

Bab III

Looping

Looping atau perulangan merupakan cara yang digunakan untuk menjalankan perintah yang berulang untuk data berbentuk kelompok seperti list, tuple atau string. Dalam melakukan perintah looping biasanya digunakan perintah For Loop dan While Loop, serta penerapan pernyataan Break dan Continue sebagai pilihan tambahan dalam membuat perintah looping.

A. For Loop

Perintah “For” digunakan dalam melakukan perulangan pada data kelompok yang sudah diketahui jumlah iterasinya dan akan berhenti jika iterasinya telah dieksekusi semuanya.

Contoh:

```
fruits = ["apel", "pisang", "mangga"]
for x in fruits:
    print("Ini adalah", x)
```

Output:

```
Ini adalah apel
Ini adalah pisang
Ini adalah mangga
```

B. While Loop

Perintah “While” digunakan dalam melakukan perulangan pada data kelompok yang tidak diketahui jumlah pasti iterasinya. Oleh karena itu, biasanya perintah while akan terus melakukan perulangan selama suatu kondisi masih terpenuhi. Selain itu, pada looping juga kita bisa tambahkan kondisi “else” seperti layaknya pada Conditional Statement pada bab sebelumnya.

Contoh:

```
n = int(input("Enter n: "))

sum = 0
i = 1

while i <= n:
    sum = sum + i
    i = i+1
else: #Penambahan kondisi else optional
    print("The sum is", sum)
```

Output:

```
Enter n: 5
The sum is 15
```

C. Break dan Continue

Penggunaan “Break” dan “Continue” digunakan untuk menambahkan kondisi perintah pada looping sehingga dalam looping tersebut akan ada kondisi “If” yang ditambahkan. Break digunakan jika kita ingin menghentikan sebuah looping jika suatu kondisi telah terpenuhi, sedangkan continue digunakan untuk melewati iterasi sekarang dan melanjutkan langsung ke iterasi selanjutnya jika kondisinya terpenuhi.

1. Break Statement

```
fruits = ["apel", "pisang", "mangga"]
for x in fruits:
    if x == "pisang":
        Break
    print("Ini adalah", x)
```

Output:

```
Ini adalah apel
```

2. Continue Statement

Sama halnya pada “For Loop” statement di atas juga dapat digunakan pada “While Loop” dengan penulisan yang sama. Menambahkan conditional statement ke dalam looping yang dibuat.

```
fruits = ["apel", "pisang", "mangga"]
for x in fruits:
    if x == "pisang":
        continue
    print("Ini adalah", x)
```

Output:

```
Ini adalah apel
Ini adalah mangga
```

D. Nested Loop

Nested loop atau perulangan bersarang adalah sebuah perulangan yang di dalamnya terdapat perulangan yang lain. Baik itu perulangan While-For, While-While, For-For atau berbagai macam kombinasi Nested Loop.

Contoh:

```
adj = ["red", "tasty"]
fruits = ["apple", "banana"]

for x in adj: #Loop Pertama
    for y in fruits: #Loop Kedua
        print(x, y)
```

Output:

```
red apple
red banana
tasty apple
tasty banana
```

E. Try-Exception

Ketika terjadi kesalahan, atau exception, Python biasanya akan berhenti dan menghasilkan pesan kesalahan. Pengecualian ini dapat ditangani menggunakan pernyataan try:

Contoh 1 :

```
x = 3
y = 'Empat'
try:
    z = x + y
except:
    z = str(x) + y

print(z)
```

```
3Empat
```

Contoh 2:

```
x = 3
y = 'Empat'
try:
    z = x + y
except:
    z = y + str(x)
else:
    print('Tidak dapat menjumlahkan keduanya')

print(z)
```

```
Empat3
```

Contoh 3:

```
x = 3
y = 'Empat'
try:
    z = x + y
except:
    z = y + str(x)
finally:
    print('!!! salah satunya dilakukan conversi type data')

print(z)
```

```
!!! salah satunya dilakukan conversi type data
Empat3
```