

BAB II

TINJAUAN PUSTAKA

II.1. Senjata Tradisional di Pulau Sumatera

Rencong menyerupai pedang kecil dengan tanduk di bagian pangkalnya dari Nanggroe Aceh Darussalam. Rencong terbuat dari perak, kuningan, besi putih, kayu dan gading. (Fatiharifah, 2015: 7). Piso Gaja Dompok adalah senjata khas suku Batak, yang merupakan pusaka Kerajaan Batak dari Sumatera Utara. Piso Gaja Dompok berarti pisau tajam yang memiliki ukiran berpenampang gajah. (Fatiharifah, 2015: 10).

Kerambit adalah pisau genggam kecil berbentuk melengkung dari Sumatera Barat. Menurut cerita rakyat, bentuk kerambit terinspirasi oleh cakar harimau yang banyak berkeliaran di hutan Sumatera pada masa itu. (Fatiharifah, 2015: 13). Keris Palembang memiliki lekukan dengan jumlah ganjil, yaitu 7, 9, dan 13. Senjata tradisional yang lainnya adalah tombak trisula. (Fatiharifah, 2015: 16). Pedang jenawi biasa digunakan oleh para dalam pertempuran tempo dulu dari provinsi Riau. Panjang pedang ini bias mencapai satu meter. (Fatiharifah, 2015: 18).

Badik Tumbuk Ladi adalah sejenis senjata tikam berukuran 27 sampai 29 cm. Sejak dulu masyarakat Melayu Kepulauan Riau menggunakan badik tumbuk lado untuk berburu dan mempertahankan diri dari serangan musuh. (Fatiharifah, 2015: 22). Keris adalah senjata tusuk genggam yang bentuknya meruncing dengan

tajaman di kedua sisi bilahnya. Keris pusaka yang terkenal di Jambi adalah Keris Siginjai. (Fatiharifah, 2015: 25).

Parang Badau memiliki bentuk seperti layar kapal dan digunakan untuk perkelahian jarak pendek dari Bangka Belitung. Selain itu, parang yang berdiameter sedang atau sekitar 40 cm dapat digunakan untuk menebang pohon karena bobot ujungnya yang lebih besar dan lebih berat. (Fatiharifah, 2015: 28). Keris termasuk senjata tradisional daerah Bengkulu. Keris yang dianggap keramat atau pemberani, panjangnya 13, jari, dan dipakai oleh kepala adat atau hulubalang. (Fatiharifah, 2015: 30). Keris untuk senjata tikam jarak dekat, sedangkan payan adalah senjata untuk pertempuran jarak menengah atau jauh dari Lampung. (Fatiharifah, 2015: 34).

II.2. Multimedia Interaktif

Yang dimaksud dengan multimedia adalah interaksi antara teks, suara, gambar statis, animasi, dan video (Perry, 1994). Ditinjau dari kata-kata pembentuknya, multimedia berarti “melibatkan berbagai media”. Dengan menggunakan multimedia, informasi dapat ditampilkan secara serentak melalui berbagai media. Anda bias melihat informasi pada layar baik berupa teks ataupun gambar dan video dan pada waktu yang bersamaan Anda bias mendengarkan paparan dalam bentuk suara melalui *speaker*. (Abdul Kadir, 2005: 302).

II.2.1. Elemen Multimedia

Sebagaimana tersirat dalam defenisi multimedia di depan, elemen multimedia meliputi:

1. Teks

Teks merupakan bentuk media yang paling umum digunakan dalam menyajikan informasi, baik yang menggunakan model baris perintah ataupun GUI. Teks dapat disajikan dengan berbagai bentuk font maupun ukuran. (Abdul Kadir, 2005: 306).

2. Suara

Suara merupakan media ampuh untuk menyajikan informasi tertentu; misalnya untuk memperdengarkan cara melafalkan sebuah kata dalam bahasa Inggris. Dengan bantuan suara, pemakai dapat mendengar bunyi suatu kata dengan tepat. Berbagai perangkat lunak dapat digunakan untuk merekam suara. Sebagai contoh, Sound Recorder yang tersedia pada Windows dapat digunakan untuk merekam suara dengan format WAVE. (Abdul Kadir, 2005: 307-308).

3. Gambar Statis

Kata pepatah, “Gambar mewakili seribu kata”. Hal ini mencerminkan bahwa sebuah gambar seringkali dapat lebih berarti bagi seseorang daripada jumlah kata. Umumnya gambar disimpan dengan cara dimampatkan. Tujuannya adalah untuk menghemat ruang dalam penyimpanan eksternal. Untuk melakukan pemampatan ini, ada dua teknik yang dikenakan. Yang pertama dinamakan teknik *lossless* dan yang kedua adalah teknik *lossy*. Teknik pemampatan *lossy* adalah suatu teknik yang memampatkan data sehingga gambar rekonstruksi hasil pemampatan mempunyai perbedaan dengan gambar asli, tetapi bagi mata manusia kelihatan sama. Dengan cara seperti ini, gambar dapat dimampatkan sekecil mungkin (terlihat dalam ukuran berkas). Teknik pemampatan *lossless* adalah suatu teknik yang menghasilkan gambar rekonstruksi yang sama dengan gambar aslinya. (Abdul Kadir, 2005: 308-309).

4. Animasi

Animasi berarti teknik untuk membuat gambar yang bergerak. Berbagai teknik animasi telah diciptakan. Cara sederhana untuk membuat gambar yang bergerak adalah dengan menggunakan sejumlah gambar yang berbeda sedikit. Teknik animasi yang lain yang sangat terkenal adalah *morphing*. Teknik ini dilakukan dengan menggunakan konsep metamorphose dari suatu gambar awal menjadi gambar akhir. (Abdul Kadir, 2005: 312-313).

5. Video

Contoh perangkat lunak terkenal yang biasa digunakan untuk membuat film dengan format AVI ataupun yang lain yaitu Adobe Premiere. Windows juga menyediakan perangkat lunak untuk keperluan ini, yaitu Windows Movie Maker. (Abdul Kadir, 2005: 316).

Pembelajaran yang menggunakan teknologi informasi dan komunikasi atau menggunakan multimedia disebut dengan media pembelajaran berbasis multimedia interaktif. Penggunaan media pembelajaran ini dimaksudkan untuk membantu dosen dalam penyampaian materi dan juga membantu mahasiswa dalam memahami materi yang diajarkan.

Selain itu muatan materi pelajaran dapat dimodifikasi menjadi lebih menarik dan mudah dipahami, tujuan materi yang sulit akan menjadi mudah, suasana belajar yang menegangkan menjadi menyenangkan. Dengan menggunakan media pembelajaran berbasis multimedia dapat memadukan media-media dalam proses pembelajaran, maka proses pembelajaran akan berkembang dengan baik, sehingga membantu dosen menciptakan pola penyajian yang interaktif. (Dwi Sarwiko;2010).

II.2.2. Animasi

Animasi adalah suatu teknik yang banyak sekali digunakan dalam dunia film dewasa ini, baik sebagai suatu kesatuan yang utuh, bagian dalam suatu film maupun bersatu dalam film live.

Dunia film sebenarnya berakar dari fotografi, sedangkan animasi berakar dari dunia gambar, yaitu ilustrasi design grafis (desain komunikasi visual).

Dapat dikatakan bahwa animasi merupakan suatu media yang lahir dari dua konversi atau disiplin, yaitu film dan gambar. Untuk dapat mengerti dan memakai teknik animasi, dua konversi tersebut harus dipahami dan dimengerti.

Film biasanya dipakai untuk merekam suatu keadaan atau mengemukakan sesuatu. Film digunakan untuk memenuhi suatu kebutuhan umum yaitu mengkomunikasikan suatu gagasan, pesan atau kenyataan. Karena keunikan dimensinya dan karena sifat hiburannya, film telah diterima sebagai salah satu media audio visual yang paling populer dan paling digemari. Karena itu juga dianggap sebagai media yang paling efektif.

Keinginan manusia untuk membuat gambar atau santiran (*image*) yang hidup dan bergerak sebagai perantara dari pengungkapan (*expression*) mereka, merupakan perwujudan dari bentuk dasar animasi yang hidup dan berkembang. (Yunita Syahfitri;2011).

II.2.3. Jenis-jenis Animasi

Ada beberapa jenis animasi, diantaranya adalah :

1. *Stop Motion* disebut juga *frame by frame*. Teknik animasi ini akan membuat objek seakan bergerak. Objek bisa bergerak karena mempunyai banyak *frame* yang dijalankan secara berurutan.
2. *Cell Animation* merupakan bentuk animasi tertua dan merupakan bentuk animasi yang paling populer.

3. *Time-Lapse*. Setiap *frame* akan di-*capture* dengan kecepatan yang lebih rendah daripada kecepatan ketika *frame* dimainkan.
4. *Claymation* dulunya disebut dengan *Clay Animation* dan merupakan salah satu bentuk dari *stop motion animation*.
5. *Cut-Out Animation*. Teknik ini digunakan untuk memproduksi animasi menggunakan karakter, properti, dan *background* dari potongan material seperti kertas, karton atau foto. Saat ini, *cut-out animation* diproduksi menggunakan komputer dengan gambar dari hasil pemindai atau grafik vektor untuk menggantikan potongan material yang digunakan.
6. *Puppet Animation*. Dalam *puppet animation*, boneka akan menjadi aktor utamanya sehingga animasi jenis ini membutuhkan banyak boneka. Animasi jenis ini dibuat dengan teknik *frame by frame*, yaitu setiap gerakan boneka di-*capture* satu per satu dengan *camera*.

Kata animasi itu sendiri sebenarnya penyesuaian dari kata animation yang berasal dari animation yang berasal dari kata dasar to animate dalam kamus umum Inggris – Indonesia berarti menghidupkan (Wojowasito, 1997). Secara umum animasi adalah suatu kegiatan menghidupkan, menggerakkan benda mati; Suatu benda mati, diberikan dorongan kekuatan, semangat dan emosi untuk menjadi hidup dan bergerak atau hanya berkesan hidup. (Yunita Syahfitri;2011).

II.3. Pengenalan 3Ds Max

3D Studio Max (3Ds Max) adalah software visualisasi (modeling dan animasi) 3 dimensi yang populer dan serbaguna. Hasil yang dibuat di 3D Studio Max sering digunakan di pertelevisian, media cetak, *games*, web, dll. (Hendi Hendratman, 2014).

3D Studio Max (3Ds Max) adalah sebuah *software* yang dikhususkan dalam pemodelan 3 dimensi ataupun untuk pembuatan animasi 3 dimensi. Selain terbukti andal untuk digunakan

dalam pembuatan objek 3 dimensi, 3Ds Max juga banyak digunakan dalam pembuatan desain *furniture*, konstruksi, maupun desain interior. Selain itu, 3Ds Max juga sering digunakan dalam pembuatan animasi ataupun film kartun. 3Ds Max merupakan *software* tiga dimensi yang dapat membuat objek gambar tampak realistis (nyata).

Keunggulan yang dimiliki 3Ds Max adalah kemampuannya dalam menggabungkan objek *image*, *vector* dan tiga dimensi, serta langsung dapat menganimasikan objek tersebut. 3Ds Max juga mampu menghasilkan objek dalam bentuk gambar ataupun dalam bentuk file interaktif seperti gambar animasi yang disimpan dalam bentuk file **.avi* (*Audio Video Interleave*) atau **.mov* (*Movie*). (Galih Pranowo, 2010: 6).

II.3.1. Area Kerja 3Ds Max

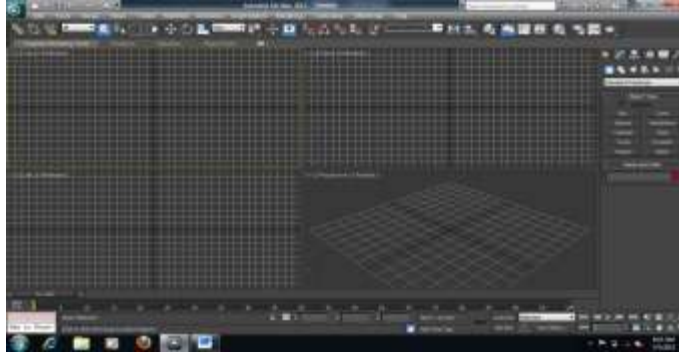
Saat pertama kali menjalankan program 3D Studio Max, maka kita akan mendapati tampilan halaman pembuka 3Ds Max seperti yang terlihat pada Gambar II.1. di bawah ini :



Gambar II.1. Tampilan Halaman Pembuka 3Ds Max

Sumber : (Galih Pranowo ; 2010)

Setelah proses *loading* program 3Ds Max selesai, maka akan tampil bagian antarmuka dari 3Ds Max 9. Area kerja 3Ds Max dapat dilihat pada Gambar II.2.



Gambar II.2. Tampilan Area Kerja 3Ds Max

Sumber : (Galih Pranowo ; 2010 : 2)

II.3.2. Command Panel

Command Panel adalah bagian yang akan sering digunakan selain *Viewport*. *Command Panel* terletak di sebelah kanan *Viewport* dan merupakan tempat-tempat parameter objek, *setting*, dan *control*. *Command Panel* dalam 3Ds Max dibagi dalam enam panel yang masing-masing dapat diakses melalui tab ikon yang berada di atas panel. Ke enam panel tersebut meliputi *Create* (untuk membuat sebuah objek), *Modify* (untuk memodifikasi suatu objek dan menambahkan modifier), *Hierarchy* (parameter-parameter untuk melakukan link dan parameter Inverse Kinematics), *Motion* (sebagai pengatur animasi), *Display* (control tampilan), dan *Utilities*. Berikut ini merupakan gambar dari Command File.

II.3.3. Rendering

Rendering merupakan proses untuk melihat hasil akhir dari pekerjaan kita di 3Ds Max. Dalam *Rendering*, semua data-data yang sudah dimasukkan dalam proses *modeling*, animasi, *texturing*, dan pencahayaan dengan parameter tertentu akan diterjemahkan dalam sebuah bentuk *output* (tampilan akhir pada model dan animasi). Berikut ini merupakan gambar dari kotak dialog *Rendering*.

II.4. Pengenalan Adobe Flash CS6

AdobeFlash (dahulu bernama *MacromediaFlash*) adalah salah satu perangkat lunak komputer yang merupakan produk unggulan *AdobeSystems*. *Adobe* digunakan untuk membuat gambar vektor maupun animasi gambar tersebut. Berkas yang dihasilkan dari perangkat lunak ini mempunyai file *extension* .swf dan dapat diputar di penjelajah web yang telah dipasang *AdobeFlashPlayer*. *Flash* menggunakan bahasa pemrograman bernama *ActionScript* yang muncul pertama kalinya pada *Flash 5*. (Kriston Radion, S.ST:2012;2)

Sebelum tahun 2005, *Flash* dirilis oleh *Macromedia*. *Flash 1.0* diluncurkan pada tahun 1996 setelah *Macromedia* membeli program animasi vektor bernama *FutureSplash*. Versi terakhir yang diluncurkan di pasaran dengan menggunakan nama '*Macromedia*' adalah *MacromediaFlash 8*. Pada tanggal 3 Desember 2005 *AdobeSystems* mengakuisisi *Macromedia* dan seluruh produknya, sehingga nama *MacromediaFlash* berubah menjadi *AdobeFlash*. (Kriston Radion, S.ST:2012;2)

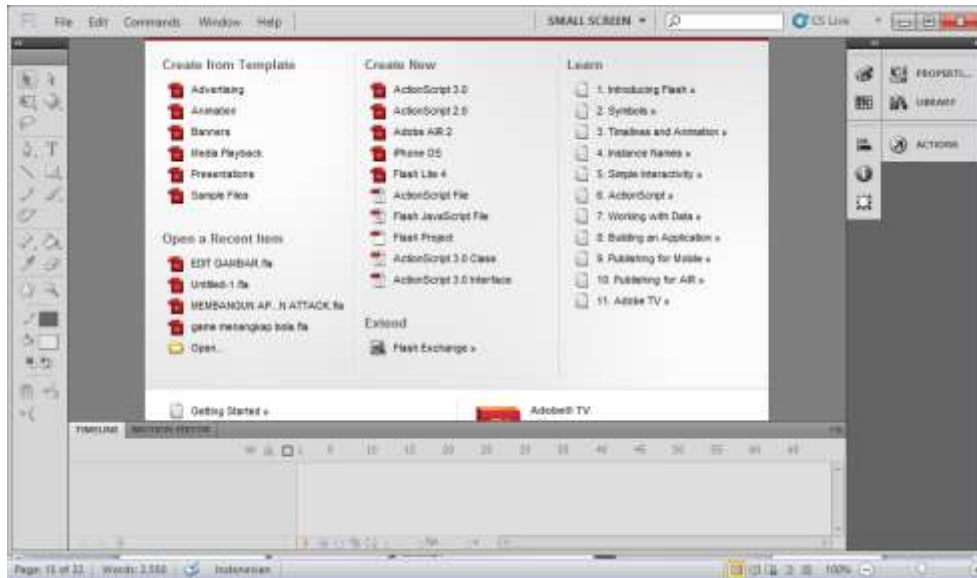
AdobeFlash merupakan sebuah program yang didesain khusus oleh *Adobe* dan program aplikasi standar *authoringtool* profesional yang digunakan untuk membuat animasi dan bitmap yang sangat menarik untuk keperluan pembangunan situs *web* yang interaktif dan dinamis. *Flash* didesain dengan kemampuan untuk membuat animasi 2 dimensi yang andal dan ringan sehingga *Flash* banyak digunakan untuk membangun dan memberikan efek animasi pada *website*, CD interaktif dan yang lainnya. Selain itu, aplikasi ini juga dapat digunakan untuk membuat animasi logo, *movie*, *game*, pembuatan navigasi pada situs *web*, tombol animasi, banner, *menuinteraktif*, *interaktiform* isian, *e-card*, *screensaver*, dan pembuatan aplikasi-aplikasi *web* lainnya. Dalam *Flash*, terdapat teknik-teknik membuat animasi, fasilitas *actionsript*, *filter*, *customeasing* dan

dapat memasukkan *video* lengkap dengan fasilitas *playback* FLV. Keunggulan yang dimiliki oleh *Flash* ini adalah ia mampu diberikan sedikit code pemrograman baik yang berjalan sendiri untuk mengatur animasi yang ada di dalamnya atau digunakan untuk berkomunikasi dengan program lain seperti HTML, PHP, dan Database dengan pendekatan XML, dapat dikolaborasi dengan *web*, karena mempunyai keunggulan antara lain kecil dalam ukuran file outputnya.(Kriston Radion, S.ST:2012;2-3).

Sebuah program grafis animasi standard profesional untuk menghasilkan produk-produk multi media seperti *Multimedia Persentation, wabsite, Computer Game, dan Animation*. Perogram ini mampu menghasilkan animasi yang begitu canggih, sehingga besar aplikasi tutorial yang interaktif, game, presentasi, dan lain-lain dibuat dengan program ini. Flase CS5.5 merupakan pengembangan dari penyempurnaan dari virsi sebelumnya (*Flash profesional* 2004). Ada beberapa panel pada plash yang harus diketahui sebagai dasar pembuatan animasi.

II.4.1. Area kerja macromedia Flash Player

Langkah untuk menjalankan program Adobe Flesh Pro CS5.5, tekan tombol **Start ► All Programs ► Adobe ► Adobe Flash CS5.5** sehingga tampil **Welcome Screen** seperti tanpak pada gambar berikut, (Madcoms Madium; 2012: 4).



Gambar II.3. Tampilan layar pertama program Adobe Flash Pro CS.5.5

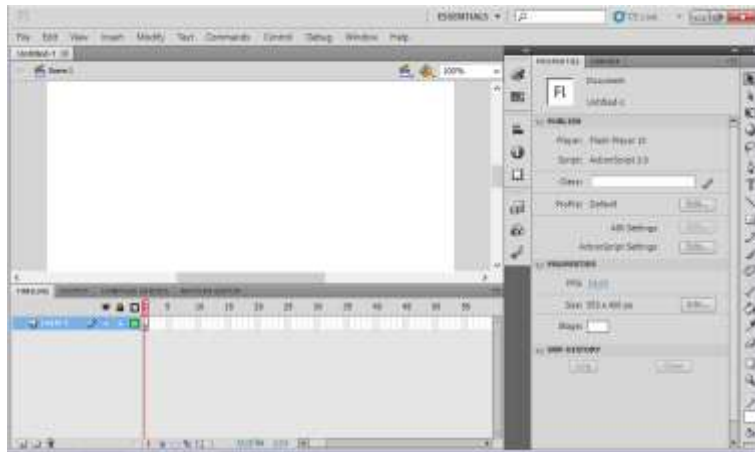
Sumber : (Madcoms Madium; 2012: 4)

Welcome Screen menampilkan empat pilihan perintah untuk memulai Adobe Flash Pro CS5.5, yaitu:

1. ***Create from Template***, berguna untuk membuka lembar kerja dengan template yang tersedia dalam program Adobe Flash Pro CS5.5.
2. ***Open a Recent Item***, berguna untuk membuka kembali file yang pernah Anda simpan atau pernah Anda buka sebelumnya.
3. ***Create New***, berguna untuk membuka lembar kerja baru dengan beberapa pilihan script yang tersedia.
4. ***Learn***, berguna untuk membuka jendela Help yang berguna untuk mempelajari suatu perintah, (Madcoms Madium; 2012: 4-5).

Jika Anda tidak ingin menampilkan jendela *Welcome Screen* lagi saat membuka program, aktifkan kotak periksa ***Don't Show again*** yang terdapat pada sisi bawah dari jendela

Welcome Screen. Sebagai contoh klik perintah **ActionScript 2.0** pada bagian Create New sehingga tampil lembar kerja seperti Gambar II.4.



Gambar II.4. Tampilan Jendela program Adobe Flash Pro CS5.5

Sumber : (Madcoms Madium; 2012: 5)

II.4.2. Toolbox

Toolbox adalah sebuah panel yang menampung tombol-tombol yang berguna untuk membuat suatu desain animasi mulai dari tombol seleksi, pen, pensil, Text, 3D Rotation, dan lain-lain



Gambar II.5. Tampilan Menu Bar

Suber : (Madcoms Madium; 2012: 6).

Dalam daftar Tabel II.2.berikut adalah simbol dan nama-nama tombol Toolbox:

Tabel II.1. Fungsi Tombol Toolbox

Nama Tombol	Fungsi
<i>Selection Tool (V)</i>	Untuk menyeleksi objek.
<i>Subselection Tool (A)</i>	Untuk menyeleksi bagian objek untuk proses editing.
<i>Free Transform Tool (Q)</i>	Untuk mengubah bentuk objek secara bebas.
<i>GradientTransform Tool(F)</i>	Untuk mengubah transformasi warna gradasi sebuah objek.
<i>3D Rotation Tool (W)</i>	Untuk melakukan transformasi bentuk dan posisi 3D pada objek berdasarkan sumbu X, Y dan Z.
<i>3D Translation Tool (G)</i>	Untuk melakukan transformasi bentuk dan posisi 3D pada simbol movie clip dengan acuan tiga sumbu X, Y dan Z.
<i>Losso Tool (L)</i>	Untuk menyeleksi objek dengan pola seleksi bebas.
<i>Pen Tool (P)</i>	Untuk menggambar objek
<i>Add Anchor</i>	Untuk menambah titik anchor pada sebuah path.

<i>Point Tool (=)</i>	
<i>Delete Anchor Point Tool (-)</i>	Untuk menghapus titik Anchor.
<i>Convert Anchor Point Tool (C)</i>	Untuk mengubah sudut lancip dari sebuah path menjadi sudut lengkung.
<i>Text Tool (T)</i>	Untuk mengetik text dan paragraf.
<i>Line Tool (N)</i>	Untuk menggambar objek garis lurus.
<i>Rectangle Tool (R)</i>	Untuk menggambar objek kotak.
<i>Oval Tool (O)</i>	Untuk menggambar objek oval atau lingkaran.
<i>RectanglePrimitive Tool (R)</i>	Untuk menggambar objek kotak dengan sudut dapat di lengkungkan.
<i>Oval Primitive Tool (O)</i>	Untuk menggambar objek lingkaran dengan berbagai variasi.
<i>PolyStar Tool</i>	Untuk menggambar objek poligon dan bintang.
<i>Pencil Tool (Y)</i>	Untuk menggambar dengan bentuk goresan pensil.
<i>Brush Tool (B)</i>	Untuk menggambar dengan bentuk polesan kuas.
<i>Spray Brush Tool (B)</i>	Untuk menggambar dengan spary, yaitu menyemprotkan warna atau simbol.
<i>Deco Tool (U)</i>	Untuk menggambar corak dekorasi dengan menggunakan simbol graphic.
<i>Bone Tool (X)</i>	Membuat animasi pertulangan dengan menggunakan titik sendi pada objek.
<i>Bind Tool (Z)</i>	Melakukan pengeditan dan modifikasi titik sendi dari piranti Bone Tool.
<i>Ink Bottle Tool (S)</i>	Untuk memberi warna dan bentuk garis outline pada sebuah objek.
<i>Paint Bucket Tool (K)</i>	Untuk memberi warna bidang objek.
<i>Eyedropper Tool (I)</i>	Untuk mengambil sample warna dari sebuah objek.
<i>Erasser Tool (E)</i>	Untuk menghapus bidang objek.
<i>Hand Tool (H)</i>	Untuk menggeser area lembar kerja atau stage.
<i>Zoom Tool (M,Z)</i>	Untuk memperbesar atau memperkecil tampilan lembar kerja atau stage.
<i>Stroke Color</i>	Untuk menentukan warna garis.
<i>Fill</i>	Untuk menentukan warna bidang objek.

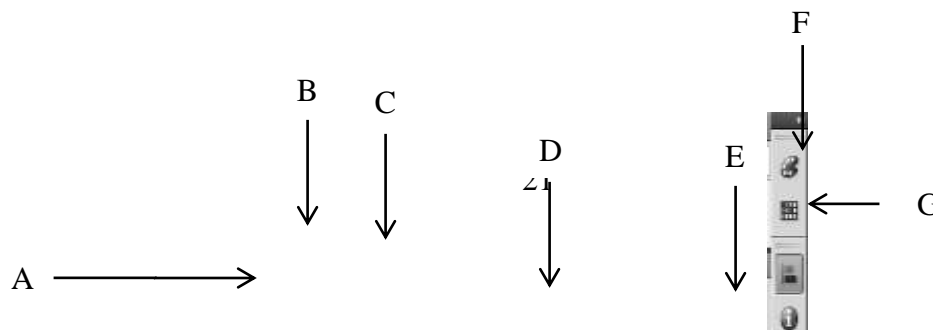
<i>Color</i>	
<i>Black and White</i>	Untuk mengubah warna garis dan bidang menjadi hitam dan putih.
<i>Swap Colors</i>	Untuk membalik warna antara warna garis dan warna bidang objek.
<i>No Color</i>	Unruk menghapus warna garis atau warna bidang objek.
<i>Snap to Objects</i>	Untuk mengaktiskan atau mematikan fungsi Snap to Objects.

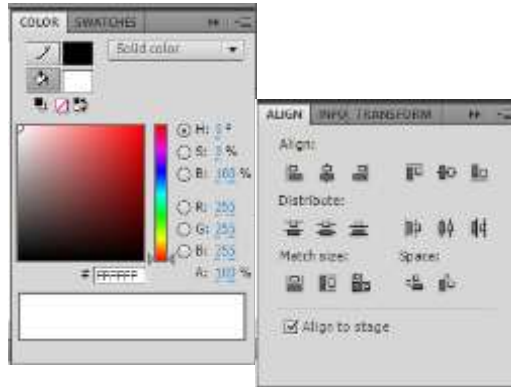
Sumber : (Madcoms Madium; 2012: 6-9)

II.4.3. Tampilan Panel

Panels, berisi kontrol fungsi yang di pakai dalam flash, yang berfungsi untuk mengganti dan memodifikasi berbagai atribut dari objek atau animasi secara cepat dan mudah. Panels biasanya terletak di bagian kanan area Flash. Untuk menampilkan panel tersebut, klik menu *Window*>(nama panel).

Program Adobe Flash Pro CS5.5 menampilkan lembar kerja yang sangat menarik serta dapat di ubah menurut selera Anda. Dengan perubahan tersebut, Anda lebih mudah mengatur dan menyusun objek yang akan dianimasikan. Perhatikan gambar berikut serta keterangannya untuk mengatur tampilan lembar kerja Adobe Flash CS5.5,(Madcoms Madium; 2012: 14).





Gambar II.6. Tampilan Panel

Sumber : Madcoms Madium; 2012: 14

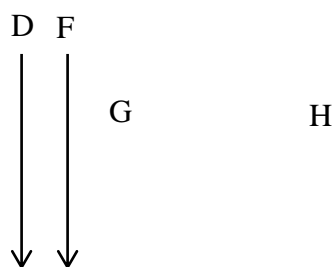
Tabel II.2. Keterangan Tampilan Panel

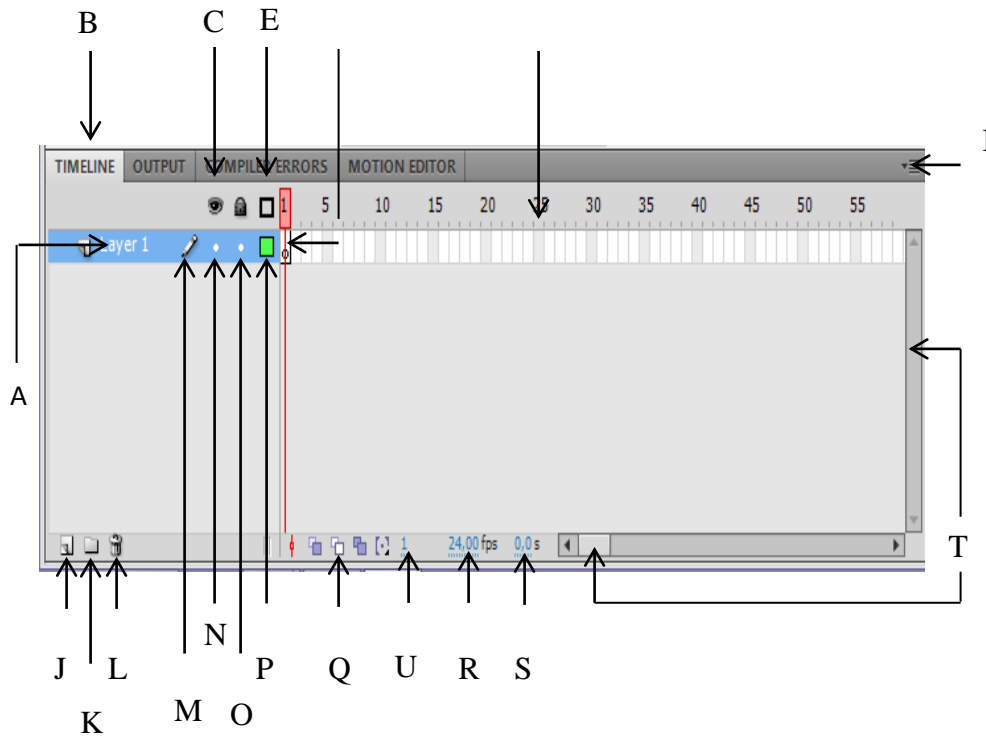
Abjad	Keterangan
A	Klik tahan untuk mengatur lembar panel.
B	Klik untuk menutup uraian panel aktif.
C	Klik untuk menutup panel aktif.
D	Klik dan geser keluar untuk melepas panel dari grupnya.
E	Klik untuk menutup uraian panel kedalam dock panel.
F	Klik untuk menguraikan semua dock panel.
G	Klik ikon panel untuk menguraikan panel dock terpilih.

Sumber : (Madcoms Madium; 2012: 14-15)

II.4.4. Timeline

Timeline berguna untuk menentukan durasi animasi, jumlah *layer*, *frame*, menempatkan *script* dan beberapa keperluan animasi lainnya. Semua bentuk animasi yang Anda buat akan diatur dan ditempatkan pada layer dalam timeline.





Gambar II.7. Tampilan Timeline Animation

Sumber : (Madcoms Madium; 2012: 9)

Tabel II.4. Keterangan Tampilan Timeline

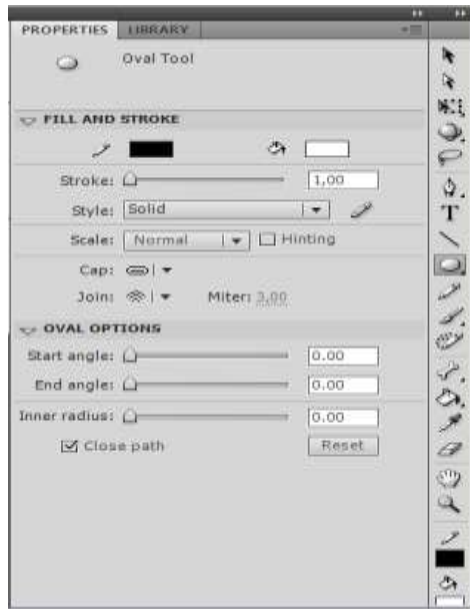
Abjad	Nama	Keterangan
A	<i>Layer</i>	Layar kerja yang menampung objek yang akan dianimasika di dalam Timeline.
B	<i>Timeline</i>	Tabulasi dari lembar kerja atau Stage yang sedang dikerjakan.
C	<i>Show/Hide All Layers</i>	Untuk menyembunyikan atau menampilkan semua isi layer.
D	<i>Lock/Unlock All Layers</i>	Untuk mengunci atau melepas kunci objek dari semua layer.
E	<i>Show All Layer as outlines</i>	Untuk menampilkan objek pada semua layer dalam bentuk outline.
F	<i>Playhead</i>	Jarum untuk membaca Frame pada saat animasi dijalankan.
G	<i>Blang Keyframe</i>	Sebuah simbol lingkaran kosong yang menampung suatu objek.
H	<i>Frame</i>	Suatu bagian dari layer yang digunakan untuk mengatur pembuatan animasi.
I	<i>Menu</i>	Untuk mengatur tampilan Frame.
J	<i>New Layer</i>	Untuk menambahkan layer baru.

K	<i>New Folder</i>	Untuk menambahkan folder baru.
L	<i>Delete</i>	Untuk menghapus layer
M	<i>Simbol Pensil</i>	Menunjukkan bahwa layer dalam kondisi terpilih atau aktif.
N	<i>Titik Show or Hide</i>	Klik untuk menampilkan atau menyembunyikan layer aktif.
O	<i>Titik Kunci</i>	Klik untuk mengunci atau melepaskan kunci layer yang aktif.
P	<i>Kotak Outline</i>	Klik untuk menampilkan objek dalam layer aktif menjadi bentuk outline.
Q	<i>Onion skinning button</i>	Untuk mengatur tampilan animasi didalam stage.
R	<i>Frame Rate</i>	Untuk mengatur kecepatan gerak animasi dalam tiap detiknya.
S	<i>Elapsed Time</i>	Menunjukkan durasi atau lamanya animasi.
T	<i>Scrollbar</i>	Menggulung jendela Timeline secara vertikal dan horisontal.
U	<i>Current Frame</i>	Menunjukkan posisi Frame aktif.

Sumber : (Madcoms Madium; 2012: 9-10)

II.4.5. Panel Properties

Panel Properties berguna untuk menampilkan parameter dari sebuah tombol yang terpilih sehingga Anda dapat memodifikasi dan memaksimalkan fungsi dari tombol tersebut. Panel Properties menampilkan parameter sesuai dengan tombol yang terpilih.

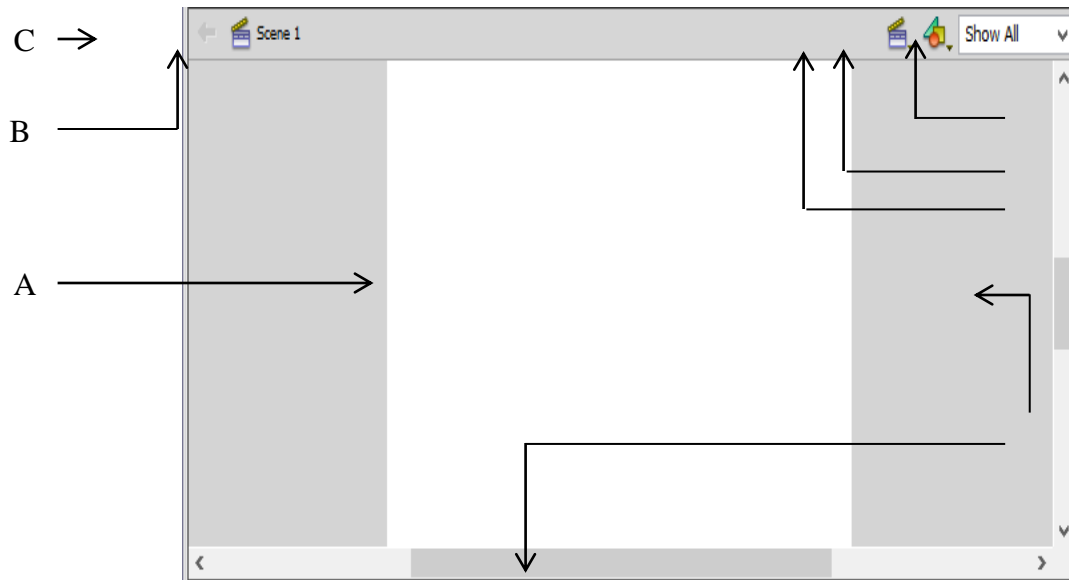


Gambar II.8. Tampilan Praperties

Sumber : (Madcoms Madium; 2012: 12)

II.4.6. Stage

Stage adalah lembar kerja yang di gunakan untuk membuat atau mendesain objek yang akan dianimasikan. Objek yang dibuat dalam lembar kerja dapat berupa bjek Vektor, Movie clip, Text, Button, dan lain-lain, (Madcoms Madium; 2012: 11).



Gambar II.9. Tampilan Stage

Sumber : (Madcoms Madium; 2012: 11)

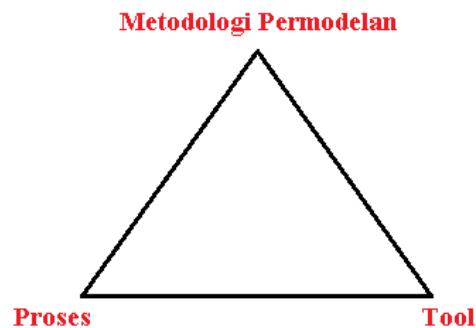
Tabel II.5. Keterangan Tampilan Stage

Abjad	Keterangan
A	Stage , lembar kerja untuk menyusun objek yang akan dianimasikan.
B	Scene , menunjukkan nama scene yang aktif.
C	Panah yang digunakan untuk berpindah dari lembar kerja simbol ke lembar kerja utama.
D	Edit Scene , untuk memilih nama scene yang akan diedit.
E	Edit Symbols , untuk memilih nama simbol yang akan diedit.
F	Zoom , untuk mengatur besarnya tampilan stage atau lembar kerja.
G	Scrolber , untuk menggulung lembar kerja secara horisontal dan vertikal.

Sumber : (Madcoms Madium; 2012: 11)

II.5. UML (*Unified Modelling Language*)

Pemodelan (*modeling*) adalah proses merancang piranti lunak sebelum melakukan pengkodean (*coding*). Model piranti lunak dapat dianalogikan seperti pembuatan blueprint pada pembangunan gedung. Membuat model dari sebuah sistem yang kompleks sangatlah penting karena kita tidak dapat memahami sistem semacam itu secara menyeluruh. Semakin kompleks sebuah sistem, semakin penting pula penggunaan teknik pemodelan yang baik. Dengan menggunakan model, diharapkan pengembangan piranti lunak dapat memenuhi semua kebutuhan pengguna dengan lengkap dan tepat, termasuk faktor-faktor seperti *scalability*, *robustness*, *security*, dan sebagainya. Kesuksesan suatu pemodelan piranti lunak ditentukan oleh tiga unsur, yang kemudian terkenal dengan sebuah segitiga sukses (*the triangle for success*). Ketiga unsur tersebut adalah metode pemodelan (*notation*), proses (*process*) dan *tool* yang digunakan. Memahami notasi pemodelan tanpa mengetahui cara pemakaian yang sebenarnya (proses) akan membuat proyek gagal. Dan pemahaman terhadap metode pemodelan dan proses disempurnakan dengan penggunaan tool yang tepat.

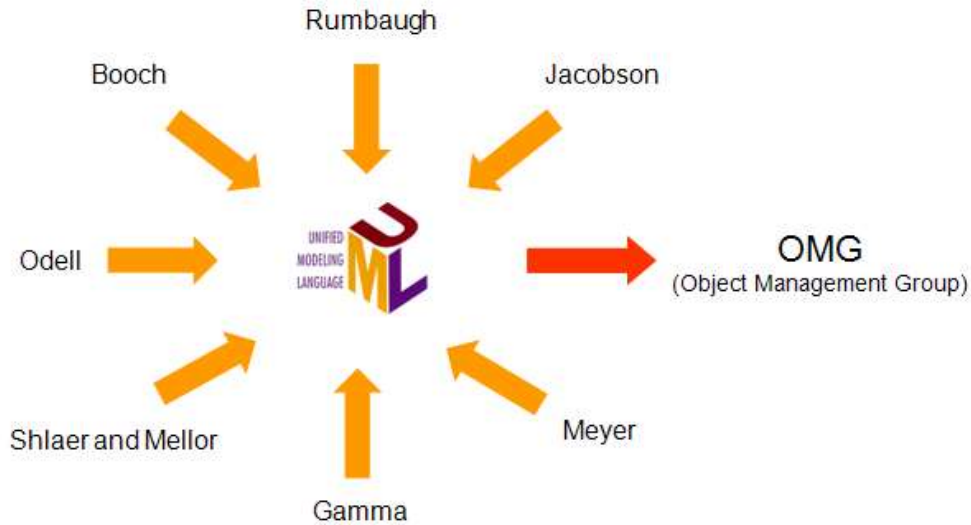


Gambar II.10. *The Triangle For Success*
Sumber : (Yuni Sugiarti ; 2013 : 33)

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut

dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*). Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: *metodologi booch*, *metodologi coad*, *metodologi OOSE*, *metodologi OMT*, *metodologi shlaer-mellor*, *metodologi wirfs-brock*, dsb. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan. Dimulai pada bulan Oktober 1994 *Booch*, *Rumbaugh* dan *Jacobson*, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan memelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management Group (OMG – <http://www.omg.org>). Tahun 1997 UML versi 1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003. Booch,

Rumbaugh dan Jacobson menyusun tiga buku serial tentang UML pada tahun 1999. Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek.



Gambar II.11. Metodologi Pemodelan Berorientasi Objek
Sumber : (Yuni Sugiarti; 2013 : 33)

Dalam hal ini, pemodelan perangkat lunak bekerja dengan cara yang cukup serupa layaknya seorang arsitek atau insinyur teknik sipil yang akan membuat sebuah bangunan/dedung berskala besar. Saat seorang arsitek atau insinyur teknik sipil akan membuat sebuah bangunan gedung berskala besar, ia biasanya membuat denah-denang atau maket-maket yang menggambarkan bentuk jadi dari bangunan/gedung.

Dari berbagai penjelasan rumit yang terdapat di dokumen dan buku-buku UML. Sebenarnya konsepsi dasar UML bisa kita rangkumkan dalam gambar dibawah.

Tabel II.6 Konsep Dasar UML

<i>Major Area</i>	<i>View</i>	<i>Diagram</i>	<i>Main Concept</i>
<i>Structural</i>	<i>Static View</i>	<i>Class Diagram</i>	<i>Class, Association, Generalization, Dependency, Realization, Interface</i>

	<i>Use Case View</i>	<i>Use Case Diagram</i>	<i>Use Case, Actor, Association, Extend, Include, Use Case Generalization</i>
	<i>Implementation view</i>	<i>Component Diagram</i>	<i>Component, Interface, Dependency, Location</i>
	<i>Deployment view</i>	<i>Deployment Diagram</i>	<i>Node, Component, Depedency, Location</i>
	<i>Interaction View</i>	<i>Sequence Diagram</i>	<i>Interaction, Object, Message, Activation</i>
		<i>Class Diagram</i>	<i>Collaboration, interaction, Collaboration Role, Message</i>
<i>Model Management Extensibility</i>	<i>Model Management View All</i>	<i>Class Diagram All</i>	<i>Package, Subsystem, Model Constraint, Stereotype, Tagged Values</i>

Sumber :(Yuni Sugiarti ; 2013 : 35)

Seperti juga tercantum pada gambar diatas UML mendefinisikan diagram-diagram sebagai berikut:

1. *Use case diagram*
2. *Class diagram*
3. *Statechart diagram*
4. *Activity diagram*
5. *Sequence diagram*
6. *Collaboration diagram*
7. *Component diagram*


8. *Deployment diagram*

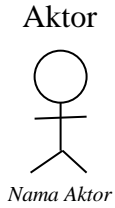
Dalam pembuatan skripsi ini penulis menggunakan diagram Use Case yang terdapat di dalam UML. Adapun maksud dari Use Case Diagram diterangkan dibawah ini.

II.5.1. *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. *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. 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.

Tabel II.7. Simbol *Use Case*

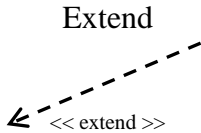
Simbol	Deskripsi
	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit dan aktor, biasanya menggunakan kata kerja di awal frase nama <i>use case</i>



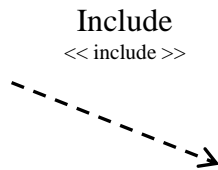
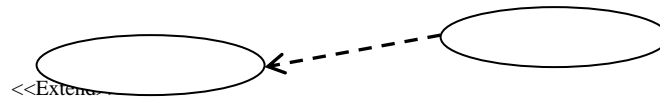
Orang atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat sistem itu sendiri. Jadi walaupun simbol aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. Biasanya menggunakan kata benda di awal frase nama aktor

Asosiasi / Association

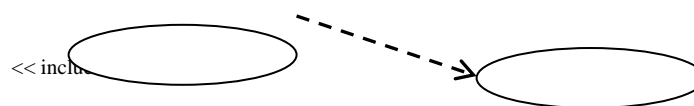
Komunikasi antar aktor dan *use case* yang berpartisipasi *use case* atau *use case* berinteraksi dengan aktor.



Relasi *usecase* tambahan ke sebuah *usecase* dimana *use case* yang ditambahkan dapat berdiri sendiri walupun tanpa *usecase* tambahan itu, mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek; biasanya *usecase* tambhan memiliki nama depan yang sama dengan *usecase* yang ditambahkan, arah panah menunjukan pada *usecase* yang dituju. Contoh :



Relasi *usecase* tambahan ke sebuah *usecase* dimana *usecase* yang ditambahkan memerlukan *usecase* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *usecase* ini. ada dua sudut pandang yang cukup besar mengenai *include* di *usecase*, *include* berarti *usecase* yang ditambahkan akan selalu dipanggil saat *usecase* tambhan dijalankan, contoh;



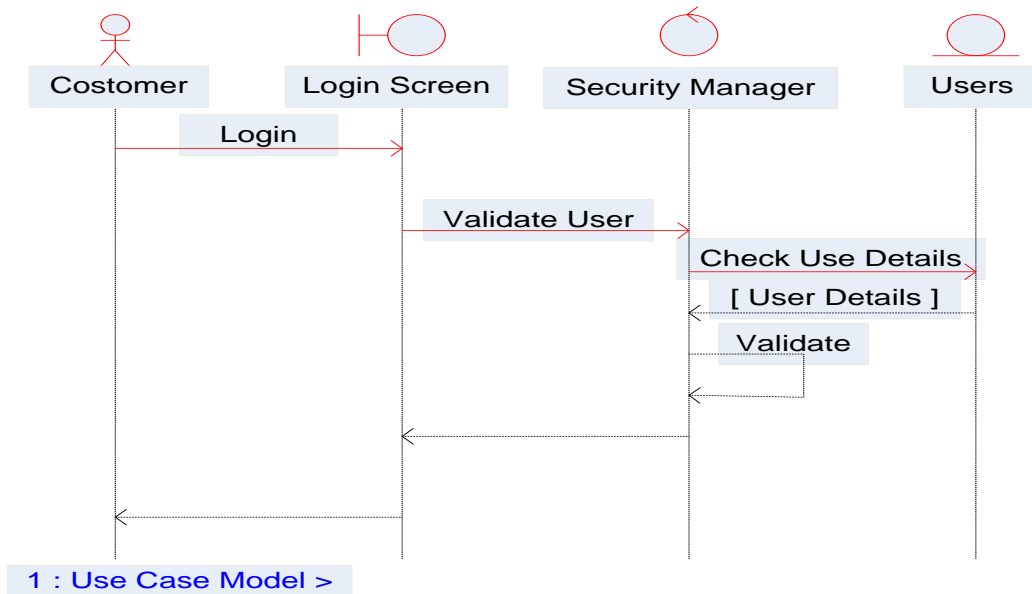
Sumber :(Yuni Sugiarti ; 2013 : 42)

II.5.2. Sequence Diagram

Diagram *Sequencemeng*gambarkan kelakuan/prilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *sequencemaka* harus diketahui objek-objek yang

terlibat dalam sebuah *usecase* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram *sequence* yang harus digambar adalah sebanyak pendefinisian *usecase* yang memiliki proses sendiri atau yang penting semua *usecase* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram *sequence* sehingga semakin banyak *usecase* yang didefinisikan maka diagram *sequence* yang harus dibuat juga semakin banyak.



Gambar

II.11. Contoh Sequence Diagram

Sumber : (Yuni Sugiarti ; 2013 : 63)

II.5.3. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

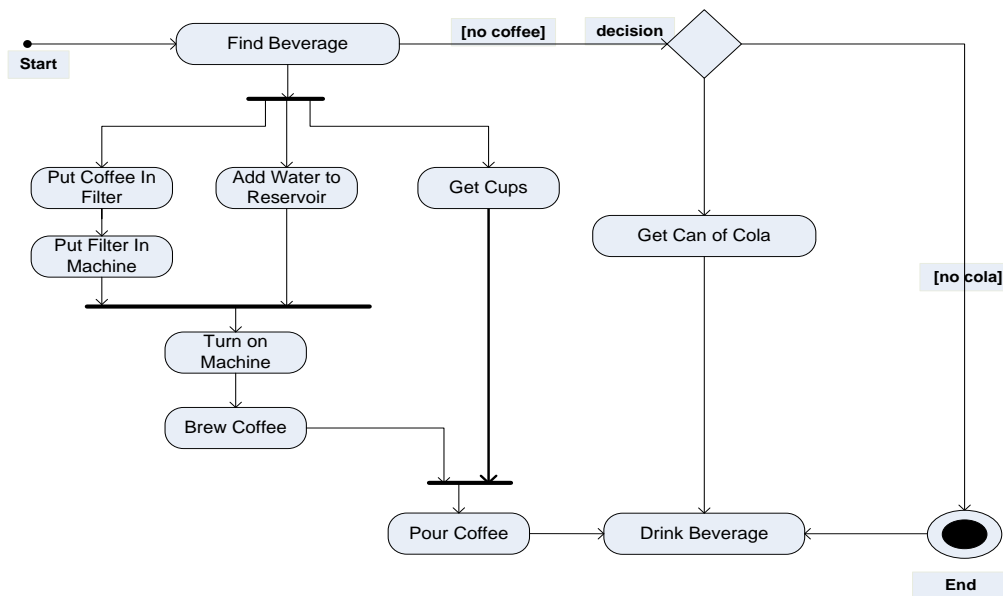
Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal*

processing). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Activity diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.



Gambar II.12. Activity Diagram
 Sumber : (Yuni Sugiarti ; 2013 : 76)