

Nama : Fadil Febriansyah

NPM : 1204029

Kelas : 3B

Matkul : Kapita Selekt

Rangkuman Brainmatics 1 Hour Online Training: 3 Alasan Dunia Tidak Butuh Software Testing

1. Kegagalan project software
 - Dibatalkan sebelum selesai
 - Selesai tapi tidak pernah dipakai
 - Tidak bermanfaat bagi pengguna
 - Tidak sesuai dengan keinginan pengguna

2. Keunikan dari software

Karakteristik	Software	Hardware
Komplektisitas	Tingkat kompleksitas dari produk software tinggi, dengan kemungkinan perubahan parameter dan fungsi yang sangat beragam	Tingkat kompleksitas produk lain rendah, dengan kemungkinan perubahan parameter dan fungsi tidak beragam
SDM	Kuantitas SDM Tidak berhubungan dengan Kualitas dan Kecepatan Kerja.	Kuantitas SDM berhubungan dengan Kualitas dan Kecepatan Kerja
Visibilitas produk	Produk tidak terlihat dengan kasat mata, termasuk bila ada cacat (defect) dari produk	Produk terlihat dengan kasat mata, termasuk bila ada cacat (defect) dari produk

3. Kesalahan Perangkat Lunak vs Kesalahan vs Kegagalan
Software errors

- Kesalahan tata bahasa dalam baris kode
- Kesalahan logis dalam menjalankan kebutuhan klien

Software Faults

- Fungsi Perangkat Lunak yang tidak benar di melaksanakan umum atau khusus aplikasi

Software failures

- Terjadi ketika kesalahan perangkat lunak diaktifkan

4. Stages of testing

a. Unit testing

- Blackbox testing
Hanya melihat input dan output dan menguji apakah unit memenuhi persyaratan yang dinyatakan dalam spesifikasi.
- Whitebox testing

Kegunaan terbatas dalam desain karena unitnya sangat kecil.

b. Integration testing

Empat jenis pengujian Integrasi:

1. Pengujian antarmuka pengguna

- Menguji setiap fungsi antarmuka
- Menelusuri setiap menu/layar

2. Pengujian kasus penggunaan

- Memastikan bahwa setiap kasus penggunaan bekerja dengan benar
- Telusuri setiap kasus penggunaan

3. Pengujian interaksi

- Mulailah dengan sebuah paket
- Setiap metode adalah sebuah rintisan
- Tambahkan metode satu per satu, uji sambil jalan
- Setelah semua paket selesai, ulangi pada level paket

4. Pengujian antarmuka sistem

- Memastikan transfer data antar sistem

c. System testing

Lima jenis pengujian sistem:

1. Pengujian Persyaratan

- Memastikan bahwa integrasi tidak menimbulkan kesalahan baru

2. Pengujian Kegunaan

- Menguji seberapa mudah dan bebas kesalahan sistem yang digunakan
- Informal atau formal

3. Pengujian Keamanan

- Memastikan bahwa fungsi keamanan ditangani dengan baik

4. Pengujian Kinerja

- Memastikan bahwa sistem bekerja di bawah volume aktivitas yang tinggi

5. Pengujian Dokumentasi

- Analis memeriksa bahwa dokumentasi dan contoh berfungsi dengan baik

d. Acceptance testing

Dua jenis pengujian:

1. Pengujian Alfa

- Ulangi tes oleh pengguna untuk memastikan mereka menerima sistem, menggunakan data yang diketahui

2. Pengujian Beta

- Menggunakan data nyata, bukan data uji

5. Seven Testing Principles

- Testing shows presence of defects
- Exhaustive testing is impossible
- Early testing
- Defect clustering
- Pesticide paradox
- Testing is context dependent
- Absence-of-errors fallacy

6. Software testing

- Test planning

- Test monitoring and control
 - Test analysis
 - Test design
 - Test implementation
 - Test execution
 - Test completion
7. **Functional testing:** Pengujian fungsional adalah jenis pengujian perangkat lunak yang berfokus pada verifikasi bahwa perangkat lunak berfungsi sebagaimana dimaksud dan memenuhi persyaratan fungsional. Pengujian fungsional dapat dilakukan secara manual atau melalui alat pengujian otomatis. Tujuan pengujian fungsional adalah untuk memastikan bahwa aplikasi perangkat lunak memenuhi persyaratan pengguna akhir dan menjalankan semua fungsi dengan benar. Contoh dari functional testing adalah pengujian fitur login pada sebuah aplikasi. Pengujian dilakukan dengan cara memasukkan username dan password yang valid dan memastikan bahwa pengguna berhasil login ke aplikasi.
 8. **Non-functional testing:** Pengujian non-fungsional adalah jenis pengujian perangkat lunak yang berfokus pada verifikasi persyaratan perangkat lunak non-fungsional, seperti kinerja, skalabilitas, keandalan, dan kegunaan. Pengujian non-fungsional dapat dilakukan secara manual atau melalui alat pengujian otomatis. Tujuan pengujian non-fungsional adalah untuk memastikan bahwa aplikasi perangkat lunak memenuhi persyaratan kinerja dan kegunaan dari pengguna akhir. Contoh dari non-functional testing adalah pengujian kinerja (performance testing) pada sebuah aplikasi. Pengujian dilakukan dengan cara memastikan bahwa aplikasi mampu menangani beban yang tinggi dan memproses data dalam waktu yang cepat.
 9. **Structural testing:** Pengujian struktural adalah jenis pengujian perangkat lunak yang berfokus pada verifikasi struktur internal aplikasi perangkat lunak, seperti cakupan kode, jalur kode, dan aliran kontrol. Pengujian struktural juga dikenal sebagai pengujian kotak putih karena melibatkan pengujian struktur internal perangkat lunak. Pengujian struktural dapat dilakukan secara manual atau melalui alat pengujian otomatis. Tujuan pengujian struktural adalah untuk memastikan bahwa aplikasi perangkat lunak memenuhi standar desain dan pengkodean. Contoh dari structural testing adalah pengujian kode sumber pada sebuah aplikasi. Pengujian dilakukan dengan cara memastikan bahwa setiap baris kode telah diuji dan diperiksa, serta memastikan bahwa semua kondisi telah dipertimbangkan dan diuji.
 10. **Testing related to Change:** Pengujian terkait perubahan adalah jenis pengujian perangkat lunak yang berfokus pada verifikasi perubahan yang dilakukan pada aplikasi perangkat lunak, seperti fitur baru, perbaikan bug, dan penyempurnaan. Pengujian terkait perubahan dapat dilakukan secara manual atau melalui alat pengujian otomatis. Tujuan pengujian terkait perubahan adalah untuk memastikan bahwa aplikasi perangkat lunak memenuhi persyaratan pengguna akhir dan melakukan semua fungsi dengan benar setelah perubahan dilakukan. Pengujian terkait dengan perubahan juga dikenal sebagai pengujian regresi karena melibatkan pengujian aplikasi perangkat lunak setelah perubahan dilakukan. Contoh dari testing related to change adalah pengujian setelah melakukan perubahan pada fitur pada sebuah aplikasi. Misalnya, pengujian setelah melakukan perbaikan bug pada fitur login, untuk memastikan bahwa setelah perubahan, fitur tersebut masih berfungsi dengan baik.
 11. **Intisari dari pelatihan 1 Hour Online Training: 3 Alasan Dunia Tidak Butuh Software Testing**
 Pengujian perangkat lunak memiliki beberapa jenis, yaitu pengujian fungsional, pengujian non-fungsional, pengujian struktural, dan pengujian terkait perubahan. Setiap jenis pengujian memiliki tujuan dan fokus yang berbeda, dan dapat dilakukan secara manual atau melalui alat pengujian otomatis. Pengujian fungsional berfokus pada verifikasi bahwa perangkat lunak

berfungsi sebagaimana dimaksud dan memenuhi persyaratan fungsional, sedangkan pengujian non-fungsional berfokus pada verifikasi persyaratan perangkat lunak non-fungsional seperti kinerja, skalabilitas, keandalan, dan kegunaan. Pengujian struktural berfokus pada verifikasi struktur internal aplikasi perangkat lunak, seperti cakupan kode, jalur kode, dan aliran kontrol. Pengujian terkait perubahan berfokus pada verifikasi perubahan yang dilakukan pada aplikasi perangkat lunak seperti fitur baru, perbaikan bug, dan penyempurnaan. Tujuan dari setiap jenis pengujian adalah untuk memastikan bahwa aplikasi perangkat lunak memenuhi persyaratan pengguna akhir dan menjalankan semua fungsi dengan benar. Contoh dari setiap jenis pengujian diberikan sebagai pengujian fitur login, pengujian kinerja, pengujian kode sumber, dan pengujian setelah melakukan perubahan pada fitur.