

BAB 6  
Design Interface  
(Perancangan Tampilan)

1. Cara Pendekatan

Terdapat 2 pendekatan dalam merancang tampilan yaitu :

- a. **User centered design approach.:** program aplikasi untuk keperluan khusus(*special purpose software*) 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. Untuk kelompok ini ada satu pendekatan yang dapat dilakukan
- b. **User design approach.:** 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.

2. Prinsip dan Petunjuk Perancangan

- a. Urutan Perancangan
  - **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.

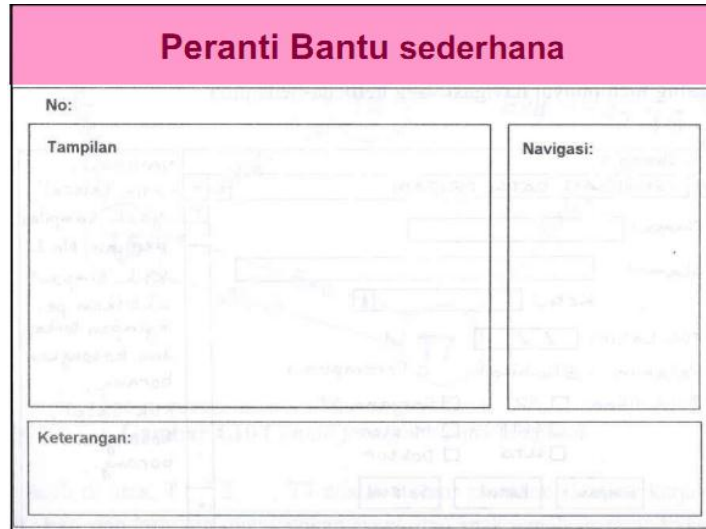
- **Perancangan struktur dialog:** Tahap kedua adalah melakukan analisis tugas dan menentukan model pengguna dari tugas tersebut untuk membentuk struktur dialog yang sesuai. Dalam tahap ini pengguna banyak dilibatkan, sehingga pengguna langsung mendapatkan umpanbalaik yang berupa diskusi informal maupun prototipe dari dialog yang nantinya akan ia digunakan.
- **Perancangan format pesan:** Pada tahap ini tata letak tampilan dan keterangan tekstual 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 mengefektifkan 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.
  
  - **Perancangan struktur data:** Struktur data yang diperlukan untuk mengimplementasikan dialog berbasis grafis jauh lebih rumit dibandingkan dengan struktur data yang diperlukan pada dialog berbasis tekstual. Meskipun demikian, sesulit atau semudah apapun struktur data yang akan digunakan, struktur data tersebut harus diturunkan dari spesifikasi antarmuka yang telah dibuat. Hal ini perlu ditekankan agar keinginan pengguna dan model sistem yang telah dirancang saling mempunyai kecocokan satu sama lain.
- b. Perancangan tampilan berbasis teks
- Urutan penyajian
  - Kelonggaran
  - Pengelompokan
  - Relevansi
  - Konsistensi
  - Kesederhanaan
- c. Perancangan tampilan berbasis Grafis
- Terdapat 5 faktor yang diperlukan yaitu :
- **Ilusi pada obyek-obyek.:** Perancangan antarmuka berbasis grafis yang efektif harus melibatkan tiga komponen. Pertama, gunakan kumpulan obyek yang disesuaikan dengan aplikasi yang akan dibuat. Jika obyek-obyek itu belum ada, kita dapat mengembangkannya sendiri. Kedua, penampilan obyek-obyek grafis harus dilakukan dengan keyakinan bahwa ia akan dengan mudah dimengerti oleh pengguna. Ketiga, gunakan mekanisme yang konsisten untuk memanipulasi obyek yang akan muncul di layar.

- Urutan visual dan fokus pengguna.: Antarmuka dapat digunakan untuk menarik perhatian pengguna antara lain dengan membuat suatu obyek berkedip, menggunakan warna tertentu untuk obyek-obyek tertentu, serta menyajikan suatu animasi yang akan lebih menarik perhatian pengguna. Tetapi, penggunaan rangsangan visual yang berlebihan justru akan membuat pengguna bingung dan merasa tidak nyaman.
- Struktur Internal: Pada pengolahan kata kita seringkali menulis beberapa kata yang berbeda dengan kata-kata yang lain, misalnya ada sekelompok kata yang ditebalkan, dimiringkan, atau diberi garis bawah. Pada salah satu pengolahan kata, kita dapat melihat apa yang disebut dengan *reveal code*, yakni suatu tanda khusus yang digunakan untuk menunjukkan adanya perbedaan *font style*. *Reveal code* ini tidak akan ikut dicetak, tetapi digunakan untuk menunjukkan kepada pengguna antara lain tentang *font style* yang digunakan, batas kiri dan batas kanan dari halaman teks serta informasi yang lain. *Reveal code* biasanya berupa suatu karakter khusus.
- Kosakata grafis yang konsisten dan sesuai.: Penggunaan simbol-simbol obyek, atau ikon, memang tidak ada standarnya, dan biasanya disesuaikan dengan kreatifitas perancangannya
- Kesesuaian dengan media.: Karakteristik khusus dari layar tampilan yang digunakan akan mempunyai pengaruh yang besar terhadap keindahan “wajah” antarmuka yang akan ditampilkan. Pada layar tampilan yang masih berbasis pada karakter, misalnya CGA, pemunculan gambar tidak akan secantik apabila kita menggunakan layar tampilan yang sering disebut dengan *bitmap* atau *raster display*. Dengan semakin canggihnya teknologi layar tampilan pada saat ini, kreatifitas perancang tampilanlah yang saat ini lebih dituntut untuk

memenuhi permintaan pengguna akan aspek kenyamanan dan keramahan antarmuka.

- d. Waktu tanggap: Waktu tanggap yang lama akan mengalihkan perhatian pengguna untuk melakukan aktifitas lain.
  - e. Penanganan Kesalahan: Penanganan kesalahan yang tepat biasanya dilakukan dengan memberikan pesan umpan balik kepada pengguna akan keadaan eksekusi program saat itu. Kesalahan pada program dapat disebabkan oleh dua hal: pertama kesalahan sintaksis yang bisa dideteksi kompiler biasa disebut *compile-time error* dan kedua kesalahan logika ketika program dijalankan biasa disebut *run-time error* atau *fatal error*.
3. Peranti Bantu Sederhana: Perancang seharusnya membuat dokumentasi akan bentuk-bentuk tampilan yang akan diimplementasikan. Peranti bantu sederhana yang dapat digunakan misalnya adalah lembar kerja tampilan/LKT (*screen design work sheet*). Pada LKT, disajikan empat bagian:
1. Nomor lembar kerja
  2. Bagian tampilan
  3. Bagian navigasi
  4. Bagian keterangan



**Lembar Kerja Tampilan**

<p>No. sheet: 5</p> <p><b>PENGISIAN DATA PRIBADI</b></p> <p>Nama: <input style="width: 150px;" type="text"/></p> <p>Alamat: <input style="width: 200px;" type="text"/></p> <p>Kota: <input style="width: 80px;" type="text"/></p> <p>TGL. Lahir: <input style="width: 50px;" type="text"/></p> <p>Kelamin: <input checked="" type="radio"/> Laki-laki <input type="radio"/> Perempuan</p> <p>Pendidikan: <input type="checkbox"/> SD <input type="checkbox"/> Sarjana S1  <input type="checkbox"/> SLTP <input type="checkbox"/> Master  <input type="checkbox"/> SLTA <input type="checkbox"/> Doktor</p> <p style="text-align: center;"> <input type="button" value="Simpan"/> <input type="button" value="Batal"/> <input type="button" value="Selesai"/> </p>	<p><b>NAVIGASI:</b></p> <ul style="list-style-type: none"> <li>- Klik 'Selesai', ganti tampilan dengan No.1.</li> <li>- Klik 'Simpan', aktifkan penyimpanan berkas dan kosongkan borang.</li> <li>- Klik 'Batal', kosongkan borang.</li> </ul>
<p><b>Keterangan:</b></p> <ul style="list-style-type: none"> <li>* Judul jendela pakai Arial, 12 point.</li> <li>* Teks pakai Arial, 10 point, hitam.</li> <li>* Medan data, Arial, 10 point, miring.</li> <li>* Teks 'push button' Arial, 12 point.</li> </ul>	

4. Jaringan Semantik Tampilan: Digunakan untuk mempermudah bagi pemrogram pada saat ia menuliskan program untuk disesuaikan dengan navigasi pada setiap lembar kerja. Pada jaringan semantik tampilan terdiri atas dua komponen: nomor tampilan (biasa diberi notasi dengan lingkaran) dan transisi yang menyebabkan perpindahan perpindahan ke tampilan yang lain (biasa diberi notasi dengan anak panah).

**Jaring Semantik Tampilan**

