

# Harus Lebih Bertanggung Jawab



Dunia *software* masih terus berubah dari kelas coba-coba ke kelas profesional. Dia ingin disetarakan dengan bidang rekayasa (*engineering*) lainnya. Munculnya bidang *software engineering*, menyetarakan bidang *software* ini dengan bidang teknik lainnya seperti teknik elektro (*electrical engineering*), teknik mesin (*mechanical engineering*), teknik sipil (*civil engineering*), dan lain-lain. Namun sayangnya, perilaku dan kultur para insinyur *software* ini masih belum profesional.

Salah satu contoh yang unik adalah adanya "*disclaimer*" pada produk *software*. Disclaimer ini berisi pernyataan bahwa sang pembuat *software* tidak bertanggung jawab dan tidak dapat dituntut apabila terjadi kesalahan atau ada akibat negatif yang ditimbulkan oleh penggunaan *software* tersebut. (Sebagai latihan, coba cari disclaimer ini pada *software* atau *web site* gratisan yang Anda gunakan.)

Mari kita bandingkan dengan bidang teknik lainnya. Apakah Anda pernah mendengar *disclaimer* dari produk jembatan yang dibangun oleh seorang insinyur sipil, "bahwa kami tidak bertanggung-jawab apabila jembatan ini runtuh, ...dan seterusnya"? Mana ada orang yang mau menyeberang dengan menggunakan jembatan tersebut jika ada disclaimer seperti itu. Ngeri... Hal yang sama juga terjadi di dunia kedokteran. Mereka semua profesional. Herannya, di dunia teknologi informasi orang dengan mudahnya menerima *disclaimer software*.

Mungkin hal ini terjadi karena bidang ilmu yang terkait dengan komputer masih bisa dikatakan bidang baru sehingga harus dimaafkan jika terjadi kesalahan? Banyak yang masih menggunakan metoda coba-coba dalam pengembangan *software*. Tahukah Anda tentang *software development life cycle*? Teori yang diajarkan di kelas itu bukan hanya untuk gagah-gagahan, tapi ada manfaatnya. Sayangnya, memang banyak dosen yang kurang berhasil menjelaskan ini kepada mahasiswanya. Atau bisa jadi penulis buku (artikel) yang kurang bisa menjelaskan hal tersebut kepada pembacanya.

Sayangnya, ketidakprofesionalan ini menjalar ke bidang lain yang masih dekat. Dalam bidang jaringan komputer, misalnya,

seorang *network engineer* dengan seenaknya memasang jaringan dan perangkat pendukungnya, seperti *firewall* dan *intrusion detection system* (IDS). Ketika terjadi kesalahan, berhasil dibobolnya sistem yang dia kelola, maka yang keluar adalah pernyataan maaf atau mengeluarkan dalih bahwa itu hanya terjadi di mesin percobaan saja. Kelalaian (*negligence*) tidak bisa hilang dengan kata maaf saja. Harus ada pertanggung-jawaban.

Di dunia teknik yang lain, ada pemisahan tanggung jawab (*separation of duties*). Misalnya ketika kita membangun sebuah jembatan, maka ada konsultan desain (yang menerjemahkan keinginan pemilik menjadi dokumen teknik yang dapat diimplementasikan), ada kontraktor (yang mewujudkan desain menjadi kenyataan dengan menggunakan dokumen teknik yang dibuat oleh desainer), dan ada pengawas (yang mengawasi atau

mengaudit pekerjaan kontraktor). Tidak bisa semuanya dijadikan satu.

Dalam bidang pengembangan *software*, jaringan, dan sistem komputer, saat ini biasanya campur aduk. Semuanya jadi satu, ya desainer dari sistem, ya implementor, ya *tester*

(*quality assurance*), ya auditor teknologi. Akibatnya tidak ada mekanisme kendali dan pengawasan yang baik. Kaidah-kaidah *engineering* dilanggar. Ini harus berubah jika kita ingin lebih baik dan lebih bertanggung jawab.

Saya pernah bertanya kepada mahasiswa saya. Jika Anda membuat sebuah *software* yang digunakan untuk mengendalikan alat pacu jantung, apakah Anda akan menggunakan alat dan *software* tersebut untuk diri Anda sendiri? Sebagian besar (hampir semua) tersenyum kecut dan menyatakan tidak berani. Lebih baik pakai produk buatan perusahaan asing saja. Padahal produk asing belum tentu lebih bagus. Jika sang programer saja tidak berani menggunakan produknya, mengapa orang lain harus menggunakan produk tersebut? Bagaimana dengan Anda? Beranikah Anda menggunakan produk yang dikendalikan oleh *software* buatan Anda sendiri? 🐱

**"Ketika terjadi kesalahan, berhasil dibobolnya sistem yang dia kelola, maka yang keluar adalah pernyataan maaf..."**