BAB II
DASAR TEORI

Pada bab ini berisi teori-teori yang mendukung dalam pembuatan Profil SMAN 2 Malang.

2.1 Gambaran Umum SMAN 2 Malang

SMA Negeri 2 Malang merupakan salah satu Sekolah Menengah Atas Negeri yang ada di Kota Malang, Provinsi Jawa Timur, Indonesia yang berrtempat di Jl. Laks. Martadinata No. 84 Malang.

SMA Negeri 2 Malang saat ini telah mengalami perkembangan yang sangat pesat, hal ini ditandai dengan adanya sarana dan prasarana yang sangat memadai untuk kegiatan belajar mengajar. Dengan jumlah murid yang mencapai 920-an setiap tahunnya, ditunjang dengan tenaga pengajar yang memadai, yaitu 78 guru dengan kualifikasi 59 orang guru tetap (PNS) dan 19 guru tidak tetap. Dari jumlah tenaga pengajar tersebut 5 orang Guru berijazah S2, dan 73 orang guru berijazah S1 membuat SMAN 2 Malang menjadi salah satu sekolah terbaik di Kota Malang.

Fasilitas yang memadai juga merupakan sarana yang mutlak harus dimiliki untuk meningkatkan kualitas pembelajaran, untuk itu saat ini SMAN 2 Malang telah memiliki 27 ruang kelas, 1 ruang lab bahasa, 1 ruang Lab Fisika, 1 ruang Lab. Kimia,1 Ruang lab. Biologi, 1 ruang lab multimedia, 1 ruang lab agama, 2 ruang lab komputer dan 1 ruang perpustakaan. Sarana penunjang lain : 1 ruang Ekstra kurikuler, 1 Ruang UKS, Ruang kepala Sekolah, ruang Waka, ruang KOPSIS, ruang KOSMA, ruang OSIS, POSKO GESANK, Tata Usaha, Ruang Tatib dan LITBANG, Ruang BK, gedung AULA, Mushola, Kamar Mandi Siswa, kamar mandi guru, Kantin, Tilpun Umum, Fotocopy dan GAZEBO, Lapangan Basket, 2 Lapangan Bola volley, Tenis Meja.

Baru-baru ini SMAN 2 Malang telah membangun sebuah front office yang digunakan untuk menerima tamu. Kegunaan dari front office ini adalah sebagai

tempat menerima tamu yang berkunjung ke SMAN 2 Malang. Para tamu pada umumnya ingin mengetahui profil SMAN 2 Malang.

Berangkat dari sana, company profile ini sangatlah dibutuhkan untuk mempermudah penjelasan tentang profil SMAN 2 Malang. Aplikasi ini diletakkan pada front office dimana ruang ini merupakan tempat pertama yang dikunjungi oleh para tamu. Dari aplikasi ini, pengunjung atau tamu dapat melihat secara lengkap profil SMAN 2 Malang yang meliputi : motto simbolis, visi dan misi, sejarah, mars dan hymne, profil guru, prestasi, smanda cup, ekstrakurikuler, album foto siswa tiap kelas, denah sekolah dan deskripsi setiap ruang, serta denah 3D yang akan memudahkan pengunjung "berjalan mengelilingi sekolah" hanya dari aplikasi ini.

2.2 Profil

Profil merupakan penjelasan mengenai sekolah termasuk produknya secara verbal maupun grafik yang mengangkat *image* sekolah serta keunggulannya dibandingkan pesaing berdasarkan *image* diatas. [KHU-08]. Nilai-nilai yang ada pada SMAN 2 Malang tercemin dalam beberapa hal berikut :

1. Sejarah Berdirinya SMAN 2 Malang

Sejarah berdirinya SMAN 2 Malang menggambarkan kepada pihak-pihak lain yang berhubungan dengan perusahaan ataupun kosumen mengenai dasar atau landasan usaha ini berdiri apakah cukup kuat secara pengalaman dan keutuhan individu yang terlibat didalamnya.

2. Visi & Misi SMAN 2 Malang

Visi merupakan cita-cita yang ingin dicapai oleh usaha kita dalam jangka panjang atau dengan kata lain perusahaan dalam periode tertentu ingin menjadi sekolah yang seperti apa.

Misi merupakan cara-cara yang digunakan usaha dalam mencapai visi usaha. Misi dapat berupa pernyataan kalimat atau kata yang mengingatkan pelaku usaha untuk bekerja sesuai Misi dalam mencapai tujuan sekolah.

3. Struktur Organisasi

Struktur organisasi berisi susunan/hirarki tanggung jawab pekerjaan dalam

perusahaan berikut nama individu pada masing-masing pekerjaan. Kegunaan struktur organisasi dalam *company profile* adalah agar konsumen atau pihak-pihak lain yang bekerjasama dengan perusahaan dapat mengetahui *person in charge* yang langsung berhubungan dengan mereka dalam pekerjaan ataupun masalah.

2.3 Multimedia

Gambar 2.1 Multimedia dalam simbol
Sumber:[SUY-09]

Multimedia adalah media yang menggabungkan dua unsur atau lebih media yang terdiri dari teks, grafik, gambar, foto, audio, dan animasi secara terintegrasi. Multimedia terbagi menjadi dua kategori, yaitu: multimedia linear, dan multimedia interaktif. Simbolisasi Multimedia dapat dilihat pada Gambar 2.1. Multimedia linear adalah suatu multimedia yang tidak dilengkapi dengan alat pengontrol apapun yang dapat dioperasikan oleh pengguna. Multimedia ini berjalan sekuensial (berurutan), contohnya TV dan film.

Multimedia interaktif adalah suatu multimedia yang dilengkapi dengan alat pengontrol yang dapat dioperasikan oleh pengguna, sehingga pengguna dapat memilih apa yang dikehendaki untuk proses selanjutnya. Contoh multimedia interaktif adalah: multimedia pembelajaran interaktif, aplikasi *game* [IST-11].

Dalam definisi ini terkandung empat komponen penting multimedia. Pertama, harus ada komputer yang mengkoordinasi apa yang dilihat dan didengar yang berinteraksi dengan kita. Kedua, harus ada link yang menghubungkan kita dengan informasi. Ketiga, harus ada alat navigasi yang memandu kita, menjelajah

jaringan informasi yang saling terhubung. Keempat, multimedia menyediakan tempat kepada kita untuk mengumpulkan, memproses, dan mengkomunikasikan informasi dan ide kita sendiri. Jika salah satu komponen tidak ada, maka bukan multimedia dalam arti luas namanya.

Misalnya jika tidak ada komputer untuk berinteraksi maka itu namanya media campuran, bukan multimedia. Jika tidak ada link yang menghadirkan sebuah struktur dan dimensi, maka namanya rak buku, bukan multimedia. Kalau tidak ada navigasi yang memungkinkan kita memilih jalannya suatu tindakan maka itu namanya film, bukan multimedia. Demikian juga jika kita tidak mempunyai ruang untuk berkreasi dan menyumbangkan ide sendiri, maka namanya televisi, bukan multimedia. Dari definisi diatas, maka multimedia ada yang *online* (internet) dan multimedia yang *offline* (tradisional).

Unsur – Unsur Multimedia

Ada beberapa unsur-unsur pendukung multimedia. Beberapa unsur tersebut sebagai berikut [SUY-09] :

1. Teks

Bentuk data multimedia yang paling mudah disimpan dan dikendalikan adalah teks. Teks merupakan yang paling dekat dengan kita dan yang paling banyak kita lihat. Teks dapat membentuk kata, surat atau narasi dalam multimedia yang menyajikan bahasa kita. Kebutuhan teks tergantung pada kegunaan aplikasi multimedia. Secara umum ada empat macam teks yaitu teks cetak, teks hasil scan, teks elektronis dan hyperteks .

2. Grafis

Alasan untuk menggunakan gambar dalam presentasi atau publikasi multimedia adalah karena lebih menarik perhatian dan dapat mengurangi kebosanan dibandingkan dengan teks. Gambar dapat meringkas dan menyajikan data kompleks dengan cara yang baru dan lebih berguna. Sering dikatakan bahwa sebuah gambar mampu menyajikan seribu kata. Tapi ini berlaku hanya ketika kita biasa menampilkan gambar yang diinginkan saat kita memerlukannya.

Multimedia membatu kita melakukan hal ini, yakni ketika gambar grafis menjadi objek suatu link. Grafis sering kali muncul sebagai backdrop (latar belakang) suatu teks untuk menghadirkan kerangka yang mempermanis teks. Secara umum ada lima macam gambar atau grafik yaitu gambar vektor (*vector image*), gambar *bitmap* (*bitmap image*), *clip art*, *digitized picture* dan *hyperpicture*.

3. Bunyi atau *Sound*

Bunyi atau *sound* dalam komputer multimedia, khusunya pada aplikasi bidang bisnis dan game sangat bermanfaat. Komputer multimedia tanpa bunyi hanya disebut *unimedia*, bukan multimedia. Bunyi atau *sound* dapat kita tambahkan dalam produksi multimedia melalui suara, musik dan efek-efek suara. Seperti halnya pada grafik, kita dapat membeli koleksi *sound* disamping juga menciptakan sendiri. Beberapa jenis objek bunyi yang biasa digunakan dalam produksi *multimedia* yakni format *waveform audio*, *compact disk audio*, MIDI *sound track* dan mp3.

4. Video

Video adalah rekaman gambar hidup atau gambar bergerak yang saling berurutan. Terdapat dua macam video yaitu video analog dan video digital. Video analog dibentuk dari deretan sinyal elektrik (gelombang analog) yang direkam oleh kamera dan dipancarluaskan melalui gelombang udara. Sedangkan video digital dibentuk dari sederetan sinyal digital yang berbentuk yang menggambarkan titik sebagai rangkaian nilai minimum atau maksimum, nilai minimum berarti 0 dan nilai maksimum berarti. Terdapat tiga komponen utama yang membentuk video digital yaitu *frame rate*, *frame size* dan *data type*. *Frame rate* menggambarkan berapa kali bingkai gambar muncul setiap detiknya, sementara *frame size* merupakan ukuran fisik sebenarnya dari setiap bingkai gambar dan *data type* menentukan seberapa banyak perbedaan warna yang dapat muncul pada saat bersamaan [SUY-09].

2.4 Animasi

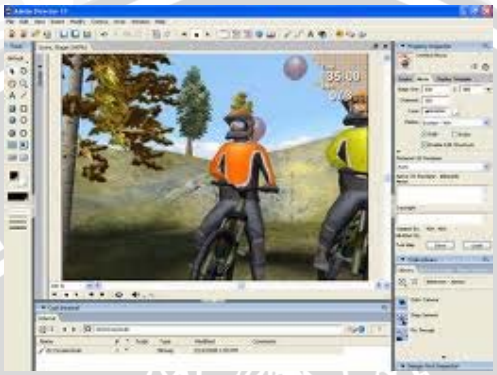
Kata animasi diambil dari kata *ANIMATION; TO ANIMATE*, dan apabila kita lihat dalam kamus bahasa inggris – indonesia artinya kurang lebih adalah hidup atau menghidupkan [SUY-09]. Animasi merupakan suatu teknik menampilkan gambar berurut sedemikian rupa sehingga penonton merasakan adanya ilusi gerakan (*motion*) pada gambar yang ditampilkan. Secara umum ilusi gerakan merupakan perubahan yang dideteksi secara visual oleh mata penonton sehingga tidak harus perubahan yang terjadi merupakan perubahan posisi sebagai makna dari istilah 'gerakan'. Perubahan seperti perubahan warna pun dapat dikatakan sebuah animasi [MAT-01].

Dalam bidang grafika pemodelan visual dapat dikategorikan sebagai dua kelompok yaitu pemodelan geometrik dan pemodelan penampilan (*appearance*). Pemodelan geometrik merupakan representasi dari bentuk objek yang ingin ditampilkan sedangkan pemodelan penampilan membuat representasi sifat visual atau penampakan objek tersebut. Contoh sifat *visual* diantaranya warna dan tekstur.

Berdasarkan definisi animasi di atas bahwa sebuah animasi disusun oleh himpunan gambar yang ditampilkan secara berurut maka animasi dapat dikatakan sebuah fungsi terhadap waktu. Gambar dapat didefinisikan sebagai koleksi deskripsi geometris dan *visual* ataupun dapat berupa citra. Pada gambar yang merupakan koleksi deskripsi, maka animasi didefinisikan sebagai fungsi yang memetakan waktu kepada perubahan parameter-parameter dari deskripsi. Pada gambar yang merupakan citra, animasi didefinisikan sebagai fungsi yang memetakan waktu kepada tiap elemen citra. Jika citra didefinisikan sebagai fungsi *domain* spasial maka animasi merupakan komposisi fungsi yang memetakan waktu kepada nilai tiap elemen citra (*pixel*).

Untuk membuat berbagai macam animasi, dibutuhkan *software-software* pendukung seperti :

2.4.1 *Adobe Director*



Gambar 2.2 Tampilan *Adobe Director*

Adobe Director adalah perangkat yang dibuat oleh *Macromedia* dan sekarang merupakan bagian dari *Adobe Systems*, yang memperbolehkan pengguna untuk membangun perangkat lunak dalam film metaphor, yang biasa digunakan untuk pembuatan : cd interaktif, edukasi (pembelajaran), katalog produk, *information kiosk, interface, game*, dan presentasi.

Istilah *User Interface* Director

Seperti pada Gambar 2.2, ada beberapa istilah pada *user interface* di Adobe Director, diantaranya sebagai berikut [HEN-08] :

1. *Stage*

Adalah tampilan untuk menunjukkan hasil tata letak objek pada waktu (*frame*) tertentu. Analoginya seperti tampilan di layar TV .

2. *Score*

Digunakan untuk mengatur urutan objek yang akan tampil agar sesuai cerita/naskah. Analoginya seperti *storyboard* dan *storyline*. Di score inilah kita menentukan mana yang akan ditampilkan terlebih dahulu dan mana yang belakangan.

3. *Cast Member*

Digunakan untuk menampung objek apa saja yang siap dan bisa ditampilkan.

4. *Panel Property Inspector*

Digunakan untuk mengatur sifat/parameter yang ada pada objek. Setiap objek mempunyai keistimewaan sendiri.

Seperti namanya, Adobe Director adalah *software* yang fungsinya adalah sebagai sutradara dalam pembuatan proyek multimedia. Dia dapat mengikutsertakan/mengimport pemain-pemain yang berbeda karakter untuk proyek multimedia, seperti : Photoshop, 3D Studio Max, MP3 dan file *audio* lain, Flash, serta *Video*.

Kelebihan Adobe Director adalah mampu mengimport format seperti :

1. *Movie* dengan format VCD, DVD, AVI, MPG/DAT, MOV, VOB, GIF.
2. *Bitmap* : PSD (Photoshop), JPG, GIF, PNG
3. *Vector* : AI (Adobe Ilustrator), SWF (Flash)
4. 3 Dimensi : W3D Shockwave 3D
5. Audio : WAV, MP3, MIDI

2.4.2 *Adobe Flash*

Gambar 2.3 Contoh Animasi dalam *Adobe Flash*

Adobe Flash adalah salah satu perangkat lunak komputer yang merupakan produk unggulan Adobe Systems. Adobe Flash digunakan untuk membuat gambar

vektor maupun animasi gambar tersebut, seperti contoh animasi pada Gambar 2.3. Berkas yang dihasilkan dari perangkat lunak ini mempunyai *file extension* .swf dan dapat diputar di penjelajah *web* yang telah dipasangi *Adobe Flash Player*. Flash menggunakan bahasa pemrograman bernama *ActionScript* yang muncul pertama kalinya pada Flash 5.

Versi terakhir yang diluncurkan di pasaran dengan menggunakan nama 'Macromedia' adalah adalah Macromedia Flash 8. Pada tahun 2005 Adobe Systems mengakuisisi Macromedia dan seluruh produknya, sehingga nama Macromedia Flash berubah menjadi Adobe Flash [HEN-08].

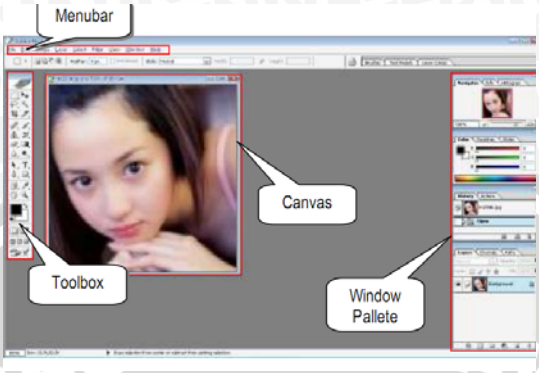
Kelebihan dari Adobe Flash:

1. Dapat membuat tombol interaktif dengan sebuah movie atau objek lain.
2. Dapat membuat perubahan transparansi warna dalam *movie*.
3. Dapat membuat perubahan animasi dari satu bentuk ke bentuk lain.
4. Dapat membuat gerakan animasi dengan mengikuti alur yang telah ditetapkan.
5. Dapat dikonversi dan dipublikasikan ke dalam beberapa tipe, diantaranya adalah .swf, .html, .gif, .jpg, .png, .exe, .mov.

2.4.3 *Adobe Photoshop*

Adobe Photoshop adalah perangkat lunak *editor* citra buatan *Adobe Systems* yang dikhususkan untuk pengeditan foto/gambar dan pembuatan efek. Perangkat lunak ini banyak digunakan oleh fotografer digital dan perusahaan iklan sehingga dianggap sebagai pemimpin pasar (market leader) untuk perangkat lunak pengolah gambar [HDM-08]. Photoshop merupakan salah satu software yang berguna untuk mengolah gambar berbasis *bitmap*, yang mempunyai *tool* dan efek yang lengkap sehingga dapat menghasilkan gambar atau foto yang berkualitas tinggi. Kelengkapan fitur yang ada di dalam Photoshop inilah yang akhirnya membuat *software* ini banyak digunakan oleh desainer grafis profesional. Dan mungkin juga sampai saat ini masih belum ada *software* desain grafis lain yang bisa menyamai kelengkapan fitur dalam Photoshop.

Bagian-bagian *Adobe Photoshop* :



Gambar 2.4 Bagian-bagian *Adobe Photoshop*

Berikut penjelasan dari Gambar 2.4:

1. *Menu Bar*

Menu bar adalah *menu pulldown* yang berisi perintah-perintah dalam photoshop seperti menu *File, Edit, Image, Layer, Select, Filter, View, Window,* dan *Help*.

2. *Toolbox*

Toolbox adalah alat-alat yang digunakan untuk memodifikasi *image* (gambar atau foto). Alat-alat ini juga dikelompokkan menurut jenisnya.

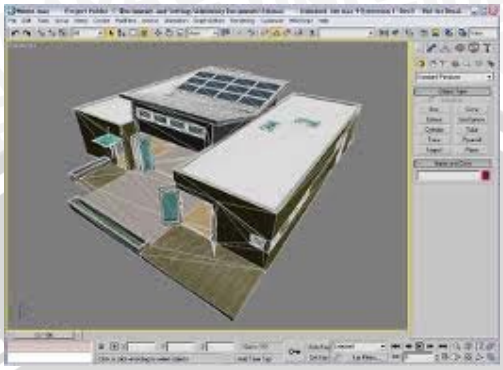
3. *Canvas*

Canvas adalah bidang yang digunakan sebagai tempat untuk meletakkan image. Biasanya ukuran canvas akan sama dengan ukuran image, tetapi dalam photoshop kita dapat merubah ukuran *canvas* dan *image* sesuai dengan kebutuhan. Kalau kita memunculkan canvas baru biasanya ada tiga pilihan yaitu *canvas* yang putih, berwarna dan transparan.

4. *Window Pallete*

Window pallete adalah *window* yang berguna untuk memilih atau mengatur berbagai parameter pada saat menyunting *image* dalam Photoshop. Untuk menampilkan *Window Pallete* dapat kita lakukan dengan cara memilih menu *Window* kemudian pilih *pallete* yang dimunculkan.

2.4.4 3D *Studio Max*



Gambar 2.5 Tampilan *3D's Max*

Sebuah perangkat lunak grafik vektor 3-dimensi dan animasi, ditulis oleh Autodesk Media & Entertainment (dulunya dikenal sebagai Discreet and Kinetix. Perangkat lunak ini dikembangkan dari pendahulunya 3D Studio fo DOS, tetapi untuk platform Win32. Kinetix kemudian bergabung dengan akuisisi terakhir Autodesk, Discreet Logic. Versi terbaru 3Ds Max pada Juli 2005 adalah 7. 3ds Max adalah salah satu paket perangkat lunak yang paling luas digunakan sekarang ini, karena beberapa alasan seperti penggunaan platform Microsoft Windows, kemampuan mengedit yang serba bisa, dan arsitektur plugin yang banyak. Tampilan awal dari 3D's Max dapat dilihat pada Gambar 2.5.

Ada 5 metode pemodelan dasar :

1. Pemodelan dengan primitif
2. NURMS (*subdivision surface*)
3. *Surface Tool*
4. NURBS
5. Pemodelan Polygon

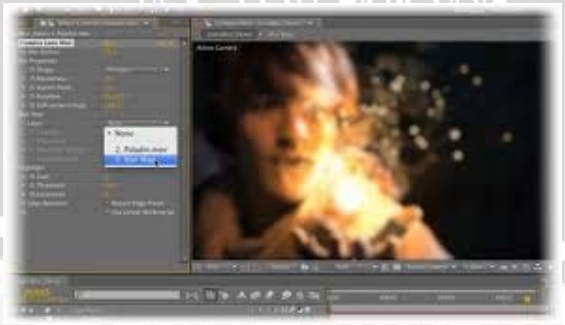
Pemodelan dengan primitif merupakan metode dasar, di mana seseorang membentuk model dengan menggunakan banyak kotak, bola, *cone*, silinder, dan objek yang telah disediakan lainnya. Seseorang juga dapat menerapkan operasi

boolean, termasuk pengurangan, pemotongan, dan penggabungan. Misalnya, seseorang dapat membuat dua bola yang dapat bekerja sebagai *blob* yang akan menyatu. Hal ini disebut “pemodelan balon”.

Mental Ray merupakan sebuah *render engine* (mesin untuk merender gambar atau video) yang terdapat pada program 3D Studio Max, selain render standar max yaitu ‘*Default Scanline’*. *Mental Ray* terintegrasi dengan 3D Studio Max sehingga tidak perlu menginstal secara terpisah. Mental ray mempunyai beberapa kelebihan yaitu dapat mengkalkulasi efek *Global Illumination* dan *Indirect Illumination*, selain itu dapat juga menggunakan *shader* pada permukaan gambar atau cahaya. *Render engine* lain selain Mental Ray adalah V-Ray, Brazil R/S, *Maxwell Render*, *Final Render*, dan sebagainya. Semua *render engine* ini memiliki kelebihannya masing-masing. yang terbaru dari 3dsmax adalah 3ds max 9.

AutoDesk 3ds Max adalah mampu dan berkuasa untuk 3D animasi, *rendering* dan *modeling*. Walaupun sasaran utamanya di digital untuk pembuatan film dan permainan, ia tetap populer untuk *photorealistic rendering* dari arsitektur dan desain produk. Aplikasi dengan lebih mendalam sebagai 3ds Max adalah baik dijelaskan di *review* ini, namun beberapa fitur dalam rilis menonjol [HDI-08].

2.4.5 *After Effect*



Gambar 2.6 Contoh Efek dalam *Adobe After Effects*

After effects adalah sebuah *software* yang sangat profesional untuk kebutuhan *Motion Graphic Design*. Dengan perpaduan dari bermacam - macam *software design* yang telah ada, After Effects menjadi salah satu *software* Design yang handal. Seperti terlihat pada Gambar 2.6, *Standart Effects* yang mencapai sekitar 50 macam lebih, bisa digunakan untuk merubah dan menganimasikan obyek. Disamping itu, membuat animasi dengan After Effects, juga bisa dilakukan dengan hanya mengetikkan beberapa kode *script* yang biasa disebut *Expression* untuk menghasil pergerakan yang lebih dinamis.

After Effects lebih lengkap fasilitasnya bila kita bandingkan dengan software Video Editing lain. Pada *After Effects*, terdapat beberapa fasilitas yang dimiliki oleh beberapa *software* lain. Misalnya; Di *After Effects* terdapat tool untuk membuat *Shape* (seperti yang terdapat pada *Photoshop*). Pada *After Effects* terdapat *Keyframe* seperti yang terdapat pada Flash (cara menganimasikannya juga hampir sama). Terdapat juga *Expression* yang hampir mirip dengan *Action Script* pada Flash, dan masih banyak lagi yang lain.

2.4.6 *Pinnacle Studio*

Gambar 2.7 Mengatur Komposisi Video dalam *Pinnacle Studio*

Pinnacle Studio adalah aplikasi yang digunakan untuk membuat film (*movie*) dalam berbagai format *file* dan memberi kemudahan bagi penggunanya untuk memberi berbagai efek menarik pada film, menambah teks, mengatur efek suara pada film dan berbagai kegiatan seperti terlihat pada Gambar 2.7.

Tiga Proses Utama dalam *Pinnacle Studio* :

1. *Capture*

Kegiatan mengimpor video ke komputer, dapat berasal dari video *digital* atau *analog*. Contoh video *digital* adalah DV (*Digital Video*) Camcoder, Digital 8 Camcoder, HDV (*High Digital Video*) Camcoder dan lain-lain. Sedangkan video *analog* misalnya VCR (*Video Cassette Recorder*) atau kamera video yang menggunakan media penyimpanan VHS (*Video Home System*).

2. *Edit*

Kegiatan memanipulasi materi video yang ada untuk kemudahan dijadikan sebuah film. Misalnya mengatur transisi antar *scene*, memberi teks, menambah gambar, menambah suara, dll.

3. *Make Movie*

Kegiatan untuk mewujudkan sebuah film dan materi-materi video yang telah diedit. Film tersebut dapat dibuat pada berbagai format file, antara lain : VCD, S-VCD, DVD, AVI, MPEG, Real Video, atau Windows Media.

2.5 Pembaruan Data

Profil ini memiliki fasilitas pembaruan data pada bagian tertentu yang dapat digunakan oleh admin untuk memperbarui isi dari aplikasi ini. Adapun cara yang digunakan agar aplikasi ini adalah memiliki fasilitas pembaruan data melalui *Adobe Director* menggunakan *Arca database xtras.*

Caranya adalah dengan memberi tombol *edit* pada setiap gambar yang ingin diubah, tombol *edit* tersebut akan mengakses halaman "*edit* data" yang sudah kami persiapkan sebelumnya. Ketika data sudah diisikan, dan tombol *browse* ditekan maka akan keluar "*open dialog*" yang berfungsi untuk memilih data mana yang akan digunakan untuk mengganti data terpilih dalam *Cast Member* di Adobe Director. Sehingga, secara otomatis data pada *Stage* akan berubah. Alur: Data > Tombol *Edit* > Halaman *Password* > Open Dialog > Data Update.

2.6 Metode Penelitian

2.6.1 Metode *Research and Development* (R&D)

Menurut [BOR-89], aplikasi berbasis multimedia dapat dikembangkan dengan Metode *Research and Development* (R&D) yang didalamnya terdapat 10 tahap, yaitu:

1. Pengumpulan Data Awal.

Dalam tahap ini dilakukan identifikasi perkiraan kebutuhan dan studi literatur.

2. Perencanaan.

Setelah mempelajari literatur selengkapnya dan memperoleh informasi yang diperlukan, langkah selanjutnya adalah merencanakan pembuatan produk.

3. Pembuatan Produk Awal.

Setelah inisiasi dalam perencanaan lengkap, langkah utama dalam tahapan R & D adalah membuat bentuk awal produk pembelajaran yang dapat diuji coba. Dalam tahap pengembangan produk ini termasuk pembuatan instrumen untuk mendapatkan umpan balik dari pengguna.

4. Uji Coba Awal

Setelah produk awal selesai dilakukan uji coba awal yaitu evaluasi teman sejawat dan evaluasi pakar. Evaluasi awal ini akan dilakukan teman sejawat dan pakar yang berkompeten dan berpengalaman dalam bidang bidang ilmu komputer, desain komunikasi visual, dan teknik informatika.

5. Perbaikan Produk Awal

Setelah dilakukan uji coba awal, tahap berikutnya adalah perbaikan produk sesuai dengan data yang diperoleh dari uji coba awal. Saran dari pakar digunakan untuk menyempurnakan produk. Dalam tahap ini juga akan dijelaskan apa saja komentar dan saran yang dilakukan oleh teman sejawat dan pakar.

6. Uji Coba Lapangan

Setelah produk awal diperbaiki sesuai dengan saran dari teman sejawat dan pakar ilmu komputer, desain komunikasi visual, dan teknik informatika, dilaksanakan uji coba lapangan untuk mendapatkan evaluasi atas produk.

7. Perbaikan Produk

Setelah dilakukan uji coba awal, tahap berikutnya adalah perbaikan produk sesuai dengan data yang diperoleh dari uji coba lapangan.

8. Uji Coba Operasional

Uji coba operasional juga dapat disebut uji coba akhir dalam pengujian aplikasi ini. Uji coba ini dilakukan selama 1 minggu terhitung sejak perbaikan produk selesai dan aplikasi sudah terpasang.

9. Perbaikan Produk Akhir

Setelah dilakukan uji coba operasional, tahap berikutnya adalah mempelajari apakah aplikasi sudah sesuai dengan tujuan yang ditentukan sebelumnya.

10. Deseminasi Nasional

Diseminasi berarti "kegiatan menyebarluaskan suatu pemikiran". Dalam konteks multimedia diseminasi berarti menyebarluaskan pengetahuan mengenai pengembangan multimedia.

2.6.2 Metode Multimedia *Developmet* Luther

Di tahap ini terdapat beberapa tahap lagi di dalamnya [LUT-94], yaitu:

a. *Concept*

Tahap *concept* (konsep) adalah tahap untuk menentukan tujuan dan siapa pengguna program (identifikasi *audience*). Selain itu menentukan macam aplikasi (presentasi atau interaktif) dan tujuan aplikasi (hiburan, pelatihan, pembelajaran, dan sebagainya).

b. *Design*

Design (perancangan) adalah tahap membuat spesifikasi mengenai arsitektur program, gaya, tampilan dan kebutuhan material/bahan untuk program.

c. *Material Collecting*

Material Collecting adalah tahap dimana pengumpulan bahan seperti teks, gambar, animasi, audio, dan video yang sesuai dengan kebutuhan dilakukan.

d. *Assembly*

Tahap *assembly* (pembuatan dan penggabungan) adalah tahap dimana semua objek atau bahan multimedia dibuat. Pembuatan aplikasi didasarkan pada *storyboard* dan struktur navigasi.

e. *Testing*

Dilakukan setelah tahap pembuatan dan penggabungan (*assembly*) selesai dengan menjalankan aplikasi/program dan dilihat apakah ada kesalahan atau tidak. Tahap ini disebut juga sebagai tahap pengujian yang akan dilakukan oleh pembuat aplikasi.

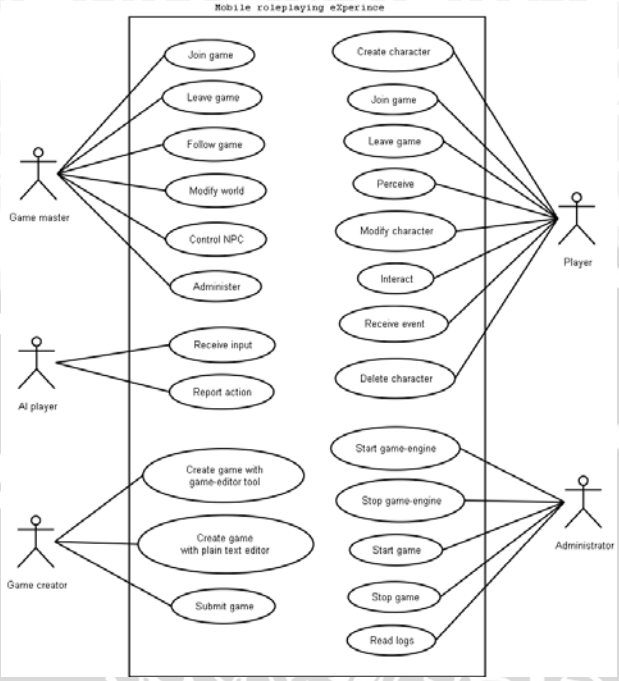
f. *Distribution*

Tahapan dimana aplikasi disimpan dalam suatu media penyimpanan. Pada tahap ini jika media penyimpanan tidak cukup untuk menampung aplikasinya, maka akan dilakukan penyimpanan di media yang lebih besar kapasitasnya.

2.7 *Use Case Diagram*

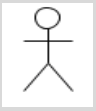
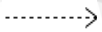
Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu [NGR-07].

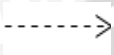
Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Adapun contoh desain *use case* diagram dapat dilihat dalam gambar 2.8 berikut:

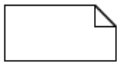


Gambar 2.8 Contoh use case diagram

Tabel 2.1 Keterangan simbol - simbol *use case* diagram

No	Gambar	Nama	Keterangan
1		*Actor*	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya sebagai elemen yang tidak mandiri (*independent*).

3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya (objek induk / *ancestor*).
4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).

10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*.

Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

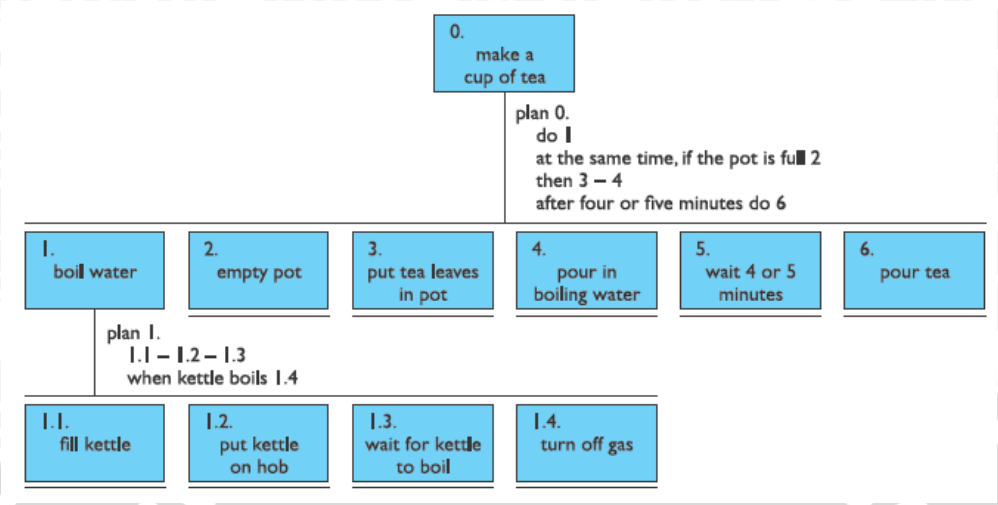
2.8 Analisis Tugas (*Task Analysis)*

Analisis tugas (*Task Analysis)* adalah proses menganalisis cara orang melakukan pekerjaan mereka: hal-hal yang mereka lakukan dan hal-hal yang perlu mereka ketahui [DIX-04]. Sebagai contoh, jika kita sedang mempertimbangkan pekerjaan rumah tangga, kita ingin mengatakan hal-hal seperti:

Untuk Membersihkan Rumah

1. keluarkan *vacuum cleaner*
2. perbaiki peralatan terkait
3. bersihkan kamar
4. ketika kantung debu penuh, kosongkan
5. letakkan kembali *vacuum cleaner* pada tempatnya jika sudah selesai

Atau untuk contoh analisis tugas berbentuk hirarki, dapat dilihat dalam gambar 2.9.



Gambar 2.9 Contoh Analisis Tugas Berbentuk Hirarki

Sumber: [DIX-04]

Untuk melakukan tugas seperti itu, kita perlu tahu beberapa hal tentang pengisap debu, lampiran mereka, kantong debu (jika digunakan), lemari (di mana *vacuum cleaner* disimpan), ruang (yang akan dibersihkan) dan sebagainya.

Analisis tugas membahas tentang sistem dan prosedur yang sudah ada, alat utamanya adalah pengamatan dalam berbagai bentuk. Satu tujuan dari analisis tugas adalah untuk membantu dalam produksi materi pelatihan dan dokumentasi.

Namun, jika sistem komputer yang baru diperlukan, analisis tugas juga memberikan kontribusi dalam laporan persyaratan dari sistem ini. Kita akan melihat bagaimana analisis tugas dapat diterapkan dalam cara yang cukup mudah dalam mendesain menu. Proses merancang sebuah sistem baru berdasarkan analisis dari sistem yang ada akan melibatkan cukup banyak wawasan profesional dan kontribusi analisis tugas untuk ini terutama adalah menjelaskan dan mengorganisir pengetahuan seseorang tentang situasi saat ini.

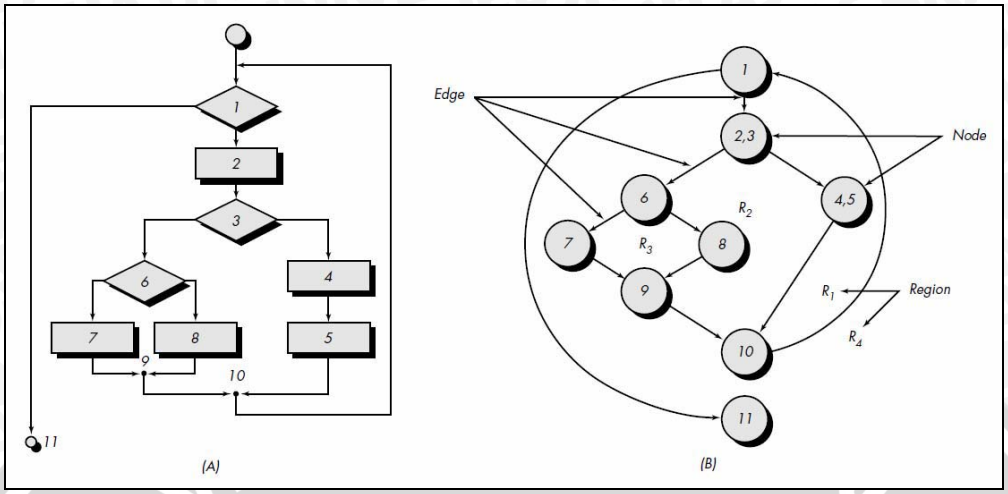
2.9 Teknik Pengujian

Pengujian perangkat lunak memerlukan perancangan kasus uji (*test case*) agar dapat menemukan kesalahan dalam waktu singkat dan usaha minimum. Berbagai macam metode perancangan kasus uji telah berevolusi. Metode-metode ini menyediakan developer pendekatan sistematis untuk pengujian. Terlebih lagi metode-metode ini menyediakan mekanisme yang dapat membantu memastikan kelengkapan dari pengujian dan menyediakan kemungkinan tertinggi untuk menemukan kesalahan-kesalahan dalam perangkat lunak [PRE-01]. Teknik atau metode perancangan kasus uji yang digunakan adalah *black-box testing* dan *white-box testing*.

2.9.1 *White-Box Testing*

White-Box testing atau *glass-box testing* merupakan sebuah metode perancangan kasus uji yang menggunakan struktur kontrol dari perancangan prosedural untuk memperoleh kasus uji [PRE-01]. Ada dua jenis pengujian yang termasuk *white-box testing* yaitu basis *path testing* dan *control structure testing*.

Pada skripsi ini menggunakan basis path testing yang diusulkan pertama kali oleh Tom McCabe [PRE-01]. Basis path testing ini memungkinkan perancang kasus uji memperoleh ukuran kompleksitas logis dari sebuah perancangan prosedural dan menggunakan pengukuran ini sebagai pedoman untuk mendefinisikan basis set dari jalur eksekusi (*execution path*). Test case yang dilakukan untuk menggunakan basis set tersebut dijamin untuk menggunakan setiap statement di dalam program paling tidak sekali selama pengujian. Sebelum metode basis path dapat diperkenalkan, notasi sederhana untuk representasi aliran kontrol yang disebut diagram alir (*flow graph*) harus diperkenalkan. Setiap representasi desain prosedural yang berupa *flow chart* dapat diterjemahkan ke dalam *flow graph*. Gambar 2.10 menunjukkan transformasi *flow chart* ke *flow graph*. Setelah *flow graph* didefinisikan maka harus ditentukan ukuran kompleksitas (*cyclomatic complexity*).



Gambar 2.10 Transformasi *flow chart* ke *flow graph*

Cyclomatic complexity adalah metriks perangkat lunak yang memberikan pengukuran kuantitatif terhadap kompleksitas logis suatu program. Bila metriks ini digunakan dalam konteks metode pengujian basis path, maka nilai yang terhitung untuk *cyclomatic complexity* menentukan jumlah jalur independen (*independent path*) dalam basis set suatu program dan memberi batas atas bagi jumlah pengujian yang harus dilakukan untuk memastikan bahwa semua statement telah dieksekusi sedikitnya satu kali.

Jalur independen adalah jalur yang melalui program yang mengenalkan sedikitnya satu rangkaian statement proses baru atau suatu kondisi baru. Untuk menentukan *cyclomatic complexity* bisa dilakukan dengan beberapa cara, diantaranya [PRE-01]:

1. Jumlah region pada *flow graph* sesuai dengan *cyclomatic complexity*.
2. *Cyclomatic complexity* V(G), untuk grafik G adalah V(G) = E – N + 2, dimana E adalah jumlah *edge*, dan N adalah jumlah *node*.
3. V(G) = P + 1, dimana P adalah jumlah *predicate node* yaitu *node* yang merupakan kondisi (ada 2 atau lebih *edge* akan keluar node ini).

2.9.2 *Black-Box Testing*

Black-box testing atau behavioral testing berfokus pada persyaratan fungsional perangkat lunak [PRE-01]. Dengan demikian, pengujian *Black-Box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk semua program. Pengujian *Black-Box* bukan merupakan alternatif dari teknik *White-Box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan daripada metode white-box.

Pengujian *Black-Box* berusaha menemukan kesalahan dalam kategori berikut :

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi.

Tidak seperti pengujian white-box, yang dilakukan pada saat awal proses pengujian, pengujian black-box cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian black-box memperhatikan struktur kontrol, maka perhatian berfokus pada domain informasi. Pengujian didesain untuk menjawab pertanyaan-pertanyaan berikut :

- Bagaimana validitas fungsional diuji ?
- Kelas input apa yang akan membuat test case menjadi baik ?
- Apakah sistem sangat sensitif terhadap harga input tertentu ?
- Bagaimana batasan dari suatu data diisolasi ?
- Kecepatan dan volume data apa yang dapat ditolerir oleh sistem ?
- Apa pengaruh kombinasi tertentu dari data terhadap operasi sistem ?

2.9.3 Interaksi Manusia Komputer

a. Menurut Baecker dan Buxton [BAE-87]

Interaksi Manusia Komputer didefinisikan sebagai mengatur proses, dialog, dan tindakan melalui - dimana pengguna (manusia) menggunakan dan berinteraksi dengan komputer". Baecker dan Buxton Lebih jauh menuliskan definisi IMK sebagai berikut: interaksi manusia komputer adalah disiplin berkaitan dengan evolusi, desain dan implementasi sistem komputer interaktif untuk digunakan manusia dan dengan studi fenomena utama di sekitar mereka.

b. Menurut penulis, Interaksi Manusia Komputer adalah sebuah hubungan antara manusia dan komputer yang mempunyai karakteristik tertentu untuk mencapai suatu tujuan tertentu dengan menjalankan sebuah sistem yang bertopengkan sebuah antarmuka *(interface)* dan memperhatikan kenyamanan pemakai, kemudahan dalam pemakaian, mudah untuk dipelajari. Para perancang antarmuka manusia dan komputer berharap agar sistem komputer yang dirancangnya dapat bersifat akrab dan ramah dengan penggunanya (*user friendly*).

Unsur Interaksi Manusia dan Komputer

Unsur unsur Interaksi Manusia Komputer [BAE-87] terdiri dari:

a. Manusia, bagaimana manusia menerima dan memproses perintah kedalam komputer.
b. Komputer, teknologi komputer yang digunakan untuk menerima, memproses, dan menghasilkan Output yang telah diprosesnya.
c. Interaksi di antara kedua elemen di atas.

Di dalam Interaksi Manusia Komputer terdapat bagian penting yang harus terpenuhi dalam perancangan sebuah sistem, yaitu *usability*. Pengertian, komponen, dan evaluasi *usability* akan dijelaskan sebagai berikut.

2.9.4 *Usability*

Usabilitas dari suatu sistem ditentukan dari bagaimana kemudahan user mememenuhi tugasnya. Definisi *usability* menurut Jakob Nielsen [NIE-93] adalah

efektifitas dalam pemakaian, efisiensi dalam penggunaan sumber daya, dan kepuasan yang didapatkan pengguna dalam menyelesaikan suatu pekerjaan, dalam sebuah lingkungan dari sebuah produk.

Banyak cara untuk memeriksa *usability* untuk aplikasi komputer yang membantu mengenali masalah-masalah *usability* pada perancangan desain antarmuka. Salah satunya adalah Evaluasi *Heuristik.*

2.9.5 Evaluasi Heuristik

Evaluasi heuristik seperti yang dijelaskan dalam [DIX-04] oleh [NIE-93] adalah *guideline*, prinsip umum dan peraturan, pengalaman yang bisa membantu suatu keputusan atau kritik atas suatu keputusan yang telah diambil, beberapa penilaian bebas terhadap suatu desain supaya kritik bisa memajukan potensi daya guna. Evaluasi *heuristik* banyak dipakai pada rancangan dengan jangka waktu perancangan yang singkat dan dengan dana yang terbatas. Evaluasi *heuristik* dipilih karena dapat dilakukan dengan cepat, mudah dan dengan biaya yang rendah, dibandingkan dengan evaluasi *usability* lainnya. Tujuan utama evaluasi *heuristik* adalah untuk mengidentifikasi masalah yang berkaitan dengan rancangan antarmuka. Metode ini dikembangkan oleh Jakob Nielsen berdasarkan pengalaman mengajar dan konsultasi selama beberapa tahun pada bidang *usability*.

Di dalam [DIX-04] dijelaskan, pada tahun 1990 [NIE-93] mengembangkan evaluasi *heuristik* yang terdiri dari 10 panduan:

1. *Visibility of System Status*

Sistem harus menginformasikan dengan jelas kepada pengguna tentang apa yang sedang terjadi pada saat menggunakan sistem melalui umpan balik yang sesuai dalam kurun waktu yang wajar.

2. *Match Between System and the Real World*

Bahasa ataupun simbol yang digunakan dalam sistem tersebut sebaiknya sinkron atau tidak jauh berbeda dengan kehidupan nyata. Sistem seharusnya berbicara dengan bahasa pengguna menggunakan kata-kata, frase-frase, dan

konsep-konsep yang familiar dengan pengguna daripada menggunakan istilah-istilah yang berorientasi pada sistem.

3. *User Control and Freedom*

Pengguna sering kali memilih fungsi-fungsi sistem dengan salah secara tidak sengaja, dan memerlukan sebuah "pintu darurat" yang tertandai dengan jelas untuk mengeluarkan pengguna dari situasi tersebut tanpa harus melalui dialog yang panjang.

4. *Consistency and Standards*

Sistem dirancang dengan konsisten sehingga tidak membingungkan *user*. Pengguna seharusnya tidak perlu berfikir terlalu lama untuk mengetahui apakah kata-kata, situasi, atau aksi yang ada bermakna sama dalam konteks yang berbeda.

5. *Error Prevention*

Sebuah konfirmasi yang mencegah kesalahan pengguna. Pengguna seharusnya dibuat sulit untuk melakukan kesalahan. Rancangan seharusnya dilakukan secara cermat untuk mencegah sebuah masalah muncul.

6. *Recognition rather than Recall*

Meminimalkan ingatan pengguna dengan membuat objek, aksi, dan fungsi-fungsi mempunyai visualisasi sendiri-sendiri. Pengguna tidak harus mengingat instruksi dan perintah yang ada, sehingga pengguna mudah kembali pada sistem setelah beberapa waktu meninggalkannya.

7. *Flexibility and Efficiency of Use*

Kemudahan pengguna dalam mencari informasi yang diinginkan. Pengguna seharusnya diijinkan untuk melakukan adaptasi terhadap aksi-aksi yang sering dilakukan.

8. *Aesthetic and Minimalist Design*

Tampilan informasi dan elemen desain haruslah relevan, tidak boleh berlebihan, karena keduanya akan bersaing secara visual dengan informasi yang lebih relevan di layar.

9. *Help Users Recognize, Diagnose and Recover from Errors*

Pesan kesalahan sebaiknya langsung menjelaskan kesalahan yang terjadi dalam bahasa yang langsung, jelas, dan secara tepat mengindikasikan problemnya. Dan secara konstruktif mengusulkan sebuah solusi bagi pengguna.

10. *Help and Documentation*

Adanya bantuan dan dokumentasi pada sistem akan sangat membantu pengguna.

Evaluasi *heuristik* pada sebuah perangkat lunak dapat mengidentifikasi masalah-masalah yang ada pada perangkat lunak tersebut. Masalah-masalah tersebut kemudian dinilai, sesuai dengan tingkat kesulitan masalah tersebut (*severity ratings*). *Severity ratings* dapat menentukan banyaknya sumber daya yang diperlukan untuk memperbaiki masalah yang ada.

Tingkat *severity ratings* pada masalah *usability* dapat ditentukan dengan skala 0 sampai 4 berikut:

1. Skala 0: Tidak ada masalah pada usability tersebut.
2. Skala 1: Kategori *cosmetic problem*, masalah tidak perlu diperbaiki, kecuali ada waktu tersisa dalam pengerjaan proyek.
3. Skala 2: Kategori minor *usability problem*, perbaikan masalah ini diberikan prioritas yang rendah.
4. Skala 3: Kategori *major usability problem*, perbaikan masalah ini diberikan prioritas yang tinggi.
5. Skala 4: Kategori *usability catastrophe*, masalah ini harus diperbaiki sebelum produk diluncurkan.