

Pemrograman dan Bahasa

Pemrograman

Lisensi Dokumen:

Copyright © 2003 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.

Hendro Subagyo

hendro-s@casl.cs.uec.ac.jp

PENDAHULUAN

Tulisan ini membahas secara singkat pemrograman dan bahasa pemrograman, rambu-rambu dalam belajar atau mengajar bahasa pemrograman, serta masalah pemilihan bahasa pemrograman pertama bagi pemula.

Kalau kita membayangkan abad informasi, yang muncul adalah kosakata seperti komputer, network, komunikasi dan informasi itu sendiri. Memang abad informasi dan komputer tidak dapat dipisahkan. Perkembangan ilmu dan teknik komputer mempercepat perkembangan informasi, atau sebaliknya permintaan akan informasi yang terus bertambah baik dari segi kuantitas maupun kualitas, mendorong percepatan perkembangan ilmu dan teknik komputer dan ilmu-ilmu lainnya.

Dengan semangat *efisiensi*, keberadaan komputer di kantor menjadi sangat penting. Di awal perkembangan komputer, program aplikasi sangat sedikit, primitif dan “sulit”, sehingga setiap kantor memerlukan programmer. Bagi Anda yang dapat menggunakan WordStar ataupun LaTeX, maka dapat dikatakan bahwa Anda pun seorang programmer, karena Anda harus mengetikkan perintah-perintah khusus untuk menghasilkan walau sebuah surat sederhana. Kini dengan munculnya aplikasi-aplikasi mutakhir dan teknologi *plug-in*, pengguna tidak perlu lagi susah-susah menjadi programmer. Apalagi dengan munculnya teknologi *visual-programming*, maka seorang programmer pun tidak perlu mengetikkan seluruh kode dari sebuah program yang dibuatnya.

Namun demikian, bagaimana pun juga, keberadaan programmer masih sangat diperlukan untuk usaha otomatisasi berbagai bidang kehidupan. Permintaan yang mendesak dari dunia bisnis, seringkali membuat dunia pendidikan kewalahan dalam mensuplai programmer yang sanggup menghadapi perkembangan teknologi komputer yang sangat cepat ini.

PEMROGRAMAN DAN BAHASA PEMROGRAMAN

Pemrograman dapat diklasifikasikan sebagai berikut [Rechenberg]:

1. Berorientasi prosedur (*procedural oriented*)
2. Berorientasi fungsi (*functional oriented*)
3. Berorientasi logik (*logic oriented*)
4. Berorientasi obyek (*object oriented*)

Masing-masing memiliki kelebihan tersendiri. Kadangkala dalam membangun suatu aplikasi dibutuhkan gabungan metode pemrograman tersebut. Misalnya dalam C++ dan Java (bahasa pemrograman berorientasi obyek), kita masih dapat menemukan tehnik-tehnik pemrograman berorientasi prosedur dalam setiap *method/function member* dalam obyek-obyeknya.

Suatu bahasa pemrograman pada asalnya hanya dapat digunakan dalam satu metode. Pascal mulanya untuk *procedural-oriented*, Lisp untuk *functional-oriented*, Smaltalk untuk *object-oriented* dan lain-lain. Seiring dengan perkembangan teknologi informasi dan komputer, puluhan bahkan ratusan bahasa pemrograman baru lahir. Masing-masing memiliki keunikan dan kelebihan spesifik. Ada yang khusus untuk jenis komputer tertentu, ada pula yang khusus untuk paradigma pemrograman tertentu.

BELAJAR PEMROGRAMAN

Lalu bagaimana seorang programmer mulai belajar memprogram? Dalam menjawab pertanyaan ini kita sering terjebak dalam masalah pemilihan bahasa pemrograman pertama bagi pemula. Bahasa pemrograman apakah yang paling tepat diajarkan kepada seorang pemula, sehingga kelak ia dapat survive di antara sekian ratus bahasa pemrograman. Sampai saat ini pun perdebatan masih terus berlanjut untuk memilih bahasa pemrograman pertama yang paling tepat. Tidak dapat diragukan lagi bahwa bahasa pemrograman hampir sebanyak bahasa manusia, atau mungkin lebih banyak lagi.

Pemilihan bahasa pemrograman pertama merupakan masalah pelik. Sebab bahasa pemrograman pertama akan mempengaruhi cara berfikir programmer di masa yang akan datang. Programmer dengan bahasa pertamanya Pascal akan lebih mudah berpindah ke bahasa pemrograman yang memiliki paradigma sama, yaitu berorientasi prosedur seperti C, Modula, Oberon. Tetapi programmer tersebut kesulitan untuk berpindah ke bahasa pemrograman dengan paradigma yang lain seperti CLOS dan Scheme, yaitu bahasa pemrograman keluarga Lisp. Selain daripada itu, setelah menentukan paradigmanya, masalah berikutnya yang muncul adalah memilih bahasa pemrograman yang paling dominan dalam paradigma tersebut. Misalnya, untuk paradigma berorientasi fungsi manakah yang paling tepat diajarkan kepada pemula, apakah CLOS atau Scheme atau EmacsLisp. Masalah kedua ini berkaitan dengan, kemudahan dalam belajar, dialek dalam keluarga bahasa pemrograman tersebut, atau bahkan kecenderungan pasar.

Pada umumnya, lembaga pendidikan memilih paradigma pemrograman berorientasi prosedur sebagai paradigma pertama siswa mereka. Bahasa yang digunakan umumnya Pascal atau C. Hal ini dikarenakan paradigma tersebut lebih sering dipakai di dalam kehidupan sehari-hari. Setelah itu, disesuaikan dengan situasi dan kondisi diajarkan paradigma lain. Tetapi ada pula lembaga pendidikan yang memilih bahasa ML, bahasa dengan paradigma berorientasi fungsi, seperti MIT. Alasan MIT,

karena sebagian mahasiswa sebelumnya telah memiliki ketrampilan dan pengetahuan dalam paradigma pemrograman berorientasi prosedur, demi alasan keadilan dipilahlah bahasa dan pemrograman yang asing untuk sebagian besar mahasiswa. Dengan demikian sebagian besar mahasiswa memulai belajar pemrograman dari awal yang sama.

Yang perlu kita perhatikan adalah mengajar/belajar pemrograman tidaklah sama dengan belajar bahasa pemrograman. Agar kita tidak salah langkah dalam mengajar/belajar pemrograman, Derek Andrew [Woodman, pp.255-276] memberikan rambu-rambu sebagai berikut:

- Kita tidak belajar atau mengajar bahasa pemrograman, tetapi belajar atau mengajar bagaimana cara memprogram
- Kita tidak belajar atau mengajar bahasa pemrograman, tetapi belajar atau mengajar bagaimana memecahkan masalah
- Kita tidak belajar atau mengajar bahasa pemrograman, tetapi belajar atau mengajar bagaimana mendesain sistem
- Kita tidak belajar atau mengajar bahasa pemrograman, tetapi belajar atau mengajar prinsip-prinsip bahasa pemrograman
- Kita tidak belajar atau mengajar bahasa pemrograman, tetapi belajar atau mengajar teori semantik
- Kita tidak belajar atau mengajar bahasa pemrograman, tetapi belajar atau mengajar teori pemrograman

Bagi seorang pengajar point terpenting adalah point pertama: perlunya mengajari ahli komputer kita bagaimana cara memprogram. Asalkan suatu bahasa pemrograman itu cukup memadai bagi kita untuk belajar atau mengajar pemrograman, apapun bahasa pemrograman yang kita gunakan tidak menjadi masalah.

KESIMPULAN

Tulisan ini membahas secara singkat pemrograman dan bahasa pemrograman, serta rambu-rambu dalam belajar atau mengajar bahasa pemrograman. Pemilihan suatu bahasa pemrograman pertama bagi pemula tergantung dari apa yang akan dikerjakan olehnya di masa mendatang.

REFERENSI

- [Woodman] Mark Woodman. *Programming Language Choice: practice and experience*. Thompson Computer Press, 1996.
- [Rechenber] Rechenber, P. *Programming Language as Thought Models*. Structured Programming, 11. 1990
- [Booch] Booch, Grady. *Object-Oriented Analysis and Design*. The Benjamin r Commings Publishing Company Inc.
- [Martin] J. Martin and J.J. Odell. *Object-Oriented Analysis and Design*. Prentice Hall, 1992.