

Jaminan Kualitas Perangkat Lunak

Disusun Oleh :

Irman Hariman, M.T

iirmanhariman@gmail.com

Software Quality Assurance [SQA]

Jaminan kualitas perangkat lunak merupakan aktivitas pelindung yang diaplikasikan pada seluruh proses perangkat lunak.

SQA meliputi :

1. Pendekatan manajemen kualitas
2. Teknologi rekayasa perangkat lunak yang efektif (metode dan peranti)
3. Kajian teknik formal yang diaplikasikan pada keseluruhan proses perangkat lunak
4. Strategi pengujian *multitiered* (deret bertingkat)
5. Kontrol dokumentasi perangkat lunak dan perubahan
6. Prosedur untuk menjamin kesesuaian dengan standar pengembangan perangkat lunak
7. Mekanisme pengukuran dan pelaporan.



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.

Konsep kunci kualitas kontrol adalah bahwa semua produk kerja memiliki spesifikasi yang telah ditentukan dan dapat diukur dimana kita dapat membandingkan output dari setiap proses. Kalang (loop) menjadi penting untuk meminimalkan cacat yang dihasilkan.



Jaminan Kualitas

Jaminan kualitas terdiri atas fungsi auditing dan pelaporan manajemen.

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



Biaya Kualitas

Menyangkut semua biaya yang diadakan untuk mengejar kualitas atau untuk menampilkan kualitas yang berhubungan dengan aktivitas. Hal ini 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 :

1. Pencegahan
2. Penilaian
3. Kegagalan.



- a) Biaya pencegahan meliputi :
- Perencanaan
 - Kajian teknis formal
 - Perlengkapan pengujian
 - Pelatihan
- b) Biaya penilaian meliputi :
- Inspeksi in-proses dan interproses
 - Pemeliharaan dan kalibrasi peralatan
 - Pengujian
- c) Biaya kegagalan
- Biaya kegagalan* adalah biaya yang akan hilang bila tidak ada cacat yang muncul sebelum produk disampaikan kepada pelanggan. 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

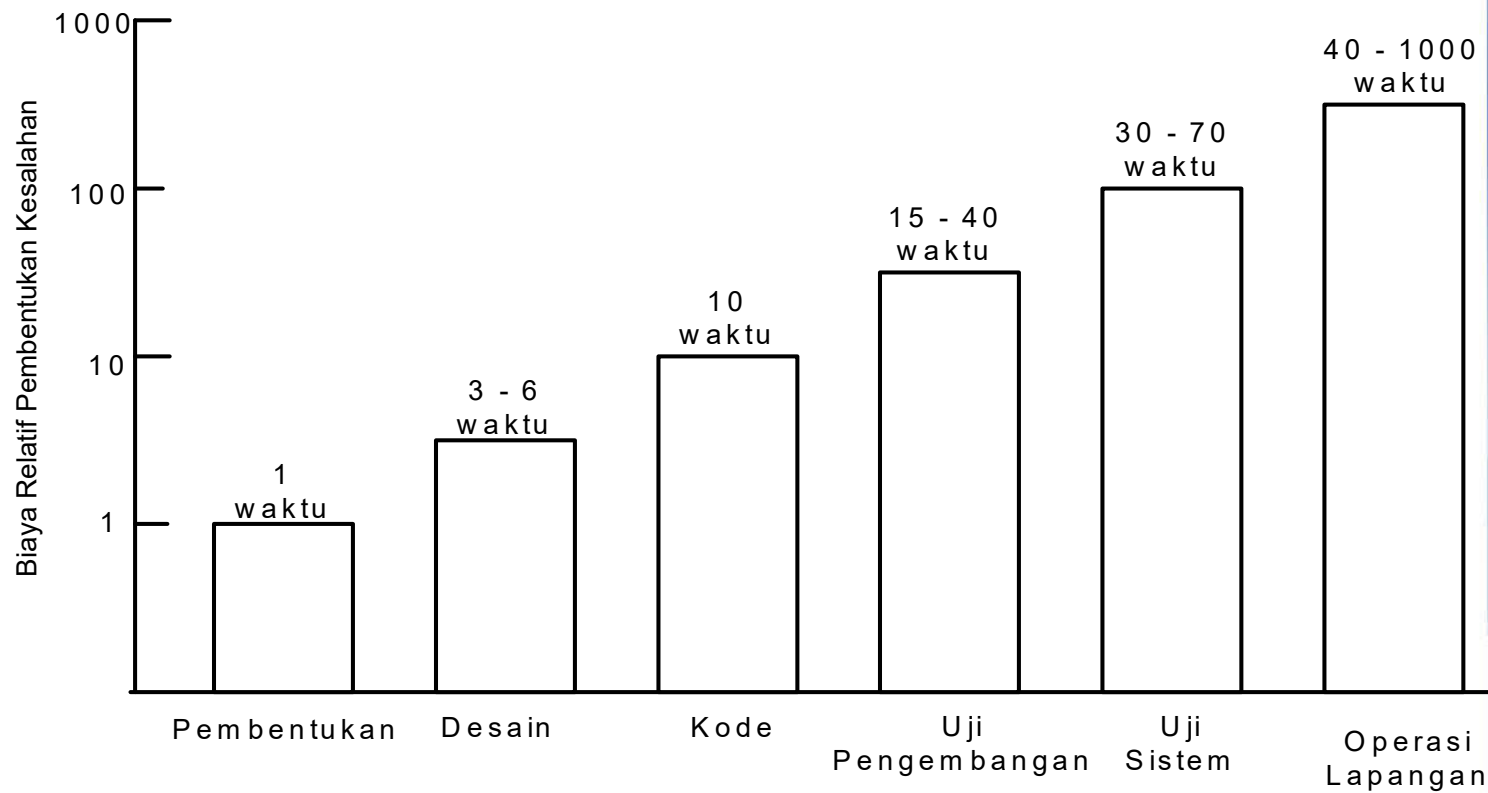
*Biaya **kegagalan eksternal*** adalah biaya yang berhubungan dengan cacat yang ditemukan setelah produk disampaikan kepada pelanggan.

Biaya kegagalan eksternal meliputi:

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

Biaya relatif mendapatkan dan membetulkan cacat bertambah secara dramatis pada saat melangkah dari pencegahan ke pendeteksian dan dari kegagalan internal ke kegagalan eksternal.





Gambar 1. *Biaya Relatif pembetulan kesalahan*

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 dikembangkan secara profesional. Definisi tersebut berfungsi untuk menekankan tiga hal penting, yaitu:

1. Kebutuhan perangkat lunak merupakan fondasi yang melaluinya *kualitas* diukur.
2. Standar yang telah ditentukan menetapkan serangkaian kriteria pengembangan yang menuntun cara perangkat lunak direkayasa.
3. Ada serangkaian *kebutuhan implisit* yang sering dicantumkan (misalnya kebutuhan akan kemampuan pemeliharaan yang baik).



Kelompok SQA berfungsi sebagai perwakilan *in-house* pelanggan, yaitu orang yang akan melakukan SQA harus memperhatikan perangkat lunak dari sudut pandang pelanggan.

Kelompok SQA harus dapat menjawab pertanyaan-pertanyaan dibawah ini untuk memastikan bahwa kualitas perangkat lunak benar-benar terjaga.

- Apakah perangkat lunak cukup memenuhi faktor kualitas
- Sudahkah pengembangan perangkat lunak dilakukan sesuai dengan standar yang telah ditetapkan sebelumnya?
- Sudahkah disiplin teknik dengan tepat memainkan perannya sebagi bagian dari aktivitas SQA?



Tugas kelompok SQA adalah membantu tim rekayasa perangkat lunak dalam pencapaian produk akhir yang berkualitas tinggi. Aktivitas yang dilakukan (atau difasilitasi) oleh kelompok SQA yang independen:

Menyiapkan rencana SQA untuk suatu proyek.

Rencana tersebut mengidentifikasi hal-hal berikut:

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



Aktivitas SQA

Jaminan kualitas perangkat lunak terdiri dari berbagai tugas yang berhubungan dengan dua konstituen yang berbeda :

- Perekraya perangkat lunak yang mengerjakan kerja teknis
- Kelompok SQA yang bertanggung jawab terhadap perencanaan jaminan kualitas, kesalahan, penyimpanan rekaman, analisis, dan pelaporan.



DEFINISI KUALITAS PERANGKAT LUNAK

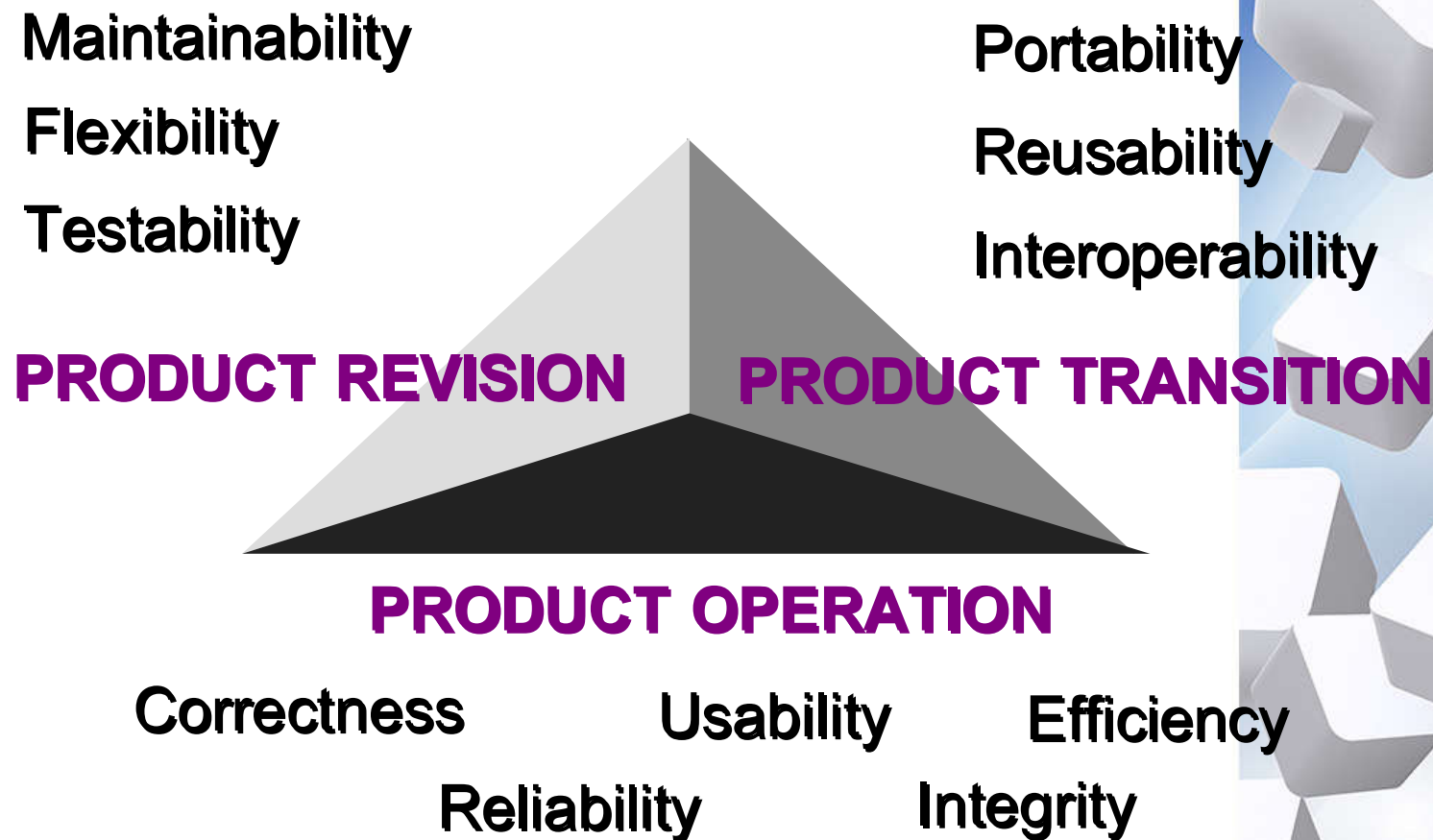
- 3 poin penting
 - Kebutuhan2 perangkat lunak merupakan fondasi darimana kualitas diukur
 - Standar2 spesifik yg menentukan kriteria pengembangan yg menuntun pembuatan suatu perangkat lunak
 - Terdapat kebutuhan2 yg implisit yg sering tidak diperhatikan (mis, keinginan untuk pemeliharaan yg terbaik)

FAKTOR KUALITAS PERANGKAT LUNAK

- Yg dapat dihitung secara langsung
 - *Error* (Kesalahan)
 - *Kilobytes Lines of Code* (KLOC)
- Dihitung secara tidak langsung
 - *Usability* (Kegunaan)
 - *Maintainability* (Pemeliharaan)



McCall's Triangle of Quality



FAKTOR KUALITAS ... (McCall)

- *Correctness* : besarnya program dapat memuaskan spesifikasi & objektivitas dari misi pelanggan
- *Reliability* : besarnya program dapat diharapkan memenuhi fungsi2 yg dikehendaki
- *Efficiency* : jumlah sumber2 & kode yg dibutuhkan program utk menjalankan fungsi2
- *Integrity* : besarnya pengontrolan pengaksesan oleh seseorang yg tidak mempunyai otorisasi terhadap perangkat lunak atau data
- *Usability* : effort (usaha) yg dibutuhkan utk mempelajari, mengoperasikan, menyiapkan input & menginterpretasi kan output program



FAKTOR KUALITAS ... (McCall) lanj.

- *Maintainability* : usaha yg dibutuhkan utk menempatkan & menetapkan suatu kesalahan pada program
- *Flexibility* : usaha yg dibutuhkan utk memodifikasi program yg dioperasikan
- *Testability* : usaha yg dibutuhkan utk menguji program utk menjamin tlh dijalankannya program yg diharapkan
- *Portability* : usaha yg dibutuhkan utk mentransfer program dari lingkungan sistem per.lunak &/ per.keras ke lingkungan lain
- *Reusability* : besarnya program dpt digunakan oleh aplikasi lain
- *Interoperability* : usaha yg dibutuhkan utk memasang-kan satu sistem dgn yg lain

PENGUKURAN KUALITAS PER. LUNAK

- *Auditability* : mudah utk dicek mengenai konfirmasi standar
- *Accuracy* : presisi komputasi & pengontrolan
- *Communication commonality* : derajat penggunaan interface, protokol & bandwidth yg standar
- *Completeness* : derajat pencapaian implementasi full dari fungsi2 yg dibutuhkan
- *Conciseness* : kepadatan program dalam *lines of code*
- *Consistency* : penggunaan teknik dokumentasi & perancangan yg seragam
- *Data commonality* : penggunaan struktur & tipe data standar
- *Error tolerance* : akibat yg timbul pada saat program menemui kesalahan

PENGUKURAN KUALITAS PER. LUNAK (lanj.)

- *Execution efficiency* : kinerja waktu eksekusi pada program
- *Expandability* : derajat dimana perancangan terprosedur, data & arsitektur dapat diperluas
- *Generality* : kelonggaran aplikasi dari komponen program
- *Hardware independence* : derajat dimana per. Lunak dipisahkan dari per. keras atau yg mengoperasikannya
- *Instrumentation* : derajat dimana program memonitor operasinya sendiri & mengindentifikasikan kesalahan2 yg timbul
- *Modularity* : kemandirian fungsional dari komponen program
- *Operability* : kemudahan pengoperasian program
- *Security* : ketersediaan mekanisme yg mengontrol atau memproteksi program & data

PENGUKURAN KUALITAS PER. LUNAK (lanj.)

- *Self-documentation* : derajat dimana *source code* menyediakan dokumentasi yg berarti
- *Simplicity* : derajat dimana program dapat dimengerti dengan mudah
- *Software system independence* : derajat dimana program berdiri sendiri dari fitur bhs pemrograman, karakteristik sistem pengoperasian & batasan lainnya yg tdk standar
- *Traceability* : kemampuan utk menelusuri representasi perancangan atau komponen program aktual, kembali ke kebutuhan
- *Training* : derajat dimana per. lunak dapat membantu pengguna yg baru dalam mengaplikasikan sistem



BIAYA KUALITAS PERANGKAT LUNAK

- Semua biaya utk mengejar kualitas atau utk menampilkan kualitas yg berhubungan dgn aktifitas.
 - Biaya pencegahan
 - Perencanaan kualitas
 - Review teknis formal
 - Perlengkapan pengujian
 - Pelatihan



BIAYA KUALITAS PERANGKAT LUNAK (lanj.)

- Biaya penilaian meliputi aktifitas memperoleh wawasan mengenai kondisi produk pertama kali pada masing2 proses, meliputi
 - Inspeksi in-proses & interproses
 - Pemeliharaan & kalibrasi peralatan
 - Pengujian
- Biaya kegagalan adalah biaya yg akan hilang bila tidak ada cacat yg muncul sebelum produk disampaikan kepada pelanggan



BIAYA KUALITAS PERANGKAT LUNAK (lanj.)

- Biaya kegagalan internal adl biaya yg diadakan bila kita mendeteksi kesalahan dlm produk sebelum produk dipasarkan, meliputi
 - Pengerjaan kembali
 - Perbaikan
 - Analisis mode kegagalan
- Biaya kegagalan eksternal adl biaya yg berhubungan dg cacat yg ditemukan setelah produk disampaikan pada pelanggan, meliputi
 - Resolusi keluhan
 - Penggantian & pengembalian produk
 - Dukungan *help line*
 - Kerja jaminan



SQA

- Merupakan kegiatan yg terpola secara sistematis dan terencana, yg dibutuhkan utk menjamin kualitas suatu perangkat lunak.
- Terdiri atas 7 aktifitas utama :
 - Aplikasi metode secara teknis
 - Review teknis formal
 - Pengujian perangkat lunak
 - Penekanan pada standar
 - Pengontrolan pada perubahan
 - Pengukuran
 - Penyimpanan dan pelaporan



SQA (lanj.)

- SQA dimulai dgn sekumpulan alat & metode teknis yg membantu analis utk mendapatkan spesifikasi yg berkualitas tinggi & bagi perancang utk merancang dg kualitas tinggi
- Setelah spesifikasi & desain dibuat, ditetapkan kualitasnya dg melakukan review teknis formal.
- Pengujian per. Lunak mengkombinasikan langkah2 strategi dg metode rancangan test-case yg dpt menjamin pendeteksian kesalahan secara efektif.
- Jika terdapat standar yg formal, berarti harus dapat dijamin bahwa standar tsb diikuti.



SQA (lanj.)

- Pengontrolan perubahan dilakukan selama pembuatan per. Lunak & pada tahap pemeliharaan. Setiap perubahan dpt menyebabkan kesalahan & efek lain yg akan menyebabkan kesalahan jg.
- Pengukuran terhadap per. Lunak mencakup pengukuran secara manajemen & teknis
- Penyimpanan hasil dari review, audit, pengontrolan perubahan, pengujian dyl sebagai bagian dari record historis utk suatu proyek dan didesiminasikan kpd para staf pengembangan sbg 'dasar utk mereka ketahui'



REVIEW PERANGKAT LUNAK

- Merupakan filter pada proses pembuatan perangkat lunak.
- Bentuknya : presentasi formal di depan pelanggan, manajemen, & staf teknisi.
- Review – suatu cara penggunaan perbedaan2 dari sekompak orang untuk
 - Menentukan peningkatan kebutuhan produk dari seseorang atau tim
 - Konfirmasi setiap bagian dari produk dimana peningkatan tsb diinginkan atau tidak diinginkan
 - Mencapai kerja teknis yg seragam, ataupun dapat lebih diprediksi, daripada tanpa review



PERTEMUAN PADA REVIEW TEKNIS FORMAL

- Batasannya
 - Antara 3 & 5 orang yg terlibat : pemimpin review, pereview & produsen
 - Persiapan sebelumnya tidak lebih 2 jam kerja per orang
 - Lama pertemuan review minimal 2 jam
- Fokus : produk, komponen program (spesifikasi kebutuhan, perancangan modul detail, listing coding utk setiap modul)
- Akhir review, harus diputuskan
 - Menerima produk tanpa modifikasi
 - Menolak produk krn kesalahan yg fatal
 - Menerima produk dg kesalahan yg kecil dan harus diperbaiki

PENDOKUMENTASIAN PADA REVIEW TEKNIS FORMAL

- Laporan review harus dapat menjawab
 - Apa yg direview
 - Siapa yg mereview
 - Apa yg ditemukan & disimpulkan
- Daftar review mempunyai 2 tujuan :
 - Mengidentifikasi area permasalahan produk
 - Menyediakan checklist setiap item kegiatan yg dapat memandu produsen utk melakukan perbaikan



PEDOMAN REVIEW TEKNIS FORMAL

- Mereview produk bukan produsen
- Membuat agenda dan mengikutinya
- Membatasi debat
- Memberitahukan area masalah, tetapi bukan utk menyelesaikan semua masalah yg ada
- Membuat catatan tertulis (di papan/bisa dilihat)
- Membatasi jumlah partisipan & menekankan persiapan awal
- Membuat checklist utk setiap produk yg direview
- Mengalokasikan sumber daya & waktu yg terjadual utk review berikutnya
- Mengadakan pelatihan utk semua pereview
- Mereview produk awal terlebih dulu (mis. Panduan review)



