

# UNIT 5

## THE ITERATOR PATTERN (1)

### UNIT OVERVIEW

Pada modul kali ini kita akan membahas mengenai penjelasan terhadap pola-pola yang berulang. Hal ini tentunya menjadi sangat penting, karena apabila pola berulang tersebut tidak kita tangani dengan baik akan mengakibatkan proses yang berulang dan dapat mengakibatkan proses yang tidak penting. Oleh karena itu, penting untuk dipahami kira-kira pola yang bagaimana yang bisa disebut sebagai pola berulang dan gimana cara mengatasinya.

### UNIT OBJECTIVES

Adapun capaian yang akan kita dapatkan pada modul ini, yaitu :

- Cara pembuatan pola berulang dengan for.
- Cara pembuatan pola berulang dengan iter.

### UNIT CONTENTS

Lesson: Lesson Name .....	1-8
Lesson: Lesson Name .....	1-38
Lesson: Lesson Name .....	1-88

## PRE LAB

Sebelum masuk ke dalam lab pertama ini, terlebih dahulu kita coba jawab beberapa pertanyaan ini berdasarkan hasil penjelasan pada modul teori!

### QUESTION

1. Menurut kalian pentingnya kita melakukan exception adalah untuk?
2. Apa bedanya statement raise dengan statement try dan except?
3. Kira-kira ketika kita membuat data berupa nama pelanggan, batas-batas atau syarat-syarat apa saja yang harus kita penuhi?

## CONTENT LESSON

### CASE STUDY / PROJECT

Seperti yang telah kita ketahui, data yang ada perpustakaan “Home Learning is Best” menjadi sangat banyak. Hal ini mengakibatkan ketika pencarian yang dilakukan oleh staf menjadi sangat lama dan membuat kinerja menjadi tidak efisien. Oleh karena itu, perusahaan tersebut ingin membuat cara untuk melakukan pencarian data yang sangat banyak tersebut. Setelah data tersebut dicari, nantinya akan dikelompokkan berdasarkan jenis buku yang digunakan. Hal ini bertujuan supaya ketika pencarian yang dilakukan selanjutnya dapat tercapai dan tidak membuang waktu yang banyak.

Yuk mari bantu kembali perpustakaan kita yang tercinta ini!

### IDENTIFICATION CONCEPT OF PROBLEM / PROJECT

Setelah kita membaca kembali studi kasus tersebut ada yang harus kita perhatikan mengenai cara yang harus kita lakukan untuk melakukan pencarian. Apakah hal tersebut dapat dilakukan sekali tahapan saja sudah cukup atau perlu dilakukan secara berulang? Setelah kita mendapatkan datanya apa yang harus dilakukan? Berdasarkan studi kasus tersebut perlu dilakukan pengelompokkan. Bagaimana cara pengelompokkan yang baik?

Sebelum kita menjelaskan solusi dari permasalahan tersebut, mari kita simak dulu beberapa penjelasan untuk membantu kita dalam menyelesaikan permasalahan yang terdapat dalam studi kasus tersebut !

## LESSON 1: PERULANGAN DENGAN FOR

### LESSON 1: OVERVIEW

Sebelumnya pada semester 1 kita pernah belajar melakukan perulangan di dalam program Python. Hal ini bertujuan untuk melakukan tindakan yang berulang tanpa harus menghabiskan baris yang banyak dan tetap sesuai dengan yang diharapkan.

Sebagai contoh, mungkin kalian pernah mendengar murid yang disuruh sama gurunya untuk menulis dipapan tulis dengan kata berupa “Saya tidak akan berbohong lagi ... !!!”. Kata tersebut tentunya bukan sekali ditulis tetapi bisa dilakukan beberapa kali sesuai dengan tingkat kesalahan yang dilakukan oleh murid tersebut (Jangan dicontoh tindakan berbohongnya ya ... !!!). Seperti contoh pada tampilan program berikut :

```

1  print("Saya tidak akan berbohong lagi ... !!!")
2  print("Saya tidak akan berbohong lagi ... !!!")
3  print("Saya tidak akan berbohong lagi ... !!!")
4  print("Saya tidak akan berbohong lagi ... !!!")
5  print("Saya tidak akan berbohong lagi ... !!!")
6  print("Saya tidak akan berbohong lagi ... !!!")
7  print("Saya tidak akan berbohong lagi ... !!!")
8  print("Saya tidak akan berbohong lagi ... !!!")
9  print("Saya tidak akan berbohong lagi ... !!!")
10 print("Saya tidak akan berbohong lagi ... !!!")

```

Figure 51 Contoh Perulangan (1)

Dari hasil program tersebut akan menghasilkan 10 baris dengan kata berupa “Saya tidak akan berbohong lagi ... !!!”. Mungkin ketika kalian lakukan hal tersebut merupakan hal yang biasa saja. Karena, hal tersebut dapat terselesaikan hanya menggunakan copy dan paste sebanyak 9 kali saja (karena yang pertama tidak perlu dilakukan copy paste). Tapi bayangkan, bagaimana kalau yang disuruh adalah 1000 baris? Apakah copy paste bisa mengatasinya? Jawabannya adalah bisa, hanya saja mau sampai kapan? Oleh karena itu, kita bisa memanfaatkan perintah for, yang digunakan untuk melakukan perulangan. Seperti contoh tampilan kode berikut :

```

for i in range(1,1001):
    print("Saya tidak akan berbohong lagi ... !!!")

```

Figure 52 Contoh Perulangan (2)

NB : Kenapa 1001? Karena, pada perulangan hanya dilakukan sampai dengan  $n - 1$  atau  $1001 - 1 = 1000$ . Sehingga, kita perlu menggunakan 1001 untuk membuat 1000 baris kode perintah untuk mencetak.

Dengan adanya 2 baris tersebut, kita bisa mencetak 1000 baris. Otomatis kita bisa menyelesaikan masalah pada murid tersebut. Hanya saja ini hanya sebagai contoh, bukan untuk ditiru. Yang ingin dijelaskan pada contoh ini adalah kita bisa menyederhanakan perintah tersebut menjadi lebih baik. Hal ini juga bisa kita lakukan untuk pengaksesan data yang ada di dalam array (sekumpulan data) seperti contoh berikut :

```

items = ["penghapus", "pulpen", "pensil", "buku", "penggaris", "tas sekolah"]
for i in range(len(items)):
    print(f>Data ke-{i+1} = {items[i]}")

```

Figure 53 Pengaksesan Index Array For (1)

Dari kode tersebut akan memunculkan hasil berikut :

```
Data ke-1 = penghapus
Data ke-2 = pulpen
Data ke-3 = pensil
Data ke-4 = buku
Data ke-5 = penggaris
Data ke-6 = tas sekolah
```

Figure 54 Hasil Pengaksesan Index Array For (1)

Dari hasil tersebut ternyata for bukan hanya sebagai mencetak saja, tapi bisa juga kita gunakan untuk mengakses data yang ada di dalam array. Hal ini tentunya dapat membantu kita apabila data tersebut terdapat di dalam sekumpulan data yang banyak, yang mengharuskan kita untuk melakukan pencarian 1 per 1. Namun, ada cara lain yang juga bisa kita gunakan untuk mengakses data tersebut dengan cara seperti kode berikut :

```
items = ["penghapus", "pulpen", "pensil", "buku", "penggaris", "tas sekolah"]
for item in items:
    print(f>Data = {item})
```

Figure 55 Pengakses Array Index For (2)

Dalam kode tersebut akan menghasilkan tampilan berikut :

```
Data = penghapus
Data = pulpen
Data = pensil
Data = buku
Data = penggaris
Data = tas sekolah
```

Figure 56 Hasil Pengaksesan Index Array For (2)

Pada hasil tersebut terlihat, bahwa cara ini juga bisa kita gunakan untuk melakukan pencetakan data tanpa harus data seberapa data tersebut. Jadi, seperti kita ingin memastikan apakah orang A ada di dalam kelas tersebut atau tidak. Hal ini tentunya dapat membuat segala sesuatu menjadi lebih jelas dan tidak perlu dilakukan secara acak.

Namun, untuk melakukan perulangan bukan hanya menggunakan for saja, kita juga bisa menggunakan iter. Ingin tahu bagaimana caranya? Yuk kita pelajari pada pelajaran berikutnya ... !!!

## LESSON 2: PERULANGAN DENGAN ITER

### LESSON 2: OVERVIEW

Kita telah mengenal teknik untuk perulangan pada pelajaran sebelumnya adalah dengan menggunakan `for`. Namun, untuk pengaksesan indeks di dalam array ternyata ada 1 lagi satu lagi caranya, yaitu dengan menggunakan fungsi `iter`. Hal ini mungkin merupakan bagian yang sangat baru untuk kalian semuanya. Namun, apabila kalian telah mengetahui cara yang dilakukan di dalam `for item in items`, mungkin kalian sudah dapat mengetahui gimana caranya bekerja.

Secara gambaran besar dapat terlihat pada tampilan grafik berikut :

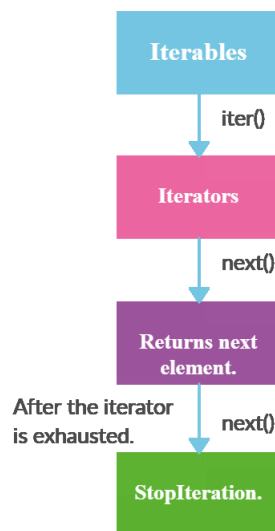


Figure 57 Tahapan Iter

Gambar mungkin telah kalian lihat pada pertemuan teori. Adapun langkah-langkah yang dilakukan proses perulangan dengan menggunakan `iter` dilakukan dengan 3 buah fungsi, yaitu : `iter()`, `next()` dan `done()`. Ketiga fungsi ini memiliki fungsinya masing-masing, yaitu :

1. `Iter()` merupakan fungsi yang bisa kita gunakan untuk membuat sebuah objek perulangan menjadi perulangan untuk bisa dilakukan proses secara berulang. Sebagai contoh array, list, dict dan beberapa objek lainnya.
2. `Next()` merupakan fungsi yang bisa kita gunakan untuk membuat penambahan (increment) ke dalam sekumpulan data tersebut. Hal ini tentunya harus dilakukan supaya data tersebut dapat sampai ke ujung dan dapat berakhir.
3. `Done()` merupakan fungsi yang bisa kita gunakan untuk mengakhiri perulangan. Meskipun fungsi ini sangat jarang sekali terlihat pada proses perulangan yang dilakukan pada `iter`, dan proses berulang juga dapat dilakukan meskipun tanpa adanya fungsi ini. Hanya saja ketika kita ingin mengakhir proses perulangan, adanya kita menggunakan fungsi untuk mengakhiri perulangan tersebut.

Setelah kita mengetahui fungsi dari 3 fungsi tersebut, mari kita kembali ke array kita sebelumnya. Bagaimana cara penerapannya yang harus kita lakukan? Silahkan perhatikan kode berikut :

```
items = ["penghapus", "pulpen", "pensil", "buku", "penggaris", "tas sekolah"]
datas = iter(items) #Inisialisasi iter
print(f>Data = {next(datas)}") #Mengakses data pertama
print(f>Data = {next(datas)}") #Mengakses data kedua
print(f>Data = {next(datas)}") #Mengakses data ketiga
print(f>Data = {next(datas)}") #Mengakses data keempat
print(f>Data = {next(datas)}") #Mengakses data kelima
print(f>Data = {next(datas)}") #Mengakses data keenam
print(f>Data = {next(datas)}") #Mengakses data ketujuh ?????
```

Figure 58 Pengaksesan Indeks Array Iter (1)

Seperti yang kita lihat sangatlah simple dan tidak perlu melakukan dengan pengaksesan indeks juga. Namun, gimana hasilnya? Kita lihat pada tampilan berikut :

```
Data = penghapus
Data = pulpen
Data = pensil
Data = buku
Data = penggaris
Data = tas sekolah
Traceback (most recent call last):
  File "contoh1.py", line 34, in <module>
    print(f>Data = {next(datas)}") #Mengakses data ketujuh ?????
StopIteration
```

Figure 59 Hasil Pengaksesan Indeks Array Iter (1)

Dari hasil tersebut terlihat hasilnya tidak jauh beda ketimbang menggunakan for yang telah kita lakukan sebelumnya. Hanya saja pada baris akhir ada error dan langsung menghasilkan StopIteration. Hal ini dikarenakan data yang ada di dalam array hanya terdapat 6, sedangkan kita mencetak sampai 7 kali. Hal ini tentunya dapat mengakibatkan kesalahan yang ada pada proses tersebut. Terus, gimana cara menangani masalah ini? Kita bisa menggunakan try except yang telah kita pelajari sebelumnya. Namun, bagaimana cara kita supaya kita tidak perlu menjalankan 1 per 1 datanya, kita bisa memanfaatkan while untuk melakukan perulangan. Hasilnya akan seperti tampilan kode berikut :

```
items = ["penghapus", "pulpen", "pensil", "buku", "penggaris", "tas sekolah"]
datas = iter(items) #Inisialisasi iter

while True:
    try:
        print(f>Data = {next(datas)}") #Mengakses array
    except StopIteration:
        break
```

Figure 60 Pengaksesan Indeks Array Iter (2)

Dari kode tersebut tentunya terlihat lebih beda dan tentunya dapat menghemat baris karena kita tidak perlu mencetak sampai dengan jumlah data yang ada. Terus, gimana hasil? Apakah sama saja? Kita lihat pada tampilan dibawah ini :

```
Data = penghapus
Data = pulpen
Data = pensil
Data = buku
Data = penggaris
Data = tas sekolah
```

Figure 61 Hasil Pengaksesan Indeks Array Iter (2)

Gimana? Sama dengan menggunakan for bukan. Mungkin kalian berpikir agak sedikit rumit, hanya saja dengan menggunakan perintah ini kita bisa lebih memahami proses yang terjadi dan pada saat akhir kita bisa pencetakan data atau kata.

Gimana? Mudah bukan. Yuk kita bahas solusi dari permasalahan studi kasus kita ... !!!

## SOLUTION

Setelah kita baca kembali studi kasus yang ada diatas, kita bisa memberikan solusi yang ada berupa melakukan perulangan. Karena, apabila datanya lebih dari 1, tentunya sangat efisien dilakukan perulangan. Tanpa perulangan tentunya akan sangat lama. Terus, teknik perulangan mana yang akan kita gunakan? Tentunya saya akan lebih memilih menggunakan iter. Supaya jelas prosesnya dan ketika tidak ditemukan datanya, kita bisa mencetak hasilnya tidak ditemukan tanpa perlu menggunakan percabangan tambahan.

Nah supaya lebih jelas, yuk kita bahas 1 per 1.

## INSTRUCTION

1. Supaya prosesnya tidak terlalu panjang, saya buat saja data kita dalam bentuk array yang dimana isinya sudah ada nama beberapa buku yang ada berupa :

```
buku = []
buku += ["Pemrograman dengan Python"]
buku += ["Struktur data"]
buku += ["Data Mining"]
buku += ["Visualisasi Data"]
buku += ["Pengolahan Citra"]
buku += ["Kecerdasan Buatan"]
buku += ["Pemrograman Web"]
```

2. Setelah kita buat datanya, mari kita buat fungsi pencarian dengan menggunakan iter. Gimana caranya? Perhatikan kode berikut :

```
#Melakukan pencarian buku dalam array buku
def cariBuku(buku, buku):
    items = iter(buku)
    i = 0
    while True:
        try:
            if(next(items) == buku):
                print(f"Buku \"{buku}\" ditemukan pada tumpukan ke-{i+1} ... !!!")
                break
            i += 1
        except StopIteration:
            print(f"Buku \"{buku}\" tidak ditemukan ... !!!")
            break
```

3. Untuk melakukan pencariannya kita hanya melakukan perintah berikut :

```
cariBuku(buku, "Database")
cariBuku(buku, "Struktur data")
cariBuku(buku, "Pemrograman C")
```

4. Setelah proses tersebut dijalankan, maka akan muncul tampilan berikut :

```
Buku "Database" tidak ditemukan ... !!!
Buku "Struktur data" ditemukan pada tumpukan ke-2 ... !!!
Buku "Pemrograman C" tidak ditemukan ... !!!
```

5. Terus gimana cara kita bisa melakukan pengelompokkan? Untuk proses ini kita akan bahas pada pertemuan berikutnya. Jadi, jangan lupa ditunggu untuk pertemuan berikutnya.

## EXERCISE

### EXERCISE OBJECTIVES

Pada latihan berikut ini mahasiswa diharapkan dapat melakukan :

- Melakukan analisis masalah terhadap studi kasus yang diberikan.
- Mencari solusi atas masalah yang diberikan.
- Menerjemahkan masalah tersebut ke dalam bentuk kode pemrograman.

### TASK 1:

Pada kampus Mikroskil tentunya sangat banyak sekali mahasiswa dalam 1 kelas. Hal ini tentunya membuat dosen untuk mengetahui siapa saja yang ada di dalam kelas tersebut. Dimana kita ketahui bahwa data mahasiswa tersebut terdiri dari NIM, Nama dan NoHP. Dapatkah kalian membantu dosen ini untuk melakukan absensi untuk kelasnya?



Ketentuan yang dilakukan pada task-01 ini adalah :

- a. Semua proses dilakukan di dalam kelas.
- b. Nama kelas tersebut diharuskan menggunakan nama kalian.
- c. Untuk tindakan yang dilakukan dikelas tersebut terdapat 2, yaitu pemasukan data dan absensi.
- d. Untuk jenis perulangan yang digunakan pada Task-01 ini adalah dengan menggunakan for.
- e. Untuk contoh data yang digunakan untuk melakukan pencarian ada 5 data. (Silahkan pastikan tidak ada data yang sama)
- f. Untuk hasil pencarian tampilkan :
  - o 1 data yang ada di dalam kelas Mikroskil.
  - o 2 data yang tidak ada di dalam kelas Mikroskil.
- g. Untuk pesan keluaran bisa disesuaikan dengan kebutuhan kalian sendiri.
- h. Dikarenakan ada contoh data, maka perlu ada tampilan hasil proses pencarian data (data ditemukan atau tidak) dalam bentuk screenshot.

---

## TASK 2:

Pada sebuah website E-Commerce terdapat berbagai jenis barang yang dijual, diantaranya hand sanitizer, ban mobil, buah-buahan serta botol minuman bayi. Barang tersebut memiliki atribut berupa kode, nama dan jumlah barang. Tentunya barang yang dijual tersebut tidak lebih dari 1 barang sehingga penjual harus tahu kira-kira barang apa saja yang mau habis. Hal ini tentunya perlu dilakukan pencarian dari data yang ada. Dapatkan kalian membantu pedagang tersebut?

Ketentuan yang dilakukan pada task-02 ini adalah :

- a. Semua proses dilakukan di dalam kelas.
- b. Nama kelas tersebut diharuskan menggunakan nama kalian.
- c. Untuk tindakan yang dilakukan dikelas tersebut terdapat 2, yaitu pemasukan data dan pencarian.
- d. Untuk jenis perulangan yang digunakan pada Task-02 ini adalah dengan menggunakan iter.
- e. Untuk contoh data yang digunakan untuk melakukan pencarian ada 5 data. (Silahkan pastikan tidak ada data yang sama)
- f. Dari data-data yang ada, tentunya harus dicari kira-kira stok mana yang sudah mau habis.
- g. Untuk pesan keluaran bisa disesuaikan dengan kebutuhan kalian sendiri.
- h. Dikarenakan ada contoh data, maka perlu ada tampilan hasil proses pencarian data (kira-kira bagaimana pesan keluaran ketika diketahui barang tersebut sudah mau habis) dalam bentuk screenshot.