

BAB 3

PENGEMBANGAN RENCANA UJI SUATU SYSTEM APLIKASI

Rencana uji aplikasi adalah dokumen perencanaan strategis yang menggambarkan apa yang perlu dilaksanakan selama pengujian. Penyiapan rencana uji sistem aplikasi merupakan tanggung jawab manajer proyek pengembangan sistem. Rencana uji merupakan cetak biru yang dipakai oleh personal proyek dan pemakai untuk memastikan bahwa sistem aplikasi mencapai tingkat kebenaran dan keandalan yang diinginkan oleh pemakai.

Langkah-langkah Rencana Uji

Rencana uji harus dapat menjelaskan siapa yang akan menguji, apa yang akan diuji, kapan pengujian dilakukan, mengapa pengujian dilakukan, bagaimana pengujian dilakukan, serta dimana pengujian dilaksanakan.

Pengembangan rencana uji dilaksanakan dengan mengikuti langkah-langkah sebagai berikut:

1. *Menentukan tingkat jaminan yang diinginkan*, pemakai harus memutuskan presisi dari proses uji, atau tingkat keakuratan dan keandalan sistem yang diinginkan.
2. *Memilih pendekatan uji*, apakah pengujian dilakukan oleh tim proyek, penguji independen, quality assurance, orang luar atau kombinasinya;
3. *Menentukan penanggung jawab uji*, seseorang harus ditugaskan untuk menentukan apakah sistem sudah diuji dengan cukup memadai.
4. *Mendefinisikan kondisi uji*, kriteria bahwa sistem aplikasi harus mencapai tingkat yang dapat diterima harus didefinisikan.
5. *Mengembangkan rencana administratif*, rencana administratif meliputi jadwal uji, orang, waktu komputer, data, anggaran, alat, dan teknik yang dipakai dalam pengujian,

Langkah I - Menentukan Tingkat Jaminan yang Diinginkan

Studi yang dilakukan oleh beberapa perusahaan besar menunjukkan kurva biaya keefektifan pada pengujian bahwa biaya-efektif untuk menguji adalah bilamana 80% kesalahan terdeteksi (lihat gambar 14), Pada titik ini, biaya meningkat dengan cepat dan biaya kesalahan yang terdeteksi mulai tidak ekonomis. Biasanya pada titik ini pengujian berhenti secara formal dan berlanjut secara informal ke status produksi,

Jika kurva biaya-keefektifan diulang beberapa kali Selama daur hidup pengembangan sistem, pengujian berlanjut menjadi biaya-efektif dalam menghilangkan kegagalan.

Gambar 15 menunjukkan bahwa jika selama fase kebutuhan kita dapat membuang sampai 80% kegagalan dan selama perancangan sampai 80% dari kegagalan yang tersisa dari fase kebutuhan ditambah kegagalan selama perancangan. dan selama pengkodean 80% dari kegagalan yang tidak terdeteksi selama kebutuhan dan perancangan, ditambah 80% dari kegagalan selama pengkodean, dst, maka hanya sejumlah kecil kegagalan akan tersisa untuk dideteksi selama fase pengujian.

Hal ini menunjukkan dua konsep ekonomi yang berbeda dari pengujian.

Konsep pertama berhubungan dengan masalah ekonomi dari pembuangan kegagalan pada suatu saat. Konsep kedua adalah beberapa pengujian pada beberapa titik yang berbeda dalam daur hidup untuk membuang lebih banyak kegagalan pada titik biaya-keefektifan dari pengujian.

Jumlah sumber rata-rata yang dialokasikan untuk pengujian adalah 5% dari usaha pengembangan total. Dilain pihak, pengujian dapat dioptimalkan sampai 30% dari usaha pengembangan total.

Bagian yang dialokasikan untuk pengujian harus dialokasikan untuk fase selain fase uji. Penempatan terlalu banyak sumber pada satu fase tertentu menunjukkan bahwa over-alokasi pada suatu fase tidak biaya-efektif.

Gambar 16 menunjukkan alokasi usaha pengujian yang direkomendasikan, yang didasarkan pada pengalaman beberapa perusahaan besar yang mengalokasikan 30% dari usaha pengembangan total untuk pengujian.

Jumlah sumber yang dialokasikan untuk pengujian harus merupakan keputusan dari pemakai dan personil pengolahan data. Pengolahan data memerlukan tingkat pengujian minimal untuk memastikan lingkungan produksi yang tak terinterupsi dan untuk menyediakan suatu jaminan kontinuitas pengolahan yang layak. Di lain pihak, pemakai mempunyai kendala waktu dan biaya, sehingga kegagalan hanya diterima di lingkungan produksi.

Empat tugas yang harus dilaksanakan dalam penentuan tingkat jaminan yang diinginkan :

1. Mendefinisikan apa yang bisa salah
2. Menentukan biaya dari masalah
3. Menentukan tingkat penerimaan dari masalah
4. Mendefinisikan objektif pengujian

Tugas 1 - Mendefinisikan Apa yang Bisa Salah

Terdapat dua metoda untuk menggambarkan masalah yang bisa salah dalam sistem aplikasi. Metoda pertama menjelaskan masalah dalam hal kesalahan yang terjadi dalam proses pengembangan, dan metoda kedua menggambarkan kesalahan dalam hal kejadian yang tidak diinginkan pada operasi dari sistem aplikasi.

Pengolah data lebih mengenal masalah yang terjadi dalam proses pengembangan. Jenis kesalahan dapat dikelompokkan sebagai berikut :

Pemakai mendefinisikan kebutuhan dengan tidak tepat

- Analisis sistem mendokumentasikan kebutuhan pemakai dengan tidak tepat
- Perancangan gagal mencapai kebutuhan pemakai yang dispesifikasikan
- Perancangan program gagal mencapai kebutuhan perancangan sistem
- Program yang dikode gagal mencapai kebutuhan perancangan program
- Bahasa pemrograman dipakai dengan tidak benar
- Kondisi uji salah

Sudut pandang pemakai terhadap masalah sistem adalah dari bagaimana sistem aplikasi dapat memberikan hasil yang salah.

Berikut ini didaftarkan beberapa aplikasi otomatis yang umum dan masalah yang berat dari sudut pandang pemakai :

- Daur pendapatan
Daur pendapatan meliputi tungsi-lungsi yang terlibat dalam penerurlaan dan penerimaan permintaan untuk barang atau layanan : pengiriman atau penyediaan barang atau layanan: permintaan kredit, penerimaan kas. dan aktivitas pengumpulan; billing; accounts receivable. komisi, jaminan. piutang. barang kembali serta pengaturan lainnya.

Kriteria masalah dalam daur pendapatan termasuk :

- Jenis barang dan layanan yang disediakan, cara penyediaan, dan otorisasi pelanggan.
 - Masalah kredit dan batasannya harus diotorisasi dengan tepat.
 - Harga dan masalah mengenai penjualan barang dan layanan harus diotorisasi dengan tepat.
 - Pengurangan dan pengaturan yang berhubungan dengan penjualan harus diotorisasi dengan tepat.
 - Pengiriman barang dan layanan harus disiapkan dengan formulir billing yang akurat dan tepat waktu.
 - Penjualan dan transaksi yang berhubungan harus direkam sesuai dengan, jumlahnya pada periode yang tepat dan harus diklasifikasikan dengan tepat di dalam account.
 - Penerimaan kas harus dicatat dengan tepat berbasis ketepatan waktu.
 - Akses ke penerimaan kas dan catatan penerimaan kas, catatan accounts receivable, dan catatan billing dan pengiriman harus dikendalikan untuk menghindari atau mendeteksi intersepsi dari penerimaan kas yang tidak tercatat.
- Daur pengeluaran
Daur pengeluaran dibagi menjadi fungsi pembelian, penggajian dan pembayaran.
Pembelian meliputi fungsi penerimaan permintaan untuk barang, aset lain dan layanan; perolehan informasi mengenai vendor yang tersedia, harga dan spesifikasi lainnya: pemesanan barang; penerimaan dan pemeriksaan barang yang dikirim atau disediakan; penghitungan jumlah yang harus dibayarkan ke vendor, termasuk pengangkutan, potongan kas. Barang kembali dan pengaturan lain.

Penggajian meliputi fungsi pengangkatan pegawai dan penentuan kompensasinya, langsung dan tak langsung; pelaporan kehadiran dan pelaksanaan pekerjaan; perhitungan biaya penggajian, potongan, keuntungan pegawai dan pengaturan lainnya. Pembayaran meliputi fungsi penyiapan, pengesahan dan pengeluaran cek atau distribusi kas.

Kriteria masalah fungsi pembelian mencakup :

- Jenis barang, aset lain, dan layanan untuk diperoleh. cara perolehan, informasi vendor, kuantitas yang diperoleh dan harga serta masalah penjualan lain harus diotorisasi dengan tepat.
- Pengaturan ke account vendor dan distribusi account harus diotorisasi dengan tepat.
- Semua barang, aset lain, dan layanan yang diterima harus dihitung dengan tepat berbasis ketepatan waktu.
- Jumlah pembayaran untuk barang dan layanan yang diterima harus dicatat sesuai dengan jumlahnya pada periode yang tepat dan harus diklasifikasikan dengan tepat di dalam account.
- Akses ke pembelian, penerimaan, dan catatan account payable harus dikendalikan untuk menghindari atau mendeteksi pembayaran yang tidak tepat atau mengalami duplikasi.
- Hanya barang, aset lain atau layanan yang diotorisasi yang dapat diterima dan atau dibayar.

Kriteria masalah fungsi penggajian mencakup:

- Pegawai, keuntungan pegawai dan penghasilan tambahan harus diotorisasi dengan tepat.
- Kompensasi harus dibuat pada tingkat yang diotorisasi untuk layanan yang diberikan dan pengurangan gaji serta pengaturan account yang berhubungan dengan penggajian harus diotorisasi dengan tepat.
- Penggajian yang direkam harus sesuai dengan tugas yang telah dilaksanakan.
- Penggajian dan pemotongan pajak yang berhubungan harus dihitung dengan tepat dan dibayarkan pada waktunya.
- Biaya penggajian harus dicatat sesuai dengan jumlahnya pada periode yang tepat dan harus diklasifikasikan dengan tepat di dalam account.
- Akses ke personal dan catatan penggajian harus dikendalikan untuk menghindari atau mendeteksi pembayaran yang tidak tepat atau mengalami duplikasi.
- Pembayaran harus diberikan hanya pada pegawai yang diotorisasi.

Kriteria masalah fungsi pembayaran mencakup:

- Pembayaran harus dibuat hanya untuk pengeluaran yang diotorisasi dengan tepat.
- Pengaturan account kas harus diotorisasi dengan tepat.
- Pembayaran harus dicatat sesuai dengan jumlahnya pada periode yang tepat dan harus diklasifikasikan dengan tepat di dalam account.
- Akses ke kas dan catatan pembayaran kas harus dikendalikan untuk menghindari atau mendeteksi pembayaran yang tidak tepat atau mengalami duplikasi.
- Pembayaran harus diberikan hanya pada pegawai yang diotorisasi.

Daur Konversi atau Produksi

Daur konversi atau produksi meliputi fungsi perencanaan dan pengendalian produk, perencanaan dan pengendalian inventori, properti dan akuntansi biaya yang ditunda, serta akuntansi biaya.

Kriteria masalah untuk daur konversi atau produksi mencakup:

- Jenis dan kuantitas barang yang akan diproduksi atau layanan yang disediakan, metoda dan materi yang dipakai, tingkat inventori atau kemampuan layanan yang dikelola, dan penjadwalan barang yang diproduksi atau layanan yang disediakan harus diotorisasi dengan tepat.
- Penyesuaian inventori, properti, biaya yang ditunda, dan biaya penjualan harus diotorisasi dengan tepat.
- Penempatan properti, inventori yang berlebihan atau kadaluwarsa harus diotorisasi dengan tepat.
- Sumber daya yang dipakai dan produksi yang telah selesai harus dicatat dengan tepat berbasis ketepatan waktu.
- inventori, biaya produksi, depresiasi properti, dan amortisasi biaya yang ditunda harus diakumulasi dengan tepat dan diklasifikasikan dalam account.
- Semua biaya penjualan harus dicatat sesuai dengan jumlahnya pada periode yang tepat dan harus diklasifikasikan dengan tepat di dalam account.
- Inventori harus diproteksi dan penggunaan atau pemindahan yang tidak diotorisasi.

- Item properti harus dikendalikan dengan tepat.
- Akses ke inventori, properti, biaya dan catatan kendali produksi harus dikendalikan untuk menghindari atau mendeteksi penempatan inventori dan properti yang tidak tepat.

Gambar 17 memberikan contoh lembar kerja untuk merekam hasil tugas I.

Tugas 2 - Mendefinisikan Biaya Masalah

Perhitungan biaya masalah dalam suatu aplikasi otomatis dihitung menggunakan formula berikut :

$$A = F * L$$

A : Ekspektasi kerugian tahunan dari masaiah

F : Frekuensi kejadian

L : Kerugian per kejadian

Rumus tersebut menunjukkan bahwa suatu estimasi perlu dibuat berkenaan dengan berapa kali masalah akan terjadi dalam 1 tahun, ditambah rata-rata kerugian yang diharapkan untuk setiap kejadian.

Berikut ini dua contoh dari penggunaan rumus ini untuk menghitung kerugian potensial.

- Perhitungan biaya kesalahan pemasukan data. Salah satu kesalahan pemasukan data dalam aplikasi billing adalah pemasukan harga produk yang salah. Dengan mengetahui kerugian potensial yang berhubungan dengan masalah ini akan menjadi dasar penentuan derajat pengujian yang harus dilakukan.
- Perhitungan biaya kesalahan pemrograman. Salah satu pertimbangan dalam pengujian adalah bahwa program akan mengandung cacat. Hasil dari kegagalan jni dapat berupa akhir yang tak normal dari suatu aplikasi untuk perioda tertentu. Hasil kerugian pengolahan tersebut harus dikuantifikasikan.

Tugas 3 - Menentukan Tingkat Penerimaan dari Masalah

Ukuran pengujian adalah faktor dari tingkat kerugian yang dapat terjadi tanpa pengujian dan tingkat kerugian yang dapat diterima pemakai. Jika tingkat kerugian dapat diterima, pengujian tidak diperlukan. Pengujian hanya perlu dilakukan jika kerugian yang diestimasi tanpa pengujian melampaui tingkat kerugian yang dapat diterima.

Tingkat yang dapat diterima dari suatu masalah di dalam sistem adalah keputusan pemakai. informasi yang dapat membantu dalam pengambilan keputusan adalah:

- Biaya masalah tanpa pengujian yang memadai.
- Dasar penentuan biaya masalah.
- Biaya pelaksanaan pengujian
- Pengurangan ukuran masalah yang dapat diharapkan berkenaan dengan pengujian.

Tugas 4 - Mendefinisikan Obiektif Penguiian

Objektif pengujian harus memenuhi karakteristik berikut :

- Digambarkan cukup rinci sehingga dapat dimengerti.
- Dapat dicapai
- Dinyatakan dengan presisi yang cukup sehingga dapat diukur

Kegunaan objektif pengujian adalah menetapkan lingkup pengujian dan menentukan bilamana pengujian selesai.

Objektif pengujian yang dapat diukur dan dapat dicapai, haruslah :

- Memastikan bahwa sistem dapat dipulihkan dalam dua jam setelah terminasi yang tidak normal.
- Memastikan bahwa struktur program berfungsi dengan tepat dengan menguji minimal 80% dari instruksi program
- Memverifikasi bahwa kondisi yang dispesifikasikan dalam matriks uji adalah benar.

Gambar 18 menunjukkan lembar kerja untuk mendefinisikan objektif pengujian.

Langkah 2 - Memilih Pendekatan Uji

Pendekatan uji mendefinisikan siapa yang akan melaksanakan pengujian. Pilihan pendekatan uji yang tersedia untuk pelaksanaan uji mencakup:

- Personal pengembangan sistem
- Personal pemakai
- Grup penguji profesional, seperti quality assurance
- Auditor internal
- Kombinasi dari yang di atas
- Memakai jasa konsultan luar.

Pendekatan yang dipilih bervariasi tergantung kepentingan aplikasi yang akan diuji.

Sistem beresiko rendah cukup diuji oleh tim proyek pengolahan data, sedangkan sistem beresiko tinggi dapat melibatkan quality assurance analyst atau jasa konsultan luar.

Langkah 3 - Menentukan Penanggung Jawab Uji

Pada pengujian secara tradisional, penerimaan sistem biasanya menjadi tanggung jawab pemakai. Akhir dari pengujian ditentukan oleh pemakai.

Setelah pengujian daur hidup diperkenalkan, pengujian tidak lagi merupakan proses berfase tunggal. Pengujian terjadi di semua fase, sehingga lebih sulit menentukan penanggung jawabnya. Pendekatan yang lebih logis adalah dengan dominasi pada fase pengembangan: Penanggung jawab untuk setiap fase pengembangan juga bertanggung jawab atas pengujian pada fase tersebut. Misalnya, departemen pengolahan data adalah penanggung jawab perancangan dan pengkodean program, maka pengujian program menjadi tanggung jawab departemen pengolahan data.

Pengujian pada setiap fase berguna untuk verifikasi kebenaran fungsi yang dilaksanakan, juga untuk memutuskan apakah fase tersebut telah diselesaikan dengan baik. Jika penanggung jawab berpendapat bahwa fase tersebut belum cukup lengkap, maka sistem aplikasi tidak diperbolehkan untuk masuk ke fase berikutnya.

Langkah 4 - Mendefinisikan Kondisi Pengujian

Proses pengembangan sistem membagi dan mendefinisikan kebutuhan menjadi tugas

yang lebih terinci. Kerumitan pengolahan data terletak pada kemampuan manusia untuk membagi tugas yang kompleks menjadi langkah-langkah yang terinci.

Pengujian dari sudut pandang pengolahan data berarti mengevaluasi apakah eksekusi berbagai proses dan data akan memberikan hasil yang diinginkan.

Pada gambar 19 ditunjukkan konsep perancangan dan pengujian secara tradisional dalam pengolahan data. Gambar tersebut memperlihatkan bahwa proses perancangan mencoba untuk membagi kebutuhan menjadi definisi pemrograman dan data. Sebaliknya, proses pengujian merupakan kebalikannya, yaitu mulai dengan definisi program dan data, kemudian mengikuti proses untuk menentukan apakah program dan data tersebut dapat menghasilkan kebutuhan sistem yang diinginkan.

Permasalahan dengan konsep ini ialah kesalahan dalam proses perancangan seringkali tidak terdeteksi di proses pengujian. Alasannya adalah karena hasil dari perancangan seringkali menjadi basis untuk pengujian. Pengalaman menunjukkan bahwa dua pertiga masalah yang terjadi dalam kebutuhan dan perancangan tidak diterima sebagai kondisi pengujian.

Solusi untuk dilema ini adalah mendefinisikan kondisi pengujian dengan memakai sumber selain definisi program dan data. Jika sumber lain digunakan, pengujian menjadi *redundancy check*. *Redundancy check* melaksanakan fungsi yang sama dalam dua cara yang berbeda dan membandingkan hasilnya.

Gambar 20 menunjukkan konsep pengujian dengan redundansi.

Pendefinisian Kebutuhan Untuk Pengujian

Metoda yang direkomendasikan untuk pendefinisian kondisi pengujian adalah bekerja dari kumpulan kebutuhan awal yang sama. Jika arah panah pada gambar 19 dibalik, baik perancangan maupun pengujian menjadi *redundancy check*. Namun demikian, proses ini membutuhkan redefinisi kebutuhan untuk pengujian.

Kondisi pengujian dapat didefinisikan ulang dengan penggambaran kebutuhan sistem dalam terminologi pemakai. Proses pendefinisian sistem mengarah pada pendokumentasian kebutuhan dalam suatu format yang melengkapi proses perancangan sistem.

Kebutuhan pengujian yang didefinisikan dalam terminologi pemakai dapat dinyatakan dalam dua dimensi :

- Kejadian ekonomis, yaitu kejadian pemakai yang mengkonsumsi atau memproduksi sumber daya
- Aksi, yaitu tugas yang dilaksanakan pada kejadian ekonomis.

Kejadian ekonomis dapat dinyatakan sebagai ohjektif pemakai, dan aksi adalah pengolahan yang harus terjadi pada data atau kejadian. Perbedaan konsep ini dengan konsep pengolahan data adalah kejadian ekonomis tidak langsung ditranslasi menjadi program komputer tunggal.

Pendefinisian kebutuhan menjadi kejadian ekonomis dan aksi bersifat tidak bergantung pada komputer. Aksi melibatkan baik pengolahan manual maupun otomatis, sedangkan kejadian ekonomis menyatakan fungsi untuk mana pemakai bertanggung jawab. Bila keduanya digambarkan dalam bentuk matriks, kotak dalam matriks mendefinisikan kondisi yang perlu diuji.

Pengertian Kejadian Ekonomis

Untuk keperluan pengujian, kejadian ekonomis didefinisikan sebagai transaksi yang berorientasi bisnis. Dengan demikian, transaksi ekonomis harus *stand-alone* sehingga menghasilkan kejadian bisnis. Kejadian biasanya melibatkan komitmen dari sumber daya atau pembangkitan pendapatan.

Contoh kejadian ekonomis :

- Kejadian ekonomis penggajian :
 - penambahan pegawai baru
 - pemberhentian seorang pegawai
 - transfer pegawai
 - Peningkatan gaji pegawai
 - perekaman jam kerja
 - perubahan pengurangan gaji
 - perekaman absensi
 - pembayaran
- Kejadian ekonomis aplikasi billing :
 - penerimaan pesanan
 - pembuatan rekening
 - pembuatan dokumen pengiriman
 - persetujuan kredit pelanggan
 - pengemhalian item pembelian
 - pemesanan kemhali karena kekurangan stok

Kejadian ekonomis umumnya dapat ditentukan dari tanggung jawab yang termasuk dalam sistem aplikasi. Dalam hanyak hal.. terdapat translasi satu ke satu antara objektif sistem dengan kejadian ekonomis. Misalnya, objektif sistem adalah menyetujui kredit pelanggan, yang dapat ditranslasi menjadi kejadian ekonomis persetujuan kredit.

Pengertian Aksi

Untuk keperluan pengujian, aksi didefinisikan sebagai Suatu tugas atau proses yang dilaksanakan untuk membantu pemenuhan kejadian ekonomis.

Contoh aksi :

- Konsep aksi generik
Aksi generik menggambarkan tugas umum yang dilaksanakan pada kejadian ekonomis. Kumpulan tugas yang dipakai adalah tugas yang dipakai dengan lebih

umum dan dapat dimengerti oleh sebagian besar personal sistem. Misalnya :

- pembuatan formulir
 - pengisian formulir
 - pengiriman data
 - pembuatan total kendali
 - peremajaan total
 - penyimpanan informasi
 - pengambilan informasi
 - pembuatan laporan
 - pelaksanaan perhitungan
- Aksi Spesifik
Aksi dapat juga dinyatakan dalam tugas yang harus dilaksanakan oleh sistem khusus.
Misalnya :
 - penerimaan pesanan
 - verifikasi informasi pelanggan
 - verifikasi informasi produk
 - verifikasi ketersediaan produk
 - permintaan ke bagian kredit untuk persetujuan kredit pelanggan

Penentuan aksi harus diarahkan untuk mengidentifikasi aksi *stand alone*. Setiap aksi harus dapat diidentifikasi sehingga hasilnya dapat ditentukan dengan mudah. Jika aksi terlalu luas, jenis kondisi uji yang harus dilakukan akan terlalu banyak. Sedangkan jika aksi terlalu spesifik, akan sulit untuk mendefinisikan kondisi uji yang berarti. Aksi yang terbaik adalah jenis aksi yang akan menghasilkan sejumlah kondisi uji yang terbatas.

Matriks Uji

Matriks uji menunjukkan hubungan antara kejadian ekonomis dengan aksi. Matriks uji yang lengkap mendefinisikan kondisi yang harus diuji selama proses uji untuk memverifikasi pelaksanaan fungsi dari sistem aplikasi.

Baris matriks menunjukkan kejadian ekonomis, sedangkan kolom matriks menunjukkan aksi. Sel matriks berisi proses yang perlu diuji.

Gambar 21 menunjukkan contoh matriks uji untuk kejadian ekonomis dari peningkatan gaji seorang pegawai.

Aksi untuk menggambarkan kondisi yang harus diuji :

1. Inisiasi aksi kejadian
2. Peningkatan aksi pengesahan
3. Aksi pemasukan data
4. Penyimpanan Formulir
5. Validasi Pemasukkan data
6. Aksi validasi logika
7. Aksi peremajaan record gaji
8. Aksi audit trail
9. Aksi pelaporan

Menyertakan hanya satu kejadian ekonomis pada matriks memberi dua keuntungan :

- Aksi dapat dibiasakan dibiasakan untuk kejadian ekonomi khusus
- Aksi pada kejadian ekonomis dapat berupa pembangkitan kejadian ekonomis baru yang menunjukkan hubungan antara kejadian

Satu kejadian ekonomis berperan pada pembangkitan kejadian yang lain yang menyebabkan beberapa matriks dipersiapkan. Gambar 22 menunjukkan *cascading matrix*, dimana satu kejadian dapat menciptakan kejadian lain yang kemudian menciptakan lagi kejadian baru.

Langkah 5 – Mengembangkan Rencana Administratif

Pengujian bersifat teknis dan administratif. Keduanya harus direncanakan agar pengujian berhasil. Seringkali, bagian teknis lebih dibandingkan bagian administratif.

Perencanaan administratif meliputi penetapan anggaran, pembuatab jadwal penugasan orang, reservasi waktu komputer, pembuatan dan penyimpanan data yang diperlukan, pemilihan dan perolehan alat dan teknik, serta beberapa konsiderasi. Gambar 23 menunjukkan rangkuman konsiderasi administratif. Tanpa perhatian yang cukup pada konsiderasi administratif, sumber daya dan fasilitas yang diperlukan untuk melaksanakan pengujian dapat tidak tersedia.

Penetapan Anggaran Pengujian

Anggaran menentukan jumlah dana yang dialokasikan untuk proses uji. Banyak organisasi tidak memisahkan biaya pengujian dari pengembangan yang menyulitkan pengendalian. Konsep pengujian daur hidup merekomendasikan 30% dari usaha pengembangan total dialokasikan untuk pengujian.

Beberapa metoda yang dipakai untuk mengestimasi dan mengukur jumlah waktu yang dialokasikan untuk pengujian sistem aplikasi :

- *Black estimating* – sejumlah anggaran pengembangan sistem yang telah ditetapkan sebelumnya dialokasikan untuk proses uji.
- *Customized estimating* – Semua karakteristik sistem dianalisis untuk mengembangkna anggaran pengujian yang dirancang secara khusus untuk aplikasi yang diuji. Anggaran akan melibatkan faktor-faktor seperti jumlah transaksi uji, jumlah program, kemampuan penguji, alat uji yang tersedia, jadwal untuk instalasi aplikasi, dsb.
- *Ball park estimates* – Estimasi yang dipersiapkan oleh estimator profesional atau supervisor didasarkan pada taksiran karakteristik proyek. Jika estimasi diterima oleh pemimpin proyek, ia harus memperkirakan penyelesaian pengujian dalam waktu yang dialokasikan. Jika estimasi tidak diterima , pemimpin proyek dapat meminta perubahan.
- *Risk attributes estimating* – Resiko aplikasi terhadap organisasi diidentifikasi dan besarnya resiko ditentukan. Tingkat pengujian didasarkan pada besarnya resiko, dan jumlah resiko yang dapat diterima pemakai dalam aplikasi setelah pengujian.

Penjadwalan pegujian:

Penjadwalan harus mempertimbangkan:

- Penetapan tanggal mulai dan berhentinya pengujian pada setiap fase
- Pemberitahuan personal yang terlibat dalam pengembangan proyek dan pengujian yang telah dijadwalkan
- Penetapan jalur uji kritis untuk memastikan bahwa semua pengujian dapat diselesaikan dalam kerangka waktu proyek.

Penugasan orang

Pengujian biasanya merupakan usaha tim yang melibatkan orang dari berbagai bagian organisasi. Jenis orang yang sebaiknya terlibat:

- Personal proyek
- Pemakai
- Penguji profesional
- Personal yang menyiapkan ICI
- Pemrograman sistem
- Operator/personal pustaka data

Reservasi waktu komputer

Pertimbangan penjadwalan waktu komputer mencakup:

- Reservasi waktu uji yang dijadwalkan dengan operasi komputer
- Penetapan mekanisme pengujian sehingga program uji akan tersedia pada pustaka yang benar dan hubungan lain yang diperlukan akan ditetapkan sehingga sistem aplikasi yang diuji akan berjalan
- Penyediaan instruksi operasi khusus untuk menjalankan pengujian.

Data Uji

Pertimbangan dalam penyiapan data uji:

- Pembuatan data uji dan atau transaksi yang diperlukan untukn memverifikasi fungsi sistem yang sebenarnya dan untuk menyebabkan kondisi lain seperti akhir yang tidak normal untuk menguji prosedur pemulihan
- Investigasi dan perolehan file komputer yang ada yang dipakai untuk pengujian
- Membuat pengaturan untuk mempertahankan data uji sebelum perioda uji
- Jika diperlukan, penyediaan media untuk penyimpanan dan pencetakan data selama pengujian

Pertimbangan Lainnya

- Penetapan tanggung jawab pemakai
- Kriteria dokumen uji penerimaan
- Prosedur administratif lokal lainnya

Rencana uji harus dipersiapkan pada awal daur hidup pengembangan sistem. Idealnya dipersiapkan sebelum fase kebutuhan berakhir. Pada saat itu rencana belum lengkap dan akan dilengkapi dan diisi selama proses pengembangan.

Umumnya penerima rencan uji adalah:

- Manajer proyek
- Pemakai sistem aplikasi
- Manajer operasi komputer
- Grup penguji profesional (mis. EDP quality assurance)
- Manajemen pengolahan data yang bertanggung jawab untuk paket aplikasi.

Referensi : Perry, W.E; A Structured Approach to Systems Testing; QED Information Science; 1983