

# SOFTWARE QUALITY ASSURANCE



## What is SQA?

TKB5351 – Penjaminan Mutu Perangkat Lunak

Chalifa Chazar  
[www.script.id](http://www.script.id)  
[chalifa.chazar@gmail.com](mailto:chalifa.chazar@gmail.com)



# Question

- What is a software?
- What is an application?

# Explanation

- What is a software?
- **Software** : sekumpulan/baris instruksi (perintah) yang memberikan tugas apa yang harus dilakukan oleh hardware.
  
- What is an application?
- **Application** : sebuah software yang menjalankan suatu tugas tertentu (spesifik) mewakili seorang user

# Definition of Software

- (IEEE) **Software** : computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system.
- (ISO) **Software memiliki 4 komponen** yaitu :
  - program komputer (kode),
  - prosedur,
  - dokumentasi dan
  - data yang dibutuhkan untuk mengoperasikan software.

# Explanation

- **Kode** : diperlukan untuk memberikan instruksi kepada komputer untuk menjalankan aplikasi.
- **Prosedur** : diperlukan untuk menentukan urutan dan jadwal program-program yang dilakukan, metode yang digunakan, dan orang yang bertanggung jawab untuk melakukan kegiatan penerapan perangkat lunak.
- **Dokumentasi** : diperlukan untuk proses pengembangan, penggunaan (manual book), dan pemeliharaan.
- **Data** : (data kode, data orang yang mengoperasikan, data kesalahan proses, data implementasi, dll) diperlukan untuk pengujian standar.

# Conclusion

- 4 komponen tsb diperlukan untuk menjamin kualitas proses pengembangan perangkat lunak dan untuk proses pemeliharaan di masa yang akan datang.
- Penjaminan mutu kualitas perangkat lunak (SQA) selalu berhubungan dengan kualitas kode, kualitas prosedur, kualitas dokumentasi dan kualitas data dari suatu perangkat lunak.

# Question

- Software xx digunakan oleh **perusahaan A** selama 3 tahun dan perusahaan A tidak pernah menemukan error software xx.
- **Perusahaan B** baru menggunakan software xx selama 2 dan menemukan banyak error dalam software xx.
- **Perusahaan C** sudah menggunakan software xx selama 4 tahun dan tidak pernah menemukan kegagalan dari software xx, tetapi 1 bulan terakhir perusahaan C menemukan beberapa error software xx.
  
- Apakah **software xx** dikatakan **software gagal**?

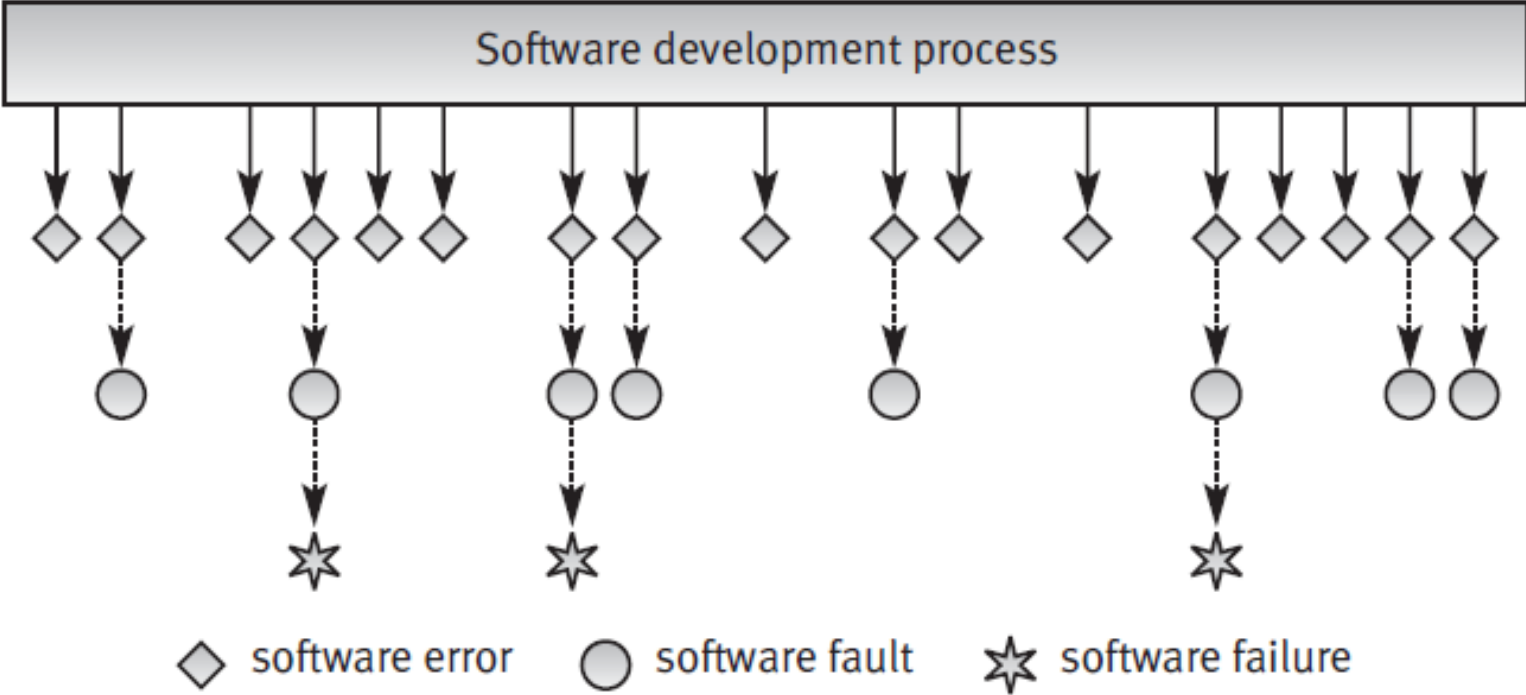
# Error, Fault & Failure

- Error?
- Fault?
- Failure?
  
- **Error** : sebuah tindakan manusia yang menghasilkan suatu hasil yang salah (kesalahan).
- **Fault** : kesalahan langkah, proses, atau definisi data dalam program komputer.
- **Failure** : ketidak mampuan sistem atau komponen untuk menjalankan fungsinya yang telah ditetapkan dalam persyaratan kinerja.



# Perbedaan Software Error, Software Fault & Software Failure

- **Software Error** adalah bagian atau sebagian dari kode yang tidak benar (*incorrect*) akibat dari tata bahasa, logika, atau kesalahan yang dibuat oleh analis sistem, programmer, atau anggota lain bagian dari pengembang software.
- **Software Fault** adalah kesalahan software yang menyebabkan tidak berjalannya fungsi dari aplikasi tertentu.
- **Software Fault** bisa menjadi **Software Failure** ketika ada seseorang (user), mencoba mengaktifkan bagian dari software yang rusak (*fault*).
- Dapat dikatakan akar dari kegagalan software (*software failure*) adalah **software error**.



# Question

- Software xx digunakan oleh **perusahaan A** selama 3 tahun dan perusahaan A tidak pernah menemukan error software xx.
- **Perusahaan B** baru menggunakan software xx selama 2 dan menemukan banyak error dalam software xx.
- **Perusahaan C** sudah menggunakan software xx selama 4 tahun dan tidak pernah menemukan kegagalan dari software xx, tetapi 1 bulan terakhir perusahaan C menemukan beberapa error software xx.
- Apakah **software xx** dikatakan **software gagal**? **Yes, Software xx gagal (software failure)**

# Software Error

- Software error dapat menyebabkan kualitas software dikatakan buruk
- Software error?
  - error kode,
  - error prosedur,
  - error dokumentasi, atau
  - error data.

# 9 Penyebab Software Error

1. Faulty requirements definition,
2. Client–developer communication failures,
3. Deliberate deviations from software requirements,
4. Logical design errors,
5. Coding errors,
6. Non-compliance with documentation and coding instructions,
7. Shortcomings of the testing process,
8. Procedure errors, dan
9. Documentation errors.




# Fault Requirement Definition

- Kekeliruan dalam mendefinisikan kebutuhan
- Tidak adanya persyaratan yang penting
- Definisi kebutuhan yang tidak lengkap
- Adanya kebutuhan yang tidak perlu (fungsi yang tidak dibutuhkan dalam waktu dekat)



# Client-Developer Communication Failures

- Kesalahpahaman instruksi client seperti yang dinyatakan dalam dokumen persyaratan kebutuhan sistem
- Kesalahpahaman terhadap perubahan kebutuhan client secara tertulis selama proses pengembangan
- Kesalahpahaman terhadap perubahan kebutuhan client secara lisan selama proses pengembangan
- Kesalahpahaman dari respon client terhadap masalah-masalah yang disajikan oleh pengembang



# Deliberate Deviations from Software Requirement

- Pengembang menggunakan modul perangkat lunak yang digunakan pada proyek sebelumnya tanpa analisis yang memadai, yang membutuhkan adaptasi dan penyesuaian terhadap kebutuhan sistem yang sedang dikembangkan
- Tekanan waktu dan anggaran, pengembang sering memutuskan untuk menghilangkan satu bagian fungsi
- Saat proses pengembangan, terdapat perubahan yang tidak disetujui oleh client, sering kali diabaikan oleh pengembang yang dapat menyebabkan kesalahan pada software




# Logical Design Error

- Kesalahan dalam mendefinisiakan kebutuhan dapat menyebabkan algoritma perangkat lunak yang salah
- Kesalahan dapat terjadi akibat sequensial
- Kesalahan dalam membatasi kondisi sistem
- Kesalahan dalam mendefinisikan lingkungan perangkat lunak yang diperlukan
- Kesalahan dalam mencantumkan definisi dalam antarmuka perangkat lunak

# Coding Error

- Kesalahan linguistik dalam bahasa pemograman
- Kesalahan dalam penerapan kasus
- Kesalahan penggunaan tool development
- Kesalahan dalam seleksi data
- dll



# Non-Compliance with Documentation and Coding Instruction

- Hampir semua unit developer memiliki cara pembuatan dokumentasi yang tidak sama, baik dalam bentuk format maupun konten maupun penulisan kode. Hal ini dapat menyebabkan:
  - Anggota tim perlu berkordinasi terhadap modul yang dikembangkan
  - Ketika ada pergantian (promosi/pensiun) dari tim developer sebelumnya, developer baru akan kesulitan untuk memahami pekerjaannya
  - Kesulitan dalam proses menterjemahkan dalam bentuk desain
  - Kesulitan dalam proses pengujian
  - Kesulitan ketika proses perbaikan (maintenance)
- Dapat mengakibatkan pengerjaan yang tidak efisien dalam sisi waktu dan anggaran



# Shortcomings of The Testing Process

- Kelalaian saat melakukan pengujian
- Tekanan waktu
- Ketidaklengkapan pengujian terhadap semua komponen

# Procedure Error

- Prosedur digunakan untuk memberikan gambaran kegiatan pada setiap tahapan proses.
- Semakin kompleks software maka prosedur juga menjadi semakin kompleks



# Documentation Error

- Setiap software harus memiliki dokumentasi.
- Kesalahan pada dokumentasi dapat menyebabkan permasalahan dalam proses pengembangan, penggunaan dan perbaikan.

# Software Quality (IEEE, 1991)

- The degree to which a system, component, or process **meet specified requirements**
- The degree to which a system, component, or process **meets customer or user or expectations**

# Software Quality (Pressman, 2000)

- Tingkat kesesuaian dengan kebutuhan fungsional tertentu, standar kualitas perangkat lunak tertentu, dan *Good Software Engineering Practices (GSEP)*.
- Pressman menunjukkan 3 syarat yang harus dipenuhi oleh developer untuk menjamin kualitas software, yaitu:
  - *Specific functional requirement* - terutama merujuk pada output dari software.
  - *The software quality standards mentioned in the contract* – standar kualitas software harus dituliskan dalam kontrak.
  - *Good Software Engineering Practices (GSEP)* – mencerminkan state-of-the-art praktek profesional, yang harus dipenuhi oleh developer meskipun tidak disebutkan secara eksplisit dalam kontrak



# Software Quality Assessment (IEEE)

- Pola terencana dan sistematis dari semua tindakan yang diperlukan untuk memberikan keyakinan bahwa item atau produk sesuai dengan persyaratan teknis yang ditentukan.
- Satu set kegiatan yang dirancang untuk mengevaluasi proses dari suatu produk yang dikembangkan atau diproduksi.

# Software Quality Assessment (ISO 9000)

- Sistematis, Satu set rencana dari tindakan yang diperlukan untuk memberikan keyakinan bahwa proses pengembangan software atau pemeliharaan sesuai dengan yang ditetapkan pada persyaratan teknis fungsional serta persyaratan manajerial dalam menjaga jadwal dan beroperasi dalam anggaran.

2 kata yang sering muncul dalam konteks **Software**

**Quality** antara lain:

- **Quality control**
- **Quality assurance**

# Perbedaan Quality Control & Quality Assurance

- **Quality Control** adalah serangkaian kegiatan yang dilakukan dengan tujuan utama adalah menahan produk dari pengiriman jika tidak memenuhi syarat.
- **Quality assurance** dimaksudkan untuk meminimalkan biaya kualitas dengan memperkenalkan berbagai kegiatan selama proses pengembangan dan pemeliharaan untuk mencegah kesalahan, mendeteksi kesalahan dan memperbaikinya pada tahapan awal proses pengembangan.
- **Quality assurance** secara substansial mengurangi tingkat produk yang non-kualifikasi

# Tujuan dari SQA

- Menjamin, dengan tingkat kepercayaan yang dapat diterima, kesesuaian dengan persyaratan teknis fungsional.
- Menjamin, dengan tingkat kepercayaan yang dapat diterima, kesesuaian dengan persyaratan manajerial dalam hal penjadwalan dan anggaran biaya.
- Menjalankan dan mengelola kegiatan untuk perbaikan dan efisiensi dalam kegiatan pengembangan software dan kegiatan SQA .

# Software Quality & Software Engineering

- Hubungan antara **Software Quality** dan **Software Engineering** :
- Rekayasa perangkat lunak (**software engineering**) adalah aplikasi yang sistematis, disiplin, pendekatan kuantitatif untuk pengembangan, pengoperasian dan pemeliharaan software.
- Karakteristik RPL, terutama dalam pendekatan sistematis, disiplin dan kuantitatif, membuat lingkungan pengembangan yang baik dalam mencapai tujuan SQA.
- Kerjasama antara pihak pengembang (developer) dan tim SQA perlu dilakukan untuk mencapai pembangunan dan pemeliharaan kegiatan yang efisien dan ekonomis, dimana pada saat yang sama dilakukan kegiatan menjamin kualitas dari suatu produk.



**</TERIMA KASIH>**

Chalifa Chazar, S.T, M.T

Email: [chalifa.chazar@gmail.com](mailto:chalifa.chazar@gmail.com)

<http://script.id>

Copyright @2020