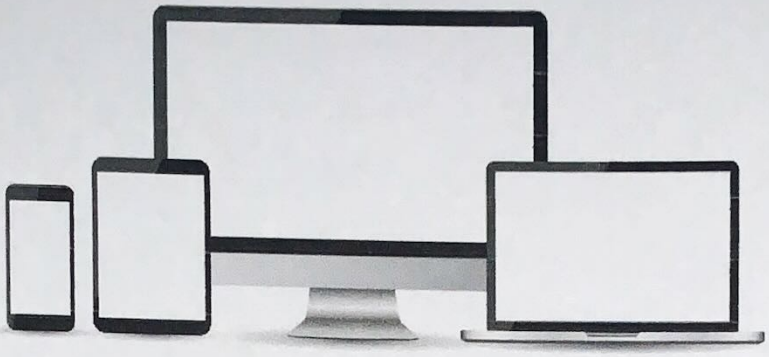
C. Front-End Testing

Pengujian *front-end* adalah pengujian yang dilakukan dari segi tampilan yang dapat dilihat oleh *user*. Pengujian dari segi *front-end* biasanya melibatkan pengujian seperti *Web Application Testing, Desktop Application Testing*, dan *Mobile Application Testing*. Adapun ilustrasi dari pengujian *front-end* sebagaimana gambar 7.3.



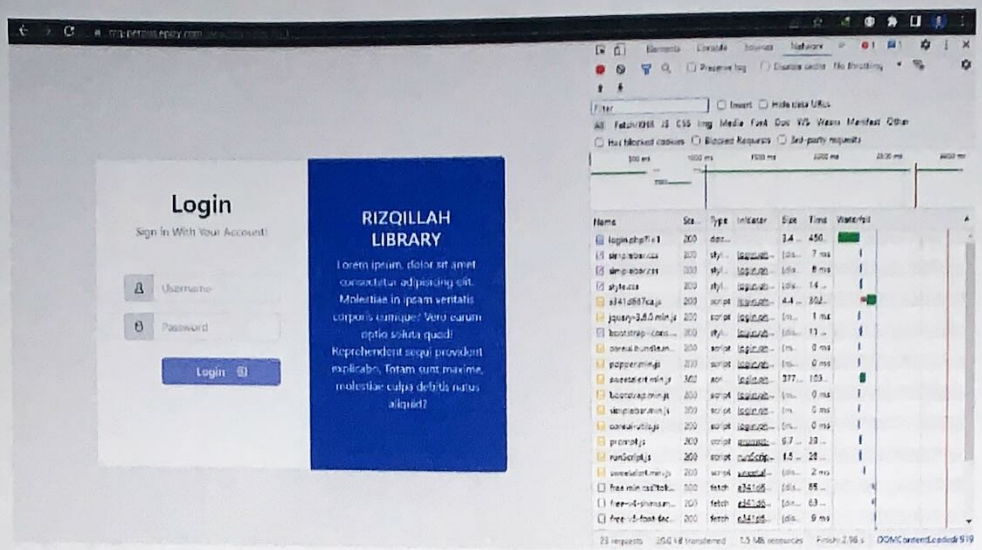
Gambar 7.3 Ilustrasi *front-end*

Dari gambar 7.3 dapat diketahui bahwa pengujian secara *front-end* akan menguji tampilan dari sebuah web, seperti jika *user* membuka web menggunakan perangkat *mobile* maka tampilannya dengan perangkat d*esktop* tentunya tidaklah sama. Oleh karena itu, pengujian secara *front-end* pada *web architecture* sangat dibutuhkan agar *website* yang dibangun *responsive* terhadap segala jenis lingkungan/*platform*.

D. How to Debug Front-End or Back-End Issues?

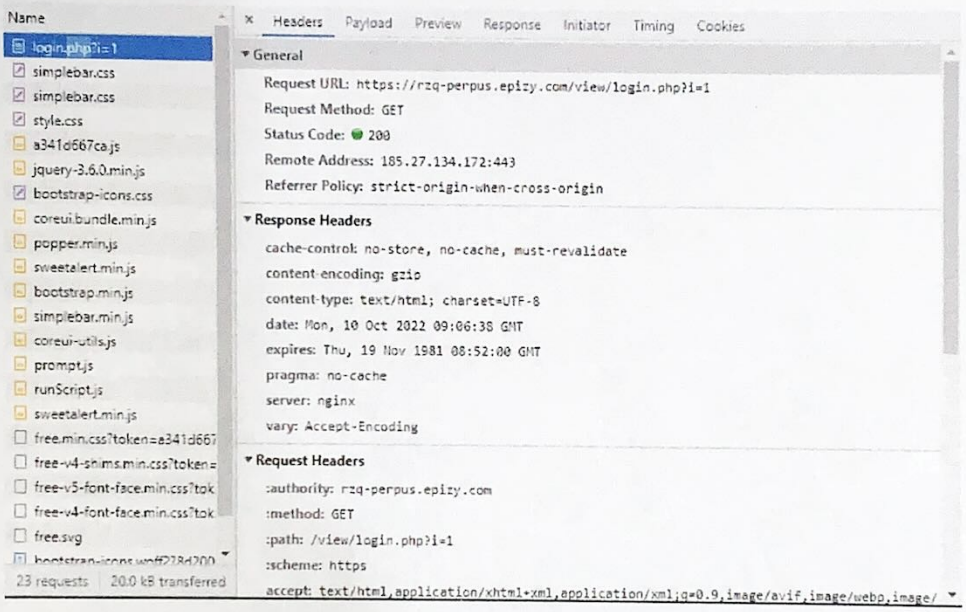
Adapun contoh pengujian terhadap *front-end* dan *back-end* adalah terhadap *website* https://rzq-perpus.epizy.com. Pengujian yang dilakukan sebagai berikut.

1. Adapun kecepatan proses pembukaan halaman web dapat dilihat pada gambar 7.4.



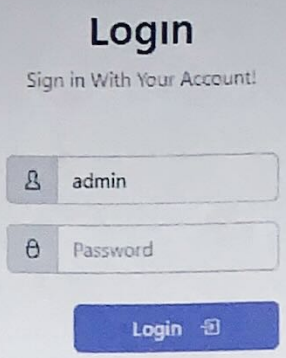
Gambar 7.4 *Network* dari *website*

2. Pada bagian file login.php, dapat dilihat informasi *load* dari file pada gambar 7.5



Gambar 7.5 *Load* file login.php

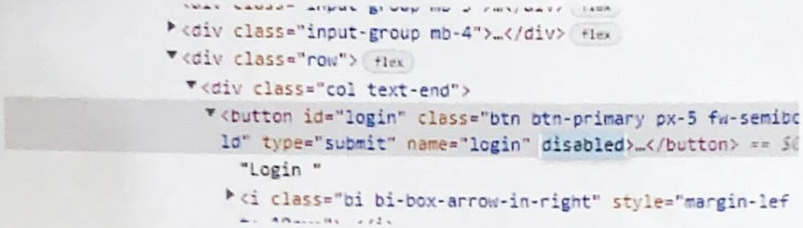
3. Mencoba mengosongkan inputan *password* pada *form login*, maka hasil dari *website* seperti pada gambar 7.6



Gambar 7.6 Tampilan *form login*

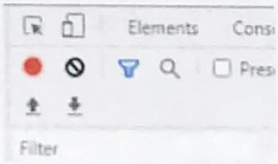
Secara otomatis tombol *login* tidak akan bisa di klik jika inputan kosong.

4. Adapun jika membuat ~~*inspect* elemen jika~~ ingin membuat tombol *login* bisa di klik , anda dapat membuatnya pada inspect element, kemudian pergi ke element button dan ~~dengan cara~~ hapus *disable* pada elemen *button*.



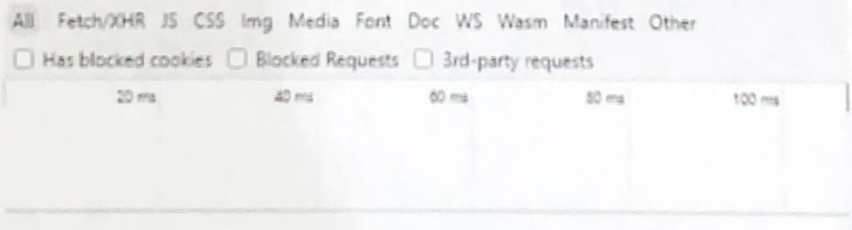
Gambar 7.7 Hapus *disabled button*

5. Buka bagian *network*, dan hapus seluruh proses yang ada, maka hasilnya seperti pada gambar 7.8



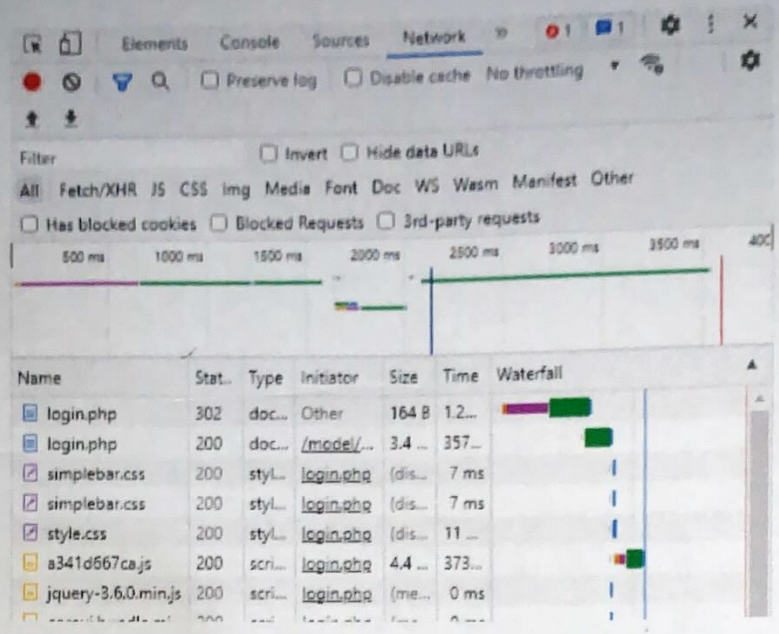
Gambar 7.8 Tombol *clear network*

Hasil setelah *clear network*



Gambar 7.9 Clear network

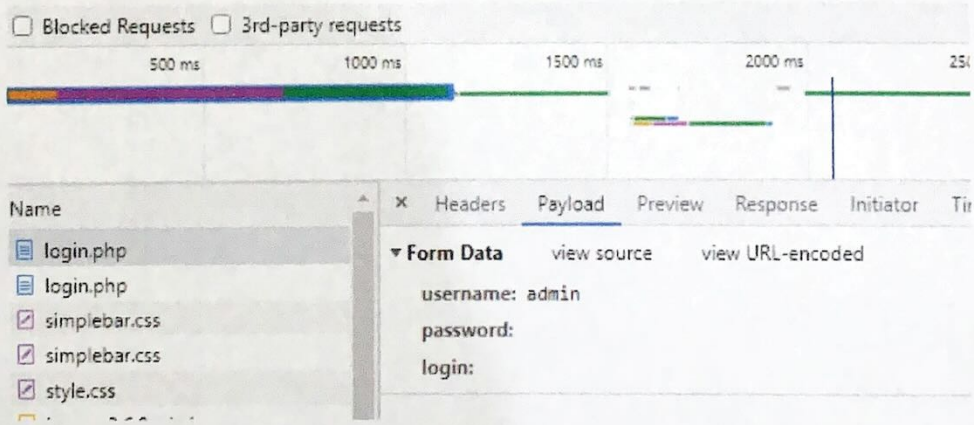
6. Kemudian klik tombol *login* dengan *password* yang kosong, maka hasil pada *network* sebagai pada gambar 7.10.



Gambar 7.10 Hasil *login* tanpa *password*

Dari gambar 7.10 dapat diketahui bahwa proses validasi terhadap kolom *login* masing kurang bagus, dikarenakan harus melakukan proses secara *front-end* dan *back-end*. Alangkah lebih bagus menggunakan proses validasi seperti menggunakan Ajax atau pengecekan javascript terlebih dahulu sebelum melakukan proses *login*.

7. Pada bagian file login dengan status 302 kita akan dapat melihat status *payload* dengan informasi dari *username* dan *password* yang diinput



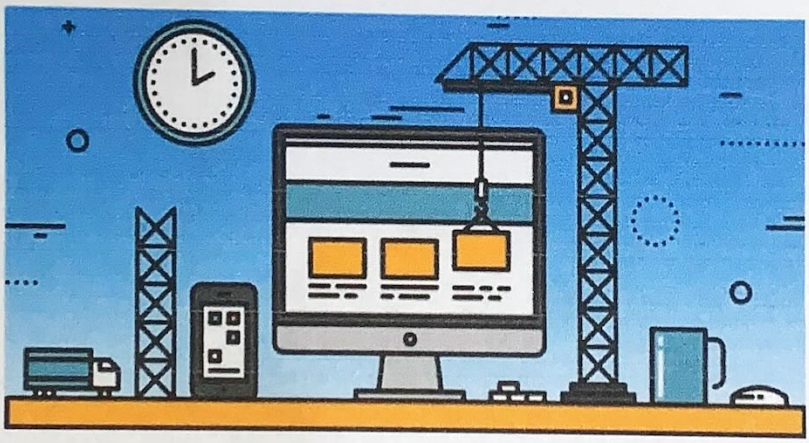
Gambar 7.11 *Payload form data*

BAB VIII
BUILD AND RELEASES

A. What is Software Build?

Setelah mengembangkan modul perangkat lunak, pengembang mengubah kode sumber menjadi bentuk mandiri atau kode yang dapat dieksekusi. Kemudian tim pengembang menyerahkan hasil pengembangan kepada tim pengujian untuk melakukan pengujian. ~~Kemudian~~ selanjutnya tim penguji perangkat lunak akan memeriksa aplikasi tersebut. Jika terdapat beberapa bug dan ~~jika~~, tidak memenuhi persyaratan, maka tim pengujian perangkat lunak menolak pengembangan aplikasi tersebut. *Build* dilakukan sebelum proses perilisan perangkat lunak ke pasar.

Build adalah proses kompilasi, di mana seluruh *source code file* (*text*) diubah menjadi sebuah kode yang dapat dieksekusi. Adapun ilustrasi dari *build* dapat dilihat pada gambar 8.1.



Gambar 8.1 Ilustrasi *build* aplikasi

Dari ilustrasi di atas, dapat dilihat bahwa proses sebelum merilis sebuah aplikasi ke publik/pasar harus melakukan kompilasi atau membuat sebuah kode teks yang dibuat menjadi dapat dieksekusi oleh *platform* tujuan aplikasi. Perbedaan utama dari *build* dan *release* terletak pada pengujian perangkat lunak. Aplikasi yang masih dalam tahap *build* wajib memerlukan pengujian mendalam, akan tetapi aplikasi yang telah masuk ke tahap *release* sudah tidak perlu melakukan pengujian apapun.

B. What is Deployment?

Pada tahapan pengembangan sistem, implementasi sistem merupakan tahapan akhir dalam pembangunan sistem. Pada tahapan implementasi, perpindahan dari sistem lama ke sistem baru dibagi menjadi 3 macam yaitu *direct* (secara langsung), *parallel* (setengah-setengah) dan juga *phased* (pergantian ke sistem baru dibuat per modul sistem).

Deployment dalam *software* dan web berarti mendorong perubahan atau pembaruan dari satu lingkungan penyebaran ke lingkungan penyebaran lainnya. Saat menyiapkan situs web, kita akan selalu memiliki situs web langsung (*localhost*), yang disebut lingkungan langsung atau lingkungan produksi. *Deployment* adalah kegiatan yang bertujuan untuk menyebarkan aplikasi yang telah dikerjakan oleh para pengembang. Penyebarannya dapat melalui beragam cara tergantung dari jenis aplikasinya.

Deployment Activites meliputi ~~melakukan~~ kegiatan tes sistem dan stres, melakukan tes penerimaan, mengonversi data yang ada, membangun materi pelatihan/melakukan pelatihan, mengonfigurasi dan mengatur lingkungan produksi, dan menyebarkan solusi. *Testing* adalah aktivitas utama implementasi beserta penyebaran dan mencakup pengujian unit, uji integrasi, tes kegunaan, tes sistem/kinerja/stres, dan tes penerimaan.

Development plan program adalah *trade-off* (perdagangan) di antara sumber daya yang tersedia, waktu yang tersedia, dan keinginan untuk mendeteksi dan memperbaiki kesalahan sebelum penyebaran sistem dilakukan.

C. What is Release?

Rilis adalah aktivitas terakhir setelah menyelesaikan pengembangan dan pengujian. Setelah menguji pembuatan aplikasi, tim pengujian mengesahkan perangkat lunak itu dan mengirimkannya kepada pelanggan. Dimungkinkan untuk satu rilis memiliki beberapa *build*. Oleh karena itu, perangkat lunak yang dikirim ke pelanggan adalah perangkat lunak yang telah menyelesaikan fase pengembangan dan pengujian.

Build mengacu pada perangkat lunak mandiri yang dihasilkan setelah mengonversi kode sumber ke kode yang dapat dieksekusi dan dapat dijalankan di komputer. Rilis juga disebut sebagai distribusi versi final suatu aplikasi. Dengan demikian, definisi-definisi ini menjelaskan perbedaan mendasar antara *build* dan rilis. Jadi perbedaan utama antara *build* dan rilis dalam pengujian perangkat lunak yaitu :

- *Build* adalah versi perangkat lunak yang diserahkan oleh tim pengembangan ke tim pengujian untuk tujuan pengujian, sedangkan
- Rilis adalah perangkat lunak yang tim uji berikan kepada pelanggan untuk dicoba

D. What is Sanity Testing?

Sanity testing adalah sejenis pengujian perangkat lunak yang dilakukan setelah menerima pembuatan perangkat lunak, dengan sedikit perubahan dalam kode, atau fungsionalitas, untuk memastikan bahwa *bug* telah diperbaiki dan tidak ada masalah lebih lanjut yang ditimbulkan karena perubahan ini. Tujuannya adalah untuk menentukan bahwa fungsionalitas yang diusulkan bekerja seperti yang diharapkan. Jika *sanity testing* gagal, *build* akan ditolak untuk menghemat waktu dan biaya yang terlibat dalam pengujian yang lebih ketat.

Tujuannya adalah "bukan" untuk memverifikasi secara menyeluruh fungsionalitas baru tetapi untuk menentukan bahwa pengembang telah menerapkan beberapa rasionalitas saat memproduksi perangkat lunak. Contoh s*anity testing*:

Dalam proyek *e-commerce*, modul utama adalah halaman *login*, halaman beranda, halaman profil pengguna, pendaftaran pengguna, dll. Terdapat kecacatan proses pada halaman *login* ketika kata sandi menerima kurang dari empat karakter dan persyaratan menyebutkan bahwa kata sandi ini ~~bidang~~ tidak boleh di bawah delapan karakter. Oleh karena itu, cacat sistem dilaporkan oleh tim pengujian ke tim pengembangan untuk menyelesaikannya. Kemudian tim pengembang~~an~~ memperbaiki cacat yang dilaporkan dan mengirimkannya ke tim pengujian untuk dibersihkan. ~~Kemudian~~ tim penguji memeriksa apakah perubahan yang dilakukan berfungsi dengan baik atau tidak. ~~Itu juga ditentukan~~ apakah ~~itu~~ berdampak pada fungsi terkait atau lainnya. Sekarang ada fungsi untuk memperbarui kata sandi di halaman profil pengguna.

Keuntungan dari s*anity testing*:

1. *Sanity testing* membantu dengan cepat mengidentifikasi cacat pada fungsionalitas inti.
2. Dapat dilakukan dalam waktu yang lebih singkat karena tidak ada dokumentasi yang diperlukan untuk *sanity testing*.
3. Jika cacat ditemukan selama *sanity testing*, proyek akan ditolak untuk menghemat waktu pelaksanaan uji regresi.
4. Teknik pengujian ini tidak begitu mahal jika dibandingkan dengan jenis pengujian lainnya.

5. Membantu untuk mengidentifikasi objek hilang yang bergantung.
6. Digunakan untuk memverifikasi fungsionalitas kecil dari aplikasi sistem apakah masih berfungsi atau tidak bahkan setelah perubahan kecil.
7. Membantu dalam skenario ketika waktu untuk pengujian produk terbatas atau memiliki lebih sedikit waktu untuk menyelesaikan pengujian.

Kekurangan dari Sanity Testing :

1. Hanya berfokus pada fungsi dan perintah dari aplikasi sistem.
2. Tidak mencakup semua kasus uji dalam skenario uji.
3. Hanya mencakup beberapa fungsi dalam aplikasi sistem. Masalah dalam fungsi yang tidak dicentang tidak dapat dipulihkan.
4. *Sanity testing* biasanya tanpa naskah. Oleh karena itu, referensi di masa mendatang tidak tersedia.
5. Tidak mencakup tingkat struktur disain dan karenanya akan sulit bagi tim pengembangan untuk mengidentifikasi dan memperbaiki masalah.

E. What is User Acceptance Testing?

User Acceptance Testing (UAT) adalah jenis pengujian yang dilakukan oleh pengguna akhir atau klien untuk memverifikasi/menerima sistem perangkat lunak sebelum memindahkan aplikasi perangkat lunak ke lingkungan produksi. UAT dilakukan pada tahap akhir pengujian setelah pengujian fungsional, integrasi dan sistem dilakukan.

Tujuan utama UAT adalah untuk memvalidasi aliran bisnis ujung ke ujung dan tidak berfokus pada kesalahan unsur pelengkap, kesalahan ejaan atau pengujian sistem. Pengujian penerimaan pengguna dilakukan dalam lingkungan pengujian terpisah dengan pengaturan data seperti produksi. UAT adalah jenis pengujian *black-box* di mana dua atau lebih pengguna akhir akan terlibat. UAT dilakukan oleh klien dan pengguna akhir.

Kebutuhan pengujian penerimaan pengguna muncul setelah perangkat lunak telah menjalani pengujian unit, integrasi dan sistem karena pengembang mungkin telah membangun perangkat lunak berdasarkan dokumen persyaratan dengan pemahaman mereka sendiri dan perubahan yang diperlukan lebih lanjut selama pengembangan mungkin tidak dikomunikasikan secara efektif kepada mereka, jadi untuk menguji apakah hasil akhir produk

perlu dilakukan

diterima oleh klien/pengguna akhir, pengujian penerimaan pengguna ~~diperlukan~~. Berikut adalah kriteria entri untuk UAT:

1. Persyaratan bisnis harus tersedia.
2. Kode aplikasi harus dikembangkan sepenuhnya
3. Pengujian unit, pengujian integrasi & pengujian sistem harus diselesaikan
4. Tidak terdapat *showstoppers*, cacat tinggi, sedang dalam fase uji integrasi sistem
5. Hanya kesalahan unsur pelengkap yang dapat diterima sebelum UAT
6. Pengujian regresi harus diselesaikan tanpa cacat besar
7. Semua cacat yang dilaporkan harus diperbaiki dan diuji sebelum UAT
8. Matriks ketertelusuran untuk semua pengujian harus diselesaikan
9. Lingkungan UAT harus siap
10. Menandatangani surat atau komunikasi dari tim pengujian sistem bahwa sistem siap untuk eksekusi UAT

BAB IX
QUALITY ASSURANCE

A. Quality Assurance (QA)

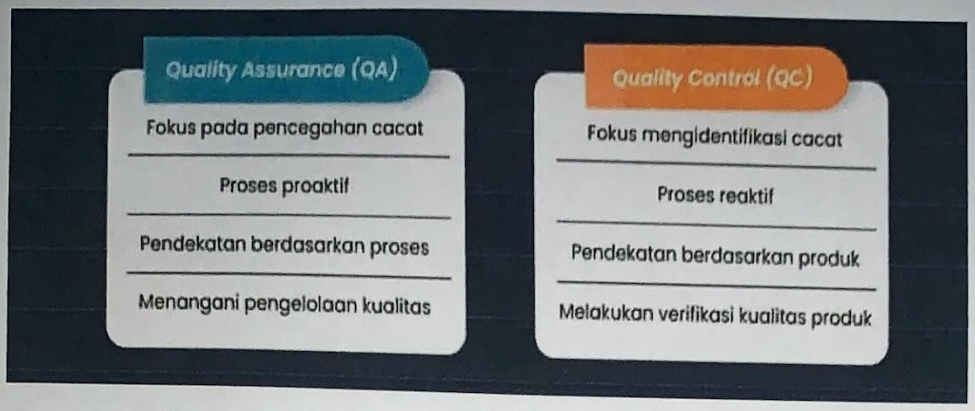
Quality Assurance adalah suatu profesi yang berperan untuk memastikan kualitas suatu produk sesuai dengan ketentuan yang berlaku dan menyiapkan segala kebutuhan dari aplikasi yang dibangun oleh perusahaan sehingga dapat bekerja dengan baik.. QA biasa ditemukan dalam perusahaan manufaktur dan perusahaan software.

Quality Assurance Testing merupakan serangkaian proses sistematis guna menentukan apakah suatu produk dan jasa harus memenuhi syarat yang ditentukan. QA akan menentukan serta menetapkan persyaratan dalam membuat atau mengembangkan produk tertentu agar memiliki kualitas yang baik. Bukan tanpa alasan mengapa kualitas suatu produk sangat penting untuk diperhatikan.

Kualitas terbaik dari adalah cara paling utama menjaga kredibilitas suatu perusahaan, selain itu juga cara meningkatkan kepercayaan konsumen, proses kerja hingga membuat perusahaan yang mampu ~~membuat mereka~~ bersaing dengan kompetitor. *Quality assurance* artinya menggunakan pendekatan proses agar tidak memunculkan produk yang cacat.

Itulah mengapa biasanya QA juga akan melakukan monitoring pembuatan produk mulai dari tahap perencanaan hingga proses pengujian. Kegiatan itu dilakukan demi mengurangi proses pengulangan pembuatan atau *rework*, sehingga proses kerja menjadi lebih efisien serta dapat menghindari keluhan dari konsumen.

B. Perbedaan Quality Assurance (QA) dengan Quality Control (QC)



Gambar 9.1 Penjelasan perbedaan antara QA dengan QC

Perbedaan *Quality Control* (QC) dan *Quality Assurance* (QA) terletak pada tugas dan tanggung jawab masing-masing. Walaupun sama-sama dalam satu departemen, QA memiliki peran dalam menjamin kualitas, sementara QC memiliki fungsi sebagai pengendali kualitas dari produk yang dihasilkan oleh perusahaan sesuai dengan standar keandalan, kegunaan, kinerja maupun standar lainnya.

QA memiliki tanggung jawab dalam memastikan sebuah produk sebelum dilepas ke pasaran, sebelum dirilis produk harus sudah memenuhi semua standar kualitas dalam setiap komponen. Pejabat staf QA memiliki kewajiban untuk aktif melakukan *monitoring* dan serangkaian pengujian dalam menetapkan kualitas pada pembeli.

Hal ini berbeda dengan QC dengan tanggung jawab memeriksa produk sebelum dan hingga setelah proses produksi menetapkan standar kualitas yang diperlukan. Pejabat QC memiliki hak menerima atau menolak produk yang akan dilepas ke pasaran, sehingga ketika ditemukan produk cacat, maka akan dikembalikan ke bagian produksi.

Kedua posisi ini memiliki keterkaitan dan dapat bekerja secara bersamaan hingga saling berkolaborasi. Jenjang karir dari QA bisa mencapai tingkatan seorang *Project Manager* jika memiliki pengalaman dalam melakukan analisis dan melakukan audit suatu produk. Selain itu jenis pekerjaan ini juga bisa menjadikan *DevOps,* tugasnya mengotomatisasi proses tahap pengembangan aplikasi.

Untuk bisa menekuni pekerjaan ini seseorang bisa memulai karir lebih dulu menjadi *Customer Experience Leader* atau *IT Management* dengan peran pentingnya dalam memegang kendali penuh dari setiap proses pengembangan produk, serta memiliki kewajiban mengutamakan kebutuhan dari konsumen atau pelanggan.