



## Perancangan Tampilan (*Design*)

- **Pengantar**
- Salah satu kriteria penting dari sebuah antarmuka adalah tampilan yang menarik. Seorang pengguna, apalagi pengguna baru, biasanya tertarik untuk mencoba sebuah program aplikasi dengan terlebih dahulu tertarik pada suatu tampilan yang ada di hadapan matanya.



- Dokumentasi rancangan dapat dikerjakan atau dilakukan dalam beberapa cara:
- Membuat sketsa pada kertas
- Menggunakan peranti prototipe GUI,
- Menuliskan tekstual yan menjelaskan tentang kaitan antara satu jendela dengan jendela yan lain,
- Menggunakan peranti bantu yan disebut **CASE** (*Computer Aided Software Engineering*).



## – CARA PENDEKATAN

- Program aplikasi, pada dasarnya dapat dikelompokkan ke dalam dua kategori besar, yakni program aplikasi untuk keperluan khusus dengan pengguna yang khusus pula (*special purpose software*) dan program aplikasi yang akan digunakan oleh banyak pengguna (*general purpose software*), yang juga sering dikenal dengan sebutan *public software*. Karena perbedaan pada calon pengguna, maka perancang program antarmuka harus benar-benar memperhatikan hal ini.



- Pada kelompok pertama, yakni pada program aplikasi untuk keperluan khusus, misalnya program aplikasi untuk inventori gudang, pengelolaan data akademis mahasiswa, pelayanan reservasi hotel, dan program-program aplikasi serupa, kelompok calon pengguna yang akan memanfaatkan program aplikasi tersebut dapat dengan mudah diperkirakan., baik dalam hal keahlian pengguna, maupun ragam antarmuka yang akan digunakan



- Pendekatan secara *user centered design* adalah perancangan antarmuka yang melibatkan pengguna. Pelibatan pengguna disini tidak diartikan bahwa pengguna harus ikut memikirkan bagaimana implementasinya nanti, tetapi pengguna diajak untuk aktif berpendapat ketika perancang antarmuka sedang menggambar “wajah” antarmuka.



- Dengan kata lain, perancang dan pengguna duduk bersama-sama untuk merancang wajah antarmuka yang diinginkan pengguna. Pengguna menyampaikan keinginannya, sementara perancang menggambar keinginan pengguna tersebut sambil menjelaskan keuntungan dan kerugian wajah antarmuka yang diinginkan oleh pengguna, serta kerumitan implementasinya. Dengan cara seperti ini, pengguna seolah-olah sudah mempunyai gambaran nyata tentang antarmuka yang nanti akan mereka gunakan



- **PRINSIP DAN PETUNJUK PERANCANGAN**
- Antarmuka pengguna secara alamiah terbagi menjadi empat komponen: **model pengguna, bahasa perintah, umpan balik, dan penampilan informasi**. Model pengguna merupakan dasar dari tiga komponen yang lain



- Model pengguna merupakan model konseptual yang diinginkan oleh pengguna dalam memanipulasi informasi dan proses yang diaplikasikan pada informasi tersebut. Setelah pengguna mengetahui dan memahami model yang ia inginkan, dia memerlukan peranti untuk memanipulasi model itu.



- Piranti pemanipulasian model ini sering disebut dengan bahasa perintah (*command language*),
- yang sekaligus merupakan komponen kedua dari antarmuka pengguna. Idealnya, program komputer kita mempunyai bahasa perintah yang alami, sehingga model pengguna dengan cepat dapat dioperasionalkan.



- Komponen ketiga adalah umpanbalik. Umpanbalik di sini diartikan sebagai kemampuan sebuah program yang membantu pengguna untuk pengoperasian program itu sendiri. Umpanbalik dapat berbentuk: pesan-pesan penjelasan, pesan penerimaan perintah, indikasi adanya obyek terpilih, dan penampilan karakter yang diketikkan lewat papan ketik



- Beberapa bentuk umpanbalik terutama ditujukan kepada pengguna yang belum berpengalaman dalam menjalankan program aplikasi itu. Umpanbalik dapat digunakan untuk memberi keyakinan bahwa program telah menerima perintah pengguna dan dapat memahami maksud perintah tersebut.



- Komponen keempat adalah tampilan informasi. Komponen ini digunakan untuk menunjukkan status informasi atau program ketika pengguna melakukan suatu tindakan. Pada bagian ini, perancang harus menampilkan pesan-pesan tersebut seefektif mungkin, sehingga mudah dipahami oleh pengguna



- **URUTAN PERANCANGAN**
- Perancangan dialog, seperti halnya perancangan sistem yang lain, harus dikerjakan secara top-down. Proses perancangannya dapat dikerjakan secara *stepwise refinement* sebagai berikut:



## Pemilihan Ragam Dialog

- Pemilihan ragam dialog dipengaruhi oleh karakteristik populasi pengguna (pengguna mula, menengah, atau pengguna ahli), tipe dialog yang diperlukan, dan kendala teknologi yang ada untuk mengimplementasikan ragam dialog tersebut. Ragam dialog yang terpilih dapat berupa sebuah program tunggal, atau sekumpulan ragam dialog yang satu sama lain saling mendukung



- **Perancangan Struktur Dialog**
- Tahap kedua adalah melakukan analisis tugas dan menentukan model pengguna dari tugas tersebut untuk membentuk struktur dialog yang sesuai. Dalam tahap ini pengguna banyak dilibatkan, sehingga pengguna langsung mendapatkan umpanbalik yang berupa diskusi informal maupun prototipe dari dialog yang nantinya akan ia digunakan.



## Perancangan Format Pesan

- Pada tahap ini tata letak tampilan dan keterangan tektual secara terinci harus mendapat perhatian lebih. Selain itu, kebutuhan data masukan yang mengharuskan pengguna untuk memasukkan data ke dalam komputer juga harus dipertimbangkan dari segi efisiensinya. Salah satu contohnya adalah dengan mengurangi pengetikan yang tidak perlu dengan cara mengaktifkan penggunaan tombol.



- **Perancangan Penanganan Kesalahan**
- Bentuk-bentuk penanganan kesalahan yang dapat dilakukan antara lain adalah:
  - Validasi pemasukan ata: misalnya jika pengguna harus memasukkan bilangan positif, sementara ia memasukkan data negatif atau nol, maka harus ada mekanisme untuk mengulang pemasukan data tersebut.
  - Proteksi pengguna: program memberi peringatan ketika pengguna melakukan suatu tindakan secara tidak sengaja, misalnya penghapusan berkas.
  - Pemulihan dari kesalahan: teredianya mekanisme untuk membatalkan tindakan yang baru saja dilakukan.
  - Penampilan pesan salah yang tepat dan sesuai dengan kesalahan yang terjadi pada waktu itu.