



# Python Basic Style Guide - Statement Gabungan, Trailing Commas, dan Anotasi Fungsi

## Statement gabungan

Usahakan untuk tidak menggabungkan >1 statement pada baris yang sama.

### Disarankan:

```
1. if foo == 'blah':
2.     do_blah_thing()
3. do_one()
4. do_two()
5. do_three()
```

### Tidak disarankan:

```
1. if foo == 'blah': do_blah_thing()
2. do_one(); do_two(); do_three()
```

Anda diperbolehkan untuk membuat sebuah konten/isi dari if/for/while yang cukup pendek untuk diletakkan dalam satu baris (program tetap berjalan). Namun pastikan tidak melakukannya jika if/for/while Anda bertingkat atau bersifat multi clause, misalnya if-else, try-finally, dan sebagainya.

### Tidak disarankan:

```
1. if foo == 'blah': do_blah_thing()
2. for x in lst: total += x
3. while t < 10: t = delay()
```

### Sangat tidak disarankan:



DIBANTU



```
2. else: do_non_blah_thing()
3. try: something()
4. finally: cleanup()
5. do_one(); do_two(); do_three(long, argument,
6.                               list, like, this)
7. if foo == 'blah': one(); two(); three()
```

## Penggunaan Trailing Commas

Koma di bagian akhir (trailing commas) umumnya bersifat opsional, satu statemen dimana ia bersifat wajib adalah saat kita membuat variabel menggunakan tipe tuple dengan satu elemen. Hal ini umumnya diperjelas dengan kurung untuk menghindari penghapusan atau pembersihan.

### Disarankan:

```
1. FILES = ('setup.cfg',)
```

### Tidak disarankan:

```
1. FILES = 'setup.cfg',
```

Saat trailing comma bersifat redundan, Anda akan merasakan kemudahannya saat menggunakan VCS (*Version Control System*), atau pada kode yang mungkin Anda tambahkan dalam beberapa waktu kedepan. Pola yang disarankan adalah meletakkan nilai atau string pada sebuah baris baru, mengikuti indentasi yang ada, dan menambahkan trailing comma, dan menutup kurung/kurawal/siku pada baris selanjutnya.

Tidak umum jika Anda meletakkan trailing comma pada baris di mana Anda menutup kurung/kurawal/siku, kecuali dalam tuple dengan satu elemen.

### Disarankan:

```
1. FILES = [
2.     'setup.cfg',
3.     'tox.ini',
4. ]
5. initialize(FILES,
6.             error=True,
7.             )
```

### Tidak disarankan:



DIBANTU



```
2. initialize(FILES, error=True,)
```

## Anotasi Fungsi

Penggunaan anotasi fungsi sebaiknya menggunakan aturan baku untuk titik dua (:) dan menggunakan spasi untuk penggunaan ->. Hal ini dijelaskan lebih lanjut di [PEP 484](#).

```
1. Yes:
2. def munge(input: AnyStr): ...
3. def munge() -> AnyStr: ...
4.
5. No:
6. def munge(input:AnyStr): ...
7. def munge()->PosInt: ...
```

Tidak menggunakan tanda sama dengan (=) untuk mengindikasikan keyword argumen atau nilai dasar/default pada parameter fungsi tanpa menggunakan anotasi.

```
1. Yes:
2. def complex(real, imag=0.0):
3.     return magic(r=real, i=imag)
4.
5. No:
6. def complex(real, imag = 0.0):
7.     return magic(r = real, i = imag)
```

Ketika melakukan kombinasi argumen anotasi dan nilai dasar/default, Anda justru disarankan untuk menggunakan spasi sebelum dan setelah tanda =.

```
1. Yes:
2. def munge(sep: AnyStr = None): ...
3. def munge(input: AnyStr, sep: AnyStr = None, limit=1000): ...
4.
5. No:
6. def munge(input: AnyStr=None): ...
7. def munge(input: AnyStr, limit = 1000): ...
```

[< Sebelumnya](#)

DIBANTU

[Selanjutnya >](#)



Dicoding Space  
Jl. Batik Kumeli No.50, Sukaluyu,  
Kec. Cibeunying Kaler, Kota Bandung  
Jawa Barat 40123



Decode Ideas  
Discover Potential

[Tentang Kami](#)

[Blog](#)

[Reward](#)

[Showcase](#)



[Tentang Kami](#)

[FAQ](#)

Penghargaan

