

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Penelitian dilakukan dengan mengacu pada penelitian terdahulu yang berjudul “Pembelajaran Fisika Melalui Komputer untuk Anak SMA Berbasis Multimedia”. Penelitian ini membuat sistem pembelajaran Fisika berbasis multimedia dengan materi-materi yang menampilkan animasi (Astrianto, 2008).

Penelitian berjudul “Pembelajaran Sistem Indera Manusia Berbasis Multimedia Untuk Siswa SMA Kelas XI”, yang memberikan pembahasan pembuatan sistem pembelajaran Biologi dengan sub topik Sistem Indera Manusia dengan materi-materi yang menampilkan animasi (Desianingsih, 2007).

Kedua penelitian tersebut menggunakan *Adobe Flash* sebagai *authoring tool* aplikasi multimedia tersebut. Permasalahan utama kedua penelitian tersebut adalah aplikasi multimedia yang dikembangkan merupakan aplikasi multimedia untuk komputer *desktop*, bukan untuk aplikasi perangkat *mobile* yang memiliki kemampuan pemrosesan dan memori yang terbatas.

## 2.2 Landasan Teori

### 2.2.1 Indeks Massa Tubuh ( IMT )

Masalah kekurangan dan kelebihan gizi pada orang dewasa (usia 18 tahun ke atas) merupakan masalah penting, karena selain mempunyai resiko penyakit-penyakit tertentu, juga dapat mempengaruhi produktifitas kerja. Oleh karena itu, pemantauan keadaan tersebut perlu dilakukan secara berkesinambungan. Salah satu cara adalah mempertahankan berat badan yang ideal atau normal.

Di Indonsia khususnya, cara pemantauan dan batasan berat badan normal orang dewasa belum jelas mengacu pada patoka tertentu. Sejak tahun 1958 digunakan cara perhitungan berat badan normal berdasarkan rumus (Supriasa, 2001):

$$\text{Berat badan normal} = 0,9 \times (\text{tinggi badan} - 100) \dots\dots\dots(1.1)$$

Rumus tersebut berlaku dengan batasan:

$$\text{Nilai minimum} = 0,8 \times (\text{tinggi badan} - 100) \text{ dan } \dots\dots\dots(1.2)$$

$$\text{Nilai maksimum} = 1,1 \times (\text{tinggi badan} - 100) \dots\dots\dots(1.3)$$

Ketentuan ini berlaku umum bagi laki-laki dan perempuan. Berat badan yang berada di bawah batas minimum dinyatakan sebagai *under weight* atau *kekurusan*, dan berat badan yang berada di atas batas maksimum dinyatakan sebagai *over weight* atau *kegemukan*. Orang-orang yang berat badannya di bawah berat normal mempunyai resiko terhadap penyakit infeksi, sementara

yang berada di atas ukuran normal mempunyai resiko terhadap penyakit degeneratif.

Laporan FAO/WHO/UNU tahun 1985 menyatakan bahwa batasan berat badan normal orang dewasa ditentukan berdasarkan nilai *Body Mass Index* (BMI), yang diterjemahkan sebagai Indeks Massa Tubuh (IMT). IMT merupakan alat yang sederhana untuk memantau status gizi orang dewasa khususnya yang berkaitan dengan kekurangan dan kelebihan berat badan. Rumus perhitungan IMT adalah sebagai berikut (Supriasa, 2001):

$$IMT = \frac{\text{Berat}_{\text{badan}}(\text{kg})}{\text{Tinggi}_{\text{badan}}(\text{m})^2} \dots\dots\dots (1.4)$$

Batas ambang IMT di setiap negara bisa berbeda, dan untuk Indonesia batas ambang IMT adalah:

Tabel 1.1 Kategori Ambang Batas IMT untuk Indonesia

|        | <b>Kategori</b>                      | <b>IMT</b>    |
|--------|--------------------------------------|---------------|
| Kurus  | Kekurangan berat badan tingkat berat | < 17,0        |
|        | Kekurangan badan tingkat ringan      | 17,0 – 18,5   |
| Normal |                                      | > 18,5 – 25,0 |
| Gemuk  | Kelebihan berat badan tingkat ringan | >25,0 – 27,0  |
|        | Kelebihan berat badan tingkat berat  | >27,0         |

(Sumber : *Penilaian Status Gizi, 2001, hal.61*).

Contoh:

Eva dengan tinggi badan 147 cm dengan berat badang 39 kg.

$$IMT = \frac{39}{(1,47)^2} = \frac{39}{2,16} = 18,05$$

Dalam hal ini Eva termasuk kategori kekurangan berat badan atau Kurang Energi Kronis (KEK) ringan. Oleh karena itu, Eva harus menaikkan berat badannya sehingga mencapai 40 kg sampai dengan 54 kg.

Tabel 1.2 Resiko Relatif Penyakit Jantung dengan Kelompok IMT

| <b>IMT</b>       | <b>20 - 25</b> | <b>&gt;25 – 30</b> | <b>&gt;30 – 35</b> | <b>35 – 40</b> | <b>&gt;40</b> |
|------------------|----------------|--------------------|--------------------|----------------|---------------|
| Kelompok         | 0              | I                  | II                 | III            | IV            |
| Resiko           | Sangat rendah  | Rendah             | Sedang             | Tinggi         | Sangat tinggi |
| Jumlah sel lemak | Normal         | Normal             | Normal (naik)      | Naik           | Naik          |

(Sumber : *Penilaian Status Gizi, 2001, hal.62*).

### 2.2.2 Nefrologi dan Hipertensi

Definisi hipertensi adalah tekanan darah sistolik  $\geq 140$  mmHg dan tekanan darah diastolik  $\geq 90$  mmHg, atau bila pasien memakai obat antihipertensi (Mansjoer, 2001).

Hipertensi sistolik terisolasi adalah hipertensi dengan tekanan sistolik sama atau lebih dari 160 mmHg, tetapi tekanan diastolik kurang dari 90 mmHg. Keadaan ini berbahaya dan memiliki peranan sama dengan hipertensi diastolik, sehingga harus diterapi.

Tabel 1.3 Klasifikasi Tekanan Darah

| <b>Kategori</b>      | <b>Sistolik</b> | <b>Diastolik</b> | <b>Rekomendasi</b>   |
|----------------------|-----------------|------------------|--|
| Normal               | <130            | <85              | Periksa ulang dalam 2 tahun  |
| Perbatasan           | 130 – 139       | 85 – 89          | Periksa ulang dalam 1 tahun  |
| Hipertensi tingkat 1 | 140 – 159       | 90 – 99          | Konfirmasi dalam 1 atau 2 bulan, anjurkan modifikasi gaya hidup      |
| Hipertensi tingkat 2 | 160 – 179       | 100 – 109        | Evaluasi atau rujuk dalam 1 bulan                                    |
| Hipertensi tingkat 3 | ≥180            | ≥110             | Evaluasi atau rujuk segera dalam 1 minggu berdasarkan kondisi klinis |

(Sumber : Buku Ajar Keperawatan Kardiovaskuler, 2001, hal.115).

Tujuan deteksi dan penatalaksanaan hipertensi adalah menurunkan risiko penyakit kardiovaskular dan mortalitas serta morbiditas yang berkaitan. Tujuan terapi adalah mencapai dan mempertahankan tekanan sistolik di bawah 140 mmHg dan tekanan diastolik di bawah 90 mmHg dan mengontrol faktor risiko. Hal ini dapat dicapai melalui modifikasi gaya hidup saja, atau dengan obat antihipertensi.

Tabel 1.4 Penatalaksanaan Berdasarkan Klasifikasi Resiko

| <b>Tekanan Darah</b> | <b>Kelompok Resiko A</b> | <b>Kelompok Resiko B</b> | <b>Kelompok Resiko C</b> |
|----------------------|--------------------------|--------------------------|--------------------------|
| 130 – 139/85-89      | Modifikasi gaya hidup    | Modifikasi gaya hidup    | Dengan obat              |
| 140-159/90-99        | Modifikasi gaya hidup    | Modifikasi gaya hidup    | Dengan obat              |
| >160/ >100           | Dengan obat              | Dengan obat              | Dengan obat              |

(Sumber : Penilaian Status Gizi, 2001, hal.63).

## 2.3 Konsep Dasar Rekayasa Perangkat Lunak

### 2.3.1 Definisi perangkat lunak

Pressman, menyatakan bahwa Perangkat lunak adalah:

1. Perintah (program komputer) yang bila dieksekusi memberikan fungsi dan unjuk kerja seperti yang diinginkan.
2. Struktur data yang memungkinkan program memanipulasi informasi secara professional.
3. Dokumen yang menggambarkan operasi dan kegunaan program.

Pengaruh perangkat lunak dalam masyarakat dan kebudayaan sangatlah besar. Sejalan dengan semakin meningkatnya peran perangkat lunak, komunitas perangkat lunak secara terus menerus mencoba mengembangkan teknologi yang akan membuatnya semakin mudah, cepat dan murah untuk menciptakan perangkat lunak komputer dengan kualitas tinggi. Sejumlah teknologi tersebut ditujukan untuk sebuah aplikasi tertentu (contoh: perancangan dan implementasi *web site*), lainnya pada sebuah teknologi (contoh: sistem berorientasi objek) dan sisanya bersifat lebih luas (contoh: sistem operasi seperti *Linux*).

### 2.3.2 Karakteristik perangkat lunak

Menyatakan bahwa, “Perangkat lunak lebih merupakan elemen logika bukan merupakan elemen sistem fisik. Dengan demikian, perangkat lunak memiliki ciri yang berbeda dari perangkat keras” (Pressman, 1997).

Karakteristik tersebut adalah:

1. Perangkat lunak merupakan dikembangkan atau direkayasa, tetapi bukan dalam pengertian pabrikasi klasik.

Terdapat sejumlah kesamaan dalam pengembangan perangkat lunak dan pabrikasi perangkat keras, tetapi pada dasarnya kedua aktivitas tersebut sangatlah berbeda. Kedua aktivitas tersebut menyatakan bahwa kualitas tinggi dapat dicapai melalui perancangan yang baik, tapi fase pabrikasi dalam perangkat keras dapat memunculkan permasalahan kualitas yang tidak dimiliki oleh perangkat lunak. Sebagian besar waktu pembuatan perangkat lunak lebih terfokus pada proses rekayasa, yang artinya proyek perangkat lunak tidak dapat dikelola sebagaimana proyek pabrikasi.

2. Perangkat lunak tidak akan pernah “usang”

Perangkat keras ketika dalam kondisi baru memiliki sejumlah permasalahan yang dapat disebabkan cacat pada saat produksi. Seiring dengan berjalannya waktu cacat tersebut diperbaiki dan dalam kurun waktu tertentu permasalahan tersebut akan semakin menurun pada tingkat minimum (perangkat keras berada dalam kondisi maksimum), tetapi jika waktu pemakaian semakin lama, dengan bertambahnya debu, getaran, pemakaian berlebihan, suhu yang ekstrim, penyakit lingkungan dan sebagainya, permasalahan-permasalahan akan muncul kembali dan jumlahnya akan meningkat lagi. Inilah yang kemudian dinyatakan sebagai “usang”. Sedangkan perangkat lunak hampir tidak pernah terkena penyakit lingkungan yang menyebabkan perangkat keras menjadi “usang”.

3. Meskipun industri semakin mengarah ke perakitan komponen, sebagian besar perangkat lunak masih berupa pengembangan secara spesifik, bukan merakit dari sejumlah komponen yang ada.

Komponen perangkat lunak berkualitas tinggi memiliki ciri *reusability*. *Reusability* diartikan bahwa sebuah komponen perangkat lunak harus didesain dan diimplementasikan sehingga dapat dipakai lagi pada berbagai program yang berbeda.

### 2.3.3 Aplikasi perangkat lunak

Perangkat lunak dapat diaplikasikan ke berbagai situasi dimana serangkaian langkah prosedural (seperti algoritma) telah didefinisikan. Kandungan (*Content*) informasi dan determinasi merupakan faktor penting dalam menentukan sifat aplikasi perangkat lunak (Pressman, 2002).

1. *System software* (perangkat lunak sistem), merupakan sekumpulan program yang ditulis untuk melayani program-program lain. Sejumlah perangkat lunak sistem (contoh: *compiler*, editor dan perangkat *file management*) merupakan proses yang kompleks, tapi menentukan struktur informasi. Dalam kasus lain, perangkat lunak sistem memiliki ciri berinteraksi terus menerus dengan perangkat lunak, digunakan oleh sejumlah pengguna pada saat yang bersamaan, operasi berurutan yang membutuhkan penjadwalan, *resource sharing* (berbagi pakai sumber daya), dan manajemen proses yang kompleks.



2. *Real-time software* (perangkat lunak waktu nyata), merupakan perangkat lunak yang memantau/menganalisa/mengendalikan kejadian di dunia nyata (*real world events*). Elemen-elemen perangkat lunak waktu nyata termasuk komponen pengumpul data yang mengumpulkan dan menyusun informasi dari lingkungan luar, menganalisa komponen yang mengubah informasi saat diperlukan oleh aplikasi, komponen kontrol/output yang bertindak terhadap lingkungan luar dan komponen pemantauan yang mengkoordinasikan semua komponen sehingga waktu tanggap nyata (umumnya 1 milidetik sampai 1 detik) dapat dijaga.
3. *Business software* (perangkat lunak bisnis), pemrosesan informasi bisnis merupakan wilayah perangkat lunak tunggal terbesar. Sistem diskrit (contoh: penggajian, akun penerimaan/pembayaran, pergudangan) telah berkembang menjadi perangkat lunak *Management Information System* (MIS) yang mengakses satu atau lebih *database* yang berisi informasi bisnis. Aplikasi-aplikasi di wilayah ini menyusun ulang data yang ada dengan cara tertentu untuk memudahkan operasi bisnis atau para pengambil keputusan.
4. *Engineering and scientific software* (perangkat lunak rekayasa dan ilmiah), merupakan perangkat lunak yang memiliki algoritma “pengolah angka”. Aplikasinya cukup luas, mulai dari astronomi sampai dengan vulkanologi, mulai dari analisa beban otomotif sampai dengan pergerakan orbital pesawat luar angkasa, dan dari biologi molekular sampai dengan pabrikasi terotomatisasi. Meskipun demikian, modern aplikasi yang termasuk dalam perangkat lunak rekayasa/ilmiah mulai menjauh dari

algoritma numerik konvensional. *Computer-aided design* (CAD), sistem simulasi dan aplikasi interaktif yang lain mulai mengarah ke karakteristik perangkat lunak waktu nyata dan bahkan perangkat lunak sistem.

5. *Embedded software* (perangkat lunak tertanam), produk-produk cerdas menjadi hal yang umum dalam pasar industri dan konsumen. Perangkat lunak tertanam berada dalam ROM (*Read Only Memory*) dan digunakan untuk mengendalikan produk dan sistem untuk pasar industri dan konsumen. Perangkat lunak tertanam melakukan fungsi-fungsi yang terbatas dan seolah-olah fungsi yang biasa, sebagai contoh tombol untuk mengendalikan *microwave oven* atau menyediakan fungsi yang signifikan dan kemampuan pengendalian, sebagai contoh kontrol bahan bakar dalam mobil, layar *dashboard* dan sistem pengereman (*braking system*).
6. *Personal computer software* (perangkat lunak komputer pribadi), yang semakin berkembang dalam dua dekade terakhir. Pengolah kata, *spreadsheet*, komputer grafis, multimedia, hiburan, pengelolaan *database*, aplikasi keuangan pribadi dan bisnis, jaringan eksternal, dan pengaksesan *database* merupakan segelintir dari ratusan jenis aplikasi
7. *Web-based software* (perangkat lunak berbasis *web*). Halaman *web* yang ditampilkan oleh *browser* merupakan perangkat lunak yang menyertakan instruksi *executable* (contoh: *CGI*, *HTML*, *Perl*, atau *Java*), dan data (contoh: *hypertext* dan sejumlah format visual dan audio).

8. *Artificial intelligence software* (perangkat lunak kecerdasan buatan). Perangkat lunak *Artificial Intelligence* (AI) memungkinkan algoritma non numerik menyelesaikan permasalahan kompleks yang tidak dapat diselesaikan oleh komputasi atau analisa linier. Sistem pakar, atau yang sering disebut sistem berbasis pengetahuan, pengenalan pola (gambar dan suara), jaringan syaraf tiruan, teorema pembuktian dan permainan (*game*) merupakan contoh dari kategori perangkat lunak AI.

#### **2.3.4 Tujuan rekayasa perangkat lunak**

Menyatakan bahwa “Tujuan rekayasa perangkat lunak adalah menyediakan sebuah kerangka kerja guna membangun perangkat lunak dengan kualitas yang lebih tinggi” (Pressman, 1997).

Rekayasa merupakan analisa, perancangan, konstruksi, verifikasi, dan pengelolaan entitas teknis atau sosial. Apapun bentuk entitas yang akan direkayasa, sejumlah permasalahan harus dijawab:

1. Permasalahan apakah yang harus diselesaikan.
2. Karakteristik entitas yang digunakan untuk menyelesaikan masalah.
3. Cara penatalaksanaan entitas dan solusinya.
4. Cara mengkonstruksikan entitas.
5. Pendekatan yang akan digunakan untuk menemukan error yang terjadi dalam perancangan dan konstruksi entitas.

6. Jenis dukungan bagi entitas untuk jangka waktu yang lama, saat terjadi perbaikan, adaptasi dan penambahan kemampuan yang diminta oleh pengguna terhadap entitas.

### **2.3.5 Fase pengembangan perangkat lunak**

Teknologi secara bersama-sama dan memungkinkan perkembangan perangkat lunak yang tepat waktu dan rasional. Metode rekayasa perangkat lunak memberikan teknik membangun perangkat lunak yang menyangkut serangkaian tugas luas, menyangkut analisis kebutuhan, konstruksi program desain, pengujian dan pemeliharaan.

Usaha yang berhubungan dengan rekayasa perangkat lunak dapat dibangkitkan dalam tiga fase umum dengan tanpa memperdulikan area aplikasi, ukuran proyek atau kompleksitasnya. Roger S. Pressman mendefinisikan fase pengembangan perangkat lunak sebagai berikut:

1. Fase definisi (*Definiton Phase*). Pada defisi ini pengembangan perangkat lunak harus mengidentifikasi informasi apa yang akan dibangun, batasan desain apa yang ada dan kriteria validasi apa yang akan dibentuk untuk mendefinisikan sistem yang sukses. Ada tiga komponen utama dalam fase ini :
  - a. Sistem rekayasa informasi.
  - b. Perancangan proyek perangkat lunak.
  - c. Analisis kebutuhan.

2. Fase pengembangan (*Development Phase*). Selama pengembangan perangkat lunak, teknik harus mendefinisikan bagaimana data dikonstruksikan, bagaimana fungsi-fungsi diimplementasikan sebagai arsitektur perangkat lunak, bagaimana desain prosedur akan diimplementasikan, bagaimana interface ditandai, bagaimana rancangan akan diterjemahkan ke dalam bahasa pemrograman serta bagaimana pengujian akan dilakukan. Tiga komponen dalam fase ini adalah :
  - a. Rancangan perangkat lunak.
  - b. Pemunculan kode.
  - c. Pengujian Perangkat lunak.
3. Fase Pemeliharaan (*Maintenance Phase*). Berfokus pada perubahan kesalahan, penyesuaian yang dibutuhkan ketika lingkungan perangkat lunak berkembang, serta perubahan dengan perkembangan yang disebabkan oleh perubahan kebutuhan pelanggan. Perangkat lunak adalah serangkaian intruksi program ketika dijalankan dapat menyediakan fungsi dan untuk kerja yang diinginkan.

#### **2.4 Teori Interaksi Manusia dan Komputer (IMK)**

Definisi interaksi manusia dan komputer adalah ilmu yang mempelajari tentang bagaimana mendesain, mengevaluasi, dan mengimplementasikan sistem komputer yang interaktif sehingga dapat digunakan oleh manusia dengan mudah (Santosa, 1997).

Interaksi manusia dan komputer tidak berfokus pada keindahan tampilan antarmuka. Aspek-aspek pemakaian, implementasi sistem rancangan, fenomena lingkungan dan lain-lain tetap harus diperhatikan. Kenyamanan pemakaian, kemudahan dalam pemakaian, kemudahan untuk dipelajari merupakan bagian dari interaksi manusia dan komputer.

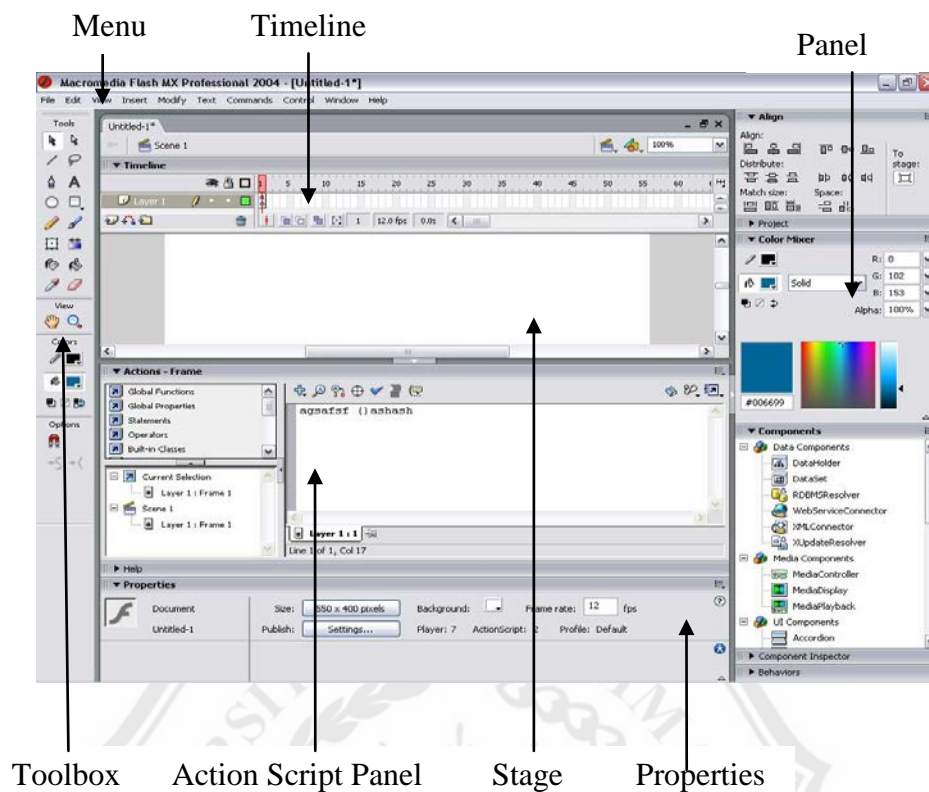
P. Insap Santosa mengemukakan bahwa hal-hal yang harus diperhatikan dalam pengembangan antarmuka adalah sebagai berikut :

1. Pengetahuan tentang mekanisme fungsi manusia sebagai pengguna komputer. Tentunya yang ada hubungannya dengan psikologi kognitif, tingkat perseptual, serta kemampuan motorik pengguna.
2. Berbagai informasi yang berhubungan berbagai informasi yang berhubungan dengan karakteristik dialog yang cukup lebar, seperti ragam dialog, struktur, isi tekstual, grafis, waktu tanggap, dan kecepatan tampilan.
3. Penggunaan prototipe yang didasarkan pada spesifikasi dialog formal yang disusun secara bersama antara calon pengguna (*user*) dan perancang sistem, serta peranti bantu yang dapat digunakan untuk mempercepat proses pembuatan prototipe.
4. Teknik evaluasi yang digunakan untuk mengevaluasi hasil proses *prototipe* yang telah dilakukan, yaitu secara *analitis* berdasarkan pada analisis atas transaksi dialog, secara *empiris* menggunakan uji coba pada sejumlah kasus, umpan balik pengguna dan beberapa analisis yang dikerjakan oleh ahli antarmuka.

## 2.5 *Macromedia Flash CS3*

*Macromedia Flash* adalah salah satu program animasi grafis yang banyak digunakan para designer untuk menghasilkan karya-karya professional, khususnya bidang animasi. Program ini cukup fleksibel dan lebih unggul dibanding program animasi lain sehingga banyak animator yang memakai program tersebut untuk pembuatan animasi. Daya kreatifitas dan selera seni dari sang animator sangat menentukan hasil akhir dari sebuah karya animasi.

*Macromedia* meluncurkan *flash tool* terbarunya yaitu *Flash Profesional CS3*. *Flash Profesional CS3* ini mempunyai banyak fitur baru, di antaranya *Flash Lite* versi 2.1 dengan *interactive mobile device emulator* dan *improved actions panel*. *Flash Lite* yaitu sebuah *flash* berdasarkan *Flash 4 scripting engine*. *Flash Lite* ini ditujukan untuk pasar ponsel yang cukup luas, yaitu ponsel yang tidak mampu mendukung *Flash Player 9* yang mempunyai banyak fitur (Siswoutomo, 2006).



Gambar 2.1 Tampilan *Macromedia Flash CS3*

### 2.5.1 Emulator

*Emulator* atau lebih tepatnya piranti lunak *emulator* memungkinkan suatu program atau piranti lunak yang dibuat pada awalnya oleh suatu sistem komputer (arsitektur dan sistem operasi) dan untuk dijalankan dalam sistem itu atau dijalankan dalam suatu sistem yang didedikasikan, dapat dijalankan dalam sistem komputer yang sama sekali berbeda. Sebagai contoh suatu program *Windows* dapat dijalankan di sistem operasi *Linux* dengan menggunakan piranti lunak *emulator WINE*. Ada pula program yang mengemulasikan suatu komputer dalam komputer, misalnya *VMware*. Contoh lain adalah program-program emulator untuk menjalankan permainan komputer yang awalnya hanya bisa dijalankan pada konsolnya masing-masing, misalnya *Nintendo*, *Atari*,



*PlayStation, Xbox* dan lain-lain. *Emulator* dibuat dengan beberapa alasan di antaranya adalah:

1. Menjalankan program-program lama, umumnya permainan komputer (*game*) yang konsol atau sistem operasinya tidak diproduksi lagi saat ini.
2. Menjalankan piranti lunak yang dibuat hanya untuk *platform-platform* tertentu.
3. Memeriksa program-program yang dibuat oleh *platform* berbeda untuk dijalankan pada platform yang lain. Sebagai contoh misalnya pembuatan program atau piranti lunak pada *PDA* atau ponsel yang terlebih dahulu diemulasikan di komputer personal.

