



## Pengecekan Style Guide PEP8

Sampai dengan saat ini, Anda tentu sudah menuliskan kode pemrograman Python cukup banyak, termasuk di antaranya kode yang Anda tulis mengalami kesalahan sintaksis atau indentasi. Lalu bagaimana agar ke depan bisa mencegah hal tersebut terjadi?

Bergantung dengan editor yang Anda gunakan untuk menulis kode Python, dimana salah satu fitur yang memudahkan penulisan, sehingga menghemat banyak waktu untuk memformat kode sesuai arahan gaya penulisan (style guide) PEP8 dan mengecek akan kemungkinan terjadinya kesalahan (error) pada kode yang ditulis. Untuk itu akan dijelaskan sejumlah aplikasi yang dapat Anda gunakan, dengan memanggil perintah atau sebaiknya diintegrasikan ke editor kode yang Anda pakai, dengan tujuan untuk membantu Anda mengecek kemungkinan kesalahan dan kesesuaian dengan PEP8.

## Lint

Lint adalah proses pengecekan kode atas kemungkinan terjadi kesalahan (error), termasuk dalam proses ini adalah mengecek kesesuaian terhadap arahan gaya penulisan kode (style guide) PEP8. Aplikasi yang digunakan untuk proses ini disebut linter. Integrasi linter dengan editor kode Anda akan membuat efisien dalam menulis kode Python. Pertimbangan ini karena keluaran dari aplikasi linter hanya berupa teks singkat berupa keterangan dan kode Error atau Warning atau Kesalahan Konvensi Penamaan (Naming Conventions), akan dicontohkan selanjutnya.

Dengan lint atau linting akan meminimalkan kode Anda mengalami error, sebagai contoh salah satunya karena kesalahan indentasi di Python. Sebelum kode Anda diproses oleh interpreter Python dengan IndentationError, lint akan memberitahunya lebih dahulu ke Anda.

Berikut akan dibahas 3 jenis aplikasi linter, silakan dicermati dahulu, tidak harus semuanya diinstal/dicoba, hanya paket yang menurut Anda sesuai kebutuhan saja yang digunakan. Hasil keluarannya mungkin mirip, tapi pada kondisi tertentu akan ada keluaran atau fitur yang mungkin sesuai dengan kebutuhan Anda menulis kode.

## Pycodestyle (sebelumnya bernama pep8)

[Pycodestyle](#) adalah aplikasi open source (berlisensi MIT/Expat) untuk membantu mengecek kode terkait gaya penulisan kode dengan konvensi PEP8.

Untuk instalasi menggunakan manajer paket pip sebagai berikut : `pip install pycodestyle`.



DIBANTU



[Pylint](#) adalah aplikasi open source (berlisensi GPL v2) untuk melakukan analisis kode Python, mengecek untuk kesalahan (error) pemrograman, memaksakan standar penulisan kode dengan mengecek penulisan kode yang tidak baik, serta memberikan saran untuk refactoring sederhana.

Untuk instalasi menggunakan manajer paket pip sebagai berikut : `pip install pylint`.

## Flake8

[Flake8](#) adalah aplikasi open source (berlisensi MIT) yang membungkus sejumlah kemampuan aplikasi lain seperti pycodestyle, pyflakes, dan sejumlah (skrip/fitur) lainnya.

Untuk instalasi menggunakan manajer paket pip sebagai berikut : `pip install flake8`

Selanjutnya, kita akan membahas sebuah contoh kode class sederhana yang akan diproses lint, misal disimpan dalam sebuah file `kal.py`. Perhatikan di baris 5, kolom 5 dan kolom 10. Ditulis berjeda antar huruf agar mudah dicek penomoran kolomnya.

1	class Kalkulator:																																					
2	"""kalkulator tambah kurang"""																																					
3	def __init__(self, _i=0):																																					
4	self.i=_i																																					
5	def tambah(self, _i): return self.i+_i																																					
6	def kurang(self, _i):																																					
7	return self.i-_i																																					
	1				5					1					1				2						2				3				3				4	
										0					5				0						5				0				5				0	

Jika diproses dengan pycodestyle. Menunjukkan error (E301) di baris 5 kolom 5, serta error (E112) di baris 7 kolom 5.

1. `pycodestyle kal.py`
2. `kal.py:5:5: E301 expected 1 blank line, found 0`
3. `kal.py:7:5: E112 expected an indented block`

Jika diproses dengan pylint. Menunjukkan error (E0001) di baris 7 kolom 10.

1. `pylint kal.py`
2. `***** Module kal`
3. `kal.py:7:10: E0001: invalid syntax (<unknown>, line 7) (syntax-error)`

Jika diproses dengan flake8. Menunjukkan error (E301) di baris 5 kolom 5, error (E112) di baris 7 kolom 5, error (E999) di baris 7 kolom 10.





- 2. kal.py:5:5: E301 expected 1 blank line, found 0
- 3. kal.py:7:5: E112 expected an indented block
- 4. kal.py:7:10: E999 **IndentationError**: expected an indented block

Setelah kodenya diperbaiki, kali ini diubah secara manual dengan mengetikkan, pada pembahasan berikutnya mengenai memformat kode akan menangani hal ini dengan otomatis.

```
1. """kalkulator"""
2.
3. class Kalkulator:
4.     """kalkulator tambah kurang"""
5.     def __init__(self, _i):
6.         self.i = _i
7.
8.     def tambah(self, _i): return self.i + _i
9.
10.    def kurang(self, _i):
11.        return self.i - _i
```

Kemudian diproses kembali dengan pycodestyle dan flake8 tidak memunculkan pesan lagi.

Diproses dengan pylint menghasilkan peringatan tentang lebih dari satu pernyataan (statement) dalam satu baris, serta kedua fungsi belum dilengkapi dengan dokumentasi. Hasilnya pylint memberikan skor 5.71 dari 10.

[< Sebelumnya](#)

[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](#)

[Hubungi Kami](#)

[FAQ](#)

Penghargaan



