

ETIKA PROFESI YANG BERLAKU BAGI PROGRAMMER DI INDONESIA

Galih Pranowo / 06071132

**Jurusan Matematika Ilmu Komputer
FAKULTAS SAINS TERAPAN
INSTITUT SAINS & TEKNOLOGI AKPRIND
YOGYAKARTA**

PENDAHULUAN

Bukanlah suatu hal yang berlebihan jika dikatakan bahwa komputer merupakan alat sosial karena kenyataannya bahwa teknologi tersebut dipergunakan secara intensif pada berbagai komunitas masyarakat seperti institusi, organisasi, perusahaan, dan lain sebagainya. Seperti halnya pada alat-alat sosial yang lain, pemanfaatan teknologi komputer dapat secara langsung maupun tidak langsung berpengaruh terhadap tatanan kehidupan masyarakat yang menggunakannya. Selain dibutuhkan moral yang didefinisikan sebagai suatu prinsip perilaku benar dan salah dan hukum, etika memegang peranan yang sangat penting. Kata Etika berasal dari bahasa Yunani *Ethos*, yang berarti karakter. Etika adalah satu set kepercayaan, standar, atau pemikiran yang mengisi suatu individu, kelompok atau masyarakat.

Semua individu bertanggung jawab kepada masyarakat atas perilaku mereka. Masyarakat dapat berupa suatu kota, negara atau profesi. Tindakan kita juga diarahkan oleh etika. Tidak seperti moral, etika dapat sangat berbeda dari satu masyarakat ke masyarakat lain. Karakteristik etika yang lebih spesifik dalam dunia komputer diperkenalkan oleh seorang profesor dari Dartmouth pada tahun 1985. James H. Moor mendefinisikan etika komputer sebagai analisis mengenai sifat dan dampak social teknologi komputer, serta formulasi dan justifikasi kebijakan dalam menggunakan teknologi tersebut secara etis.

Menjadi seorang Programmer atau Jasa Pemrograman sistem harus ditunjang pengalaman yang cukup. Bukan hanya harus mengerti secara detail tentang bagaimana mendvelop suatu sistem, tetapi juga harus bisa menjiwai sistem tersebut.

Etika profesi yang berlaku bagi programmer di indonesia 1

Dalam arti, dalam pengembangan suatu sistem khususnya kepada klien tidak harus mengikuti 100% apa yang diinginkan oleh kliennya, tetapi juga harus dipikirkan kebutuhan masa depan klien dan kemudahan untuk pengembangan berikutnya. Seringkali seorang programmer hanya mengikuti perintah dari kliennya, sampai keinginan klien semua dikerjakan, tetapi lupa tentang bagaimana jika dimasa depan sistem tersebut akan dikembangkan. Kalau sudah salah mendvelop, akhirnya bisa berakibat fatal yakni jika ingin dikembangkan tidak bisa, karena harus merubah seluruh sistem, program dan database yang ada. Seorang programmer harus berupaya meresapi pemikiran para klien dan apa yang mereka harapkan serta mengarahkan mereka melihat sesuatu lebih ke depan sehingga kesadaran klien bisa digugah. Dengan demikian program yang akan dibuat bisa lebih mudah diterima oleh klien atau calon klien.

KELOMPOK BIDANG PROGRAMMER

Profesi seorang programmer dapat dikelompokkan menjadi beberapa bidang sebagai berikut :

1. Programmer

Programmer adalah individu yang bertugas dalam hal rincian implementasi, pengemasan, dan modifikasi algoritma serta struktur data, dituliskan dalam sebuah bahasa pemrograman tertentu.

Deskripsi Pekerjaan :

- a) Menulis program (coding) dengan menggunakan pemrograman tertentu (VB, VB.NET, Java).
- b) Memahami konsep basis data.
- c) Mampu mengoperasikan aplikasi basis data.
- d) Melakukan pengujian terhadap aplikasi program.
- e) Melakukan analisis terhadap aplikasi program.
- f) Melakukan riset , desain, dokumentasi dan modifikasi aplikasi software.
- g) Melakukan analisis dan memperbaiki kerusakan (error) pada software dengan tepat dan cara yang akurat.

h) Menyediakan status laporan aplikasi yang diperlukan.

2. Database Programmer

Programmer yang menguasai perancangan dan pemrograman database menggunakan Aplikasi Ms. Access dan Ms. Sql Server ATAU PL/SQL dan Oracle Form Developer 10g

3. Web Programmer

Programmer web yang menguasai pengembangan aplikasi web berbasis HTML dan ASP.Net ATAU aplikasi web berbasis HTML dan J2EE serta Struts Framework ATAU aplikasi web berbasis MySQL dan PHP.

4. Multimedia Programmer

Programmer multimedia yang menguasai penggunaan teknologi dan pengembangan aplikasi berbasis multimedia

5. Embedded Programmer

Programmer yang menguasai arsitektur sistem mikroprosesor, interfacing dan pemrograman embedded

KODE ETIK PROGRAMMER

Pemrograman komputer membutuhkan sebuah kode etik, dan kebanyakan dari kode-kode etik ini disadur berdasarkan kode etik yang kini digunakan oleh perkumpulan programmer internasional.

Kode etik seorang programmer adalah sebagai berikut :

1. Seorang programmer tidak boleh membuat atau mendistribusikan Malware.
2. Seorang programmer tidak boleh menulis kode yang sulit diikuti dengan sengaja.
3. Seorang programmer tidak boleh menulis dokumentasi yang dengan sengaja untuk membingungkan atau tidak akurat.
4. Seorang programmer tidak boleh menggunakan ulang kode dengan hak cipta kecuali telah membeli atau telah meminta izin.
5. Tidak boleh mencari keuntungan tambahan dari proyek yang didanai oleh pihak kedua tanpa izin.

6. Tidak boleh mencuri software khususnya development tools.
7. Tidak boleh menerima dana tambahan dari berbagai pihak eksternal dalam suatu proyek secara bersamaan kecuali mendapatkan izin.
8. Tidak boleh menulis kode yang dengan sengaja menjatuhkan kode programmer lain untuk mengambil keuntungan dalam menaikkan status.
9. 9. Tidak boleh membeberkan data-data penting karyawan dalam perusahaan.
10. Tidak boleh memberitahu masalah keuangan pada pekerja dalam pengembangan suatu proyek.
11. Tidak pernah mengambil keuntungan dari pekerjaan orang lain.
12. Tidak boleh memermalukan profesinya.
13. Tidak boleh secara asal-asalan menyangkal adanya bug dalam aplikasi.
14. Tidak boleh mengenalkan bug yang ada di dalam software yang nantinya programmer akan mendapatkan keuntungan dalam membetulkan bug.
15. Terus mengikuti pada perkembangan ilmu komputer.

Pada umumnya, programmer harus mematuhi “Golden Rule”: Memperlakukan orang lain sebagaimana kamu ingin diperlakukan. Jika semua programmer mematuhi peraturan ini, maka tidak akan ada masalah dalam komunitas.

FAKTOR YANG MEMPENGARUHI KUALITAS PRODUK DAN PRODUKTIVITAS PROGRAMMER

1. Kemampuan pribadi :

- a) Dua aspek dasar kemampuan : kecakapan umum dan terbiasa dengan aplikasi tertentu.
- b) Seorang yang cakap dalam pemrograman belum tentu cakap pula dalam aplikasi sains, atau sebaliknya.
- c) Ketidakakraban dengan lapangan aplikasi akan menghasilkan produktivitas rendah dan kualitas yang buruk.
- d) Yang dimaksud dengan kecakapan umum adalah kemampuan dasar dalam menulis program komputer dengan benar sedangkan ukuran produktivitas seorang programmer adalah banyak baris yang dihasilkan oleh programmer

tersebut per hari.

2. Komunikasi team :

- a) Meningkatnya ukuran produk yang dihasilkan akan menurunkan produktivitas programmer akibat meningkatnya kerumitan antara komponen-komponen program dan akibat meningkatnya komunikasi yang perlu dilakukan antara programmer, manajer, dan pelanggan.
- b) Jumlah lintasan komunikasi antar programmer yang terjadi dalam sebuah proyek adalah $n(n-1)/2$, dimana n adalah jumlah programmer yang terlibat dalam proyek tersebut.
- c) Penambahan lebih banyak programmer dalam sebuah proyek yang sedang berjalan akan menurunkan produktivitas, kecuali jika para programmer baru tersebut mempunyai tugas yang tidak bergantung kepada hasil kerja programmer lama.
- d) Hukum Brooks : Adding more programmers to a late project may make it later.

3. Kerumitan produk

Tiga level kerumitan produk : program aplikasi, program utility, program level sistem.

4. Notasi yang tepat

Bahasa pemrograman menetapkan notasi (baca : token, reserve word) baku, terutama untuk hal-hal yang berkaitan dengan matematika. penetapan notasi antar programmer (baca : perancang produk) harus dilakukan sehingga dapat dimengerti dengan jelas.

5. Pendekatan sistematis

Sistem menetapkan teknik dan prosedur baku. pembakuan dalam pengembangan dan pemeliharaan perangkat lunak masih belum mantap.

6. Kendali perubahan

Kelenturan sebuah produk perangkat lunak merupakan sebuah kekuatan, tetapi di pihak lain juga merupakan sumber kesulitan dalam proses perancangannya. perubahan terhadap produk harus tetap meminta persetujuan manajer sebagai

penanggung jawab proyek. Dampak perubahan harus dapat ditelusuri, diuji, dan didokumentasikan.

7. Tingkat teknologi

Peran penggunaan teknologi dalam proyek perangkat lunak misalnya menyangkut bahasa pemrograman, lingkungan mesin yang digunakan, teknik pemrograman, dan penggunaan tools tertentu. Bahasa pemrograman modern menyediakan fasilitas penyesuaian pendefinisian dan penggunaan data, konstruksi aliran kendali, fasilitas modular, dan concurrent programming

8. Tingkat keandalan

Setiap produk harus mempunyai keandalan standar. Peningkatan keandalan dihasilkan melalui perhatian yang sangat besar pada tahap analisa. Peningkatan keandalan akan menurunkan produktivitas. Boehm : rasio produktivitas antara dua produk dengan keandalan terendah dengan yang tertinggi adalah 2:1.

9. Pemahaman permasalahan

Pelanggan adalah penyumbang utama terhadap kegagalan dalam memahami masalah adalah :

- a) Tidak memahami permasalahan perusahaannya,
- b) Tidak mengerti kemampuan dan keterbatasan komputer,
- c) Tidak mempunyai pengetahuan dasar tentang logika dan algoritma.
- d) Software engineer tidak memahami lapangan aplikasi, gagal mendapatkan informasi kebutuhan pelanggan karena pelanggan bukan seorang end user.

10. Ketersediaan waktu

- a) Penetapan lama proyek dan jumlah programmer terlibat harus mempertimbangkan kemampuan pribadi setiap programmer serta kemampuan komunikasi antar mereka.
- b) Jumlah programmer yang makin banyak akan meningkatkan overhead di antaranya akibat keperluan komunikasi.
- c) Jumlah programmer yang makin sedikit berarti memperbanyak beban kerja kepada setiap programmer.
- d) Proyek 1 bulan dengan 6 programmer bisa saja diganti dengan proyek 6 bulan

dengan 1 programmer atau proyek 3 bulan dengan 3 programmer.

11. Persyaratan keterampilan

Berbagai keterampilan harus ada dalam sebuah proyek perangkat lunak, misalnya :

- a) Keterampilan berkomunikasi dengan pelanggan untuk memastikan keinginannya dengan sejelas-jelasnya
- b) Kemampuan dalam pendefinisian masalah dan perancangan.
- c) Kemampuan implementasi dengan penulisan program yang benar.
- d) Kemampuan debugging secara deduktif dengan kerangka “what if”.
- e) Dokumentasi.
- f) Kemampuan bekerja dengan pelanggan.
- g) Semua keterampilan tersebut harus senantiasa dilatih.

12. Fasilitas dan sumber daya

Fasilitas non teknis yang tetap perlu diperhatikan yang berkaitan dengan motivasi programmer misalnya : mesin yang baik, serta tempat yang tenang, atau ruang kerjanya dapat ditata secara pribadi.

13. Pelatihan yang cukup

Banyak programmer yang dilatih dalam bidang-bidang : ilmu komputer, teknik elektro, akuntansi, matematika, tetapi jarang yang mendapat pelatihan dalam bidang teknik perangkat lunak.

14. Kemampuan manajemen

Seringkali manajer proyek tidak mempunyai, atau hanya sedikit mengetahui, latar belakang teknik perangkat lunak. Di sisi lain terjadi promosi jabatan menjadi manajer dimana yang berpromosi tidak atau kurang mempunyai kemampuan manajemen. Sasaran yang tepat

Sasaran utama dari teknik perangkat lunak adalah pengembangan produk-produk perangkat lunak yang tepat untuk digunakan.

15. Peningkatan kualitas

Dua aspek yang menimbulkan keinginan untuk meningkatkan kualitas produk adalah seberapa banyak fungsi, keandalan, dan kemampuan dapat diberikan melalui

sejumlah pengembangan, masalah mendasar dari keterbatasan teknologi perangkat lunak

KETERAMPILAN YANG HARUS DIMILIKI SEORANG PROGRAMMER

Membaca kode sumber sebuah program, dari yang sederhana hingga relatif kompleks, adalah keterampilan yang harus dimiliki oleh seorang programmer. Istilah program di sini digunakan tidak hanya dalam pengertian aplikasi utuh, namun juga mencakup level segmen kode, fungsi/subrutin, pustaka atau modul yang digunakan oleh sebuah aplikasi utuh. Manfaat dari terbiasanya kita membaca program adalah semakin banyaknya sumber informasi yang dapat kita gunakan untuk memecahkan masalah.

Implementasi sebuah algoritma yang memodifikasi sebetulnya struktur data dalam sebuah program dapat kita adaptasikan ke dalam program yang kita buat sendiri dengan konteks dan struktur data yang sama sekali berbeda. Terlepas dari manfaat pedagogisnya, sayangnya keterampilan ini tidak banyak diajarkan (dan setahu saya tidak pernah menjadi matakuliah tersendiri di perguruan tinggi), namun baru terasa urgensinya saat kita disodori kewajiban memodifikasi program yang ditulis dan sebelumnya dikelola oleh orang lain. Dengan hanya berbekal kode sumber dan dokumentasinya (yang, patut disayangkan, tidak selalu komprehensif), dalam keadaan seperti ini seorang programmer harus mampu melanjutkan pengelolaan program tersebut tanpa menghancurkan organisasi kode yang telah ada atau menambah kompleksitas yang tidak perlu ada. Di sisi lain keadaan ini akan jarang ditemui oleh rata-rata programmer, karena lebih sering penulis asli dari program itulah yang harus melanjutkan pengelolaan, setelah memindahkan perhatian pada hal lain selama beberapa waktu.

Karenanya dapat disimpulkan bahwa secara umum, keterampilan yang harus dimiliki seorang programmer terkait dengan komprehensi kode sumber program, dengan derajat urgensi menurun (Hargo, 2008), adalah:

1. Memahami kode sumber yang ditulis sendiri pada saat ia tidak lagi mengingat detail mekanisme dari program tersebut.

2. Melanjutkan pengelolaan, menyesuaikan, mengembangkan dan (bila perlu) merombaknya untuk menyesuaikan program dengan kebutuhan pengguna tanpa mengorbankan kemudahan perawatan di masa mendatang.
3. Memiliki kemampuan sebagaimana dijelaskan dalam point 1. dan 2. untuk program yang ditulis dan didokumentasikan oleh programmer lain.
4. Membaca program untuk memperkaya perkakas yang dimiliki seorang programmer untuk memecahkan masalah.

KEWAJIBAN PROGRAMMER

Buat para *programmer*, Coding adalah kewajiban. Sama halnya menghafal obat-obatan bagi dokter, maupun menghafal undang-undang bagi pengacara. Sebagian *programmer* menganggap dirinya spesial, karena tidak semua orang IT dapat menguasai salah satu (atau bahkan beberapa) bahasa pemrograman.

Seorang programmer memiliki kewajiban sebagai berikut :

1. Memahami konsep dasar sistem operasi.

Kebanyakan dari programmer Indonesia biasanya membuat aplikasi di atas sistem operasi, sehingga banyak yang berpendapat bahwa tidak perlu memahami cara kerja sistem operasi. Untuk programmer profesional, pemahaman ini akan membuat programmer lebih siap untuk membuat aplikasi server yang biasanya *multithreaded* dan harus efisien digunakan dalam waktu yang lama. Pemahaman mendalam di salah satu sistem operasi juga merupakan nilai tambah yang signifikan. Dengan mengetahui struktur internal sistem operasi (misalnya Linux), programmer dapat mengetahui berbagai pertimbangan dalam merancang aplikasi besar yang terus berkembang.

2. Memahami konsep dasar jaringan.

Sebuah aplikasi tidak dapat berjalan sendiri. Aplikasi tersebut pasti harus berhubungan dengan internet, melayani banyak pengguna, atau berhubungan dengan perangkat lain seperti handphone atau PDA. Untuk itu, pemahaman atas konsep jaringan sangat penting.

3. Memahami konsep dasar relational database.

Setiap aplikasi pasti memiliki sebuah database dalam penyimpanan datanya untuk itu programmer khususnya Database Programmer ditekankan menguasai relational database.

4. Karena sekarang jaman internet, maka wajib memahami protokol HTTP, FTP, POP3, SMTP, SSH. Protokol HTTP sekarang adalah protokol yang paling banyak digunakan di internet.
5. Karena sekarang jaman globalisasi, maka wajib memahami Unicode. Unicode itu penting supaya aplikasi kita tetap bisa diinstal di komputer mana saja.
6. Lebih dari satu bahasa pemrograman.

Pemahaman lebih dari satu bahasa itu penting agar wawasan programmer lebih terbuka. Bahwa tidak ada bahasa yang one-fit-all, bahwa ada cara berpikir yang berbeda dalam tiap bahasa, bahwa komunitas tiap bahasa berbeda budayanya. Semua ini akan berkontribusi dalam pendewasaan seorang programmer dalam berdiskusi dan menanggapi perbedaan (terutama pendapat).

7. Cara menggunakan Version Control.

Dalam dunia kerja, penggunaan version control adalah wajib. Ini standar (de facto) internasional. Jika mempunyai project opensource, baik di Sourceforge, Apache, Codehaus, dan semua hosting project opensource, pasti programmer akan diberikan version control.

SIKAP PROGRAMMER TERHADAP KLIEN

1. Mempunyai sikap & kepribadian baik, komunikatif, mudah beradaptasi dengan lingkungan kerja, cekatan & fleksibel.
2. Mampu bekerja berorientasi jadwal, mengatur pekerjaan multiple project dan bekerja sama dalam team.
3. Membuat kontrak kerja dengan klien.
4. Menyukai dan mengerti dasar-dasar pemrograman.