

# SOFTWARE QUALITY ASSURANCE



## Software Testing

TKB5351 – Penjaminan Mutu Perangkat Lunak

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

Perbandingan Formal Design Review, Inspection dan Walkthrough

# **REVIEW TUGAS**



Properties	Formal Design Review	Inspection	Walkthrough
Tujuan Langsung	<ol style="list-style-type: none"><li>1. Deteksi error</li><li>2. Identifikasi risiko baru</li><li>3. Persetujuan di tahapan desain</li></ol>	<ol style="list-style-type: none"><li>1. Deteksi error</li><li>2. Identifikasi penyimpangan standar</li></ol>	Deteksi error
Tujuan tidak langsung	Tambahan pengetahuan	<ol style="list-style-type: none"><li>1. Tambahan pengetahuan</li><li>2. Dukungan kegiatan perbaikan</li></ol>	Tambahan Pengetahuan
Pimpinan review	CIO atau CSE atau staff senior	Moderator	Koordinator
Anggota review	Top-level staff dan perwakilan konsumen	Kelompok (Peers)	Kelompok (Peers)
Partisipasi pimpinan proyek	Ya	Ya	Ya, biasanya pada inisiasi review



Properties	Formal Design Review	Inspection	Walkthrough
Ahli atau spesialis dalam team	-	<ol style="list-style-type: none"><li>1. Designer</li><li>2. Coder atau implementer</li><li>3. Tester</li></ol>	<ol style="list-style-type: none"><li>1. Standards enforcer</li><li>2. Maintenance expert</li><li>3. User representative</li></ol>
<b>Proses review:</b>			
Overview meeting	Tidak ada	Ada	Ada
Participants preparation	Ada	Ada	Ada
Review session	Ada	Ada	Ada
Follow-up of correction	Ada	Ada	Tidak ada



Properties	Formal Design Review	Inspection	Walkthrough
<b>Infrastruktur:</b>			
Formal training of participants	Tidak ada	Ada	Tidak ada
Use checklist	Tidak	Ya	Tidak
<b>Error-related data collection:</b>	Tidak secara langsung	Dibutuhkan	Tidak secara langsung
<b>Review documentation:</b>	Laporan formal design review	<ol style="list-style-type: none"><li>1. Laporan temuan sesi inspeksi</li><li>2. Laporan hasil pelaksanaan sesi inspeksi</li></ol>	Laporan penemuan sesi walkthrough

Software Quality Assurance – Penjaminan Mutu Kualitas PL

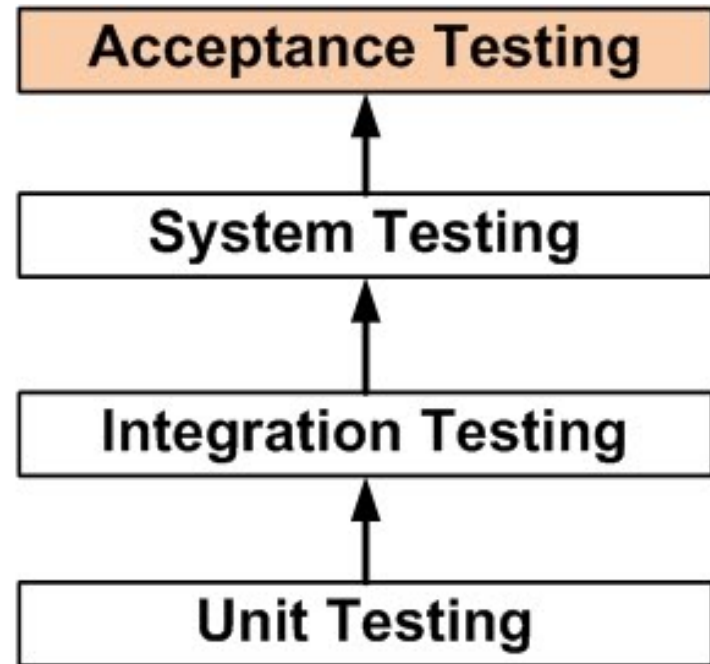
# **SOFTWARE TESTING**

# Fakta (Perry, 1995)

- Rata-rata 24% dari anggaran proyek dialokasikan untuk pengujian.
- Rata-rata 27% waktu dijadwalkan untuk kegiatan pengujian.
- Namun, kegiatan pengujian dilakukan pada akhir *development*, tekanan menjelang akhir proyek sering kali menyebabkan waktu pengujian relatif dikurangi.
- Hal ini dapat menyebabkan **kegagalan software**.

# Software Testing

Software testing merupakan salah satu bentuk penerapan untuk mengontrol kualitas suatu produk sebelum di implementasikan/digunakan user.







# Definition (IEEE Std 610.12, 1999)

- 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 or component.”
- 2) “The process of analyzing a software item to detect the differences between existing and required conditions (that is, bugs) and to evaluate the features of the software item.”

# Definition

“Software testing is a **formal** process carried out by a **specialized testing team** in which a software unit, several integrated software units or an entire software package are examined by **running the programs on a computer**. All the associated tests are performed according to **approved test procedures** on **approved test cases**.”

# Karakteristik Software Testing

- **Formal** → rencana pengujian software merupakan bagian dari proyek pengembangan dan perencanaan kualitas (*project's development and quality plans*) sesuai dengan jadwal yang di tertera dalam perjanjian kontrak.
- **Specialized testing team** → tim khusus atau konsultan eksternal yang khusus dalam bidang pengujian.
- **Running the programs** → segala bentuk kegiatan jaminan kualitas yang tidak menjalankan program, tidak dianggap sebagai pengujian.

# Karakteristik Software Testing

- **Approved test procedures** → prosedur pengujian dilakukan berdasarkan metode yang sudah diakui.
- **Approved test cases** → pelaksanaan pengujian sesuai dengan rencana.

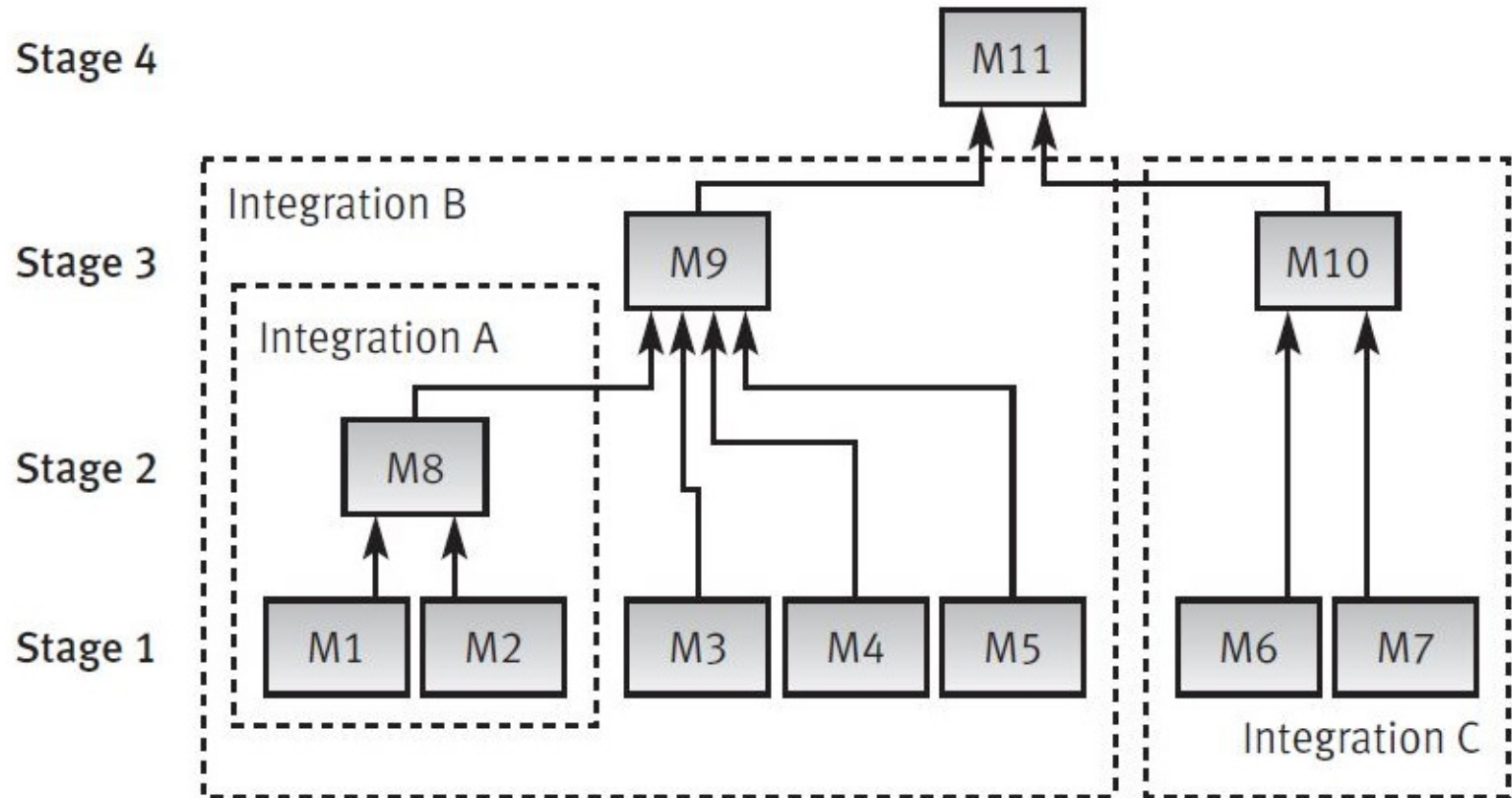
# Objectives

- **Tujuan langsung:**
  - Mengidentifikasi dan menemukan kesalahan sebanyak mungkin dari PL yang diuji.
  - Memastikan tingkat kualitas PL yang dapat diterima.
  - Melaksanakan pengujian yang diperlukan secara efektif dan efisien dalam anggaran dan keterbatasan waktu
- **Tujuan tidak langsung:**
  - Pencatatan kesalahan PL yang kemudian dapat digunakan untuk mencegah kesalahan (tindakan korektif dan preventif)

# Strategi Dasar Pengujian

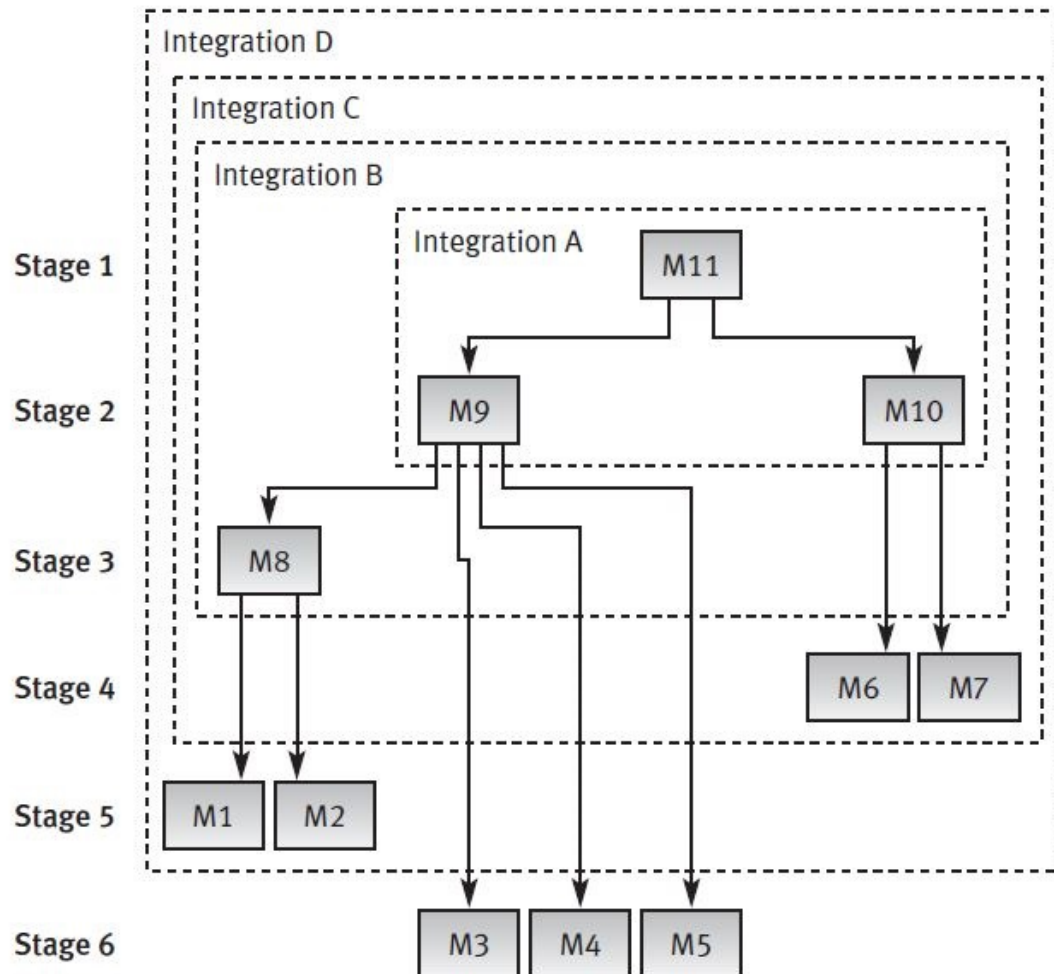
- **Big Bang Testing** → pengujian PL keseluruhan.
- **Incremental testing** → pengujian secara bertahap. Bisa secara top-down atau bottom-up.

# Bottom-up



(a) Bottom-up testing

# Top-down



(b) Top-down testing



# Bottom-up vs Top-down

- **Bottom up**
  - Keuntungan → relatif lebih mudah (kinerjanya).
  - Kelemahan → keterlambatan pemeriksaan PL secara keseluruhan.
- **Top-down**
  - Keuntungan → dapat menunjukkan keseluruhan fungsi program, sehingga memungkinkan analisa identifikasi awal kesalahan (algoritma, persyaratan fungsional,,dll)
  - Kelemahan → relatif sulit karena biasanya kompleks dan kesulitan menganalisis hasil tes.

# Big Bang vs Incremental

- Big bang lebih banyak memberikan kelemahan, karena tidak ada arahan dalam pengujian, dan relatif lebih rumit walaupun sumberdaya besar.
- Incremental lebih banyak memberikan keuntungan karena pengujian secara bertahap maka proses relatif lebih mudah (walaupun untuk software skala besar). Dan identifikasi kesalahan dan koreksi lebih mudah, walaupun dengan sumberdaya yang lebih kecil.

# Klasifikasi Pengujian

- Black box
- White box

# Black Box

- Mengidentifikasi bug hanya berdasarkan perangkat lunak berfungsi dengan baik.
- Mengabaikan mekanisme internal sistem atau komponen dan hanya berfokus pada keluaran yang dihasilkan sebagai respon terhadap input yang dipilih dan dieksekusi.
- Pengujian dilakukan untuk mengevaluasi pemenuhan sistem atau komponen dengan kebutuhan fungsional yang ditentukan.

# White Box

- Pengujian yang memperhitungkan mekanisme internal sistem atau komponen untuk mengidentifikasi bug.
- Metode ini meneliti karakteristik dasar, untuk menyelidiki kebenaran struktur kode.

**Table 9.2: White box and black box testing for the various classes of tests**

Test classification according to requirements	White box testing	Black box testing
1.1 Output correctness tests		+
1.2 Documentation tests		+
1.3 Availability (reaction time) tests		+
1.4 Data processing and calculations correctness tests	+	
1.5 Software qualification tests	+	
2. Reliability tests		+
3. Stress tests (load tests and durability tests)		+
4. Software system security tests		+
5.1 Training usability tests		+
5.2 Operational usability tests		+
6. Maintainability tests	+	+
7. Flexibility tests		+
8. Testability tests		+
9. Portability tests		+
10. Reusability tests	+	
11.1 Software interoperability tests		+
11.2 Equipment interoperability tests		+

---



**</TERIMA KASIH>**

Chalifa Chazar, S.T, M.T

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

<http://script.id>

Copyright @2020