

Jadi, Apa Kegunaan dari kode `if __name__ == "__main__"` pada Python?

Mungkin kita familiar atau setidaknya pernah melihat kode `if __name__ == "__main__"` pada salah satu skrip dalam bahasa pemrograman Python (py).

Namun sebenarnya apa kegunaan dari kode tersebut?

Biasanya kode tersebut sering kali ditemukan pada akhir dari skrip program. Apakah itu sebuah keharusan dalam menempatkan kode ini di akhir skrip? Ternyata tidak. Kode `if __name__ == "__main__"` tidak selalu harus berada di akhir skrip, tetapi bisa ditempatkan di mana pun, selama kode yang ingin dijalankan saat skrip tersebut dieksekusi secara langsung di dalam suatu blok, dan tidak akan mempengaruhi bagian lain dari kode. Namun, umumnya diletakkan di bagian bawah untuk menjaga struktur kode tetap jelas dan mudah dipahami.

Kode ini digunakan untuk melakukan blok atau istilahnya adalah “pengaman” dalam skrip program dimaksud. Mengapa bisa diistilahkan sebagai “pengaman”? mari kita pahami secara perlahan.

Dengan adanya kode ini, pertama kita dapat memastikan bahwa kode tersebut hanya dijalankan jika file tersebut dieksekusi secara langsung, bukan saat diimpor sebagai suatu modul di dalam skrip lain. Ini mencegah kode yang tidak perlu dijalankan secara otomatis ketika modul diimpor. Artinya dengan kode ini, **eksekusi kode menjadi terkontrol**.

Lebih detailnya mari kita lihat contoh berikut:

```
trial.py > ...
1  def greet(name):
2      return f"Hello, {name}!"
3
4  def add(a, b):
5      return a + b
6
7  if __name__ == "__main__":
8      print(greet("friends"))
9      print(add(1,2))

Hello, friends!
3
```

Gambar 1 (Contoh skrip dan output penggunaan kode)

Pada Gambar 1 merupakan suatu skrip berjudul *trial.py*. Penggunaan kode `if __name__ == "__main__"` digunakan untuk memanggil *function* pada serangkaian blok kode yang ada pada susunan skrip tersebut dan bisa dijalankan secara langsung. Misalnya, fungsi *greet* akan memunculkan *string* (baris karakter) *greeting* yaitu “Hello, (nama – dapat disesuaikan sesuai input)!”. Dan fungsi *add* merupakan penjumlahan dari *a+b*.

```

import trial.py
1  import trial
2  __name__ = "trial"
3  print(trial.greet('Melisa'))

```

Hello, Melisa!

Gambar 2 (Contoh penggunaan kode setelah di-*import* ke skrip lain dan *output*-nya)

Pada Gambar 2 merupakan skrip lain di mana melakukan *import* dari skrip berjudul *trial.py*. karena sudah melakukan *import* skrip tersebut, maka skrip baru yang berjudul “import trial.py” dapat memanggil fungsi-fungsi yang terdapat pada skrip *trial.py* sesuai dengan keinginan tanpa langsung menjalankan fungsi-fungsi (misalnya *greet* dan *add*) jika tidak terdapat perintah sebelumnya pada skrip *import trial.py*. Dalam gambar 2 ini, fungsi *add* tidak ditulis atau dipanggil sehingga tidak dijalankan.

Contoh lainnya mari kita lihat apabila tidak menggunakan kode `if __name__ == "__main__"`.

```

trial.py > ...
1  def greet(name):
2      return f"Hello, {name}!"
3
4  def add(a, b):
5      return a + b
6
7  print (greet('friends'))
8  print (add(1,2))
9

```

Hello, friends!

3

Gambar 3 (Contoh skrip yang tidak menggunakan kode `if __name__ == "__main__"` serta *output*-nya)

Pada Gambar 3, skrip *trial.py* ini tetap berjalan langsung seperti skrip pada Gambar 1.

```

import trial.py
1  import trial
2  __name__ = "trial"
3  print(trial.greet('Melisa'))
4

```

Hello, friends!

3

Hello, Melisa!

Gambar 4 (Contoh penggunaan tanpa kode `if __name__ == "__main__"` setelah di-*import* ke skrip lain berjudul *import trial.py* serta *output*-nya)

Pada Gambar 4, skrip *import trial.py* berjalan tidak sesuai perintah, dan menampilkan seluruh fungsi yang terdapat pada skrip *trial.py*.

Kedua, kode `if __name__ == "__main__"` penting untuk mengorganisir kode dengan baik atau istilahnya **lebih modular**. Setelah kita melihat contoh perbandingan pada Gambar 2 dan Gambar 4,

maka lebih jelas lagi bahwa `if __name__ == "__main__"` bisa mendefinisikan fungsi dalam skrip, lalu hanya menjalankan bagian tertentu saja saat file dijalankan secara langsung.

Ketiga, dengan menggunakan kode blok ini, orang lain akan lebih mudah membaca kode yang kita buat dan lebih mudah memahami bagaimana program bekerja dan cara menjalankannya. Dapat membantu kolaborasi antara *programmer*, pemeliharaan kode di masa yang akan datang, dan pengembangan yang lebih baik secara keseluruhan.

Selamat mencoba!

Referensi:

Geeksforgeeks. (2023). *What Does if __name__ == "__main__": ?*. Diakses pada 19 September 2024, dari <https://dev.notnoob.com/python/fungsi-if-main-name-python/>

Gruppetta, Stephen. (2024). *if __name__ == "__main__" Python: Complete Explanation*. Diakses pada 19 September 2024, dari <https://www.datacamp.com/tutorial/if-name-equals-main-python>

noTnoob dev. (2023). *Apa Itu Fungsi if __name__ == "__main__" Pada Python?*. Diakses pada 19 September 2024, dari <https://dev.notnoob.com/python/fungsi-if-main-name-python/>