

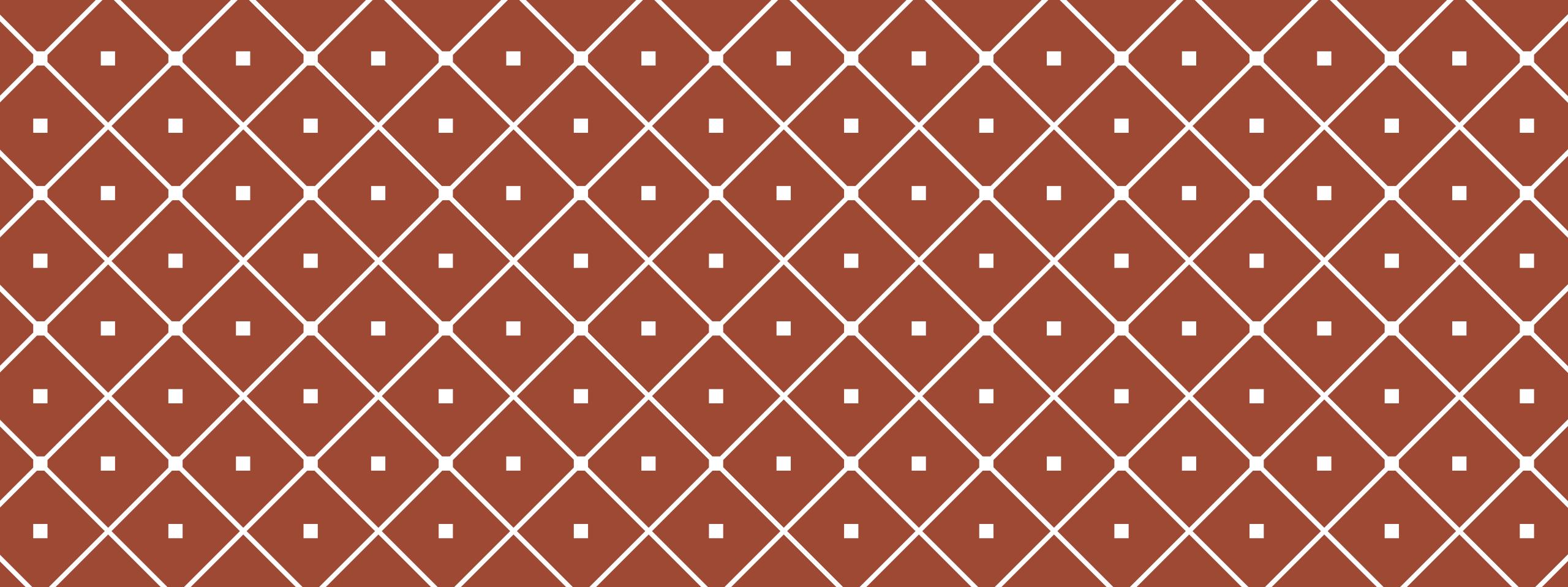


PENGUJIAN PERANGKAT LUNAK

-KONSEP WHITEBOX TESTING-

Eka Widhi Yunarso (EWD)
2017-2

*Digunakan dilingkungan internal prodi
D3 Manajemen Informatika, Fakultas
Ilmu Terapan, Universitas Telkom*



White Box Testing



White box testing... ?

Mesin ATM



White box testing...?

Mesin dispenser kopi



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



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

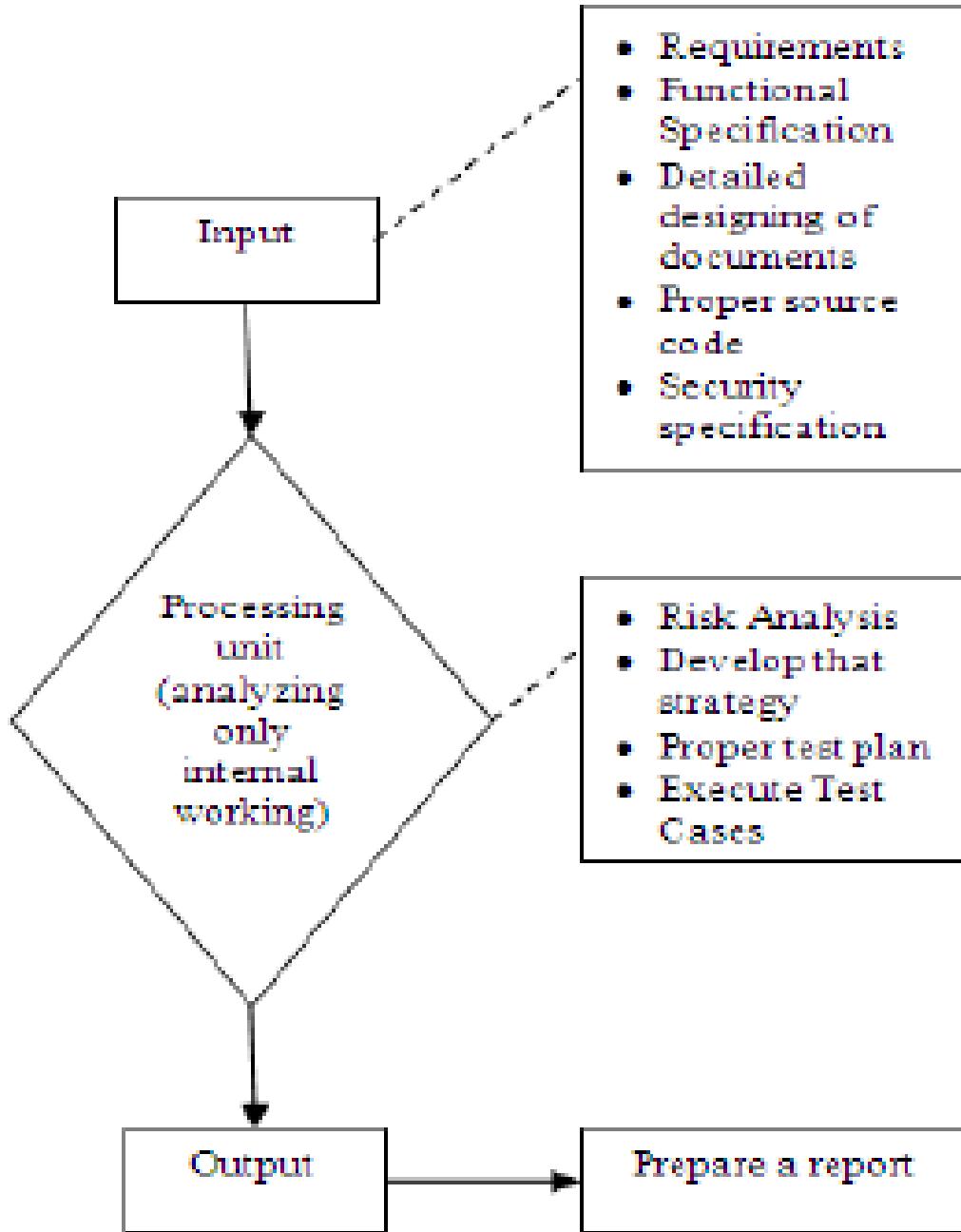


TAHAPAN DALAM WHITE BOX TESTING

Tahap 1	Input	<ul style="list-style-type: none">• Requirement• Functional spesification• Detailed designing of documents• Proper source code
Tahap 2	Processing Unit (Analyzing internal working only)	<ul style="list-style-type: none">• Perform risk analysis to guide whole testing process• Proper test plan• Execute test cases and communicate results
Tahap 3	Output (Prepare final report)	<ul style="list-style-type: none">• Test cases document and result• Acceptance test document



WORKING PROCESS PADA WBT



LOGICAL PATH

Conditions

- If .. Then ..
- If .. Then .. Else ..
- If .. Then .. Else if .. Then ..
- Case .. Of ..

Loop

- While .. Do ..
- Repeat .. Until ..
- For .. To .. Do ..



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



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

$$10^{14} = 100.000.000.000 \text{ (seratus triliun)}$$



MASIH ‘NAMUN’...

Setiap jalur harus diuji secara “manual” untuk dibuktikan kebenarannya.

Manual:

- Menulis
- Mengeksekusi
- Memverifikasi hasil

1 jalur logika = 1 kegiatan manual



LAGI-LAGI ‘NAMUN’...

Jika 1 manual dilakukan selama 5 menit, maka: $5 \text{ menit} \times 100 \text{ triliun} = \pm 1 \text{ miliar tahun!}$

Jika 300 kali lebih cepat (1 manual = 1 detik), maka:
 $1 \text{ detik} \times 100 \text{ triliun} = \pm 3.2 \text{ juta tahun}$



PERTANYAAN

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

Jawaban terletak pada ketidak-sempurnaan perangkat lunak:

- Adanya kesalahan logika (logic errors)
 $65 < x \leq 75 \rightarrow \text{if } (x < 65) \text{ AND } (x \leq 75) \text{ then...}$
- Adanya ketidaksesuaian asumsi (incorrect assumptions)
1 bulan = 30 hari \rightarrow bulan Februari? Juli? dst...
- Adanya kesalahan menulis kode (typographical errors)
`volume:= P * L * T;` \rightarrow `polume:=P * L * T;`



MUNGKINKAH DILAKUKAN?

Ya!

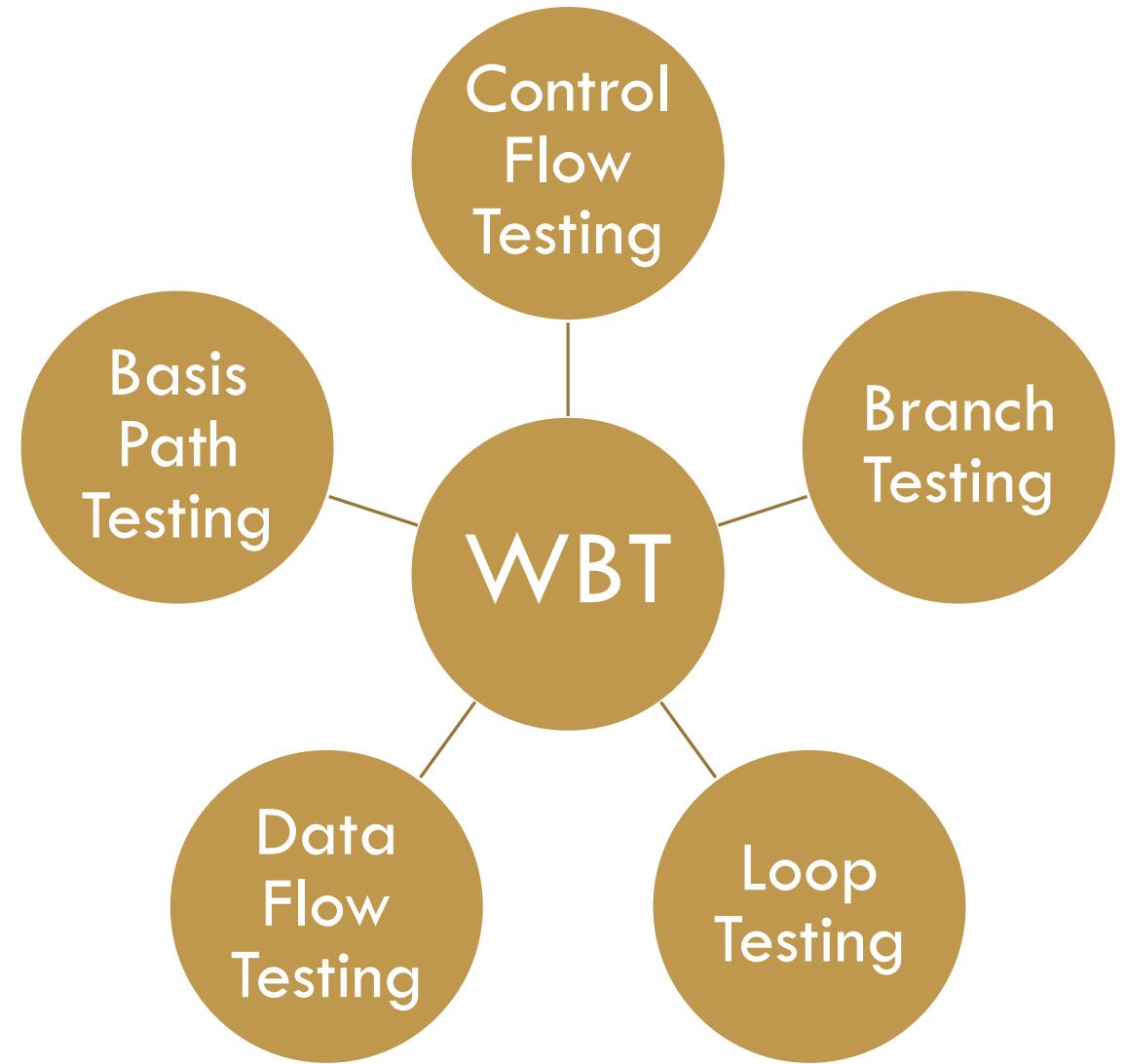
Tidak dilakukan secara menyeluruh.

Cukup dilakukan pada jalur logika yang penting.

Kombinasikan dengan *black box testing*.



BEBERAPA BENTUK WBT



REFERENCES

- <http://softwaretestingfundamentals.com>
- Pressmann, R.S (2010). Software Engineering A Practitioner's approach. New York: McGraw-Hill.
- Agus Pratondo, d. (2009). Jaminan Mutu Sistem Informasi. Bandung: Politeknik Telkom.
- Eka Widhi Yunarso (2013). Student Workbook Jaminan Mutu Sistem Informasi. Yogyakarta: DeePublished

