PENGUJIAN PERANGKAT LUNAK (DPH2C2)

PROGRAM STUDI D3 MANAJEMEN INFORMATIKA – UNIVERSITAS TELKOM
SEMESTER GENAP TAHUN AKADEMIK 2016-2017

PERTEMUAN 3

MATERI : KONSEP WHITE BOX TESTING

Telkom University

White Box Testing

- White box testing... ?

- Mesin ATM

Telkom University

- White box testing...?

- Mesin dispenser kopi

Telkom University

White Box Testing

- Structural Testing or Logic-driven Testing or Glass Box Testing
- Yang dibutuhkan → Source code
- Menguji lebih "dekat" tentang detil prosedur perangkat lunak.
- Yang diselidiki: *logical path* (jalur logika) perangkat lunak

Telkom University

Mengapa 'Source Code'?

- Dengan source code, dapat dilakukan pengujian tentang:
  - Structural Testing process
  - Program Logic-driven Testing
  - Design-based Testing
  - Examines the internal structure of program

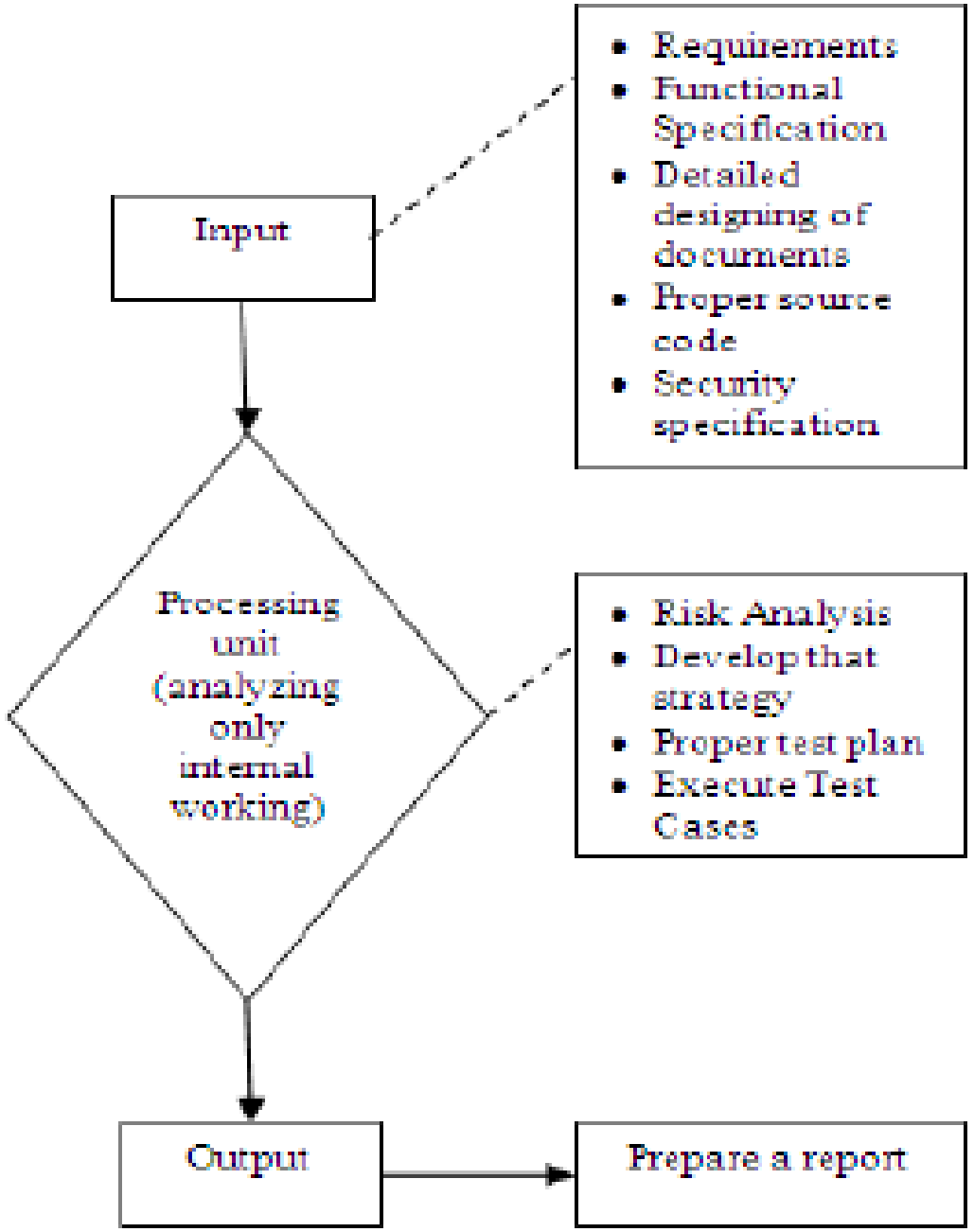
Telkom University

Tahapan dalam White Box Testing

Tahap 1	Input	• Requirement • Functional spesification • Detailed designing of documents • Proper source code
Tahap 2	Processing Unit (Analyzing internal working only)	• Perform risk analysis to guide whole testing process • Proper test plan • Execute test cases and communicate results
Tahap 3	Output (Prepare final report)	• Test cases document and result • Acceptance test document

Telkom University

Working Process pada WBT



Telkom University

Logical Path

- Conditions
  - If .. Then ..
  - If .. Then .. Else ..
  - If .. Then .. Else if .. Then ..
  - Case .. Of ..
- Loop
  - While .. Do ..
  - Repeat .. Until ..
  - For .. To .. Do ..

Telkom University

Keuntungan

- Seringkali *programmer* melakukan kesalahan-kesalahan umum pada saat membangun perangkat lunak. Keunggulan dari *white box testing* adalah kemampuannya untuk mendeteksi kesalahan umum tersebut, yaitu:
  - Kesalahan logika *(logic errors)*
  - Ketidaksesuaian asumsi *(incorrect assumptions)*
  - Kesalahan lain yang juga sering terjadi adalah kesalahan dalam menuliskan kode program atau sering disebut sebagai 'salah ketik' *(typographical errors)*.

Syarat Kesempurnaan Hasil WBT

- Mendefinisikan semua logical path
- Membangun kasus untuk pengujian
- Mengevaluasi hasilnya
- Menguji secara **menyeluruh**

Telkom University

Namun...

- Pengujian secara menyeluruh justru menimbulkan masalah sumber daya
- Program yang kecil bisa menghasilkan banyak sekali jalur logika, contoh:
  - Terdapat 1 (satu) buah loop yang berulang sebanyak 20 kali
  - Di dalamnya terdapat nested if yang terdiri dari 4 set pernyataan if..then..else
  - Menghasilkan 10^{14} jalur

    10^{14} = 100.000.000.000 (seratus triliun)

Telkom University

Masih ‘namun’...

- Setiap jalur harus diuji secara “manual” untuk dibuktikan kebenarannya.
- Manual:
  - Menulis
  - Mengeksekusi
  - Memverifikasi hasil
- 1 jalur logika = 1 kegiatan manual

Telkom University

Lagi-lagi ‘namun’...

- Jika 1 manual dilakukan selama 5 menit, maka:
  5 menit X 100 triliun = ± 1 miliar tahun!

- Jika 300 kali lebih cepat (1 manual = 1 detik), maka:

  1 detik X 100 triliun = ± 3.2 juta tahun

Telkom University

Pertanyaan

"Bukankah *black box testing* jauh lebih cepat dan lebih mungkin dilakukan?"

- Jawaban terletak pada ketidak-sempurnaan perangkat lunak:
  - Adanya kesalahan logika (logic errors)

    65 < x <=75 → if (x < 65) AND (x <= 75) then…
  - Adanya ketidaksesuasian asumsi (incorrect assumptions)

    1 bulan = 30 hari → bulan Februari? Juli? dst…
  - Adanya kesalahan menulis kode (typographical errors)

    volume:= P * L * T; → polume:=P * L * T;

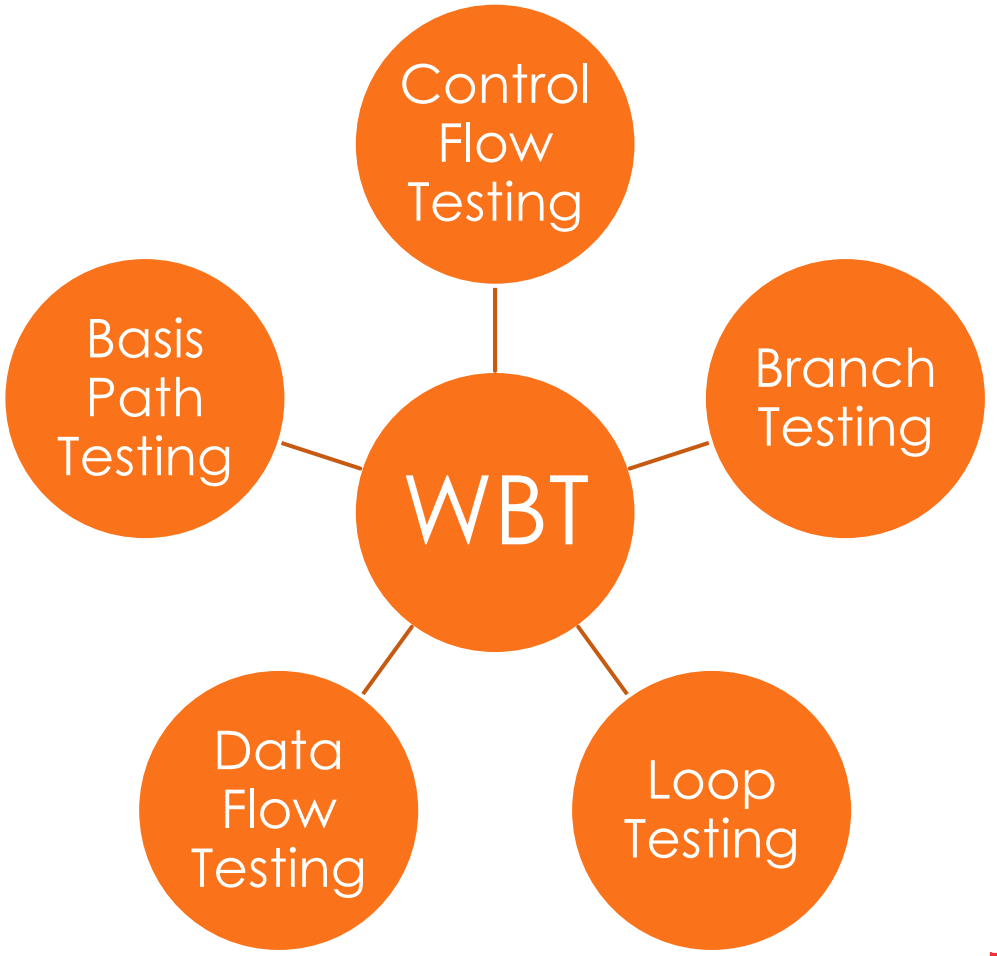
Telkom University

Mungkinkah dilakukan?

- Ya!
- Tidak dilakukan secara menyeluruh.
- Cukup dilakukan pada jalur logika yang penting.
- Kombinasikan dengan *black box testing*.

Telkom University

Beberapa Bentuk WBT



Telkom University