

Pengujian Perangkat Lunak

Disusun Oleh :

Irman Hariman, M.T

iirmanhariman@gmail.com

Definisi Pengujian

- Menurut Hetzel 1973:
Testing ialah proses pemantapan kepercayaan akan kinerja program atau sistem sebagaimana yang diharapkan.
- Menurut Myers 1979:
Testing ialah proses eksekusi program atau sistem secara intens untuk menemukan error.
- Menurut Hetzel 1983 (Revisi):
Testing adalah tiap aktivitas yang digunakan untuk dapat melakukan evaluasi suatu atribut atau kemampuan dari program atau sistem dan menentukan apakah telah memenuhi kebutuhan atau hasil yang diharapkan.
- Menurut Standar ANSI/IEEE 1059:
Testing adalah proses menganalisa suatu entitas software untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan (defects / errors / bugs) dan mengevaluasi fitur-fitur dari entitas software



Pengujian / testing

- *Testing is the exposure of a system to trial input to see whether it produces correct output*
- *Adalah proses eksekusi suatu program dengan maksud menemukan kesalahan*
- Testing merupakan tugas yang tak dapat dihindari di tiap bagian dari tanggung jawab usaha pengembangan suatu sistem software

William Howden



Pengujian Perangkat Lunak

- *Elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain dan pengkodean*



Pengertian Pengujian

- Pengujian Perangkat Lunak adalah proses menjalankan dan mengevaluasi sebuah PL secara manual maupun otomatis untuk menguji apakah PL sudah memenuhi persyaratan atau belum atau untuk menentukan perbedaan antara hasil yang diharapkan dengan hasil sebenarnya.
- Pelaksanaan pengujian PL biasanya disesuaikan dengan metodologi pembangunan PL yang digunakan.
- Pada umumnya pengujian dilakukan sesudah tahap pemograman,namun demikian perencanaan pengujian dilakukan mulai tahap analisis.



Pengujian

- Pengujian Perangkat Lunak merupakan suatu tahapan penting dalam pembangunan perangkat lunak.
- Pengujian dilakukan dengan cara mengevaluasi kofigurasi perangkat lunak yang terdiri dari spesifikasi kebutuhan, deskripsi perancangan dan program yang dihasilkan.
- Hasil evaluasi kemudian dibandingkan dengan hasil uji yang diharapkan.
- Jika ditemukan kesalahan maka perbaikan perangkat lunak harus dilakukan untuk kemudian diuji kembali.



Peran Pengujian

Hal ini terkait dengan aktivitas apapun yang menghasilkan produk. Perannya sebagai proses perubahan.

Terdapa tiga kategori proses :

- Manufacturing : menghasilkan produk sejenis
- Job shop : banyak karakteristik yang sama
- Professional : produk unik

Setiap penyimpangan yang dihadapi oleh tester ini disebut 'cacat'



Tujuan Pengujian PL

- Menilai apakah perangkat lunak yang dikembangkan telah memenuhi kebutuhan pemakai
- Menilai apakah tahap pengembangan perangkat lunak telah sesuai dengan metodologi yang digunakan
- Membuat dokumentasi hasil pengujian yang menginformasikan kesesuaian perangkat lunak yang diuji dengan spesifikasi yang telah ditentukan



Yang Harus Dilakukan Saat Pengujian

- Mengidentifikasikan dan menemukan beberapa kesalahan yang mungkin ada dalam Perangkat Lunak yang diuji
- Setelah Perangkat Lunak dibetulkan, diidentifikasi ulang kesalahan dan dites ulang untuk menjamin kualitas level penerimaan
- Membentuk tes yang efisien dan efektif dengan anggaran dan jadwal yang terbatas
- Mengumpulkan daftar kesalahan untuk digunakan dalam daftar pencegahan kesalahan (tindakan corrective dan preventive)



Fokus Pengujian PL

1. Kualitas dari proses pengetesan menentukan kesuksesan pengujian
2. Mencegah perambatan cacat dengan menguji pada seluruh siklus hidup perangkat lunak
3. Menggunakan alat pengujian/Testing tools
4. Harus ada seseorang ahli yang bertanggung jawab untuk memperbaiki pross pengujian
5. Pengujian adalah disiplin professional yang membutuhkan orang terlatih dan berkeahlian
6. Menumbuhkan tim yang bersikap positif



Kategori Cacat

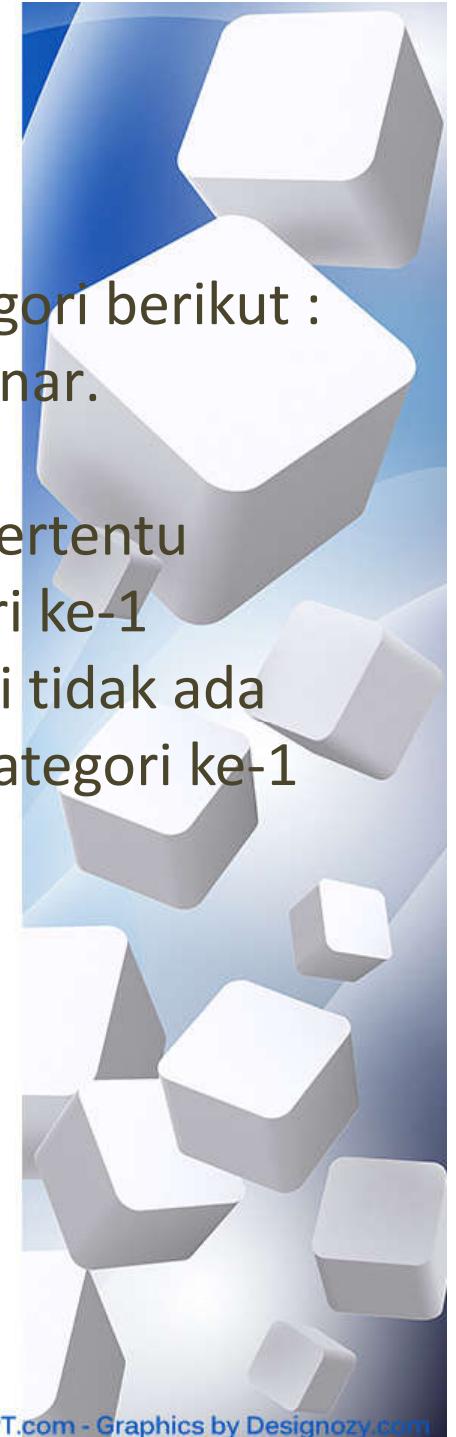
- Cacat dari spesifikasi : produk dibangun menyimpang dari produk tertentu.
- Macam-macam pengguna/harapan dari pelanggan: Pengguna ingin tetapi dalam produk tersebut tidak ada spesifikasi yang diinginkan.
- Sebuah cacat yang menyebabkan kesalahan dalam operasi disebut kegagalan



Cacat Perangkat Lunak (DEFECT)

Defect pada umumnya berupa salah satu dari 3 kategori berikut :

1. Wrong (salah) : Spesifikasi dibuat secara tidak benar.
Merupakan cacat kategori ke-2
2. Missing (Hilang/kurang) : spesifikasi kebutuhan tertentu tidak ada pada produk. Merupakan cacat kategori ke-1
3. Extra : Fungsi yang terdapat dalam produk, tetapi tidak ada dalam spesifikasi kebutuhan. Merupakan cacat kategori ke-1



Defect & Failure - Cacat & Kegagalan

- Defect : merupakan cacat/kekurangan (wrong/missing/extraneous) didalam sistem software, namun tidak memiliki dampak apabila tidak mempengaruhi customer/user dan sistem operasional organisasi.
- Failure : cacat yang menyebabkan error pada operasi atau secara negatif memberikan dampak pada user/customer.
- Defect dapat berubah menjadi Failure yang dapat menyebabkan gangguan/kerusakan pada organisasi



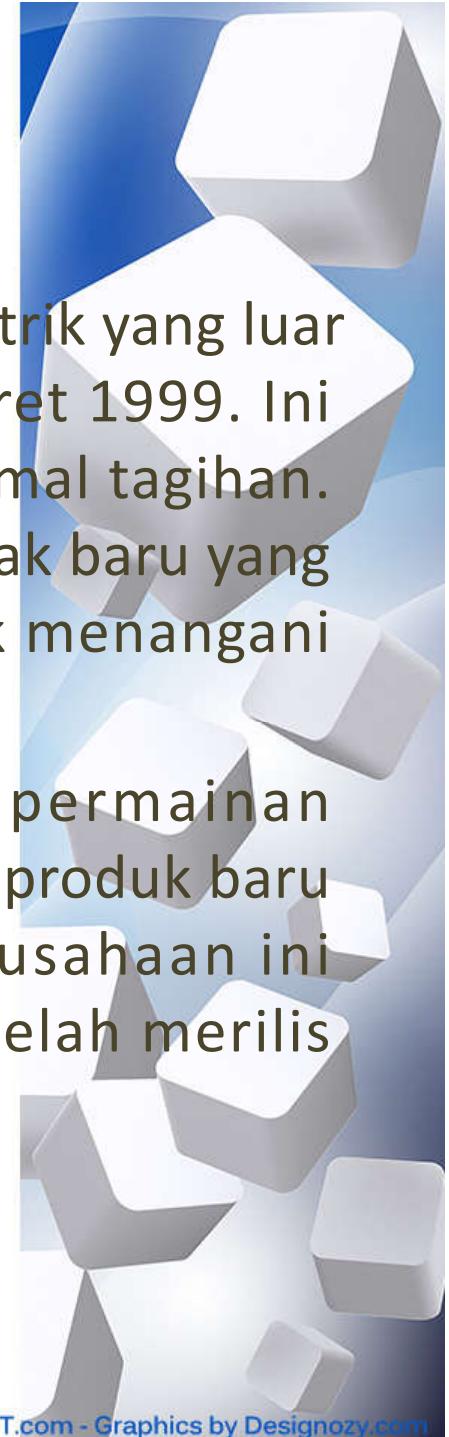
Kegagalan PL

- Pada awal tahun 2000, masalah utama yang dilaporkan dengan sistem komputer baru di sebuah sekolah daerah pinggiran kota besar publik AS dengan 100.000 + siswa; masalah termasuk 10.000 kartu laporan yang keliru dan murid terbengkalai oleh karena kegagalan sistem pendaftaran kelas; distrik tersebut CIO dipecat
- Pada bulan Oktober 1999 \$ 125,000,000 NASA Orbiter Mars Iklim - pesawat ruang angkasa itu diyakini akan hilang dalam ruang karena kesalahan konversi data sederhana.
- Mars Polar Lander misi, yang gagal karena alasan-alasan yang tidak diketahui pada bulan Desember 1999. Beberapa panel menyelidiki untuk menentukan kegagalan proses yang memungkinkan kesalahan yang terjadi.



Kegagalan PL

- Sebuah kota kecil di Illinois menerima tagihan listrik yang luar biasa besar bulanan sebesar \$ 7 juta pada Maret 1999. Ini tentang 700 kali lebih besar daripada yang normal tagihan. Ternyata terjadi karena bug dalam perangkat lunak baru yang telah dibeli oleh perusahaan listrik lokal untuk menangani masalah software Y2K.
- Pada awal tahun 1999 sebuah perusahaan permainan komputer besar menarik kembali semua salinan produk baru yang populer karena masalah software. Perusahaan ini membuat permintaan maaf ke publik karena telah merilis produk yang belum siap.



Hartford Coliseum Collapse (1978)



- Setelah beberapa jam ribuan fans keluar dari coliseum, atap tempat tersebut runtuh akibat beban timbunan salju.
- Terjadi akibat kesalahan oleh programmer CAD software yang digunakan untuk men-design coliseum tersebut
- Kerugian sebesar \$ 70 juta, dan \$ 20 juta untuk ekonomi lokal.



CIA Gives the Soviets Gas (1982)



- Kendali software menjadi tak terkendali dan menghasilkan tekanan intens pada pipa gas Trans-Siberia sehingga menghasilkan ledakan non-nuklir terbesar dalam sejarah
- Hal ini diakibatkan sabotase software oleh CIA terhadap software Soviet saat itu.



Medical Machine Kills (1985)



- Mesin terapi radiasi Canada's Therac-25 menjadi malfungsi dan memberikan radiasi lebih pada pasien.
- Diakibatkan adanya bug dalam software pengendali mesin tersebut.
- 3 orang meninggal, 3 orang luka parah.



Patriot Fails Soldiers (1991)



- Dalam masa Perang Gulf pertama, sebuah sistem American Patriot Missile di Saudi Arabia gagal “menepis” kedatangan misil Iraqi Scud. Misil tersebut menghancurkan sebuah beberapa barrack.
- Hal ini diakibatkan kesalahan kalkulasi kedatangan misil.
- 28 tentara meninggal, 100 tentara luka parah.



Ariane Rocket Goes Boom (1996)



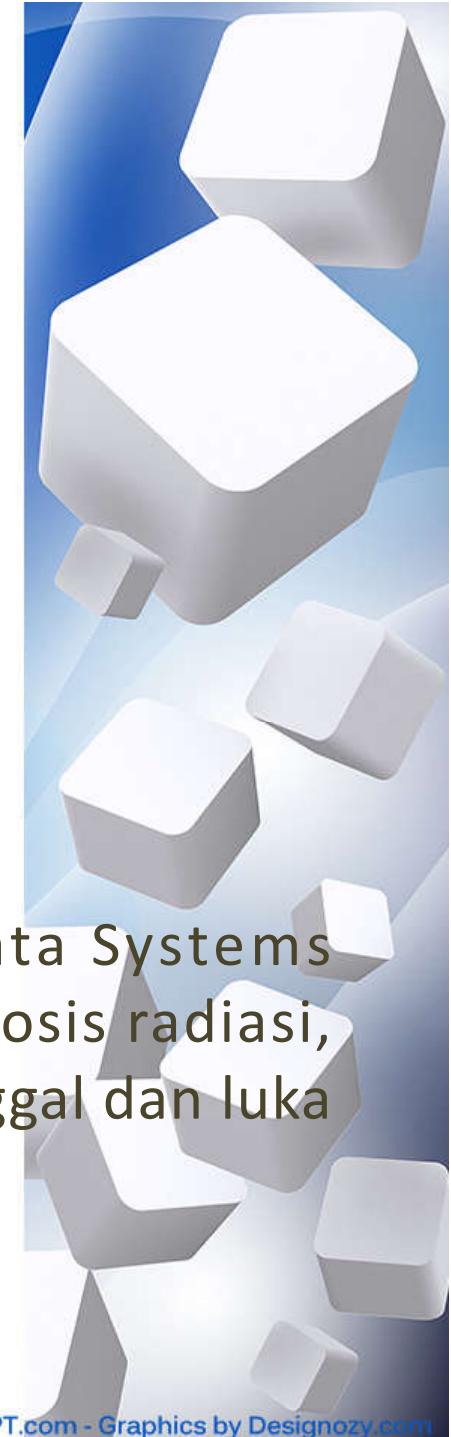
- Ariane 5, roket terbaru eropa, tiba-tiba hancur beberapa detik setelah peluncuran perdananya. Juga menghancurkan 4 satelit yang digunakan untuk mempelajari interaksi magnet bumi dengan *solar wind*.
- Diakibatkan sistem komputer pemandu tiba-tiba shut down.



Cancer Treatment to Die For (2000)



- Delapan orang meninggal, 20 orang luka parah.
- Radiation therapy software buatan Multidata Systems International salah dalam memperhitungkan dosis radiasi, sehingga menyebabkan beberapa pasien meninggal dan luka parah.



Pengujian SDLC

- Studi di IBM mengungkapkan bahwa pengujian sebelum pengkodean adalah 50% efektif; setelah coding adalah 80% efektif.
- Memperbaiki kesalahan setelah pengkodean adalah 10 kali lebih mahal seperti sebelumnya, dan 100 kali lebih mahal untuk memperbaiki kesalahan produksi.
- Oleh karena itu, lebih ekonomis untuk menggunakan pengujian SDLC, karena diketahui bahwa 64% dari kesalahan terjadi selama tahap analisis dan desain.
- Requirements (persyaratan):
 - bahwa persyaratan yang ditentukan memenuhi kebutuhan organisasi
- Design and programs (desain dan program) :
 - bahwa desain dan menyelesaikan program telah memenuhi persyaratan yang ditentukan



Pengujian SDLC

- Test and installation (pengujian dan instalasi) :
 - Bahwa sistem yang diterapkan memenuhi spesifikasi
- Maintenance (pemeliharaan) :
 - Perubahan pekerjaan



Setiap Tahap Dilakukan

- Menganalisis struktur yang diproduksi pada fase ini untuk testability internal dan kecukupan
- Menghasilkan pengujian set berdasarkan struktur pada tahap ini
- Selain itu, selama desain dan pemrograman
 - Menentukan bahwa struktur yang konsisten dengan struktur yang dihasilkan selama tahap sebelumnya
 - Menyempurnakan dan merumuskan kembali set pengujian yang dihasilkan sebelumnya



Verifikasi Kegiatan

- Requirements (persyaratan) :
 - Menentukan pendekatan verifikasi
 - Menentukan kecukupan persyaratan
 - Menghasilkan data uji fungsional
 - Menentukan konsistensi desain dengan persyaratan
- Design (desain) :
 - Menentukan kecukupan desain
 - Menghasilkan data pengujian struktural dan fungsional
 - Menentukan konsistensi dengan desain
- Programming/Coding (pemrograman) :
 - Menentukan kecukupan pelaksanaan
 - Menghasilkan data pengujian struktural & fungsional utk program
- Test (pengujian) : Pengujian sistem aplikasi
- Installation (instalasi) : Sistem diuji dalam produksi
- Maintenance (pemeliharaan) : Memodifikasi dan pengujian ulang



Pengujian Waterfall

Tahap pengujian meliputi pengujian tahap analisis, perancangan, implementasi,intalasi dan pemeliharaan.

Prinsip pengujian :

1. Dapat dilacak hingga ke persyaratan atau dokumen SRS.
2. Pengujian harus direncanakan sebelum pelaksanaan pengujian.
3. Pengujian harus dimulai dari hasil yang kecil, diteruskan ke hal-hal yang besar.
4. Pengujian yang berlebihan tidak akan mungkin dilaksanakan.
5. Pengujian sebaiknya dilakukan pihak ketiga.



Tahap Pengujian

1. Tentukan apa yang akan diukur melalui pengujian.
2. Bagaimana pengujian akan dilaksanakan.
3. Membangun suatu kasus uji (test case) yaitu sekumpulan data atau situasi yang akan digunakan dalam pengujian.
4. Tentukan hasil yang akan diharapkan atau hasil yang sebenarnya.
5. Jalankan Kasus pengujian.
6. Bandingkan hasil pengujian dan hasil yang diharapkan.



Pengujian Tahap Analisis

- Pengujian pada tahap analisis ditekankan pada validasi terhadap kebutuhan, untuk menjamin bahwa kebutuhan telah dispesifikasi dengan benar.
- Tujuan pengujian pada tahap ini adalah untuk mendapatkan kebutuhan yang layak dan untuk memastikan apakah kebutuhan tersebut sudah dirumuskan dengan baik.



Faktor-faktor pengujian yang dilakukan pada tahap analisis :

1. Kebutuhan yang berkaitan dengan metodologi.
2. Pendefenisian spesifikasi fungsional
3. Penentuan spesifikasi kegunaan.
4. Penentuan kebutuhan portabilitas.
5. Pendefenisian antar muka sistem.



Pengujian Tahap Perancangan

- Pengujian tahap perancangan bertujuan untuk menguji struktur perangkat lunak yang diturunkan dari kebutuhan.
- Kebutuhan yang bersifat umum dirinci menjadi bentuk yang lebih spesifik.



Faktor-faktor pengujian yang dilakukan pada tahap perancangan :

1. Perancangan yang berkaitan dengan kebutuhan.
2. Kesesuaian perancangan dengan metodologi dan teori.
3. Portabilitas rancangan.
4. Perancangan perawatan.
5. Kebenaran rancangan berkaitan dengan fungsi dan aliran data.
6. Kelengkapan perancangan antar muka.



Pengujian Tahap Implementasi

- Pengujian pada tahap ini merupakan pengujian unit-unit yang dibuat sebelum diintegrasikan menjadi aplikasi keseluruhan.
- Faktor-faktor pengujian yang dilakukan pada tahap ini :
 1. Kendali integritas data.
 2. Kebenaran program.
 3. Kemudahan pemakaian.
 4. Sifat Coupling.
 5. Pengembangan prosedur operasi.



Pengujian Tahap Pengujian

- Tujuan pengujian pada tahap ini adalah untuk menilai apakah spesifikasi program telah ditulis menjadi instruksi-instruksi yang dapat dijalankan pada mesin.
- Selain itu juga untuk menilai apakah instruksi yang ditulis tersebut telah sesuai dengan spesifikasi program.



- Faktor-faktor pengujian yang dilakukan pada tahap ini meliputi :
 1. Pengujian fungsional.
 2. Dukungan manual.
 3. Kemudahan operasi.



Pengujian dengan Kasus Uji

- Pengujian yang dilakukan meliputi pengujian unit (berupa prosedur atau fungsi) dan pengujian sistem.
- Dalam pengujian unit, unit-unit yang diuji meliputi unit-unit yang ada dalam sistem, sedangkan pengujian sistem dilakukan terhadap sistem secara keseluruhan.
- Setiap pengujian dilakukan dengan menggunakan berbagai data masukan yang valid maupun tidak.



Jaminan Kualitas PL

Kualitas Menurut Para Ahli

- ***W E Perry.*** Kualitas adalah pemenuhan terhadap standart.
- ***R Glass.*** Kualitas adalah tingkat kesempurnaan.
- ***J Juran.*** Kualitas adalah tepat guna.

Definisi Kualitas Berdasarkan ***ISO 8402.***

Kualitas adalah keseluruhan dari fitur yang menjadikan produk dapat memuaskan / dipakai sesuai kebutuhan dengan harga yang terjangkau.



Kualitas Software

- Lima sudut Pandang tentang kualitas software:
 - Kerumit
 - user
 - Produksi
 - Produk
 - berdasarkan nilai
- Kualitas Perangkat Lunak dalam hal faktor kualitas dan kriteria
 - faktor kualitas adalah karakter dari sebuah sistem
 - Contoh : ketepatan, kehandalan, efisiensi, dan testability
 - Kriteria kualitas adalah bagian dari faktor kualitas yang berkaitan dengan pengembangan software
 - n attribute of software architecture
- Model pengukuran kualitas a.l :
 - contoh: ISO 9126, CMM, TPI, and TMM



Peranan Pengujian

Penilaian kualitas software dibagi menjadi dua kategori:

- Analisis Statis
 - Menguji coding dan penyebab dari semua kejadian yang mungkin akan muncul apabila software dijalankan
 - contoh: review coding, dan analisis algoritma
- Analisis Dinamis
 - Eksekusi program untuk mengetahui kemungkinan kegagalan program
 - Mengamati pola kerja program dan menyimpulkan mengenai kualitas dari program
- Analisis statis & dinamis saling melengkapi satu sama lain
- kedua pendekatan ada baiknya dilakukan utk mendapatkan hasil pengujian maksimal



Verifikasi dan Validasi

- Verifikasi
 - Evaluasi dari perangkat lunak yang membantu dalam menentukan apakah produk yang dikembangkan memenuhi kebutuhan yang ditetapkan sebelum pembuatan perangkat lunak dimulai
 - Membuat software dengan tepat
- Validasi
 - Evaluasi perangkat lunak yang membantu dalam menentukan apakah produk memenuhi kegunaan yang diinginkan
 - Membuat software yang tepat



Teknik Dalam Pengujian

Ada 2 macam

- Blackbox testing
 - Tidak perlu mengetahui struktur software
 - Juga disebut dengan specification-based atau functional testing
- Whitebox testing
 - Harus mengetahui struktur dan implementasi dari software



Pengujian PL

- Jenis pengujian sistem untuk sistem berbasis perangkat lunak, meliputi:
 - Pengujian Pemulihan
 - Pengujian Keamanan
 - Pengujian Stres
 - Pengujian Deployment



Pengujian PL

Pengujian Pemulihan

- Sistem harus dapat menoleransi kesalahan yaitu kesalahan pengolahan tidak harus menyebabkan keseluruhan fungsi berhenti
- Pengujian Pemulihan (*recovery testing*) adalah pengujian sistem yang memaksa PL untuk gagal dalam berbagai cara dan memverifikasi bahwa pemulihan dilakukan dengan benar



Pengujian PL

Pengujian Keamanan

- Pengujian Keamanan mencoba untuk memverifikasi mekanisme perlindungan yang dibangun ke dalam sistem, yang pada kenyataannya melindungi dari penetrasi yang tidak benar



Pengujian PL

Pengujian Stres

- Pengujian stres menjalankan sistem dengan cara meminta sumber daya dalam jumlah, frekuensi atau volume yang abnormal



Pengujian PL

Pengujian Deployment

- Dalam banyak kasus, PL harus dijalankan pada berbagai *platform* dan berada di beberapa Sistem Operasi
- Pengujian Deployment sering disebut juga dengan Pengujian Konfigurasi merupakan pengujian yang menjalankan PL di setiap lingkungan dimana PL tsb beroperasi
- Pengujian Deployment juga memeriksa semua prosedur dan instalasi PL (*installer*) yang akan digunakan serta semua dokumentasi bantuan PL



Kaidah Umum Pengujian

- Dimulai dari pengujian tingkat komponen menuju integrasi.
- Titik yang berbeda dimungkinkan memakai teknik pengujian yang berbeda.
- Pengujian dilakukan oleh developer dan (untuk proyek yang besar) tim independen.
- Testing dan debugging adalah berbeda. Namun debugging pasti berkaitan dengan strategi testing apapun.

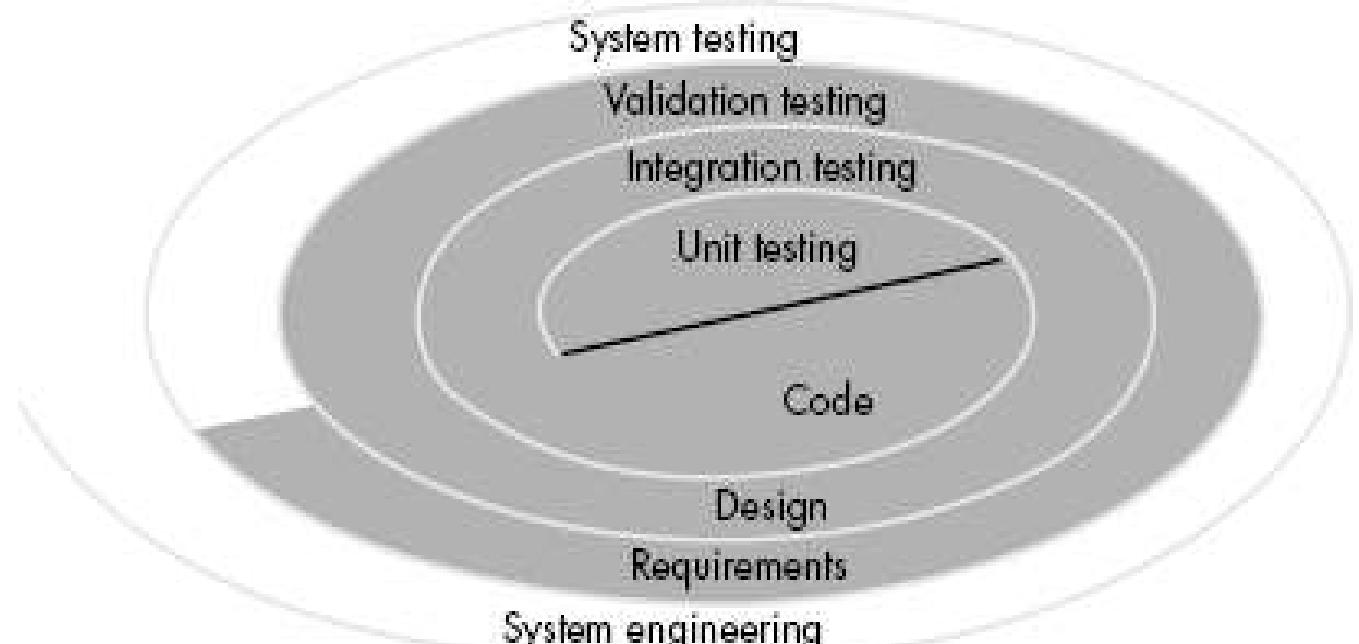


Strategi Pengujian

- Dimulai dari unit testing terhadap source code hingga system testing terhadap spesifikasi kebutuhan.

FIGURE 18.1

Testing strategy



Langkah Pengujian PL

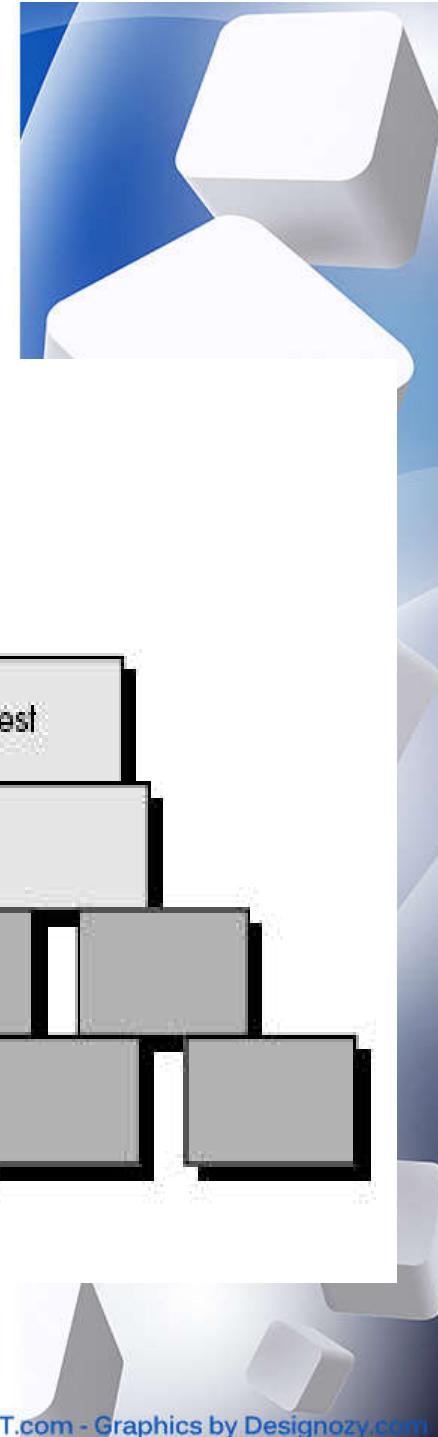
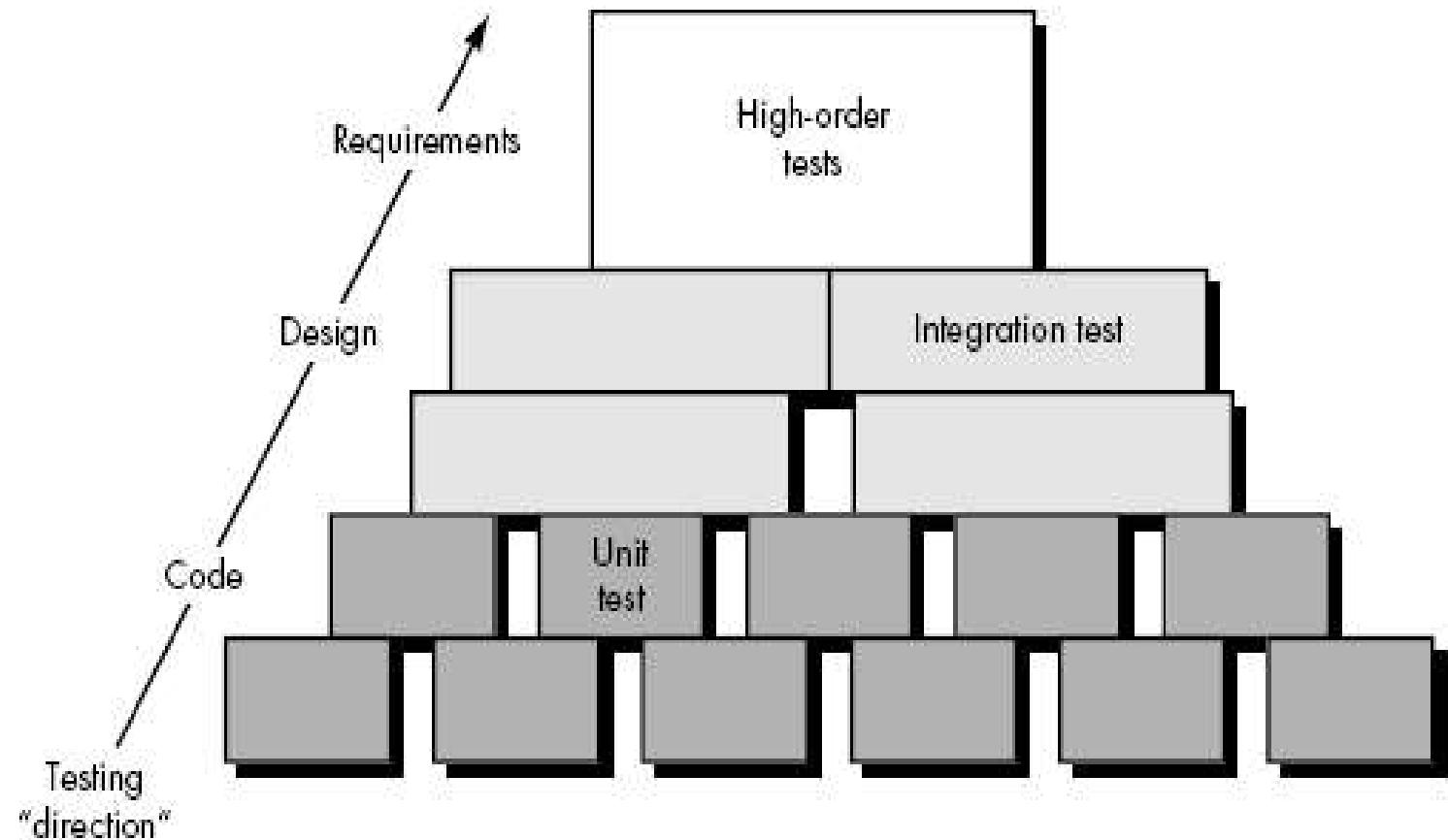


FIGURE 18.2

Software
testing steps



Unit Testing

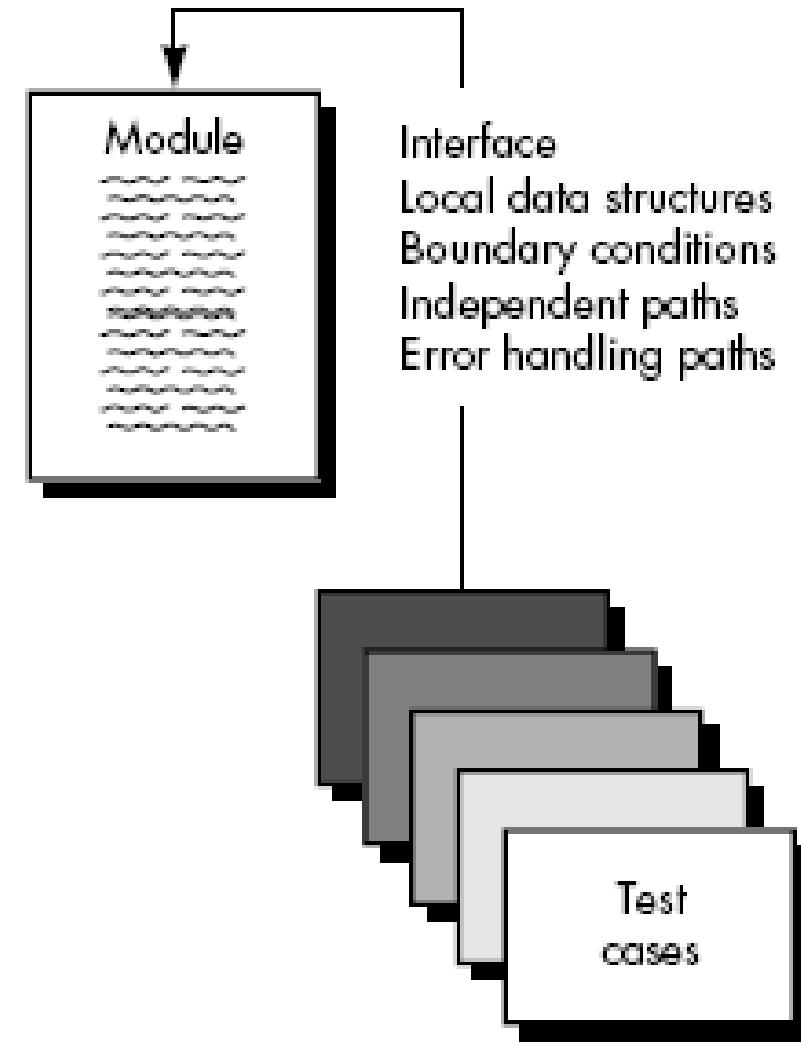


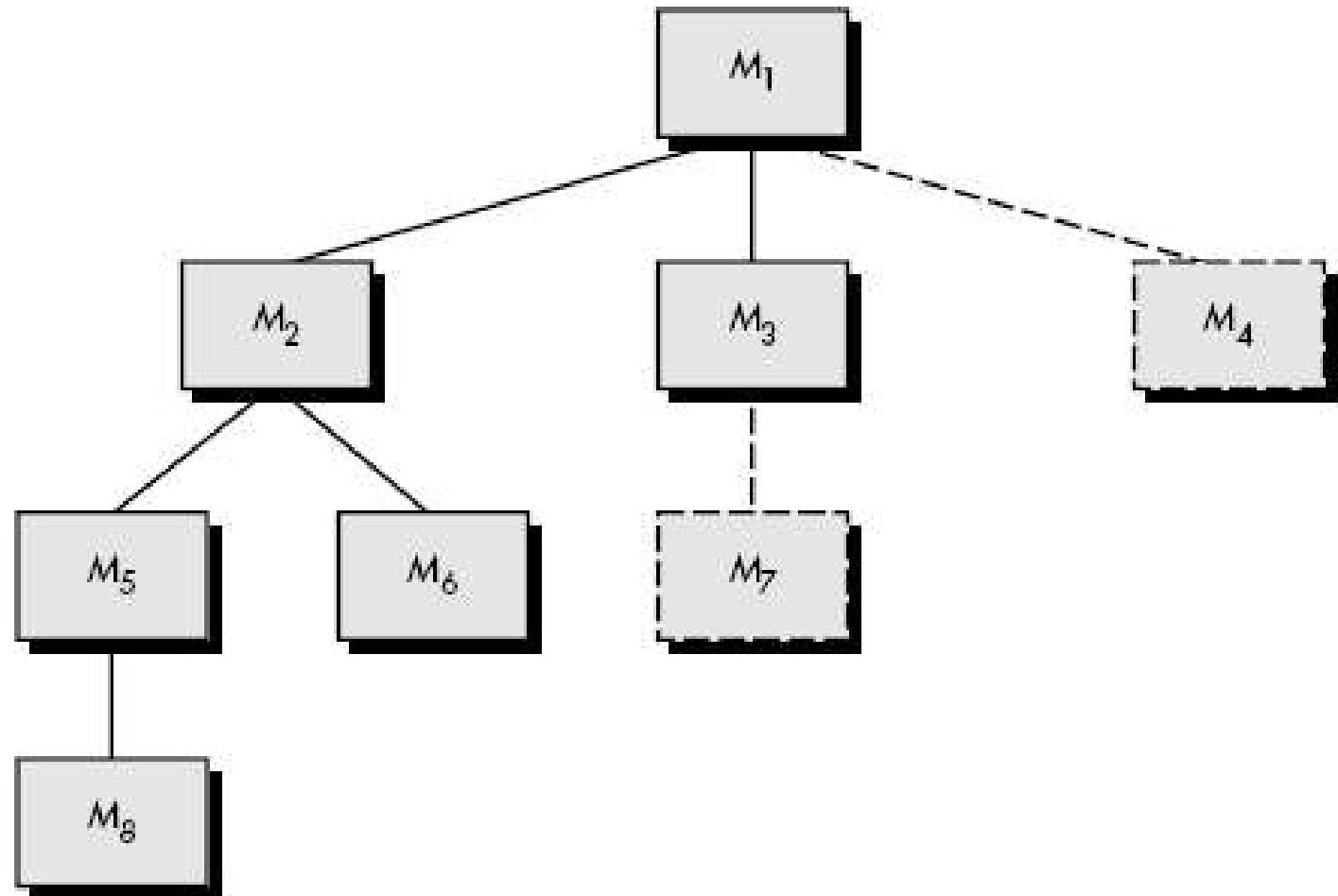
FIGURE 18.4
Unit test

Integration Testing

- Top – down integration

FIGURE 18.6

Top-down
integration



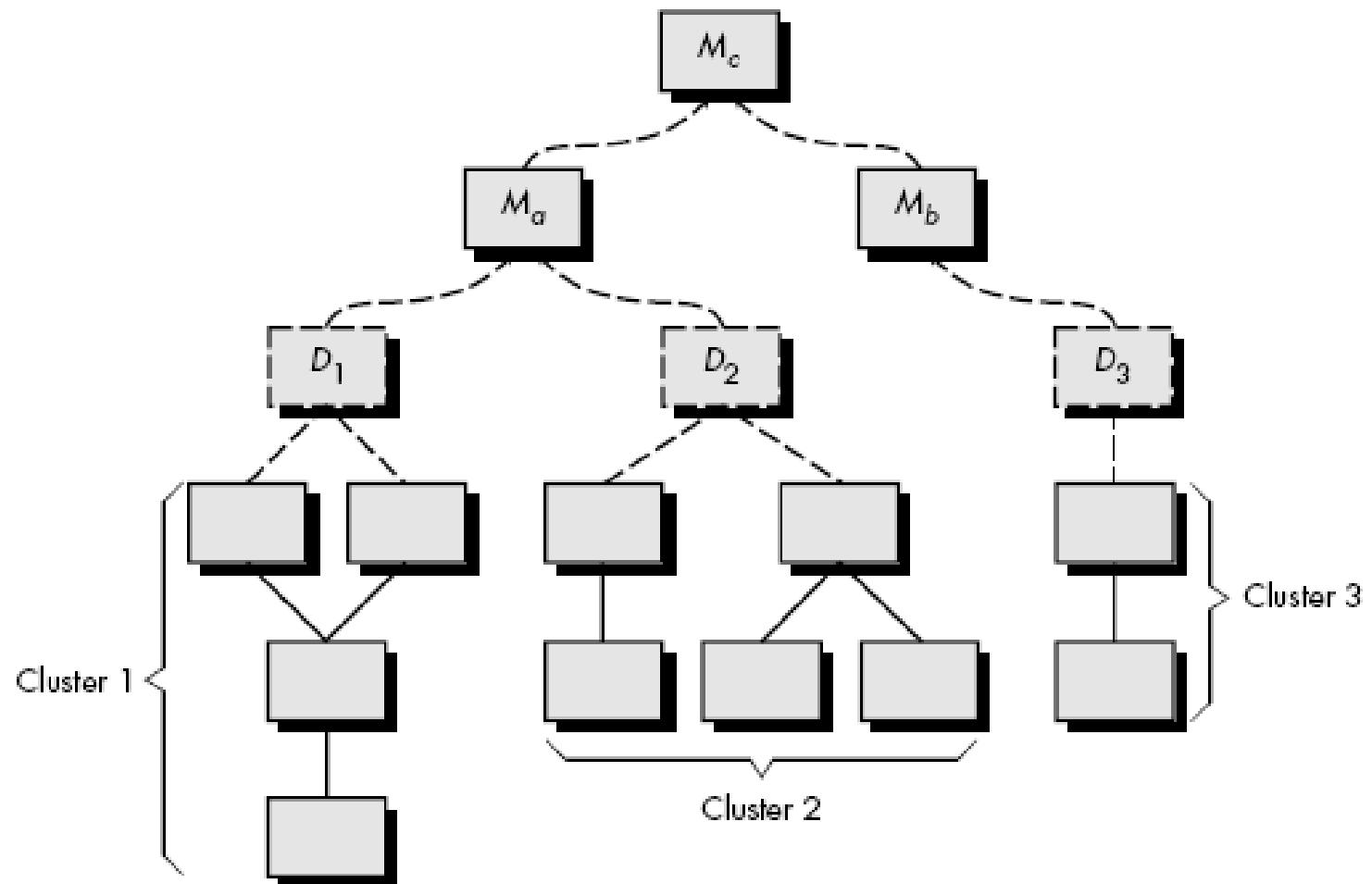
Integration Testing



- Bottom – up integration

FIGURE 18.7

Bottom-up
integration



Integration Testing

- Regression testing: dilakukan pengujian setiap kali ada modul baru yang diintegrasikan atau ada modul yang berubah.
- Smoke testing: test daily, untuk proyek jenis kritis-waktu.



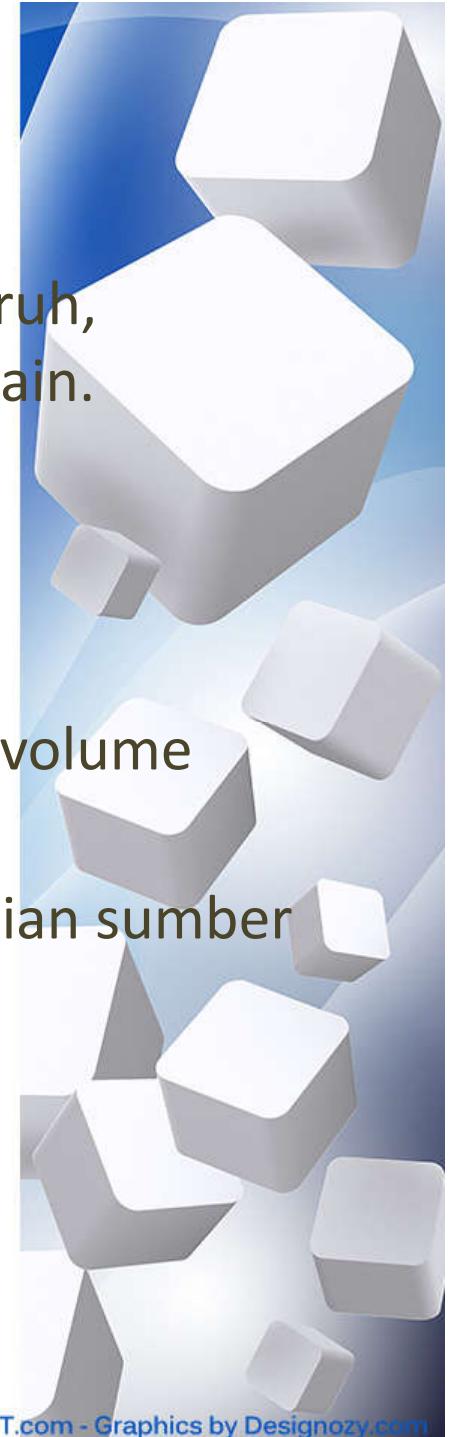
Validation Testing

- Disebut sukses jika fungsi P/L dapat diterima oleh customer (berdasarkan dokumen SKPL).
- Alpha test: dilakukan di tempat developer oleh customer pada lingkungan yang terkendali.
- Beta test: dilakukan di tempat customer tanpa melibatkan developer pada lingkungan yang tak terkendali.



System Testing

- Meguji sistem berbasis komputer secara menyeluruh, termasuk juga hubungannya dengan sistem yang lain.
- Diantaranya:
 - Recovery testing, jika system failure.
 - Security testing, jika terjadi serangan.
 - Stress testing, terhadap jumlah, frekuensi dan volume pekerjaan.
 - Performance testing, untuk mengukur pemakaian sumber daya.



Debugging

- Memperbaiki error yang ditemukan pada saat testing (yang sukses).
- Kaidah dasar sebelum debug:
 - Apakah penyebab bug dihasilkan kembali oleh bagian program yang lain?
 - Apakah bug selanjutnya yang mungkin muncul jika bug diperbaiki?
 - Apa yang bisa dilakukan untuk mencegah bug terjadi untuk pertama kalinya?



Prinsip-prinsip Testing

Terdapat 6 kunci prinsip-prinsip testing, yaitu :

- Testing yang komplit tidak mungkin.
- Testing merupakan pekerjaan yang kreatif dan sulit.
- Alasan yang penting diadakannya testing adalah untuk mencegah terjadinya *errors*.
- Testing berbasis pada resiko.
- Testing harus direncanakan.
- Testing membutuhkan independensi.



Prinsip-prinsip Testing

- **Testing yang komplit tidak mungkin**

Faktor yang menyebabkan adalah :

1. Domain Masukan
2. Kompleksitas
3. Jalur Program



Prinsip-prinsip Testing

- **Testing merupakan pekerjaan yang kreatif dan sulit**

Dikatakan Sulit karena :

1. Seorang Tester harus paham Sistem Keseluruhan
2. Sistem tidak sederhana untuk dipahami

Beberapa Skill yang dibutuhkan Tester :

1. Kreatifitas
2. Pengetahuan bisnis
3. Pengalaman testing
4. Metodologi testing



Prinsip-prinsip Testing

Testing bukanlah suatu hal yang sederhana, karena :

- Untuk melakukan testing secara efektif, harus mengetahui keseluruhan sistem
- Sistem tidak sederhana/tidak mudah dipahami

Untuk dapat sukses dalam melakukan testing dibutuhkan hal-hal sebagai berikut :

- Kreatifitas
- Pengetahuan Bisnis
- Pengalaman Testing
- Metodologi Testing

Testing (Kunci Pengaruh)

- Wawasan dan Kreatifitas
- Pengetahuan dan Pemahaman



Testing Berbasis Resiko

4 Poin penting yang menyebabkan testing berbasis resiko :

1. Resource ; Sumber daya dan biaya yang dibutuhkan untuk melakukan testing berdasarkan pada skala prioritas, kompleksitas dan kesulitan testing
2. Cost keterlambatan Pengiriman barang ; Biaya dari keterlambatan pengiriman produk (dimana salah satu kemungkinan besar penyebabnya adalah testing)
3. Defect/cacat ; Kemungkinan adanya suatu defect (berdasarkan pengalaman beroperasi dan prioritas sejarah terjadinya defect)
4. Cost of Defect ; Biaya yang disebabkan oleh defect , bilamana defect tersebut menyebabkan error yang akan membawa kerugian baik secara langsung ataupun tak langsung bagi pelanggan (berkaitan dengan kewajiban bisnis bagi pengembang terhadap kerugian yang terjadi pada pelanggan).



Konsep siklus dari testing

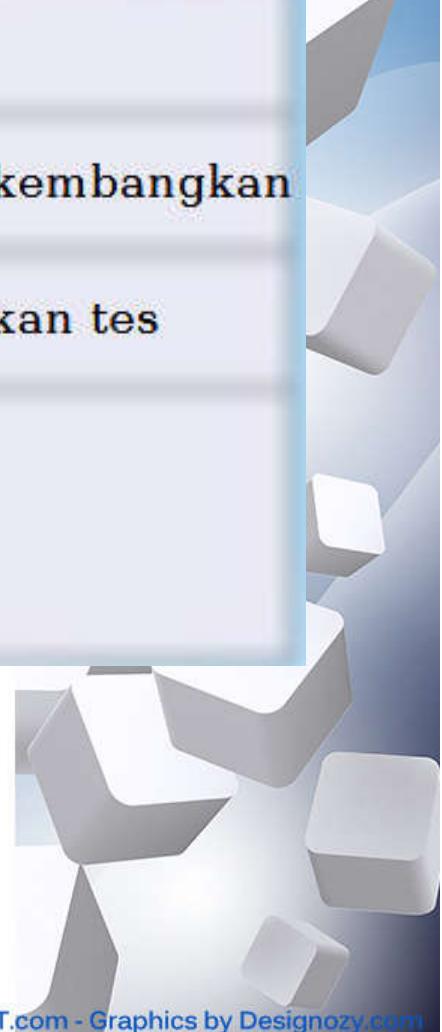
- Testing bukan untuk satu fase pengembangan saja.
- Hasil Testing diasosiasikan pada tiap fase pengembangan.



Testing harus direncanakan



Rencana Tes	Disain Tes
Pernyataan obyektifitas testing	Spesifikasi tes yang dikembangkan
Deskripsi pendekatan tes	Deskripsi pengelompokan tes
Sekelompok tugas untuk mencapai obyektifitas testing	

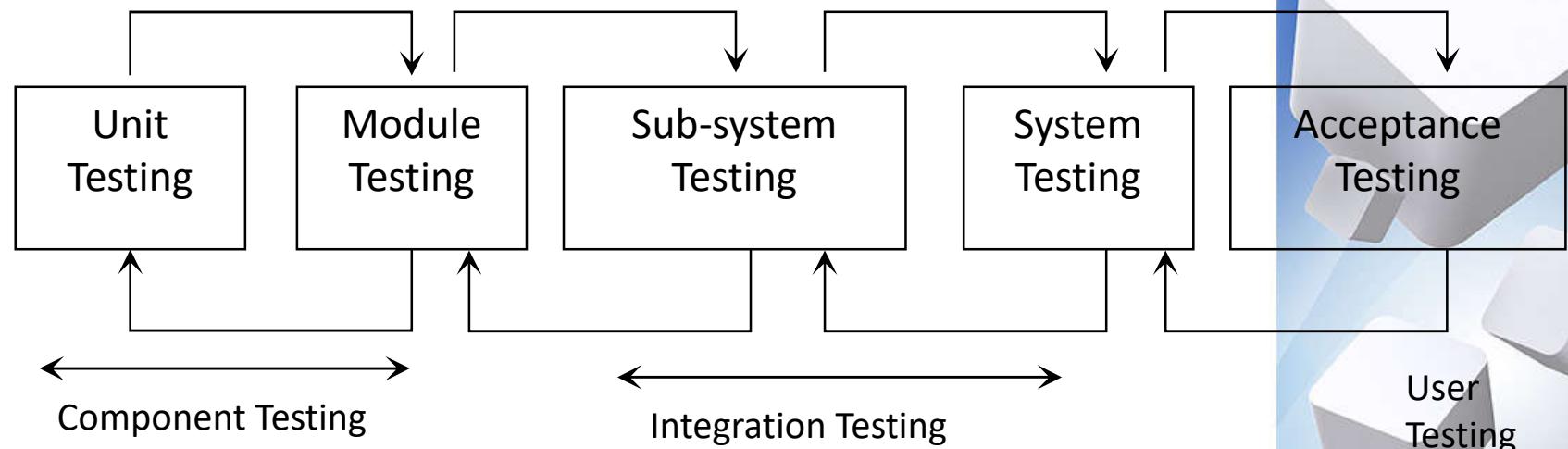


Perencanaan Pengujian PL

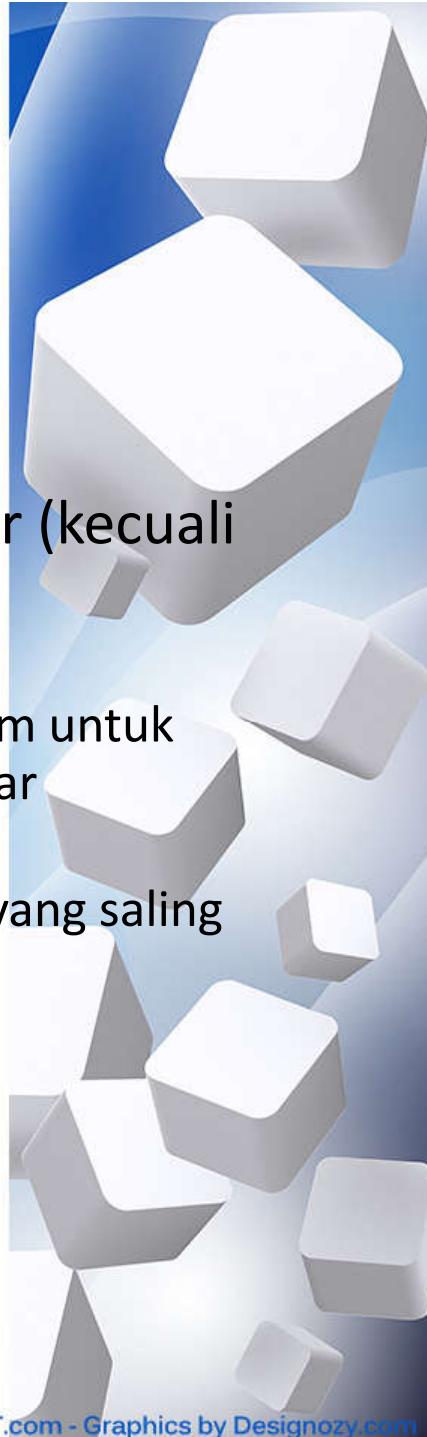
- Proses testing
 - Deskripsi fase-fase utama dalam pengujian
- Pelacakan Kebutuhan
 - Semua kebutuhan user diuji secara individu
- Item yg diuji
 - Menspesifikasi komponen sistem yang diuji
- Jadual Testing
- Prosedur Pencatatan Hasil dan Prosedur
- Kebutuhan akan Hardware dan Software
- Kendala-kendala
 - Mis: kekurangan staff, alat, waktu dll



Proses Testing



Proses Pengujian PL

- ❖ Component testing
 - Pengujian komponen-komponen program
 - Biasanya dilakukan oleh component developer (kecuali untuk system kritis)
 - 1. Unit testing
 - Pengujian masing-masing unit komponen program untuk meyakinkan bahwa sudah beroperasi secara benar
 - 2. Module Testing
 - Pengujian terhadap koleksi unit-unit komponen yang saling berhubungan.
- 

Proses Pengujian PL

❖ Integration testing

- Pengujian kelompok komponen-komponen yang terintegrasi untuk membentuk sub-system ataupun system
- Dilakukan oleh tim penguji yang independent
- Pengujian berdasarkan spesifikasi sistem

1. Sub-system Testing

- Pengujian terhadap koleksi module-module yang membentuk suatu sub-system (aplikasi)

2. System Testing

- Pengujian terhadap integrasi sub-system, yaitu keterhubungan antar sub-system



Proses Pengujian PL

❖ User testing

1. System Testing

- Pengujian terhadap integrasi sub-system, yaitu keterhubungan antar sub-system

2. Acceptance Testing

- Pengujian terakhir sebelum sistem dipakai oleh user.
- Melibatkan pengujian dengan data dari pengguna sistem.
- Biasa dikenal sebagai “alpha test” (“beta test” untuk software komersial, dimana pengujian dilakukan oleh potensial customer)

