

Dosen Pengampu: Adri Arisena



PRAKTIKUM 2

Pengulangan dalam Python

Asisten Laboratorium:
Najlia Intani



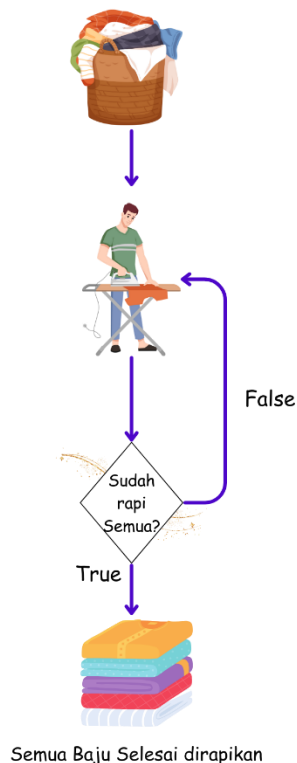
"Modul ini diperuntukkan bagi Mata Kuliah Pemrograman Dasar dan Basis Data Program Studi Agribisnis"

A. PENDAHULUAN LOOP/PERULANGAN

Secara umum, pernyataan pada bahasa pemrograman akan dieksekusi secara berurutan. Pernyataan pertama dalam sebuah fungsi dijalankan pertama, diikuti oleh yang kedua, dan seterusnya. Tetapi akan ada situasi dimana Anda harus menulis banyak kode, dimana kode tersebut sangat banyak. Jika dilakukan secara manual maka Anda hanya akan membuang-buang tenaga dengan menulis beratus-ratus bahkan beribu-ribu kode. Untuk itu Anda perlu menggunakan pengulangan di dalam bahasa pemrograman Python.

Iterasi merupakan proses eksekusi blok kode secara berulang sampai suatu kondisi tidak terpenuhi lagi atau ketika adanya sintaks yang menghentikannya dari dalam blok tersebut.

Perhatikan diagram analogi berikut:



Seseorang memiliki setumpuk pakaian yang kusut dan perlu disetrika. Dia mengambil satu potong pakaian, meletakkannya di papan setrika, menyetrika bagian depan, menyetrika bagian belakang, dan melipat pakaiannya. Dia mengulangi aksi ini untuk setiap potong pakaian di tumpukan. Setiap potong pakaian yang dia setrika adalah satu iterasi dalam perulangan. Dia berhenti menyetrika setelah semua pakaian sudah disetrika.

B. WHILE LOOP

Pengulangan While Loop di dalam bahasa pemrograman Python dieksekusi statement berkali-kali selama kondisi bernilai benar atau True. Blok lainnya yang dapat digunakan untuk melakukan iterasi adalah while. Berbeda dengan for loop, iterasi menggunakan while memerlukan pengecekan kondisi bernilai True agar dapat dieksekusi. Selama kondisi tersebut masih terpenuhi (bernilai True) maka blok while akan terus dieksekusi. Oleh karena itu, penting untuk memastikan terdapat kode untuk merubah kondisi pada while menjadi False, jika tidak maka iterasi akan berlangsung tanpa henti.

Struktur perulangan While adalah sebagai berikut:

```
while kondisi :  
    blok pernyataan yang akan diulang selama kondisi True
```

Contoh 1

```
#Contoh penggunaan While Loop  
#Catatan: Penentuan ruang lingkup di Python bisa menggunakan tab alih-alih  
menggunakan tanda kurung  
  
count = 0  
while (count < 9):  
    print ("The count is: ", count)  
    count = count + 1  
  
print ("Good bye!")
```

Contoh 2

```
a = 3  
# blok while akan dieksekusi  
# selama a > 0  
while a > 0:  
    print("a =", a, "> 0")  
    a = a - 1  
    print("    a sekarang :", a)
```

Contoh 3

```
# contoh INFINITE LOOP
# !!!hentikan program secara manual!!!
# nilai b = 1 dan 1 > 0
# karena nilai b tidak pernah berubah di dalam loop
# maka kondisi akan selalu bernilai True
b = 1      # b akan selalu bernilai 1
while b > 0:
    print("infinite loop")
```

Contoh 4

```
jawab = 'ya'
hitung = 0

while(jawab == 'ya'):
    hitung += 1
    jawab = input("Ulang lagi tidak? ")

print("Total perulangan: ",hitung)
```

C. FOR LOOP

Pengulangan for pada Python memiliki kemampuan untuk mengulangi item dari urutan apapun, seperti list atau string. Pada for loop, iterasi dilakukan sebanyak n kali tergantung banyaknya elemen pada objek iterable yang digunakan. Contoh objek iterable pada python adalah range, list dan tuple.

Struktur perulangan For adalah sebagai berikut:

```
for variable in iterable:
    blok kode yang dijalankan untuk setiap item dalam iterable
```

Contoh 5

```
#Contoh pengulangan for sederhana
angka = [1,2,3,4,5]
for x in angka:
    print(x)
```

Contoh 6

```
#Contoh pengulangan for
buah = ["nanas", "apel", "jeruk"]
for makanan in buah:
    print ("Saya suka makan", makanan)
```

Apabila ingin melakukan perulangan berdasarkan rentang suatu nilai, dapat menggunakan fungsi `range()`.

Struktur dari fungsi `range()` adalah sebagai berikut:

```
range(start, stop, step)
```

Contoh 7

```
ulang = 10

for i in range(ulang):
    print(f"Perulangan ke-{i}")
```

Contoh 8

```
# mencetak "Pikachu" sebanyak 5x
for _ in range(5):
    print("Pikachu")
```

Contoh 9

```
# mencetak nilai kuadrat 1-10 dengan step=2
for val in range(1, 10, 2):
    val_kuadrat = val**2
    print(f"{val}^2 adalah {val_kuadrat}")
```

Contoh 10

```
# iterasi pada list
skills = ["Python", "R", "Julia", "Scala"]
for skill in skills:
    print("Anda menguasai", skill)
print("-----")
# iterasi pada list dengan menyertakan indeks (key)
for idx, skill in enumerate(skills):
    print(f"Skill ke-{idx+1} : {skill}")
```

D. NESTED LOOP

Bahasa pemrograman Python memungkinkan penggunaan satu lingkaran di dalam loop lain. Bagian berikut menunjukkan beberapa contoh untuk menggambarkan konsep tersebut.

Struktur Nested Loop adalah sebagai berikut:

```
for variabel1 in iterable1:
    for variabel2 in iterable2:
        blok kode yang dijalankan untuk setiap item dalam iterable
```

Contoh 11

```
# Program Python dengan Nested Loop

# Input jumlah baris
baris = int(input("Masukkan jumlah baris: "))

# Loop untuk membuat pola
for i in range(1, baris + 1):
    for j in range(1, i + 1):
        print("*", end=" ")
    print() # Pindah ke baris berikutnya
```

Contoh 12

```
#Menamai Pasangan Hewan
hewan1 = ["Kucing", "Kelinci", "Hamster"]
hewan2 = ["Besar", "Kecil"]
print("Pasangan hewan:")
for h1 in hewan1:
    for h2 in hewan2:
        print(h1, h2)
```

Contoh 13

```
#Daftar Hari dan Waktu
hari = ["Senin", "Selasa", "Rabu"]
waktu = ["Pagi", "Siang", "Malam"]
print("Jadwal hari dan waktu:")
for h in hari:
    for w in waktu:
        print(h, w)
```

Contoh 14

```
#Permainan Tebak Angka
angka_rahasia = [1, 3, 5]
angka_tebakan = [2, 3, 4, 5]
print("Hasil tebakkan angka:")
for rahasia in angka_rahasia:
    for tebakkan in angka_tebakan:
        if rahasia == tebakkan:
            print(f"Benar! Angka {tebakkan} cocok.")
        else:
            print(f"Angka {tebakkan} salah.")
```

E. ELSE, BREAK, CONTINUE

Pada proses iterasi menggunakan for loop maupun while loop dapat ditambahkan beberapa sintaks lainnya yang dapat merubah aliran proses didalamnya yaitu continue dan break. Selain itu kita juga dapat menambahkan blok else pada for loop maupun while loop.

continue: keyword continue akan menghentikan eksekusi sisa blok untuk iterasi yang sedang berjalan dan melanjutkan lagi proses untuk iterasi berikutnya.

break: keyword break berfungsi untuk menghentikan total iterasi walaupun kondisi pada while loop masih True atau elemen pada for loop masih ada.

else: blok ini dapat ditambahkan pada for maupun while dan akan dieksekusi ketika kondisi pada while loop menjadi False atau elemen list pada for loop sudah habis.

Contoh 15

```
# ===== WHILE DENGAN ELSE ===== #

b = 1
while b < 5:
    print("b masih lebih kecil dari 5")
    b = b + 1
else:
    print("b sekarang sama dengan 5")

# break : menghentikan iterasi total

print("\n-----BREAK-----")

skill = ["Python", "R", "Julia", "Scala"]      # data dalam bentuk list

print("start loop")
for val in skill:
    # jika val = "Julia" hentikan iterasi
    if val == "Julia":
        break

    print("  Bahasa", val)

print("end loop")

# continue ; menghentikan iterasi saat ini
# dan langsung melanjutkan iterasi berikutnya

print("\n-----CONTINUE-----")
```



```
print("start loop")
a = 0
while a < 5:
    a = a + 1

    # jika nilai a 2 atau 3 : continue
    if a == 2 or a == 3:
        continue

    print("  nilai a =", a)

print("end loop")
```

LATIHAN PRAKTIKUM 2

Seorang pelanggan berbelanja di sebuah toko. Ia membeli beberapa barang, dan total belanja dihitung berdasarkan harga barang dan jumlah barang yang dibeli. Anda diminta membuat program Python yang menggunakan looping untuk menghitung total belanja dari beberapa barang.

Ketentuan:

1. Pelanggan memasukkan jumlah barang yang dibeli.
2. Untuk setiap barang, pelanggan memasukkan harga satuan dan jumlah unit barang tersebut.
3. Program menghitung total harga untuk setiap barang dan menjumlahkan semuanya untuk menghitung total belanja.