BAB II

LANDASAN TEORI

Komputer sebagai salah satu bentuk teknologi yang dapat digunakan sebagai alat bantu dalam melaksanakan kegiatan pembelajaran. Dengan bantuan komputer para guru dapat memanfaatkan berbagai sumber informasi yang ada di sekelilingnya sebagai sumber belajar. Para guru dapat menggunakan berbagai program komputer untuk membuat pembelajarannya lebih kaya informasi dan sekaligus lebih menarik, sehingga dapat menimbulkan motivasi belajar pada para siswanya.

Guru diharapkan dapat memanfaatkan berbagai sumber belajar yang mudah digunakan baik sebagai bahan belajar mandiri, seperti halnya CD Pembelajaran yang biasa disebut CAI (*Computer Asissted Instruction*). Dalam pembahasan ini istilah CAI diterjemahkan ke dalam bahasa Indonesia menjadi PBK (Pembelajaran Berbantuan Komputer).

2.1 Pembelajaran Berbantuan Komputer (PBK)

Pembelajaran berbantuan komputer adalah segala sesuatu aktivitas pembelajaran yang dilakukan melalui komputer. Jadi pada PBK aktivitas pemberian materi, menarik minat siswa untuk mengikuti pelajaran, memberikan tes dan memberikan umpan balik semuanya dilakukan oleh komputer.

2.1.1 Kelebihan dari Pembelajaran Berbantuan Komputer (PBK)

Kelebihan dari pembelajaran berbantuan computer (PBK) adalah sebagai berikut:

1. Meningkatkan Interaksi

   Interaksi di sini adalah aktivitas pertukaran informasi antara komputer dengan penggunanya dalam hal ini siswa. Ketika komputer menampilkan suatu pesan maka siswa harus meresponnya. Karena kerja komputer berdasarkan respon yang diberikan siswa, maka pelajaran dalam PBK terikat langsung oleh respon yang diberikan siswa. Ketika siswa sedang berfikir mencari jawabnya maka komputer akan menunggu dan ketika siswa belum memahami materi maka petunjuk tambahan sudah tersedia sehingga siswa dengan mudah memahami pelajaran. Dengan PBK maka interaksi antara siswa dengan materi lebih banyak karena siswa langsung menyimak materi tanpa ada rasa takut, terlalu cepat dan sebagainya.

2. Individualisasi

   Interaksi merupakan kontribusi utama dalam keefektifannya PBK, tetapi individualisasi lebih condong kepada efisiensi, hal ini dapat digambarkan sebagai berikut: Interaksi Keefektifan Individualisasi Efisiensi

3. Efektifitas biaya

   Alasan digunakannya PBK adalah masalah administrasi, karena penggunaan pelayanan dalam PBK tidak membutuhkan kehadiran seorang guru, PBK dapat digunakan dimalam hari, hari-hari libur yang dimana biasanya guru tidak bisa hadir.

4. Motivasi

   Banyak siswa yang menganggap bahwa PBK sangat menarik perhatian mereka, walaupun alasan ketertarikan mereka terhadap PBK sangat beragam

5. Umpan balik

   Umpan balik lebih cepat diterima dalam penggunaan PBK dibandingkan media lain yang sulit atau tidak bisa menerima umpan balik, jawaban siswa bisa dievaluasi dengan cepat.

6. Kemudahan penyimpanan data

   Pelajaran-pelajaran dalam PBK diprogram secara otomatis terhadap segala aspek penyimpanan (Splittgetber, 1979). Hasil-hasil belajar dapat dicetak, nilai-nilai dapat disimpan.

7. Keutuhan pelajaran

   Dengan PBK beberapa bentuk aktifitas seperti membaca, melihat video tape dapat ditampilkan dalam satu layar. Melalui PBK dapat meyakinkan bahwa topik-topik akan disajikan secara utuh.

8. Kendali peserta belajar

   Salah satu hal yang menarik dari siswa dan PBK adalah terjaminnya kewenangan penuh (otoritas) siswa dalam mengambil keputusan-keputusan penting selama proses instruksional untuk memperbesar hasil belajar individu (Caldwel, 1980)

2.1.2 Kelemahan Pembelajaran Berbantuan Komputer

Adapun kelemahan dari pembelajaran berbantuan komputer (PBK) adalah sebagai berikut:

1. Butuh hardware khusus dan mahal

   Keterbatasan terbesar dari PBK adalah membutuhkan perangkat keras yang harganya mahal dan sulit didapat.

2. Sulit Mengulang Topik

   Berbeda dengan buku bacaan, pelajaran-pelajaran dalam PBK sulit diakses untuk pelajaran yang berikutnya, maksudnya bila hardware tidak tersedia, kita akan mengalami kesulitan untuk mengulang pelajaran yang sebelumnya, atau mengalami kesulitan apabila kita hendak membuatnya sebagai acuan atau referensi ketika mengaplikasikan kemampuan belajar.

3. Tergantung pada kemampuan membaca dan visual.

   Untuk dapat menggunakan PBK dibutuhkan CRT (Catode Ray Tube), yang juga dikenal sebagai video display/monitor. Oleh karena itu akan sangat membutuhkan pengetahuan visual siswa, karena mayoritas isi pelajaran berupa teks ada dilayar, kemampuan membaca siswa menjadi faktor utama yang mempengaruhi efektifitas dalam pelajaran PBK.

4. Grafik tidak realistik.

   Banyak kasus, komputer menampilkan grafik atau gambar yang terkadang tidak mewakili objek yang sebenarnya. Terbatas pada warna, kontras, dan membutuhkan waktu untuk menghasilkan suatu gambar yang berkualitas baik.

5. Butuh keterampilan pengembangan tambahan.

   Seorang perancang PBK harus memiliki keahlian dan pengetahuan diluar kemampuan yang dibutuhkan dalam memproduksi media pembelajaran lainnya.

6. Butuh waktu pengembangan yang lama

Walaupun menggunakan bahasa pemrograman komputer yang dapat mempersingkat waktu dalam memprogram pengembangan PBK, tetapi masih ada kendala lain yang membuat waktu yang lama.

7. Kemampuan belajar insindental terbatas.

   Pada PBK akan sangat sulit belajar insidental karena PBK berlangsung berdasarkan program yang sudah dibuat sebelumnya.

8. Kaku

   PBK hanya dapat merespon pada input-input tertentu PBK hanya menerima apakah tombol-tombol dalam keyboard telah ditekan dengan benar.

2.2 Karakteristik-karakteristik dari PBK yang efektif

Karakteristik-karakteristik dari PBK yang efektif

1. PBK yang efektif sesuai dengan tujuan pembelajaran

   PBK yang efektif harus sesuai dengan tujuan pembelajaran yang akan dicapai.

2. PBK yang efektf menyesuaikan dengan karakteristik siswa

   PBK yang efektif harus sesuai dengan karakteristik siswa

3. PBK yang efektif memaksimalkan interaksi

   Mungkin keuntungan yang paling besar dalam pembeelajaran yang dikomputerisasi daripada pembelajaran berdasarkan buku teks dan media lainnya adalah potensi untuk melakukan interaksi selama pelajaran berlangsung.

4. PBK yang efektif diindividualisasikan

Beberapa program memperbolehkan murid untuk mengawasi suatu pelajaran seperti tingkat kesulitan, kapan harus beralih pada bahasan selanjutnya atau ketika bantuan diberikan program tertentu bahkan memperbolehkan untuk memberikan

5. PBK yang efektif menarik minat siswa

   Jangan mengasumsikan bahwa belajar dengan menggunakan komputer maka akan memotivasi siswa. Walaupun beberapa siswa lebih suka bentuk pembelajaran dengan komputer, tetapi hal itu tidak akan berlangsung lama apabila isi dari komputer itu tidak menarik minat mereka.

6. PBK yang efektif melakukan pendekatan yang positif kepada siswa

   Sifat dari PBK yang efektif harus menyerupai seperti antara guru dengan murid pada pertemuan tatap muka. Hindari pemberian hukuman kepada siswa yang masih anak-anak.

7. PBK yang efektif menyediakan feedback yang beragam

   Siswa yang masih anak-anak senang atau bahkan membutuhkan umpan balik yang positif yang menunjukkan bahwa mereka telah melakukan sesuatu.

8. PBK yang efektif sesuai dengan lingkungan pembelajaran

   Banyak pelajaran PBK dikembangkan di sekolah untuk digunakan siswa secara perseorangan dengan lingkungan kelas pada umumnya.

9. PBK yang efektif mengevaluasi prestasi sesering mungkin

   Kemampuan untuk menyesuaikan diri dengan pembelajaran dan kebutuhan siswa merupakan kemampuan yang dimiliki oleh PBK. Hal yang harus dilakukan adalah sebagai berikut:

a. Memberikan pertanyaan yang sesuai
b. Pertanyaan-pertanyaan yang akan diberikan kepada siswa harus sesuai dengan apa yang tertera di dalam TIK dan dapat diukur sejauh mana mereka telah mampu menerima suatu pelajaran.
c. Hindari makna ganda atau pertanyaan yang kurang baik
d. Menulis soal tes yang akan ditafsirkan dengan benar oleh siswa adalah sulit. Soal test harus berisi kosa kata yang sama dengan pengajaran di kelas sehingga siswa dapat menjawab dengan benar. Oleh karena itu dalam menuliskan soal tes harus hati-hati. Soal tes tersebut harus dapat dimengerti oleh siswa.
e. Menilai jawaban dari awal hingga akhir
f. Ketika komputer menerima jawaban tak terduga, komputer dapat memberi tahu siswa bahwa jawaban yang diberikan kurang tepat dan meminta siswa untuk memperbaiki jawabannya. Hal ini harus dapat dilakukan oleh PBK dari awal pelajaran sampai akhir pelajaran.
g. Jangan keliru ketidakmampuan untuk merespon dengan mengabaikan jawaban yang benar

10. PBK yang efektif menggunakan sumber daya komputer yang baik
Perancang PBK yang efektif harus mengetahui kemampuan dari sistem komputernya untuk mengembangkan pelajaran dan mampu membuat pelajaran lebih efektif.
11. PBK yang efektif mengacu pada prinsip desain pembelajaran

Sebuah desain pembelajaran yang bagus dapat memotivasi siswa, memberitahu siswa tentang tujuan pembelajaran, menampilkan perintah yang tersusun rapi, mengevaluasi perkembangan secara berkala, menyediakan variasi umpan balik.

PBK yang efektif harus dapat menlakukan itu semua.

12. PBK yang efektif sudah dievaluasi terus-menerus

    PBK harus dievaluasi di setiap level baik kualitas pembelajaran, pertimbangan afektif, tampilan fisik maupun relevansinya dengan kurikulum.

2.3 Rekayasa Perangkat Lunak

Definisi menurut Fritz Bauer *"Rekayasa Perangkat Lunak adalah pengembangan dan penggunaan prinsip pengembangan suara untuk memperoleh perangkat lunak secara ekonomis yang reliable dan bekerja secara efisien pada mesin nyata"*.

Menurut IEEE [IEE93] mengembangkan definisi yang lebih komprehensif rekayasa, rekayasa perangkat lunak yaitu :

1. Aplikasi dari sebuah pendekatan *kuantifiabel*, disiplin, dan sistematis kepada pengembangan, operasi, dan pemeliharaan perangkat lunak.
2. Studi tentang pendekatan-pendekatan seperti pada nomor satu.

2.4 Definisi Aliran Data Berorientasi Objek

Pemrograman berorientasi objek, atau sering juga disebut dengan OOP (*Object Oriented Programming*), merupakan paradigma baru dalam

pengembangan sistem/perangkat lunak. Tidak seperti pendahulunya, yaitu pemrograman dengan teknik terstruktur (*structured programming*) yang sering mengalami kegagalan dalam hal sistem/perangkat lunak yang tidak sesuai dengan kebutuhan dan harapan pengguna (*user's need and expectations*) serta kerap kali tidak dapat memenuhi jadwal yang telah ditentukan sebelumnya. Pemrograman berorientasi objek dengan paradigma barunya, menjanjikan tingkat penggunaan ulang (*reusability*), kinerja (*performance*), serta kehandalan (*reliability*) yang lebih tinggi. (Roger Pressman, 2000).

Saat ini, dunia pemrograman berorientasi objek mengenal UML (*Unified Modelling Language*) sebagai alat utama dalam analisis dan perancangan sistem. UML merupakan alat yang sangat sesuai dengan pemrograman berorientasi objek sebab konsep dasarnya adalah memodelkan kelas-kelas (beserta atribut serta operasi di dalamnya) bersamaan dengan relasi-relasi yang terjadi antar kelas yang bersangkutan.

2.5 Definisi UML (*Unified Modeling Language*)

UML (*Unified Modelling Language*) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

UML adalah bahasa grafis untuk mendokumentasikan, menspesifikasikan dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung

bahasa dan teknologi, pemaduan beberapa notasi di beragam metodologi, usaha bersama dari banyak pihak, didukung oleh kakas-kakas yang diintegrasikan oleh XML. Standar UML dikelola oleh OMG (*Object Management Group*).

UML (*Unified Modelling Language*) adalah bahasa pemodelan untuk menspesifikasikan, memvisualisasikan, membangun dan mendokumentasikan artifak-artifak dari sistem.

1. Di dalam *system intensive process*, metode diterapkan sebagai proses untuk menurunkan atau mengevolusikan sistem.
2. Sebagai bahasa, UML digunakan untuk komunikasi yaitu alat untuk menangkap pengetahuan mengenai satu subyek dan mengekspresikan pengetahuan (sintaks) yang memperdulikan subyek untuk maksud komunikasi. Subyek adalah sistem yang dibahas.
3. Sebagai bahasa pemodelan, UML fokus pada pemahaman subyek melalui formulasi model dari subyek. Model memuat pengetahuan pada subyek, dan aplikasi dari pengetahuan ini berkaitan dengan intelejensia.
4. Berkaitan dengan unifikasi.UML memadukan praktek rekayasa terbaik sistem informasi dan industri, meliputi beragam tipe sistem (perangkat lunak dan non perangkat lunak), domain (bisnis, perangkat lunak) dan proses siklus hidup.
5. begitu diterapkan untuk menspesifikasikan sistem, UML dapat digunakan untuk mengkomunikasikan “apa” yang diperlukan dari sistem dan “bagaimana” sistem dapat direalisasikan.
6. Begitu diterapkan untuk menvisualisasikan sistem, UML dapat digunakan untuk menjelaskan sistem secara visual sebelum direalisasikan.

7. Begitu diterapkan untuk membangun sistem, UML dapat digunakan untuk memandu realisasi sistem serupa dengan “blueprint”.
8. Begitu diterapkan untuk mendokumentasikan sistem, UML dapat digunakan untuk menangkap pengetahuan mengenai sistem pada seluruh siklus hidup.

UML bukanlah :

1. Bahasa pemrograman visual, tapi bahasa pemodelan visual.
2. Spesifikasi kakas, tapi spesifikasi bahasa pemodelan.
3. Proses, tapi yang memungkinkan proses-proses.

Tujuan utama perancangan UML adalah :

1. Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti.
2. Menyediakan mekanisme perluasan dan spesifikasi untuk memperluas konsep-konsep inti.
3. Mendukung spesifikasi independen bahasa pemrograman dan proses pengembangan tertentu.
4. Menyediakan basis formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar kakas berorientasi objek.
6. Mendukung konsep-konsep pengembangan level lebih tinggi seperti komponen, kolaborasi, *framework* dan *pattern.*
7. Diagram Teknik Pemodelan UML

Diagram mengemukakan banyak hal, penggunaan notasi yang terdefinisi baik dan ekspresif adalah penting pada proses pengembangan perangkat lunak, yaitu :

1. Notasi standar memungkinkan pengembang mendeskripsikan skenario atau rumusan arsitektur dan kemudian mengkomunikasikan secara tidak ambigu.
2. Notasi yang bagus membebaskan otak untuk berkonsentrasi pada masalah-masalah yang lebih lanjut.
3. Notasi yang baik memungkinkan mengeleminasi keperluan pemeriksaan konsistensi dan kebenaran keputusan keputusan dengan menggunakan *tool* terotomatisasi.

2.5.1 Diagram Struktur

Diagram ini untuk memvisualisasi, menspesifikasikan, membangun dan mendokumentasikan aspek statik dari sistem. Diagram struktur di UML terdiri dari :

1. Diagram kelas (*Class diagram*)

   Diagram ini menunjukkan sekumpulan kelas, *interface* dan kolaborasi dan keterhubungannya.

2. Diagram objek (*Object diagram*)

   Diagram ini menunjukkan sekumpulan objek dan keterhubungannya.

3. Diagram komponen (*Component diagram*)

   Diagram ini menunjukkan organisasi dan kebergantungan di antara sekumpulan komponen. Diagram ini merupakan pandangan statik terhadap implementasi sistem.

4. Diagram deployment (*Deployment diagram*)

   Diagram ini menunjukkan konfigurasi pemrosesan saat jalan dan komponen komponen yang terdapat didalamnya.

2.5.2 Diagram Perilaku

Diagram ini untuk memvisualisasi, menspesifikasi, membangun dan mendokumentasikan aspek dinamis dari sistem. Diagram perilaku di UML terdiri dari :

1. Diagram *use-case* (*Use case diagram*)

   Diagram ini menunjukkan sekumpulan kasus fungsional dan *actor* (jenis kelas khusus) dan keterhubungannya.

2. Diagram sekuen (*Sequence diagram*)

   Diagram ini menunjukkan interaksi yang terjadi antar objek. Diagram ini merupakan pandangan dinamis terhadap sistem.

3. Diagram kolaborasi (*Collaboration diagram*)

   Diagram ini juga merupakan diagram interaksi. Diagram ini menekankan pada organisasi struktur dari objek-objek yang mengirim dan menerima pesan.

4. Diagram *statechart* (*Statechart diagram*)

   Diagram ini adalah *state-machine* diagram, berisi state, transisi, kejadian dan aktivitas. Statechart merupakan pandangan dinamis dari sistem.

5. Diagram aktivitas (*Activity Diagram*)

   Diagram ini untuk menunjukkan aliran aktivitas di sistem. Diagram ini adalah pandangan dinamis terhadap sistem.

2.5.3 Notasi dalam UML (*Unified Modeling Language*)

UML menyediakan beberapa notasi dan artifak standar yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan desain.

Artifak di dalam UML didefinisikan sebagai informasi dalam berbagai bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat lunak.

1. *Actor*

   Actor adalah segala sesuatu yang berinteraksi dengan sistem aplikasi komputer. Jadi *actor* ini bisa berupa orang, perangkat keras, atau mungkin objek lain dalam sistem yang sama. Biasanya yang dilakukan *actor* adalah memberikan informasi pada sistem dan atau memerintahkan sistem untuk melakukan sesuatu.

2. *Class*

   Class merupakan pembentuk utama dari sistem berorientasi objek karena *class* menunjukkan kumpulan objek yang memiliki atribut dan operasi yang sama. *Class* digunakan untuk mengabstraksikan elemen-elemen dari sistem yang sedang dibangun.

3. *Interface*

   Interface merupakan kumpulan operasi tanpa implementasi dari suatu *class.*

4. *Use Case*

   Use case menjelaskan urutan kegiatan yang dilakukan *actor* dan sistem untuk mencapai suatu tujuan tertentu walaupun menjelaskan kegiatan. Namun *use case* hanya menjelaskan “apa” yang dilakukan oleh *actor* dan sistem, bukan “bagaimana” *actor* dan sistem melakukan kegiatan tersebut.

5. *Intercation*

   Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar objek.

6. *Package*

   Package adalah kontainer atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari sistem yang sedang di bangun, sehingga bisa dibuat model yang lebih sederhana. Tujuannya adalah untuk mempermudah penglihatan (*visibility*) dari model yang sedang dibangun.

7. *Note*

   Note digunakan untuk memberikan keterangan dan komentar tambahan dari satu elemen sehingga bisa langsung terlampir dalam model. *Note* ini bisa ditampilkan ke semua elemen notasi yang lain.

8. *Dependency*

   Dependency merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain.

9. *Boundary Class*

   Boundary class adalah *class* yang menghubungkan *user* dengan sistem.

10. *Control Class*

    Control class adalah *class* yang mengkoordinasi aktivitas dalam sistem. *Class* ini menghubungkan *boundary class* dengan *entity class*.

11. *Entity Class*

    Entity Class adalah *class* yang menghubungkan dengan data atau informasi yang digunakan oleh sistem. *Entity class* ini adalah *class* yang menyimpan dan mengelola data dapat dikombinasikan.

2.6 Perangkat Lunak untuk Analisis dan Perancangan Sistem

Perangkat lunak (*software*) adalah sebuah program yang berisi instruksi-instruksi yang harus dikerjakan oleh perangkat keras (*hardware*). *Software* merupakan perangkat yang dapat dilihat oleh mata telanjang, tetapi tidak dapat diraba, *software* juga sering digunakan untuk menunjuk kepada semua program yang dapat dipakai dalam sistem komputer.

Perangkat lunak dapat dikategorikan kedalam tiga bagian, yaitu :

1. Perangkat Lunak Sistem (*Operation Software*)

   Yaitu program yang ditulis untuk mengendalikan dan mengkoordinasikan kegiatan dari sistem komputer.

2. Perangkat Lunak Bahasa (*Language Software*)

   Yaitu program yang digunakan untuk menterjemahkan intruksi-instruksi yang ditulis dalam bahasa pemograman ke dalam bahasa mesin agar dapat dimengerti oleh komputer.

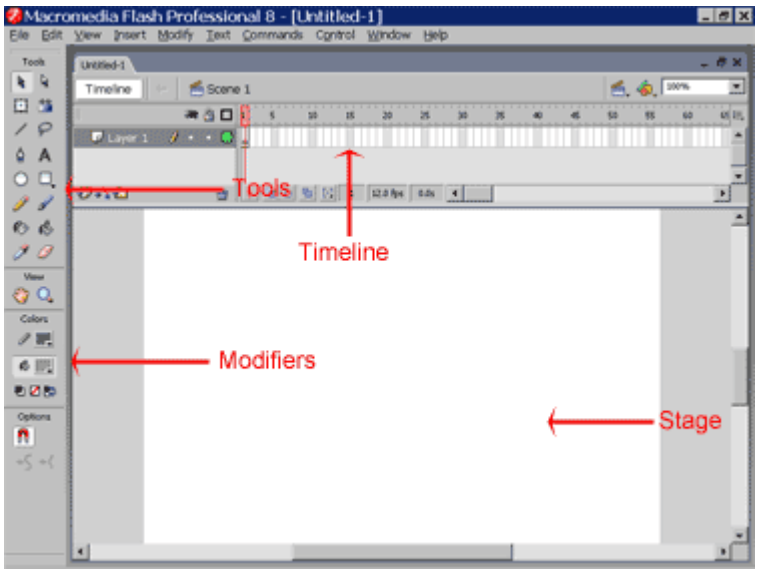
3. Perangkat Lunak Aplikasi (*Application Software*)

   Yaitu program yang ditulis dan diterjemahkan oleh *language software* untuk menyelesaikan suatu aplikasi tertentu.

2.6.1 *Macromedia Flash*

Flash adalah salah satu program pembuatan animasi yang sangat handal. Kehandalah flash dibandingkan dengan program yang lain adalah dalam hal ukuran file dari hasil animasinya yang kecil. Untuk itu animasi yang dihasilkan oleh program flash banyak digunakan untuk membuat sebuah web agar menjadi tampil lebih interaktif.

Seperti telah dijelaskan pada bab sebelumnya bahwa salah satu kendala dari pembuatan sebuah web adalah ukurannya yang harus kecil, sehingga kesulitan untuk memasukkan animasi kedalamnya disebabkan oleh ukuran file animasi yang umumnya sangat besar. Program flash mengatasi kendala kendala diatas dengan baik, sehingga hampir semua web yang interaktif saat ini menggunakan flash sebagai bagian didalamnya.



Gambar 2.1 Interface Macromedia Flash

Berikut adalah deskripsi dari gambar diatas :

1. Toolbar,Toolbal berisi kumpulan tool yang digunakan untuk membuat dan memilih isi di dalam Timeline dan Stage. Toolbar terbagi menjadi tool dan modifier. Setiap tool memiliki ukuran Modifier tertentu yang ditampilkan ketika kita memilih tool tersebut. Sebagai contoh, jika kita memilih Text tool maka modifier seperti huruf, ukuran huruf, warna huruf dapat kita atur.

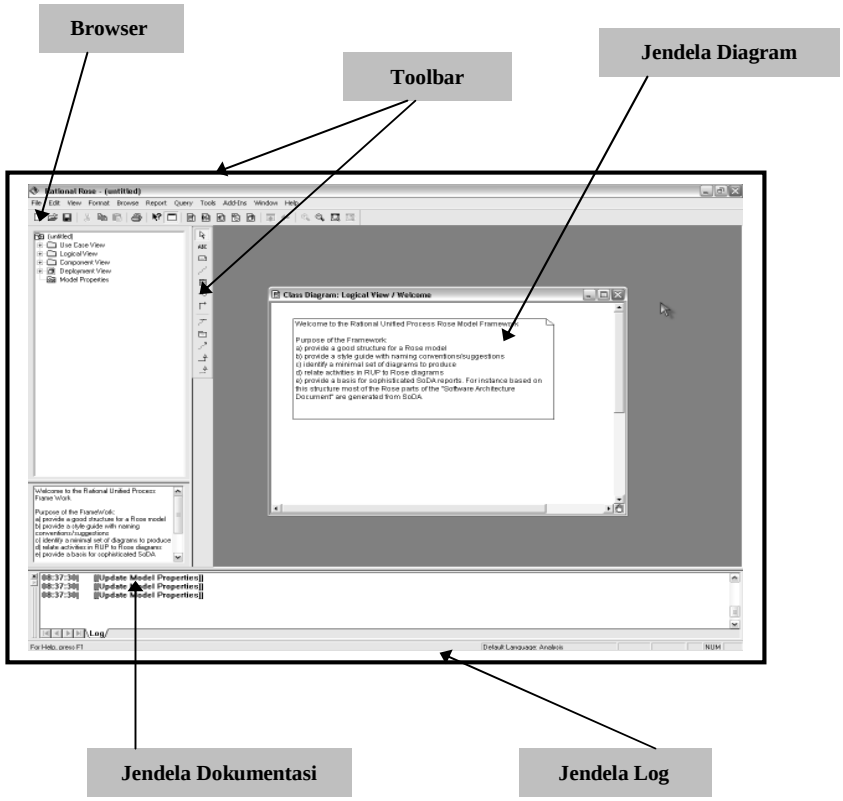
2. Timeline,adalah organisasi dan pengaturan tool yang besar dan kompleks. Anda akan menggunakan keuntunngannya ketika anda membuat animasi. Kemudian anda akan perlu untuk lebih mendalami komponen-komponennya pada bab selanjutnya. Sekarang anda hanya memerlukan untuk mengerti Timeline secara umum.
3. Stage, adalah daerah yang berisi semua elemen gambar yang membentuk movie Flash.. Kita akan selalu bekerja membuat gambar, membuat animasi, dll di tempat ini. Pada Flash, anda dapat mengontrol seberapa besar layarnya, dan apa warnanya, melalui kotak dialog Movie Properties

2.6.2 *Rational Rose*

Rational Rose adalah kakas (*tools*) pemodelan visual untuk pengembangan sistem berbasisi objek yang cukup handal untuk digunakan sebagai bantuan bagi para pengembang dalam melakukan analisis dan perancangan sistem.

Rational Rose mendukung pemodelan bisnis, yang membantu para pengembang untuk memahami sistem secara komprehensif. Ia juga membantu analisis sistem dengan cara pengembang membuat diagram *use case* untuk melihat fungsionalitas sistem secara keseluruhan sesuai dengan harapan dan keinginan pengguna. Kemudian, ia juga menuntut pengembang untuk mengembangkan interaction diagram untuk melihat bagaimana objek-objek saling bekerjasama dalam menyediakan fungsionalitas yang diperlukan. Dan masih

banyak lagi fitur-fitur yang disediakan dalam *tool* ini yang lebih memudahkan pengembang untuk membuat *blue print* sistem yang dibuatnya.



Gambar 2.2 Interface Rational Rose

Berikut adalah deskripsi dari gambar diatas :

1. Jendela *Browser*, jendela ini berfungsi untuk secara cepat bergerak dalam model. Segala sesuatu seperti *actor*, *use case,* kelas, objek, komponen, dan sebagainya yang kita tambahkan dalam model akan didaftarkan dan ditampilkan dalam jendela *browser.*
2. Jendela Diagram, jendela ini berfungsi untuk membuat, menampilkan (*display*), serta menyunting (*edit*) satu atau lebih diagram UML. Dalam diagram ini perubahan yang dilakukan pada jendela *browser*

maka akan secara otomatis akan ditampilkan dalam jendela diagram ini.

3. Jendela Dokumentasi, jendela ini berguna untuk melihat atau memperbaharui (*update*) dokumentasi unsur-unsur model. Sebagai contoh, mungkin kita akan menulis definisi singkat untuk masing-masing aktor serta *use case* pada model kita. Saat kita menambahkan dokumentasi pada suatu kelas, apapun yang kita tulis pada jendela dokumentasi kelak akan hadir sebagai komentar-komentar (tidak diproses) pada kode-kode yang akan dihasilkan.
4. Jendela Log, jendela ini berguna untuk melihat kesalahan-keslahan dan melaporkan hasil-hasil dari berbagai perintah yang diberikan pada *rational rose*.
5. *Toolbar*, selain menu yang dapat di akses dengan meng-klik nama-nama menu yang bersangkutan untuk menjalankan perintah-perintah tertentu, *rational rose* juga menyediakan *toolbar* yang sangat *intuitif* untuk menjalankan perintah-peritah yang sering digunakan.