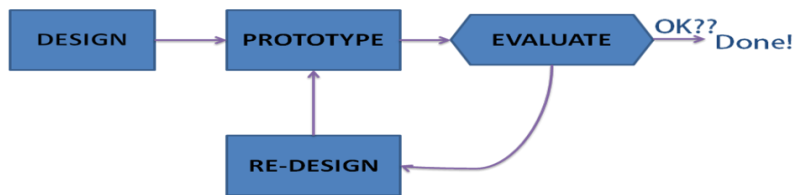


# BAB V

## PROTOTYPING

### 5.1. DEFINISI PROTOTYPE

- Proses membangun model dari suatu sistem.
- Bentuk awal (contoh) atau standar ukuran dari sebuah objek.
- Salah satu metode pengembangan perangkat lunak.



**Gambar 5.1. Prototype**

Dengan metode prototyping ini pengembang dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem.

Kunci agar model prototype ini berhasil dengan baik adalah dengan mendefinisikan aturan-aturan main pada saat awal, yaitu pelanggan dan pengembang harus setuju bahwa prototype dibangun untuk mendefinisikan kebutuhan. Prototype akan dihilangkan sebagian atau seluruhnya dan perangkat lunak aktual direalisasikan dengan kualitas dan implementasi yang sudah ditentukan

#### 5.1.1 Keuntungan dari Prototyping:

- End user* dapat berpartisipasi aktif.
- Penentuan kebutuhan lebih mudah diwujudkan.
- Mempersingkat waktu pengembangan SI.
- Waktu pengembangan bisa berkurang.
- Mengurangi biaya pengembangan.
- Membutuhkan keterlibatan pengguna.
- Pengembang menerima umpan balik pengguna terukur.
- Memfasilitasi implementasi sistem karena pengguna tahu apa yang diharapkan.
- Hasil di kepuasan pengguna yang lebih tinggi.

- j. Menghadapkan para pengembang untuk potensi perangkat tambahan sistem yang akan datang.

### **5.1.2 Beberapa Kerugian dari Prototyping:**

- a. Proses analisis dan perancangan terlalu singkat.
- b. Mengesampingkan alternatif pemecahan masalah.
- c. Biasanya kurang fleksible dalam mengahdapi perubahan.
- d. Protitype yang dihasilkan tidak selamanya mudah dirubah.
- e. Prototype terlalu cepat selesai.

### **5.1.3 Langkah-langkah pembuatan prototipe :**

- a. Permintaan bermula dari kebutuhan user.
- b. Bangunlah sistem prototipe untuk menemukan kebutuhan awal yang diminta.
- c. Biarkan user menggunakan prototipe. Analis harus memberikan pelatihan, membantu dan duduk bersama-sama dengan user, khususnya untuk pertama kali. Anjurkan perubahan. User harus melihat fungsi-fungsi dan sifat dari prototipe, lihat bagaimana ia memecahkan masalah bisnis dan mengusulkan perbaikan.
- d. Implementasikan saran-saran perubahan.
- e. Ulangi langkah ketiga sampai user merasa puas.
- f. Merancang dan membangun suatu sistem akhir seperti sebelumnya.

### **5.1.4 Alasan membuat Prototype :**

- a. Evaluasi dan umpan balik sangat penting dalam perancangan.
- b. Para Stakeholders dapat lebih mudah melihat, memegang, dan berinteraksi dengan prototype dari pada dengan dokumen atau gambar
- c. Anggota tim bisa berkomunikasi secara lebih efektif
- d. Lebih mudah mencoba ide-ide baru
- e. Mendorong pemikiran lebih dalam aspek perancangan yang sangat penting.
- f. Prototype mendukung perancang dalam memilih alternatif rancangan berlangsung.

## 5.2. RAPID PROTOTYPING

Rapid Prototyping (RP) dapat didefinisikan sebagai metode-metode yang digunakan untuk membuat model berskala (prototipe) dari mulai bagian suatu produk (part) ataupun rakitan produk (assembly) secara cepat dengan menggunakan data Computer Aided Design (CAD) tiga dimensi. Rapid Prototyping memungkinkan visualisasi suatu gambar tiga dimensi menjadi benda tiga dimensi asli yang mempunyai volume. Selain itu produk-produk rapid prototyping juga dapat digunakan untuk menguji suatu part tertentu.

### 5.2.1 Beberapa Metode Rapid Prototyping

- a. *Stereolithography* (SLA atau SL = teknologi cetak 3D yang digunakan untuk memproduksi model, prototipe, pola, dan bagian produksi
- b. *Laser Sintering Selektif* (SLS) adalah teknik manufaktur aditif yang digunakan untuk produksi rendah volume model prototipe dan komponen fungsional.
- c. *Laminated Object Manufacturing* (LOM) adalah permodelan di dalamnya, lapisan kertas perekat berlapis, plastik, atau laminasi logam berturut-turut direkatkan dan dipotong menjadi berbentuk dengan pisau atau cutter laser.
- d. *Fused Deposition Modelling* (FDM)
- e. *Solid Ground Curing* (SGC)

## 5.3. DIMENSI PROTOTYPING

- a. Penyajian / Representasi
- b. Ruang lingkup,
- c. executability dan
- d. maturation.

## 5.4. TERMINOLOGI PROTOTYPING

### 5.4.1 Prototype Horizontal

Sangat luas, mengerjakan sebagian besar interface, tetapi tidak mendalam, kemampuan sistem hanya ditampilkan sebagian.

### 5.4.2 Prototype Vertikal

Semua interface ditampilkan tetapi kemampuannya tidak ditampilkan.

- a. Early Prototyping (prototipe cepat)
- b. Late Prototyping (prototipe lambat)
- c. Low-fidelity Prototyping (prototype dengan tingkat ketepatan yang rendah).

**Contoh (1) storyboard:**

- Digunakan di awal desain.
- Biasanya digunakan dengan skenario, lebih terinci, dan dapat diputar ulang.
- Kumpulan dari sketsa/frame individual.
- menyajikan urutan inti cerita.
- menunjukkan bagaimana kemungkinan user dapat mengalami peningkatan melalui setiap aktifitas.

**Contoh (2) sketsa:**

- Sketsa sangat penting untuk low-fidelity prototyping.
- Jangan takut dengan kemampuan menggambar.
- Menyajikan “tampilan” yang kotor dan cepat dari interface, konsep desain, dll.

**Contoh (3) “wizard-of-oz”:**

- User berpikir mereka berinteraksi dengan komputer, tapi developer lebih menanggapi hasilnya daripada sistemnya.
- Biasanya dilakukan di awal desain untuk memahami apa yang diharapkan oleh user.

- d. Mid-fidelity prototyping (prototype dengan tingkat ketepatan sedang)

Contoh tools yang digunakan: **powerpoint, illustrator, dll.**

- e. High-fidelity prototyping (prototype dengan tingkat ketepatan yang tinggi).

Tools umum yang digunakan: **Macromedia Director, Visual Basic, Flash, illustrator.**

## **5.5. PROTOTYPING TOOLS**

### **5.5.1 Draw/Paint Program, contoh: Photoshop, Coreldraw**

- Menggambar setiap layar, baik untuk dilihat.
- Prototype horisontal, tipis.
- Adobe Photoshop.

Contoh :



Gambar 5.2. Tampilan Adobe Photoshop

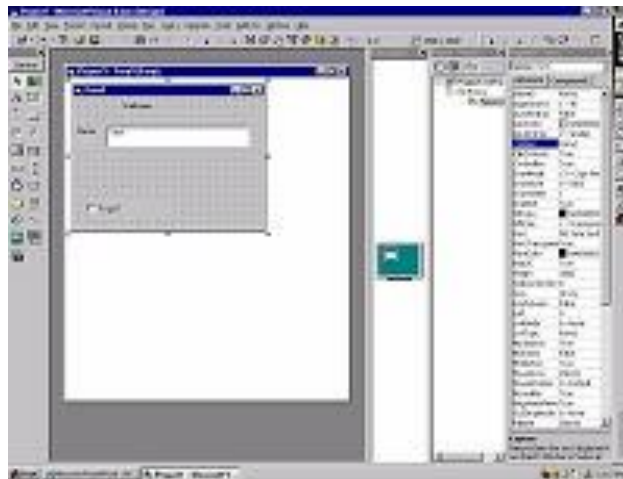
### 5.5.2 Scripted Simulations/Slide Show

Contoh: Powerpoint, Hypercard, Macromedia Director, HTML.

- Letakkan tampilan seperti storyboard dengan (animasi) perubahan diantaranya.
- Dapat memberikan user catatan yang sangat spesifik.
- Disebut *chauffeured prototyping*.
- Macromedia Director.

### 5.5.3 Interface Builders

Contoh: Visual Basic, Delphi, UIMX. Tools untuk menampilkan jendela, kendali, dan lain-lain dari interface.



Gambar 5.3. Tampilan Visual Basic

### Soal dan Pembahasan

Soal:

Sebutkan metode dalam Rapi Prototyping

Pembahasan :

*Interaksi Manusia dan Komputer / Anisya, S. Kom., M. Kom*

- a. *Stereolithography* (SLA atau SL = teknologi cetak 3D yang digunakan untuk memproduksi model, prototipe, pola, dan bagian produksi
- b. *Laser Sintering Selektif* (SLS) adalah teknik manufaktur aditif yang digunakan untuk produksi rendah volume model prototipe dan komponen fungsional.
- c. *Laminated Object Manufacturing* (LOM) adalah permodelan di dalamnya, lapisan kertas perekat berlapis, plastik, atau laminasi logam berturut-turut direkatkan dan dipotong menjadi berbentuk dengan pisau atau cutter laser.
- d. *Fused Deposition Modelling* (FDM)
- e. *Solid Ground Curing* (SGC)

# BAB VI

## DIALOG

### 6.1 DESAIN DIALOG

Dialog dalam arti umum adalah percakapan antara dua kelompok atau lebih. Sedangkan dialog dalam konteks perencanaan user interface adalah struktur dari percakapan antara user dan sistem komputer.

### 6.2 DIALOG STYLE

Ragam dialog merupakan cara yang digunakan untuk mengorganisasikan berbagai teknik dialog, sedang dialog itu sendiri merupakan “*interface*” yang digunakan oleh komputer agar manusia bisa memberikan sejumlah parameter yang akan diolah oleh komputer, yang kemudian memberikan hasil berdasarkan data yang diinputkan pada *interface* tersebut. Beberapa sifat penting yang perlu dimiliki oleh setiap ragam dialog adalah:

#### 6.2.1 Inisiatif

Inisiatif merupakan sifat dasar dari sembarang dialog, karena inisiatif akan menentukan keseluruhan ragam komunikasi sehingga dapat ditentukan tipe-tipe pengguna yang dituju oleh sistem yang dibangun. Karakteristik utamanya adalah bahwa dialog itu terdiri atas sekumpulan pilihan yang telah didefinisikan sebelumnya. Sebaliknya, inisiatif oleh pengguna mempunyai sifat keterbukaan yang lebih luas: pengguna diharapkan memahami sekumpulan perintah yang harus ditulis menurut aturan (sintaks) tertentu.

#### 6.2.2 Keluwesan

Sistem yang luwes atau fleksibel adalah sistem yang mempunyai kemampuan untuk mencapai suatu tujuan lewat sejumlah cara yang berbeda. Karakteristik penting dalam mencapai keluwesan suatu sistem adalah bahwa sistem harus dapat menyesuaikan diri dengan keinginan pengguna dan bukan pengguna harus menyesuaikan diri dengan kerangka sistem yang telah ditetapkan oleh perancang sistem. Keluwesan juga dapat dilihat dari adanya kesempatan bagi pengguna untuk melakukan *customizing* dan memperluas antarmuka dari sebuah sistem untuk memenuhi kebutuhan pribadinya.

### **6.2.3 Kompleksitas**

Seorang perancang sistem tidak perlu membuat atau menggunakan antarmuka lebih dari apa yang diperlukan, karena tidak ada keuntungan yang dapat diperoleh, malahan akan menjadikan implementasinya menjadi lebih sukar.

### **6.2.4 Kekuatan**

Kekuatan didefinisikan sebagai jumlah kerja yang dapat dilakukan oleh sistem untuk setiap perintah yang diberikan oleh pengguna.

### **6.2.5 Beban Informasi**

Agar penyampaian informasi itu dapat berdaya guna dan berhasil guna, beban informasi yang terkandung didalam suatu ragam dialog seharusnya disesuaikan dengan tingkat pengguna. Jika beban itu terlalu tinggi, pengguna akan merasa sangat terbebani yang akan berakibat negatif dalam hal kemampuan pengolahan kognitif dan tingkah laku pengguna akan merasa bahwa sistemnya seolah-olah menyembunyikan kinerja penggunanya sendiri.

### **6.2.6 Konsistensi**

Konsistensi harus diterapkan pada perancangan antarmuka pengguna. Contoh pada format pemasukan data dan format tampilan data.

### **6.2.7 Umpan Balik**

Pada program komputer yang tidak ramah, pengguna sering harus menunggu proses yang sedang berjalan, sementara pengguna tidak mengetahui status proses saat itu, apakah sedang melakukan komputasi, sedang mencetak hasil, atau bahkan komputernya macet (*hang*) karena suatu sebab. Program yang baik akan selalu memberikan umpan balik kepada pengguna atas apa yang dikerjakan saat itu.

### **6.2.8 Observabilitas**

Sistem dikatakan mempunyai sifat observabilitas apabila sistem itu berfungsi secara benar dan nampak sederhana bagi pengguna meskipun sesungguhnya pengolahan secara internalnya sangat rumit.



### 6.2.9 Kontrolabilitas

Kontrolabilitas merupakan kebalikan dari observabilitas, dan hal ini berimplikasi bahwa sistem selalu berada dibawah kontrol pengguna. Agar hal ini tidak tercapai, antarmukanya harus mempunyai sarana yang memungkinkan pengguna untuk dapat melakukan kendali.

### 6.2.10 Efisiensi

Efisiensi dalam sistem komputer yang melibatkan unjuk kerja manusia dan komputer secara bersama-sama adalah hasil yang diperoleh dari kerjasama antara manusia dan komputer. Sehingga, meskipun efisiensi dalam aspek rekayasa perangkat lunak sistem menjadi sangat penting jika mereka berpengaruh pada waktu tanggap atau laju penampilan sistem, seringkali perancang lebih memilih untuk memanfaatkan hasil teknologi baru untuk meminimalkan ongkos pengembangan sistem. Sebaliknya, tidak dapat dipungkiri bahwa biaya personal dari seorang ahli akan semakin meningkat dari waktu ke waktu.

## 6.3 KATEGORI RAGAM DIALOG

### 6.3.1 Dialog Berbasis Perintah Tunggal (*Command Line Dialogue*)

Merupakan ragam dialog yang paling konvensional. Perintah-perintah tunggal yang dapat dioperasikan biasanya tergantung dari sistem komputer yang dipakai dan berada dalam suatu *domain* yang disebut bahasa perintah (*command language*). Bahasa perintah harus dirancang sedemikian rupa sehingga mereka mempunyai sifat alamiah, yakni mudah dipelajari dan diingat oleh kebanyakan pengguna. Meskipun bersifat buatan, bahasa buatan ini tetap mempunyai struktur leksikal, sintaksis dan semantik tertentu. Berikut keuntungan dan kerugian penggunaan dialog berbasis perintah tunggal:

<b>Keuntungan</b>	<b>Kerugian</b>
<ul style="list-style-type: none"><li>- Cepat</li><li>- Efisien</li><li>- Akurat</li><li>- Ringkas</li><li>- Luwes</li><li>- Inisiatif oleh pengguna</li><li>- Appealing</li></ul>	<ul style="list-style-type: none"><li>- Memerlukan pelatihan yang lama</li><li>- Membutuhkan penggunaan yang teratur</li><li>- Beban ingatan yang tinggi</li><li>- Jelek dalam menangani kesalahan</li></ul>

### 6.3.2 Dialog Berbasis Bahasa Pemrograman (*Programming language dialogue*)

Dalam keadaan tertentu, penggunaan dialog berbasis perintah tunggal sering tidak memadai, khususnya ketika pengguna terus memberikan sederetan perintah-perintah yang sama setiap kali ia menjalankan program aplikasi tersebut. Dialog berbasis bahasa pemrograman merupakan ragam dialog yang memungkinkan pengguna untuk mengemas sejumlah perintah kedalam suatu berkas yang sering disebut *batch file*

### 6.3.3 Antarmuka Berbasis Bahasa Alami (*Natural Language Interface*)

Dalam film fiksi ilmiah kita seringkali melihat komunikasi antara manusia dengan komputer lewat suatu bahasa ucapan yang secara jelas memanfaatkan bahasa alami (*natural language*).

Berikut keuntungan dan kerugian penggunaan antarmuka berbasis bahasa alami.

Keuntungan	Kerugian
<ul style="list-style-type: none"><li>- Tidak memerlukan sintaksis</li><li>- Luwes dan powerful</li><li>- Alamiah</li><li>- Merupakan inisiatif campuran</li></ul>	<ul style="list-style-type: none"><li>- Mempunyai dualism</li><li>- Bertele-tele</li><li>- Perancangan perangkat lunak yang rumit</li><li>- Tidak efisien</li></ul>

### 6.3.4 Sistem Menu

Sistem menu merupakan pilihan yang tepat untuk menunjukkan kemampuan dan fasilitas yang dimiliki oleh sebuah program aplikasi kepada pengguna. Menu adalah daftar sejumlah pilihan dalam jumlah terbatas, yang biasanya berupa suatu kalimat atau kumpulan beberapa kata.

Ditinjau dari teknik penampilan pilihan-pilihan pada sebuah sistem menu, dikenal dua jenis sistem menu:

#### a. Sistem menu datar

Dalam sistem menu datar, kemampuan dan fasilitas yang dimiliki oleh suatu program aplikasi akan ditampilkan secara lengkap, dan biasanya menggunakan kalimat-kalimat yang cukup panjang. Contoh sistem menu datar.



**Gambar 6.1 Sistem Menu Datar**

**b. Sistem Menu Tarik (*Pulldown*) yang Berbasis pada Struktur Hirarki Pilihan (Struktur Pohon Pilihan)**

Sebuah menu tarik pada dasarnya adalah sistem menu yang pilihan-pilihannya dikelompokkan menurut kategori tertentu atau menurut cara tertentu sehingga mereka membentuk semacam hirarki pilihan. Pada hirarki paling tinggi, pilihan-pilihan itu disebut dengan pilihan/menu utama. Sebagian atau semua pilihan/menu utama dapat mempunyai salah satu atau lebih subpilihan/submenu. Sebuah subpilihan/submenu dari suatu pilihan/menu utama dapat mempunyai satu atau lebih sub-pilihan dan seterusnya. Struktur ini membentuk semacam struktur pohon.



**Gambar 6.2. Sistem Menu Tarik**

Keuntungan dan kerugian penggunaan sistem menu adalah:

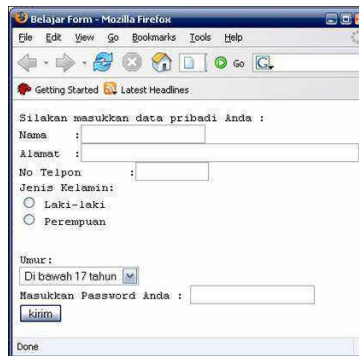
Keuntungan	Kerugian
<ul style="list-style-type: none"> <li>- Memerlukan sedikit pengetikan</li> <li>- Beban memori rendah</li> <li>- Strukturnya terdefiniskan dengan baik</li> <li>- Tersedianya piranti bantu CAD</li> </ul>	<ul style="list-style-type: none"> <li>- Seringkali lambat</li> <li>- Memakan ruang layar (khususnya untuk menu datar)</li> <li>- Tidak cocok untuk aktivitas pemasukan data</li> </ul>

	- Tidak cocok untuk dialog antar pengguna
--	---

### 6.3.5 Dialog Berbasis Pengisian Borang (*form filling dialogue*)

Teknik dialog pengisian borang (*form filling dialogue*) merupakan suatu penerapan langsung dari aktifitas pengisian borang dalam kehidupan sehari-hari dimana pengguna akan dihadapkan pada suatu bentuk borang yang ada di layar komputer yang mereka gunakan. Perlman membuat perbedaan antara menu dan borang. Menu adalah dialog yang menampilkan sejumlah alternatif pilihan yang pilihan-pilihan itu dapat dipilih pengguna dengan cara tertentu pada setiap daur aktifitas. Borang adalah tampilan dari sejumlah persyaratan (*requirement*) yang menampilkan sejumlah pilihan dan berbagai nilai parameter yang telah ditentukan dan diintegrasikan kedalam sebuah tampilan pada layar.

Contoh dialog berbasis pengisian borang



**Gambar 6.3 Dialog Berbasis Pengisian Borang**

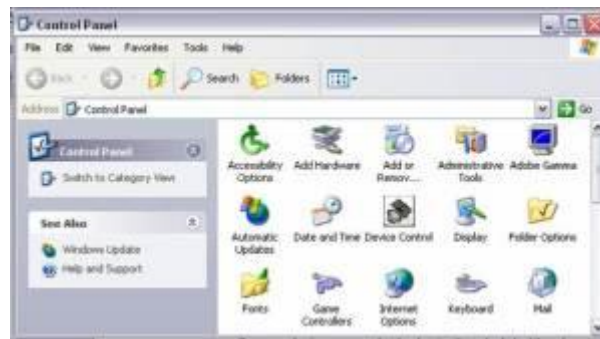
Keuntungan dan kerugian penggunaan dialog berbasis pengisian borang adalah:

<b>Keuntungan</b>	<b>Kerugian</b>
<ul style="list-style-type: none"> <li>- Pengguna sudah terbiasa dengan pengisian boring</li> <li>- Isian data yang disederhanakan</li> <li>- Diperlukan sedikit pelatihan</li> <li>- Beban memori rendah</li> <li>- Perancangan yang mudah</li> <li>- Tersedianya berbagai peranti bantu perancangan tampilan</li> </ul>	<ul style="list-style-type: none"> <li>- Seringkali lambat</li> <li>- Memakan ruang layar (khususnya untuk menu datar)</li> <li>- Tidak cocok untuk pemilihan perintah / instruksi</li> <li>- Memerlukan pengontrol kursor</li> <li>- Mekanisme pengontrol kursor</li> <li>- Mekanisme navigasinya tidak terlihat secara eksplisit</li> <li>- Memerlukan suatu bentuk pelatihan</li> </ul>

### 6.3.6 Antarmuka Berbasis Icon

Antarmuka sering memanfaatkan simbol-simbol dan tanda-tanda dari kehidupan kita sehari-hari untuk memberitahukan pengguna akan kemampuan dan fasilitas yang dimiliki oleh suatu program aplikasi. Ragam dialog yang banyak menggunakan simbol-simbol dan tanda-tanda untuk menunjukkan suatu aktifitas tertentu disebut dengan antarmuka berbasis ikon (*icon based user interface*).

Contoh dialog berbasis ikon pada fasilitas Control Panel yang ada di dalam Microsoft Windows XP Profesional.



**Gambar 6.4. Antarmuka Berbasis Ikon**

### 6.3.7 Sistem Penjendelaan (*Windowing system*)

Secara umum yang disebut dengan jendela (*window*) adalah bagian dari layar yang digunakan untuk menampilkan suatu informasi. Informasi disini dapat berupa informasi tekstual maupun grafis. Untuk menunjukkan daerah yang digunakan untuk menampilkan suatu informasi biasanya daerah yang umumnya berupa persegi panjang dibatasi dengan semacam garis pembatas dengan ketebalan tertentu. Sistem penjendelaan adalah sistem antarmuka yang memungkinkan pengguna untuk menampilkan berbagai informasi baik sendiri-sendiri maupun secara bersama-sama ke dalam bagian-bagian layar yang tidak saling mempengaruhi.

### 6.3.8 Manipulasi Langsung (*Direct Manipulation*)

Karakteristik yang sangat penting dari ragam dialog ini adalah adanya penyajian langsung suatu aktifitas oleh sistem kepada pengguna sehingga aktifitas itu akan dikerjakan oleh sistem komputer ketika pengguna memberikan instruksi lewat manipulasi langsung dari semacam kenyataan maya (*virtual reality*) yang terpampang lewat tampilan yang muncul dilayar. Penerapan manipulasi langsung pada berbagai bidang diantaranya:

### a. Kontrol Proses

Kontrol proses didalam berbagai industri, misalnya pada pembangkitan dan penyaluran listrik dan industri makanan berskala besar biasanya memanfaatkan tampilan visual yang berupa panel-panel kontrol yang dihubungkan kesuatu sistem pengontrol berbasis komputer.

### b. Editor Teks

Konsep WYSIWYG (*What You See Is What You Get*) merupakan fenomena pengolahan kata modern yang banyak sekali memberikan kemudahan pada pengguna. Pengguna akan mendapatkan hasil cetakan seperti apa yang muncul pada layar tampilan. Sehingga, ketika pengguna melakukan manipulasi ke atas suatu teks pada layar, maka sebenarnya ia juga melakukan manipulasi atas hasil cetakan yang ia harapkan.

### c. Simulator

Simulator merupakan sistem miniatur yang mencoba menirukan kerja suatu sistem yang berskala sangat besar atau sangat kecil jika dilihat dari kaca mata orang awam. Misalnya pada simulator penerbangan. Didalam simulator penerbangan, seorang calon pilot seolah-olah sedang berada didalam sebuah pesawat yang menjadi tanggung jawab secara penuh.

### d. Kontrol Lalu Lintas Penerbangan

Sistem radar berpegang pada adanya kemampuan untuk mengenal pola bentuk-bentuk tertentu seperti yang dapat dikerjakan oleh manusia. Dalam hal ini, dunia tiga dimensi tempat sebuah pesawat sedang menjelajahi dunia ini akan diterjemahkan ke dalam layar dua dimensi yang ada di hadapan seorang operator. Lewat tampilan dua dimensi, yang merefleksikan dunia tiga dimensi itulah seorang operator dapat mengontrol lalu lintas penerbangan

### e. Perancangan Bentuk/Model (Computer Aided Design)

Saat sekarang kita dapat meliha berbagai program aplikasi untuk perancangan suatu sistem. Contoh yang pali populer barangkali adalah sebuah program yang bernama engan Auto CAD. Dengan program ini kita dapat merancang suatu model pesawa terbang, baik dengan tampilan yang disebut denagn wireframe model maupun berujut tampilan seperti sebuah pesawat yang sesungguhnya dengan memanfaatkan fasilitas rendering yang dimilikinya. Program ini juga serin dimanfaatkan untuk melakukan pemetaan berbasis komputer (*computer-based mapping*)

Keuntungan dan kerugian dialog berbasis manipulasi langsung

Keuntungan	Kerugian
- Mempunyai analogi yang jelas dengan suatu	- Memerlukan program yang rumit dan berukuran

pekerjaan nyata <ul style="list-style-type: none"> <li>- Mengurangi waktu pembelajaran</li> <li>- Memberikan tantangan untuk eksplorasi pekerjaan yang nyata</li> <li>- Penampilan visual yang bagus</li> <li>- Mudah dioperasikan</li> <li>- Tersedianya berbagai perangkat bantu untuk merancang ragam dialog manipulasi langsung</li> </ul>	besar <ul style="list-style-type: none"> <li>- Memerlukan tampilan grafis berkinerja tinggi</li> <li>- Memerlukan peranti masukan seperti mouse, trackball</li> <li>- Memerlukan perancangan tampilan dengan kualifikasi tertentu</li> </ul>
--	--

### 6.3.9 Antarmuka Berbasis Interaksi Grafis.

Secara umum cukup sulit membedakan antarmuka berbasis manipulasi langsung dengan antarmuka berbasis interaksi grafis. Pada program-program yang berkemampuan *hypertext*, program akan memberitahu pengguna bahwa pengguna dapat melakukan *links* ke teks atau dokumen yang lain. Ketika kursor *mouse* berada pada teks yang mempunyai *link* ke teks lain, maka bentuk kursor biasanya berubah (umumnya berubah menjadi bentuk tangan menunjuk). Keuntungan dan kerugian teknik antarmuka berbasis interaksi grafis sama dengan keuntungan dan kerugian teknik antarmuka menggunakan manipulasi langsung.

## 6.4 USER INTERFACE SOFTWARE

*Interface* adalah sebuah titik, wilayah, atau permukaan di mana dua zat atau benda berbeda bertemu; dia juga digunakan secara metafora untuk perbatasan antara benda. Kata *interface* kadangkala (biasanya dalam bidang teknik) disingkat menjadi “i/f”. Bentuk kerja dari *interface* berarti menghubungkan dua atau lebih benda pada suatu titik atau batasan yang terbagi, atau untuk menyiapkan kedua benda untuk tujuan tersebut. Kata *interface* juga memiliki arti khusus, yaitu: antarmuka pengguna adalah fungsi dan atribut sensor dari suatu sistem (aplikasi, perangkat lunak, kendaraan, dll) yang berhubungan dengan pengoperasiannya oleh pengguna.

Antarmuka pemakai (*User Interface*) merupakan mekanisme komunikasi antara pengguna (user) dengan sistem. Antarmuka pemakai (*User Interface*) dapat menerima informasi dari pengguna (user) dan memberikan informasi kepada pengguna (user) untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan suatu solusi.

*User Interface*, berfungsi untuk menginputkan pengetahuan baru ke dalam basis pengetahuan sistem pakar (ES), menampilkan penjelasan sistem dan memberikan panduan

pemakaian sistem secara menyeluruh step by step sehingga user mengerti apa yang akan dilakukan terhadap suatu sistem. Yang terpenting dalam membangun user interface adalah kemudahan dalam memakai/ menjalankan sistem, interaktif.

### **Soal dan Pembahasan**

#### **Soal :**

Jelaskan maksud dari ragam dialog.

#### **Pembahasan :**

Merupakan cara yang digunakan untuk mengorganisasikan berbagai teknik dialog, sedang dialog itu sendiri merupakan “*interface*” yang digunakan oleh komputer agar manusia bisa memberikan sejumlah parameter yang akan diolah oleh komputer, yang kemudian memberikan hasil berdasarkan data yang diinputkan pada *interface* tersebut