3 Alasan Dunia Tidak Butuh Software Testing

KAPITA SELEKTA

Disusun Oleh

1204041 BRYAN SAMPERURA

Program Studi Diploma IV Teknik Informatika

Universitas Logistik Dan Bisnis Internasional

2023

Pemahaman 7 prinsip testing, Functional testing, Non Functional testing, Structural testing dan Testing related to Change

1. Pemahaman 7 prinsip testing: Tujuh prinsip dasar dalam testing adalah:

- Testing shows the presence of defects (Testing menunjukkan keberadaan defect)
- Exhaustive testing is impossible (Pengujian secara menyeluruh tidak mungkin dilakukan)
- Early testing saves time and money (Pengujian awal dapat menghemat waktu dan uang)
- Defect clustering occurs (Terjadi penumpukan defect)
- Testing is context dependent (Pengujian bergantung pada konteks)
- Absence-of-errors fallacy (Kesalahan penyangkalan kesalahan)
- Pesticide paradox (Paradoks pestisida)

2. Functional testing:

Functional testing adalah proses pengujian perangkat lunak untuk memastikan bahwa setiap fitur dan fungsionalitas dari aplikasi berfungsi sebagaimana mestinya sesuai dengan persyaratan bisnis dan kebutuhan pengguna. Tujuan dari pengujian fungsional adalah memastikan bahwa perangkat lunak dapat melakukan apa yang seharusnya dilakukan dengan benar.

3. Non-Functional testing:

Non-functional testing meliputi pengujian performa, keamanan, keterandalan, keamanan, ketersediaan, dan skala perangkat lunak. Pengujian non-fungsional membantu menentukan seberapa baik perangkat lunak dapat menangani beban kerja, seberapa cepat aplikasi berjalan, dan seberapa kuat perangkat lunak dapat mencegah ancaman keamanan.

4. Structural testing:

Structural testing adalah jenis pengujian perangkat lunak yang dilakukan untuk memastikan bahwa kode program yang ditulis bekerja dengan benar. Pengujian

struktural melibatkan pengujian metrik program seperti kode sumber, objek, dan bahasa pemrograman lainnya.

5. Testing related to Change:

Testing yang berkaitan dengan perubahan dilakukan untuk memastikan bahwa perubahan yang diterapkan pada sistem tidak mengganggu fungsionalitas atau kinerja sistem secara keseluruhan. Pengujian ini termasuk pengujian integrasi, pengujian regresi, dan pengujian ketergantungan.

6. Pengujian regresi:

Pengujian regresi adalah jenis pengujian perangkat lunak yang dilakukan untuk memastikan bahwa perubahan apa pun pada aplikasi tidak mengganggu fungsionalitas atau kinerja sistem secara keseluruhan.

7. Pengujian integrasi:

Pengujian integrasi adalah jenis pengujian perangkat lunak yang dilakukan untuk memastikan bahwa berbagai komponen dari sistem dapat bekerja bersama dengan baik dan berfungsi sebagaimana mestinya. Tujuan pengujian integrasi adalah untuk memastikan bahwa sistem dapat berinteraksi dengan sistem eksternal dengan benar dan bahwa aplikasi berfungsi secara keseluruhan.

Contoh Functional testing, Non Functional testing, Structural testing dan Testing related to Change

Berikut adalah contoh dari beberapa jenis pengujian perangkat lunak:

1. Contoh Functional Testing:

a. Pengujian Fungsional pada Aplikasi E-Commerce: Memastikan bahwa fitur seperti pendaftaran, login, pengecekan harga, menambahkan produk ke keranjang belanja, checkout, dan pembayaran bekerja dengan benar dan sesuai dengan persyaratan bisnis dan kebutuhan pengguna.

b. Pengujian Fungsional pada Aplikasi Mobile Banking: Memastikan bahwa fitur seperti login, pengecekan saldo, transfer uang, pembayaran tagihan, dan pengaturan

akun bekerja dengan benar dan sesuai dengan persyaratan bisnis dan kebutuhan pengguna.

2. Contoh Non-Functional Testing:
a. Pengujian Performa pada Aplikasi Web: Memastikan bahwa aplikasi dapat menangani beban kerja yang tinggi dengan baik dan menghadapi permintaan yang tinggi.
b. Pengujian Keamanan pada Aplikasi Perbankan: Memastikan bahwa data pengguna dan informasi keuangan terlindungi dengan baik dari serangan dan ancaman keamanan.

3. Contoh Structural Testing:
a. Pengujian Unit pada Kode Program: Memastikan bahwa setiap unit atau bagian dari kode program berfungsi sebagaimana mestinya.
b. Pengujian Integrasi pada Sistem: Memastikan bahwa berbagai komponen dari sistem bekerja bersama dengan baik dan berfungsi sebagaimana mestinya.

4. Contoh Testing Related to Change:
a. Pengujian Regresi pada Perubahan Aplikasi: Memastikan bahwa perubahan yang diterapkan pada aplikasi tidak mengganggu fungsionalitas atau kinerja sistem secara keseluruhan.
b. Pengujian Integrasi pada Perubahan Sistem: Memastikan bahwa perubahan yang diterapkan pada sistem tidak mengganggu fungsionalitas atau kinerja sistem secara keseluruhan.

Apa intisari dari pelatihan tersebut?

Terkadang tim pengembang tidak melakukan pengujian perangkat lunak karena beberapa alasan, salah satunya adalah karena pengujian dapat memakan waktu dan biaya tambahan. Ada tekanan untuk menyelesaikan proyek dalam waktu yang singkat dengan biaya minimal, sehingga proses pengujian yang memakan waktu dan biaya tambahan dapat menjadi penghalang dalam mencapai tujuan tersebut. Keterbatasan sumber daya seperti SDM dan peralatan pengujian yang tidak memadai juga dapat menjadi faktor yang menghalangi penggunaan software testing secara sistematis.

Namun, keyakinan berlebihan pada pengalaman proyek sebelumnya juga dapat menjadi alasan tim pengembang tidak menggunakan software testing secara terstruktur.

Meskipun demikian, software testing sangat penting dalam pengembangan perangkat lunak karena dapat membantu menemukan bug atau masalah yang mungkin terjadi pada perangkat lunak sebelum diperkenalkan ke pasar atau digunakan oleh pengguna. Perangkat lunak yang tidak diuji dengan baik dapat menyebabkan risiko kegagalan pada perangkat lunak, yang pada akhirnya dapat merugikan pengguna atau perusahaan.

Oleh karena itu, sebaiknya tim pengembang mempertimbangkan dan mengalokasikan waktu dan sumber daya yang memadai untuk proses pengujian perangkat lunak. Dengan melakukan software testing yang baik dan terstruktur, maka dapat membantu memastikan kualitas dan keamanan perangkat lunak yang dihasilkan, serta menghindari kerugian yang lebih besar di kemudian hari. Seharusnya tim pengembang tidak hanya mengandalkan pengalaman proyek sebelumnya, namun memperhatikan pentingnya software testing dalam memastikan kualitas perangkat lunak yang dihasilkan.