**Perbedaan \ dengan / pada sebuah file**

/ adalah pemisah jalur pada sistem Unix dan Unix-like. Windows modern umumnya dapat menggunakan keduanya \ dan / interchangeably untuk filepaths, tetapi Microsoft telah menganjurkan untuk penggunaan \ sebagai pemisah jalan selama beberapa dekade.

Hal ini dilakukan untuk alasan historis yang tanggal sejauh 1970-an, mendahului Windows lebih dari satu dekade. Pada awalnya, MS-DOS (dasar untuk Windows awal) tidak mendukung direktori. Unix memiliki dukungan direktori menggunakan / karakter sejak awal. Namun, ketika direktori ditambahkan di MS-DOS 2.0, Microsoft dan IBM sudah menggunakan / karakter untuk [switch perintah](https://en.wikipedia.org/wiki/Command-line_interface#Command-line_option), dan karena parser ringan DOS (diturunkan dari [QDOS](https://en.wikipedia.org/wiki/86-DOS), dirancang untuk dijalankan pada perangkat keras kelas bawah), mereka tidak dapat menemukan cara yang layak untuk menggunakan / karakter tanpa merusak kompatibilitas dengan aplikasi yang ada.

Jadi, untuk menghindari kesalahan tentang "hilang saklar" atau "beralih tidak sah" ketika melewati filepath sebagai argumen ke perintah seperti ini:

cd/ <---- no switch specified

dir folder1/folder2 <---- /folder2 is not a switch for dir

diputuskan bahwa \ karakter akan digunakan sebagai gantinya, sehingga Anda bisa menulis perintah-perintah seperti ini

cd\

dir folder1\folder2

tanpa error.

Kemudian, Microsoft dan IBM berkolaborasi pada sistem operasi yang tidak terkait dengan DOS yang disebut [OS / 2](https://en.wikipedia.org/wiki/OS/2). OS / 2 memiliki kemampuan untuk menggunakan kedua pemisah, mungkin untuk menarik lebih banyak pengembang Unix. Kapan [Microsoft dan IBM berpisah pada tahun 1990](https://en.wikipedia.org/wiki/OS/2#1990:_Breakup), Microsoft mengambil kode apa yang mereka miliki dan buat [Windows NT](https://en.wikipedia.org/wiki/Windows_NT), di mana semua versi modern Windows didasarkan, membawa agnostisisme pemisah ini dengannya.

Karena kompatibilitas mundur adalah nama permainan untuk Microsoft dari semua transisi OS utama yang telah mereka lakukan (DOS ke Win16 / DOS, ke Win16 / Win32, ke Win32 / WinNT), kekhasan ini macet, dan mungkin ada untuk sementara waktu.

Karena alasan inilah perbedaan ini ada. Seharusnya tidak berpengaruh pada apa yang Anda lakukan karena, seperti yang saya katakan, WinAPI umumnya dapat menggunakannya secara bergantian. Namun, aplikasi pihak ke-3 mungkin akan rusak jika Anda lulus / ketika mereka mengharapkan suatu \ antara nama direktori. Jika Anda menggunakan Windows, tetap dengan \. Jika Anda menggunakan Unix atau [URI](https://en.wikipedia.org/wiki/Uniform_Resource_Identifier)s (yang memiliki landasan mereka di jalur Unix, tapi itu cerita lain sepenuhnya), kemudian gunakan /.

**Dalam konteks C #:** Perlu dicatat, karena ini *aku s* secara teknis pertanyaan C #, bahwa jika Anda ingin menulis lebih banyak kode C "portabel" yang bekerja pada Unix dan Windows (meskipun C # adalah bahasa Windows), Anda mungkin ingin menggunakan Path.DirectorySeparatorChar lapangan sehingga kode Anda menggunakan pemisah yang disukai pada sistem itu, dan digunakan Path.Combine()untuk menambahkan jalur dengan benar.