



TESTING DAN IMPLEMENTASI

PERTEMUAN 9

JENIS TESTING SOFTWARE QUALITY ASSURANCE (SQA)

INTRO

Software Quality Assurance adalah suatu proses sistematis untuk memeriksa apakah sebuah software telah dikembangkan sesuai dengan kebutuhan yang telah ditentukan sebelumnya.

Proses ini, bisa dilaksanakan oleh seorang *QA Tester* atau oleh seorang *QA Engineer*.

INTRO

Seorang **Quality Assurance/Tester** bertugas **mempersiapkan** kasus pengujian untuk setiap **case** dan **skenarionya** untuk menyelesaikan aktivitas persyaratan suatu sistem. Banyak kasus uji yang mewakili berbagai macam situasi, baik yang **normal maupun tidak normal** (kejadian tertentu) yang harus disiapkan untuk masing - masing skenario.



8 JENIS TESTING WAJIB SEORANG SOFTWARE TESTER / SOFTWARE QA:

1. Performance Testing
2. System Testing
3. Unit Testing
4. Integration Testing
5. Usability Testing
6. Smoke Testing
7. Stress Testing
8. User Acceptance Test (UAT)

1. PERFORMANCE TESTING

Performance test adalah integration dan usability test (tes kegunaan), yang menentukan apakah system or subsystem dapat memenuhi kriteria kinerja berbasis waktu seperti response time atau throughput (hasil).

- Response time menentukan batas waktu maksimum/delay waktu normal yang diijinkan dari respon suatu software saat terjadi transaksi.
- Throughput (hasil) menentukan hasil transaksi yang harus diproses per menit atau per jam.

2. SYSTEM TESTING

System test adalah **integration test** dari **behavior** (tingkah laku) seluruh sistem atau **independent subsystem**.

System testing biasanya dilakukan pertama kali oleh **pengembang** atau personil pengujian untuk memastikan bahwa keseluruhan sistem telah berfungsi dan bahwa sistem telah memenuhi persyaratan pengguna (user requirement).

Jadi test ini biasanya dilakukan setiap kali melakukan penambahan menu/submenu/files baru dsb

3. UNIT TESTING

Unit testing adalah proses metode **pengujian individual, class**, atau komponen **sebelum mereka terintegrasi dengan perangkat lunak lainnya**.

Tujuan dari unit testing adalah untuk mengidentifikasi dan memperbaiki kesalahan sebanyak mungkin sebelum modul - modul digabungkan menjadi unit perangkat lunak yang lebih besar, seperti program, class dan subsistem.

Kelemahan dari test ini adalah Kesalahan menjadi lebih sulit ditemukan dan diperbaiki ketika banyak unit telah digabungkan.

Jadi test ini biasanya dilakukan saat penggabungan dengan **unit lain, class lain, komponen lain, fitur lain, tools lain, aplikasi lain, subsistem lain dsb**

4. INTEGRATION TESTING

Integration test adalah **mengevaluasi behavior** (tingkah laku) dari **kelompok method atau class**.

Tujuan dari integration test adalah untuk mengidentifikasi kesalahan yang tidak dapat dideteksi oleh unit testing. Kesalahan tersebut mungkin disebabkan oleh beberapa masalah, diantaranya:

- **Interface incompatibility**, misalnya terjadi kesalahan dalam menentukan tipe data ke method lainnya → (karena terjadi penggabungan fitur/subsistem maka terjadi kerancuan dalam tipe data, fields yang sama tetapi beda tipe data)
- **Parameter values**, misalnya terjadi perhitungan nilai yang salah atau tidak terduga seperti nomor negatif untuk value/harga tertentu → (stok barang bisa minus)

Jadi test ini melakukan **evaluasi Kembali dari unit testing yang telah dilakukan sebelumnya**

5. USABILITY TESTING

Usability test merupakan testing yang paling umum digunakan, yaitu untuk menentukan apakah system telah **sesuai dengan kebutuhan pengguna**/user atau memenuhi persyaratan pengguna.

Oleh karena banyaknya tipe persyaratan system, baik yang **fungsional (manfaat langsung)** maupun **non-fungsional (tidak langsung)**, maka banyak tipe dari usability test yang harus dilakukan di waktu yang berbeda.

Tujuan utama dari usability test adalah **mengevaluasi persyaratan fungsional dan kualitas dari user interface**. Pengujian ini sering dilakukan untuk mendapatkan feedback yang cepat dalam meningkatkan interface dan mengoreksi kesalahan dalam komponen perangkat lunak.

User berinteraksi dengan sistem untuk menentukan apakah :

- **Sistem telah berfungsi sebagaimana yang diharapkan** dan
- **apakah user interface membuat sistem dapat mudah digunakan.**

6. SMOKE TESTING

Smoke testing adalah system test yang biasanya dilakukan **setiap hari** atau beberapa kali **per minggu**. Build and smoke test sangat penting karena menyediakan **feedback** yang cepat dalam masalah yang signifikan.

Smoke testing hanya menentukan masalah penjadwalan dalam melakukan testing system. Tujuan utama dalam smoke testing adalah **feedback dari user**

7. STRESS TESTING

Stress testing juga dikenal sebagai pengujian daya tahan untuk menentukan sampai dimana batas dari perangkat lunak atau perangkat keras. Hal ini juga untuk memastikan sistem berjalan efektif meskipun dalam **kondisi ekstrem**.

Tujuan dari Stress testing adalah untuk menguji **stabilitas** dan **keandalan** sistem

Misalnya untuk mengetahui sekuat apa server website saat menampung visitor (pengunjung) dalam website tersebut, dengan cara melakukan hit dummy ke website tersebut, misalnya dengan melakukan kunjungan ke satu website secara bersamaan, menambah jumlah user sebanyak mungkin dalam satu waktu, dsb

8. USER ACCEPTANCE TEST (UAT)

User acceptance test digunakan untuk menentukan apakah sistem yang dikembangkan telah memenuhi kebutuhan pengguna. Dalam beberapa proyek, acceptance testing dilakukan pada **putaran terakhir proses pengujian** yaitu **sebelum sistem diserahkan kepada user**.

Acceptance Testing biasanya dilakukan setelah rangkaian testing seperti Unit Testing, Integration Testing, dan System Testing selesai dan menggunakan **metode Black Box Testing**, dengan menggunakan dokumen test case untuk dipresentasikan di akhir ke user / client.