



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	<71230972>
Nama Lengkap	<Oktavian Christ Putranto>
Minggu ke / Materi	05 / 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
2024

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.

Perulangan

Dalam membuat program baik itu berupa program sederhana maupun program yang rumit kita akan selalu bertemu dengan 3 struktur control pemrograman yang berupa skensial, perulangan, dan percabangan, pada pertemuan di minggu sebelumnya kita sudah banyak mempelajari percabangan kini kita akan mempelajari perulangan pada dasarnya perulangan sendiri dapat dimanfaatkan untuk membuat program melakukan hal yang sama berulang kali, Melakukan suatu hal secara bertahap, di mana setiap tahap sebenarnya memiliki langkah yang sama, dan Mengakses sekumpulan data dalam suatu struktur data, misalnya: List, Tuple, Queue, Stack dan beberapa struktur data lainnya.

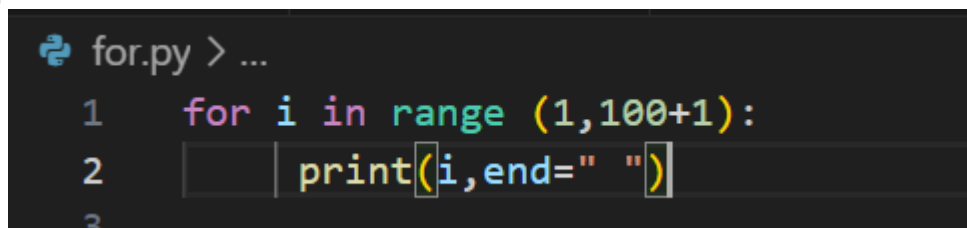
Pada Python, perulangan dapat dilakukan dengan menggunakan for, while maupun dengan cara rekursif. Untuk pertemuan ini, kita akan mempelajari struktur perulangan memakai for dan while.

Perulangan For

For sederhananya ialah sebuah bentuk perulangan dalam Bahasa pemrograman python yang biasa digunakan ketika batas akhir dari perulangan tersebut sudah di temukan, kondisi kondisi dalam perulangan For:

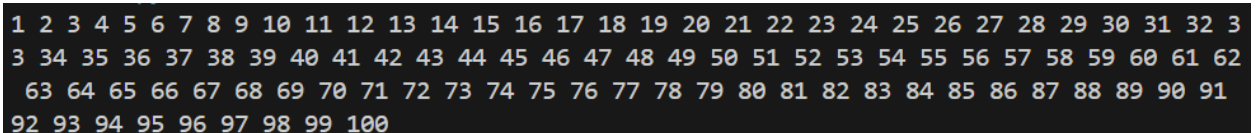
- Jumlah perulangan sudah diketahui sejak awal. Contoh sederhananya ketika kita diminta untuk membuat penomoran dari 1-100, maka kita dapat membuat perulangan for untuk menuliskannya secara berurutan

Input :



```
for.py > ...  
1   for i in range (1,100+1):  
2       print(i,end=" ")  
3
```

Output :



```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 3  
3 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62  
63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91  
92 93 94 95 96 97 98 99 100
```

Perulangan ini berguna untuk mengeluarkan output secara otomatis dari 1 sampai seratus menggunakan perulangan For ketika menggunakan perulangan For maka harus diperhatikan batas awal dan akhirnya ditandai dengan in range (1,100+1) yang berarti output akan menampilkan bilangan dari 1 sampai 100.

- Perulangan terjadi karena operasi yang sama pada suatu rentang data atau rentang nilai. Misalnya dalam mencari jumlah dari 100 bilangan pertama, maka secara berturut-turut dilakukan penjumlahan $1 + 2 + 3 + \dots + 100$. Berarti dilakukan dalam rentang mulai dari 1 sampai 100

Input :

```

for.py > ...
1  batas=int(input("batas:"))
2  tot=0
3  for i in range (1,batas+1):
4      if i<batas:
5          print (i,end='+')
6      if i>=batas:
7          print (batas,end="=")
8      tot=tot+i
9  print (tot,end='')
10

```

Output :

```

batas:100
1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20+21+22+23+24+25+26+27+28+29+30+31+32+3
3+34+35+36+37+38+39+40+41+42+43+44+45+46+47+48+49+50+51+52+53+54+55+56+57+58+59+60+61+62
+63+64+65+66+67+68+69+70+71+72+73+74+75+76+77+78+79+80+81+82+83+84+85+86+87+88+89+90+91+
92+93+94+95+96+97+98+99+100=5050

```

Pada perulangan ini kita akan membuat batas akhir dari bilangan yang ingin dijumlahkan dalam contoh batasnya ialah 100 sehingga pada perulangan for ketika angka lebih kecil dari batas maka akan muncul bilangan dalam range tersebut dan diakhiri dengan + fungsi ini termasuk dalam struktur percabangan yaitu `if i<batas: print (i,end='+')` kemudian ketika bilangan sudah melebihi batas maka bilangan akan diakhiri dengan = kedua fungsi ini ketika conditional pertama tidak terpenuhi yaitu `if i>=batas: print(batas,end="=")`. Setelah perulangan tersebut di selesaikan maka diakhiri dengan total nilai nya yaitu `tot=tot+i`.

Dari kedua contoh perulangan yang sudah saya buat dapat diperhatikan bahwa dalam setiap perulangan for yang dibuat maka diperlukan adanya bantuan fungsi `range()`, yang bentuknya sebagai berikut:

- `range(stop)`. Digunakan untuk menghasilkan rentang dari 0 sampai `stop-1`. Misalnya `range(6)`, berarti menghasilkan rentang 0-5.

- `range(start, stop, [step])`. Digunakan untuk menghasilkan rentang dari start, sampai stop dengan peningkatan sejumlah step.

Contoh penggunaan fungsi For in range :

```
bil.py > ...
1  for i in range(30, 1, -3):
2      print(i)
3
```

```
PS C:\Users\AS...
ts/kuliah/msys...
thon/bil.py"
30
27
24
21
18
15
12
9
6
3
```

Dalam contoh ini kita menetapkan range 30 sebagai titik awal, 1 sebagai titik stop dan, -3 menandakan jumlah step nya sehingga ketika dituliskan sebuah perulangan seperti ini maka program akan mengeluarkan output angka dari 30 sampai 1 dengan jarak pengurangan sebesar-3 tiap step nya, alhasil data output yang ditampilkan akan terlihat seperti contoh diatas.

Contoh lain :

```
bil.py > ...
1  for i in range(10):
2      print('halo')
3
```

```
thon/bil.py
halo
halo
halo
halo
halo
halo
halo
halo
halo
halo
```

Untuk contoh kedua kita dapat melihat bahwa bentuk perulangan for dapat menampilkan data string 'halo' sejumlah 10 kali Dimana hal ini dapat terjadi karena ketika range tidak diberi nilai start berarti dia akan mulai dari 0-10(batas akhir) dan kita dapat mengeluarkan output sesuai yang kita mau yaitu `print('halo')` selama perulangan masih berjalan.

Perulangan While

Perulangan dalam bentuk while biasa dilakukan ketika kita tidak mengetahui batas akhir dari program yang ingin dibuat sederhana program perulangan while akan membuat titik stop bergantung pada conditional yang dibuat terpenuhi atau tidak.

Contoh penggunaan while :

```
genap.py > ...
1  bilangan = 0
2  genap = False
3  while genap == False:
4      bilangan = int(input('Masukkan bilangan genap: '))
5      if bilangan % 2 == 0:
6          genap = True
7  print(bilangan, 'yang anda masukkan adalah bilangan genap')
```

```
Masukkan bilangan genap: 1
Masukkan bilangan genap: 13
Masukkan bilangan genap: 15
Masukkan bilangan genap: 2
2 yang anda masukkan adalah bilangan genap
```

Pada contoh diatas struktur perulangan while dibuat agar program terus berjalan sampai menemukan input user yang merupakan bilangan genap, hal ini dapat terjadi karena kita memakai persyaratan/conditional Dimana ketika `genap==false` maka perulangan akan terus berjalan dan di dalam perulangan tersebut kita akan mengecek input user Dimana ketika bilangan di modulus 2 = 0 maka genap akan menjadi True dan membuat perulangan berhenti jadi hingga bilangan tidak genap maka perulangan akan terus menerus terjadi dan program akan mengecek input user terus menerus, ketika perulangan selesai maka output akan berupa print bilangan genap tersebut.

Penggunaan Break dan Continue

Perulangan dapat dikontrol dengan menggunakan break dan continue. Secara umum break digunakan untuk menghentikan perulangan, sedangkan continue digunakan untuk melanjutkan perulangan ke iterasi berikutnya

Contoh awal :

```
break.py > ...
1  for i in range(1, 11):
2      print(i, end=' |')
3  print('Selesai')
```

```
thon/break.py"
1 2 3 4 5 6 7 8 9 10 Selesai
```

Penggunaan break:

```
break.py > ...
1  for i in range(1, 11):
2      if i == 5:
3          break
4      else:
5          print(i,end=' ')
6  print('Selesai')
```

1 2 3 4 Selesai

Dari dua contoh yang ada diatas dapat terlihat perbedaan output Dimana pada contoh pertama output yang ditampilkan adalah angka 1-10 setelah itu 'selesai' sedangkan pada contoh kedua angka yang ditampilkan ialah 1-4 kemudian 'selesai' perbedaan ini dapat terjadi karena pada contoh kedua diterapkan konsep Break Dimana ketika $i == 5$ maka break terjadi dan perulangan selesai sehingga angka yang ditampilkan hanyalah 1-4 kemudian selesai.

Contoh awal untuk penggunaan continue:

```
cont.py > ...
1  for i in range(1,6):
2      print(i,end=' ')
3  print('Selesai')
```

1 2 3 4 5 Selesai

Contoh setelah memakai continue:

```
cont.py > ...
1  for i in range(1,6):
2      if i == 3:
3          continue
4      else:
5          print(i,end=' ')
6  print('Selesai')
```

1 2 4 5 Selesai

Pada contoh pertama dan kedua Nampak perbedaan output Dimana ketika continue di pakai dalam perulangan maka output yang awalnya "1 2 3 4 5 selesai" menjadi "1 2 4 5 Selesai" hal ini terjadi karena pada dasarnya continue akan menskip bilangan yang bernilai 3 ditandai dengan $i == 3$: continue dan lanjut ke perulangan berikutnya hingga program selesai

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.

SOAL 1

Input:

```
lat_1.py > perkalian
1  def perkalian(bil_1,bil_2):
2      print (f"{bil_1}x{bil_2}",end="=")
3      for i in range (1,bil_1):
4          i=bil_2
5          print(i,end="+")
6      hasil=bil_1*bil_2
7      print(bil_2,end="=")
8      print(hasil)
9  bil_1=int(input("bilangan 1:"))
10 bil_2=int(input("bilangan 2:"))
11 perkalian(bil_1,bil_2)
```

Output :

```
bilangan 1:6
bilangan 2:5
6x5=5+5+5+5+5+5=30
PS C:\Users\ASUS VivoBook\Downloads\LapPrak5_vian
rs/ASUS VivoBook/Downloads/LapPrak5_vian
bilangan 1:7
bilangan 2:10
7x10=10+10+10+10+10+10+10=70
```

Pada soal pertama kita diminta untuk membuat program yang menerapkan perhitungan perkalian dengan menggunakan penjumlahan pada soal pertama saya membuat program dengan memakai function dengan nama perkalian(bil_1,bil_2) :

Pada function ini pertama tama kita akan menampilkan bil_1 dikali bil_2 terlebih dahulu dengan code yang ada di line 2 yaitu `def perkalian(bil_1,bil_2):` kemudian masuk kedalam perulangan Dimana `for i in range (1,bil_1)` karena kita ingin mencetak bil_2 sebanyak bilangan pengalinya yaitu bil_1 setelah itu kita buat value i menjadi bil_2 dan `print(i,end="+")` sehingga bil_2 terprint sejumlah bil_1 – 1 dan diakhiri dengan "+". Kemudian pada line 6,7,dan 8 ialah mengakhiri perkalian tersebut dengan hasil perkalian nya codenya ialah `hasil=bil_1*bil_2` dan `print(hasil)` dan untuk output yang sama seperti test case yang diberikan oleh soal saya pun memprint bil_2 terakhir (agar jumlah bilangan 2 nya sejumlah bil_1) yang kemudian diakhiri dengan "=" agar sesuai ketentuan output yang diminta.

Setelah fungsi selesai dibuat maka saya membuat input user untuk memberikan value pada bil_ 1 dan bil_2 nya setelah itu fungsi perkalian yang telah memiliki parameter akan berjalan dan menghasilkan output sesuai dengan yang diinginkan.

SOAL 2

Input :

```
lat_2.py > ...
1  def ganjil(bawah,atas):
2      if bawah>atas:
3          for i in range(bawah,atas-1,-1):
4              if i%2!=0:
5                  if i!=atas and i!=atas+1:
6                      print(i,end=",")
7              else:
8                  print(i,end=".")
9      else:
10         for i in range(bawah,atas+1):
11             if i%2!=0:
12                 if i!=atas and i!=atas-1:
13                     print(i,end=",")
14             else:
15                 print(i,end=".")
16
17  bawah=int(input("batas bawah:"))
18  atas=int(input("batas atas:"))
19  ganjil(bawah,atas)
```

```
batas bawah:10
batas atas:30
11,13,15,17,19,21,23,25,27,29.
```

```
batas bawah:97
batas atas:82
97,95,93,91,89,87,85,83.
```


Pada soal nomor 2 kita diminta untuk membuat program yang dapat menampilkan deret bilangan ganjil dari batas bawah dan batas atas yang diberikan oleh pengguna. Jika ternyata batas atas < batas bawah, berarti deret tersebut dimulai dari batas atas, sampai batas bawah (negatif range).

Pada program ini saya membuat fungsi ganjil (bawah,atas) Dimana parameter bawah dan atas ditentukan oleh input user, jadi pada fungsi ini ketika value bawah lebih besar dari atas maka program akan mengeksekusi perulangan dari line 3 yang mana berfungsi untuk menampilkan negative range dari bilangan ganjil antara bawah-atas-1.

Untuk menampilkan bilangan yang ganjil maka kita memakai rumus conditional Dimana ketika bilangan di modulus 2 dan menghasilkan nilai yang tidak sama dengan 0 maka bilangan tersebut sudah dipastikan ialah bilangan ganjil maka bilangan ganjil itu akan di tampilkan di output.

Untuk menentukan apakah bilangan tersebut diakhiri oleh “,” atau “.” Maka diperlukan conditional baru Dimana ketika bilangan ganjil tersebut nilainya tidak sama dengan atas dan atas +1 (hal ini berarti bilangan ganjil tersebut bukanlah bilangan terakhir) maka bilangan tersebut akan diakhiri dengan tanda koma “,” dan ketika bilangan tersebut merupakan bilangan terakhir maka bilangan ganjil tersebut akan diakhiri dengan tanda titik “.”

Untuk line 9 yang merupakan else Dimana value bawah lebih kecil dari atas maka perulangan dibuat untuk menampilkan bilangan ganjil diantara batas bawah dan atas. Konsep kerja conditionalnya sama dengan negative range nya hanya perlu penyesuaian dalam parameternya saja.

SOAL 3

```
lat_3.py > ips
1 def ips():
2     juMat=int(input("total matakuliah:"))
3     i=0
4     rata_rata=0
5     while i<juMat:
6         huruf=input(f"nilai MK {i+1} :")
7         if huruf=="A":
8             angka=4
9         elif huruf=="B":
10            angka=3
11        elif huruf=="C":
12            angka=2
13        elif huruf=="D":
14            angka=1
15        rata_rata=rata_rata+angka
16        i+=1
17
18    return round(rata_rata/i,2)
19 total=ips()
20 print(f"rata ratanya adalah {total}")
```

```
total matakuliah: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
rata ratanya adalah 2.67
```

```
total matakuliah:6
nilai MK 1 :A
nilai MK 2 :A
nilai MK 3 :A
nilai MK 4 :A
nilai MK 5 :A
nilai MK 6 :A
rata ratanya adalah 4.0
```

```
total matakuliah:6
nilai MK 1 :A
nilai MK 2 :B
nilai MK 3 :D
nilai MK 4 :D
nilai MK 5 :A
nilai MK 6 :D
rata ratanya adalah 2.33
```

Pada soal ketiga kita diminta untuk membuat sebuah program untuk melakukan penghitung nilai Indeks Prestasi Semester (IPS).

Input bagi program:

- Jumlah mata kuliah
- Nilai A, B, C, dan D untuk setiap mata kuliah mahasiswa. Diasumsikan sks setiap mata kuliah selalu 3. Kemudian bobot dari masing-masing nilai adalah: A=4, B=3, C=2, D=1. Output program ialah hasil IPS yang didapatkan.

Pada soal 3 saya membuat fungsi dengan nama ips() yang pertama akan meminta input user untuk JuMat(Jumlah Matakuliah)

Kemudian set value untuk i=0 dan rata_rata=0 kedua nilai ini berfungsi untuk menghitung total ips dan juga jumlah matakuliahnya

Pada line berikutnya saya membuat perulangan memakai while ketika i lebih kecil dari JuMat maka perulangan akan terus berjalan

(Dalam perulangan)

```
huruf=input(f"nilai MK {i+1} :")
```

fungsi code tersebut adalah Meminta input user untuk mengisi variable huruf mulai dari 0(value i) + 1 sehingga MK yang ditampilkan pertama ialah 1 dst.. hingga akhirnya value i setara dengan 6 dan perulangan berhenti.

Selanjutnya memakai conditional untuk memberi nilai pada inputan huruf Dimana ketika input "A" maka value angka akan menjadi 4 dan jika value huruf "B" maka value angka menjadi 3 dst.. sesuai dengan ketentuan

Setelah itu rata rata yang awalnya 0 akan ditambah valuenya dengan value angka dari inputan tersebut dan hal ini akan bertumpuk karena berada dalam perulangan.

Diakhir nilai akan ditambah 1 setiap perulangan berjalan hal ini

Setelah memiliki nilai total yang di input oleh user dan i sudah memiliki value yang sama dengan JuMat maka perulangan berhenti dan rata_rata sudah menampung jumlahan seluruh nilai yang di input yang kemudian akan dihitung rata-rata ips nya dengan rumus $\text{rata_rata} / i$ (nilai i sama dengan JuMat) kemudian di rata rata

Setelah itu nilai yang didapat dari fungsi disama dengankan dengan total sehingga output yang ditampilkan ialah nilai totalnya.

Source: <https://github.com/vianchr/LapPrak5> vian

