



Pengenalan Python

Python adalah bahasa pemrograman multifungsi yang dibuat oleh Guido van Rossum dan dirilis pada tahun 1991. GvR, begitu ia biasa disebut di komunitas Python, menciptakan Python untuk menjadi interpreter yang memiliki kemampuan penanganan kesalahan (*exception handling*) dan mengutamakan sintaksis yang mudah dibaca serta dimengerti (*readability*). Didesain untuk memudahkan dalam prototyping, Python menjadi bahasa yang sangat mudah dipahami dan fleksibel.

Python juga memilih untuk menggunakan indentasi untuk mengelompokkan blok kode, berbeda dengan beberapa bahasa lain yang menggunakan simbol tertentu, misalnya kurung kurawal, atau sintaksis begin-end. Sehingga secara visual pun, blok kode Python didesain untuk mudah dipahami. Salah satu yang paling dikenal adalah, penggunaan titik koma atau semicolon (;) tidak wajib di Python dan penggunaan semicolon cenderung dianggap bukan cara khas Python (*non-pythonic way*), meskipun ia tetap dapat digunakan, misalnya untuk memisahkan dua statement dalam baris yang sama.

```
1. print("Hello World"); print("Welcome to Python")
```

Python juga memilih untuk mengadopsi *dynamic typing* secara opsional, yakni variabel yang dibuat tidak akan diketahui tipenya hingga ia dipanggil pertama kali atau dieksekusi, tidak perlu deklarasi variabel (meskipun dimungkinkan), dan memungkinkan tipe data berubah dalam proses eksekusi program. Sejak Python versi 3.6, sudah tersedia pilihan untuk *static typing*.

Python pun terus berkembang dalam penggunaannya, sehingga fitur-fitur baru dibutuhkan untuk dikembangkan. Versi 2.0 dirilis Oktober 2000 dengan beberapa pengembangan fitur termasuk *Garbage Collector* dan *Memory Management* yang juga menjadi fitur pada beberapa bahasa pemrograman modern lainnya, di antaranya Java dan C#.

Python 3.0 adalah versi perubahan mayor yang dirilis pada Desember 2008, yang didesain sebagai versi yang tidak *backward-compatible* dengan versi-versi sebelumnya. Beberapa sintaksis/statement yang sebelumnya berjalan di versi 2.x, kini tidak lagi berjalan. Semua hal ini didasarkan pada keinginan bahasa Python yang kembali ke “inti”, yakni *readable, consistent & explicit*. Contohnya, fungsi print yang sebelumnya adalah statement di python 2.x, menjadi function di python 3.x.

Lebih jauh tentang Python 3.10 kunjungi tautan berikut:

<https://docs.python.org/3.10/whatsnew/3.10.html>.

Versi terbaru Python pada saat pembaruan modul ini adalah 3.10.1 yang dirilis pada 6 Januari 2022. Pada saat tulisan ini dibuat, Python 3.11 sedang dikembangkan.



DIBANTU



Saat ini, Python dikelola oleh lembaga non-komersial Python Software Foundation (PSF). Namun sebelumnya, GvR dijuluki sebagai Benevolent dictator for life (BDFL) karena hampir semua keputusan pengembangan Python diambil oleh GvR, berbeda dengan bahasa lain yang misalnya menggunakan voting dan semacamnya. Pasca tahun 2000, dibentuklah beberapa sistem yang memungkinkan Python menjadi lebih sustain, misalnya Python Enhancement Proposals (PEP) untuk pengembangan Python dan tentunya Python Software Foundation (PSF).

Jika PSF menjadi lembaga yang mengelola dan mengadvokasi Python, PEP menjadi panduan dalam pengembangan Python. Beberapa PEP memuat misalnya bagaimana sintaksis dan bagaimana Bahasa Python akan berevolusi, bagaimana modul akan dinyatakan usang (*deprecated*), dan sebagainya. Setelah kurang lebih 30 tahun dalam pengembangan Python, GvR memutuskan untuk tidak lagi menjabat BDFL pada 12 Juli 2018.

Salah satu patokan dalam pengembangan Python adalah PEP 20 yang berjudul Zen of Python.

Zen of Python (<https://www.python.org/dev/peps/pep-0020/>)

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!

Jika ada pengembangan fitur Python, maka PEP 20 inilah yang menjadi dasar/akar dalam mengambil keputusan.

[< Sebelumnya](#)

[Selanjutnya >](#)



DIBANTU



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



Discover Potential

[Tentang Kami](#)

[Reward](#)

[Showcase](#)

[FAQ](#)

Penghargaan



© 2022 Dicoding | Dicoding adalah merek milik PT Presentologics, perusahaan induk dari PT Dicoding Akademi Indonesia.

[Terms](#) • [Privacy](#)



DIBANTU