

Bagaimana Mahasiswa Ilmu Komputer Belajar: Mengkritisi Kurikulum dan Gaya Pendidikan Kita

Romi Satria Wahono

Lisensi Dokumen:

Copyright © 2003-2006 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Sepulang dari study di Jepang tahun 2004, saya banyak mengajar di beberapa Universitas di Jakarta, terutama di fakultas atau jurusan yang berhubungan dengan ilmu komputer dan teknik informatika. Saya mengajar mata kuliah yang memang saya kuasai, dan terkait langsung dengan tema penelitian saya. Diantaranya adalah mata kuliah Software Engineering (Rekayasa Perangkat Lunak), Algoritma dan Bahasa Pemrograman (Algorithm and Programming Language), dan Basis Data (Database). Kebanyakan mata kuliah tersebut diajarkan setelah semester 5 (tingkat 3 atau 4). Dalam interaksi belajar mengajar di kelas, saya menemukan beberapa fenomena menarik berhubungan pengetahuan mahasiswa dan kurikulum yang diajarkan di universitas.

Saya menemukan tipe mahasiswa yang ketika saya terangkan dia kesulitan menangkap beberapa konsep yang seharusnya sudah dia dapat di semester sebelumnya. Katanya, itu tidak diajarkan di universitas tersebut. Fenomena ini terjadi dalam universitas yang memotong (mengubah) beberapa kurikulum yang seharusnya diajarkan, karena tidak ada SDM pengajar (dosen). Di lain pihak, saya menemukan fenomena lain dimana mahasiswa mengatakan bahwa dia mengenal beberapa konsep yang saya singgung, hanya dia lupa mata kuliah yang mengajarkannya. Fenomena ini terjadi di universitas yang mencekoki mahasiswanya dengan mata kuliah berlebih, dengan argumentasi bahwa supaya mahasiswa mendapat pengetahuan secara lengkap. Sering dosen mengajar bukan pada bidang yang dikuasai, hal itu terpaksa dilakukan oleh universitas untuk mengejar mata kuliah yang harus jalan. Dua-duanya ternyata membuat mahasiswa jadi linglung, yang satu linglung karena memang tidak pernah diajarkan, dan yang lain linglung karena terlalu banyak yang diajarkan. Intinya sih kedua-duanya sama-sama nggak ngerti.

Fenomena aneh lain tentunya masih banyak, misalnya mahasiswa tingkat 3 jurusan teknik informatika (atau ilmu komputer) yang tidak kenal siapa Dennis Ritchie, tidak bisa membuat program meskipun hanya untuk sebuah fungsi untuk memunculkan Hello World (apalagi mengkompilanya), tidak paham tentang paradigma pemrograman, juga

tidak paham apa itu kompiler, shell, pointer, fungsi, array, dan tentu semakin mual-mual kalau saya sebut algoritma atau struktur data.

Bagaimana seorang mahasiswa Ilmu Komputer belajar? Saya mencoba memberi gambaran umum dengan mengambil studi kasus bagaimana jurusan ilmu komputer di Saitama University mengatur kurikulumnya. Saitama University bukan termasuk universitas yang terbaik untuk ilmu komputer, umurnya masih sangat muda dengan SDM pengajar (professor) yang juga terbatas, bahkan beberapa professor diambil dari jurusan elektro untuk beberapa mata kuliah tertentu. Ini tidak mengurangi keseriusan universitas untuk menyajikan pendidikan dan kurikulum terbaik untuk mahasiswa-mahasiswanya.

Saya mulai program undergraduate (S1) di Department of Information and Computer Sciences, Saitama University tahun 1995. Tingkat I (semester 1 dan 2), mata kuliah dasar (**kiso kamoku**) sangat dominan. Kalkulus, statistik, probabilitas, fisika dasar, kimia dasar, discrete mathematics, dan mata kuliah dasar lain banyak diajarkan. Semester 2 sudah ada beberapa mata kuliah jurusan (**senmon kamoku**) yang diajarkan, diantaranya adalah bahasa pemrograman, bahasa C (prosedural), HTML, dengan praktek lab untuk mengenal Unix, shell, text editor (emacs), LaTeX (TeX), gnuplot, kompiler, teknik typing 10 jari, dsb. Pada saat masuk tingkat II (semester 3), saya menyadari bahwa mata kuliah tingkat I membekali saya dengan beberapa tool dan konsep dasar, sehingga saya bisa survive mengikuti proses belajar mengajar di tingkat selanjutnya. Lab komputer hanya berisi Unix terminal. Seluruh laporan dan tugas harus ditulis dengan LaTeX dengan text editor emacs, apabila memerlukan bahasa pemrograman harus dibuat dalam bahasa C dan dikompilasi dengan GCC. Apabila ada data yang harus ditampilkan dalam bentuk grafik, bisa menggunakan Gnuplot. Setiap mahasiswa harus mempunyai situs web (homepage), dimana selain berisi aktifitas pribadi, juga berisi seluruh laporan dan tugas yang dikerjakan. Selain lewat situs web, laporan harus dikirim dengan menggunakan email ke professor pengajar, dalam format PS atau PDF dengan source dari LaTeX.

Yang menarik, bahwa gaya pendidikan yang ditempuh menganut konsep korelasi, berhubungan, saling mendukung dan terarah dari semester 1 sampai akhir. Skill terhadap komputer dan bahasa pemrograman juga cukup dalam, karena ada kewajiban menguasai bahasa C, HTML, Unix, Linux, Shell, dsb yang bukan untuk ritualitas mata kuliah semata, tapi untuk bekal sang mahasiswa supaya bisa survive di jenjang semester berikutnya. Apakah tidak diajarkan paradigma dan bahasa pemrograman lain? jawabannya adalah diajarkan, tetapi untuk konsumsi mahasiswa tingkat 3 (semester 5 dan 6). Pemrograman berorientasi objek (Java), functional programming (LISP dan Scheme), dan Prolog diajarkan pada semester 5 dan 6 untuk membidik supaya sang murid “nyantol” ketika mengikuti mata kuliah Kecerdasan Buatan (Artificial Intelligence) dan Rekayasa Perangkat Lunak (Software Engineering). Dan dengan sebelumnya menguasai bahasa prosedural seperti C, kita semakin “ngeh” tentang pentingnya paradigma berorientasi objek ketika mendalami mata kuliah tentang pemrograman berorientasi objek.

Korelasi mata kuliah ini nampak juga dari deretan gaya pengajaran, setelah mahir berbahasa C, kita diminta ngoprek Minix yang terbuat dari bahasa C (sistem operasi buatan Andrew S. Tanenbaum, yang menginspirasi Linus Torvald membuat Linux) pada

mata kuliah Operating System (Sistem Operasi), membuat sendiri shell (dengan fungsi yang mendekati bash dan cshell) diatas sistem operasi yang sudah kita oprek, dan diminta mendesain dan mengembangkan bahasa pemrograman sendiri di mata kuliah Compiler Engineering (teknik kompilasi). **Berurutan, berhubungan, tetap fokus dan mendalam**, itu mungkin resep desain kurikulum yang diajarkan.

Pada saat tingkat 2 dan 3 itulah sang mahasiswa diarahkan untuk menuju arah kompetensi sesuai dengan yang diinginkan. Dan yang pasti, hampir seluruh mahasiswa mendapatkan “bekal” dan “skill” yang relatif sepadan untuk bergerak. Mahasiswa yang ingin melanjutkan karier menjadi seorang **Programmer**, disiapkan mata kuliah Struktur Data, Algorithm, Programming Language, Compiler Engineering, Automaton dan Formal Language. Yang ingin jadi **Software Engineer**, harus fokus mengikuti mata kuliah Software Engineering, Industrial Software Engineering, System Development Engineering, Software Project Management, dsb. Yang ingin berkarier di perusahaan **animasi dan grafis**, harus serius mengikuti mata kuliah Computer Graphics, Image Processing, CAD Engineering, Pattern Recognition, dsb. Yang siap bergelut di perusahaan **Telekomunikasi**, harus melahap mata kuliah Information Theory, Communication System, Signal Processing, Speech Processing, dsb. Yang ingin ke arah **Hardware**, harus menguasai mata kuliah Electronic Circuits, Electronic Devices, Computer Architecture, Quantum Mechanics, Logic Circuits, dsb. Bagaimana dengan yang tertarik dengan **Kecerdasan Buatan**? harus mau berpusing-pusing ria di mata kuliah Artificial Intelligence, Expert System, Knowledge Engineering, Neural Network, dsb.

Rencana pengembangan karier ini semakin matang dan tertata ketika masuk ke tingkat 4, seluruh mahasiswa harus menjalani 1 tahun terakhir di grup penelitian yang dipimpin oleh seorang professor. Penelitian dan thesis (tugas akhir) sifatnya wajib dilakukan, untuk memperdalam dan memahami implementasi riil dari bidang ilmu peminatan yang direncanakan dan dicita-citakan sang mahasiswa. Apa itu bidang ilmu peminatan? Ya bidang yang sudah saya sebut diatas tadi. Programming, Software Engineering, Communication System, Computer Graphics, Artificial Intelligence, Computer Hardware, Networking, dsb. Masing-masing professor dengan grup penelitian biasanya fokus di satu atau dua bidang ilmu peminatan, termasuk didalamnya penelitian yang dilakukan dan mata kuliah yang diajar. Tidak ada seorang professor Software Engineering yang mendapat jatah mengajar mata kuliah Computer Graphics, karena memang bukan bidangnya. Kalaupun bisa memberikan, tentu tidak menguasai **the root problem** (akar permasalahan) yang ada di bidang tersebut, ini yang membuat mata kuliah jadi hambar, tidak mendalam dan mahasiswa jadi bingung memahami apa hakekat dari mata kuliah tersebut.

Jadi masing-masing mata kuliah ada arah, ada desain yang ingin dicapai, dan ini yang dijelaskan di awal perkuliahan. Tidak ada kegiatan OSPEK yang berisi penyiksaan dan penghinaan, tidak ada hura-hura pesta masuk perguruan tinggi, yang ada adalah penjelasan tentang kurikulum secara komprehensif. Sang mahasiswa ingin menjadi apa, tertarik di bidang apa, itu yang dibidik dan diarahkan oleh universitas dengan penjelasan desain kurikulum beserta dengan mata kuliah apa yang sebaiknya diambil oleh sang mahasiswa. Jumlah kredit untuk syarat kelulusan S1 juga tidak sepadat Indonesia, hanya

sekitar 118, sudah termasuk didalamnya penelitian dan tugas akhir yang dihitung sekitar 10-12 kredit. Jadi total kredit dari mata kuliah hanya sekitar 106. Kelonggaran waktu yang ada dapat kita gunakan untuk kerja parttime di perusahaan-perusahaan IT, mengasah kemampuan jadi programmer, network engineer, admin, software designer, dsb. Mahasiswa mendapatkan konsep di kelas, dan mematangkan diri di lapangan, tempat kita menggarap project maupun tempat kerja. Itu adalah strategi penting dalam mengkader para computer scientist.

Universitas di Indonesia yang membuka fakultas/jurusan Ilmu Komputer dan Teknik Informatika harus berbenah. Tidak hanya berambisi mengejar jumlah murid karena konsep aji mumpung (mumpung TI sedang booming, terima mahasiswa sebanyak banyaknya), tapi juga harus bertanggungjawab terhadap figur dan karakter hasil didikan dan lulusan universitasnya. Untuk para calon mahasiswa, pilihlah Universitas yang memiliki kurikulum dan dosen pengajar yang baik. Jangan memilih jurusan karena trend, ikut-ikutan teman, atau alasan tidak logis lainnya. Pilihlah karena memang kita berminat untuk berkarier di bidang tersebut.

Biografi Penulis



Romi Satria Wahono. Menamatkan SMU di SMU Taruna Nusantara, Magelang pada tahun 1993. Menyelesaikan program S1 dan S2 di *Department of Computer Sciences, Saitama University, Jepang* tahun 1999 dan 2001. Saat ini program program S3 pada jurusan yang sama. Peneliti di Lembaga Ilmu Pengetahuan Indonesia (LIPI). Kompetensi inti pada bidang *Software Engineering, eLearning System, dan Knowledge Management*. Aktif sebagai penulis, dimana ratusan tulisan berupa scientific paper, artikel, dan tutorial telah diterbitkan dalam berbagai proceedings conference, jurnal ilmiah, majalah, koran dan portal, bertaraf nasional maupun internasional. Mendapatkan penghargaan dari PBB pada pertemuan puncak WSIS (*World Summit on Information Society*) tahun 2003 di jenewa, sebagai pendiri dari IlmuKomputer.Com. CEO dari PT Brainmatics Cipta Informatika, perusahaan IT yang bergerak di bidang training dan consulting.

Informasi lebih lanjut tentang penulis bisa didapat melalui:

Email: romi@romisatriawahono.net

URL: <http://romisatriawahono.net>

YM: [romi_sw](#)