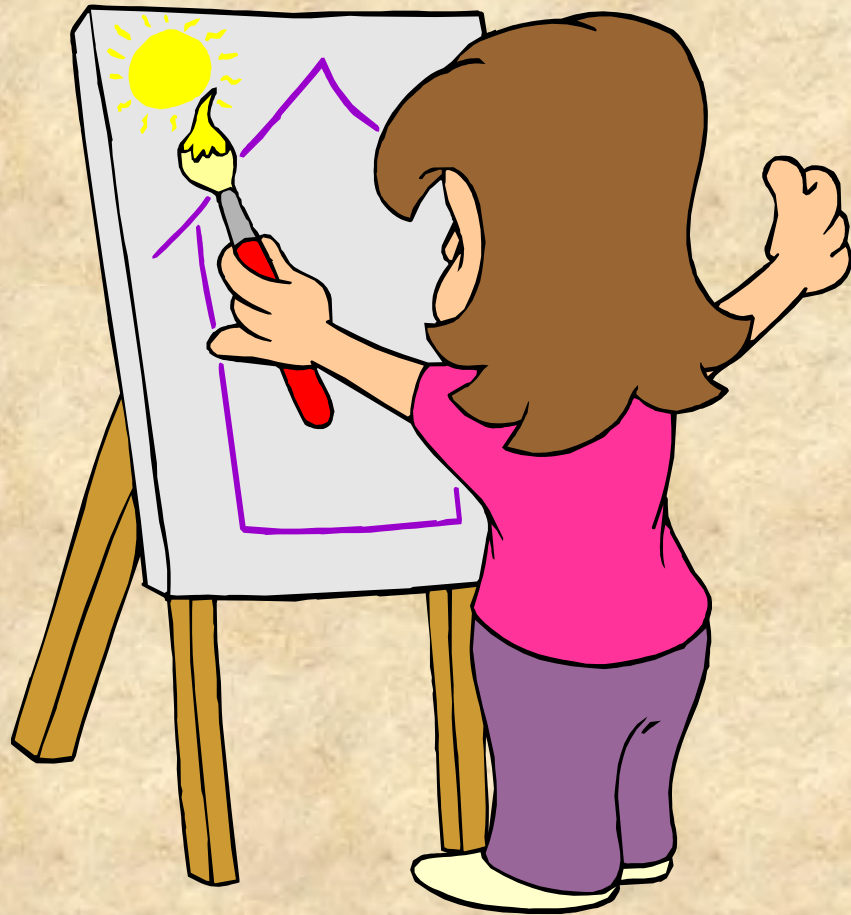


Perancangan Tampilan



Pendahuluan

- Salah satu kriteria penting dari sebuah antarmuka adalah tampilan yang menarik
- Yang harus dimiliki oleh perancang tampilan:
 - Jiwa seni yang memadai
 - Mengetahui selera user secara umum
- Seorang perancang tampilan HARUS mendokumentasikan semua pekerjaan yang telah dia kerjakan selama ini, sebagai bahan evaluasi pembuatan tampilan yang baru atau memperbaiki tampilan sebelumnya
- PROGRAM YANG DIBUAT BUKAN UNTUK ANDA, NAMUN UNTUK ORANG LAIN



Dokumentasi Rancangan

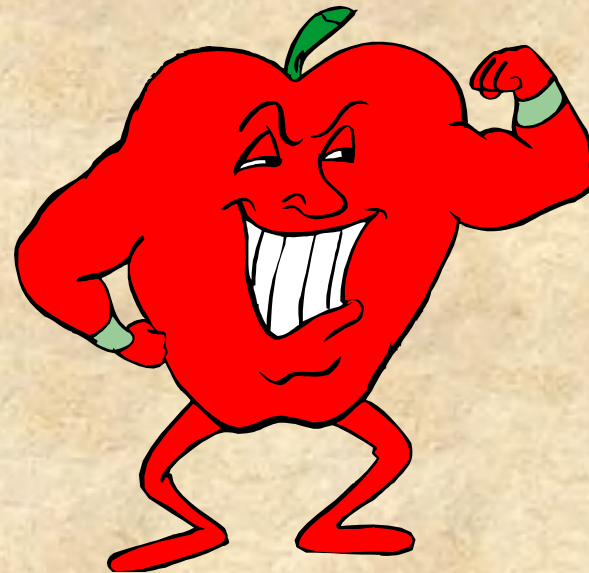
Dokumentasi rancangan dapat dikerjakan dalam beberapa cara :

1. Membuat Sketsa pada kertas
2. Menggunakan piranti prototipe GUI
3. Menuliskan tekstual yang menjelaskan tentang kaitan antara satu jendela dengan jendela lainnya.
4. Menggunakan piranti bantu yang disebut CASE (*Computer Aided Software Engineering*)

Cara Pendekatan

Jenis program aplikasi:

- Special purpose software
- General purpose software



Special Purpose Software

- Yaitu program aplikasi untuk keperluan khusus dengan user yang khusus pula (*special purpose software*)
- *User-centered design approach* adalah perancangan antarmuka yang melibatkan pengguna. Perancang dan pengguna duduk bersama-sama untuk merancang antarmuka yang diinginkan pengguna, sehingga dengan cara ini pengguna seolah-olah sudah mempunyai gambaran nyata tentang antarmuka yang nanti akan mereka gunakan.
- Kelompok user dapat dengan mudah diperkirakan, baik dari segi keahlian maupun ragam antarmuka yang akan digunakan

Special Purpose Software...Ljt

- Mis: program inventori gudang, pengelolaan data akademis mahasiswa, pelayanan reservasi hotel
- Pendekatan yang digunakan:
 - *User-centered design approach*: perancang dan user bersama-sama membuat tampilan antarmuka
 - *User design approach*: hanya user yang membuat tampilan antarmuka

General Purpose Software

- Yaitu program aplikasi yang akan digunakan oleh berbagai tingkat kepandaian dan karakteristik yang sangat beragam, perlu memiliki salah satu kunci penting dalam pembuatan modul antarmuka dengan melakukan customizazion, sehingga pengguna dapat menggunakan program aplikasi dengan wajah antarmukanya yang sesuai dengan selera masing-masing pengguna.
- Perancang dapat melakukan 'pemaksaan' kepada user untuk menerima tampilan antarmukanya, walaupun dapat memberikan dampak program tersebut tidak laku

General Purpose Software Lanjutan.....

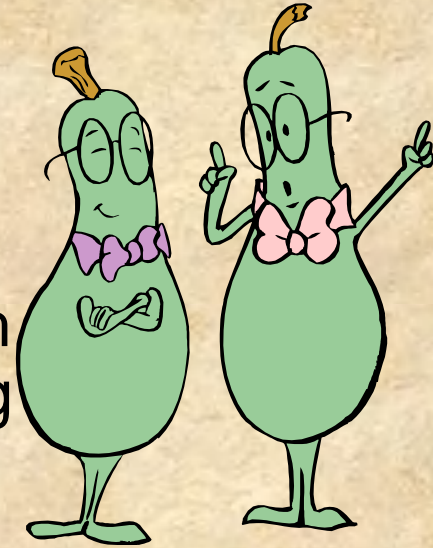
- Salah satu cara yang dapat dilakukan adalah user dapat melakukan perubahan tampilan sesuai dengan keinginannya sendiri (*customization*), mis: merubah warna dasar, pengaturan *desktop*, *wallpaper*, *screensaver*, dll

Komponen Antarmuka Pengguna

1. **Model pengguna;** memungkinkan user untuk mengembangkan pemahaman yang mendasar tentang apa yang dikerjakan oleh program, bahkan oleh user yang sama sekali tidak mengetahui teknologi komputer
2. **Bahasa perintah;** sedapat mungkin menggunakan bahasa alami
3. **Umpanbalik;** kemampuan sebuah program yang membantu user untuk mengoperasikan program itu sendiri
4. **Tampilan informasi;** digunakan untuk menunjukkan status informasi atau program ketika user melakukan suatu tindakan

Urutan Perancangan

1. **Pemilihan ragam dialog**, dipengaruhi oleh karakteristik populasi pengguna, tipe dialog yang diperlukan, dan kendala teknologi yang ada untuk mengimplementasikan ragam dialog tersebut.
2. **Perancangan struktur dialog**; melakukan analisis tugas dan menentukan model pengguna dari tugas tersebut
3. **Perancangan format pesan**; tata letak tampilan, keterangan tektual secara terinci, dan efisiensi inputing data harus mendapat perhatian lebih



Urutan Perancangan

4. **Perancangan penanganan kesalahan;** untuk menghindari adanya kesalahan yang timbul, maka adanya kondisi yang disebut abnormal termination, yaitu eksekusi program berhenti karena terjadi kesalahan. Bentuk penanganan kes
5. Kesalahan :
 - Validasi pemasukan data
 - Proteksi pengguna
 - Pemulihan dari kesalahan
 - Penampilan pesan salah yang tepat dan sesuai dengan kesalahan yang terjadi pada waktu itu.

Urutan Perancangan

5. Perancangan struktur data

Setelah semua aspek antarmuka dipertimbangkan, anda harus menentukan struktur data yang dapat digunakan untuk menyajikan dan mendukung fungsional komponen-komponen antarmuka yang diperlukan. Struktur data ini harus dipetakan langsung ke dalam model pengguna yang telah dibuat.

Hal ini perlu ditekankan agar keinginan pengguna dan model sistem yang telah dirancang saling mempunyai kecocokan satu sama lain.

PERANCANGAN TAMPILAN BERBASIS TEKS

- Pada perancangan tampilan untuk antar muka berbasis teks, ada enam faktor yang harus dipertimbangkan agar diperoleh tata letak tampilan yang berkualitas tinggi, yaitu :
 1. Urutan Penyajian
 2. Kelonggaran (Spaciousness)
 3. Pengelompokkan
 4. Relevansi
 5. Konsistensi
 6. Kesederhanaan

Perancangan Tampilan Berbasis Teks...Lanjutan

- Urutan penyajian; disesuaikan dengan model pengguna
- Kelonggaran; mis: dengan adanya jarak spasi antar perintah, atau penempatan yang khusus
- Pengelompokkan data yang saling berkaitan
- Relevansi; hanya pesan-pesan yang relevan saja yang ditampilkan di layar
- Konsistensi; menggunakan suku kata yang konsisten untuk menjelaskan sesuatu
- Kesederhanaan; menggunakan singkatan yang dipahami oleh umum

PERANCANGAN TAMPILAN BERBASIS GRAFIS

- Dengan antarmuka berbasis grafis berbagai kemudahan dalam hal pengontrolan format tampilan dapat dikerjakan dengan lebih mudah dan fleksibilitas tampilan dapat semakin dirasakan oleh perancang tampilan maupun penggunaannya.
- Disisi lain, kita harus memperhatikan beberapa kendala dalam penerapan antarmuka berbasis grafis, antara lain : waktu tanggap, kecepatan tampilan, lebar tampilan, dan tipe tampilan

Perancangan Tampilan Berbasis Grafis....Lanjutan

- Ilusi pada obyek-obyek yang dapat dimanipulasi, mis: gambar disket, printer, dll
- Urutan visual dan fokus pengguna, mis: tanda kedip untuk posisi kursor, penggunaan warna yang berbeda
- Struktur internal; berguna untuk menunjukkan bahwa obyek yang sedang dihadapi dapat dimodifikasi sesuai dengan keinginan user
- Kosakata grafis yang konsisten dan sesuai, mis: gambar disket, printer, dll
- Kesesuaian dengan media/informasi yang akan disampaikan

Contoh beberapa kemampuan pada Microsoft Windows

- Pengguna tidak harus mengingat perintah-perintah yang sering kali cukup panjang, tetapi cukup dikerjakan dengan melihat dan menunjuk ke suatu gambar yang mewakili suatu aktifitas (ikon).
- Penggunaan borang property atau option untuk mengatur kenampakan (wajah) desktop.
- Kemampuan WYSIWYG
- Perintah-perintah yang berlaku umum, seperti move, delete atau copy dll.

5 faktor yang perlu diperhatikan pada perancangan antarmuka berbasis grafis

1. Ilusi pada objek-objek yang dapat dimanipulasi.

Perancangan antarmuka berbasis grafis harus melibatkan tiga komponen :

- Kumpulan Objek
- Penampilan objek-objek
- Mekanisme yang konsisten

2. Urutan visual dan Fokus pengguna

- Antar muka grafis dapat digunakan untuk menarik perhatian pengguna antara lain dengan membuat objek yang berkedip, menggunakan warna tertentu, serta menyajikan suatu animasi yang akan lebih menarik perhatian pengguna

3. Struktur Internal

- Pada pengolahan kata kita sering menulis beberapa kata yang berbeda dengan kata-kata yang lain, misalnya ada sekelompok kata yang ditebalkan, dimiringkan atau diberi garis bawah.
- Pada antarmuka berbasis grafis, khususnya pada objek-objek yang dapat dimanipulasi, perancang juga harus memberikan struktur internal (*reveral structure*) dalam bentuk yang berbeda dengan yang digunakan pada dokumen tekstual, untuk memberi tahu pengguna sejauh mana pengguna dapat mengubah atau memanipulasi objek tersebut.

4. Kosa kata grafis yang konsisten dan sesuai

Pada program aplikasi yang berbeda, penggunaan simbol biasanya disesuaikan dengan kreatifitas perancangnya.

5. Keseuaian dengan pengguna

- Karakteristik dari layar tampilan yang digunakan akan mempunyai pengaruh yang besar terhadap keindahan “wajah” antarmuka yang akan ditampilkan.
- Dengan semakinnya canggihnya teknologi layar tampilan pada saat ini, kreatifitas perancang tampilan lah yang saat ini lebih dituntut untuk memenuhi permintaan pengguna akan aspek kenyamanan dan keramahan antarmuka.



Penanganan Kesalahan

Validasi pemasukan data, mis: jika user harus memasukkan bilangan positif, namun dia memasukkan data negatif atau nol, maka harus ada mekanisme untuk mengulang pemasukan data tersebut

- Proteksi user; program memberi peringatan ketika user melakukan suatu tindakan secara tidak sengaja, mis: penghapusan berkas
- Pemulihan dari kesalahan: tersedianya mekanisme untuk membatalkan tindakan yang baru saja dilakukan
- Penampilan pesan salah yang tepat dan sesuai dengan kesalahan yang terjadi pada waktu itu

Kesalahan dibagi 2, yaitu :

- Kesalahan pada saat implementasi program yaitu kesalahan sintaks yang secara langsung akan dideteksi oleh kompiler (*compile-time error*), terjadi pada saat program sedang dikompilasi.
- Kesalahan Logika ketika program sedang dijalankan, terjadi pada saat program dijalanka (*run-time error atau fatal error*). Kesalahan ini akan mengakibatkan terhentinya program secara abnormal.

Contoh Kesalahan Sintaks

[1] var A, B, C1 : integer;

[2] begin

[3] readln(A); readln(B)

[4] C1:= A/B;

[5] writeln(C1);

[6] end.

Contoh

- Dalam pascal, operator / (bagi) tidak dapat dilakukan pada peubah yang bertipe integer. Ketika program dikompilasi akan terdapat pesan yang berbunyi “*type mismatch*”
- Tetapi ketika baris ke-4 diubah menjadi `C1:= A Div B;` maka kesalahan itu akan hilang yang berarti program berhasil dikompilasi dan siap dijalankan.

Contoh Kesalahan Logika

[1] var A, B, C1 : integer;

[2] begin

[3] readln(A); readln(B)

[4] C1:= A div B;

[5] writeln(C1);

[6] end.

Contoh kesalahan Logika.....Ijt

- Penyebab kesalahan logika bisa dari Pengguna ketika mengisikan data yang tidak sah dan dari proses eksekusi.
- Pada saat anda mengganti nilai $B=0$ maka akan terdapat pesan yang berbunyi *Divide by Zero* dan program tidak memberikan hasil.
- Dalam operasi matematis pembagian dengan nol tidak mungkin dilaksanakan.

Program setelah diubah

```
[1] var A, B, C1 : integer;  
[2] begin  
[3] readln(A); readln(B)  
[4] if B = 0 then;  
[5] writeln(pembagian dengan nol tidak diijinkan);  
[6] else  
[7] begin  
[8] C1 := A div B;  
[9] writeln(C1);  
[10] end;  
[11] end.
```