Macam – Macam Testing Sistem Informasi

Edy Supriyanto
Fakultas Teknologi Informasi, Universitas Stikubank Semarang
e-mail : edys@unisbank.ac.id

ABSTRAK: Testing software merupakan suau kegiatan yang dibahas pada hampir semua metode software engineering. Ada 3 kategori testing dalam hal ini, yaitu : testing input, testing output dan testing proses. Berbagai jenis testing tersebut dibahas dan diberikan contohnya pada pembahasan tulisan ini.

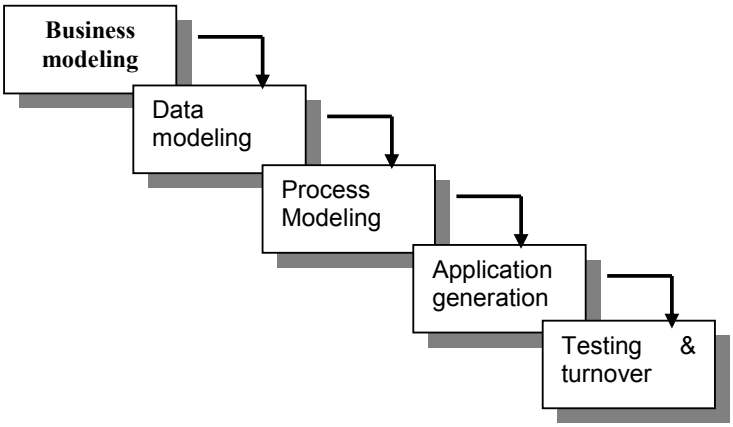
Kata kunci : testing, sistem informasi

PENDAHULUAN

Pada saat kita akan memakai suatu software, sebaiknya kita sudah melakukan pengujian (testing) dulu atas software tersebut. Kalaupun tidak melakukan pengujian sendiri, kita juga sudah yakin bahwa software tersebut sudah pernah diuji sebelumnya oleh orang/pihak lain. Software yang berasal dari pabrik misalnya (Microsoft), tentu sudah lolos dari uji, karena masalah pengujian telah menjadi bahasan yang vital di pabrik tersebut, Ini sudah dilakukan jauh-jauh hari sebelum produk itu di-*release* kepada user.

Testing software kali ini diasumsikan jika kita membuat software sendiri. Testing software ini tentu harus dipertimbangkan pada saat pembuatan software. Sehingga dalam pembuatan software dapat memperhatikan pada bagian mana yang perlu mendapat pengujian. Dengan begitu, software hasil bikinan kita sendiri dapat lebih baik dan sesuai dengan harapan.

Seperti yang telah di ketahui bersama, pada hampir semua model pengembangan rekayasa perangkat lunak mempunyai tahapan testing. Seperti pada model *RAD* (Rapid Application Development) yang disampaikan oleh Roger S. Pressman, maka tahapan RAD mempunyai fase sebagai berikut : *business modeling, data modeling, processing model, application generation*, dan *testing and turnover*. Untuk lebih jelasnya, berikut ini disajikan gambar dari tahapan tersebut.



Gambar 1. Tahapan RAD

Secara umum, pengujian (testing) software dapat dibagi dalam tiga kategori, yaitu : Testing saat Input Data, Testing saat Pemrosesan, dan Testing saat Output. Testing saat Input Data adalah tindakan untuk menguji edit dan kontrol dalam pemasukan data, misalnya : validasi, cek digit. Testing saat Pemrosesan bertujuan untuk meyakinkan bahwa program telah bekerja seperti yang diharapkan. Misalnya : keyakinan bahwa tabel yang diproses telah benar-benar direvisi dengan kondisi masukan terakhir, dan kalkulasi dalam program telah benar sesuai rumus yang diinginkan. Sedangkan Testing saat Output berguna untuk meyakinkan bahwa laporan yang dihasilkan telah dibuat dengan format yang benar dan mempunyai informasi yang valid.

Dalam melakukan testing software, kadang-kadang kita juga melakukan tugas lain yang bersamaan pelaksanaannya.. Kegiatan ini misalnya konversi data, pemeriksaan hardware, kemampuan system operasi dan keandalan sekuritas.

Selain itu, yang perlu diperhatikan pada saat testing adalah : semua kejadian dalam testing harus didokumentasi dengan baik, semua testing harus mendapat ijin dari otoritas yang berwenang, dan hasil pengujian harus dilaporkan kepada otoritas yang memberi ijin/tugas serta disetujui hasilnya. Banyaknya pengujian terhadap suatu field atau suatu proses juga harus menjadi perhatian. Dengan adanya testing dapat dilihat bahwa system informasi telah bekerja dengan baik, sehingga diharapkan system informasi dapat menolak data yang tidak akurat (tidak cocok).

TESTING INPUT DATA

Berikut ini disajikan berbagai pengontrolan saat input data, yaitu :

1. **Character checks (pengecekan karakter)**

   Suatu tindakan untuk melihat apakah suatu field itu dapat menerima karakter tertentu saja atau tidak.

   Misalnya : pada kolom pengisian data, user diharapkan untuk mengisi data yang berupa karakter saja, atau numeric saja, atau data berupa karakter dan numeric.

2. **Numeric Value Checks (Pengecekan Nilai Numerik)**

   Ini merupakan character checks yang dikhususkan pada karakter bilangan.

   Misalnya : pengisian yang membolehkan nilai negatif.

3. **Check Digit (Digit cek)**

   Sejumlah angka mempunyai check digit, maka sistem akan menolak sembarang masukan yang banyaknya angka (digit) tidak akurat.

   Misalnya : password kartu ATM.

4. **Limit Tests (Pengujian Batas)**

   Pada suatu field, kadang-kadang nilai yang harus diisikan terbatas jangkauan nilainya.

   Misalnya : banyaknya SKS yang harus diambil seorang mahasiswa pada satu semester.

5. **Reasonableness Tests (Pengujian Ke-logisan)**

   Seperti pada limit test, tetapi pembatasannya pada hal yang logis (beralasan).

   Misalnya : jumlah anak.

6. **Internal Compatibility (Kompatibilitas Internal)**

   Suatu data yang sudah dimasukkan, sebaiknya kompatibel dengan data yang lain dalam satu aplikasi tertentu.

   Misalkan : field NIM mahasiswa dapat dipakai di program absensi, program kartu ujian, kartu KRS pada aplikasi TRANSKRIP, dengan mengacu pada nama mahasiswa yang sama.

7. **Cross Checks with data in other applications**

   Suatu data akan diperiksa secara silang dengan aplikasi yang lain, sehingga jika terjadi kesalahan maka pesan kesalahan akan diberikan. Ini bermaksud untuk memeriksa apakah fungsinya telah berjalan dengan benar.

Misalnya : file gaji dan file absensi harus saling berrelasi dengan baik dan berasal dari master file yang sama

8. Duplicate Transactions (Transaksi Ganda)

Suatu system sebaiknya dibuat dapat menolak transaksi yang ganda.

Misalnya : pada pembayaran bulanan PLN. Jika sudah dibayar pada suatu tempat, maka tempat pembayaran lain yang on-line harus diblok agar tidak dapat dibayar lagi oleh orang lain.

9. Table Look Ups

Jika suatu kode tertentu dimasukkan dalam suatu field, maka system akan mengakses table yang tepat dan memberikan informasi yang benar.

Misalnya : pada isian field yang berbentuk combo box atau list box.

10. Existence of Required Data (Keberadaan Data yang Dibutuhkan)

Jika suatu data yang dibutuhkan tidak ada, maka system harus memberikan pesan bahwa data yang dibutuhkan tidak ada. Jadi harus ada kejelasan tentang data.

Misalnya : nomor rekening bank

11. Confirmation Screens (Layar Konfirmasi)

Suatu tampilan yang akan memberikan konfirmasi bahwa data yang dimasukkan adalah data yang benar.

Misalnya : memasukkan nomor password kartu ATM.

12. Field lengths and Overflow checks (Cek panjang field dan overflow)

Panjang field dapat diberikan dengan ukuran tertentu. Begitu juga dengan banyaknya data yang dapat dimasukkan dalam suatu field.

Misalnya : nomor rekening bank pasti mempunyai nomor tertentu dan banyaknya angka juga tertentu, dan harus diisi secara penuh semua kolom-kolomnya. Juga pada suatu field dapat terjadi mempunyai keterbatasan harga. Misalnya field jumlah anak, harus dibatasi sebanyak 2 digit. Jika seseorang mengisi yang berlebihan (overflow), maka system sebaiknya langsung memberikan pesan kesalahan.

13. Edit Over-rides (Edit Penumpukan)

Dapatkah suatu pengeditan data dengan penumpukan (penulisan di atasnya) ? Jika ini memang terjadi dan boleh, maka yakinlah bahwa tampilan penumpukan tersebut bekerja dengan semestinya. Ini biasanya untuk masalah keamanan, misalnya pada penulisan password yang harus diisi dua kali dan ditulis dengan tanda ***** (tanpa user tahu), padahal isian field-field password tersebut harus sama walaupun datanya tidak dapat diketahui.

Misalnya : penggantian password.

14. Arithmetic Accuracy/Tolerance Levels

Keakuratan perhitungan aritmetika juga harus menjadi perhatian khusus, sehingga nantinya jika ada kalkulasi terhadap angka tersebut tidak terjadi kesalahan, apalagi jika salah maka program menjadi mogok (hang).

Misalnya : perhitungan pada aplikasi perbankan.

15. Date-Driven Edits (Edit Kendali Data Tanggal)

Ini berguna untuk mengedit masukan yang bergantung pada data tahun/tanggal.

Misalnya : ingin mengetahui data mahasiswa, maka harus memasukkan nomor NIM dan biasanya data nomor NIM mahasiswa tersebut berhubungan dengan tahun masuk mahasiswa yang bersangkutan. Contoh, NIM 038101009B, ini menunjukkan tahun masuk adalah 2003.

16. Penanda khusus/counter/flag yang menunjukkan bahwa suatu transaksi belum selesai.

Misalnya : pada pengisian KRS mahasiswa yang menunjukkan angkatan tertentu harus mengambil sejumlah mata kuliah, maka ada counter untuk memberi pesan bahwa isian belum selesai dan harus diisi.

TESTING SAAT PEMROSESAN DATA

1. Delete vs Reverse (Hapus lawan Mundur)

Kita harus tahu bahwa data yang telah dimasukkan, nantinya akan dapat dihapus atau ditelusuri mundur (historisnya). Ketika suatu transaksi telah diupdate (dimutakhirkan), transaksi itu mungkin perlu dihapus atau ditelusuri ke belakang (mundur). Dan kegiatan delete atau reverse itu harus dapat berjalan dengan baik.

Misalnya : aplikasi perbankan untuk tabungan harus ada fasilitas reverse, agar dapat sebelumnya dapat ditelusuri.

2. Automatically Triggered Processing (Pemrosesan Terpicu Secara Otomatis)

Jika suatu system mempunyai sifat automatically triggered processing, maka testing yang dilakukan harusnya dapat meyakinkan bahwa kalkulasi atau pemrosesan telah dilakukan dengan benar. Dalam hal ini testing dilakukan untuk meyakinkan bahwa parameter yang tepat telah digunakan dan output dari pemrosesan telah akurat.

Misalnya : untuk aplikasi robotika, maka sensor-sensor harus dapat memberikan responsive dari pemrosesan yang dapat terpicu secara otomatis.

3. Updating (Pemutakhiran)

Testing dijalankan untuk meyakinkan bahwa sistem telah di-update secara benar dengan data yang telah dimasukkan. Dalam hal ini, table yang tepat telah di-update dengan informasi yang benar. Bisa jadi, ada lebih ari satu table yang di-update karena adanya satu transaksi.

Misalnya : pada pembuatan rekening PLN, maka pemutakhiran data merupakan hal yang penting.

4. Audit Trails (Jejak Pemeriksaan)

Seperti pada nomor 3, maka testing dilakukan untuk meyakinkan bahwa log system dan jejak untuk pemeriksaan telah bekerja dengan baik.

Misalnya : pada aplikasi perbankan untuk tabungan, maka jejak penabung sejak menabung harus dapat ditelusuri, sehingga jika buku tabungan hilang atau pencetakan buku tabungan yang baru dapat berjalan sesuai dengan yang diharapkan.

5. Table Values (Nilai Tabel)

Testing ini dijalankan pada prosedur untuk peng-updatean parameter system dan tabel kode. Pelaksanaan dari update transaksi harus bekerja dengan tepat, termasuk dalam hal ini adalah pengeditan pemasukan data.

Misalnya : nilai table nilai mahasiswa adalah karakter, sedangkan nilai table untuk KRS adalah kode-kode mata kuliah.

6. Initialization and Purge (Pengawalan dan Pembersihan data)

Testing juga dilakukan pada saat awal pengisian record dan pembersihan data yang lama. Ini dilakukan agar tidak terjadi data yang rancu (tumpang tindih).

Misalnya : pada pencetakan sisa saldo pada buku tabungan yang baru.

7. Back-up & Recovery (Penggandaan & Penyegaran)

Proses dari penggandaan data dari system dan penyegaran system untuk menghindari kejadian yang mengalami kegagalan juga harus diadakan pengujian.

Misalnya : setiap file harus di-back up agar aman dan masih mempunyai arsipnya. Sedangkan data yang perlu ada perubahan pada saat tertentu, harus diadakan penyegaran, misalnya : data lama bekerja yang setiap tahun harus bertambah.

8. Arithmetic Calculations (Kalkulasi Aritmetika)

Tes ini ditujukan kepada semua kalkulasi aritmetika yang telah dilakukan sudah benar sesuai rumus yang diinginkan. Untuk meyakinkan report telah benar, maka jika perlu ada cek silang dengan kalkulasi yang telah dilakukan.

Misalnya : kalkulasi untuk penghitungan bunga bank.

9. Volume Testing (Pengujian Volume)

Sistem harus diuji pada kondisi level beban normal dan level puncak untuk meyakinkan bahwa pada system tidak terjadi bottleneck. Pengujian street juga dikerjakan untuk melihat kondisi system untuk menerima batas limit yang tertinggi.

Misalnya : penentuan operasional aplikasi perbankan, harus melihat saat nasabah terjadi *rush*, maka aplikasi komputer harus dapat mengatasinya.

10. Live Transaction Testing (Pengujian Transaksi Hidup)

Testing pemrosesan system harus dijalankan dengan data hidup (dengan catatan data yang lama juga harus disimpan dalam system). Ini bertujuan agar berbagai variasi pemrosesan data telah dapat diuji dengan benar.

Misalnya : pada aplikasi ATM perbankan.

11. Database Management System Testing (Pengujian Sistem Manajemen Database)

Struktur database juga harus diuji untuk meyakinkan bahwa disain telah dibuat dengan benar.

Misalnya : relasi file dan query-nya harus dapat berjalan dengan baik.

12. Interface with other modules/systems (Penghubung dengan modul/system lainnya)

Testing juga dijalankan untuk meyakinkan bahwa modul-modul yang lain telah dapat di-update dengan baik terhadap data yang telah dimasukkan dan diproses sebelumnya. Sehingga antara modul satu dengan yang lainnya dapat terjadi sinergi yang baik, sesuai yang diharapkan.

Misalnya : model absensi harus ada hubungan dengan modul penggajian.

13. Pengujian pada sembarang pemrosesan batch bertujuan untuk meyakinkan bahwa system yang ada betul-betul sesuai yang diinginkan.

Misalnya : pada proses batch pembayaran rekening PLN.

14. Hal-hal yang berhubungan dengan telekomunikasi juga harus diuji, walaupun untuk hal-hal yang bersifat teknis, kita juga harus memanggil ahli bidang.

Misalnya : dicoba berbagai komputer dari berbagai tempat dengan fasilitas telekomunikasi.

TESTING OUTPUT

1. Ringkasan laporan dapat diperiksa sebagai hal untuk meyakinkan bahwa format dan isi dari laporan sesuai yang dibutuhkan.

   Misalnya : transkrip sebagai bentuk laporan hasil kuliah seorang mahasiswa.

2. Yakinlah bahwa dalam laporan hal-hal yang bersifat aritmetika telah berjalan dengan benar.

   Misalnya : perhitungan dalam transkrip.

3. Sebaiknya seluruh laporan harus disajikan dan lihatlah ada kesalahan atau tidak dengan laporan sebelumnya.

   Misalnya : hubungan transkrip dengan KHS dari seorang mahasiswa.

4. Jika ada laporan yang bersifat khusus, maka testing juga harus dapat menjawab bahwa laporan khusus ini ditujukan untuk meyakinkan bahwa data yang diekstrak dari suatu tempat harus cocok dan lengkap dengan kriteria khusus yang diharapkan.

   Misalnya : laporan pemasaran suatu barang dapat dibuat harian, mingguan, dan bulanan.

DAFTAR PUSTAKA

1. Manitoba, www.umanitoba.ca/university
2. Pressman R.S., 1997, *Software Engineering Concepts*, Fourth Edition, McGraw-Hill Book Company.