Permasalahan Testing dan Strateginya

SI-318 Testing dan Implementasi Sistem

Rosa Ariani Sukamto, ST

Sejarah Definisi Pengujian (*Testing*)

- 1. Establishing confidence that program does what it is supposed to do (Hetzel, 1973)
- 2. The process of executing a program or system with the intent of finding errors (Myers,1979)
- 3. Detecting specification errors and deviation from the specification
- 4. Any activity aimed at evaluating an attribute or capability of a program or system (Hetzel, 1983)
- 5. The measurement of software quality (Hetzel, 1983)
- 6. The process of evaluating a program or system
- 7. Verifying that a system satisfies its specified requirements or identifying differences between expected and actual results
- 8. Confirming that a program performs its intended function properly

• IEEE/ANSI

- 1. The process of operating a system or component under specified conditions, observing or recording the results, and making an evaluation of some aspect of the system/component
- 2. The process of analyzing a SW item to detect the difference between existing and required conditions (that is, bugs) and to evaluate the features of the SW items.

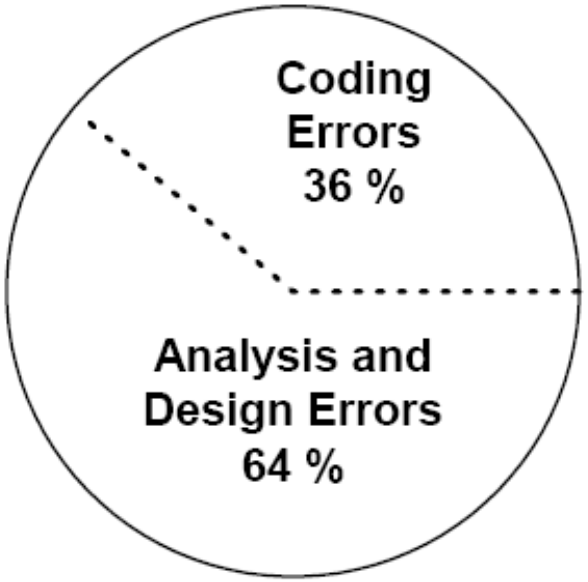
Pihak-pihak yang Terlibat Pengujian

- SW Customer
  – The Group that contracts for SW to be developed
- SW User
  – The Group that will use the SW
- SW Developer
  – The Group that develops the SW
- SW Tester
  – The Group that performs the check function on the SW
- Information Services Management
  – The group with the responsibility for fulfilling the Information Services
- Mission
  - Senior Organization Management
    – The executive
- Auditor
  - The group having the responsibility to evaluate the effectiveness, efficiency and the adequacy of control in information services area.

- **Testing is considered a control by the audit function**

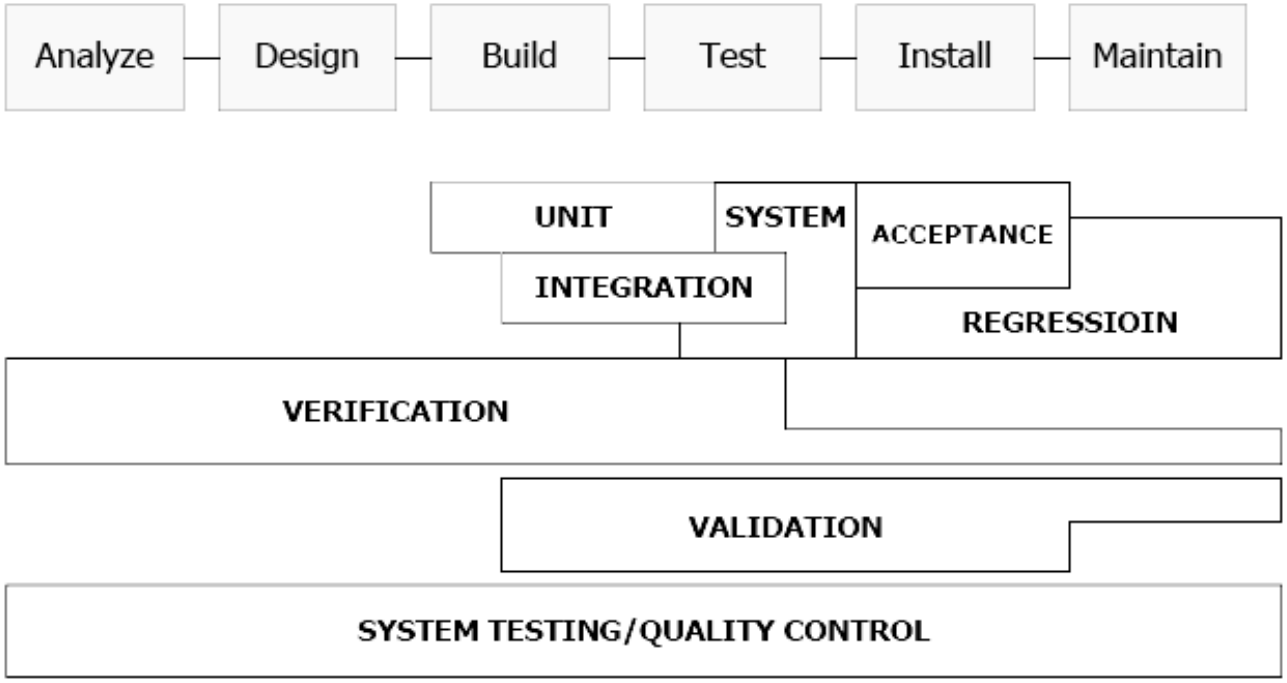
Permasalahan Umum pada Komputer

- Incomplete Design or Erroneous decision-making criteria
  - actions have been incorrect
  - inappropriate decision making criteria
- Fail to meet customer requirement
  - logic error or programming error
- Omitting needed edit checks for completeness of output data
  - Data Problems
    - Incomplete Data
    - Incorrect Data
    - Obsolete Data



Testing in Life Cycle

- Verification is the process of evaluating a system/component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase (IEEE/ANSI)
- Validation is the process of evaluating a system/component during or at the end of the development process to determine whether it satisfies specified requirements (IEEE/ANSI)

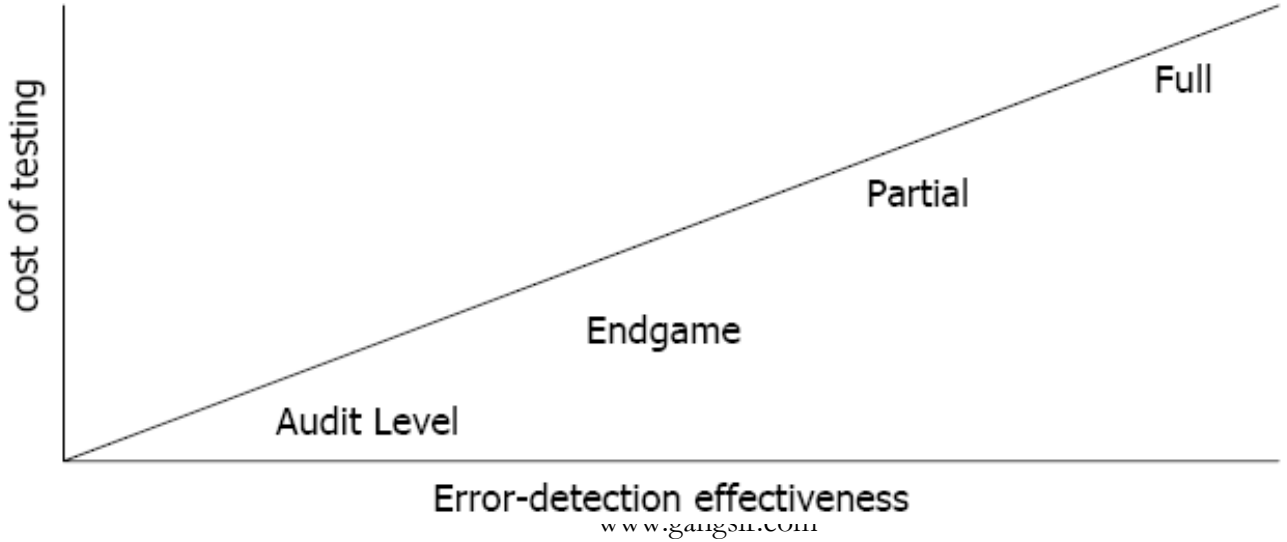


Pengujian Dasar

- Full Testing
  - starts no later than the requirement phase and continues through acceptance testing
- Partial Testing
  - begins any time after functional design has been completed, with less than optimal

influence on requirements and functional design

- Endgame Testing
  - Is highly validation oriented, with no influence on requirements or functional design
- Audit-Level Testing
  - is a barebones audit plans, procedures, and products for adequacy, correctness, and
  - compliance to standards.



Strategi Resiko Sistem Komputer

- A risk is a condition that can result in a loss
- A risk is related to the probability of a loss
- The risk is always exists, although the loss may not occur
- Risk can not be eliminated, but the impact of the loss can be reduced
- The most effective methods to reduce the impact of the loss is testing

- Types of strategic risk
  - Incorrect result
  - unauthorized transaction
  - lost of integrity of file
  - processing can not be reconstructed
  - lost of continuity of processing
  - degradation of services for user
  - security, unreliable result
  - difficulties to use and operate
  - unmaintainable program
  - not portable
  - not be able to interconnect
  - unacceptable performance level

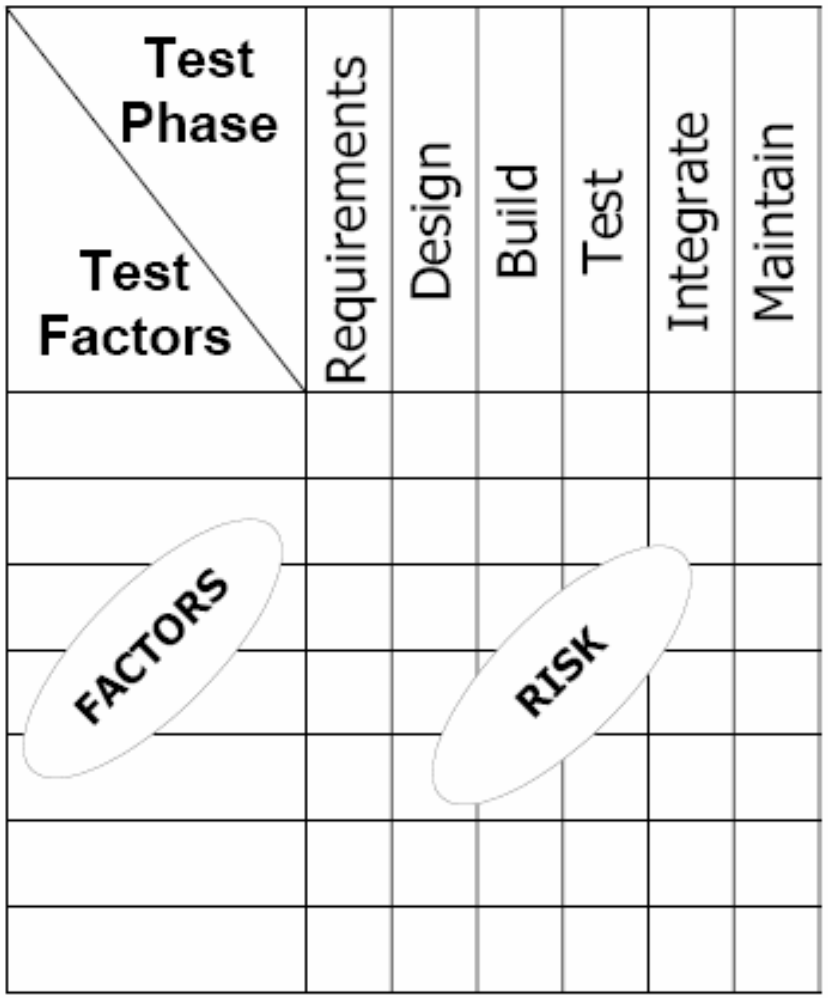
Strategi Pengujian

- A strategy must address the risks and present a process that can reduce those risks
- Two component of Testing Strategy
  - **Test Factor** - The risk or issue that needs to be addressed as part of the test strategy:

    Correctness, authorization, file integrity, audit trail, continuity of process, service levels, access control, compliance, reliability, ease of use, maintainability, portable, coupling, performance, ease of operation
  - **Test Phase** - The phase of the SDLC in which testing will occur
- Note:
  - The risk associated with testing will be called “Test Factors”
  - Not all test factors will be applicable
  - The test phase will vary based on the testing methodology used

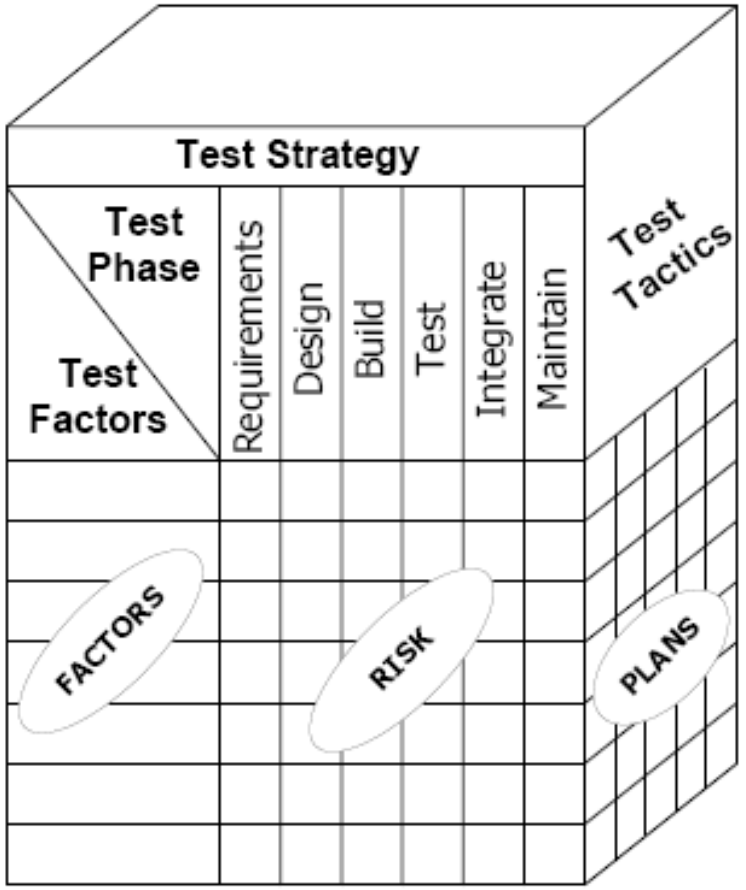
Mengembangkan Strategi Pengujian

- Select and rank test factor
- Identify the system development phase
- Identify the business risk associated with the system under development
- Place risks in the matrix



Strategical/Tactical Testing Cube

- The testing methodology cube represents a detailed work program for testing application system
- The tactics add the test plans, test criteria, testing techniques and testing tools used in validating and verifying the SW system under development
- A detailed testing work program is important to ensure that the test factors have been adequately addressed at each phase.
- The first and most important dimensions are the test factors



Developing Testing Tactics

- Acquire and study the test strategy
- Determine the type of development project
- Determine the type of software system
- Determine the project Scope
- Identify the tactical risks
- Determine when testing should occur
- Build the system test plan
- Build the unit test plans

Perancangan Kasus Pengujian

- Setiap kasus uji diidentifikasi secara unik dan diasosiasikan secara eksplisit dengan kelas yang akan diuji.
- Tujuan pengujian harus dinyatakan
- Daftar tahapan pengujian harus dibuat untuk setiap pengujian :
  - daftar keadaan dari objek yang diuji
  - daftar pesan dan operasi yang muncul
  - daftar exception yang mungkin muncul saat objek diuji
  - daftar kondisi eksternal informasi tambahan yang membantu dalam memahami pengujian atau mengimplementasikan pengujian.

Perancangan Kasus Pengujian

- **Implikasi Konsep Berorientasi Objek :**
  - Enkapsulasi menyebabkan informasi dari status objek yang sedang diuji sulit diperoleh
  - Inheritance menyebabkan perlu dilakukannya pengujian setiap reuse (jika subclass digunakan dalam konteks yang berbeda dengan superclass-nya)
  - Multi inheritance menyebabkan penambahan konteks yang harus diuji
- ***Applicability* Metoda Perancangan Kasus Uji Konvensional**
  - White Box - pengujian operasi
    - *basis path*, *loop testing* , atau *data flow* untuk memastikan bahwa setiap pernyataan dalam operasi telah diuji
  - Black Box - untuk menguji sistem
  - Use case - untuk membuat input dalam perancangan *black box* dan pengujian *state-based*

Perancangan Kasus Pengujian

- **Pengujian *Fault-Based* :**
  - Untuk merancang pengujian yang mempunyai kemungkinan besar untuk menemukan kesalahan yang masuk akal
  - Perancangan awal perlu pengujian ini dalam tahap Analisis
  - Efektivitas pengujian tergantung dari cara pandang penguji dalam melihat kesalahan
  - Pengujian integritas
    - mencari kesalahan yang masuk akal dalam *message connection*. Ada 3 jenis kesalahan yang dapat muncul dalam konteks ini :
      - hasil yang tidak diharapkan
      - kesalahan penggunaan message/operasi
      - pemanggilan yang salah

Perancangan Kasus Pengujian

- **Perancangan Pengujian Scenario-Based**
  - Pengujian *Fault-based* mengabaikan dua kesalahan utama :
    - Kesalahan spesifikasi
    - Kesalahan interaksi antar subsistem
  - Pengujian *scenario-based* memusatkan pada apa yang dilakukan oleh user, bukan pada apa yang dilakukan produk
  - *Use cases* digunakan untuk menentukan apa yang dilakukan oleh user kemudian menerapkannya sebagai pengujian
  - Pengujian *scenario-based* cenderung untuk memeriksa banyak subsistem dalam suatu pengujian tunggal.

Perancangan Kasus Pengujian

- Contoh perancangan pengujian *scenario-based* untuk text editor:
  - **Use Case** : *Fix the Final Draft*
  - **Background** : Use case ini menggambarkan urutan *event* yang muncul
    1. Cetak seluruh dokumen
    2. Periksa dokumen, ubah halaman tertentu
    3. Untuk setiap halaman yang diubah, cetak halaman tersebut
    4. Kadang-kadang sederetan halaman dicetak
  - Kebutuhan user :
    - metoda untuk mencetak satu halaman
    - metoda untuk mencetak beberapa halaman berurutan
  - Yang diuji : pengeditan setelah pencetakan
  - Penguji berharap untuk menemukan bahwa fungsi pencetakan menyebabkan kesalahan dalam fungsi pengeditan

Perancangan Kasus Pengujian

- **Testing Surface Structure dan Deep Structure**
  - **Surface Structure :**
    - struktur yang terlihat secara eksternal dari program berorientasi objek
    - ssstruktur yang langsung berhubungan dengan *end user*
    - menggunakan daftar semua objek dari semua objek sebagai *checklist* pengujian
    - Perancangan kasus uji harus menggunakan objek dan operasinya sebagai petunjuk yang menuntun pada task yang terabaikan
  - **Deep Structure :**
    - detil teknis internal dari program berorientasi objek
    - Struktur yang dimengerti dengan memeriksa perancangan dan/atau kode dirancang untuk memeriksa ketergantungan, kelakukan, dan mekanisme komunikasi yang telah dibuat