**Nama : Rania Annisa**

**NIM : 1957301036**

**Kelas : TI 4B**

**Modul Software Quality Assurance QA**

**(Pada video 3-15)**

1. **Software Development Life Cycle (SDLC)**

SDLC (Software Development Life Cycle) adalah kerangka kerja atau model manajemen proyek terstruktur yang menguraikan fase-fase yang diperlukan untuk membangun sistem TI, dari awal hingga hasil akhir. Tujuan dari SDLC ini untuk menciptakan proses produksi yang efektif dan berkualitas tinggi yang memenuhi harapan pelanggan dalam anggaran dan jadwal yang ditentukan. SDLC memiliki cara kerja dengan mengurangi biaya pengembangan perangkat lunak, meningkatkan kualitas dan mempersingkat waktu produksi. SDLC juga dapat membantu tim mencapai tujuan mereka dengan mengevaluasi kekurangan sistem.

Berikut penjelasan sederhana tentang SDLC, yang dianalogikan dengan pengalaman restoran. Ada pelanggan yang memesan sesuatu yang dinginkan, tapi cara bicara pelanggan dan cara memasak chef memiliki bahasa yang berbeda. Oleh karena itu, dibutuhkan penerjemah bahasa yang disebut sebagai pemilik produk. Dalam kasus restoran ini, dapur adalah pengembangan tim (development tim). Juru masak (chef) sebagai pengembang yang melakukan sesuatu yang mana chef ini yang dapat merasakan terlebih dahulu sesuatu yang mereka masak. Jadi, sebelum makanan dari chef tersebut di hidangkan ke pelanggan, maka hanya chef yang mengetahui rasa dan penampilan dari makanan tersebut yang sesuai dengan permintaan pelanggan. Apabila, pesanan pelanggan sudah sesuai dengan keinginannya dan kualitas cukup baik untuk dilayani, maka akan dihidangkan ke pelanggan. Berikut gambaran dari SDLC sederhana yang dianalogikan sebagai restoran:



1. **Testing Definition**

Untuk memastikan kualitas dari suatu aplikasi, testing harus dilakukan yaitu dengan cara menguji apakah sistem data yang dihasilkan sesuai dengan testing yang telah dilakukan. Tujuan dari testing ini sendiri antara lain, yaitu:

* Memastikan aplikasi berjalan sebagaimana mestinya
* Mendeteksi terjadinya error serta memvalidasi apakah sudah memenuhi keinginan user
* Melakukan pengecekan/pengetesan entitas-entitas, termasuk software,untuk pemenuhan dan konsistensi dengan melakukan evaluasi hasil terhadap kebutuhan yang telah ditetapkan
* Melakukan validasi untuk dapat melihat kebenaran sistem, apakah proses yang telah ditulis dalam spesifikasi adalah apa yang sebenarnya.
* Mendeteksi adanya error, testing untuk mendeteksi kesalahan secara insentif, yaitu menentukan apakah suatu hal tersebut terjadi bilamana tidak seharusnya terjadi.

Terdapat dua tahapan proses perangkat lunak bebas bug dan memenuhi semua persyaratan, yaitu:

1. Verifikasi

Verifikasi adalah  suatu pengecekan ataupun pengetesan entitas- entitas, termasuk aplikasi, untuk pemenuhan serta konsistensi dengan melakukan penilaian hasil terhadap kebutuhan yang telah ditetapkan.

“are we building the product right?” (menurut Boehm)

“apakah kita telah membuat produk dengan benar?”

1. Validasi

Validasi yaitu setelah melaksanakan proses verifikasi berikutnya adalah melihat kebenaran sistem, apakah proses yang ditulis dalam spesifikasi adalah apa yang sesungguhnya diinginkan ataupun dibutuhkan oleh pengguna.

“are we building the right product?” (menurut Boehm)

“apakah kita telah membuat produk yang benar?”

Berikut 5 kegiatan utama yang akan dibahas pada testing definition, yaitu:



1. Bug Hunting 🡪 aktivitas untuk mencari bug dan ketika bug sudah ditemukan akan dilaporkan.
2. Bug reporting 🡪 pemantauan dalam melacak status dan riwayat, yang mana memerlukan tempat pusat dalam proses pemantauan.
3. Prioritization 🡪 tidak semuanya akan diperbaiki, hanya memprioritaskan kekurangannya yang masuk akal saja.
4. Test cases 🡪 menulis, mengeksekusi, dan melaporkan kasus yang akan diuji.
5. Story testing 🡪 untuk melakukan beberapa pengujian terhadap persyaratan, sehingga dapat menemukan bug, melaporkan dan menulis kasus yang diuji.
6. **Software Defect (Bug)**

Kesalahan dalam program komputer yang menyebabkan hasilnya tidak benar atau tidak terduga, atau berperilaku dengan cara yang tidak diinginkan.

Berikut ini adalah beberapa jenis bug yang paling sering terjadi selama proses pengembangan software.

1. Functional error

Functional error merupakan sebuah kategori luas yang mencakup masalah yang terkait fungsionalitas sebuah program. Bug jenis ini bervariasi, mulai dari tombol yang tidak dapat di-klik hingga masalah pada kegunaan aplikasi itu sendiri.

1. Performance defects

Performance defects adalah kategori bug yang terkait dengan kecepatan, stabilitas, response time, dan penggunaan sumber daya dari sebuah software.

1. Usability defects

Usability defects adalah jenis bug yang menyebabkan pengguna tidak dapat memanfaatkan sebuah software secara maksimal. Jenis bug ini biasanya menyebabkan software sulit atau tidak nyaman untuk digunakan. Selain masalah pada kode software, usability defects juga dapat disebabkan oleh desain UI yang terlalu rumit sehingga pengguna kesulitan menemukan fungsi yang mereka cari.

1. Compatibility error

Compatibility error merujuk pada masalah software yang tidak dapat berjalan sebagaimana mestinya dalam situasi tertentu. Pada umumnya, compatibility error akan muncul ketika dijalankan aplikasi lama pada sistem operasi yang lebih baru.

1. Security error

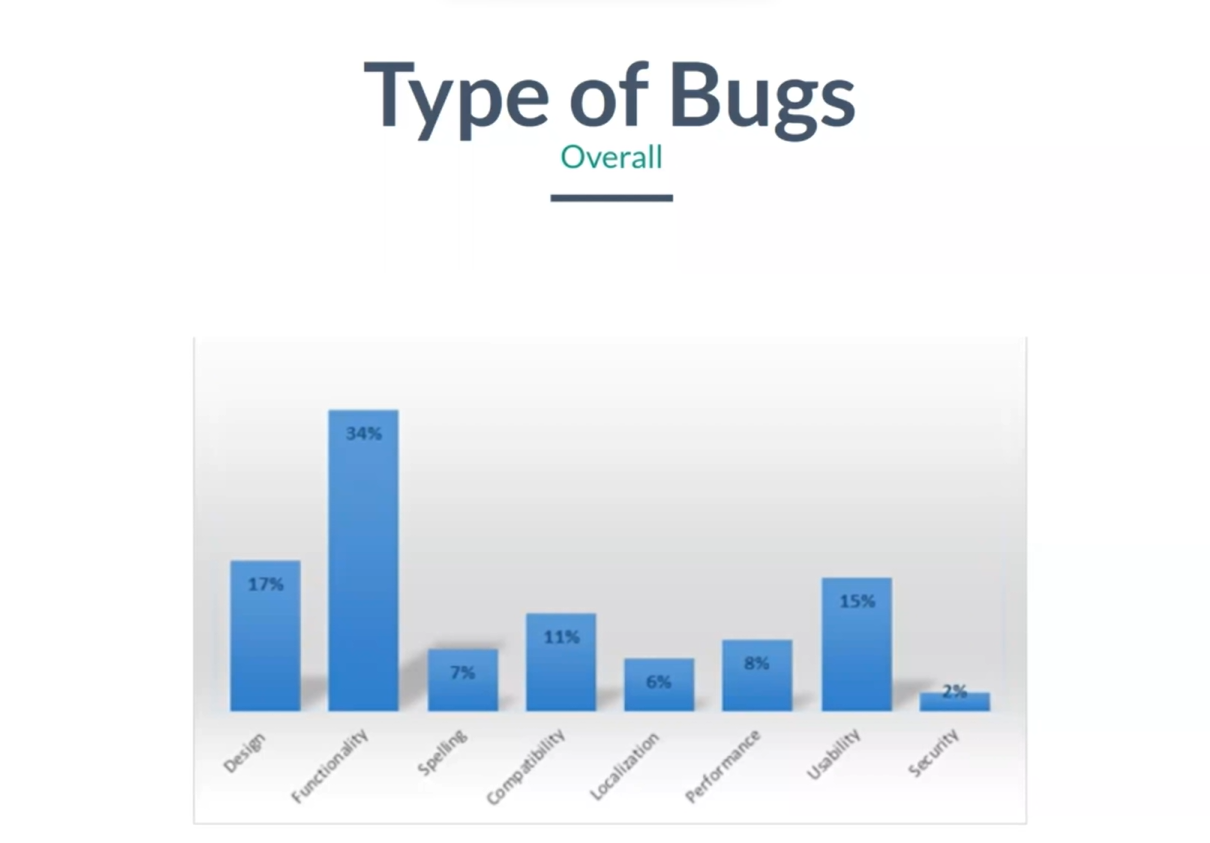
Security error merupakan jenis bug yang cukup berbahaya karena terkait langsung dengan sistem keamanan sebuah software. Sebuah bug dalam sistem keamanan software, besar atau kecil, akan membuka celah untuk cracking dan mencuri data atau merusak software tersebut.

1. Syntax error

Syntax error adalah jenis bug yang terjadi pada source code sebuah program. Jenis bug ini akan menghambat proses pengembangan software dan menyebabkan software tidak dapat di-compile dengan baik.

1. Logic error

Logic error adalah jenis bug lain yang terjadi pada source code sebuah program. Jenis bug ini biasanya disebabkan oleh penggunaan kode yang tidak sesuai sehingga menghasilkan output yang salah.



Pada grafik ini, functionality merupakan hal utama yang menjadi masalah /defect yang sangat berpengaruh terhadap software.

1. **Tipe-tipe pada pengujian**
2. UI Testing

Kerusakan/ketidaksempurnaan pada UI/design dapat dikategorikan dalam 5 kategori, yaitu:

1. Layout

Kerusakan pada layout dapat terlihat pada ketidaksejajaran margin, ketidaksempurnaan model kotak IE6(Internet Explorer 6), overlapping, kerusakan gambar, jarak lebar dan tinggi garis, ruang yang tidak diingankan diantara daftar item, jarak vertikal yang memiliki kecacatan.

1. Font

Kerusakan pada font dapat terlihat dari tipe font yang berbeda, baik dari family font yang digunakan maupun ukuran font yang digunakan.

1. Color

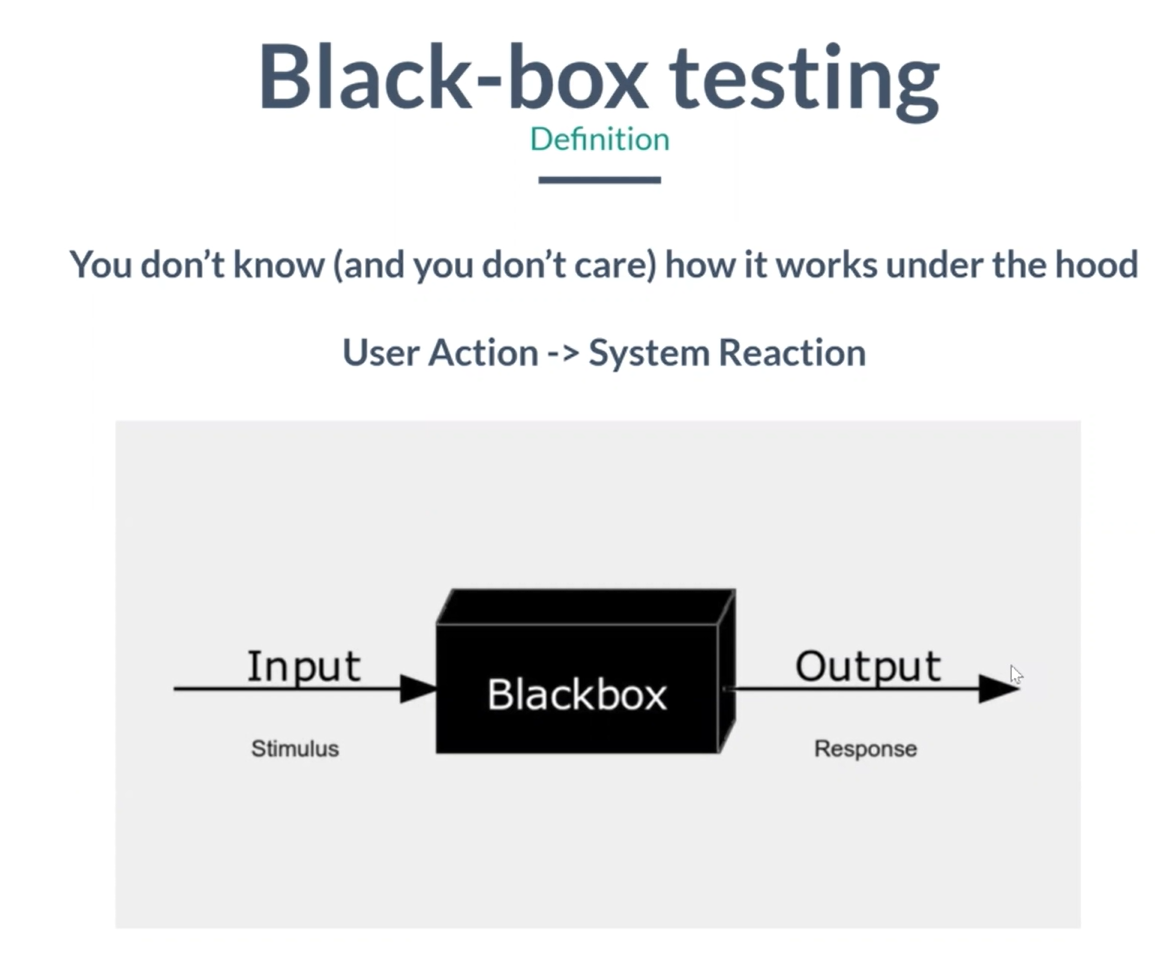
Kerusakan pada warna dapat terlihat dari tidak adanya keseragaman warna yang digunakan dan warna tombol yang tidak berfungsi sesuai dengan kebutuhan.

1. Content

Kerusakan pada content dapat terlihat dari perbedaan besar kecilnya suatu ukuran font, tidak adanya keseragaman kata, kerusakan gambar serta kesalahan dalam ejaan kata.

1. Functional Testing

Fungsional testing adalah sistem yang diuji terhadap fungsional persyaratan atau spesifikasinya. Jenis pengujian ini juga memverifikasi bahwa setiap fungsi aplikasi telah beroperasi sesuai dengan kebutuhan, yang berfokus pada manual dan automation testing. Fungsional testing ini merupakan salah satu jenis pengujian metode black box testing di mana setiap fungsionalitas aplikasi diuji dengan memberikan satu set input, yang diberikan untuk mengetahui reaksi aplikasi yang sebenarnya dan kemudian membandingkannya dengan hasil yang diharapkan sesuai dengan spesifikasi yang diberikan. Berikut contoh gambaran dari black box testing:



Pengguna memberikan aksi sebagai input dan sistem memberikan reaksi/respon sebagai output. Black box testing hanya akan menjangkau input dan output sistem software tanpa adanya pengetahuan terkait internal program. Metode pengujian black box ini sangat penting agar bisa menemukan gangguan atau bug dalam suatu aplikasi sebelum di publikasikan secara resmi.

Adapun beberapa jenis dari functional testing, yaitut:

1. Unit Testing: Pengujian secara individual dan independen dari bagian aplikasi terkecil yang dapat diuji.
2. Smoke Testing: Pengujian ini dilakukan pada pembuatan perangkat lunak awal untuk memeriksa apakah fungsi kritis berfungsi dengan baik atau tidak untuk melakukan pengujian lebih lanjut.
3. Sanity Testing: Pengujian yang dilakukan pada perangkat lunak yang stabil, yang mengalami perubahan kecil dalam kode atau fungsionalitas. Tujuannya adalah untuk memastikan bahwa kesalahan telah diselesaikan dan memastikan bahwa tidak ada lagi kesalahan/bug yang disebabkan oleh perubahan baru.
4. Integration Testing: Pengujian yang dilakukan untuk memeriksa fungsi perangkat lunak yang benar ketika komponen digabungkan sebagai satu unit.
5. Regression Testing: Pengujian Regresi dilakukan ketika bug diperbaiki atau fungsionalitas baru telah diperkenalkan ke suatu program. Ini memastikan bahwa fungsionalitas yang ada tidak rusak karena pengenalan perubahan baru.
6. User Acceptance Testing: ini adalah jenis pengujian perangkat lunak di mana suatu sistem diuji untuk penerimaan pelanggan. Satu-satunya tujuan pengujian ini adalah untuk mengevaluasi kepatuhan sistem dengan persyaratan bisnisnya dan untuk mengevaluasi apakah dapat diterima untuk pengiriman. Ini juga memverifikasi apakah perangkat lunak dapat melakukan tugas tertentu dalam situasi nyata.
7. Compatibility Testing

Compatibility testing merupakan bagian dari non fungsional testing pada perangkat lunak, yaitu pengujian yang dilakukan pada aplikasi untuk mengevaluasi kompatibilitas aplikasi dengan komputasi. Komputasi ini mencakup beberapa aspek seperti hardware, sistem operasi, database, browser, dan sebagainya. Compatibility testing digunakan untuk memeriksa apakah sistem yang dikembangkan mampu berjalan pada hardware, software, sistem operasi, atau pada jaringan yang berbeda. Tidak hanya itu, compatibility testing juga digunakan untuk mengetahui dan menentukan versi *minimum* untuk dapat menjalankan aplikasi pada perangkat yang sedang dilakukan uji coba. Semakin sistem dapat dijalankan di banyak jenis perangkat yang berbeda, maka semakin baik aspek kompatibilitasnya.

1. Localization Testing

Pengujian lokalisasi yaitu memeriksa aplikasi sesuai dengan setting lokal untuk menyesuaikan bahasa dan negara yang sesuai dengan keinginan pengguna. Oleh karena itu, format pada mata uang, tanggal dan waktu akan berubah sesuai dengan pengaturan lokal yang dipilih oleh pengguna.

1. Performance Testing

Performance testing yaitu untuk menentukan kecepatan akses, daya tanggap, dan stabilitas program perangkat lunak. Pada pengujian ini, seorang software tester / QA tidak dituntut untuk mencari error melainkan mencari hambatan-hambatan yang dapat membuat sebuah aplikasi loading terlalu lama atau mungkin gagal diakses. Aplikasi yang loading-nya lama tentu akan memberikan penilaian yang buruk pada pengguna. Adapun tujuan dari *Performance testing* yaitu mengevaluasi output aplikasi, kecepatan pemrosesan, kecepatan transfer data, penggunaan bandwidth jaringan, pengguna bersamaan maksimum, pemanfaatan memori, efisiensi beban kerja, dan waktu respon perintah.

Terdapat tiga faktor utama dalam memenuhi harapan/hasil dari performance testing, yaitu sebagai berikut:

* Speed: Melihat kecepatan sistem dalam memberikan respon dari setiap request.
* Scalability: Menentukan berapa jumlah load maksimum yang dapat ditangani sistem.
* Stability: menganalisis kondisi sistem saat diberikan beban yang bervariasi, apakah stabil atau tidak.

1. Usability testing

Usability testing adalah metode yang digunakan untuk mengevaluasi seberapa mudah situs web menggunakan User Experience (UX) yang dilakukan pada produk yang fokus pada identifikasi masalah kegunaan, mengumpulkan data kualitatif dan kuantitatif, dan menentukan kepuasan dari pengguna produk. Umumnya usability test mengevaluasi persyaratan fungsional dan kualitas dari user interface. Usability test ini juga memiliki hubungan antara pengguna dengan sistem yang dilakukan untuk menentukan apakah fungsi telah sama seperti yang diharapkan dan apakah user interface membuat sistem dapat mudah digunakan. Pengujian ini sering dilakukan untuk mendapatkan hasil yang cepat dalam meningkatkan interface dan mengkoreksi kesalahan dalam komponen perangkat lunak. Salah satu contoh usability box new away 2018, ketika semua orang dari negara bagian mendapatkan pesan peringatan hanya karena kesalahan seseorang meng-klik sesuatu yang salah dikomputer, karena masalah penggunannya yang mana memiliki dua tombol yang sama disamping satu sama lain. Oleh karena itu, lebih baik untuk menggunakan warna yang berbeda agar dapat membedakan antara satu tombol dengan tombol yang bersampingan.

1. Security Testing

Security testing yaitu jenis pengujian perangkat lunak yang dilakukan untuk mengidentifikasi kerentanan serta memastikan bahwa data dan sumber daya sistem di dalamnya sudah terlindungi dengan baik dari para penyusup. Pengujian ini dilakukan dengan tujuan untuk menemukan semua celah dan kelemahan sistem yang dapat mengakibatkan hilangnya informasi. Adapun tahapan-tahapan dari security testing, sebagai berikut:

1. Asessment yaitu analisis keamanan untuk persyaratan dan periksa kasus penyalahgunaan/penyalahgunaan
2. Design yaitu analisis risiko keamanan untuk perancangan. Pengembangan Rencana Uji termasuk uji keamanan
3. Develop yaitu panduan kode bersama dengan pengujian untuk unit dasar.
4. Testing yaitu pengujian black box, pengujian statis dan dinamis dan pengujian white box, dan pemindaian kerentanan.
5. Deploy yaitu pengujian penetrasi, pemindaian kerentanan
6. Monitor yaitu analisis dampak patch, pengujian regresi dan tinjauan berkala terhadap perkembangan lebih lanjut.
7. Ad-hoc or Monkey Testing

Ad-hoc testing yaitu salah satu cara untuk menemukan bug dalam pengujian yang mengacu pada beberapa tes yang pada dasarnya terdapat jenis pengujian yang hanya dilakukan pengujian secara acak. Setelah bug ditemukan, maka akan dilakukan tangkapan layar yang mana cara dalam menangkap layar tergantung pada sistem yang digunakan. Berikut shortcut dalam tangkapan layar pada beberapa sistem, yaitu:



1. **Cara Melaporkan Bug**

Untuk melaporkan bug dapat menggunakan Jira, berikut 4 langkah utama dalam log bugs dalam Jira:



1. Type

Pilih jenis masalah bug, yang mana tergantung pada proyek yang telah disiapkan di JIRA. Jenis masalah dapat berisi nilai berikut: bug, story, epic, task dan sebagainya untuk melaporkan kerusakan.



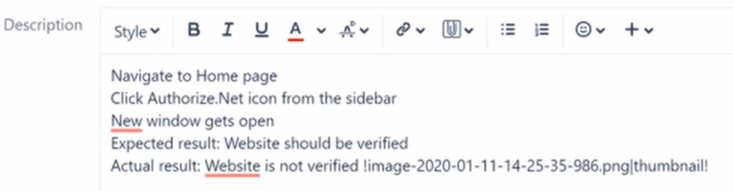
1. Summary

Menulis ringkasan singkat tentang apa dan dimana yang salah. Dapat ditulis di summary yaitu [Authorize.net] Verification is missing.



1. Description

Deskripsi ini adalah bidang yang paling penting ketika membuat laporan bug, yang mana deskripsi ini sedikit lebih deskriptif dari dari ringkasan (summary).



1. Screenshot

Screenshot jendela Authorize.Net untuk lebih terpercaya dan aman, maka dapat ditampilkan screenshot jendela Authorize.Net. Setelah itu, masukkan file image di attachment.

