

# DESAIN

## 5.1 Tujuan Instruksional Umum

Mahasiswa dapat menguasai konsep-konsep interaksi manusia dan komputer dengan baik, sehingga dapat mengimplementasikannya dalam mendesain *software* sesuai dengan prinsip-prinsip *User Centered Design*.

## 5.2 Tujuan Instruksional Khusus

Mahasiswa dapat mengerti dan menjelaskan prinsip desain, cara mendapatkan ide , tantangan dalam membuat design grafik yang baik.

## 5.3 Komponen Antarmuka

Antarmuka pengguna memiliki empat komponen yaitu :

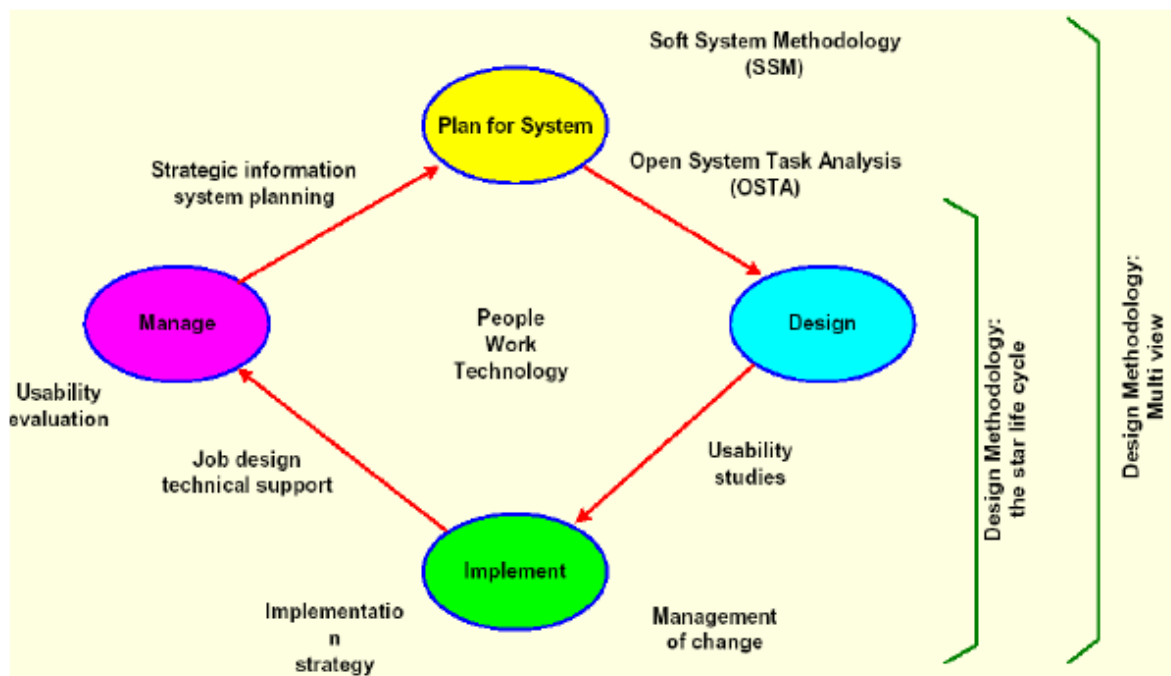
1. Model pengguna, model konseptual yang diinginkan pengguna dalam memanipulasi informasi dan proses yang diaplikasikan pada informasi tersebut.
2. Bahasa perintah (*command language*), peranti untuk memanipulasi model, idealnya dgn menggunakan bahasa alami.
3. Umpan balik, untuk memberi keyakinan bahwa program telah menerima perintah pengguna dan dapat memahami maksud perintah tersebut termasuk kemampuan sebuah program yang membantu pengguna untuk mengoperasikan program itu sendiri.
4. Tampilan informasi, digunakan untuk menunjukkan status informasi atau program ketika pengguna melakukan suatu tindakan.

## 5.4 Prinsip Design

Desain antarmuka yang baik berdasarkan pada user yang dinamakan User-Centered Design (UCD). UCD adalah filosofi perancangan yang menempatkan pengguna sebagai pusat dari proses pengembangan sistem. Saat ini pendekatan UCD telah didukung berbagai teknik, metode, tools, prosedur, dan proses yang membantu perancangan sistem interaktif yang lebih berpusat pada pengguna. Pengguna tidak hanya memberi komentar tentang ide perancangan. Mereka harus secara intensif dilibatkan dalam semua aspek, termasuk

bagaimana implementasi sistem yang baru akan mempengaruhi pekerjaannya. Pengguna juga dilibatkan dalam pengujian awal, evaluasi serta perancangan secara iteratif. Eason (1992) menggambarkan empat langkah kunci dalam pengembangan yaitu :

- Perencanaan
- Perancangan
- Implementasi
- Pengelolaan sistem.



Gambar 5.1 Metode UCD

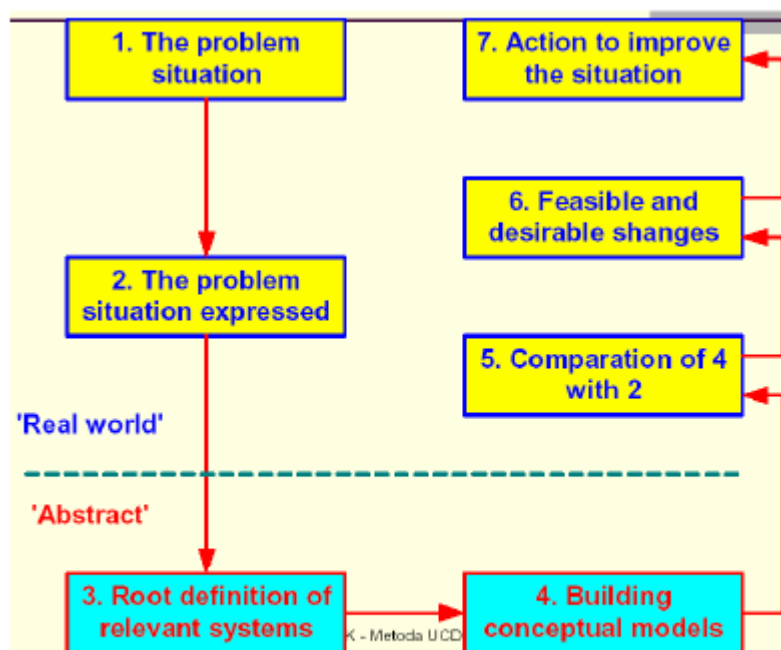
Pada gambar di atas terdapat empat pendekatan dalam pengembangan sistem yaitu:

- Soft System Methodology (SSM), berfokus pada perencanaan.

Penekanan SSM tidak pada pencarian solusi untuk suatu masalah, tetapi pada pemahaman situasi dimana masalah yang dirasakan dianggap bukan merupakan esensi/akar masalah sebenarnya. Langkah-langkah SSM yaitu :

- Langkah 1 dan 2 difokuskan pada pencarian pernyataan yang lengkap atas situasi permasalahan.

- Pada langkah ini dilakukan pertemuan yang melibatkan seluruh pihak yang berkepentingan (*stakeholders*). Umumnya *stakeholders* yang berbeda akan mempunyai pandangan yang berbeda pula tentang tujuan sistem.
- Perbedaan pandangan ini tidak perlu dipertentangkan, karena mereka menekankan pada aspek yang berbeda atas situasi keseluruhan dan mereka harus difasilitasi.
- Langkah 3 mencoba untuk membuat definisi sistem yang presisi.
- Langkah 4 menggunakan hasil dari langkah 3 untuk membuat model konseptual yang berupa pernyataan sistem secara abstrak.
- Membawa langkah 3 dan 4 dari dunia nyata yang memungkinkan untuk menghasilkan suatu pernyataan yang tidak dipengaruhi oleh batasan-batasan dalam dunia nyata.
- Checkland (1981) dan Wilson (1984) menyediakan sejumlah metoda yang dapat digunakan untuk menghasilkan suatu definisi yang formal dan komprehensif tentang sistem.
- Checkland menamakannya sebagai *root definition*, yang dapat membantu perancang untuk memastikan bahwa mereka telah mencakup seluruh aspek dari sistem dan menghasilkan definisi akar yang kuat.
- Definisi tersebut adalah elemen CATWOE: *Clie*n atau pelanggan, *Actors*, *Transformation*, *Weltanschauung* (pandangan dunia), *Owners*, dan *Environment*.



Gambar 5.2 Soft System Methodology (SSM)

- Open System Task Analysis (OSTA), berfokus pada langkah awal perencanaan.
- Multiview, metodologi yang lengkap mulai dari perencanaan sampai dengan implementasi.
- Star Life Cycle, berfokus pada perancangan

Konsep UCD (User Centered Design ) yaitu :

- Berpusat pada user
- Tujuan/sifat-sifat, konteks dan lingkungan produk semua berdasarkan pada pengalaman user.
- Selanjutnya ditetapkan model pekerjaan pengguna yang akan didukung sistem.

Prinsip *user-centered design lifecycle* [J. Gould, 1995] yaitu :

- Pertama-tama fokus pada pengguna (*user requirement*)
  - Perancang harus mempunyai hubungan langsung dengan pengguna sesungguhnya atau calon pengguna – melalui interview, survei, partisipasi dalam workshop perancangan.
  - Tujuan utama untuk memahami kognisi, karakter dan atitud pengguna serta karakteristik antropometri.
  - Aktifitas utama mencakup pengambilan data, analisis dan integrasinya ke dalam informasi perancangan dari pengguna tentang karakteristik tugas, lingkungan teknis dan organisasi.
- Perancangan terintegrasi (*prototyping*)
  - Integratif? Perancangan harus mencakup antarmuka pengguna, sistem bantuan, rencana pelatihan, dokumentasi pengguna, dukungan teknis serta prosedur instalasi dan konfigurasi.
  - Perancangan tersebut harus dikembangkan secara parallel, dan tidak secara sekuensial serta harus berada di bawah satu proses manajemen.
- Dari awal berlanjut pada pengujian pengguna (*usability measurement*)
  - Pengukuran penggunaan produk secara empiris

- Dibutuhkan observasi tentang kelakuan pengguna, evaluasi umpan balik yang cermat, wawasan pemecahan masalah dan motivasi yang kuat untuk mengubah rancangan.
- Umpan balik yang berasal dari pengguna dikumpulkan secara langsung atau tidak langsung dinyatakan dalam bentuk rekomendasi dan keputusan perancangan.
- Perancangan interatif (spiral model) Iteratif?

Sistem yang sedang dikembangkan harus didefinisikan, dirancang, dites berulang kali berdasarkan hasil tes kelakuan dari fungsi, antarmuka, sistem bantuan, dokumentasi pengguna dan pendekatan pelatihannya.

User dari user centered design yaitu :

- Pengguna  
Orang yang akan menggunakan sistem
- Pengguna langsung (*end user*).  
Orang yang menggunakan sistem untuk menyelesaikan pekerjaannya.
- Pengguna tidak langsung.  
Orang yang menggunakan sistem untuk penggunaan lainnya, seperti : system administrators, installers dan demonstrators.
- Stakeholder.  
Orang yang terpengaruh oleh sistem atau yang dapat mempengaruhi proses pengembangan, seperti : staf pemasaran dan pembeli.
- Usability engineers.  
Orang yang mempunyai latar belakang dalam psikologi dan dapat membantu dalam menetapkan panduan perancangan, menentukan konteks penggunaan dan melaksanakan wawancara kebutuhan penggunaan dan sesi pengujian.
- Technical staff dan software developers.  
Orang yang merinci spesifikasi fungsionalitas sistem dan mengembangkan use case model dan prototipe antarmukanya.

Aturan dalam User centered design yaitu :

1. Perspective, pengguna selalu benar.

Jika terdapat masalah dalam penggunaan sistem, maka masalahnya terdapat pada sistem bukan pengguna

2. Instalasi

Pengguna mempunyai hak untuk dapat menginstal atau meng-uninstal perangkat lunak dan perangkat keras sistem secara mudah tanpa adanya konsekuensi negatif.

3. Pemenuhan

Pengguna mempunyai hak untuk mendapatkan sistem dapat bekerja persis seperti yang dijanjikan.

4. Instruksi

Pengguna mempunyai hak untuk dapat menggunakan instruksi secara mudah (buku petunjuk, bantuan secara *online* atau kontekstual, pesan kesalahan). Tujuan dari instruksi tersebut agar pengguna mempunyai pemahaman dalam menggunakan sistem untuk mencapai tujuan yang diinginkan secara efisien dan terhindar dari masalah.

5. Control

Pengguna mempunyai hak untuk mengontrol sistem dan mampu membuat sistem menanggapi dengan benar atas permintaan yang diberikan.

6. Umpan balik

Sistem menyediakan informasi yang jelas, dapat dimengerti dan akurat tentang tugas yang dilakukan dan kemajuan yang dicapai.

7. Keterkaitan

Pengguna mempunyai hak untuk mendapatkan informasi yang jelas tentang semua prasyarat yang dibutuhkan sistem untuk memperoleh hasil terbaik

8. Scope

Pengguna mempunyai hak untuk mengetahui batasan kemampuan sistem

9. Assistance

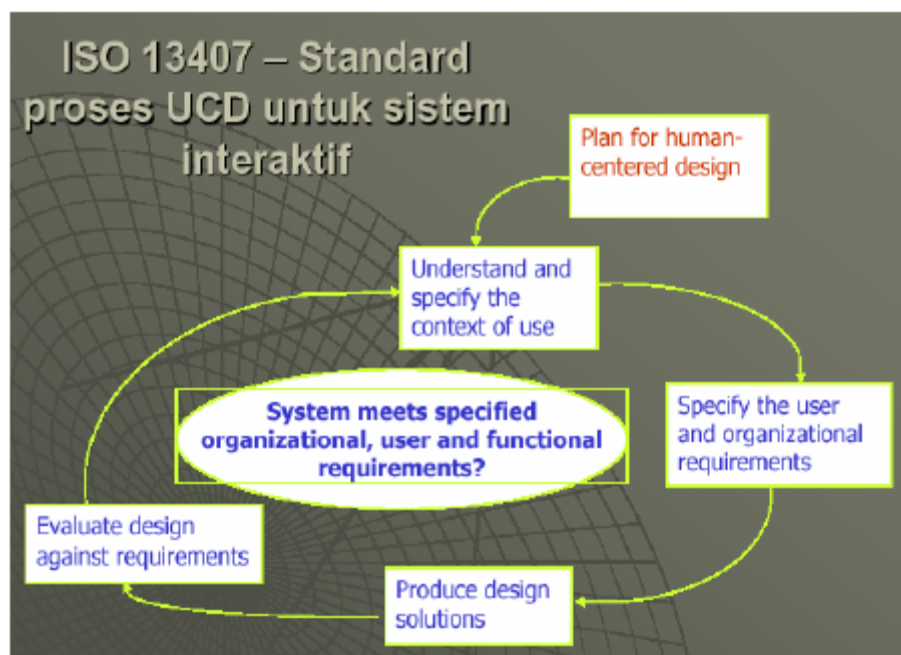
Pengguna mempunyai hak untuk berkomunikasi dengan penyedia teknologi dan menerima pemikiran dan tanggapan yang membantu jika diperlukan.

10. Usability

Pengguna harus menjadi penguasa teknologi perangkat lunak dan perangkat keras bukan sebaliknya. Produk harus dapat digunakan secara alami dan intuitif.

Lima sifat yang harus diubah yaitu :

1. Selama proses pengembangan penekanan dan fokus hanya ditujukan pada mesin atau sistem, bukan pada orang yang akan menggunakan sistem.
2. Dengan penetrasi teknologi ke pasar pelanggan, maka audiens terbesar telah berubah dan terus berubah secara dramatis. Perkembangan organisasi masih lambat dalam mensikapi evolusi ini.
3. Perancangan sistem yang *usable* adalah sulit, tidak dapat diprediksi dan perlu usaha keras. Namun saat ini banyak organisasi yang mensikapinya sebagai hal yang biasa.
4. Organisasi menggunakan tim dan pendekatan yang sangat khusus dalam menghasilkan dan mengembangkan sistem, namun gagal untuk mengintegrasikannya satu dengan yang lain.
5. Saat ini penekanan dan kebutuhan adalah pada aspek perancangan, sementara banyak perancang memposisikan pikiran dan keterampilannya hanya untuk implementasi teknis.



Gambar 5.3 Standar Proses UCD

Menentukan kebutuhan pengguna dan organisasi dengan cara :

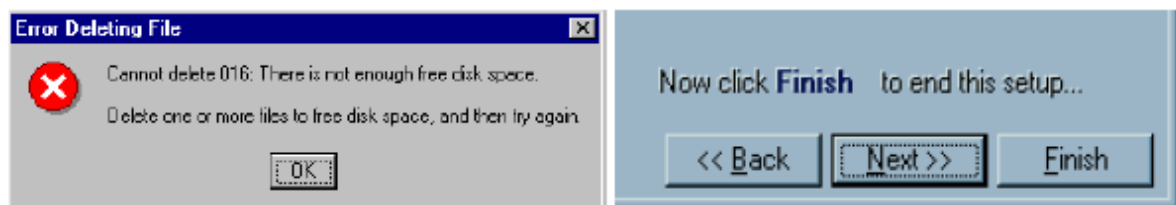
1. Kualitas perancangan interaksi manusia dan komputer serta *workstation*.
2. Kualitas dan isi tugas pengguna, contoh : apakah operator bertanggung jawab melakukan konfigurasi sistem seperti kenyamanan, keselamatan, kesehatan dan khususnya motivasi.
3. Kinerja tugas yang efektif khususnya dalam hal transparansi aplikasi ke pengguna.
4. Kerjasama dan komunikasi yang efektif diantara pengguna dan pihak ketiga yang relevan.
5. Dibutuhkan kinerja sistem baru terhadap tujuan finansial.

Solusi perancangan yang dihasilkan :

- Dengan menggunakan pengetahuan yang ada (standards, contoh petunjuk sistem lain, dan lain-lain), untuk mengembangkan suatu proposal solusi perancangan.
- Membuat solusi perancangan lebih konkrit( dengan menggunakan simulasi, prototipe , dan lain-lain)
- Memperlihatkan prototipe ke pengguna dan mengamatinya saat melakukan tugas yang spesifik, dengan atau tanpa bantuan evaluator
- Menggunakan umpan balik untuk perbaikan rancangan.
- Mengulangi proses sampai tujuan perancangan terpenuhi.

Secara umum prinsip untuk membangun antarmuka ini yaitu :

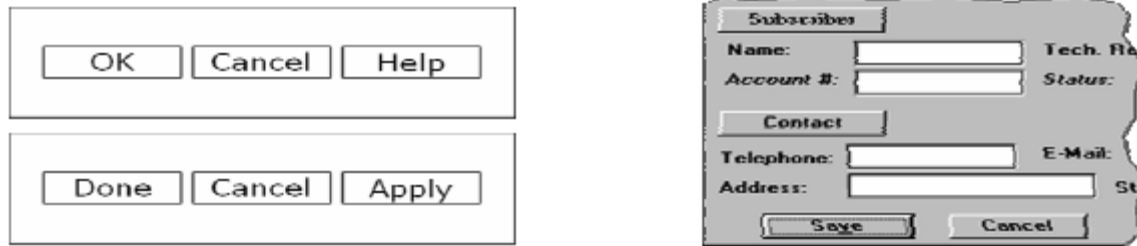
- Gunakan dialog yang sederhana dan alami, menghindari penggunaan jargon dengan menampilkan *techno-speak* seperti gambar di bawah :



Gambar 5.4 Techno-Speak

- Berusaha untuk konsisten, pada urutan, aksi, perintah, layout dan daftar katakata penting seperti gambar di bawah :





Gambar 5.5 Perintah yang konsisten

- Sediakan umpan balik yang informatif, secara terus-menerus memberitahu pengguna tentang yang terjadi.
- Minimalkan beban ingatan user, mendeskripsikan format *input* yang diperlukan seperti contoh dan default.

Contohnya:

Date \_\_ - \_\_ - \_\_ (DD-Mmm-YY, e.g., 02-Aug-93)

- Memberikan perubahan aksi yang mudah, seperti memberikan *undo*
- Menyediakan jalan keluar yang jelas, misal untuk mengatasi salah dalam memilih perintah.
- Menyediakan shortcut, memungkinkan user untuk menjalankan operasi yg sering dipakai dengan cepat.
- Mendukung fokus internal dari kontrol, seperti penggunaan *enter next command* lebih baik daripada *Ready for next command*
- Menyediakan penanganan kesalahan, memberi petunjuk tentang kesalahan yg dilakukan user dan langkah perbaikannya
- Menyediakan help dan dokumentasi, untuk membantu pengguna khususnya pengguna baru dalam menggunakan *software*.

## 5.5 Desain Grafik

Desain grafik mempunyai sifat *look* dan *feel*, maksudnya :

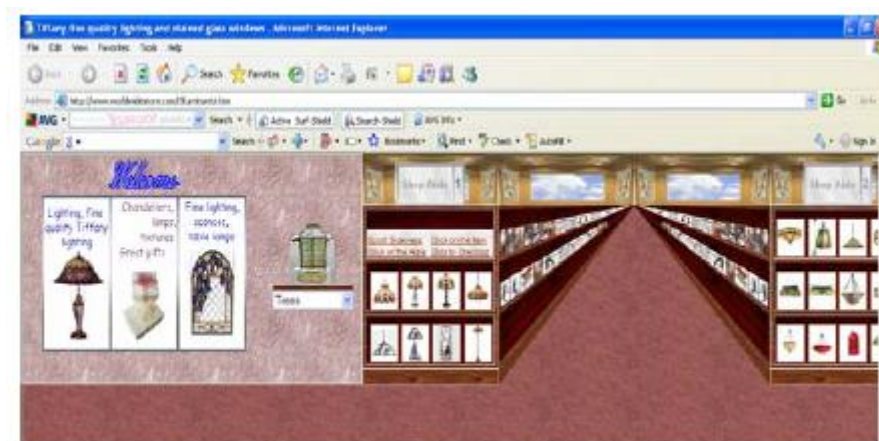
- *Look* terlihat pada tampilan secara visualisasi seperti:
- simbolisasi/*icon*
- *enable/disable*

- *active/inactive*, dan lain-lain.
- *feel* akan terasa ketika pengguna melakukan seperti :
- *drag, drag* dan *drop*
- *click/dblclick*.

Prinsip dalam membuat desain grafik yaitu :

#### ☐ Metaphor

Penggunaan presentasi dan unsur visual untuk beberapa item yang relevan. Contoh: pada *website* toko furniture *online* dengan visualisasi etalase di toko dibuat mirip dengan aslinya, seperti gambar di bawah ini:



Gambar 5.6 Visualisasi toko online

☐ Kejelasan, tiap-tiap elemen pada suatu interface harus mempunyai suatu tujuan. Contoh: penggunaan *white space* mempunyai pengaruh antara lain:

- Menyediakan keseimbangan dan simetri dalam penggunaannya
- Memperkuat dampak pesan yg disampaikan
- Mengijinkan mata untuk istirahat dari unsur aktivitas
- Digunakan untuk meningkatkan kesederhanaan, kerapian, kelas.



Gambar 5.7 Penggunaan white space

- Konsistensi, pada layout, warna, gambar, ikon, teks, di dalam *screen*, antar *screen* dan lain-lain.
- Alignment, penggunaan model seperti :
  - *western world*, dimulai dari kiri atas
  - *grid*, garis vertikal dan horizontal untuk membantu tata letak window.
  - *Left, center* atau *right*



Gambar 5.8 Grid

□ Pendekatan(Proximity), item dikelompokkan berdasarkan hal-hal yang berkaitan. Jarak dalam proximity tidak menyiratkan suatu hubungan.

Nama :

Nama :

Name <input type="text"/> Addr1 <input type="text"/> Addr2 <input type="text"/> City <input type="text"/> State <input type="text"/> Phone <input type="text"/> Fax <input type="text"/>	Name <input type="text"/> Addr1 <input type="text"/> Addr2 <input type="text"/> City <input type="text"/> State <input type="text"/> Phone <input type="text"/> Fax <input type="text"/>	Name <input type="text"/> Addr1 <input type="text"/> Addr2 <input type="text"/> City <input type="text"/> State <input type="text"/> Phone <input type="text"/> Fax <input type="text"/>
--	--	--

Gambar 5.9 Proximity

□ Kontras, digunakan untuk membedakan kontrol yang aktif atau tidak dan mengatur item yang paling penting dengan *highlight*(menyorotnya).



Gambar 5.10 Penggunaan Kontras

Pada gambar di atas untuk form yang harus diisi menggunakan warna yang kontras dari tampilan lainnya.

- Tipografi
- Karakter dan symbol harus nyata dan dapat dibedakan.

- Hindari penggunaan semua huruf besar
- Readability(keadaaan yg dapat dibaca)
- Bagaimana agar mudah membaca teks yg banyak
- Legibility (Sifat mudah dibaca)
- Bagaimana agar mudah untuk mengenali teks pendek yg muncul secara tiba-tiba
- Jenis huruf=font

Serif font – readability

Times, Bookman Sans serif font - legibility

Tahoma, Arial

- Petunjuk
  - Gunakan serif utk teks yg panjang; sans serif utk teks utama
  - Gunakan 1-2 font (3 maks)
  - Jangan gunakan bold, italic, kapital utk teks yg panjang
  - gunakan ukuran maks 1-3 point
  - hati-hati penggunaan teks untuk latar belakang dengan warna

#### • Warna

- Kita melihat dunia melalui refleksi model warna
- cahaya menerangi suatu permukaan dan dicerminkan pada mata kita
- printer
- Pada monitor, susunan khas RGB, nilai 0-255 setiap red, green, blue
- Gunakan untuk satu tujuan, bukan hanya menambah beberapa warna pada tampilan
- Tampilkan image berwarna pada background hitam
- Pilih foreground dengan warna terang (white, bold green,...)
- Hindari coklat dan hijau sebagai warna background.
- Pastikan warna foreground kontras dengan warna background
- Gunakan warna untuk menarik perhatian, komunikasi organisasi, untuk menandai status, untuk menentukan hubungan
- Hindari penggunaan warna untuk tugas yang tidak berhubungan
- Warna berguna utk mendukung pencarian
- Selalu konsisten dgn asosiasi pekerjaan dan budaya

- Red
  - hot, warning, love
- Pink
  - female, menarik
- Orange
  - musim, hangat, Halloween
- Yellow
  - happy, hati-hati, suka cita
- Brown
  - warm, fall, ko, daratan
- Green
  - subur, pastoral, iri/cemburu
- Purple (ungu)
  - meriah, canggih.
  - Ikon
- Mewakili objek atau aksi yg lazim dan dapat dikenali
- Membuat ikon lebih menonjol dari background
- Pastikan ikon yg terpilih tampak terlihat jelas dari ikon yg tidak terpilih
- Buat setiap ikon yg berbeda
- Hindari rincian yg berlebihan



Gambar 5.11 Desain ikon

## 5.6 Bahan Diskusi

Menurut Anda, Mengapa dalam membuat desain interface membutuhkan desain yang *universal* ( diterima secara umum ) ?

## 5.7 Rangkuman

- ☐ Salah satu kriteria penting dalam membuat antarmuka adalah tampilan yang menarik
- ☐ Komponen antarmuka antara lain : model pengguna, bahasa perintah, umpan balik dan tampilan informasi.
- ☐ Desain yang baik berdasarkan pada user (*user-centered Design* )
- ☐ Desain grafik mempunyai sifat *look* dan *feel*
- ☐ Prinsip desain grafik yaitu metaphor, kejelasan, konsistensi, alignment, proximity dan kontras.

## 5.8 Latihan Soal-soal

1. Apakah sebenarnya tujuan dari desain itu ? Kaitkan juga dengan prinsip-prinsip yang mendukung penggunaan!
2. Apakah yang harus kita lakukan agar fokus pada desain?
3. Dalam merancang antar muka agar mendapatkan hasil yang maksimal kita dapat menggunakan beberapa "Prinsip". Sebutkan dan jelaskan prinsip-prinsip tersebut!
4. Apa saja prinsip dari desain itu?
5. Mengapa membuat suatu desain, kita harus memperhatikan aspek manusia dan komputer
6. Mengapa dalam membuat desain itu mempertimbangkan aspek konsistensi?
7. Apa pengaruhnya jika desain yang kita buat tidak mempertimbangkan user friendly?
8. Jelaskan tentang prinsip dalam desain grafik?