

7 Aturan Sintaks 🐍

Bahasa pemrograman Python adalah bahasa yang dieksekusi oleh sebuah interpreter. Interpreter tersebut bertugas untuk memarsing sintaks python, dan kemudian mengkonversinya menjadi sebuah instruksi mesin satu persatu (maksudnya adalah satu baris-satu baris) [\[1\]](#).

Terdapat beberapa aturan penulisan sintaks pada bahasa pemrograman Python. Aturan-aturan ini harus ditaati oleh programmer agar interpreter bisa memarsing dan menjalankan aplikasi dengan baik. Jika tidak, maka aplikasi akan memproduksi sebuah error.

Apa saja aturan-aturan penulisan sintaks python?

Aturan penulisan python terbagi menjadi beberapa sub bahasan; mulai dari struktur baris kode, statemen, komentar, penugasan, indentasi, dan lain sebagainya.

Secara umum, sintaks penulisan python bersifat:

- **Case Sensitive**
- **Tidak Menggunakan Titik Koma**
- **Indentasi Sebagai Pembentuk Struktur**
- **Tidak ketat terhadap tipe data**
- **Human friendly**

Kita akan bahas satu persatu.

NB: Beberapa contoh dari tutorial ini diambil dari materi-materi yang akan datang. Sehingga sangat wajar jika kita masih belum memahami beberapa istilah atau beberapa kode program yang dijadikan contoh.

Case Sensitive

Bahasa pemrograman python bersifat *case sensitive*. Ia akan membedakan antara huruf kecil dan huruf besar walaupun sebuah kata itu terlihat sama.

Contoh:

```
ibu_kota = 'Jakarta'  
print(iBu_kota)
```

Jika dijalankan, kita akan mendapatkan error:

```
Exception has occurred: NameError
name 'iBu_kota' is not defined
File "case-sensitive.py", line 3, in <module>
    print(iBu_kota)
```

Kenapa?

Karena variabel yang kita definisikan adalah `ibu_kota`, dengan huruf kecil semuanya. Sedangkan variabel yang berusaha kita panggil adalah `iBu_kota` yang mana huruf `B` nya adalah huruf kapital, dan interpreter python menganggap keduanya berbeda.

Tidak Menggunakan Titik Koma

Sebelum membahas lebih lanjut tentang titik koma, ada baiknya kita mengetahui terlebih dahulu pengertian statement pada bahasa pemrograman (lebih khusus pada python).

Apa itu statement?

Statemen adalah sebuah pernyataan atau instruksi yang akan dieksekusi oleh mesin [\[2\]](#). Interpreter python akan bertugas untuk menginterpretasikan setiap statemen menjadi perintah mesin yang sesuai.

Penulisan Statement

Di dalam python, penulisan antar statemen **tidak diakhiri dengan titik koma** –berbeda dengan bahasa pemrograman lain pada umumnya di mana setiap statement akan dibedakan berdasarkan adanya karakter titik koma (`;`).

Lalu, bagaimana cara interpreter python membedakan antar satu statemen dengan statemen lainnya?

Dengan karakter ganti baris (`\n`).

Setiap pergantian baris, interpreter akan menganggap bahwa **sebuah statemen telah sempurna**.

Perhatikan contoh berikut:

```
a = 5
b = 8
c = a + b

print(c)
```

Pada program di atas, terdapat 4 buah statemen. Dan setiap statemen dipisahkan oleh sebuah karakter `\n` atau karakter yang menandakan bahwa suatu baris telah selesai.

Kita tetap bisa menggunakan titik koma

Pada kasus-kasus tertentu, kita tetap bisa menggunakan titik koma.

Kapan?

Ketika terdapat **lebih dari 1 statemen** dalam satu baris.

Perhatikan contoh berikut:

```
a = 5; b = 8; c = a + b;
print(c)
```

Pada contoh di atas, terdapat 5 buah statement:

- 4 di antaranya ditulis dalam satu baris sekaligus.
- dan sisanya ditulis pada baris yang berbeda.

Satu Statemen Multi Baris

Kebalikannya...

Kita bisa memecah satu statemen menjadi lebih dari satu baris.

Dalam kasus-kasus tertentu, satu statemen bisa sangat panjang. Sehingga kode programnya melebihi ukuran layar.

Contoh:


```
kondisi = 10 < 5 and 10 > 9 or 11 == 6 + 5 and 0 == 100 * 5 / (25 - 15)
```

Hal ini tentu saja membuat kode program menjadi lebih susah dibaca, didebug, dan dipahami.

Kita inginnya kode program terlihat lebih ringkas dan mudah dibaca.

Sedangkan jika kita buat baris baru, statemen di atas akan terputus, yang kemudian akan dianggap error oleh interpreter python.

Solusinya?

Kita bisa memecah satu statemen panjang menjadi multiple baris dengan tanda *backslash* ()

Seperti contoh berikut:

```
kondisi = 10 < 5 \
and 10 > 9 \
or 11 == 6 + 5 \
and 0 == 100 * 5 / (25 - 15)
```

Meskipun kode program di atas memiliki total 4 baris, tapi interpreter tetap menganggapnya sebagai satu statemen utuh.

Indentasi Sebagai Pembentuk Struktur

Pada bahasa pemrograman lain, umumnya indentasi adalah sesuatu yang tidak penting. Bahkan cenderung diabaikan oleh mesin.

Indentasi hanya digunakan untuk mempermudah manusia dalam membaca kode program.

Tapi tidak dengan python.

Dalam python, **indentasi adalah hal yang super-super penting**, karena ia bertugas untuk mendefinisikan struktur blok kode program.

Sehingga, **melakukan kesalahan indentasi** juga bisa berujung pada sebuah error (yang mungkin akan sulit dipecahkan bagi yang belum terbiasa). Atau dalam istilah lain: kita menggunakan indentasi untuk mengelompokkan blok kode program di dalam python.

Apa itu Indentasi?

Indentasi adalah penulisan paragraf yang agak menjorok masuk ke dalam [\[3\]](#). Biasanya jika kita membaca majalah atau koran, kita akan dapati indentasi pada kalimat awal setiap paragrafnya.

Kesalahan-Kesalahan Penulisan Indentasi pada Python

Untuk memahami betapa pentingnya indentasi pada python, mari kita perhatikan contoh-contoh berikut tentang kesalahan indentasi yang umum terjadi pada bahasa pemrograman python.

Perhatikan contoh berikut:

```
print('Selamat')
    print('Pagi')
print('Dunia')
```

Kode program di atas, secara sekilas tidak ada masalah. Tapi, ketika kita eksekusi, ternyata interpreter python memberitahukan kita bahwa terdapat error yang disebabkan oleh indentasi yang tidak pada tempatnya:

```
IndentationError: unexpected indent
```

Yang benar harusnya tiga perintah `print()` berada pada satu indentasi, karena memang ketiganya berada dalam satu blok yang sama.

```
print('Selamat')
print('Pagi')
print('Dunia')
```

Contoh kode program yang benar ✓:

```
a = 10

if a > 5:
    print('nilai a lebih dari 5')
```

Contoh yang salah ✗:

```
a = 10

if a > 5:
print('nilai a lebih dari 5')
```

Pesan error:

```
IndentationError: expected an indented block
```

Contoh yang salah tapi samar (kalau di-copy paste kemungkinan akan tetap working) ✗:

```
a = 10

if a > 5:
    print('nilai a lebih dari 5')
print('nilai a lebih dari 5')
```

Pesan error:

```
TabError: inconsistent use of tabs and spaces in indentation
```

Kenapa yang terakhir error?

Karena ke-tidak-konsisten-an penggunaan tab. Yang pertama menggunakan **4 spasi untuk tab**, sedangkan yang kedua menggunakan **1 tab asli (bukan spasi)**.

Meski secara visual terlihat sama, tapi tidak bagi interpreter python.

Oiya, jika kita menggunakan teks editor atau IDE yang canggih, **error seperti ini tidak akan terjadi** karena IDE atau Teks Editor yang kita gunakan akan otomatis mengkonversi semua tab menjadi seragam, entah menggunakan spasi atau menggunakan **tab** itu sendiri.

Error seperti ini biasanya akan terjadi jika:

- Kita melakukan *copy-paste* kode program dari internet
- Dan kita tidak menggunakan teks editor atau IDE yang kekinian

Tidak Ketat Terhadap Tipe Data

Sifat yang berikutnya adalah: insensitifitas terhadap tipe data.

Artinya:

Kita bisa memberi dan mengubah nilai apa pun dari tipe data apa pun ke dalam sebuah variabel.

Bagi yang baru mempelajari bahasa pemrograman, ini mungkin biasa saja. Tapi bagi *old-school* yang pernah mempelajari Java dan semisalnya, akan merasa aneh. Karena pada bahasa-bahasa tersebut, satu variabel hanya diperuntukkan untuk satu tipe data saja.

Contoh ketidak-ketat-an python terhadap tipe data:

```
# nilai awal berupa integer
a = 5
# kita ubah menjadi string dan tidak error
a = 'Nurul Huda'
```

Tanda Petik dan Tanda Petik Dua

Dalam bahasa pemrograman python, kita bisa mendefinisikan string dengan tanda petik satu `'` maupun tanda petik dua `"`.

Contoh yang benar ✓:

```
nama = 'Nurul Huda'
asal = "Indonesia"
```

Contoh yang salah ✗:

```
nama = 'Nurul Huda"
asal = "Indonesia'
```

Penulisan Komentar

Komentar adalah sebuah baris kode atau statemen yang diabaikan oleh interpreter python. Ia hanya ditulis dengan tujuan **agar dibaca oleh manusia**, bukan mesin.

Komentar juga sangat penting sebagai penjelasan alur dari kode program yang kita tulis. Jika tidak, kita sendiri (si penulis kode) bisa lupa dan kebingungan jika harus menjelaskan kode program lama yang pernah kita tulis pada masa lalu.

Penulisan komentar pada python **terdiri dari 2 jenis**:

- satu baris
- dan multi baris

Komentar **satu baris** ditulis dengan tanda **#**. Sedangkan komentar **lebih dari satu baris** ditulis dengan *triple doublequote* (tanda petik dua sebanyak 3x).

Contoh:

```
# variabel a merepresentasikan panjang
a = 5
b = 10 # variabel b merepresentasikan tinggi

"""
Dan variabel c merepresentasikan luas
persegi dari hasil perkalian
variabel a dan variabel b
"""
c = a * b
```