

BAB I

Pengujian Perangkat Lunak

1.1 Pengertian Pengujian

Pengujian Perangkat Lunak (Software Testing) adalah suatu teknik yang digunakan untuk menentukan bahwa perangkat lunak yang dihasilkan telah memecahkan masalah. Pengujian Perangkat Lunak termasuk salah satu langkah dalam metodologi pengembangan sistem (SDLC: *System Development Life Cycle*). Namun, pada setiap aktivitas SDLC yang dilakukan pengujian tetap harus dilakukan.

Tiga konsep yang harus diperhatikan dalam Pengujian Perangkat Lunak, yaitu:

- Demonstrasi validitas perangkat lunak pada setiap tahapan pembangunan sistem.
- Penentuan validitas sistem akhir terhadap pemakai dan kebutuhan.
- Pemeriksaan implementasi sistem dengan menjalankan sistem pada suatu contoh data uji

1.2 Kadar Kritis dan Resiko Sistem

Faktor utama dari pengujian perangkat lunak adalah kadar kritis serta resiko yang ditimbulkannya. Kadar kritis ini di dalam pengujian akan memvalidasi cara mengatasi/solusinya. Sistem komputer untuk pengaturan pendaratan pesawat atau aplikasi transfer keuangan membutuhkan ekstra perhatian daripada program pengaturan parkir mobil misalnya, karena pada program semacam itu bila terjadi kesalahan akan mengakibatkan kerugian yang fatal. Dalam validasi solusi harus memperhatikan masalah ukuran proyek, keunikan, kadar kritis, biaya terjadi kesalahan, dan biaya proyek.

Resiko adalah kondisi yang dapat menyebabkan kerugian. Dalam hal ini resiko berhubungan dengan kemungkinan terjadinya kehilangan. Situasi ini selalu terjadi, walau kehilangan akhirnya tidak terjadi. Misalnya bahaya kebakaran selalu ada tetapi jarang terjadi kebakaran. Walau bagaimanapun, kita harus berjaga-jaga terhadap resiko tersebut, misalnya dengan memasang alarm khusus, alat pemadam kebakaran, sehingga kemungkinan terjadi kebakaran akan berkurang. Kita tidak dapat menghilangkan resiko, tapi kita dapat mengurangi kemunculannya dan/atau dampak kehilangan.

Dalam pengembangan dan instalasi sistem komputer, resiko akan selalu terlibat. Resiko ini harus diperhitungkan dalam tiap tahapan proses pengembangan untuk mengurangi kemungkinan kehilangan pada tiap tingkat. Metode efektif yang dapat digunakan untuk mengurangi resiko, adalah dengan pengujian.

Tiap resiko ini akan berakibat pada fungsi system. Ada beberapa contoh jenis resiko yang berhubungan dengan pengembangan dan instalasi perangkat lunak:

- Hasil yang salah
- Transaksi yang ilegal diterima oleh sistem
- Integritas file hilang
- Pemrosesan tidak bisa dilakukan
- Pelayanan terhadap pemakai akan berkurang
- Keamanan sistem akan dipertanyakan
- Pemrosesan tidak sesuai dengan kebijakan organisasi atau peraturan pemerintah

- Hasil dari sistem tidak dapat diandalkan
- Sistem akan sulit untuk digunakan.
- Program tidak terawat
- Sistem tidak dapat di-port ke perangkat lain
- Sistem tidak dapat berhubungan dengan perangkat lain
- Tingkat perfomansi jelek
- Sistem akan sulit beroperasi

Pendekatan yang efektif dalam pengujian adalah dengan mengenali dan mengevaluasi resiko pada sistem komputer. Suatu keputusan dapat dibuat untuk menentukan seberapa besar suatu resiko dapat diterima dan suatu rencana pengujian harus dilakukan untuk mendapatkan hasil yang baik. Dengan alasan ekonomis, dapat ditentukan apakah dampak bagi sistem dapat diterima, atau seberapa jauh diterimanya. Penentuan bagus tidaknya suatu pengujian diserahkan dari suatu tim yang terdiri dari programmer/sistem analis dengan mempertimbangkan keputusan bisnis.

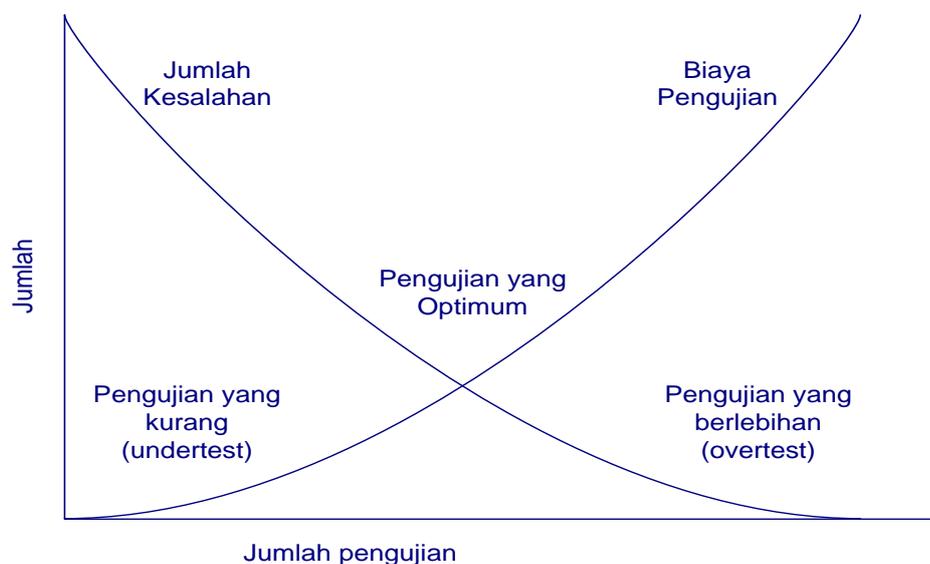
1.3 Faktor Ekonomi dalam Pengujian

Seorang manajer EDP mengatakan pengujian sebagai "*Too little testing is a crime, too much testing is a sin*". Jika pengaturan dipandang sebagai resiko, maka ada resiko atas pengujian yang terlalu banyak (*overtesting*) atau terlalu sedikit (*undertesting*). Resiko yang diakibatkan terlalu sedikit akan berakibat langsung pada sistem saat digunakan. Sedangkan resiko dari pengujian terlalu banyak adalah akan menimbulkan tidak efektif dan efisien, dilihat dari penggunaan sumber daya dan biaya.

Masalah yang sering muncul berhubungan dengan proses pengujian diantaranya adalah:

- Gagal mendefinisikan tujuan pengujian
- Pengujian dilakukan pada tingkat yang salah pada suatu siklus pengembangan sistem
- Penggunaan teknik pengujian yang tidak efektif.

Kondisi efektifitas sistem pengujian dapat dilihat pada gambar berikut:



Pada gambar tersebut terlihat bahwa ketika biaya pengujian bertambah, maka jumlah dampak yang tidak terdeteksi menurun. Pada sisi kiri menggambarkan situasi *undertest* dimana biaya pengujian kurang dari kehilangan karena dampak yang tidak terdeteksi. Pada suatu titik kedua garis bertemu, dan kondisi *overtest* dimulai. Pada situasi ini, biaya pengujian untuk mendapatkan kelemahan akan meningkat, melebihi biaya kehilangan bila kelemahan itu tidak ditemukan. Pengujian dari sudut pandang efektivitas biaya, berarti pengujian harus dilakukan sampai dicapai kondisi optimum.

Beberapa instansi sudah membentuk dasar pengukuran efektivitas dari pengujian. Hal ini menyulitkan bagi sistem analyst/programmer dalam menentukan efektivitas sistem. Tanpa pengujian yang standard, efektivitas dari suatu proses tidak dapat dievaluasi dengan cukup detail.

Penggunaan metodologi yang standard, dapat menentukan hubungan sebab akibat. Dengan kata lain, efek dari perubahan metodologi dapat dievaluasi untuk menentukan apakah efek tersebut dihasilkan dari jumlah dampak yang lebih kecil atau lebih besar. Pembentukan hubungan ini adalah langkah yang perlu dalam meningkatkan pemrosesan sistem.

1.4 Masalah Sistem Komputer yang Umum

Pada umumnya, masalah yang dihadapi sistem komputer ada dua yaitu masalah perangkat lunak dan masalah data. Masalah itu menyebabkan situasi yang tidak ekonomis atau menyebabkan suatu aksi yang salah.

1.4.1 Masalah perangkat Lunak

Persoalan yang dapat diidentifikasi dalam perangkat lunak, yang sering menyebabkan keputusan yang buruk, antara lain:

- Perancangan perangkat lunak dengan kriteria pengambilan keputusan yang tidak lengkap atau salah. Aksi akan menjadi salah, karena logika pengambilan keputusan telah menghilangkan faktor yang seharusnya dimasukkan. Pada kasus lain, kriteria pengambilan keputusan kurang cocok, baik pada saat perancangan atau sesudahnya, karena perubahan suasana.
- Kegagalan untuk memprogram perangkat lunak yang sesuai dengan keinginan pemakai atau perancang akan mengakibatkan kesalahan logika yang kadang disebut sebagai kesalahan pemrograman.
- Penghilangan pemeriksaan edit yang dibutuhkan untuk menentukan kelengkapan data masukan. Misalnya suatu data kritis dibiarkan kosong, dan karena pemeriksaan kurang, maka aplikasi akan memproses transaksi dengan data yang tidak lengkap.

1.4.2 Masalah Data

Kualitas pemasukan data sering menjadi masalah. Karena data ini adalah bagian yang integral dari proses pengambilan keputusan, maka kualitas yang jelek akan berpengaruh pada aksi perangkat lunak. Masalah data yang umum adalah:

- Data yang tidak lengkap
- Data yang salah
- Data yang usang

1.5 Pengujian adalah Masalah Organisasi

Pengujian pada sistem pemrosesan data tidak hanya masalah pemrosesan data, tapi menyangkut masalah organisasi. Bagian pemrosesan data dapat memverifikasi struktur sistem bekerja dengan benar, juga sesuai dengan permintaan, tapi bagian itu tidak dapat menguji untuk apakah sistem yang akan digunakan itu sesuai dengan kebutuhan organisasi.

Pengujian yang efektif harus terdiri dari suatu tim yang terdiri dari para profesional dan pemakai. Jika pemakai tidak dapat hadir, maka mereka akan digantikan oleh kelompok penguji yang profesional. Kelompok ini memiliki nama yang berbeda-beda misalnya pengujian MIS, Quality Control Quality Assurance dan Inspectors. Kebijakan dari pengujian dikembangkan oleh bagian pemrosesan data, dengan melihat filosofi keseluruhan organisasi.

Perkembangan teknologi juga dapat menyebabkan pendekatan organisasi di dalam proses pengujian pengujian, yaitu::

- *Integrasi* - teknologi makin terintegrasi dengan masalah-masalah bisnis, sedemikian rupa sehingga bisnis tidak akan jalan tanpa teknologi komputer.
- *Hubungan Sistem (System Chain)* - Suatu sistem komputer biasanya saling berhubungan dengan sistem lain, sehingga kadang-kadang suatu kesalahan bisa berakibat pada sistem lain.
- *Efek Domino*: Suatu kesalahan tunggal, dapat menyebabkan ratusan kesalahan yang sama dalam waktu singkat.
- *Keandalan peralatan elektronik*
- *Multiple User*

1.6 Mengembangkan Kebijakan Pengujian

Terdapat empat kriteria dalam membangun/mengembangkan proses pengujian, yaitu:

- *Definisi Pengujian*. Definisi pengujian harus jelas, singkat dan tidak meragukan
- *Sistem Pengujian*. Metode di mana pengujian dapat dilakukan
- *Evaluasi*. Bagaimana manajemen pemroses data akan diukur dan dievaluasi pengujian.
- *S'tandard*. Standar dimana pengujian akan diukur.

Pembangunan dari kebijaksanaan pengujian, adalah tanggung jawab manajemen pemrosesan data. Tiga metode dapat digunakan untuk membentuk kebijaksanaan pengujian, antara lain:

- *Arahan Manajemen*. Salah seorang dari manajemen yang dianggap senior bisa memberikan kebijaksanaan. Mereka menentukan apa yang diharapkan dari pengujian, dokumen yang terlibat. Cara ini menguntungkan secara ekonomis serta efektif. Kerugiannya ialah kebijaksanaannya bersifat lebih ke arah manajemen pemrosesan data dibandingkan dengan organisasi.
- *Kebijaksanaan pemrosesan data berdasarkan konsensus*. Pihak manajemen membentuk kelompok yang terdiri dari senior dan orang-orang yang dipercaya untuk mengembangkan kebijaksanaan. Dari pihak senior akan diharapkan tanggung jawabnya untuk menerima dan menyebarkan kebijaksanaan, sedangkan pengembangan kebijaksanaan adalah lebih banyak dari hasil pemikiran bagian pemroses data daripada pihak senior. Cara ini memberikan keuntungan bahwa karena manajemen atas terlibat, maka bawahannya akan terdorong untuk mengikuti. Kerugiannya ialah cara ini lebih bersifat kebijaksanaan bagian pemrosesan data dibandingkan kebijaksanaan organisasi.

- Pertemuan pemakai. Anggota senior dari pihak manajemen pemakai bertemu dengan pihak pemrosesan data untuk bersama-sama mengembangkan kebijaksanaan pengujian. Dengan cara ini keuntungannya ialah bersifat lebih ke arah kebijaksanaan organisasi dan melibatkan semua bidang. Kerugiannya ialah cara ini membutuhkan waktu, serta pihak pemrosesan data diharuskan untuk menerima karena sudah konsensus bersama serta hasilnya biasanya bukan seperti yang diinginkan oleh pihak pemrosesan data.

1.7 Pendekatan Terstruktur untuk Pengujian

Dalam pendekatan daur pengembangan system terstruktur tradisional melibatkan tahapan berikut:

- Analisa kebutuhan
- Perancangan
- Implementasi (Peng-kode-an)
- Integrasi
- Pengujian
- Operasi dan perawatan

Pengujian dilakukan setelah tahapan integrasi dan sebelum tahapan pengoperasian dan perawatan. Sering bahwa pengujian setelah dilakukan peng-kode-an adalah satu-satunya teknik verifikasi untuk menentukan keandalan sistem. Jika pengujian dibatasi pada fase tunggal seperti teknik tradisional tersebut, maka konsekuensi lain bisa timbul. Pengujian dapat menghabiskan 50% dari biaya pengembangan.

Jika kesalahan ditemukan di bagian lain, kesalahan yang sama harus dibayar empat kali, yaitu:

Pertama : Biaya pengembangan program yang salah, yang akan dapat melibatkan spesifikasi yang salah, peng-kode-an yang salah, dan dokumentasi yang tidak sesuai.

Kedua : Sistem harus diuji untuk pendeteksian kesalahan.

Ketiga : Spesifikasi dan peng-kode-an yang salah harus dihilangkan dan dibuat spesifikasi, peng-kode-an dan dokumentasi yang benar.

Keempat : Sistem harus kembali diuji untuk memeriksa kebenarannya.

Jika dalam pengujian lebih dipentingkan biaya yang rendah dan kualitas yang tinggi, maka verifikasi harus dilakukan pada tiap tahapan pengembangan. Suatu studi menunjukkan bahwa kesalahan sistem muncul lebih banyak pada tahap perancangan.

Contoh aspek-aspek yang diuji dalam proses pengujian akan diperlihatkan pada tabel dibawah ini dan verifikasi dilakukan untuk tiap fase pengembangan sistem :

Fase Pengembangan	Aktivitas Verifikasi
Analisa Kebutuhan	<ul style="list-style-type: none"> • Penentuan pendekatan verifikasi • Penentuan batasan kebutuhan • Buat data tes yang fungsional • Penentuan konsistensi disain dengan kebutuhan
Perancangan (Disain)	<ul style="list-style-type: none"> • Penentuan batasan rancangan • Buat data tes yang fungsional dan terstruktur • Penentuan konsistensi dalam perancangan
Peng-kode-an	<ul style="list-style-type: none"> • Penentuan batasan implementasi • Buat data tes yang fungsional dan terstruktur untuk program
Pengujian	<ul style="list-style-type: none"> • Uji sistem aplikasi
Instalasi	<ul style="list-style-type: none"> • Sistem yang sudah diuji coba
Perawatan	<ul style="list-style-type: none"> • Modifikasi dan ulangi pengujian

1.8 Metodologi Pengujian

Tujuan dari pengujian ialah mengurangi resiko. Oleh karena itu, metoda yang digunakan harus dapat menangani resiko. Dalam perancangan metodologi pengujian, faktor resiko menjadi dasar atau tujuan dari pengujian. Resiko yang dikaitkan dengan pengujian disebut faktor uji. Proses pengujian harus mengurangi faktor uji tersebut hingga suatu tingkat tertentu.

Resiko pengujian, menjadi faktor yang harus diperhatikan dalam strategi pengembangan pengujian. Faktor-faktor pengujian tersebut adalah:

- *Kebenaran (Correctness)*. Jaminan bahwa data dapat dimasukkan, diproses dan dikeluarkan oleh aplikasi dengan akurat dan lengkap, dengan pengontrolan transaksi dan elemen data.
- *Integritas File*. Jaminan bahwa data yang dimasukkan ke sistem aplikasi dapat diambil kembali tanpa ada perubahan. Dalam faktor ini harus dijamin penggunaan file yang benar.
- *Otorisasi (Authorization)*. Jaminan bahwa data diproses sesuai dengan keinginan manajemen. Dalam sistem aplikasi, terdapat otorisasi umum dan khusus dalam pemrosesan transaksi.
- *Audit trail*. Kemampuan untuk melakukan substansi terhadap pemrosesan yang telah dilakukan. Pemrosesan data dapat didukung dengan memperhatikan masalah keakuratan, kelengkapan, *timeliness* dan otorisasi data.
- *Kontinuitas Pemrosesan (Continuity of Processing)*. Kemampuan untuk menunjang pemrosesan bila muncul suatu event atau kejadian. Dengan demikian akan melibatkan suatu prosedur dan pembuatan back-up untuk mengatasi masalah tersebut sehingga integritas tidak hilang.
- *Tingkat Pelayanan (Access Control)*. Jaminan bahwa hasil yang diharapkan dapat tersedia selama waktu tertentu.
- *Kontrol Akses (Access Contro/)*. Jaminan bahwa sumber daya sistem diproteksi terhadap modifikasi yang tidak disengaja.

- *Compliance*. Jaminan bahwa sistem dirancang sesuai dengan metodologi organisasi, kebijaksanaan, prosedur dan standard. Kebutuhan itu harus diidentifikasi, diimplementasikan dan dirawat sesuai dengan kebutuhan aplikasi.
- *Keandalan (Reliability)*. Jaminan bahwa aplikasi akan menjalankan fungsinya dengan benar untuk waktu yang lama. Kebenaran dari suatu proses berhubungan dengan kemampuan sistem untuk memproses transaksi yang benar, sementara keandalan berhubungan dengan kemampuan sistem dapat menjalankan fungsinya untuk waktu yang lama.
- *Kemudahan dalam Penggunaan*. Faktor uji ini berhubungan dengan kemudahan pemakai yang menggunakannya.
- *Perawatan*. Usaha untuk mencari dan membetulkan kesalahan selama sistem sudah berjalan.
- *Portable*. Usaha untuk mengubah program dari suatu konfigurasi perangkat keras/lunak ke konfigurasi lain. Dalam usaha ini akan melibatkan masalah konversi data, perubahan program, sistem operasi dan perubahan dokumentasi.
- *Coupling*. Usaha untuk berinteraksi dari suatu sistem aplikasi ke aplikasi lain dalam lingkungan pemrosesan, baik dalam hal menerima atau mengirim data.
- *Performansi*. Jumlah sumberdaya dan kode yang dibutuhkan oleh sistem untuk melakukan fungsinya.
- *Kemudahan dalam pengoperasian*. Usaha yang dihasilkan untuk mengintegrasikan sistem ke lingkungan yang operasional. Prosedur yang digunakan dapat bersifat manual atau otomatis.

Tugas :

Buatlah proses pengujian sistem pada suatu kasus yang anda ketahui.

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