



Memformat Kode

Jika proses lint atau linting hanya melakukan pengecekan, terkait arahan gaya penulisan kode Anda bisa menggunakan aplikasi tambahan untuk memformat kode agar sesuai PEP8. Sehingga untuk Anda yang menggunakan editor kode yang sangat ringan dan ringkas fitur sekalipun, misalnya tanpa fasilitas lint, tetap bisa menghasilkan kode sesuai konvensi PEP8.

Proses memformat kode dengan menggunakan aplikasi, cukup sederhana, setelah melakukan instalasi, akan muncul perintah dengan nama yang sama dengan nama aplikasinya. Kemudian perintah itu dapat dieksekusi untuk memformat satu file ataupun satu direktori kode Python.

Berikut akan dibahas 3 jenis aplikasi untuk memformat kode, silakan dicermati dahulu, tidak harus semuanya diinstal, hanya paket yang menurut Anda sesuai kebutuhan saja yang digunakan.

Black

[Black](#) adalah proyek *open source* yang dikembangkan di repository Python Software Foundation (PSF) dengan lisensi MIT. Untuk mendapatkan gambaran, versi online (tidak resmi) ada di <https://black.now.sh>.

Untuk instalasi menggunakan manajer paket pip sebagai berikut: *pip install black*.

YAPF (Yet Another Python Formatter)

[YAPF](#) adalah proyek open source yang dikembangkan di repository Google, dengan lisensi Apache.

Untuk mendapatkan gambaran, versi online (tidak resmi) ada di <https://yapf.now.sh>

Untuk instalasi menggunakan manajer paket pip sebagai berikut: *pip install yapf*.

Autopep8

[Autopep8](#) adalah proyek open source (berlisensi MIT) yang termasuk paling awal untuk memformat kode, dengan bantuan lint pycodestyle. Untuk instalasi menggunakan manajer paket pip sebagai berikut: *pip install autopep8*.

Kita akan menggunakan contoh kode sebelumnya kal.py yang berisi class Kalkulator, versi awal sebelum diperbaiki dan dicek dengan linter, sebagai berikut.



DIBANTU



```
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
```

Berikut adalah hasil kode diproses dengan black, dengan mengeksekusi perintah: `black kal.py`

```
1. class Kalkulator:
2.     """kalkulator tambah kurang"""
3.
4.     def __init__(self, _i=0):
5.         self.i = _i
6.
7.     def tambah(self, _i):
8.         return self.i + _i
9.
10.    def kurang(self, _i):
11.        return self.i - _i
```

Kemudian jika diproses dengan yapf, tidak langsung mengubah isi file kal.py tetapi tampil ke layar, dengan mengeksekusi perintah: `yapf kal.py`.

```
1. class Kalkulator:
2.     """kalkulator tambah kurang"""
3.     def __init__(self, _i=0):
4.         self.i = _i
5.
6.     def tambah(self, _i):
7.         return self.i + _i
8.
9.     def kurang(self, _i):
10.        return self.i - _i
```

Sama dengan yapf, untuk pemrosesan dengan autopep8, tidak langsung mengubah isi file kal.py tetapi tampil ke layar, dengan mengeksekusi perintah: `autopep8 kal.py`.

**DIBANTU**



```
2.     """kalkulator tambah kurang"""
3.
4.     def __init__(self, _i=0):
5.
6.         self.i = _i
7.
8.     def tambah(self, _i): return self.i + _i
9.
10.    def kurang(self, _i):
11.
12.        return self.i - _i
```

Setelah mempelajari kedua mekanisme: pengecekan gaya penulisan (style guide) dan proses memformat kode, Anda tinggal fokus dengan penulisan indentasi dalam kode Python, untuk format sisanya dapat dibantu oleh aplikasi-aplikasi yang telah kita pelajari di atas.

Jika Anda menulis dengan editor kode yang sangat sederhana, anggap saja seperti notepad di Windows atau pico/nano di Linux, maka dalam menuliskan kode Python cukup perhatikan indentasi untuk setiap baris pernyataan (statement). Setelah selesai menuliskan kodenya, simpan kodenya sebagai file .py, lalu eksekusi perintah linter atau langsung eksekusi perintah aplikasi untuk memformat kode. Hasilnya, kode Anda sudah dirapikan sesuai arahan gaya penulisan PEP8 juga dilakukan pengecekan terhadap kemungkinan kesalahan (jika Anda lakukan linting).

Untuk Anda yang mengikuti instalasi editor kode PyCharm pada modul Persiapan, secara bawaan sudah menggunakan fitur inspeksi dengan kemampuan yang kurang lebih sama, meskipun jika Anda mau, bisa juga menambahkan aplikasi lint yang sudah dijelaskan sebagai tambahan dari yang bawaan. Demikian juga untuk fitur format ulang kode juga tersedia secara bawaan di aplikasi PyCharm.

[< Sebelumnya](#)

[Selanjutnya >](#)



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



Penghargaan



Decode Ideas
Discover Potential

[Tentang Kami](#)

[Blog](#)

[Reward](#)

[Showcase](#)

[Hubungi Kami](#)

[FAQ](#)





[terms](#) • [Privacy](#)



DIBANTU