

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Perancangan**

Perancangan adalah kegiatan awal dari suatu rangkaian kegiatan dalam proses pembuatan produk. Dalam tahap perancangan tersebut dibuat keputusan-keputusan penting yang mempengaruhi kegiatan-kegiatan lain yang menyusulnya. (Harsokusoemo; 2000:1). Perancangan yang dimaksud adalah sebuah proses membuat beberapa *output* media komunikasi *visual* yang didasari adanya sebuah kebutuhan atau suatu permasalahan, mulai dari perencanaan, pengumpulan dan analisa data hingga membuat desain yang *efisien* dan sesuai tujuan, (Adithya Yustanto; 2011: 3).

#### **II.2. Pengertian Animasi**

Film animasi berasal dari dua disiplin ilmu, yaitu film yang berakar pada dunia fotografi dan animasi yang berakar pada dunia gambar. Animasi dipandang sebagai suatu hasil proses dimana obyek-obyek yang digambarkan atau divisualisasikan tampak hidup. Gambar digerakkan melalui perubahan sedikit demi sedikit dan teratur sehingga memberikan kesan hidup.

Animasi dipandang sebagai salah satu hasil proses dimana obyek-obyek yang digambarkan atau divisualisasikan tampak hidup. Kehidupan tersebut dapat dinyatakan dari suatu pergerakan. Meskipun demikian animasi tidak secara jelas dinyatakan pada obyek-obyek mati yang kemudian digerakkan. Benda-benda mati, gambaran, deformasi bentuk yang digerakkan memang dapat dikatakan suatu bentuk animasi, akan tetapi esensi dari animasi tidak sebatas pada unsure menggerakkaniru sendir, jika kehidupan memang didentikkandengan

pergerakan, maka kehidupan itu sendiri juga mempunyai karakter kehidupan.(Cahaya Budi Wibawa, 2011)

Dalam dunia penyiaran ada ketentuan dalam penentuan resolusi animasi. Resolusi tersebut berpengaruh pada frame per *secondnya*. menurut NTSC (*Nationaltelevition Standard Comitee*) ukuran dasar yang digunakan atar *frame per second* adalah 24 *frame per second* (24fps).(Chabib Syafrudin ; 2013: 389)

### 1. Jenis Film Animasi

- a. Animasi 2 Dimensi (2D)
- b. Animasi 3 Dimensi (3D)
- c. Animasi Tanah Liat (*ClayAnimation*)
- d. Animasi Jepang (*Anime*)

### 2. Bentuk Film Animasi

- a. Film *Spot* (10 sampai 60 detik )
- b. Film *Pocket* Cartoon (50 detik sampai 2 menit)
- c. Film Pendek (2 sampai 20 menit)
- d. Film Setengah Panjang (20 sampai 50 menit)
- e. Film Panjang (minimal 50 menit)

### 3. Proses Pembuatan Produksi Film Animasi

- a. Pembuatan
- b. Tema
- c. *Logline*
- d. Sinopsis
- e. *CharacterDevelompment*

- f. Membuat *storyboard*
- g. Membuat gambar
- h. Pewarnaan *digital (coloring)*
- i. *Dubbing*
- j. Konversi ke CD

### **II.2.1. Sejarah Animasi**

Animasi merupakan suatu seni untuk membuat dan menggerakkan sebuah obyek, baik berbentuk 2 dimensi maupun 3 dimensi dan dibuat dengan berbagai cara, misalnya menggunakan kertas, komputer dan lain sebagainya. Animasi saat ini telah menjadi industri besar yang memberikan dampak ekonomi dan sosial yang begitu besar bahkan cukup signifikansi terhadap pendapat sebuah negara.

Animasi dari kontibusi hasil karya seorang yang bernama Walt Disney yang lahir dengan nama Walter alias Disney. Walt Disney mempunyai ide untuk membuat animasi bergerak dengan seekor tikus.

Dari proses tersebut maka lahirlah Mickey Mouse yang merupakan cikal bakal dari kartun animasi pertama yang pernah dibuat. Awal mula bentuk mickey masih sederhana. Dengan beberapa cirri tikus yang dimanipulasi seperti bentuk telinga dibuat bulat, bentuk mata yang dibuat besar hamper sebesar wajahnya, dan bentuk tubuh yang kurus ditambah atribut celana pendek yang dibuat sederhana sehingga memudahkan untuk memproduksinya.(Cahya Budi Wibawa, 2011)

## **II.3. 3 Dimensi**

### **II.3.1. Konsep Dasar Objek 3D**

Objek 3D adalah representasi dari data geometrik 3 dimensi sebagai hasil dari pemrosesan dan pemberian efek cahaya terhadap grafika komputer 2d. Hasil ini kadang kala ditampilkan secara waktu nyata (real time) untuk keperluan simulasi. Secara umum prinsip yang dipakai adalah mirip dengan objek 2d, dalam hal: penggunaan algoritma, grafika vektor, model frame kawat (wire frame model), dan grafika rasternya.

Objek 3D sering disebut sebagai model 3D. Namun, model 3D ini lebih menekankan pada representasi matematis untuk objek 3 dimensi. Data matematis ini belum bisa dikatakan sebagai gambar grafis hingga saat ditampilkan secara visual pada layar komputer atau printer. Proses penampilan suatu model matematis ke bentuk citra 2d biasanya dikenal dengan proses 3D rendering (Angga Prasetio Romadhon ; 2013 : 222)

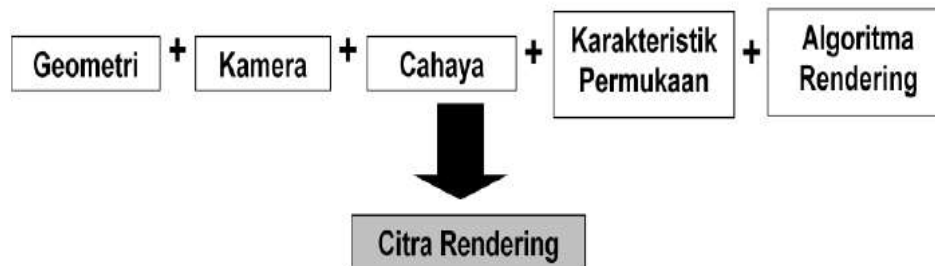
### **II.3.2. Konsep Dasar Rendering Objek 3D**

Rendering merupakan sebuah proses untuk menghasilkan sebuah citra 2D dari data 3D. Proses ini bertujuan untuk memberikan visualisasi pada user mengenai data 3D tersebut melalui monitor atau pencetak yang hanya dapat menampilkan data 2D.

Gambar yang dibuat melalui proses rendering digital adalah gambar digital atau raster image, jenis gambar yang sama dengan yang biasa kita lihat sehari-hari pada desktop komputer atau wallpaper. Gambar digital tersebut dibuat melalui proses rendering digital sebagai langkah besar terakhir sebelum disusun menjadi animasi. Animasi sebagai tujuan akhir biasa digunakan dalam film, video game, permainan komputer, simulator, dan untuk efek khusus di televisi. Masing-masing menggunakan proses rendering digital yang menggunakan fitur dan teknik berbeda untuk mencapai hasil yang diinginkan. (Angga Prasetio Romadhon ; 2013 : 223)

Secara umum, proses untuk menghasilkan rendering dua dimensi dari objek-objek 3D melibatkan 5 komponen utama :

- a. Geometri
- b. Kamera
- c. Cahaya
- d. Karakteristik Permukaan
- e. Algoritma Rendering



**Gambar II.1. Skema Proses Rendering**

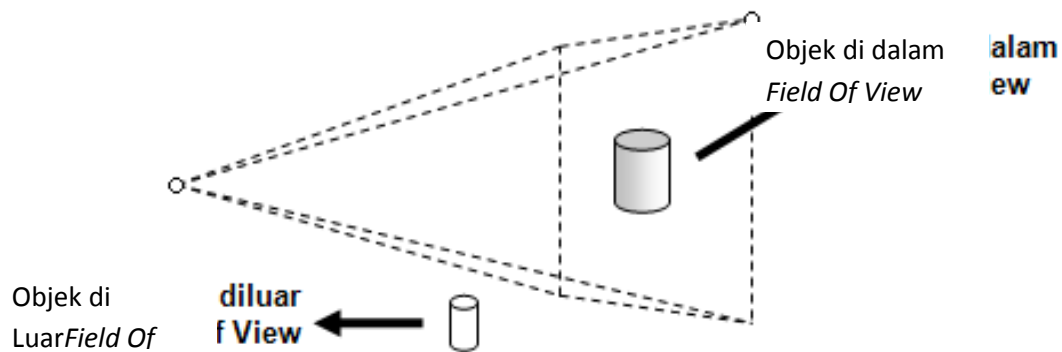
*Sumber : (Angga Prasetyo Romadhon ; 2013 : 223)*

### **II.3.3. Kamera**

Dalam grafika 3D, sudut pandang (*point of view*) adalah bagian dari kamera. Kamera dalam grafika 3D biasanya tidak didefinisikan secara fisik, namun hanya untuk menentukan sudut pandang kita pada sebuah world, sehingga sering disebut virtual camera.

Pada kamera, dikenal field of view yaitu daerah yang terlihat oleh sebuah kamera. Field of view pada grafika 3D berbentuk piramid, karena layar monitor sebuah komputer berbentuk segiempat. Objek-objek yang berada dalam field of view ini akan terlihat dari layar monitor, sedang objek-objek yang berada di luar field of view ini tidak terlihat pada layar monitor. Field of view ini sangat penting dalam pemilihan objek yang akan diproses dalam rendering. Objek-

objek diluar field of view biasanya tidak akan diperhitungkan, sehingga perhitungan dalam proses rendering, tidak perlu dilakukan pada seluruh objek. (Angga Prasetio Romadhon ; 2013 : 223)



**Gambar II.2. Sudut *Field of view***

**Sumber : (Angga Prasetio Romadhon ; 2013 : 223)**

#### **II.3.4. Cahaya**

Sumber cahaya pada grafika 3D merupakan sebuah objek yang penting, karena dengan cahaya ini sebuah world dapat terlihat dan dapat dilakukan proses rendering. Sumber cahaya ini juga membuat sebuah world menjadi lebih realistis dengan adanya bayangan dari objek-objek 3D yang ada.

Sebuah sumber cahaya memiliki jenis. Pada grafika 3D dikenal beberapa macam sumber cahaya, yaitu :

##### **1. *Pointlight***

Memancar ke segala arah, namun intensitas cahaya yang diterima objek bergantung dari posisi sumber cahaya. Tipe ini mirip seperti lampu pijar dalam dunia nyata.

## **2. *Spotlight***

Memancarkan cahaya ke daerah tertentu dalam bentuk kerucut. Sumber cahaya terletak pada puncak kerucut. Hanya objek-objek yang terletak pada daerah kerucut tersebut yang akan nampak.

## **3. *Ambientlight***

Cahaya latar/alam. Cahaya ini diterima dengan intensitas yang sama oleh setiap permukaan pada benda. Cahaya latar tersebut dimodelkan mengikuti apa yang terjadi di alam, dalam keadaan tanpa sumber cahaya sekalipun, benda masih dapat dilihat.

## **4. *Directionallight***

Memancarkan cahaya dengan intensitas sama ke suatu arah tertentu. Letak tidak mempengaruhi intensitas cahayanya. Tipe ini dapat menimbulkan efek seolah-olah sumber cahaya berada sangat jauh dari objek.

## **5. *Parallelpont***

Sama halnya dengan directional light, hanya pencahayaan ini memiliki arah dan posisi.

## **II.4. Istana Maimun**

Istana Maimun merupakan ikon sekaligus kebanggaan warga Medan. Istana warna kuning yang berlokasi di Jalan Brigjen Katamso-Medan, terletak di Kelurahan Sukaraja

kecamatan Medan Maimun, Kota Medan. Istana Maimun yang merupakan peninggalan Kerajaan Deli ini dibangun pada 26 Agustus 1888 dan baru diresmikan pada 18 Mei 1891. Bangunan Istana menghadap ke timur dimana terdiri dari dua lantai dengan tiga bagian yaitu bangunan induk, sayap kiri dan sayap kanan. Di bagian depan sekitar 100 meter berdiri Masjid Al-Maksum yang dikenal sebagai Masjid Raya Medan.

Istana yang memiliki luas 2.772 m<sup>2</sup> ini mempunyai arsitektur yang unik dengan perpaduan beberapa unsur kebudayaan Melayu bergaya Islam, Spanyol, India dan Itali. Perpaduan ini menyuguhkan keunikan inilah yang memberikan karakter khas bangunannya. Istana Maimun merupakan salah satu dari sekian istana yang paling indah di Indonesia.

Kemewahan interior dan bangunan fisik Istana Maimun ini dimungkinkan karena sejak 2 abad silam wilayah Deli berada dibawah Kesultanan Deli yang mengelola hasil perkebunan, minyak, dan rempah-rempah yang melimpah. Hasil bumi yang luar biasa tersebut memberikan penghasilan sangat besar kepada Kesultanan Deli dengan bukti kehadiran Istana Maimun yang megah.([www.pusakaindonesia.org](http://www.pusakaindonesia.org))

#### **II.4.1. Sejarah Istana Maimun**

Istana Maimun merupakan istana peninggalan kerajaan Deli yang dipimpin Sultan Al Rasyid Perkasa Alamsyah pada tahun 1973. Istana Maimun sempat ditempati oleh 4 Sultan Melayu yang memerintah saat itu. Istana ini dibangun pada tahun 1888 yang diarsiteki oleh TH Van Erp yang bekerja juga sebagai Konijnlijk *Nederlands-Indische Leger* (KNIL), atau tentara Kerajaan Hindia-Belanda.

Istana Maimun sebuah bangunan peninggalan sejarah masa kerajaan melayu Sultan Deli ke- IX mulai dibangun pada 26 Agustus 1888 dan selesai selama 3 tahun yang sekaligus

diresmikan pada tanggal 18 Mei tahun 1891. Bangunan begitu kokoh dan megah hingga saat ini didesain oleh arsitektur asal Italia yang bernama Ferari. Pembangunan ini menghabiskan dana setara satu juta gulden jika dikurskan dengan mata uang Belanda, konsep arsitekturnya unik, cantik, dan memiliki karakter unsur tradisional yang khas Indonesia dengan sentuhan Melayu, baik bentuk maupun ornamennya dipengaruhi oleh berbagai kebudayaan, antara lain Melayu, Islam, Spanyol, China, India dan Itali. Bangunan ini juga didominasi dengan warna kuning keemasan yang identik dengan etnis Melayu. Bangunan dengan dua lantai yang memiliki luasnya 772 m<sup>2</sup> dan mempunyai 30 bilik (kamar) yang didalamnya terdapat berbagai macam perabotan dengan gaya Eropa, seperti lemari, kursi dan lampu-lampu Kristal. Bangunan dua lantai ini dibagi menjadi tiga bagian ruangan.: Ruang Utama, sayap kanan dan sayap kiri. Ruang Utama atau ruangan induk disebut dengan Balairung Sri yang luasnya 412 m<sup>2</sup> ruangan ini biasanya digunakan untuk acara-acara adat kerajaan, menerima tamu ataupun acara penobatan Sultan Deli, ruangan ini dihiasi dengan koleksi peninggalan-peninggalan jaman dahulu seperti senjata tua dan foto-foto keluarga, selain itu pada bagian belakang ada dapur, gudang dan ruangan penjara.

Keunikan perpaduan tradisi Melayu dengan kebudayaan Eropa pada bangunan interiornya. Sedangkan *influence* Islam dapat terlihat dari bentuk kurva di beberapa bagian atap istana. Kurva yang berbentuk seperti kapal terbalik atau yang kerap dikenal dengan Persia Curve yang sering dijumpai pada bangunan di kawasan Timur Tengah, India, atau Turki.

Banyak barang-barang peninggalan kerajaan Deli seperti foto-foto, baju adat, maupun senjata kerajaan Deli dipajang di dalam Kompleks istana ini. Masih didalam Kompleks istana terdapat tempat Meriam Putung yang memiliki nilai mistik sendiri bagi orang Medan. Kono meriam ini merupakan jelmaan dari adik Putri Hijau yang berasal dari Kerajaan Deli Tua. Sang Putri menjelma di saat terjadi perang antara kerajaan Aceh dan Melayu. Hal ini karena pinangan

Raja Aceh ditolak oleh Putri Hijau. Pecahan dari meriam ini tersimpan di Kabupaten Karo, Sumatera Utara.(www.pusakaindonesia.org)

## **II.5. Multimedia**

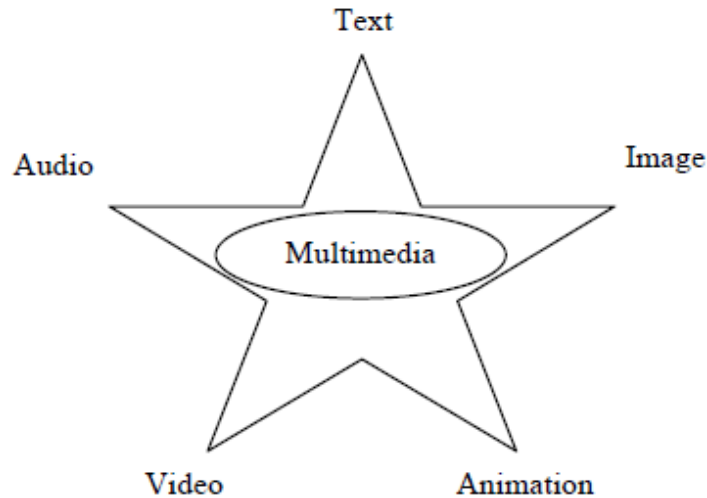
### **II.5.1. Pengertian multimedia**

Multi-banyak, Media-sarana berkomunikasi untuk melewatkan informasi. Suatu sistem yang terdiri dari perangkat keras, perangkat lunak dan alat – alat lain seperti televisi, monitor video dan sistem piringan optik atau sistem stereo yang dimaksudkan untuk menghasilkan penyajian audio visual yang utuh. Beberapa pakar mengartikan multimedia sebagai berikut :

1. Multimedia secara umum merupakan kombinasi 3 elemen yaitu suara, gambar dan teks.
2. Multimedia adalah kombinasi dari paling sedikit 2 media input atau output dari data, media ini dapat audio (suara, musik), animasi, video, teks, grafik, dan gambar.
3. Multimedia merupakan alat yang dapat menciptakan prestasi yang dinamis dan intraktif yang mengkombinasikan teks grafik, animasi, audio dan gambar video.
4. Multimedia adalah pemanfaatan komputer untuk membuat dan menggabungkan teks, grafik, audio, gambar bergerak (video dan animasi) dengan menggabungkan link dan tool yang memungkinkan pemakai melakukan navigasi, berintraksi, berkreasi dan berkomunikasi.

### **II.5.2. Kelebihan Multimedia**

Dari berbagai media informasi, multimedia memiliki suatu kelebihan tersendiri yang tidak dapat digantikan oleh penyajian media informasi lainnya. Kelebihan dari multimedia adalah menarik indra dan menarik minat, karena merupakan gabungan antara pandangan, suara dan gerakan. Multimedia terbagi dalam beberapa element-element multimedia, seperti yang terlihat dalam gambar dibawah ini. (Chrisna Atmadji ; 2010 : 59)



**Gambar II.3. Komponen Multimedia**

*Sumber : (Chrisna Atmadji ; 2010 : 60)*

#### **II.5.2.1. Teks**

Bentuk data multimedia yang paling mudah disimpan dan dikendalikan adalah teks. Teks dapat membentuk kata, surat atau narasi dalam multimedia yang menyajikan bahasa.

#### **II.5.2.2. Image (grafik)**

Alasan untuk menggunakan gambar dalam presentasi atau publikasi multimedia adalah karena lebih menarik perhatian dan dapat mengurangi kebosanan dibandingkan dengan teks. Gambar dapat meringkas menyajikan data yang kompleks dengan cara yang baru dan lebih berguna.

#### **II.5.2.3. Bunyi (audio)**

PC multimedia tanpa bunyi hanya disebut unimedia, bukan multimedia. Bunyi dapat ditambahkan dalam multimedia melalui suara, musik dan efek-efek suara.

#### **II.5.2.4. Video**

Video menyediakan sumber daya yang kaya dan hidup bagi aplikasi multimedia.

#### **II.5.2.5. Animasi**

Dalam multimedia, animasi merupakan penggunaan komputer untuk menciptakan gerak pada layar.

#### **II.5.2.6. *Virtual Reality***

Virtual reality merupakan penggunaan multimedia untuk penerapan secara langsung.

### **II.6. *3D Max***

*3ds Max* adalah sebuah software yang dikhususkan dalam pemodelan 3 dimensi ataupun untuk pembuatan animasi 3 dimensi. Selain terbukti handal untuk digunakan dalam pembuatan objek 3 dimensi, *3ds Max* juga banyak digunakan dalam pembuatan desain furnitur, konstruksi, maupun desain interior. Selain itu, *3ds Max* juga sering digunakan dalam pembuatan animasi atau film kartun.

*3ds Max* yang dilengkapi dengan bahasa *scripting* (*MaxScripting*) terbukti ampuh untuk membuat game 3 dimensi, mulai dari yang sederhana hingga yang rumit sekalipun. Dengan kemampuan tersebut, banyak orang maupun instansi memanfaatkan software *3ds Max* untuk membuat suatu desain atau iklan yang berguna sebagai media publikasi produk atau karya mereka kepada publik. *3ds Max* memungkinkan pengguna untuk membuat tampilan 3 dimensi yang sangat menarik. (Galih Purnowo ; 2010 : 1)

#### **II.6.1. Area Kerja *3Ds Max***

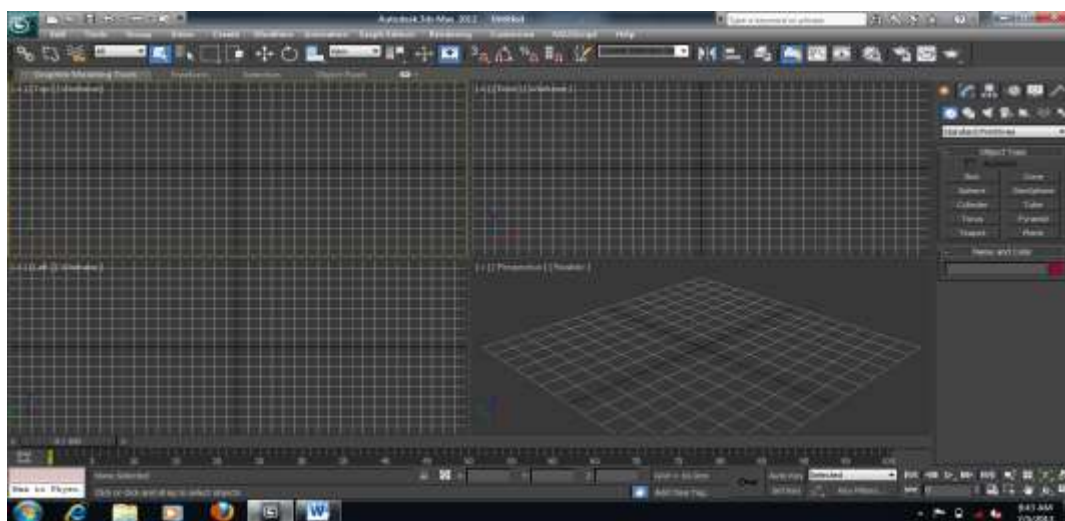
Saat pertama kali mengerjakan program 3D Studio Max12 (3Ds Max12), maka kita akan mendapati tampilan halaman pembuatan 3Ds Max12 seperti yang dilihat pada Gambar II.1. berikut:



**Gambar II.4. Tampilan Halaman Pembuka 3Ds Max**

*Sumber : (Galih Pramowo: 2010; 2)*

Setelah proses *loading* program 3Ds Max 2010 selesai, maka akan tampil bagian antarmuka dari 3Ds Max 2010. Area kerja 3Ds Max 2010 dapat dilihat pada Gambar II.2.

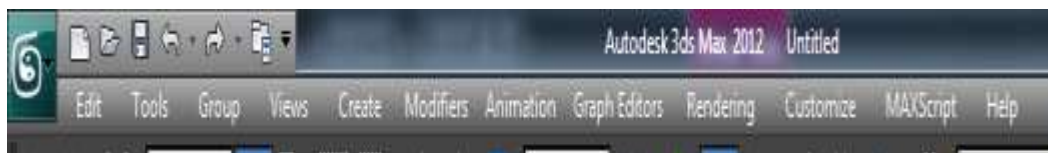


## Gambar II.5. Tampilan Area Kerja 3Ds Max

*Sumber : (Galih Pramowo: 2010; 2)*

### II.6.2. Menu Bar

*Menu Bar* pada 3Ds Max adalah sebuah menu bar standart Windows yang memuat menu *File, Edit, Tools, Group, Views, Create, Modifiers, Reactor, Animation, Graph Editors, Rendering, Customize, MAXScript, Help*. Berikut ini adalah merupakan gambar dari *Menu Bar*.



### Gambar II.6. Tampilan Menu Bar

*Sumber : (Galih Pramowo: 2010; 2)*

1. **Tools** yaitu memuat berbagai perintah pengeditan yang sebagian besar juga ditampilkan pada Main Toolbar.
2. **Group** yaitu memuat perintah-perintah yang berkaitan dengan pengelompokan objek agar dapat jadi satu bagian dari beberapa bagian objek.
3. **Views** yaitu memuat perintah-perintah untuk mengeset atau mengatur dan mengontrol viewport.
4. **Create** yaitu perintah-perintah untuk membuat satu objek.
5. **Modifiers** yaitu perintah-perintah untuk memodifikasi suatu objek
6. **Animation** yaitu perintah-perintah untuk memuat dan mengendalikan suatu animasi.
7. **Graph Editors** yaitu perintah-perintah untuk memberikan akses grafis untuk mengedit objek dan animasi.

8. **Rendering** yaitu perintah-perintah untuk melakukan rendering, mengatur Video Post, radiosity, dan environment.

9. **MAXScript** yaitu perintah-perintah untuk menggunakan MAXScript atau bahasa scripting dalam 3D Studio Max, (Galih Pramowo: 2010; 2).

### **II.6.3. Main Toolbar**

*Main Toolbar* terletak tepat di bawah Menu Bar. *Main Toolbar* menyediakan *shortcut* instruksi-instruksi praktis mulai dari penyelesaian objek, Material Editor, hingga Rendering. Adapun tombol dari *Main Toolbar* adalah sebagai berikut : *Select and Link*, *Unlink Selection*, *Bind to Space Warp*, *Selection Filter List*, *Select Object*, *Select by Name*, *Selection Region Flyout*, *window/Crossing*, *Select and Move*, *Select and rotate*, *Select and Scale*, *Snap Toggle*, *Percent Snap Toggle*, *Mirror*, *Layar Maneger*, *Material Editor*, *Reder Setup*, *Rendered Frame Window*, *Render Production*, *Render Iterative*, *ActiveShade*. Berikut ini merupakan gambar dari *Main Toolbar*, (Galih Pramowo: 2010; 4-7).



**Gambar II.7. Tampilan *Main Toolbar***

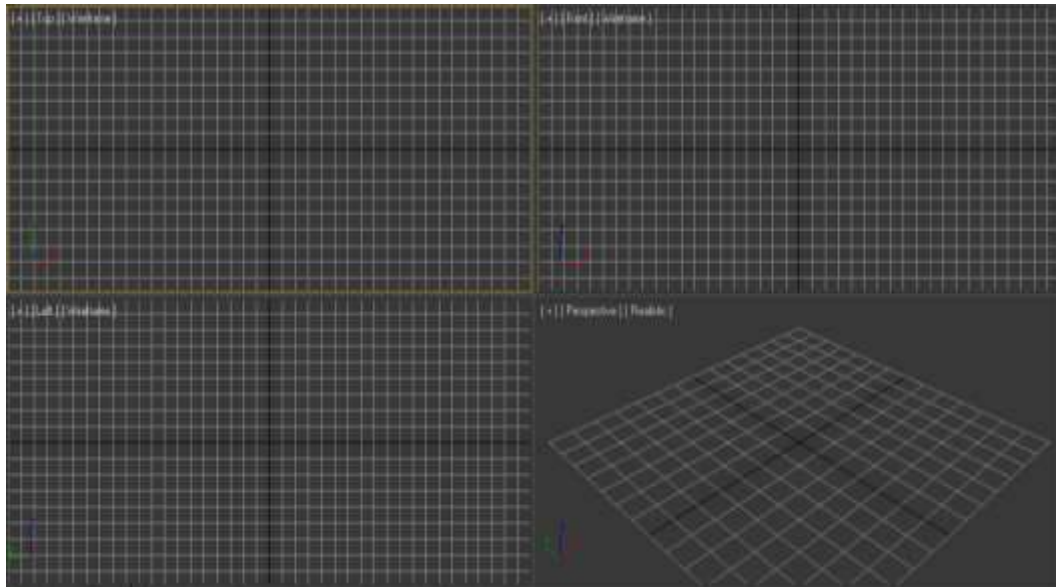
***Sumber : (Galih Pramowo : 2010; 4)***

### **II.6.4. Viewport**

*Viewport* dalam *3Ds Max* adalah ruang kerja atau layar kerja dimana kitadapat melakukan pekerjaan membuat animasi. *Viewport* juga akan menjadi tempat yang paling sering digunakan, baik dalam pemodelan maupun animasi. Secara *dafault*, *Viewport* terbagi menjadi empat bagian,

yaitu *Top viewport* (kiri atas), *Front Viewport* (kiri bawah), *Left Viewport* (kanan atas), dan *Perspective Viewport* (kanan bawah). Untuk mengubah *Viewport*, klik kanan pada label *Viewport* yang bersangkutan dan pilih *Viewport* yang digunakan, (Galih Pramowo: 2010; 10-11).

Berikut ini merupakan gambar dari *Viewport*.



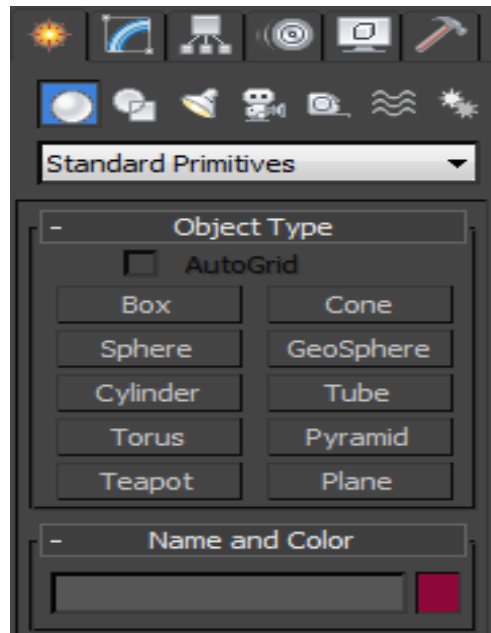
**Gambar II.8. Tampilan *Viewport***

***Sumber : (Galih Pramowo: 2010; 12)***

### **II.6.5. *Command Panel***

*Command Panel* adalah bagian yang akan sering digunakan selain *viewport*. *Command Panel* terletak di sebelah kanan *Viewport* dan merupakan 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 sebuah objek dan menambahkan modifier), *Hierarchy* (parameter-parameter untuk melakukan link dan parameter Inverse Kinematics), *Motion* (sebagai pengatur animasi), *Display* (control tampilan), dan ***Utilities***, (Galih

Pramowo: 2010; 8).Berikut ini merupakan Gambar dari Command File.



**Gambar II.9. Tampilan *Command Panel***

***Sumber : (Galih Pramowo; 2010: 8)***

### **II.6.6. *Timeline Animation***

*Timeline Animation* merupakan fasilitas yang disediakan 3Ds Max untuk melakukan animasi atau sebagai pencatat aktifitas objek kapan harus tampil dan kapan harus menghilang. Selain itu, *Timeline Animation* juga berguna untuk melakukan pengeditan animasi dengan tombol-tombol yang sesuai dengan fungsinya masing-masing. Pada bagian ini juga di sediakan pasilitas untuk mengontrol animasi, dan sebagainya, (Galih Pramowo: 2010; 13). Berikut ini merupakan gambar dari *Timeline Animation*.

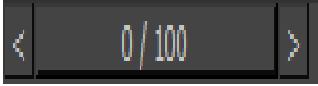
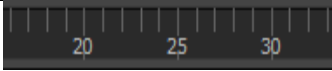

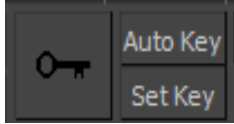


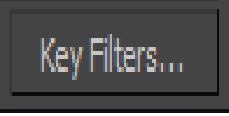
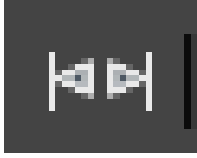

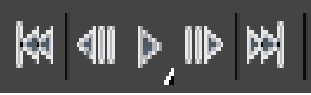
**Gambar II.10. Tampilan *Timeline Animation***

**Sumber : (Galih Pramowo; 2010: 13).**

*Timeline Animation* sering digunakan dalam membuat pergerakan suatu animasi dan untuk mengatur waktu animasi. Bagi para *Animator* atau pembuat animasi, *Timeline* adalah hal penting yang harus diperhatikan agar nantinya dapat menghasilkan sebuah animasi yang sempurna. Tabel berikut ini adalah bagian-bagian dari *Timeline Animation 3Ds Max* yang sering digunakan ketika membuat animasi.

**Tabel II.1. Ikon-Ikon *Timeline Animation***

Ikon	Nama Fasilitas	Keterangan
	<i>Slinder Timeline</i>	Merupakan fasilitas untuk memindah <i>frame</i> dari <i>timeline</i> atau untuk membentuk objek berada pada urutan <i>frame</i> tertentu.
	<i>Frame</i>	<i>Frame</i> adalah sebuah kolom yang berada pada <i>timeline</i> yang berfungsi untuk membuat suatu pergerakan objek dari suatu titik ke titik yang lainnya.
	<i>Open Mini Curve Editor</i>	Tombol ini berguna untuk membuka kotak editor pengontrol animasi dan suara
	<i>Auto Key, Set Key</i>	Tombol ini berfungsi untuk mengaktifkan dan mengunci objek dalam <i>frame</i> yang telah ditentukan pada <i>timeline</i> .

	<i>Key Filters</i>	Berfungsi untuk membuka kotak dialog pilihan <i>Set Key Filter</i> untuk menentukan posisi, rotasi, dan skala pada animasi.
	<i>Set Mode Toggle</i>	Berfungsi untuk mengaktifkan <i>mode Toggle</i> dalam animasi.
	<i>Timeline Configuration</i>	Berfungsi untuk membuka kotak dialog <i>timeline Configuration</i> untuk mengatur panjang pendek <i>frame</i> , tampilan waktu, <i>frame rate</i> , serta kecepatan dalam animasi.
	<i>Go to start, Previous Frame, Play Animation, Next Frame, Go to End</i>	Berfungsi untuk mengatur jalannya animasi, kembali ke awal animasi, kembali ke frame sebelumnya, memainkan animasi, menuju ke frame berikutnya, dan menuju ke akhir animasi.

**Sumber : (Galih Pramowo: 2010; 13-14)**

## **II.7. Adobe Flash CS6**

*Adobe Flash CS6* merupakan sebuah *software* yang didesain khusus oleh Adobe dan program aplikasi standar *authoring tool professional* yang telah digunakan untuk membuat animasi dan *bitmap* yang sangat menarik untuk keperluan pembangunan situs web yang interaktif dan dinamis. *Adobe Flash CS6* menyediakan berbagai macam fitur yang akan sangat membantu para animator untuk membuat animasi menjadi semakin mudah dan menarik. *Adobe Flash CS6* telah mampu membuat dan mengolah teks maupun objek dengan efek tiga dimensi, sehingga hasilnya tampak lebih menarik.

*Flash* didesain dengan kemampuan untuk membuat animasi 2 dimensi atau 3 dimensi

yang handal dan ringan sehingga Flash banyak digunakan untuk membangun dan memberikan afek animasi pada *website*, CD Interaktif dan yangn lainnya, selain itu *software* ini juga dapat digunakan untuk membuat animasi logo, movie, game, pembuatan navigasi pada situs *website* atau *blog*, tombol animasi, *banner*, menu interaktif, interaktif *form* isian, e-card, *screen sever* dan pembuatan aplikasi-aplikasi *website* lainnya.

### II.7.1. Area Kerja *Macromedia Flash Player*

Langkah untuk menjalankan program Adobe Flash Pro CS6, tekan tombol **Start ► All Programs ► Adobe ► Adobe Flash CS6** sehingga tampil **Welcome Screen** seperti tanpak pada gambar berikut, (M. Leo Agung; 2013: 4).



Gambar II.11. Tampilan Halaman *Macromedia Flash 8*

(Sumber : M. Leo Agung, 2013:4)

## II. 8. UML (*Unified Modelling Language*)

*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 mempelopori

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.(Yuni Sugiarti ; 2013 : 33)


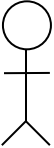

Dalam pembuatan skripsi ini penulis menggunakan diagram Use Case yang terdapat di dalam UML. Adapun maksud dari Use Case Diagram diterangkan dibawah ini.

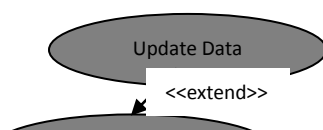
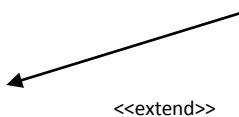
### ***1. Use Case Diagram***

*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. (Yuni Sugiarti ; 2013 : 41)

**Tabel II.2. Simbol-simbol Use Case Diagram**

Simbol	Deskripsi
Usecase 	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case.
Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan dengan menggunakan kata benda diawal frase nama aktor.
Assosiasi / Association 	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.



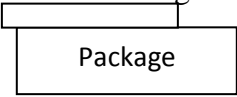
Exetnd	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan, arah panah menunjuk pada use case yang dituju. Contoh :
Include	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsi atau sebagai syarat dijalankan use case ini. Ada dua sudut pandang yang cukup besar mengenai include di use case, include berarti use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan, contoh :

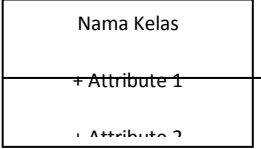
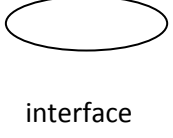
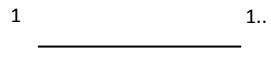

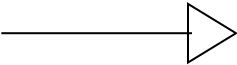
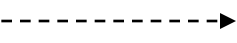
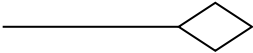
*Sumber : (Yuni Sugiarti ; 2013 ; 42)*

## 2. Class Diagram

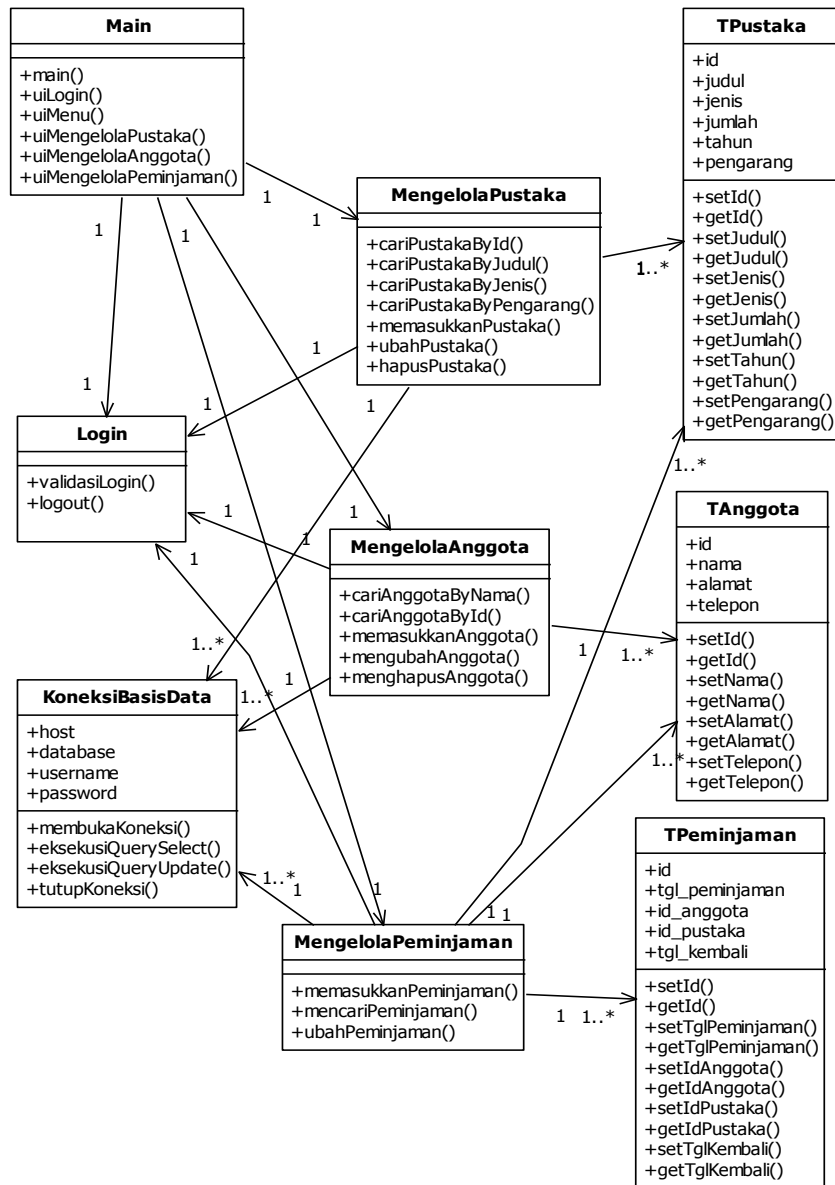
Diagram kelas atau *class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Berikut adalah simbol-simbol pada diagram kelas :

**Tabel II.3. Simbol-simbol *Class* Diagram**

Simbol	Deskripsi
	Package merupakan sebuah bungkusan dari satu atau lebih kelas

<p style="text-align: center;"><b>Onerasi</b></p> 	<p>Kelas pada struktur sistem</p>
<p style="text-align: center;"><b>Antarmuka/Interface</b></p> 	<p>Sama dengan konsep interface dalam pemrograman berorientasi objek</p>
<p style="text-align: center;"><b>Asosiasi</b></p> 	<p>Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.</p>
<p style="text-align: center;"><b>Asosiasi</b></p> <p style="text-align: center;"><b>Berarah/directed</b></p> <p style="text-align: center;">assosiasi</p> 	<p>Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.</p>
<p style="text-align: center;"><b>Generasi</b></p> 	<p>Relasi antar kelas dengan makna generalisasi-spesialisasi (umum – khusus)</p>
<p style="text-align: center;"><b>Kebergantungan/Defedency</b></p> 	<p>Relasi antar kelas dengan makna kebergantungan antar kelas</p>
<p style="text-align: center;"><b>Agregasi</b></p> 	<p>Relasi antar kelas dengan makna semua bagian (whole-part)</p>

**Sumber : (Yuni Sugiarti ; 2013:59)**



Gambar II.14. Contoh Class Diagram

Sumber : (Yuni Sugiarti ; 2013 : 63)

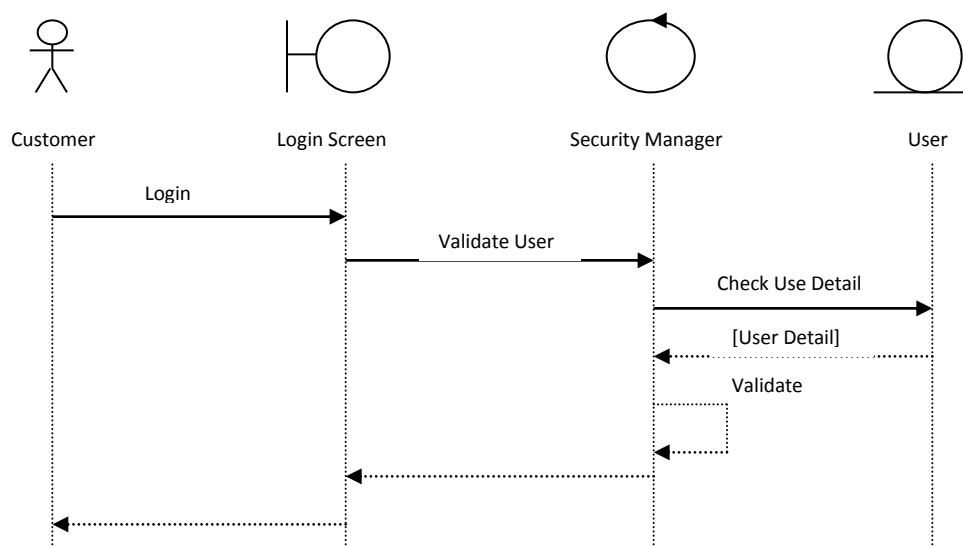
### 3. Sequence Diagram

Diagram *Sequence* menggambarkan 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 *sequence* maka harus diketahui objek-objek

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

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



**Gambar II.15. Contoh *Sequence Diagram***

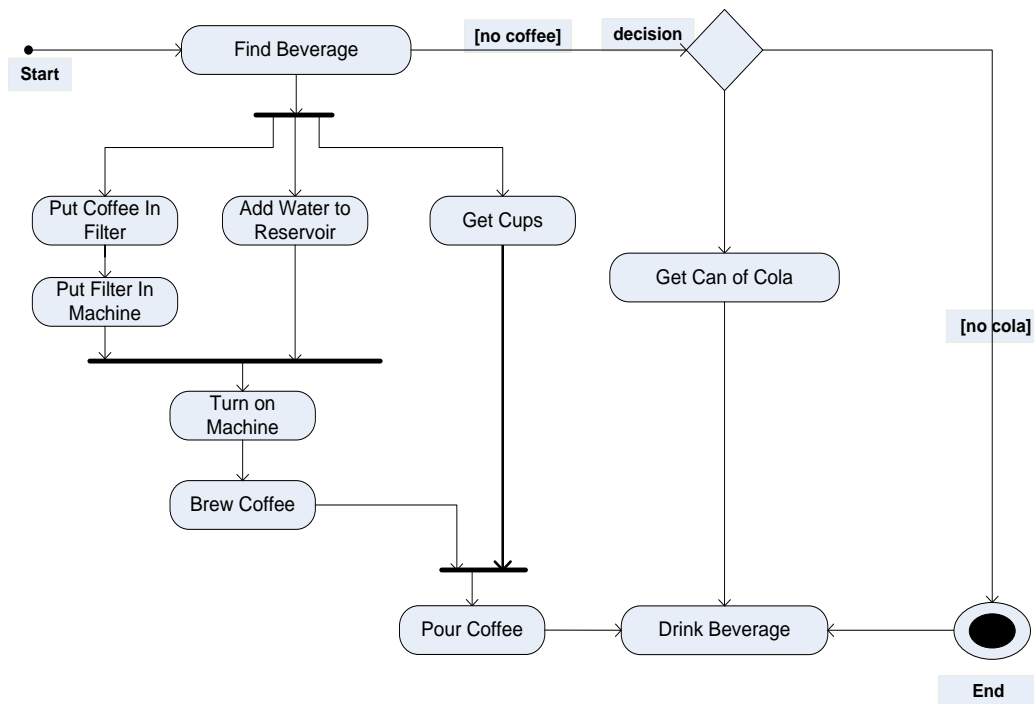
*Sumber : (Yuni Sugiarti ; 2013 : 63)*

#### **4. 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.



**Gambar II.16. Activity Diagram**

*Sumber : (Yuni Sugiarti ; 2013 : 76)*