

Testing dan Implementasi Sistem

Pendahuluan

Materi:

Definisi Testing

Definisi Sederhana Kualitas

Hubungan Testing dan Kualitas

Faktor Kualitas secara Umum

Kualitas Software Penting bagi Organisasi Software

Definisi Testing

Menurut Hetzel 1973: Testing adalah proses pemantapan kepercayaan akan kinerja program atau sistem sebagaimana yang diharapkan.

Menurut Myers 1979: Testing adalah proses eksekusi program atau sistem secara intens untuk menemukan *error*.

Menurut Hetzel 1983 (Revisi): Testing adalah tiap aktivitas yang digunakan untuk dapat melakukan evaluasi suatu atribut atau kemampuan dari program atau sistem dan menentukan apakah telah memenuhi kebutuhan atau hasil yang diharapkan.

Menurut Standar ANSI/IEEE 1059: Testing adalah proses menganalisa suatu entitas *software* untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan (*defects / errors / bugs*) dan mengevaluasi fitur-fitur dari entitas *software*.

Pandangan Praktisi Tentang Testing

Melakukan cek pada program terhadap spesifikasi.

- ☑ Menemukan *bug* pada program.
- ☑ Menentukan penerimaan dari pengguna.
- ☑ Memastikan suatu sistem siap digunakan.
- ☑ Meningkatkan kepercayaan terhadap kinerja program.
- ☑ Memerlihatkan bahwa program berkerja dengan benar.
- ☑ Membuktikan bahwa *error* tidak terjadi.
- ☑ Mengetahui akan keterbatasan sistem.
- ☑ Mempelajari apa yang tak dapat dilakukan oleh sistem.
- ☑ Melakukan evaluasi kemampuan sistem.
- ☑ Verifikasi dokumen.
- ☑ Memastikan bahwa pekerjaan telah diselesaikan.

Testing *software*

Testing *software* adalah proses mengoperasikan *software* dalam suatu kondisi yang di kendalikan, untuk **verifikasi** apakah telah berlaku sebagaimana telah ditetapkan (menurut spesifikasi), mend**eteksi** *error*, dan **validasi** apakah spesifikasi yang telah ditetapkan sudah memenuhi keinginan atau kebutuhan dari pengguna yang sebenarnya

Verifikasi

Verifikasi adalah pengecekan atau pengetesan entitas-entitas, termasuk *software*, untuk pemenuhan dan konsistensi dengan melakukan evaluasi hasil terhadap kebutuhan yang telah ditetapkan.

(Are we building the system right ?)

Deteksi *error*

Deteksi *error*: Testing seharusnya berorientasi untuk membuat kesalahan secara intensif, untuk menentukan apakah suatu hal tersebut terjadi bilamana tidak seharusnya terjadi atau suatu hal tersebut tidak terjadi dimana seharusnya mereka ada

Validasi

Validasi melihat kebenaran sistem, apakah proses yang telah ditulis dalam spesifikasi adalah apa yang sebenarnya diinginkan atau dibutuhkan oleh pengguna. (*Are we building the right system?*)

Tujuan Akhirnya Testing

Apakah *software* telah siap digunakan?

Apa saja risikonya?

Apa saja kemampuannya?

Apa saja keterbatasannya?

Apa saja masalahnya?

Apakah telah berlaku seperti yang diharapkan?

Definisi Sederhana Kualitas

CROSBY: Kualitas adalah pemenuhan terhadap kebutuhan.

ISO-8402: Kualitas adalah keseluruhan dari fitur yang menjadikan produk dapat memuaskan atau dipakai sesuai kebutuhan dengan harga yang terjangkau.

W.E. Perry: Kualitas adalah pemenuhan terhadap standar.

R. Glass: Kualitas adalah tingkat kesempurnaan.

J. Juran: Kualitas adalah tepat guna.

Hubungan Testing dan Kualitas

Definisi *software* berkualitas adalah *software* yang bebas *error* dan *bug* secara obyektif, tepat waktu dan dana, sesuai dengan kebutuhan atau keinginan dan dapat dirawat (*maintainable*).

Faktor Kualitas secara Umum

Fungsionalitas (Kualitas Luar)

Rekayasa (Kualitas Dalam)

Adaptabilitas (Kualitas ke Depan)

Fungsionalitas (Kualitas Luar)

Kebenaran (*Correctness*)

Reliabilitas (*Reliability*)

Kegunaan (*Usability*)

Integritas (*Integrity*)

Rekayasa (Kualitas Dalam)

Efisiensi (*Efficiency*)

Testabilitas (*Testability*)

Dokumentasi (*Documentation*)

Struktur (*Structure*)

Adaptabilitas (Kualitas ke Depan)

Fleksibilitas (*Flexibility*)

Reusabilitas (*Reusability*)

Maintainabilitas (*Maintainability*)

Kualitas *Software* Penting bagi Organisasi *Software*

Kualitas *software* dipengaruhi oleh biaya dan jadwal disebabkan kemampuan rekayasa *software* dari pihak pengembang yang tak mencukupi, dan kemampuan pelanggan yang sangat kurang (bahkan tak mampu) untuk memberikan spesifikasi kebutuhan dari sistem.

Organisasi *software harus* melakukan proses analisa, evaluasi dan pengembangan yang berkesinambungan agar *software* yang semakin efektif, efisien, terukur, terkendali dan dapat diulang secara konsisten dalam menghasilkan suatu produk (*software*) yang berkualitas, tepat waktu dan pendanaan.