

Standard Testing

24 Juli 2002

1

Kualitas Sistem

1.1 Model pendekatan kualitas

Beberapa model pendekatan kualitas pada umumnya secara definisi dapat dikelompokkan menjadi:

- **Pendekatan kualitas dengan pendekatan transendensi**
Pendekatan ini dikenal dan merupakan sinonim untuk sebuah konsistensi standar tinggi untuk fungsionalitas dari satu produk. Dalam konteks ini maka kualitas tidak dapat diukur dan didefinisikan secara tepat. Selain itu ada satu postulat yang mengatakan bahwa kualitas tidak dapat dievaluasi hanya berdasarkan pada pengalaman saja. Konsep kualitas tidak dapat didefinisikan dengan implikasi, hanya sebagai konsep “keindahan” yang tidak dapat didefinisikan dari implikasi.
- **Pendekatan yang berhubungan dengan produksi**
Pada pendekatan ini maka kualitas dapat diukur dengan presisi. Pada pendekatan ini maka dapat dimungkinkan membuat ranking dari produk berbeda dari kategori yang sama
- **Pendekatan yang berhubungan dengan pemakai**
Pada pendekatan ini maka kualitas lebih ditentukan oleh pemakai produk dibandingkan daripada produk itu sendiri. Menurut pendekatan ini maka produk yang memiliki kualitas baik adalah produk yang dapat memuaskan keinginan pemakai secara baik.
- **Pendekatan proses**
Pendekatan ini berhubungan dengan reanilitas untuk melaksanakan spesifikasi yang telah ditetapkan dan menjaga kondisi ideal dari fabrikasi. Pendekatan ini banyak digunakan pada model industri yang ada pada saat ini. Pengecekan kualitas secara presisi ditujukan untuk mereduksi biaya kesalahan dan revisi.
- **Pendekatan rasio hubungan harga/kemampuan pemakaian (Leistungsfähigkeit)**
Pada pendekatan ini melibatkan hubungan antara harga dan kualitas. Pada konteks ini, maka kualitas dari satu produk adalah produk yang dirancang untuk pemakaian tertentu dengan harga yang dapat terjangkau oleh pemakai atau sebagai konfirmasi antara spesifikasi dan biaya.

1.2 Kriteria Kualitas Perangkat Lunak

Kualitas Perangkat Lunak adalah tingkat kemampuan perangkat lunak memproses atribut dari kombinasi yang diinginkan. Atribut masing masing didefinisikan lebih lanjut. Faktor penentu kualitas adalah sebagai berikut:

- **Efisiensi pemanfaatan sumber daya komputasi yang ada**
Kemampuan atribut yang berhubungan dengan tingkat performance dari Sumber Daya yang digunakan dalam kondisi tertentu.
- **Efisiensi Waktu**
Kemampuan perangkat lunak untuk memproses satu perintah fungsi tertentu atau melaksanakan fungsi tertentu pada rentang waktu tertentu
- **Efisiensi Sumber Daya**
Kemampuan perangkat lunak untuk melakukan fungsi tertentu pada kondisi tertentu atau hasil kondisi tertentu terhadap penggunaan sejumlah sumber daya.
- **Fungsionalitas**
Atribut yang menggambarkan keberadaan tertentu dari properti properti dan fungsi fungsi tertentu yang memberikan kondisi kepuasan atau keinginan pemakai.
- **Kelengkapan**
Tingkat kualitas dimana perangkat lunak memproses fungsi yang diperlukan untuk memenuhi keinginan pemakai
 - **Kebenaran**
Tingkat dimana semua fungsi memiliki spesifikasi sendiri
 - **Sekuriti**
Tingkat kualitas dimana perangkat lunak dapat mendeteksi dan mencegah kelemahan, kehilangan ilegal informasi dan kerusakan data sumber daya,
 - **Kompatibilitas**
Tingkat kualitas dimana perangkat lunak baru dapat diinstal tanpa perubahan kondisi dan lingkungan kerja dengan tersedianya kemungkinan peyiapan dan penggantian perangkat lunak.
 - **Interopalibitas**
Tingkat kualitas dimana perangkat lunak dapat berkomunikasi dengan mudah dengan sistem lain dan diporeasikan dengan mudah
- **Pemeliharaan.**
Satu atribut yang menggambarkan kemampuan yang diperlukan untuk modifikasi khusus.
- **Perbaikan**
Tingkat kualitas yang berhubungan dengan kemampuan untuk membenahi kesalahan atau error dan pembenahan dapat ditangani sesuai dengan keinginan pemakai.
 - **Pengembangan atau Kemampuan untuk Ekspansi**
Tingkat kualitas dari usaha yang diperlukan untuk meningkatkan atau memodifikasi dari efisiensi kerja atau fungsi fungsi perangkat lunak.

- **Pengujian**
Tingkat kualitas pengujian perangkat lunak seperti yang diperlukan untuk melakukan pengujian perangkat lunak
- **Portabilitas**
Atribut kualitas yang berhubungan dengan kemampuan perangkat lunak untuk ditransfer dari lingkungan tertentu ke lingkungan yang lain
- **Ketergantungan pada perangkat keras**
Tingkat kualitas yang menunjukkan ketergantungan perangkat lunak pada perangkat keras tertentu. Seharusnya perangkat lunak tidak tergantung dari perangkat keras
 - **Ketidaktergantungan pada perangkat lunak bantuan**
Tingkat kualitas perangkat lunak akan driver pada spesifik perangkat lunak tertentu
 - **Kemampuan Instalasi**
Tingkat usaha untuk mengatur perangkat lunak untuk mencapai kondisi tertentu, pada tingkat mana perangkat lunak dapat digunakan kembali pada aplikasi yang lain daripada hanya untuk satu aplikasi original.
- **Reliabilitas**
Atribut yang menggambarkan kemampuan perangkat lunak untuk mempertahankan tingkat performance level nya pada kondisi tertentu dan rentang waktu tertentu
 - **Ketiadaan defisiensi**
Tingkat kualitas dimana perangkat lunak tidak terdapat atau ditemukan eror atau kesalahan yang tak terdeteksi.
 - **Toleransi Kesalahan atau Error**
Tingkat kualitas yang menggambarkan batasan toleransi dimana perangkat lunak akan terus berfungsi tanpa adanya kesalahan yang menyebabkan kerusakan langsung dan tidak langsung pada data pemakai. Dalam konteks ini termasuk di dalamnya adalah tingkat kemampuan beroperasinya fungsi yang ada untuk recovery dan beroperasi kembali
 - **Ketersediaan**
Tingkat kualitas yang menggambarkan kemampuan perangkat lunak akan masih tetap beroperasi pada saat adanya kesalahan sistem
- **Usabilitas**
Atribut yang menggambarkan usaha untuk menggunakan (termasuk persiapan penggunaan dan evaluasi hasil) dan menggambarkan tugas dari kegunaan fungsi dari pemakai
- **Kepahaman**
Jumlah pemakai yang memerlukan usaha ekstra untuk bisa mengerti dan memahami perangkat lunak
 - **Kemudahan untuk belajar**
Tingkat kualitas yang menggambarkan usaha yang diperlukan oleh pemakai untuk mengerti perangkat lunak. Kondisi ideal adalah usaha yang dilakukan seminimal mungkin.

– **Operabilitas**

Tingkat kualitas yang menggambarkan kesesuaian perangkat lunak pada tujuan dan sasaran pemakaian serta menggambarkan karakteristik psikologis pemakai termasuk didalamnya faktor ergonomis seperti warna, ukuran, suara dan lain sebagainya.

– **Kemampuan berkomunikasi**

Tingkat kualitas yang menggambarkan bahwa perangkat lunak dirancang sesuai dengan karakteristik psikologis dari pemakai.

2

Pengukuran Kinerja

Untuk memahami dan menginterpretasikan pengujian kinerja tersebut berikut ini dijabarkan beberapa dasar teoritis yang melatar-belakangi suatu pengukuran kinerja sistem komputer.

Kinerja pada suatu sistem komputer didefinisikan dengan :

“is the degree to which a computing system meets the expectation of the person involved with it “ (Doherty, W.J.(1970), Scheduling TSS/360 for Responsiveness, AFIPS Proc. FJCC, 1970, pp. 97-111)

Definisi kinerja sistem komputer berdasarkan aspek perikayasaan perangkat lunak (Software Engineering):

“is the effectiveness with which the resource of the host computer system are utilized toward meeting the objective of the software system “ (Graham, R, M. (1973). Performance prediction, Advances Courses on Software Engineering, Springer Verlag, 1974, pp.395-463.)

Sehingga dapat disimpulkan bahwa kinerja suatu sistem komputer adalah : *“seberapa baik suatu sistem memungkinkan pengguna melakukan yang diinginkannya”*. Dan dapat diinterpretasikan pula *“seberapa baik sistem bekerja sesuai dengan yang direncanakan”*.

Pengukuran kinerja sistem dibutuhkan untuk beberapa evaluasi :

- **Comparative Evaluation** . Kinerja suatu sistem dievaluasi relatif kepada sistem lainnya. Kegunaan dari evaluasi ini misal untuk proses pembelian perangkat lunak baru, atau perangkat keras baru, memilih service komputasi, dan juga mengevaluasi perubahan sistem untuk modifikasi.
- **Analytic Evaluation** . Kinerja dari sistem komputer dievaluasi berdasarkan beberapa parameter sistem. Tujuan dari evaluasi ini adalah untuk meningkatkan kinerja sistem (performance tuning), melakukan perawatan sistem (performance control), dan mendisain serta mengimplementasi sistem baru.

Kinerja suatu sistem sendiri dapat mengalami degradasi karena beberapa hal, misal :

- Kecepatan atau kapasitas dari beberapa komponen sistem menyebabkan komponen sistem lainnya tak dapat bekerja dengan kecepatan maksimum.

- Interferensi yang disebabkan oleh permintaan secara simultan dari 2 atau lebih dari komponen tersebut untuk saling berkomunikasi ketika permintaan tersebut dapat diproses secara sekuensial.
- Karakteristik dari beban kerja (workload) sistem.

Pengukuran kinerja hanya dapat dispesifikasikan kepada jenis dan tujuan dari sistem yang dievaluasi, workload, dan fungsi dari evaluasi itu. Pengukuran kinerja ini harus terdefinisi dengan baik dan jelas, karena merupakan “framework” bagi seluruh proses evaluasi. Serta dapat melakukan reproduksi hasil atau perulangan pengukuran dengan tepat. Seperti diketahui kinerja dari suatu sistem komputer adalah fungsi dari :

- Konfigurasi sistem
- Kebijakan pengelolaan sumber daya (Resource management policies)
- Efisiensi program dari sistem yang digunakan
- Efektifitas instruction set dari processor yang digunakan
- Kecepatan perangkat keras
- Untuk sistem yang terhubung dengan jaringan kinerja subsistem jaringan juga termasuk

Komponen suatu sistem akan mempengaruhi kinerja sistem secara keseluruhan melalui interaksi yang mutual antar masing komponen. Sehingga kinerja bisa didiskusikan dalam posisi yang berbeda :

- **Effectiveness** dari sistem ketika menangani aplikasi tertentu. Hal inilah yang terlihat oleh pengguna sistem. Untuk tujuan ini digunakan pengukuran dengan pendekatan stimulus (**Stimulus approach**). Pada pendekatan ini sistem dianggap suatu kotak hitam (black box) yang memiliki fungsi tertentu yang diketahui. Pengukuran dilakukan dengan mengobeservasi sistem berdasarkan suatu benchmark atau suatu simulator. Metoda ini memang cepat tapi kurang detail memberi gambaran akan kinerja sistem.
- **Internal efficiency**, yaitu melakukan pengamatan terhadap mekanisme sistem mengatur sumber daya yang dimilikinya untuk menangani beban kerja. Untuk tujuan ini dilakukan pengukuran dengan pendekatan analitis (**Analytic approach**). Pengukuran ini berusaha mengukur perilaku internal dari sistem. Dengan tujuan untuk menguji apakah sistem bekerja dengan benar, mengisolasi kemungkinan kesalahan, serta memahami sistem dan lingkungan kerjanya.

Pada dasarnya pada kedua pendekatan di atas memiliki parameter yang berbeda untuk menjabarkan kinerja sistem. Pengukuran yang dilakukan oleh Mindcraft pada dasarnya adalah pengujian “efektifitas” sistem maka berikut ini akan dijabarkan parameter yang biasanya digunakan pada suatu pengukuran sistem dengan pendekatan tersebut (Wiryana, 1998) :

- **Throughput**. Jumlah kerja yang diselesaikan untuk satu satuan waktu pada beban kerja yang diberikan.
- **Relative throughput**. Waktu yang dibutuhkan untuk memproses beban kerja pada sistem 1 relatif ke waktu yang dibutuhkan pada sistem 2 untuk beban yang sama.

- **Kapabilitas (Kapasitas)** . Jumlah kerja maksimal yang dapat ditangani pada satu satuan waktu untuk beban kerja yang diberikan.
- **Turnaround time** . Waktu antara suatu pekerjaan diberikan pada suatu sistem dan hasil yang diterima
- **Response time** . Waktu untuk suatu transaksi pada sistem interaktif atau sistem sesungguhnya.
- **Availabilitas** , prosentase sistem dapat tersedia dan bekerja melayani pengguna

2.1 Workload

Untuk menguji kinerja sistem komputer biasanya digunakan suatu “workload buatan”. Workload sesungguhnya biasanya tidak dapat direproduksi. Tetapi bila sifat statistik dari workload sistem tidak berubah maka secara statistik dapat dikatakan workload dapat direproduksi. Sehingga dengan asumsi workload ini memungkinkan untuk :

- Menentukan karakteristik workload dengan distribusi dari permintaan yang dilakukan pada resource sistem.
- Mendefinisikan suatu satuan “kerja” dan menyatakan beban kerja dengan “satuan kerja dasar” ini.

Untuk menentukan workload yang digunakan maka perlu dipertimbangkan beberapa hal :

- Representativeness. Apakah suatu model beban kerja dapat mewakili kondisi sesungguhnya.
- Fleksibilitas. Apakah beban kerja tersebut dapat diubah.-ubah dengan mudah atau hanya memiliki satu model yang tetap.
- Kesederhanaan penyusunan.
- Compactness. Apakah model beban kerja tersebut merupakan satu kesatuan sistem pengukuran atau menggunakan berbagai sistem.
- Biaya untuk menggunakan model tersebut.
- Ketergantungan platform. Apakah model tersebut dapat diterapkan pada platform yang berbeda atau hanya berlaku untuk satu jenis platform
- Reprodusibilitas. Apakah model beban kerja tersebut dapat diulang untuk melakukan pengukuran yang sejenis. Ini bertujuan untuk dapat melakukan validasi terhadap pengukuran kinerja.
- Kompatibilitas. Tidak saja terhadap sistem perangkat keras tetapi juga terhadap perangkat lunak yang digunakannya.

Kinerja suatu sistem komputer adalah suatu reaksi untuk suatu workload (beban kerja) yang spesifik. Sehingga untuk mengevaluasi kinerja suatu sistem komputer workload yang tepat haruslah dipilih. Karakter workload haruslah mampu mendemonstrasikan faktor-faktor yang penting. Workload ini bekerja sebagai suatu model bagi sistem sesungguhnya. Fungsi dari penggunaan model workload :

- Menyediakan suatu representasi dari workload untuk kebutuhan evaluasi kinerja sebagai perbandingan pada sistem yang berbeda.
- Menyediakan suatu lingkungan terkendali yang dapat mereproduksi hasil percobaan pengukuran kinerja untuk tujuan studi optimasi.
- Mengurangi kuantitas data yang harus dianalisis.
- Memberikan workload pada sistem pada bentuk yang dibutuhkan oleh model sistem.

Satu-satunya benchmark yang valid adalah benchmark yang dilakukan sendiri, dengan menggunakan aplikasi yang akan digunakan pada lingkungan komputasi produksi atau lingkungan pemakaian sesungguhnya, dengan menggunakan jaringan sesungguhnya. Benchmark yang dilakukan di tempat lain sebetulnya tidak menggambarkan hal yang sama.

3

Pengujian usabilitas

3.1 Usabilitas

Pada bahasan User Interface yang lebih diutamakan adalah **usability** (usabilitas) sistem. Usabilitas dari suatu sistem ditentukan dari bagaimana kemudahan user memenuhi tugasnya. Definisi usability menurut Shackel(1993) adalah :

The capability of human functional terms to be used easily and effectively by the specified range of users, given specific training and user support, to fulfil the specified range of tasks, within the specified range of environment scenario

Memang definisi tersebut masih kental dari sudut pandang **anthropometric** ataupun **biomekanis** . Artinya kenyamanan fisik lebih dititik beratkan sehingga Chapmanis (1993) mendefinisikan usability lebih jauh sebagai :

The usability of computer is measured by how easily and effectively the computer can be used by a specific set of users, given particular kinds of supports, to carry out

Dari kedua definisi tersebut jelas suatu pengukuran usabilitas tidak dilakukan ketika user baru memakai sistem. Tetapi dilakukan setelah user melalui masa pelatihan dan mendapat dukungan yang cukup (manual dan sebagainya) Pada sudut pandangan teknologi informasi usability dapat diformulasikan sebagai berikut (Sweeney et al., 1993).

Usability is emergent quality of an optimum design which is reflected in the effective and satisfying use of the IT

Berdasarkan kriteria tersebut usability dapat diukur dengan mempertimbangkan beberapa hal :

- **Optimum** . Terkait dengan analisis cost-benefit, dan kompromi.
- **Efektif** . Berkaitan dengan efisiensi dan produktifitas penggunaan, di antaranya kecepatan, diselesaikannya tugas, betul tidaknya tugas diselesaikan, yang dicapai oleh interaksi pengguna dengan sistem. Hal ini sangat kontekstual sifatnya.
- **Kepuasan** . Ini berkaitan dengan emosi kepuasan pengguna terhadap produk yang dipakai. Pengukuran tidak hanya berlangsung pada awal penggunaan tetapi setelah melalui pelatihan dan penggunaan pada waktu tertentu.

- **User, task and environment** . Karakteristik ini harus juga dipertimbangkan.
- **Biaya** . Hal ini berkaitan dengan investasi yang harus dilakukan pengguna untuk mencapai usabilitas pada level yang diinginkan, termasuk biaya fisik, dan non fisik.

Usability dapat dipandang dari tiga faktor utama :

- **Learnability** , yaitu kemudahan pengguna baru untuk mulai secara efektif berinteraksi dan mencapai kinerja yang optimal.
- **Fleksibilitas** , yaitu beragam cara yang disediakan oleh sistem untuk memungkinkan user dan sistem bertukar informasi. Sehingga fleksibilitas ini harus menjamin bahwa si user tidak hanya **terpaku pada satu pola dialog** .
- **Robustness** , tingkatan berbagai dukungan yang disediakan bagi user untuk menentukan tercapainya tujuan. Artinya bila suatu cara gagal maka masih ada cara lain untuk mencapai tujuan tersebut.

Satu persatu akan dibahas prinsip dasar dari penilaian suatu user interface. Pertama adalah prinsip '**learnability**' yang sering disalah artikan sebagai **user friendly** . Learnability sendiri mempertimbangkan bagaimana suatu user interface memungkinkan pengguna baru pertama kali menggunakan sistem tersebut dapat segera menggunakan sistem tersebut. Prinsip ini sendiri memiliki prinsip-prinsip :

- **Predictability** . User dapat menentukan akibat kemudian atas aksi yang dilakukan berdasarkan interaksi yang pernah dilakukan sebelumnya. Berdasarkan sejarah apa yang dilakukan user dapat memperkirakan apa yang akan terjadi selanjutnya ketika suatu aksi dilakukan. Hal ini terkait dengan kemampuan pengguna menentukan efek dari operasi pada sistem. Prinsip ini terkait dengan **Operation Visibility** yaitu kemampuan pengguna untuk mengetahui operasi yang dapat dilakukan
- **Synthesizability** . Memungkinkan user memperkirakan langkah yang sebelumnya dilakukan berdasarkan kondisi saat ini. Ini berarti user telah membentuk suatu mental model bagaimana sistem bekerja. Prinsip ini berkaitan dengan **Immediate/Eventual honesty** .
- **Familiarity** . Berdasarkan pemahaman user atas pengetahuan dan pengalaman yang dimilikinya baik pada dunia nyata maupun pada komputer dapat dimanfaatkan untuk berinteraksi dengan sistem yang baru. Yang perlu diperhatikan adalah, setiap kultur memiliki representasi simbol yang berbeda, sehingga sistem yang memiliki 'keluwesan' tinggi untuk membentuk metaphora akan menjamin terbentuknya familiarity ini. Terkait dengan **guessability** yaitu kemampuan pengguna menebak fungsi suatu obyek, dan **Affordance** , yaitu kemampuan pengguna memperkirakan operasi yang dapat dilakukan oleh obyek tersebut.
- **Generalizability** . Sistem memungkinkan pengguna mempraktekan pengetahuan umumnya untuk interaksi khusus berdasarkan aplikasi yang berbeda ketika berada pada situasi yang serupa.
- **Consistency** . Kemiripan perilaku pada situasi serupa pada obyek task yang serupa.

Sedangkan faktor **flexibility** terdiri dari

- **Dialogue initiative** . Memberikan kebebasan pada user akan terbatasnya dialog masukan yang disediakan oleh sistem. Sehingga user dapat memilih untuk melakukan operasi apakah dengan dialog dengan metaphora click atau dengan menggunakan command line. Berkaitan dengan prinsip **System/User pre-emptiveness** , artinya apakah user dapat memulai interaksi atau hanya menunggu awal interaksi dilakukan oleh sistem. Secara umum biasanya diinginkan bahwa pengguna lebih menentukan saat dimulainya interaksi.
- **Multi-threading** . Kemampuan sistem untuk mendukung interaksi pengguna melakukan task lebih dari satu pada saat yang sama. Sehingga sistem tidak diam setelah user memberikan perintah. Berkaitan dengan prinsip **Concurrent vs interleaving** , concurrent memungkinkan komunikasi simultan untuk task yang berbeda, sedangkan interleaving memungkinkan pelaksanaan task yang berbeda dan pada saat tertentu dapat dilakukan bersamaan, **multimodality** berkaitan dengan jumlah kanal untuk melakukan interaksi lebih dari satu.
- **Task migrability** . Kemampuan memberikan kendali untuk mengeksekusi tugas tertentu sehingga menjadi pertukaran kendali antara user dan sistem dapat berlangsung dengan baik. Pengguna dengan mudah menghentikan suatu proses, meneruskan kembali proses tersebut, mengatur proses berjalan di background atau di depan, tanpa harus menunggu sistem menyelesaikan tugasnya.
- **Substitutively** . Memungkinkan pemberian nilai masukan yang ekuivalen. Misal entah menggunakan pemilihan list file, atau memasukkan nama file. Begitu juga sebaliknya. Sehingga di sinilah berperannya penilaian atas kemampuan user interface menangani **regular expression** . Berkaitan dengan prinsip **Representation multiplicativy, equal opportunity**
- **Customizability** , kemudahan user interface dimodifikasi oleh pengguna atau oleh sistem. Sehingga user tidak terbatas dengan menggunakan metaphora, ataupun mekanisme tunggal. Sistem harus secara fleksibel dapat dengan mudah dikustomisasi. Prinsip ini berkaitan dengan prinsip **adaptitivty** adalah proses pengubahan yang iniatif dilakukan oleh pengguna dan **adaptability** adalah proses pengubahan yang intiiatif dilakukan oleh sistem.

Sedangkan aspek **robustness** terdiri dari prinsip :

- **Observability** . Kemungkinan pengguna mengevaluasi kondisi internal sistem dari representasi yang diterima saat ini. Sehingga user tidak 'buta' apa yang terjadi pada sistem. Hal ini berkaitan dengan prinsip **Browsability** yaitu kemampuan pengguna mengeksplorasi kondisi internal dari sistem berdasarkan interpretasi di user interface, **static/dynamic defaults** pengisian nilai default ini akan menolong user untuk memberikan masukan, **reachability** berkaitan dengan kemungkinan bernavigasi diantara keadaan sistem yang dapat diobservasi, **persistence** tenggang waktu pengaruh dari suatu aksi komunikasi yang dilakukan oleh user, **operation visibility** yaitu kemungkinan pengguna mengetahui operasi yang sedang dilakukan oleh sistem.
- **Recoverability** , kemampuan pengguna untuk memperbaiki aksi yang dilakukan ketika diketahui terjadi suatu kesalahan. Terkait dengan prinsip **reachability** , **forward/backward recovery** , proses perbaikan kesalahan ini dapat dilakukan inisiatif oleh user ataupun sistem baik pada arah maju ataupun mundur, **commensurate effort** , jika sangat sulit untuk melakukan 'undo' pada suatu pengaruh pada suatu state maka akan sulit pula melakukan

akis tersebut. Sebaliknya bila mudah melakukan 'undo' maka harus mudah pula mengulanginya (redo).

- **Responsiveness** , bagaimana user menerima laju komunikasi dengan sistem. Sehingga ketika user berinteraksi tidak terlalu lama menunggu. Terkait dengan prinsip **stability** yang berkaitan dengan variasi tengang waktu sistem bereaksi haruslah sekecil mungkin..
- **Task conformance** . Suatu sistem interaktif memiliki tujuan agar pengguna dapat melakukan tugasnya untuk mencapai tujuan yang diharapkan pada suatu domain. Sehingga perlu dipertimbangkan apakah sistem mendukung penyelesaian tugas pada domain tersebut, atau apakah sistem mendukung hal tersebut sesuai dengan yang diinginkan oleh pengguna. **Task completeness** berkaitan tentang cakupan permasalahan tugas dan **task adequacy** berkaitan dengan pemahaman pengguna terhadap tugas yang harus dilakukannya.

Sehingga menjadi wajar untuk mempertanyakan pernyataan **userfriendly yang sudah mendarah daging** , berarti hal yang dibicarakan adalah tingkat **familiarity** dari pengguna terhadap sistem yang lama, bukan usability sistem.

Seringkali orang secara cepat dan gampang melakukan **penilaian atas user friendly** Tidak bisa dengan langsung melihat bisa dilakukan penilaian sekilas. Jelas terjadi pemahaman yang salah akan istilah **user friendly** dengan **familiarity** yang mendorong ke arah **easy to learn** (sengaja digunakan terminologi asli yang digunakan dalam bahasa ilmiah pada bidang user interface).

Belum tentu suatu aplikasi yang easy to learn adalah user friendly, begitu juga belum tentu suatu aplikasi yang user friendly akan easy to learn. Dua hal ini harus dibedakan, yang penting adalah '**usabilitas**' dari aplikasi. Saya kutipkan lagi dari buku User Interface Design, suatu disain sistem user interface yang baik haruslah :

- Spesifik untuk tugas tertentu
- Harus memiliki kinerja yang terprediksi
- Proses disain haruslah bersifat iteratif
- Disain harus memiliki kemampuan mengendalikan daripada melakukan evaluasi
- Disain harus **sederhana**

Dari persyaratan di atas jelas bahwa fleksibilitas untuk melakukan kustomisasi adalah berperah penting untuk menjaga disain yang baik. User interface yang memberikan harga mati sangat sulit untuk dikembangkan agar sesuai dengan tugas tertentu dan user tertentu.

Dalam menilai suatu user interface apakah buruk atau baik, tidak bisa dilakukan secara serampangan **hanya dengan cara mencap kurang user friendly** . Metoda evaluasi telah tersedia, misal salah satunya telah ada metoda yang secara standard dapat dijadikan acuan, misal **ISO 9241** .

Usability Objective	Effectiveness measures	Efficiency measures	Satisfaction Measures
Suitability for the task	Percentage of goals achieved	Time to complete a task	Rating scale for satisfaction
Appropriate for trained users	Number of power features used	Relative efficiency compared with an expert user	Rating scale for satisfaction with power features
Learnability	Percentage of functions learned	Time to learn criterion	Rating scale for ease of learning
Error tolerance	Percentage of errors corrected successfully	Time spent on correcting errors	Rating scale for error handling

Dari standard tersebut tampak bahwa untuk menilai suatu user interface learnability yang dipengaruhi familiarity dari '*mental model*' dan '*conceptual model*' hanyalah merupakan salah satu komponen. Sebagaimana digambarkan pada gambar di atas. Ketika si disainer merancang sistem. Maka dia mengira-ngira bekerjanya sistem ini yang disebut dengan '*conceptual model*'. Begitu juga si disainer juga mengira-ngira bagaimana kerja si pengguna ini yang dikenal dengan istilah '*cognitive model*'. Ketika sistem dijalankan, maka dalam benak si pengguna dia memiliki bayangan harus bagaimana sistem bekerja ini yang disebut '*mental model*', begitu juga di mesin dieksekusi suatu model (secara abstraksi) mengenai bagaimana perilaku dari si pengguna ini yang disebut '*system model of operator*'.

Obyek dari user interface akan berusaha membuat agar antara '*system model of operator*' cocok dengan '*cognitive model*', dan '*mental model*' yang terbentuk di operator sesuai dengan '*conceptual model*' dari disainer. Bila kondisi ini tercapai maka user interface akan memberikan hasil yang baik.

3.2 Metoda penguujian usabilitas

Untuk dapat menilai suatu user interface (apalagi sistem Window) tidak bisa sekedar melihat dan menggunakan dalam beberapa menit. Beberapa metoda evaluasi haruslah diterapkan, misal pada saat **disain** dilakukan metoda evaluasi seperti berikut :

- **Cognitive walkthrough** . Metoda ini diilhami oleh teori psikologi, evaluasi cara ini dilakukan secara informal dan bersifat subyektir. Dilakukan dengan cara menjelajahi penggunaan sistem. Tujuannya untuk mengevaluasi sistem terutama ketika pengguna belajar melakukan tugas tertentu. Biasanya evaluasi ini bertujuan untuk mengetahui :
 - Apakah akibat interaksi tersebut pada user
 - Proses kognitif apakah yang terjadi
 - Permasalahan 'learning' apakah yang timbul
- **Heuristic evaluation** . Metoda ini hampir mirip dengan cognitive walkthrough akan tetapi melibatkan seorang pakar yang menilai disain berdasarkan kriteria usabilitas. Kriteria ini misalnya :
 - Perilaku sistem dapat diprediksikan

- Perilaku sistem konsisten
 - Umpan balik disediakan oleh sistem
 - Memori pengguna tidak terlalu dibebani
 - Dialog bersifat task-oriented
- **Review based evaluation** . Pada evaluasi ini seorang reviewer akan menguji dan melakukan eksperimen. Evaluasi akan mempertimbangkan perbedaan antara hasil eksperimen dengan disain yang ada.
 - **Model based evaluation** . Pada metoda ini dilakukan evaluasi berdasarkan kriteria model dan tiap bagian disain.

Untuk mengevaluasi sistem yang **telah diimplementasikan** dapat dimanfaatkan beberapa metoda misal

- **Empirical method** : dengan evaluasi melalui eksperimen. Memanfaatkan alat bantu statistik, untuk menghitung performance pengguna pada sistem. Jumlah dan jenis percobaan haruslah mewakili keadaan sesungguhnya
- **Observational techniques** . evaluasi ini bisa dilakukan dengan berbagai teknik misal :
 - **Think aloud** . Pada teknik evaluasi ini, pengguna sebenarnya pada suatu sistem diobservasi ketika berinteraksi dengan sistem. Biasanya mereka diminta melakukan suatu task, dan ketika melakukan hal itu mereka akan 'meneriakan' apa yang ada dipikrannya, pertimbangannya, dan keputusan aksi yang dilakukannya. Cara ini cukup menarik karena :
 - * Proses ini termasuk mudah dan dapat digunakan
 - * Pengguna didorong untuk memberikan kritik
 - * Evaluator dapat menemukan letak interaksi yang membingungkan dan dapat meningkatkan efektifitas sistem dengan cara menentukan permasalahannya
 - **Protocol analysis** . Dengan cara ini dilakukan pencatatan aksi yang dilakukan user. Dapat digunakan beragam metoda dan alat bantu :
 - * Kertas dan pensil
 - * Audio recording
 - * Video recording
 - * Computer logging
 - * User notebooks
 - **Automatic protocol analysis** . Dengan perangkat bantu analisis ini maka dapat dilakukan analisis interaksi user interface. Misal beberapa tool :
 - * Experimental Video Annotator (EVA)
 - * Workplace project dari Xerox PARC
 - **Post-task walkthrough** . Kadang kadang data evaluasi dari observasi langsung pada suatu saat tertentu tak dapat diinterpretasikan. Untuk mengetahui alasan pelaksanaan aksi maka perlu dilakukan evaluasi cara ini. Sebab kadang pengguna tak dapat 'bercerita' ketika sedang melakukan suatu aksi, terutama ketika saat kritis dan sebagainya.

- **Query technique** . Metoda yang tidak terlalu formal ini dilakukan dengan bertanya pada user. Biasanya merupakan evaluasi pelengkap. Bisa dilakukan dengan :
 - **Interview** . Biasanya pengguna akan ditanya, kesan-kesan menggunakan sistem ketika melakukan sesuatu tugas. Interview akan dilakukan baik bila telah dilakukan perencanaan di awal. Interview ini akan merupakan evaluasi 'high-level' yang akan mencoba menangkap, preferensi user, impresi dan perilaku.
 - **Questioner** . Cara ini memang kurang fleksibel dibanding interview. Tetapi lebih mudah dilakukan dan memakan waktu lebih singkat. Biasanya dapat dilakukan variasi pertanyaan antara lain :
 - * Pertanyaan bersifat umum. Untuk memahami latar belakang pengguna.
 - * Pertanyaan dengan jawaban bebas (open ended). Untuk meminta saran dari user akan sistem.
 - * Skalar. Hal ini bertujuan agar pengguna menilai dengan menggunakan skala numerik.
 - * Pilihan ganda. Biasanya untuk menentukan hal manakah yang lebih disukai oleh user
 - * Rangkaian. Biasanya untuk mengetahui preferensi pengguna.