

SINERGI ISO 9001:2000 - CMMI PADA INDUSTRI PENGEMBANG PERANGKAT LUNAK

Leo Willyanto Santoso, Yulia

Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra

e-mail: leow@petra.ac.id, yulia@petra.ac.id

ABSTRAK: Semua hal dalam hidup ini selalu memperhatikan masalah kualitas, baik itu berupa kualitas produk ataupun kualitas layanan. Pada industri perangkat lunak, hal ini tidak dibedakan, kebutuhan untuk menghasilkan produk yang berkualitas akan meningkatkan kekompleksan dari sebuah sistem. Standarisasi dalam penentuan kualitas, ISO sudah menjadi keharusan bagi para pengembang perangkat lunak. Apabila diteliti lebih lanjut, ISO 9001:2000 masih mempunyai banyak kekurangan, khususnya untuk diaplikasikan dalam produk perangkat lunak. Pada penelitian ini, akan dibahas analisis perbandingan dengan melakukan pendekatan kualitas untuk meningkatkan proses perangkat lunak dengan melakukan mapping antara ISO 9001:2000 dengan CMMI. Hal-hal yang dibahas berupa aspek-aspek dari standard ISO dan CMMI versi 1.1. Pada penelitian ini, dapat diperoleh strategi untuk menerapkan ISO 9001:2000 dan CMMI secara sinergi ketika merencanakan untuk melakukan peningkatan proses perangkat lunak. Selain itu, dihasilkan juga analisis bagaimana untuk beralih dari ISO 9001:2000 ke CMMI. Dalam penerapannya pada industri perangkat lunak, PT 'X', diperoleh berbagai keuntungan yang sebelumnya tidak didapatkan.

Kata kunci: ISO, ISO 9001:2000, *capability maturity model integrated*.

ABSTRACT: All sectors of society including software industry, have a big concern about quality, both of product quality and service quality. The need to manage risk, has a strong relation with quality and increases system complexity. Standarization in determining quality, ISO is a must for all software developer. If we study further, ISO 9001:2000 still has some weakness especially for implementation in software product. This paper presents a comparative analysis of quality approaches to risk management. This paper also provide a general overview of all the risks aspects treated in ISO 9001:2000 standards and CMMI model version 1.1. In this research, we obtain a strategy to implement ISO 9001:2000 and CMMI together when we make a planing to increase software process. Besides it produces analysis how to change position from ISO 9001:2000 to CMMI. In the implementation on software industry PT 'X', obtain get various benefits that has never been obtained before.

Keywords: ISO, ISO 9001:2000, *capability maturity model integrated*.

PENDAHULUAN

Tujuan akhir dari proses rekayasa perangkat lunak adalah menghasilkan perangkat lunak berkualitas tinggi. Philip Crosby (Crosby, 1979), dalam bukunya yang terkenal tentang kualitas, menerangkan bahwa manajemen kualitas bukanlah suatu hal yang tidak diketahui. Para pengembang perangkat lunak percaya bahwa kualitas perangkat lunak merupakan sesuatu yang harus kita perhatikan sejak tahap awal proses rekayasa perangkat lunak.

Apabila kita mengamati sebuah item dengan didasarkan pada sifat pengukurannya, ada dua jenis kualitas yang ada, yaitu kualitas disain dan kualitas konformansi.

Kualitas desain mengacu pada karakteristik yang ditentukan oleh desainer terhadap suatu item tertentu. Nilai material, toleransi, dan spesifikasi kinerja, semua memberikan kontribusi terhadap kualitas desain. Karena material dengan nilai yang lebih tinggi digunakan dan toleransi yang lebih ketat serta tingkat kinerja yang lebih baik ditentukan, maka kualitas desain dari suatu produk bertambah, bila

produk dihasilkan sesuai dengan spesifikasi yang ditentukan.

Kualitas konformansi adalah tingkat dimana spesifikasi desain terus diikuti selama pembuatan. Semakin tinggi tingkat konformansi, semakin tinggi tingkat kualitas konformansi. Dalam pengembangan perangkat lunak, kualitas desain mencakup syarat, spesifikasi, dan desain sistem. Kualitas konformansi adalah suatu masalah yang difokuskan pada implementasi. Bila implementasi mengikuti desain dan sistem yang dihasilkan memenuhi persyaratan serta tujuan kinerja, maka kualitas konformansi menjadi tinggi.

STANDARISASI PERANGKAT LUNAK

Untuk memberikan penilaian terhadap proses perangkat lunak secara berimbang, maka diperlukan adanya standar yang baku. Ada beberapa jenis standar yang banyak digunakan oleh masyarakat industri, yaitu antara lain berdasarkan terminologi, klasifikasi, metode atau proses, produk, dan kode pembuatan.

Antara satu standar dengan standar yang lain mempunyai persamaan dan perbedaan yang signifikan. Standar-standar itu dikeluarkan oleh sebuah organisasi atau badan dunia, yaitu antara lain ISO, IEC, CEN, BSI, ANSI dan DIN. Sebenarnya masih banyak lagi organisasi yang mengeluarkan standar, tetapi cukup kita batasi yang berhubungan dengan segi kualitas saja.

Secara umum, tujuan dari berbagai standarisasi yang ada adalah:

1. Untuk menyediakan komunikasi antar pelbagai kelompok yang lebih berarti.
2. Untuk memperkenalkan *economy of effort*.
3. Untuk melindungi kepentingan pelanggan, terutama tentang kualitas.
4. Untuk memajukan *quality of life*, yaitu keamanan, kesehatan dan perlindungan lingkungan sekitar.
5. Untuk membantu mengurangi *trade barrier*.

Secara ringkas, beberapa standarisasi perangkat lunak yang sudah ada antara lain:

- ISO/IEC 9126, kualitas produk perangkat lunak
- ISO/IEC 14598, evaluasi produk perangkat lunak
- ISO 9000 series (BS 5750, EN 29000) sistem kualitas, tidak spesifik untuk perangkat lunak. Sebagai petunjuk pengaplikasian ke perangkat lunak, maka digunakan ISO 9000-3.
- Software Process Improvement and Capability dTermination (SPICE), menggabungkan standar dari ISO/IEC 15504

STANDAR KUALITAS ISO 9000

Sistem jaminan kualitas dapat didefinisikan sebagai struktur, tanggung jawab, prosedur, proses, dan sumber-sumber daya organisasi untuk mengimplementasikan manajemen kualitas (ANSI/ASQC, 1987). ISO 9000 menjelaskan elemen jaminan kualitas dalam bentuk yang umum yang dapat diaplikasikan pada berbagai bisnis tanpa memandang produk dan jasa yang ditawarkan.

Agar terdaftar dalam satu model sistem jaminan kualitas yang ada pada ISO 9000, sistem kualitas dan operasi perusahaan diperiksa oleh auditor bagian ketiga untuk memeriksa kesesuaiannya dengan standar dan operasi efektif. Bila registrasi itu berhasil, perusahaan diberi sertifikat dari badan registrasi yang diwakili oleh auditor. Audit pengawasan tengah tahunan terus dilakukan untuk memastikan kesesuaiannya dengan standar yang sudah ditetapkan.

Model jaminan kualitas ISO 9000 memberlakukan perusahaan sebagai jaringan proses yang saling terhubung (interkoneksi). Suatu sistem kualitas, supaya sesuai dengan ISO, proses-prosesnya harus

menekankan pada area yang telah diidentifikasi pada standar ISO, dan harus didokumentasi dan dipraktikkan sebagaimana dijelaskan. Pendokumentasian proses membantu organisasi untuk memahami, mengontrol, dan mengembangkan jaringan proses yang mungkin dapat mendatangkan keuntungan terbesar bagi organisasi yang merancang dan mengimplementasikan sistem kualitas yang sesuai dengan ISO.

ISO 9000 menggambarkan elemen sebuah sistem jaminan kualitas secara umum. Elemen-elemen tersebut mencakup struktur, prosedur, proses, organisasi, dan sumber daya yang dibutuhkan untuk mengimplementasikan rencana kualitas, kontrol kualitas, jaminan kualitas, dan pengembangan kualitas. Tetapi, ISO 9000 tidak menggambarkan bagaimana organisasi seharusnya mengimplementasi elemen-elemen kualitas tersebut. Sebagai konsekuensi, ada tantangan dalam mendesain dan mengimplementasi suatu sistem jaminan kualitas yang memenuhi standar sesuai dengan produk, layanan dan budaya perusahaan.

ISO 9001 adalah standar jaminan kualitas yang berlaku untuk rekayasa perangkat lunak. Standar tersebut berisi 20 syarat yang harus ada untuk mencapai sistem jaminan kualitas yang efektif. Karena standar ISO 9001 dapat diaplikasikan pada semua disiplin rekayasa/engineering, maka dikembangkan sekumpulan khusus pedoman ISO (ISO 9000-3) untuk membantu menginterpretasikan standar untuk digunakan pada proses perangkat lunak.

Dua puluh syarat yang digariskan oleh ISO 9001 menekankan topik-topik berikut (Pressman, 1997):

1. Tanggung jawab manajemen.
2. Sistem kualitas.
3. Kajian kontrak.
4. Kontrol desain.
5. Kontrol data dan dokumen.
6. Pembelian.
7. Kontrol terhadap produk yang disuplai oleh pelanggan.
8. Identifikasi dan kemampuan penelusuran produk.
9. Kontrol proses.
10. Pemeriksaan dan pengujian.
11. Kontrol pemeriksaan, pengukuran, dan perlengkapan pengujian.
12. Pemeriksaan dan status pengujian
13. Kontrol ketidaksesuaian produk.
14. Tindakan preventif dan korektif.
15. Penanganan, penyimpanan, pengepakan, preservasi, dan penyampaian.
16. Kontrol terhadap catatan kualitas.
17. Audit kualitas internal.

18. Pelatihan
19. Pelayanan
20. Teknik statistik

Untuk dapat didaftar dalam ISO 9001, organisasi perangkat lunak harus membuat kebijakan dan prosedur yang memberi tekanan pada masing-masing syarat tersebut dan kemudian dapat menunjukkan bahwa prosedur dan fungsi itu telah diikuti. Untuk penjelasan lebih lanjut, pembaca yang tertarik dengan informasi mengenai ISO 9001 dianjurkan melihat (Schulmeyer, 1994) dan (Essential, 1995).

CAPABILITY MATURITY MODEL (CMM)

Standarisasi ISO 9001 mempunyai beberapa keterbatasan, yaitu: (1) Tidak dibuat secara khusus untuk perangkat lunak dan tidak mudah diinterpretasikan untuk perangkat lunak meskipun sudah tersedianya ISO 9000-3, (2) hanya menyediakan satu level penilaian saja, (3) perangkat lunak tidak bisa dianggap seperti produk lainnya.

Untuk mengatasi beberapa keterbatasan dari ISO, maka telah dikembangkan sebuah model yang dinamakan Capability Maturity Model Integrated (CMMI). Perbedaan utama antara ISO dan CMMI dapat dilihat pada Tabel 1.

Tabel 1. Perbedaan utama ISO dengan CMMI

ISO	CMMI
- Merupakan sebuah standar	- Merupakan sebuah model
- Membutuhkan interpretasi untuk organisasi dengan banyak program.	- Bersifat institusional dan mudah diimplementasikan
- Diimplementasikan secara luas	- Mengakomodasi banyak organisasi dengan pelbagai program.
- Merupakan kumpulan dari kebutuhan yang harus dipenuhi semuanya.	- Langkah-langkahnya bersifat progresif dan lebih jelas.

Capability Maturity Model dikembangkan oleh Software Engineering Institute di Pittsburgh pada tahun 1987. Capability Maturity Model mendukung proses penilaian secara bertingkat. Penilaiannya tersebut berdasarkan kuisioner. Capability Maturity Model ini dikembangkan secara khusus untuk perangkat lunak yang juga mendukung peningkatan proses.

Capability Maturity Model ini dikembangkan berdasarkan beberapa model perangkat lunak lainnya, yaitu model TRILLIUM untuk telecom, model BOOTSTRAP di eropa, dan *Software Diagnostic/Process Professional* di skotlandia.

PENINGKATAN PROSES PERANGKAT LUNAK

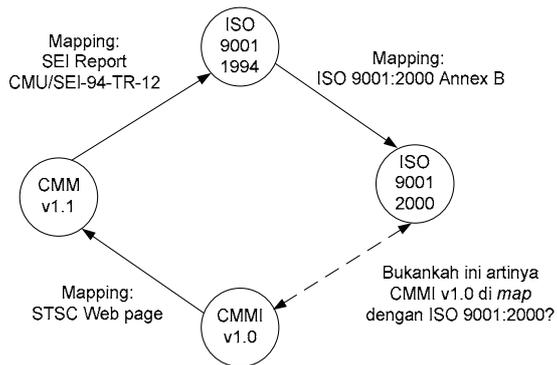
Industri perangkat lunak di Indonesia berkembang dengan pesat. Menurut Badan Pusat Statistik (BPS), pertumbuhan industri perangkat lunak di Indonesia mencapai 40% pada tahun 2003. Sudah banyak diantara industri pengembang perangkat lunak tersebut yang berhasil memperoleh standarisasi mutu, ISO.

Diantara industri perangkat lunak yang ada sekarang ini, masih banyak yang pengelolaannya belum terorganisasi dengan baik, atau bersifat *immature*. Hal ini ditandai dengan kualitas produk yang buruk, fungsionalitas perangkat lunak yang kurang, biaya yang membengkak dan waktu penyelesaian proyek yang terlambat. Yang membuat ironi, ternyata industri pengembang perangkat lunak yang sudah memperoleh ISO 9001 sekalipun, masih sering mengalami kegagalan-kegagalan dalam project pengembangan perangkat lunak seperti diatas.

Hal ini yang menggugah penulis untuk mengetahui penyebab di balik fenomena seperti itu. Untuk mengatasi hal itu, sebenarnya bisa dilakukan dengan beberapa cara, salah satunya adalah dengan menerapkan CMM setelah kita memperoleh sertifikasi ISO. Untuk melakukan hal tersebut tidak mudah. Untuk berpindah ke sebuah model yang baru, kita dapat melakukan langkah-langkah berikut ini (Ketola, 2000):

1. Analisis ISO 9001:2000 dan ISO 9004:2000.
Dalam analisis ini, kita membandingkan antara ISO 9001:2000 dengan ISO 9001:1994 dan ISO 9000-3:1994. Setelah itu, kita analisa perbedaan yang ada di antara standarisasi ISO tersebut.
2. Analisis CMMI.
Dalam analisis ini, kita membandingkan antara CMM v1.1 dengan CMMI, kemudian kita analisa perbedaan antara keduanya.
3. Mencari sinergi antara ISO 9001:2000 dan CMMI
4. Menentukan status peningkatan proses dari organisasi yang menggunakan standar ISO ataupun CMMI
5. Menentukan hal-hal yang bersifat umum dalam berbagai kesempatan peningkatan proses berdasarkan pada sinergi antara ISO dan CMMI.
6. Membuat sebuah *transition plan*.

Apabila kita melakukan proses mapping antara standarisasi ISO dengan model CMM, maka diperoleh hasil seperti pada Gambar 1. berikut ini.



Gambar 1. Mapping antara ISO dan CMM

Dari hasil mapping, kita dapat mengetahui masing-masing kelebihan. Kelebihan dari ISO adalah:

- Diterapkan secara luas oleh banyak industri.
- Meningkatkan fungsionalitas suatu institusi/organisasi.
- Mendapatkan pengakuan internasional.
- Memberikan kebebasan kepada siapapun yang ingin menerapkannya.

Sedangkan kelebihan dari CMMI adalah:

- Mempunyai fitur-fitur yang bersifat institusional, yaitu komitmen, kemampuan untuk melakukan sesuatu, analisis dan pengukuran serta verifikasi implementasi.
- Tersedianya “Road Map” untuk peningkatan lebih lanjut.

Selain itu, kita juga membuat *mapping* untuk fitur-fitur umum dari ISO - CMMI, hasilnya adalah:

- 4.0 Sistem Manajemen Kualitas
 - GP 2.1 Menentukan Kebijakan Organisasi
 - GP 2.2 Merencanakan Proses
 - GP 2.3 Menyediakan Sumber daya
 - GP 3.1 Menentukan Proses
 - GP 2.6 Mengatur Konfigurasi
 - GP 2.8 Memantau dan Mengontrol Proses
 - GP 3.2 Mengumpulkan Informasi Pengembangan
 - GP 2.9 Mengevaluasi Ketaatan dengan Objektif
- 5.0 Tanggung jawab Manajemen
 - GP 2.1 Menentukan Kebijakan Organisasi
 - GP 2.2 Merencanakan Proses
 - GP 2.4 Memberikan Tanggung jawab
 - GP 2.6 Mengatur Konfigurasi
 - GP 2.7 Mengidentifikasi *Stakeholder* yang sesuai
 - GP2.10 Mereview Status dengan Manajemen
 - GP 3.1 Menentukan Proses
- 6.0 Manajemen Sumber daya
 - GP 2.3 Menyediakan Sumber daya
 - GP 2.5 Melatih Pekerja

- 7.0 Realisasi Produk
 - GP 2.2 Merencanakan Proses
 - GP 2.6 Mengatur Konfigurasi
- 8.0 Pengukuran, analisis dan pengembangannya
 - GP 2.6 Mengatur Konfigurasi
 - GP 2.8 Memantau dan Mengontrol Proses
 - GP 2.9 Mengevaluasi Ketaatan dengan Objektif
 - GP 3.2 Mengumpulkan Informasi Pengembangan

Dari hasil *mapping*, kita dapat menentukan apa saja yang tidak ada pada ISO dan CMMI. Yang tidak ada dalam standarisasi ISO 9001:2000 adalah:

- Pembuatan dan pengelolaan aset-aset proses dari organisasi (*repository* dan *database*)
- Area Proses (PA)
 - Training Organisasi
 - Manajemen Resiko
 - Analisis dan Resolusi Keputusan
 - Manajemen Proyek Kuantitatif
 - Analisis Kasual
- Institusionalisasi

Sedangkan yang tidak tercover di CMMI adalah:

- *Customer focus* dan kepuasan pelanggan
- Tanggung jawab manajemen untuk kualitas
- Infrastruktur (bangunan, peralatan dan tempat kerja)
- Kepemilikan *customer*

Dari hasil tersebut, apabila kita berkeinginan untuk mengimplementasikan ISO 9001:2000 – CMMI secara sinergi, maka kita harus melakukan langkah-langkah berikut ini:

1. Mengembangkan infrastruktur.
 - CMMI: memperhatikan semua GP yang ada.
 - ISO: memperhatikan ketentuan 4.0 dan 5.0
 - Membuat SEPG dan *Management Steering Group*
 - Mengembangkan OSSP (OPD)
2. Memilih PA yang akan diimplementasikan pertama kali.
 - CM, PPQA, MA dan PP
 - PMC, RM, SAM
3. Mengimplementasikan PA yang berhubungan dengan registrasi ISO.
4. Mengimplementasikan kebutuhan-kebutuhan ISO yang spesifik.
 - *Management Representative*
 - Lingkungan kerja
5. Mengimplementasikan PA lainnya yang penting
 - Manajemen risiko
 - Analisis dan Resolusi Keputusan

UJI COBA

Penulis telah melakukan penelitian di salah satu industri pengembang perangkat lunak, PT ‘X’, dimana mempekerjakan lebih dari 20 orang *programmer* dan *system analyst*. Dalam usaha untuk meningkatkan proses perangkat lunak, PT ‘X’ yang telah mendapatkan ISO 9001 berusaha untuk mengadopsi model dalam CMMI.

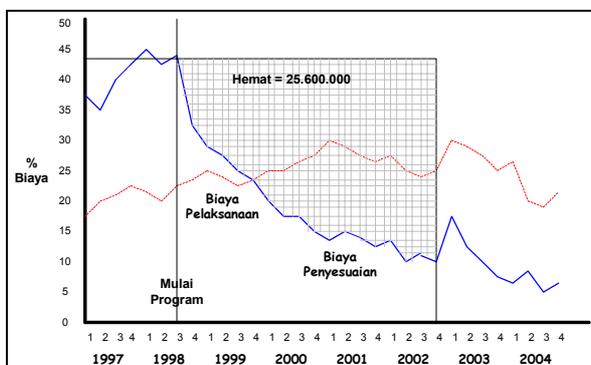
Hasil penilaian secara periodik menunjukkan bahwa level 1 CMMI dapat dicapai pada tahun 1997. Pada tahun 1999, level 2 sudah tercapai dan satu tahun kemudian, level 3 juga sudah tercapai. Level 4 dapat dicapai pada tahun 2004 ini. Untuk sampai mencapai level 4 ini, biaya yang dikeluarkan tidak kurang dari 20 juta rupiah. Biaya kualitas disini dibedakan menjadi empat macam, yaitu:

1. Biaya Pelaksanaan / *Performance*
Biaya yang dikeluarkan untuk bangunan dan gedung
2. Biaya Penyesuaian / *Nonconformance*
Biaya yang dikeluarkan untuk *rework*.
3. Biaya Penilaian / *Appraisal*
Biaya yang dikeluarkan untuk proses testing.
4. Biaya Pencegahan / *Prevention*
Biaya yang dikeluarkan untuk mempertahankan segala sesuatu yang sudah dicapai.

Untuk mengetahui biaya masing-masing pembiayaan per tahunnya, dapat dilihat pada Tabel 2, sedangkan grafik perbandingannya dapat dilihat pada Gambar 2.

Tabel 2. Biaya kualitas PT ‘X’

Tahun	Tingkat	Perform.	Nonconf.	Appraise.	Prevent.
1997	1	34%	41%	15%	7%
1999	2	55%	18%	14%	12%
2000	3	66%	11%	23%	20%
2004	4	76%	6%	18%	15%



Gambar 2. Penghematan Kerja

Keuntungan yang diperoleh dari PT ‘X’ adalah:

1. Pengurangan kerja
Berhasil mengurangi waktu *integration test* dan *defect fixing* dengan skala 5:1
2. Meningkatkan produktifitas hampir 3 kali lipat dalam 5 tahun terakhir.
3. Mengurangi *defect rate* sampai 4 kali lipat.
4. Meningkatkan *performance* kerja dengan mengurangi project-project yang terlambat penyelesaiannya.
5. Memperoleh bonus Rp. 15.800.000 untuk project-project yang dapat diselesaikan lebih awal dari jadwal yang telah ditetapkan.

KESIMPULAN

Beberapa kesimpulan yang dapat ditarik dari penelitian ini adalah sebagai berikut:

1. ISO 9001:2000 dan CMMI bersifat sinergi.
2. Peningkatan proses perangkat lunak dengan menerapkan CMMI akan membentuk industri pengembang perangkat lunak yang profesional.
3. Organisasi dan industri yang telah menerapkan CMMI level 3 telah siap untuk registrasi ISO 9001:2000 dengan sedikit penyesuaian.
4. Organisasi dan industri yang telah memperoleh ISO 9001:2000 akan membutuhkan tenaga ekstra untuk mencapai CMMI level 2 atau 3.

DAFTAR PUSTAKA

1. ANSI/ASQC A3-1987, *Quality Systems Terminology*, 1987.
2. CMMI Documents, www.sei.cmu.edu
3. Crosby, P., *Quality is Free*, McGraw-Hill, 1979
4. Essential Software Engineering, *ISO 9000 Software Development*, R.S. Pressman & Associates, Inc., 1995.
5. Ince, D., *ISO 9001 and Software Quality Assurance*, McGraw Hill, 1994.
6. ISO Documents, www.iso.ch
7. Ketola, Jeanne, Kathy Roberts, *ISO 9000:2000 In a Nutshell*, Paton Press, Chico, CA, 2000.
8. Pressman, Roger S., *Software Engineering: A Practitioner’s Approach*, 4th edition McGraw Hill, 1997.
9. Sommerville, Ian, *Software Engineering*, 6th edition, Addison-Wesley Publishers, 2001.
10. Schulmeyer, G.C., dan J.I. McManus (eds), *Handbook of Software Quality Assurance*, Van Nostrand Reinhold, 1987.
11. Schmauch, C.H., *ISO 9000 for Software Developers*, ASQC Quality Press, Milwaukee, WI, 1994.
12. Wallmuller, E., *Software Quality Assurance: A Practical Approach*, Prentice-Hall, 1995.