

JAMINAN KUALITAS PERANGKAT LUNAK

Jaminan kualitas perangkat lunak (*Software Quality Assurance / SQA*) adalah aktivitas pelindung yang diaplikasikan pada seluruh proses perangkat lunak.

SQA meliputi :

- Pendekatan manajemen kualitas.
- Teknologi rekayasa perangkat lunak yang efektif (metode dan piranti).
- Kajian teknik formal yang diaplikasikan pada keseluruhan proses perangkat lunak.
- Strategi pengujian *multitiered* (deret bertingkat).
- Kontrol dokumentasi perangkat lunak dan perubahan yang dibuat untuknya.
- Prosedur untuk menjamin kesesuaian dengan standar pengembangan perangkat lunak (bila dapat diaplikasikan).
- Mekanisme pengukuran dan pelaporan.

A. KONSEP KUALITAS

1. Kualitas

American Heritage Dictionary mendefinisikan kata kualitas sebagai "sebuah karakteristik atau atribut dari sesuatu." Sebagai atribut dari sesuatu, kualitas mengacu pada karakteristik yang dapat diukur, sesuatu yang dapat kita bandingkan dengan standar yang sudah diketahui, seperti panjang, warna, sifat kelistrikan, kelunakan, dsb. Tetapi perangkat lunak, yang sebagian besar merupakan entitas intelektual, lebih menantang untuk dikarakterisasi daripada objek fisik.

Pengukuran karakteristik program benar-benar ada. Properti tersebut mencakup kompleksitas siklomatik, kohesi, jumlah *function point*, baris kode, dll.

Bila kita mengamati sebuah item dengan didasarkan pada sifat pengukurannya, ada dua jenis kualitas yang ada, yaitu **kualitas desain** 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.

2. Kontrol Kualitas

Kontrol kualitas merupakan serangkaian pemeriksaan, kajian, dan pengujian yang digunakan pada keseluruhan siklus pengembangan untuk memastikan bahwa setiap produk memenuhi persyaratan yang ditetapkan.

Kontrol kualitas mencakup *loop* (kalang) umpan balik pada proses yang menciptakan produk kerja. Kombinasi pengukuran dan umpan balik memungkinkan kita memperbaiki proses bila produk kerja yang diciptakan gagal memenuhi spesifikasi mereka. Pendekatan tersebut memandang kontrol kualitas sebagai bagian dari proses manufaktur.

3. Jaminan Kualitas

Jaminan kualitas terdiri atas fungsi auditing dan pelaporan manajemen.

Tujuan jaminan kualitas untuk memberikan data yang diperlukan oleh manajemen untuk menginformasikan masalah kualitas produk, sehingga dapat memberikan kepastian dan konfidensi bahwa kualitas produk dapat memenuhi sasaran.

4. Biaya Kualitas

Biaya kualitas menyangkut semua biaya yang diadakan untuk mengejar kualitas atau untuk menampilkan kualitas yang berhubungan dengan aktivitas. Studi tentang biaya kualitas dilakukan untuk memberikan garis dasar bagi biaya kualitas yang sedang digunakan, untuk mengidentifikasi kemungkinan pengurangan biaya kualitas serta memberikan basis perbandingan yang ternormalisasi.

Biaya kualitas dapat dibagi ke dalam biaya-biaya yang dihubungkan dengan pencegahan, penilaian, dan kegagalan.

Biaya pencegahan meliputi :

- Perencanaan kualitas
- Kajian teknis formal
- Perlengkapan pengujian
- Pelatihan

Biaya penilaian meliputi aktivitas untuk memperoleh wawasan mengenai kondisi produk “pertama kali” pada masing-masing proses.

Contoh biaya penilaian meliputi :

- Inspeksi in-proses dan interproses
- Pemeliharaan dan kalibrasi peralatan
- Pengujian

Biaya kegagalan adalah biaya yang akan hilang bila tidak ada cacat yang muncul sebelum produk disampaikan kepada pelanggan. Biaya kegagalan dapat dibagi lagi ke dalam biaya **kegagalan internal** dan **eksternal**.

Biaya kegagalan internal adalah biaya yang diadakan bila kita mendeteksi suatu kesalahan dalam produk sebelum produk dipasarkan.

Biaya kegagalan internal meliputi :

- Pengerjaan kembali
- Perbaikan
- Analisis mode kegagalan

Biaya kegagalan eksternal adalah biaya yang berhubungan dengan cacat yang ditemukan setelah produk disampaikan kepada pelanggan.

Contoh biaya kegagalan eksternal meliputi :

- Resolusi keluhan
- Penggantian dan pengembalian produk
- Dukungan *help line*
- Kerja jaminan

B. JAMINAN KUALITAS PERANGKAT LUNAK

Kualitas perangkat lunak didefinisikan sebagai :

Konformansi terhadap kebutuhan fungsional dan kinerja yang dinyatakan secara eksplisit, standar perkembangan yang didokumentasikan secara eksplisit, dan karakteristik implisit yang diharapkan bagi semua perangkat lunak yang dikembangkan secara profesional.

Definisi tersebut berfungsi untuk menekankan tiga hal penting, yaitu :

- **Kebutuhan perangkat lunak merupakan fondasi yang melaluinya kualitas diukur. Kurangnya penyesuaian terhadap kebutuhan juga menunjukkan rendahnya kualitas.**
- **Standar yang telah ditentukan menetapkan serangkaian kriteria pengembangan yang menuntun cara perangkat lunak direayasa. Jika kriteria tersebut tidak diikuti, hampir pasti menimbulkan kualitas yang kurang baik.**
- **Ada serangkaian kebutuhan implisit yang sering tidak dicantumkan (misalnya kebutuhan akan kemampuan pemeliharaan yang baik). Bila perangkat lunak dapat berhasil menyesuaikan dengan kebutuhan eksplisitnya, tetapi gagal memenuhi kebutuhan implisitnya, maka kualitas perangkat lunak tersebut perlu diragukan.**

Aktivitas SQA

Jaminan kualitas perangkat lunak terdiri dari berbagai tugas yang berhubungan dengan dua konstituen yang berbeda, perekayasa perangkat lunak yang mengerjakan kerja teknis dan kelompok SQA yang bertanggung jawab terhadap perencanaan jaminan kualitas, kesalahan, penyimpanan rekaman, analisis, dan pelaporan.

Tugas kelompok SQA adalah membantu tim rekayasa perangkat lunak dalam pencapaian produk akhir yang berkualitas tinggi. The Software Engineering Institute merekomendasikan serangkaian aktivitas SQA yang menekankan rencana jaminan kualitas, kesalahan, penyimpanan rekaman, analisis, dan pelaporan.

Berikut ini aktivitas yang dilakukan (difasilitasi) oleh kelompok SQA yang independen :

1. Menyiapkan rencana SQA untuk suatu proyek.

Rencana itu dikembangkan selama perencanaan proyek dan dikaji oleh semua kelompok yang tertarik. Aktivitas jaminan kualitas yang dilakukan oleh tim rekayasa perangkat lunak dan kelompok SQA diatur oleh rencana.

Rencana tersebut mengidentifikasi hal-hal berikut :

- Evaluasi yang dilakukan
- Audit dan kajian yang dilakukan
- Standar yang dapat diaplikasikan pada proyek
- Prosedur untuk pelaporan dan penelusuran kesalahan
- Dokumen yang dihasilkan oleh kelompok SQA
- Jumlah umpan balik yang diberikan pada tim proyek perangkat lunak

2. Berpartisipasi dalam pengembangan deskripsi proses pengembangan proyek. Tim rekayasa perangkat lunak memilih sebuah proses bagi kerja yang akan dilakukan.

3. Mengkaji aktivitas rekayasa perangkat lunak untuk memverifikasi pemenuhan proses perangkat lunak yang sudah ditentukan. Kelompok SQA mengidentifikasi, mendokumentasi, dan menelusuri deviasi proses dan membuktikan apakah koreksi sudah dilakukan.

4. Mengaudit produk kerja perangkat lunak yang ditentukan untuk membuktikan kesesuaian dengan produk kerja yang ditentukan tersebut sebagai bagian dari proses perangkat lunak.

5. Memastikan bahwa deviasi pada kerja dan produk kerja perangkat lunak didokumentasi dan ditangani sesuai prosedur pendokumentasian.

6. Mencatat ketidak-sesuaian dan melaporkannya kepada manajemen senior. Item-item yang tidak sesuai ditelusuri sampai item itu diubah.

C. KAJIAN PERANGKAT LUNAK

Kajian perangkat lunak adalah suatu “filter” bagi proses rekayasa perangkat lunak, yaitu kajian yang diterapkan pada berbagai titik selama pengembangan perangkat lunak dan berfungsi untuk mencari kesalahan yang kemudian akan dihilangkan. Kajian perangkat lunak berfungsi untuk “memurnikan” produk kerja perangkat lunak yang terjadi sebagai hasil dari analisis, desain, dan pengkodean.

D. KAJIAN TEKNIK FORMAL

Kajian Teknik Formal (*Formal Technique Research*) adalah aktivitas jaminan kualitas perangkat lunak yang dilakukan oleh perancang perangkat lunak.

Tujuan FTR adalah :

- Menemukan kesalahan dalam fungsi, logika, atau implementasinya dalam berbagai representasi perangkat lunak.
- Membuktikan bahwa perangkat lunak di bawah kajian memenuhi syarat.
- Memastikan bahwa perangkat lunak disajikan sesuai dengan standar yang sudah ditentukan sebelumnya.
- Mencapai perangkat lunak yang dikembangkan dengan cara seragam.
- Membuat proyek lebih dapat dikelola.

Sebagai tambahan, FTR berfungsi sebagai dasar pelatihan yang memungkinkan perancang junior mengamati berbagai pendekatan yang berbeda terhadap analisis perangkat lunak, desain dan implementasi. FTR juga berfungsi untuk mengembangkan *backup* dan kontinuitas karena sejumlah orang mengenal baik bagian-bagian perangkat lunak yang tidak mereka ketahui sebelumnya.

E. RELIABILITAS PERANGKAT LUNAK

Tidak diragukan lagi bahwa reliabilitas sebuah program komputer merupakan suatu elemen yang penting. Bila sebuah program berkali-kali gagal untuk melakukan kinerja, maka sedikit meragukan apakah faktor kualitas perangkat lunak yang lain dapat diterima.

Reliabilitas perangkat lunak, tidak seperti faktor kualitas yang lain, dapat diukur, diarahkan, dan diestimasi dengan menggunakan data pengembangan historis. Reliabilitas perangkat lunak didefinisikan dalam bentuk statistik sebagai **"kemungkinan operasi program komputer bebas kegagalan di dalam suatu lingkungan tertentu dan waktu tertentu."**

Contoh :

Program X diperkirakan memiliki reliabilitas 0,96 pada delapan jam pemrosesan yang dilalui. Dengan kata lain, jika program X akan dieksekusi 100 kali dan membutuhkan delapan jam waktu pemrosesan yang dilalui (waktu eksekusi), dia akan beroperasi dengan benar (tanpa kegagalan) 96 kali dari 100 kali pelaksanaan.

Keamanan Perangkat Lunak dan Analisis Risiko

Keamanan perangkat lunak dan analisis risiko adalah aktivitas jaminan kualitas perangkat lunak yang berfokus pada identifikasi dan penilaian risiko potensial yang mungkin berpengaruh negatif terhadap perangkat lunak dan menyebabkan seluruh sistem menjadi gagal. Jika risiko dapat diidentifikasi pada awal proses rekayasa perangkat lunak, maka ciri-ciri desain perangkat lunak dapat ditetapkan sehingga akan mengeliminasi atau mengontrol risiko potensial.

F. STANDAR KUALITAS ISO 9000

Sistem jaminan kualitas dapat didefinisikan sebagai struktur, tanggung jawab, prosedur, proses, dan sumber-sumber daya organisasi untuk mengimplementasi manajemen kualitas.

ISO 9000 menjelaskan elemen jaminan kualitas dalam bentuk yang umum yang dapat diaplikasikan pada berbagai bisnis tanpa memandang produk dan jasa yang ditawarkan. Elemen-elemen tersebut mencakup struktur, prosedur, proses, organisasi, dan sumber daya yang dibutuhkan untuk mengimplementasi rencana kualitas, kontrol kualitas, jaminan kualitas, dan pengembangan kualitas.

Agar terdaftar dalam satu model sistem jaminan kualitas yang ada pada ISO 9000, sistem kualitas dan operasi perusahaan diperiksa oleh auditor untuk memeriksa kesesuaiannya dengan standar dan

operasi efektif. Bila registrasi itu berhasil, perusahaan diberi sertifikasi dari badan registrasi yang diwakili oleh auditor. Audit pengawasan tengah tahunan terus dilakukan untuk memastikan kesesuaiannya dengan standar yang sudah ditetapkan.

Standar ISO 9001

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, yaitu :

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 ketisaksesuaian 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.

G. RANGKUMAN

- Jaminan kualitas perangkat lunak merupakan “aktivitas pelindung” yang diaplikasikan pada setiap langkah dalam proses perangkat lunak. SQA mencakup berbagai prosedur untuk aplikasi yang efektif dari metode dan piranti, kajian teknik formal, strategi dan teknik pengujian, prosedur untuk mengontrol perubahan, prosedur untuk menjamin kesesuaian dengan standar yang ada, serta pengukuran dan mekanisme pelaporan.
- SQA kompleks karena kualitas perangkat lunak pun bersifat kompleks, atribut program komputer yang didefinisikan sebagai “penyesuaian dengan syarat yang ditetapkan secara eksplisit dan implisit.” Tetapi bila diperhatikan secara lebih umum, kualitas perangkat lunak mencakup banyak faktor produk dan proses yang berbeda serta metrik yang berhubungan.
- Kajian perangkat lunak merupakan salah satu aktivitas SQA yang terpenting. Kajian berfungsi sebagai filter bagi proses perangkat lunak, menghapus kesalahan bila biaya yang diperlukan untuk menemukan dan mengoreksi kesalahan relatif tidak mahal. Kajian teknik formal atau *walkthrough* adalah pertemuan kajian yang disesuaikan dengan kebutuhan yang terbukti sangat efektif untuk menemukan kesalahan.
- Untuk melakukan jaminan kualitas perangkat lunak secara tepat, maka data mengenai proses rekayasa perangkat lunak harus dikumpulkan, dievaluasi, dan disebar. SQA Statistik membantu meningkatkan kualitas produk dan proses perangkat lunak. Model reliabilitas perangkat lunak memperluas pengukuran, memungkinkan data cacat yang telah dikumpulkan diekstrapolasi ke dalam tingkat kegagalan yang diproyeksikan. Selain itu, memungkinkan juga untuk memprediksi reliabilitas.