

## Software Testing Strategies

Oleh :  
Ir. I Gede Made Karma, MT

1

## What Is It?

Perancangan kasus untuk pengujian efektif adalah penting, begitu juga strategi yang dipergunakan untuk mengeksekusinya.

1. Perlukah kita mengembangkan rencana formal pengujian?
2. Perlukah kita menguji keseluruhan program atau cukup sebagian kecil saja?
3. Perlukah kita mengulangi pengujian yang telah dilakukan ketika kita menambahkan komponen baru pada sistem yang besar?
4. Perlukah kita melibatkan pelanggan?

→ Terjawab pada saat mengembangkan strategi pengujian software.

2

## Who Does It?

Sebuah strategi pengujian software dikembangkan oleh :

1. Manajer proyek,
2. Software engineer,
3. Spesialis pengujian.

3

## Why Is It Important?

Pengujian sering mendapat perhatian lebih dibandingkan dengan aktivitas lain.

Apabila pengujian dilaksanakan asal-asalan, akan buang-buang waktu, perlu usaha lebih, dan tidak ada gunanya dan kesalahan tidak tersentuh.

4

## What Are The Step?

Pengujian diawali dengan yang kecil dan berlanjut ke yang besar. Artinya :

1. Awal pengujian difokuskan pada satu komponen dan menerapkan pengujian *white-* dan *black-box* untuk mengatasi kesalahan logik dan fungsi program.
2. Pengujian dilanjutkan pada komponen yang terintegrasi.
3. Pengujian sebagai kesatuan software.
4. Diakhiri dengan sekumpulan pengujian yang dilakukan sekali ketika program lengkap dioperasikan. Pengujian ini dirancang untuk mengatasi kesalahan *requirement*.

5

## What Is The Work Product?

Sebuah Spesifikasi Pengujian yang mendokumentasikan pendekatan pengujian yang berisi sebuah rencana yang menggambarkan keseluruhan strategi dan prosedur tentang tahapan pengujian yang spesifik dan pengujian yang akan dilaksanakan.

6

## How Do I Ensure That I've Done It Right?

Dengan mengkaji Spesifikasi Pengujian sebelum pengujian, kita dapat menilai kelengkapan kasus pengujian dan kegiatan pengujian. Rencana dan prosedur pengujian yang efektif akan mendahului tahapan pembangunan software dan menemukan kesalahan pada setiap tahapan proses pembangunan.

7

## Pendekatan Strategis

Pengujian adalah sekumpulan aktifitas yang dapat direncanakan dan dilaksanakan secara sistematis. Karakteristik dari strategi pengujian software :

1. Pengujian dimulai pada tingkatan komponen dan melebar ke integrasi dari keseluruhan sistem berbasis komputer.
2. Teknik pengujian yang berbeda untuk poin yang berlainan.
3. Pengujian dilakukan oleh pengembang software dan grup penguji independen.
4. Pengujian dan *debugging* adalah aktifitas berbeda, tetapi *debugging* harus diakomodasikan dalam beberapa strategi pengujian.

8

## Verifikasi dan Validasi

- Verifikasi : Apakah kita membangun produk dengan benar?  
→ mengacu pada sekumpulan aktifitas yang menjamin bahwa software secara benar mengimplementasikan fungsi khusus.
- Validasi : Apakah kita membangun produk yang benar?  
→ mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa software yang dibangun sesuai dengan keinginan pelanggan.

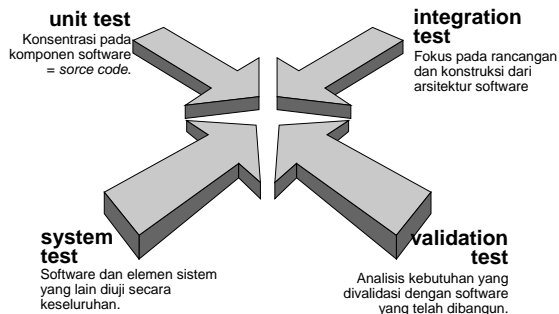
9

## Pengorganisasian Pengujian Software

1. Bahwa pengembang software sebaiknya tidak melakukan pengujian secara keseluruhan.
  2. Bahwa software sebaiknya di "tossed over the wall" kepada orang asing yang akan mengujinya tanpa ampun.
  3. Bahwa penguji terlibat dalam proyek hanya ketika tahap pengujian akan dimulai.
- Pernyataan di atas adalah keliru.
1. Pengembang bertanggung jawab pada pengujian setiap komponen program, menjamin masing-masing melaksanakan fungsi seperti yang dirancang.
  2. Grup penguji independen berfungsi untuk menghilangkan konflik kepentingan dalam pengembang.
  3. Pengembang dan GPI bekerja secara dekat selama proyek.

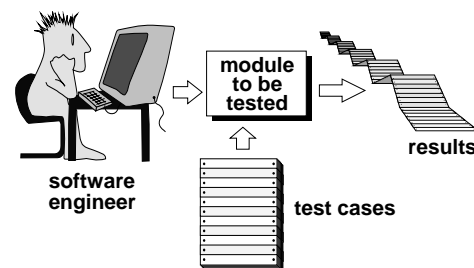
10

## Strategi Pengujian



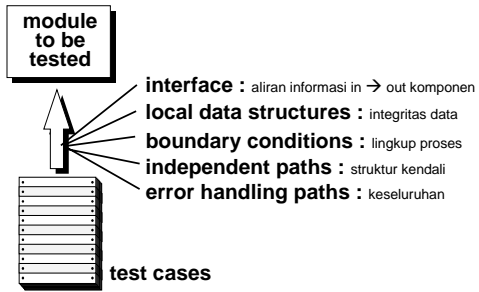
11

## Unit Testing (1)



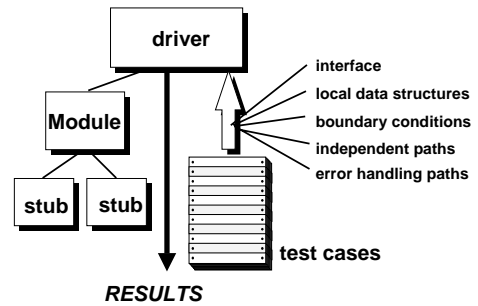
12

## Unit Testing (2)



13

## Unit Test Environment



14

## Strategi Pengujian Integrasi

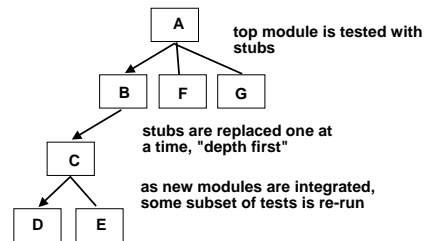
Pilihan :

- pendekatan "big bang"
- strategi pembangunan bertahap



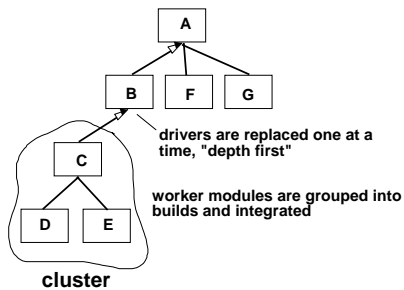
15

## Top Down Integration



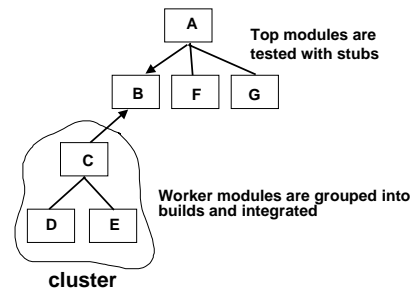
16

## Bottom-Up Integration



17

## Sandwich Testing



18

## High Order Testing

- validation test
- system test
- alpha and beta test
- other specialized testing

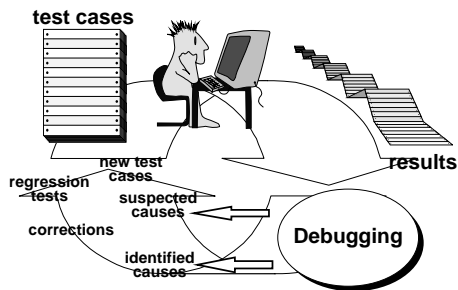
19

## Debugging: A Diagnostic Process



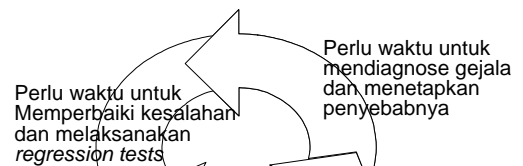
20

## The Debugging Process



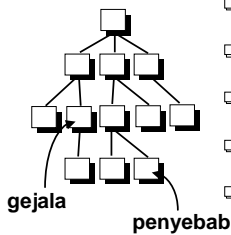
21

## Debugging Effort



22

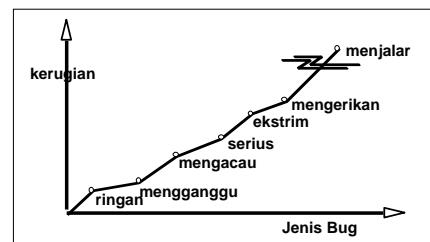
## Gejala & Penyebab



- gejala dan penyebab dapat terpisah secara lokasi
- gejala bisa hilang ketika masalah lain dibetulkan
- penyebab dapat berupa kombinasi dari bukan kesalahan
- penyebab dapat berupa kesalahan sistem atau kompuler
- penyebab dapat berupa asumsi yang dipercaya semua orang
- gejala dapat terjadi berselang

23

## Konsekuensi dari Bugs



**Bug Categories:** function-related bugs, system-related bugs, data bugs, coding bugs, design bugs, documentation bugs, standards violations, etc.

24

## Debugging Techniques

- ❑ brute force / testing
- ❑ backtracking
- ❑ induction
- ❑ deduction

25

## Debugging: Final Thoughts

1. Don't run off half-cocked, think about the symptom you're seeing.
2. Use tools (e.g., dynamic debugger) to gain more insight.
3. If at an impasse, get help from someone else.
4. Be absolutely sure to conduct regression tests when you do "fix" the bug.

26