

RENCANA TES PENERIMAAN

Tujuan : mendapatkan pernyataan tertulis dari user bahwa produk (dalam hal ini sistem) yang dikirim sesuai dengan yang dijanjikan.

METODE TES PENERIMAAN

1. PERIODE PERCOBAAN ATAU PARALLEL RUN (THE TRIAL PERIOD OR PARALLEL RUN)

Periode percobaan atau *parallel run* adalah pendekatan yang paling umum untuk penerimaan. Menggunakan pendekatan 'Periode Percobaan' tim proyek mudah memasang sistem baru untuk dicoba oleh user. Pendekatan '*Parallel Run*' menambahkan dimensi untuk peralihan sistem lama yang sudah berjalan dengan baik sebagai perbandingan dan cadangan.

Dalam kedua kasus ini klien menggunakan sistem baru selama 'X' hari. Jika tidak ada masalah maka user menerimanya, jika ada masalah maka tim proyek berusaha memperbaikinya dan melakukan kembali percobaan selama 'X' hari.

Kekurangan :

1. Masalah kecil dapat membuat anda menjalankan kembali selama 'X' hari untuk jangka waktu yang tidak terbatas. Kadang-kadang sistem software yang rumit tidak pernah 100% *di-debug*.

2. Mungkin sulit untuk mencari penyebab dari suatu masalah. Jika 10 user berada pada sistem yang interaktif dan sistem tersebut rusak, ini merupakan tantangan untuk menemukan dengan tepat apa yang menyebabkan sistem tersebut rusak.
3. Tidak ada jaminan bahwa semua kelebihan sistem akan dicoba dalam 'X' hari. Penulis pernah melihat sebuah sistem akuntansi yang diterapkan pada awal tahun fiskal baru. Sistem itu berjalan baik selama masa percobaan (6 bulan) sampai mengalami kegagalan pada akhir tahun fiskal ketika akuntan mencoba untuk melakukan tutup buku. Sayangnya garansinya telah habis dan penjual (*vendor*) tidak mau memperbaikinya.
4. Biarkan *end user* masuk ke sistem pada hari pertama yang penerapannya tidak selalu bermanfaat. Karena dalam hal ini faktor penampilan lebih berperan. Seperti dalam roman, kesan pertama sangat penting.

2. SOLUSI : PENERIMAAN YANG LENGKAP SEDIKIT DEMI SEDIKIT (Solution : A Thorough But Piecemeal Acceptance)

Pendekatan yang lebih baik adalah menemukan serangkaian tes yang mendemonstrasikan semua fungsi yang dijanjikan.

Penerimaan akan dilakukan secara resmi melalui seluruh tes ini kepada pelanggan. Keberhasilan tes diakhiri satu per satu.

Jika sebuah tes gagal, Tim proyek dengan penuh harapan memperbaiki masalah langsung di tempat pengujian.

Jika itu masalah utama maka tes ditunda sampai masalah dapat diperbaiki.

Dalam teori hanya tes yang gagal yang diulang, walaupun user memiliki hak untuk menjalankan kembali tes yang diterimanya sesudah perbaikan.

Serangkaian tes dan perintah yang dijalankan sistem disebut **Rencana Tes Penerimaan (Acceptance Test Plan / ATP)**.

Manfaat / Keuntungan :

1. Anda dapat mendemonstrasikan semua fungsi yang dijanjikan.
2. Sebuah tindakan yang menyebabkan masalah selalu diketahui – anda mengetahui dengan tepat siapa yang mengetik ketika masalah terjadi.
3. User tidak merasa takut tentang semuanya.

Kerugian utama :

Memerlukan banyak pekerjaan untuk menulis ATP. Dan lagi user mungkin tidak lazim dengan pendekatan ini. Tetapi anda dapat membiasakan mereka dengan metode baru sebelumnya.

Cantumkan laporan singkat dalam proposal, yang merupakan sebuah dokumen yang ditandatangani. Selengkapnya ada dalam Spesifikasi Fungsi, dokumen lainnya yang ditandatangani. Dia akan selalu melihat dan menyetujui ATP sebelum penerimaan. Seharusnya tidak ada keengganan untuk menerima dan membayar jika metode ini digunakan.

3. MEMASTIKAN BAHWA SEMUA YANG DIJANJIKAN AKAN DIUJI (ENSURING THAT ALL THE PROMISES ARE TESTED)

Untuk memastikan semua yang dijanjikan akan dites langsung melalui Spesifikasi Fungsi halaman demi halaman, paragraf demi paragraf, dan buat daftar semua fungsi yang dapat dites. Perhatikan tabel seperti gambar ini :

FS REF SEC/PAR	FUNCTION TO BE TESTED	TEST METHOD	TEST NUMBER
3.1	Main menu appears at start-up	T	1.0
3.2	Registrar menu appears when...	T	2.1
3.3	Manager menu appears when...	T	2.2
7.7	Store 10.000 student records	I	7.8
10.2	Students by course by city report	T,A	4.5

T – test I – inspection A – Analysis (Hand calc, or use another pgm)
N/A – not applicable

LANGKAH MENYUSUN TES PENERIMAAN

1. MENGGUNAKAN DISAIN (*USING THE DESIGN*)

1. Pekerjaan menyusun ATP dilakukan setelah disain dikerjakan. Kita memerlukan Spesifikasi Fungsi untuk menghasilkan ATP.
2. Disain membantu untuk mengelompokkan tes ke dalam serangkaian tes yang mendemonstrasikan fungsi utama dari sistem.
3. Yang terpenting adalah tes dapat dimengerti dengan baik oleh user anda.
4. Prinsip desain tes ini adalah dapat mendemonstrasikan semua menu, kemudian seluruh keterangan yang diminta, diikuti dengan semua update, dsb. Cara lain untuk mengelompokkan kumpulan tes adalah dengan fungsi. Melalui semua fungsi Registrasi, diikuti oleh fungsi Administrator, dsb.

2. MENULIS PERCOBAAN (*WRITING TEST*)

Anda sudah siap menentukan bagaimana anda akan menguji item ketika pengisian pada METODE PERCOBAAN seperti pada gambar 8.1. di atas.

Berikut ini adalah contoh percobaan nomor 4.5., laporan '***Students by Course by City***'. [Catatan dalam tanda kurung siku-siku tidak ditampilkan – ini adalah keterangan].

TEST NO:	4.5
TEST PURPOSE:	Demonstrate the production of the Students by Course by City report.
F.S. REFERENCE (SEC/PAR):	10.2, 12.8, 11.3 [Note how one test can demonstrate several functions.]
SETUP:	Ensure files STUDENT.DAT and COURSE.DAT contain data.Star system.
INPUT:	Choose Selection 1. From Main Menu using mouse click and drag method.
OUTPUT:	"CHOOSE REPORT TYPE" menu (format per FS pg 17, Figure 8.15) appears. [Refer to the Functional Spec. whenever possible.]
INPUT:	Choose Selection 5. (Students by Course by City report) by UP/DOWN arrow and RETURN method.
OUTPUT:	Message 'Report being prepared.' Appears. No longer than 60 seconds later, message 'Report being printed' appears and printer starts printing. The terminal can be used to enter any other command. User will try up to 3 commands of his choice. [There is danger in stating, "User may type any number of commands."He just may!] When the report is complete (printer stops,) inspect it to ensure it is of format FS pg.23,Figure 12.12. Total columns will be checked by hand calculated addition of the attendance figures printed.
USER SIGNATURE :	
PROJECT TEAM SIGNATURE :	
DATE :	
COMMENTS :	

Gambar Contoh Typical test

3. MEMBUAT DAFTAR RENCANA TES PENERIMAAN (THE ACCEPTANCE TEST PLAN CHECKLIST)

Gunakan hal berikut sebagai daftar pengecekan untuk semua kegiatan yang diperlukan untuk rencana penerimaan :

- Hasilkan Fungsi vs. Tabel Percobaan dan semua FS yang dijanjikan telah dialamatkan.
- Definisikan percobaan dan kumpulan percobaan.
- Tetapkan tanggung jawab untuk menulis percobaan.
- Klien dan Tim proyek mengetahui bahwa ATP akan ditinjau kembali, direvisi jika perlu, dan ditandatangani oleh user. Klien mengetahui bahwa keberhasilan penyelesaian dari percobaan akan mempengaruhi penerimaan sistem. Lihat bentuk contoh ATP pada bagian 10 di Appendix A.
- Tanggung jawab untuk percobaan data telah ditetapkan. Data untuk percobaan seharusnya disediakan oleh tim proyek dan juga user. Jika user dapat menyediakan data yang sesuai dengan keadaan yang sebenarnya, percobaan terhadap sistem akan berjalan dengan baik, ditambah user akan merasa nyaman dengan keakuratan percobaannya.

KESIMPULAN UNTUK RENCANA TES PENERIMAAN (CONCLUSION TO THE ACCEPTANCE TEST PLAN)

Anjurkan user untuk menulis ATP jika dia mampu. Hal ini akan memberikan dia perasaan mengawasi – tim proyek harus membangun sistem melalui percobaan.

Anda dapat melakukan tes penerimaan secara berlebihan. Membandingkan biaya tes dengan biaya risiko itu adalah suatu masalah. Anda dapat tidak melakukan semua percobaan, khususnya dalam sistem *multi-user* yang interaktif.

KESIMPULAN UNTUK TAHAP DISAIN (CONCLUSION TO THE DESIGN PHASE)

Pada akhir tahap disain kita menempuh beberapa kejadian penting sebagai berikut :

1. Dokumen Spesifikasi Disain memuat disain akhir tingkat atas melalui disain tingkat menengah.
 2. Tanggung jawab ATP disahkan dan dimulai. Ini tidak perlu diselesaikan sampai tahap penerimaan.
- Rencana proyek, khususnya perkiraan perlu ditinjau kembali.
 - Tahap pemrograman mungkin akan menjadi tahap yang sangat mahal dan membutuhkan waktu yang sangat banyak dalam keseluruhan kerja proyek.
 - Disain akan memberikan perkiraan perhitungan jumlah modul-modul dan kerumitannya.
 - Dengan menunjukkan desain kita bisa melihat siapa programmer-programmer yang dapat diandalkan, sehingga anda dapat mempertimbangkan faktor produktivitas mereka.
 - Statistik menunjukkan bahwa pada akhir tahap disain diperkirakan seharusnya tidak lebih dari 10%.