



Perulangan di Python

Notebook ini akan mengajarkan Anda materi mengenai perulangan di bahasa Python. Pada akhir lab ini, Anda diharapkan memahami cara menggunakan perulangan pada Python, terutama perulangan `for` dan `while`

Daftar Isi

- [Perulangan](#)
 - [Range \(range\)](#)
 - [Perulangan for \(for\)](#)
 - [Perulangan while \(while\)](#)
- [Kuis tentang perulangan](#)

Estimasi waktu pengerjaan: **20 menit**

Perulangan

Range

Kadangkala, Anda ingin menjalankan suatu perintah berulang kali. Sebelum membahas perulangan, akan lebih baik jika mengetahui perintah `range` terlebih dahulu. `range` merupakan fungsi yang menghasilkan `list` dengan nilai berurutan. Sebagai contoh, jika Anda menginginkan `list` yang terdiri dari 3 elemen berurutan dari 0 sampai 2, gunakan perintah berikut:

In [1]:

```
# Penggunaan range
```

```
range(3)
```

Out[1]:

```
range(0, 3)
```

range(3)



[0,1,2]

Perulangan for

Perulangan `for` digunakan untuk mengeksekusi suatu blok kode berulang kali. Sebagai contoh, Anda dapat menggunakannya untuk menampilkan setiap elemen dari sebuah `list`. Pada contoh di bawah ini, perintah `for` digunakan untuk menampilkan setiap tahun yang berada pada `list` `dates` :

In [2]:

```
# Contoh perulangan for

dates = [1982,1980,1973]
N = len(dates)

for i in range(N):
    print(dates[i])
```

```
1982
1980
1973
```

Kode yang terletak pada indentasi (menjorok masuk) dijalankan sebanyak `N` kali, setiap kali dieksekusi nilai `i` akan dinaikkan sebesar 1. Pernyataan yang dijalankan dalam perulangan `for` adalah fungsi `print` untuk menampilkan nilai `list` pada indeks ke- `i` sebagai berikut:

```
for i in range(N):  
    print(dates[i])  
  
Dates=[1982,1980,1973]
```

Pada contoh di bawah ini, urutan angka dari 0 sampai 7 akan ditampilkan:

In [3]:

```
# Contoh perulangan for  
  
for i in range(0, 8):  
    print(i)
```

```
0  
1  
2  
3  
4  
5  
6  
7
```

Pada Python, setiap elemen pada list dapat diakses secara langsung:

In [4]:

```
# Contoh perulangan for untuk List  
  
for year in dates:  
    print(year)
```

1982

1980

1973

Pada setiap iterasi, nilai dari variabel `year` sama dengan nilai variabel `dates[i]` pada contoh sebelumnya:

```
for year in dates:
```

```
    print(year)
```

```
Dates=[1982,1980,1973]
```

Elemen-elemen pada list dapat diubah dalam perulangan:

In [5]:

```
# Gunakan perulangan for untuk mengubah nilai elemen pada list

squares = ['red', 'yellow', 'green', 'purple', 'blue']

for i in range(0, 5):
    print("Before square ", i, 'is', squares[i])
    squares[i] = 'weight'
    print("After square ", i, 'is', squares[i])
```

```
Before square 0 is red
After square 0 is weight
Before square 1 is yellow
After square 1 is weight
Before square 2 is green
After square 2 is weight
Before square 3 is purple
After square 3 is weight
Before square 4 is blue
After square 4 is weight
```

Pengaksesan indeks dan elemen pada list dapat pula dilakukan dengan cara sebagai berikut:

In [6]:

```
# Perulangan pada list disertai pengaksesan indeks dan elemen secara bersamaan

squares=['red', 'yellow', 'green', 'purple', 'blue']

for i, square in enumerate(squares):
    print(i, square)
```

```
0 red
1 yellow
2 green
3 purple
4 blue
```

Perulangan while

`for` dapat digunakan untuk melakukan perulangan, terutama jika kita mengetahui berapa kali perulangan tersebut akan dilakukan. Namun, bagaimana jika kita melakukan perulangan tanpa mengetahui berapa kali perulangan dilakukan? Bagaimana jika kita ingin melakukan perulangan terus-menerus sampai suatu kondisi tertentu terpenuhi? Perintah `while` digunakan untuk melakukan perulangan berdasarkan suatu kondisi tertentu. Blok kode pada `while` akan dieksekusi sampai kondisi berhenti bernilai **False**

Misalkan kita ingin menampilkan tahun pada list `dates` dan berhenti pada tahun 1973 serta menampilkan banyaknya iterasi. Kode di bawah ini menggunakan `while` untuk melakukan perulangan:

In [7]:

```
# Contoh perulangan while

dates = [1982, 1980, 1973, 2000]


i = 0
year = 0

while(year != 1973):
    year = dates[i]
    i = i + 1
    print(year)

print("It took ", i , "repetitions to get out of loop.")
```

```
1982
1980
1973
It took 3 repetitions to get out of loop.
```

A while loop iterates merely until the condition in the argument is not met, as shown in the following figure:



```
albums = 250
total_albums = 0
i=0;
while( year!=1973):
    year=dates[i]
    i=i+1
    print(year)

print("it took",i, "outloop")
```

dates=[1982,1980,1973,1992]



Kuis mengenai Perulangan

Tuliskan kode program menggunakan perulangan `for` untuk menampilkan bilangan di antara **-5** dan **5** menggunakan fungsi `range`

In [4]:

```
# Tulis kode anda dibawah ini dan tekan Shift+Enter untuk melakukan eksekusi
for i in range(-5, 5):
    print(i)
```

```
-5
-4
-3
-2
-1
0
1
2
3
4
```

Double-click **di sini** untuk membuka jawaban.

Tampilkan setiap elemen dari list berikut : Genres=['rock', 'R&B', 'Soundtrack', 'R&B', 'soul', 'pop']

In [12]:

```
# Tulis kode anda dibawah ini dan tekan Shift+Enter untuk melakukan eksekusi
Genres = ['rock', 'R&B', 'Soundtrack', 'R&B', 'soul', 'pop']
print ("Semua Genres Lagu : ada {} Genres".format(len(Genres)))
for Genre in Genres:
    print(Genre)
```

```
Semua Genres Lagu : ada 6 Genres
rock
R&B
Soundtrack
R&B
soul
pop
```

Tuliskan perulangan for untuk menampilkan list berikut : squares=['red', 'yellow', 'green', 'purple', 'blue']

In [6]:

```
# Tulis kode anda dibawah ini dan tekan Shift+Enter untuk melakukan eksekusi
squares=['red', 'yellow', 'green', 'purple', 'blue']

for i, square in enumerate(squares):
    print(i, square)
```

```
0 red
1 yellow
2 green
3 purple
4 blue
```

Double-click **di sini** untuk membuka jawaban.

Tulislah perulangan `while` untuk menampilkan nilai Rating yang disimpan pada list `PlayListRatings` .
Jika nilai kurang dari 6, keluar dari perulangan. List `PlayListRatings` adalah sebagai berikut:
`PlayListRatings = [10, 9.5, 10, 8, 7.5, 5, 10, 10]`

In [40]:

```
# Tulis kode anda dibawah ini dan tekan Shift+Enter untuk melakukan eksekusi
PlaylistRating = [10, 9.5, 10, 8, 7.5, 5, 10, 10]

i = 0
rating = 0

while(rating < 6):
    rating = PlaylistRating[i]
    i = i + 1
    print("nilai rating lagunya adalah",rating)
```

nilai rating lagunya adalah 10

Double-click **di sini** untuk membuka jawaban.

Tulislah perulangan `while` untuk menyalin string `'orange'` dari list `squares` ke list `new_squares` .
Keluar dari perulangan jika menjumpai elemen list yang bukan `'orange'` :

In [59]:

```
# Tulis kode anda dibawah ini dan tekan Shift+Enter untuk melakukan eksekusi

squares = ['orange', 'orange', 'purple', 'blue ', 'orange']
newsquares = []

i= 0
while (squares [i] == 'orange'):
    newsquares.append(squares [i])
    i=i+1

print (newsquares)
```

```
['orange', 'orange']
```

Double-click **di sini** untuk membuka jawaban.

About the Authors:

[Joseph Santarcangelo](https://www.linkedin.com/in/joseph-s-50398b136/) (<https://www.linkedin.com/in/joseph-s-50398b136/>) is a Data Scientist at IBM, and holds a PhD in Electrical Engineering. His research focused on using Machine Learning, Signal Processing, and Computer Vision to determine how videos impact human cognition. Joseph has been working for IBM since he completed his PhD.

Other contributors: [Mavis Zhou](http://www.linkedin.com/in/jiahui-mavis-zhou-a4537814a) (www.linkedin.com/in/jiahui-mavis-zhou-a4537814a), [James Reeve](https://www.linkedin.com/in/reevejamesd/) (<https://www.linkedin.com/in/reevejamesd/>).

Copyright © 2018 IBM Developer Skills Network. This notebook and its source code are released under the terms of the [MIT License](https://cognitiveclass.ai/mit-license/) (<https://cognitiveclass.ai/mit-license/>).