Menggunakan GIT

Sebagai seorang pengembang perangkat lunak yang berkutat pada programming, tentu kamu punya pengalaman pahit saat salah menghapus baris pada skrip, atau malas melakukan penambahan fitur karena takut akan merusak seluruh aplikasi? Dan pada semua kejadian tersebut, kamu tidak punya backup, atau cukup malas untuk menulis ulang alur pemrograman. Pernahkan memakai sistem kontrol revisi?

Pada tutorial kali ini, saya akan membahas tentang Git yang akan menyelamatkan kamu dari bencana kerusakan aplikasi akibat salah mengubah baris-baris pemrograman.

Tentang Tutorial ini:

* Tingkat Kesulitan: Pemula
* Durasi: 30 menit
* Materi: Git

**Apa itu Git?**

[Git](http://git-scm.com/) adalah sistem kontrol revisi terdistribusi (terjemahan bebas dari Distributed Revision Control) [diciptakan Linus Torvalds](http://en.wikipedia.org/wiki/Git_%28software%29#Early_history) untuk mendukung pengembangan kernel Linux pada tahun 2005, dengan kelebihan pada performa dan skalabilitas besar.

**Kenapa harus Git?**

Saya pernah menggunakan Subversion (SVN) saat awal pertama kali belajar pemrograman, saat itu saya menyadari begitu pentingnya mengetahui perubahan apa saja yang saya lakukan di masa lalu, agar saya bisa *move-on* melanjutkan alur aplikasi yang saya buat. Saya memilih SVN karena masih memakai Windows dan Git belum mendukung dengan baik. Namun, di kemudian hari saya menyadari ketika SVN sangat tidak bagus untuk kolaborasi dan membuat directory aplikasi saya membengkak berkali lipat.

Saat ini Git sudah mendukung hampir semua sistem operasi (Mac OSX, Linux, Windows, dll), performa tinggi, dan mudah untuk berkolaborasi, dan lebih sedikit memakan kuota penyimpanan.

**Bagaimana memasang Git pada sistem operasi kamu?**

- Windows: unduh dan pasang [mysgit](http://code.google.com/p/msysgit/downloads/list).

- Linux: gunakan package manager dari masing-masing distro.

- Mac OSX: unduh dan pasang [git-osx-installer](http://code.google.com/p/git-osx-installer/downloads/list).