



# Python Programming

**Looping**



01

# Perulangan

# Perulangan

- Perulangan atau *looping* merupakan perintah untuk menyuruh program melakukan sesuatu secara berulang-ulang.
- Di Python, terdapat dua perintah untuk melakukan *looping* yaitu fungsi `for` dan `while`.
- `For` digunakan apabila kita telah mengetahui seberapa banyak program harus mengulang
- `For` disebut dengan perulangan yang terhitung atau *counted loop*
- `While` digunakan apabila kita belum mengetahui seberapa banyak program harus mengulang
- `While` melakukan pengulangan jika memiliki syarat dan tidak tentu berapa banyak pengulangan tersebut
- `While` disebut dengan *uncounted loop*



02

# Looping: For

# Perulangan: for

- For digunakan untuk mengulangi urutan dan bisa digunakan di beberapa tipe data seperti *list*, *tuple*, *dictionary*, *set*, dan *string*
- Bentuk umum untuk pengulangan for yaitu:

```
1 for index in range(jumlah_perulangan):  
2     # eksekusi program ini sebanyak jumlah_perulangan  
3     # eksekusi program ini sekali
```

- Contoh program untuk melakukan print dengan for

```
1 jumlah_perulangan = 3  
2  
3 for index in range(jumlah_perulangan):  
4     print(f"Print ke-{index}")
```

```
Print ke-0  
Print ke-1  
Print ke-2
```

- Perintah print akan mengulangi sebanyak nilai jumlah\_perulangan, yaitu tiga kali dimulai dari index ke-0

# Perulangan: for

- Contoh implementasi for untuk melakukan perulangan dari suatu string

```
1 for i in "Python":  
2     print(f"Huruf: {i}")
```

Huruf: P  
Huruf: y  
Huruf: t  
Huruf: h  
Huruf: o  
Huruf: n

- Implementasi for dengan menggunakan range()

```
1 for i in range(1, 6):  
2     print(i)
```

1  
2  
3  
4  
5

# Perulangan: for

- Contoh implementasi for untuk melakukan perulangan dari suatu list

```
1 buah = ["apel", "belimbing", "ceri", "durian"]
2
3 for i in buah:
4     print(f"Buah: {i}")
```

Buah: apel  
Buah: belimbing  
Buah: ceri  
Buah: durian

- Contoh implementasi for untuk melakukan perulangan dari suatu list dengan range()

```
1 buah = ["apel", "belimbing", "ceri", "durian"]
2
3 for i in range(len(buah)):
4     print(f"Buah: {buah[i]}")
```

Buah: apel  
Buah: belimbing  
Buah: ceri  
Buah: durian



03

# Looping: While



# Perulangan: while

- Pengulangan while akan dieksekusi selama statement memenuhi atau bernilai True
- Bentuk umum untuk pengulangan for yaitu:

```
1 while(kondisi):  
2     # jalankan ini selama kondisi bernilai True  
3     # jalankan perintah ini hanya sekali
```

- Contoh program untuk melakukan while

```
1 angka = 0  
2 while(angka < 8 ):  
3     print(f"Angka sekarang: {angka}")  
4     angka += 1
```

```
Angka sekarang: 0  
Angka sekarang: 1  
Angka sekarang: 2  
Angka sekarang: 3  
Angka sekarang: 4  
Angka sekarang: 5  
Angka sekarang: 6  
Angka sekarang: 7
```

# Perulangan: while

- Eksekusi perulangan menggunakan while juga mungkin bersifat infinite loop atau kondisi tidak pernah bernilai False

```
1 angka = 0
2 while(angka < 8 ):
3     print(f"Angka sekarang: {angka}")
```

Angka sekarang: 0  
Angka sekarang: 0  
Angka sekarang: 0  
Angka sekarang: 0  
Angka sekarang: 0  
Angka sekarang: 0  
Angka sekarang: 0  
Angka sekarang: 0  
Angka sekarang: 0  
Angka sekarang: 0



04

# Control the Flow

# Perulangan: Break

- Fungsi break akan menghentikan perulangan sebelum perulangan melewati seluruh item

```
1 angka = [1, 2, 3, 4, 5]
2
3 for i in angka:
4     print(i)
5     if i == 3:
6         break
```

1  
2  
3

```
1 angka = [1, 2, 3, 4, 5]
2
3 for i in angka:
4     if i == 3:
5         break
6     print(i)
```

1  
2

# Perulangan: Continue

- Fungsi continue adalah untuk menghentikan iterasi loop saat ini dan melanjutkan ke iterasi yang berikutnya

```
1 angka = [1, 2, 3, 4, 5]
2
3 for i in angka:
4     if i == 3:
5         continue
6     print(i)
```

```
1
2
4
5
```



05

**Range**

# Perulangan: Range

- Dalam slide sebelumnya, untuk melakukan perulangan dapat menggunakan fungsi `range()`
- Fungsi `range()` mengembalikan urutan angka mulai dari 0 (*default*), bertambah 1 (*default*) hingga pada angka yang ditentukan

```
1 for i in range(6):  
2     print(i)
```

0  
1  
2  
3  
4  
5

- Pada `range(6)` nilainya adalah dari 0 hingga ke 5

# Perulangan: Range

- Secara *default* nilai awal pada range yaitu 0, apabila ingin menambahkan parameter kita bisa menggunakan range(nilai\_awal, nilai\_akhir)

```
1 for i in range(2, 6):  
2     print(i)
```

```
2  
3  
4  
5
```

- Secara default nilai kenaikan urutan adalah 1, namun kita bisa mengubahnya dengan menambahkan parameter range(nilai\_awal, nilai\_akhir, jumlah\_kenaikkan)

```
1 for i in range(0, 10, 2):  
2     print(i)
```

```
0  
2  
4  
6  
8
```



# Perulangan: Range

- Selain menggunakan range untuk pengulangan secara naik ke atas, kita juga bisa melakukan pengulangan menurun

```
1 for i in range(10, 0, -1):  
2     print(i)
```

```
10  
9  
8  
7  
6  
5  
4  
3  
2  
1
```



06

# Nested Loops

# Perulangan: Nested Loops

- *Nested loop* atau *looping* bersarang adalah adanya sebuah *looping* dalam *looping*
- *Looping* dalam akan dieksekusi sebanyak satu kali setiap iterasi dari *looping* luar

```
1 buah = ["apel", "belimbing", "ceri"]  
2 angka = [1, 2]  
3  
4 for i in buah:  
5     for j in angka:  
6         print(i)  
7         print(j)
```

```
apel  
1  
apel  
2  
belimbing  
1  
belimbing  
2  
ceri  
1  
ceri  
2
```



07

# Loop with Conditional Statement

# Perulangan: If Else di Loop

- Looping juga bisa dikombinasikan dengan *conditional statement*
- Contoh penggunaan if dan continue

```
1 string = "Python"
2
3 for karakter in string:
4     if karakter == "y":
5         continue
6     print(f"Huruf: {karakter}")
```

Huruf: P  
Huruf: t  
Huruf: h  
Huruf: o  
Huruf: n

# Perulangan: If Else di Loop

- Melakukan cetak bintang dengan while dan if

```
1 baris = 0
2 bintang = 0
3
4 while baris < 10:
5     if (bintang >= baris + 1):
6         print()
7         baris += 1
8         bintang = 0
9         continue
10    print("*", end="")
11    bintang += 1
```

```
*
**
***
****
*****
*****
*****
*****
*****
*****
```

# Perulangan: Else setelah While

- *Statement else* dapat digunakan untuk menjalankan perintah selama satu kali Ketika kondisi tidak benar

```
1 i = 0
2 while i < 10:
3     print(i)
4     i += 1
5 else:
6     print(f"Nilai i sudah tidak memenuhi kondisi karena i = {i}")
```

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

Nilai i sudah tidak memenuhi kondisi karena i = 10



07

# Loop di Type Data



# Perulangan: Looping di List

- For digunakan untuk mengulangi urutan dan bisa digunakan di beberapa tipe data seperti *list*, *tuple*, *dictionary*, *set*, dan *string*
- Mengakses item di suatu *list* menggunakan *for*

```
1 list = ["apel", "anggur", "mangga", "jeruk", "melon"]
2
3 # tanpa range
4 for item in list:
5     print(item)
```

```
apel
anggur
mangga
jeruk
melon
```

```
1 list = ["apel", "anggur", "mangga", "jeruk", "melon"]
2
3 # dengan range
4 for item in range(len(list)):
5     print(list[item])
```

```
apel
anggur
mangga
jeruk
melon
```

# Perulangan: Looping di List

- Selain hanya untuk mengakses setiap itemnya, kita juga bisa melakukan looping untuk mengubah karakter item dalam suatu *list* menjadi huruf kapital

```
1 list = ["apel", "anggur", "mangga", "jeruk", "melon"]
2 listBaru = []
3
4 for i in list:
5     listBaru.append(i.upper())
6
7 print(listBaru)
```

['APEL', 'ANGGUR', 'MANGGA', 'JERUK', 'MELON']

# Perulangan: Looping di Tuple

- Mengakses item dari suatu tuple

```
1 tuple = ("apel", "anggur", "mangga", "jeruk", "melon")
2
3 for item in tuple:
4     print(item)
```

apel  
anggur  
mangga  
jeruk  
melon

```
1 tuple = ("apel", "anggur", "mangga", "jeruk", "melon")
2
3 for item in range(len(tuple)):
4     print(tuple[item])
```

apel  
anggur  
mangga  
jeruk  
melon

# Perulangan: Looping di Set

- Karena set bersifat *unordered* maka urutan item yang diakses tidak akan teratur susunannya
- Karena bersifat *unindexed* juga, maka tidak bisa menggunakan index untuk mengakses set

```
1 set = {"apel", "anggur", "mangga", "jeruk", "melon"}  
2  
3 for item in set:  
4     print(item)
```

```
anggur  
mangga  
apel  
jeruk  
melon
```

# Perulangan: Looping di Dictionary

- Saat melakukan *looping* di tipe data *dictionary*, nilai yang akan dikembalikan adalah *keys* namun terdapat metode untuk mengembalikan *values* dari *dictionary* juga

```
1 kucing = {  
2     "nama": "Kuro",  
3     "umur": 2,  
4     "ras": "Persian",  
5     "jantan": True,  
6     "hobi": ["makan", "tidur"]  
7 }  
8  
9 for kunci in kucing:  
10    print(kunci)
```

```
nama  
umur  
ras  
jantan  
hobi
```

```
1 kucing = {  
2     "nama": "Kuro",  
3     "umur": 2,  
4     "ras": "Persian",  
5     "jantan": True,  
6     "hobi": ["makan", "tidur"]  
7 }  
8  
9 for i in kucing:  
10    print(kucing[i])
```

```
Kuro  
2  
Persian  
True  
['makan', 'tidur']
```

# Perulangan: Looping di Dictionary

- Metode lain untuk mengakses *key* dan *value* dari suatu *list* menggunakan `keys()` dan `values()`

```
1 kucing = {  
2     "nama": "Kuro",  
3     "umur": 2,  
4     "ras": "Persian",  
5     "jantan": True,  
6     "hobi": ["makan", "tidur"]  
7 }  
8  
9 for x in kucing.keys():  
10     print(x)
```

nama  
umur  
ras  
jantan  
hobi

```
1 kucing = {  
2     "nama": "Kuro",  
3     "umur": 2,  
4     "ras": "Persian",  
5     "jantan": True,  
6     "hobi": ["makan", "tidur"]  
7 }  
8  
9 for x in kucing.values():  
10     print(x)
```

Kuro  
2  
Persian  
True  
['makan', 'tidur']

# Perulangan: Looping di Dictionary

- Metode lain untuk mengakses *key* dan *value* dari suatu list

```
1 kucing = {  
2     "nama": "Kuro",  
3     "umur": 2,  
4     "ras": "Persian",  
5     "jantan": True,  
6     "hobi": ["makan", "tidur"]  
7 }  
8  
9 for x, y in kucing.items():  
10     print(x, y)
```

```
nama Kuro  
umur 2  
ras Persian  
jantan True  
hobi ['makan', 'tidur']
```

# Quiz

1. Isilah tiga bagian code yang kosong:

```
buah = ["apel", "anggur", "mangga", "jeruk", "melon"]  
____ x ____ ____:  
    print(x)
```

2. Isilah bagian code yang kosong untuk melakukan *looping* dimana ketika nilai x adalah 5 maka program langsung ke iterasi berikutnya:

```
x = 0  
____ x < 10:  
    x += 1  
    if i ____ 5:  
        ____  
    print(i)
```





07

# Studi Kasus

# Studi Kasus

Leni kembali menemui sebuah masalah baru. Kali ini dia ingin sekali mengetahui deret angka dari 1 sampai 50 dimana ketika bertemu dengan kelipatan 5 maka outputnya adalah “pass”. Tolong bantu Leni dengan masalahnya kali ini.

# Terimakasih

Do you have any questions?

addyouremail@freepik.com

+91 620 421 838

yourcompany.com

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

Please keep this slide for attribution