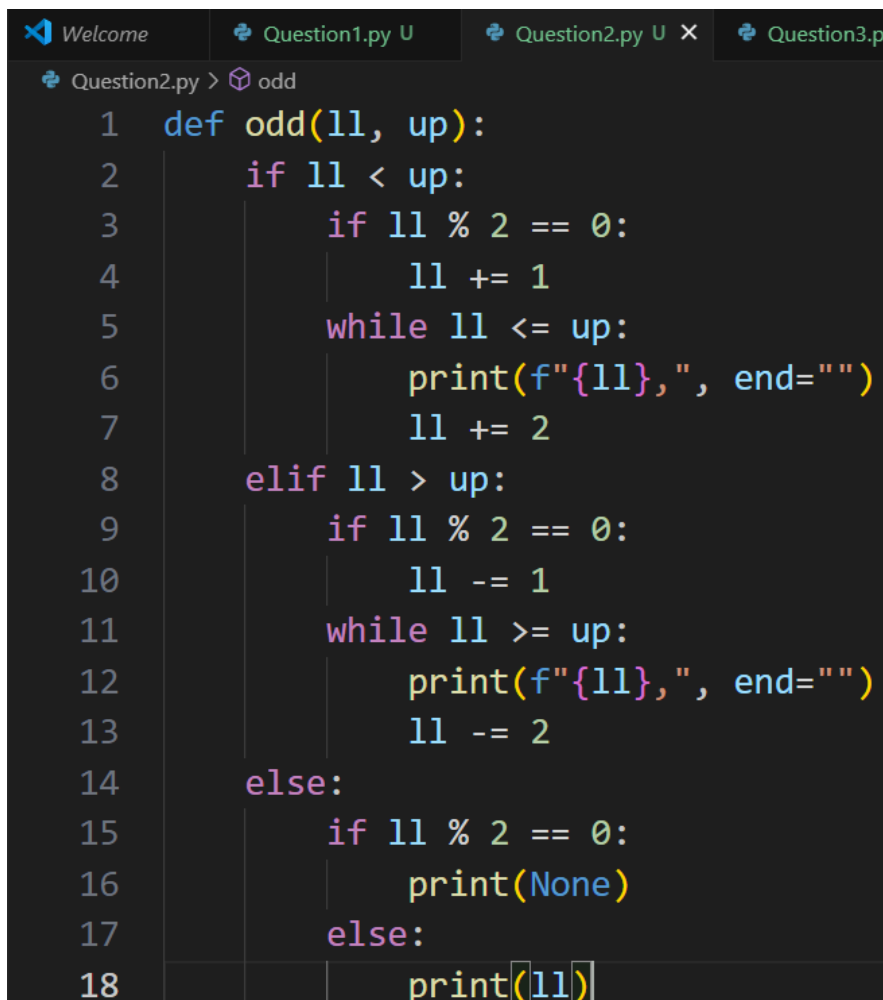
Kode ini berfungsi menggunakan 2 angka dari input user dan mengeluarkan output berupa visualisasi perkalian dalam bentuk penjumlahan berulang dan hasil dari penjumlahan / perkalian tersebut. Kode ini menggunakan komponen-komponen: input, fungsi, while loop, percabangan if else, dan try except.

Kode dimulai dari line 14 dimana diminta 2 input user berupa angka. Setelah itu akan masuk try except untuk mengubah input user dari string menjadi integer, karena jika user memasukkan input selain angka (semisal Dua puluh) maka kode dapat berhenti dan memberi user sebuah "Exception" error. Setelah lolos try except, maka fungsi "mult()" akan di call.

Fungsi mult() akan menggunakan while loop untuk membuat visualisasi perkalian dalam bentuk penjumlahan dimana 2 input yang diminta tadi akan dikeluarkan dalam bentuk print() berkali-kali sebanyak salah satu input dan pada akhirnya jumlah akhir akan di print juga.

SOAL 2

A. Source Code



```
1 def odd(ll, up):
2     if ll < up:
3         if ll % 2 == 0:
4             ll += 1
5         while ll <= up:
6             print(f"{ll},", end="")
7             ll += 2
8     elif ll > up:
9         if ll % 2 == 0:
10            ll -= 1
11        while ll >= up:
12            print(f"{ll},", end="")
13            ll -= 2
14    else:
15        if ll % 2 == 0:
16            print(None)
17        else:
18            print(ll)
```

```

19 in1 = input("Lower limit: ")
20 in2 = input("Upper limit: ")
21 try:
22     in1 = int(in1)
23     in2 = int(in2)
24 except:
25     print("Please use numbers, not words")
26     exit()
27 print("Ganjil: ", end="")
28 odd(in1, in2)

```

B. Output Result

```

PS C:\Users\micha\Documents\College\2SMT\TPAP1\tugaspralpro5
nts\College\2SMT\TPAP1\tugaspralpro5\Question2.py
Lower limit: 66
Upper limit: 100
Ganjil: 67,69,71,73,75,77,79,81,83,85,87,89,91,93,95,97,99,

```

C. Explanation

Kode ini berfungsi menggunakan 2 input dari user berupa angka dan mengeluarkan output berupa semua angka ganjil yang berada di antara kedua angka tersebut. Komponen dari kode ini berupa: input, fungsi, while loop, percabangan if else, dan try except. Input untuk meminta 2 input dari user, try except untuk mengatasi kesalahan input user, fungsi untuk mempermudah calling dan menyimpan proses, dan percabangan if else serta while loop untuk membuat proses output dengan print lebih mudah dan single stream.

SOAL 3

A. Source Code

```
Welcome | Question1.py U | Question2.py U | Question3.py U x | SrcCd.py U
Question3.py > point_check
1 def average(a, b, c, d, n):
2     total = ((a * 4) + (b * 3) + (c * 2) + (d * 1)) * 3
3     avrg = total / (n * 3)
4     return avrg
5
6 def point_check(n):
7     a = 0
8     b = 0
9     c = 0
10    d = 0
11    count = 0
12    for i in range(1, n+1):
13        point = input(f"Nilai MK {i}: ")
14        if point == "A":
15            a += 1
16        elif point == "B":
17            b += 1
18        elif point == "C":
19            c += 1
20        elif point == "D":
21            d += 1
22        else:
23            print("Invalid")
24        count += 1
25    result = round(average(a, b, c, d, count), 2)
26    print("Rata-rata = ", result)
27
28
29    amnt = input("Jumlah MK: ")
30
31    try:
32        amnt = int(amnt)
33    except:
34        print("Please use numbers, not words")
35        exit()
36
37    point_check(amnt)
```

B. Output Result

```
PS C:\Users\micha\Docu
● nts/College/2SMT/TPAP1
Jumlah MK: 7
Nilai MK 1: A
Nilai MK 2: B
Nilai MK 3: C
Nilai MK 4: D
Nilai MK 5: A
Nilai MK 6: A
Nilai MK 7: B
Rata-rata = 3.0
```

C. Explanation

Kode ini berfungsi dengan menggunakan input awal dari user berupa jumlah mata kuliah dan setelah itu akan meminta input lanjutan berupa nilai matkul tersebut dalam jangka nilai A-D yang setelah itu akan dikeluarkan output berupa rata-rata hasil nilai IPS dari semua inputan user. Komponen kode ini berupa: Input, fungsi, for loop, percabangan if else, round, dan try except.

Kode akan menggunakan fungsi `point_check()` untuk mengulang permintaan input berupa nilai MK sebanyak jumlah MK yang dimasukkan dengan menggunakan for loop range dimana start dimulai dari 1 dan stop saat Jumlah MK. Lalu fungsi `average()` akan dicall untuk melakukan kalkulasi hasil rata-rata nilai IPS dari inputan yang user berikan saat for loop pada `point_check()`, lalu hasil tersebut akan dibulatkan ke 2 angka dibelakang koma menggunakan `round()`, dan akhirnya dikeluarkan menggunakan `print`.