# RELATIONAL DAN OPERATOR LOGIS

PERTEMUAN 9

# NILAI-NILAI KARAKTER ASCIL

Masing-masing dari 47 tombol di bagian tengah mesin tik keyboard dapat menghasilkan dua karakter, dengan total 94 karakter. Menambahkan 1 untuk karakter yang dihasilkan oleh spasi membuat 95 karakter. Terkait dengan karakter-karakter ini adalah angka mulai dari 32 hingga 126. Nilai-nilai ini, yang disebut nilai ASCII karakter, Tabel 6.1 menunjukkan beberapa nilai ASCII.

**Tabel 6.1.** Karakter Nilai-nilai ASCII

| No | Karakter | No | Karakter |
|---|---|---|---|
| 32 | (space) | 66 | B |
| 33 | ! | 90 | Z |
| 34 | “ | 97 | A |
| 35 | # | 98 | b |
| 48 | 0 | 122 | z |
| 49 | 1 | 123 | { |
| 57 | 9 | 125 | } |
| 65 | A | 126 | ~ |

Standar ASCII juga memberikan karakter ke beberapa angka di atas 126. Tabel 6.2 menunjukkan beberapa nilai ASCII yang lebih tinggi.

**Tabel 6.2.** Karakter Nilai-nilai ASCII yang lebih tinggi

| No | Karakter | No | Karakter |
|---|---|---|---|
| 162 | ¢ | 181 | µ |
| 169 | © | 188 | ¼ |
| 176 | ° | 189 | ½ |
| 177 | ± | 190 | ¾ |
| 178 | ² | 247 | ÷ |
| 179 | ³ | 248 | Ø |

Jika n adalah angka non-negatif, maka

```
chr(n)
```

adalah string karakter tunggal yang terdiri dari karakter dengan nilai ASCII n. Jika str adalah string karakter tunggal, maka

```
ord(str)
```

adalah nilai karakter dari ASCII. Misalnya, pernyataan

```
print(chr(65))
```

menampilkan angka 65

```
print(ord('A'))
```

menampilkan huruf A.

Rangkaian dapat digunakan dengan ***chr*** untuk mendapatkan string menggunakan karakter ASCII yang lebih tinggi. Misalnya, pernyataan

```
print("32" + chr(176) + " Fahrenheit")
```

menampilkan 32° Fahrenheit.

# OPERATOR RELASIONAL

Operator relasional kurang dari (<) dapat diterapkan pada angka, string, dan objek lainnya. Suatu angka ***a*** dikatakan lebih kecil dari angka ***b*** jika a terletak di sebelah kiri ***b*** pada garis bilangan. Misalnya, 2 < 5, -5 < -2, dan 0 < 3.5.

String ***a*** dikatakan kurang dari string ***b*** jika ***a*** mendahului ***b*** saat menggunakan karakter ASCII. Digit mendahului huruf besar, yang mendahului huruf kecil. Dua string dibandingkan karakter dengan karakter (bekerja dari kiri ke kanan) untuk menentukan string mana yang harus mendahului yang lain. Jadi, "kucing" < "anjing", "kereta" < "kucing", "kucing" < "katalog", "9W" < "kelelawar", "Anjing" < "kucing", dan "sales_99" < "sales_retail". Jenis statemen ini disebut ***leksikografis***. Tabel 6.3. menunjukkan operator relasional yang berbeda dan artinya.

**Tabel 6.3.** Operator Relasional

| Operator | Contoh | Penjelasan |
|---|---|---|
| Sama dengan == | 1 == 1 | bernilai True Jika masing-masing operan memiliki nilai yang sama, maka kondisi bernilai benar atau True. |
| Tidak sama dengan != | 2 != 2 | bernilai False Akan menghasilkan nilai kebalikan dari kondisi sebenarnya. |
| Lebih besar dari > | 7 > 3 | bernilai True Jika nilai operan kiri lebih besar dari nilai operan kanan, maka kondisi menjadi benar. |
| Lebih kecil dari < | 7 < 3 | bernilai True Jika nilai operan kiri lebih kecil dari nilai operan kanan, maka kondisi menjadi benar. |
| Lebih besar atau sama dengan >= | 7 >= 3 | bernilai True Jika nilai operan kiri lebih besar dari nilai operan kanan, atau sama, maka kondisi menjadi benar. |
| Lebih kecil atau sama dengan <= | 7 <= 3 | bernilai True Jika nilai operan kiri lebih kecil dari nilai operan kanan, atau sama, maka kondisi menjadi benar. |
| in, not in | | Membership (Keanggotaan) |

# CONTOH

```
# buat variabel a dan b dengan teknik squence ordering
a, b = 5, 10
print(a, '>', b, '=', a > b)
print(a, '<', b, '=', a < b)
print(a, '==', b, '=', a == b)
print(a, '!=', b, '=', a != b)
print(a, '>=', b, '=', a >= b)
print(a, '<=', b, '=', a <= b)
```

```
5 > 10 = False
5 < 10 = True
5 == 10 = False
5 != 10 = True
5 >= 10 = False
5 <= 10 = True
```

# Operator Logis

- Operator logika adalah operator yang sangat penting. Operator ini sangat **berkaitan erat** dengan operator perbandingan. Dan kedua-duanya juga mengembalikan nilai dengan tipe data yang sama yaitu boolean.

# Berikut ini tabel dari operator logika pada python.

| Simbol | Tugas | Contoh |
| --- | --- | --- |
| `and` | Mengembalikan `True` jika dua statement sama-sama benar | `True and True` |
| `or` | Mengembalikan `True` jika salah satu statement bernilai benar | `2 > 5 or 1 < 3` |
| `not` | Menegasikan hasil. `True` menjadi `False` dan sebaliknya | `not(1 > 5)` |

# Contoh.

```
print(True and True)
print(1 + 2 == 3 and True)
print('----')
print(False or 1 > 5)
print(False or 5 > 2)
print('----')
print(not(1 > 5))
print(not(1 < 5))
```

```
True
True
----
False
True
----
True
False
```

# Tipe data Boolean/Bool

Selain tipe data ***int*** dan **float,** dalam python juga ada tipe data boolean . Tipe data ini berupa biner yang berisi dengan salah satu dari 2 nilai: True atau False. Tipe data boolean banyak dipakai untuk percabangan kode program atau untuk memutuskan apa yang mesti dijalankan ketika sebuah kondisi terjadi. Sebagai contoh, kita bisa membuat kode program untuk menentukan apakah status kelulusan berdasarkan input dari pengguna. Untuk keperluan ini kita harus mengecek terlebih dahulu apakah status tersebut bisa dibagi 2 (untuk status lulus atau tidak). Tipe data boolean bisa dipakai untuk menampung kondisi seperti ini, yakni benar atau salah (*True* atau *False*).

Pernyataan program

```
print(condition)
```

akan menampilkan Benar atau Salah. Objek Benar dan Salah dikatakan memiliki tipe data Boolean atau tipe data bool. Baris-baris berikut kode menampilkan hasil **False**

```
x = 5
print((3 + x) < 7)
```

Baris-baris kode berikut menampilkan hasil **True**:

```
x = 2
y = 3
var = x < y
print(var)
```

# Tiga Metode yang mengembalikan nilai boolean

Jika str1 dan str2 adalah string, maka kondisinya

```
str1.startswith(str2)
```

memiliki nilai True jika dan hanya jika str1 dimulai dengan str2, dan kondisinya

```
str1.endswith(str2)
```

memiliki nilai True jika dan hanya jika str1 diakhiri dengan str2.

Misalnya, dua kondisi berikut ini benar:

```
"fantastic".startswith("fan")
"fantastic".endswith("stic")
```

Jika var1 memiliki nilai "fantastis" dan var2 memiliki nilai "Fant", maka dua kondisi berikut ini salah:

```
var1.startswith(var2)
"elephant".endswith(var2)
```

Jika item adalah literal atau variabel, maka kondisi formulir

```
isinstance(item, dataType)
```

memiliki nilai **True** jika dan hanya jika nilai item memiliki tipe data yang ditentukan, di mana tipe data adalah semua tipe data (seperti int, float, str, bool, list, atau tuple). Sebagai contoh, kondisi isinstance ("32", int) memiliki nilai **False** dan kondisi isinstance (32, int) memiliki nilai **True**.

Tabel 6.5 menunjukkan beberapa metode string lain yang mengembalikan nilai Boolean. Dalam tabel, asumsikan bahwa str1 bukan string kosong. Setiap metode dalam tabel mengembalikan False ketika str1 adalah string kosong.

**Tabel 6.5.** Metode yang mengembalikan **True** atau **False**

| Metode | Pengembalian True saat |
|---|---|
| **str1.isdigit()** | semua karakter str1 adalah digit |
| **str1.isalpha()** | semua karakter str1 adalah huruf-huruf alfabet |
| **str1.isalnum()** | semua karakter str1 adalah huruf alfabet atau digit |
| **str1.islower()** | str1 memiliki setidaknya 1 karakter alfabet dan semua karakter alfabetnya adalah huruf kecil |
| **str1.isupper()** | str1 memiliki setidaknya 1 karakter alfabet dan semua karakter alfabetnya adalah huruf besar |
| **str1.isspace()** | str1 hanya berisi karakter spasi |

# THANKS