

BAB II

TINJAUAN PUSTAKA

II.1. Perancangan

Perancangan merupakan spesifikasi umum dan terinci dari pemecahan masalah berbasis komputer yang telah dipilih selama tahap analisis. Perancangan adalah menyeleksi dan menghubungkan pengetahuan, fakta, imajinasi, asumsi untuk masa yang akan datang dengan tujuan memvisualisasi dan memformulasi hasil yang diinginkan, dan juga urutan kegiatan yang diperlukan. Perancangan dalam pengertian ini menitik beratkan kepada usaha untuk menyeleksi dan menghubungkan sesuatu dengan kepentingan masa yang akan datang serta usaha untuk mencapainya.

II.2. Animasi

Animasi berasal dari bahasa latin yaitu “anima” yang berarti jiwa, hidup, nyawa, semangat. Sedangkan animasi secara utuh diartikan sebagai gambar yang memuat objek yang seolah-olah hidup. Disebabkan oleh kumpulan gambar itu berubah beraturan dan bergantian ditampilkan, objek dalam gambar bisa berupa tulisan, bentuk benda, warna dan efek spesial (Jevri Setia Nugraha ; 2012 : 64).

II.2.1. Sejarah Animasi

Animasi berawal dari kontribusi hasil karya seorang yang bernama Walt Disney yang lahir dengan nama Walter alias Disney, Walt lahir di Chicago, Illinois dengan orang tua bernama Elias Disney dan Flora Call. Pada 1906, mereka semua pindah ke sebuah peternakan dekat Marceline, Missouri, Amerika Serikat. Walt sejak kecil sudah memiliki hobi menggambar, khususnya gambar kartun, mulai dari sketsa hingga kartun yang berwarna.

II.2.2. Jenis-Jenis Animasi

Menurut Patmore (2003 : 2-3) ada beberapa jenis animasi, diantaranya adalah:

1. *Stop Motion*

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*

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. Contohnya :

- a) Gerakan bunga yang terlihat ketika mekar.
- b) Pergerakan matahari yang terlihat dari terbit sampai tenggelamnya.

4. *Claymation*

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*.

II.3. Unified Modeling Language (UML)

Mengingat perkembangan UML yang banyak digunakan untuk perancangan sistem khususnya dalam disain piranti lunak. Perlu diingat kembali bahwa UML hanyalah merupakan salah satu cara dalam menjelaskan perancangan sistem.

Unified Modeling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem

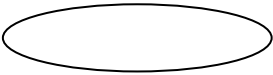
piranti lunak. UML mendefinisikan notasi dan syntax/semantik. (Haviluddin ; 2011 : 11).

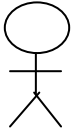

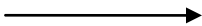
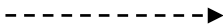

II.3.1. Use Case

Use Case adalah deskripsi fungsi dari sebuah sistem dari *perspektif* pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Model use case adalah bagian dari *requirement*. *Use case* adalah alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang suatu sistem dari sudut pandangnya. Dengan demikian diharapkan akan bisa dibangun suatu sistem yang bisa membantu pengguna, perlu diingat bahwa *use case* mewakili pandangan diluar sistem. Diagram *Use case* menunjukkan 3 aspek dari sistem yaitu: actor, *use case* dan *system / sub system boundary*. Actor mewakili peran orang, *system* yang lain atau ketika berkomunikasi dengan use case. Berikut gambar notasi use case:

Berikut ini gambar table simbol-simbol dari *Use Case* Diagram adalah:

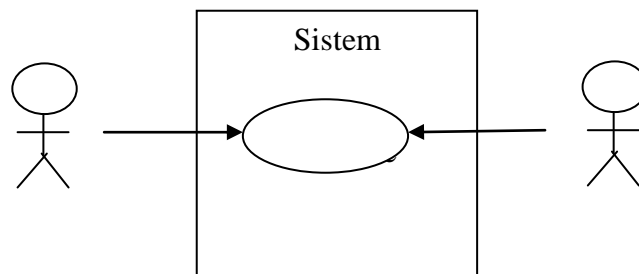
Tabel II.1. Simbol-simbol dari Use Case Diagram

Gambar	Keterangan
	Use case menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja awal nama use case

	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berintraksi dengan use case, tetapi tidak memiliki control terhadap use case</p>
	<p>Asosiasi antara aktor dan use case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta intraksi secara langsung dan bukannya mengidikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan use case yang menggunakan panah terbuka untuk mengidikasikan bila aktor berintraksi secara pasif dengan sistem.</p>
	<p><i>Include</i>, merupakan di dalam use case lain (required) atau pemanggilan use case oleh use case lain, contohnya adalah pemanggilan sebuah fungsi program.</p>
	<p><i>Extend</i>, merupakan perluasan dari use case lain jika kondisi atau syarat terpenuhi.</p>

(Sumber : Windu Gata ;2013 :4)

Berikut contoh dari *Use case* Diagram:



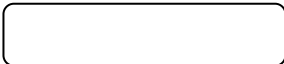
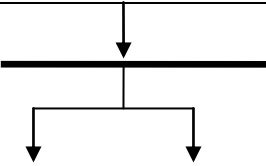
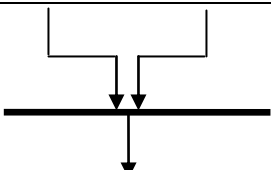
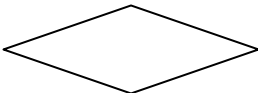


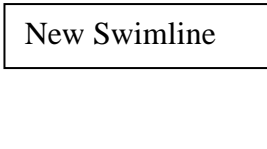
Gambar II.1. Contoh *Use case* Diagram
(Sumber : Adi Nugroho ; 2010:86)

II.3.2. Activity Diagram

Activity Diagram teknik untuk mendeskripsikan logika procedural, prose bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya *flow chart*, akan tetapi perbedaannya dengan *activity diagram* adalah *activity diagram* bias mendukung perilaku parallel sedangkan *flowchart* tidak bisa. Berikut adalah symsbol yang ada pada *activity diagram*.

Tabel II.2. Simbol Pada Activity Diagram

Gambar	Keterangan
	Start point, dilektakkan pada pojok kiri atas dan merupakan awal aktifitas
	End point, akhir aktifitas
	Activites, menggambarkan suatu proses/kegiatan bisnis.
	Fork(Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu.
	Join(Penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan,true,false.

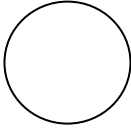
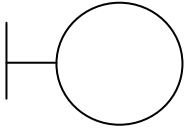
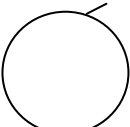
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.
---	--


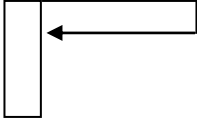


(Sumber : Windu Gata ; 2013:6)

II.3.3. Sequence Diagram

Sequence diagram menggambarkan intraksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. Sequence diagram biasa digunakan untuk menggambarkan scenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu:

Tabel II.3. Simbol Pada Sequence Diagram

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> antara satu atau lebih actor dengan sistem, seperti tampilan <i>formentry</i> dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.

	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari panjang, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis-garis titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terhadap <i>activation</i>

(Sumber : Windu Gata ; 2013:7)

II.3.4. Class Diagram

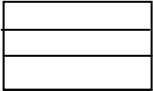

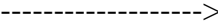
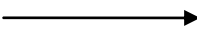
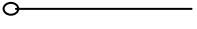
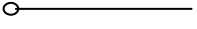
Class adalah hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

Class diagram secara khas meliputi: Kelas (*Class*), *Relasi*, *Associations*, *Generalization* dan *Aggregation*, Atribut(*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut.

Adapun simbol-simbol yang terdapat pada Sequence diagram:

Tabel II.4. Simbol Pada *Sequence Diagram*

Gambar	Keterangan
	Class,menambahkan kelas baru pada diagram
	Interface,menambahkan kelas antarmuka pada diagram
	<i>Association</i> ,menggambarkan relasi asosiasi
	<i>Generalization</i> ,menggambarkan relasi generalisasi
	<i>Realize</i> , menggambarkan relasi
	<i>Aggregation</i> ,menggambarkan relasi agregasi

(Sumber : Windu Gata ; 2013:9)

Hubungan antar kelas mempunyai keterangan yang disebut dengan multiplicity atau kardinaliti.

Tabel II.5. Kadinality Pada *Sequence Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada,maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 4

(Sumber : Windu Gata ; 2013:9)

II.4. Bangunan Gereja

Pandangan gereja adalah sebuah gedung atau bangunan merupakan pandangan yang salah yang paling banyak dimengerti oleh setiap orang tentang arti gereja. Gereja adalah rumah tetapi bukanlah bangunan gereja. Gereja mula-mula saat itu merupakan sekumpulan orang percaya yang bersekutu untuk beribadah kepada Tuhan. Dengan adanya perkembangan gereja yang semakin luas pada setiap jamannya, maka kemudian gereja dibagi kedalam wilayah-wilayah dan tempat yang tetap untuk beribadah.

Dilihat dari asal usulnya Gereja dalam bahasa Indonesia merupakan kata serapan dari Bahasa Latin yang diserap pula dari Bahasa Yunani "*ekklesia*" yang berarti dipanggil keluar (*ek* = Portugis "*igreja*"). Dalam Bahasa Portugis merupakan serapan dari Bahasa Latin yang diserap pula dari Bahasa Yunani "*ekklesia*" yang berarti dipa(nggil keluar (*ek* = keluar; *klesia* dari kata *kaleo* = memanggil). Jadi *ekklesia* berarti kumpulan orang yang dipanggil ke luar (dari dunia ini) untuk dapat memuliakan nama Allah (Surya; 2011 : 13).

II.5. Gereja HKBP

Gereja Huria Kristen Batak Protestan (HKBP) merupakan satu dari berbagai macam gereja Kristen Protestan yang ada di Indonesia. HKBP merupakan gereja Kristen yang berkembang dengan kebudayaan Batak. Gereja HKBP ini memiliki penyebaran yang cukup jelas di Indonesia, ini seiring dengan penyebaran orang-orang

yang berasal dari suku Batak di seluruh Indonesia. HKBP memiliki anggota mayoritas suku Batak. (Ricky ; 2013 : 2).

Demi mencapai prestasi belajar yang memuaskan tersebut dengan sistem pendidikan perkuliahan yang semakin maju dan didukung juga perkembangan teknologi. Teknologi multimedia telah menjanjikan potensi besar dalam merubah cara seseorang untuk belajar, untuk memperoleh informasi, menyesuaikan informasi dan sebagainya. (Yustecia ; 2013 : 15).

II.6. 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. (Galih Prabowo ; 2012 : 6)

3ds Max atau 3D Studio Max merupakan sebuah aplikasi standar desain grafis dan animasi berbasis Windows yang paling populer dan inovatif saat ini. Aplikasi ini digunakan untuk membuat film animasi 3 dimensi maupun *finishing object*, baik untuk keperluan film, iklan, maupun arsitektur. (Mickael Sugianto ; 2011 : 4)

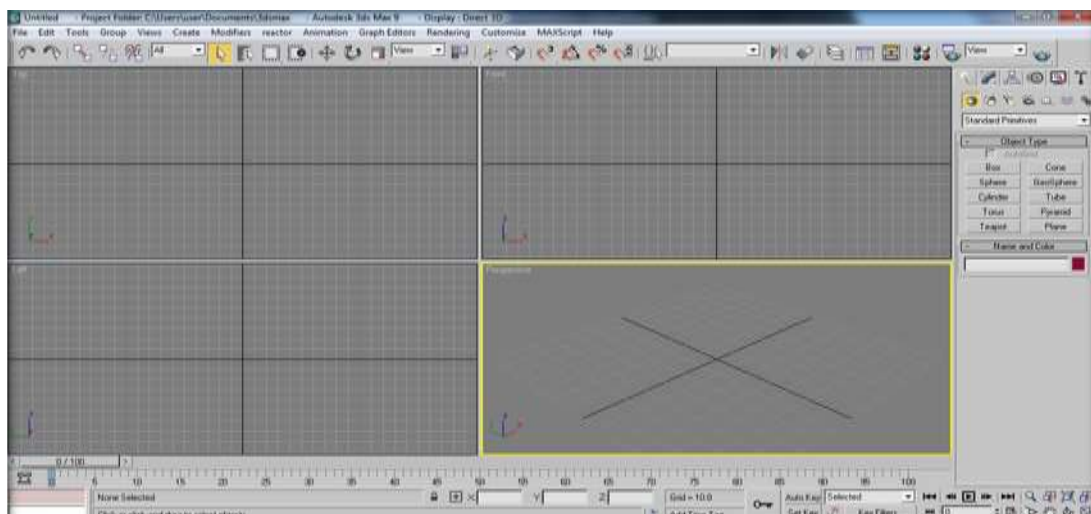
II.6.1. Area Kerja 3Ds Max 9

Saat pertama kali menjalankan program 3D Studio Max 9 (3Ds Max 9), maka kita akan mendapati tampilan halaman pembuka 3Ds Max 9 seperti yang terlihat pada Gambar II.1 di bawah ini :



Gambar II.2. Tampilan Halaman Pembuka 3Ds Max
Sumber : (Galih Pramowo: 2010; 2).

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

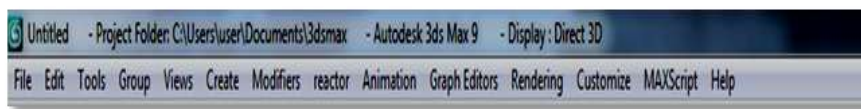


Gambar II.3. Tampilan Area Kerja 3Ds Max

Sumber : (Galih Pramowo: 2010; 2).

II.6.2. Menu Bar

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



Gambar II.4. Tampilan Menu Bar

Sumber : (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-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, Angle Snap Toggle, Percent Snap Toggle, Mirror, Layer Manager, Material Editor, Render Setup*. Berikut ini merupakan gambar dari *Main Toolbar*.

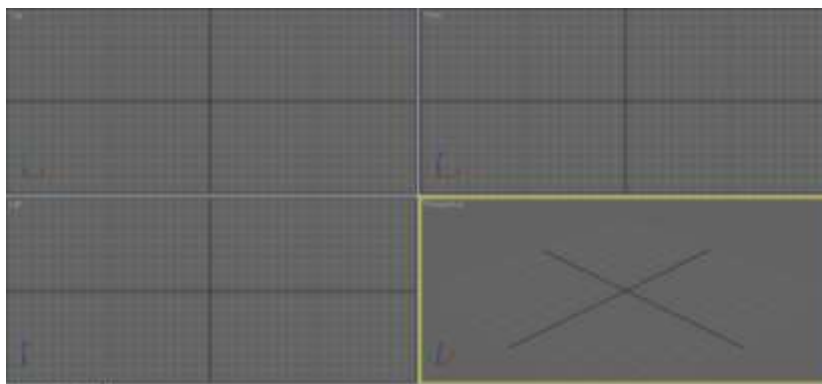


Gambar II.5. Tampilan Main Toolbar

Sumber : (Galih Pramowo : 2010; 4).

II.6.4. Viewport

Viewport dalam 3Ds Max adalah ruang kerja atau layer kerja dimana kita dapat melakukan pekerjaan membuat animasi. *Viewport* juga akan menjadi tempat yang paling sering digunakan, baik dalam pemodelan maupun animasi. Secara *default*, *Viewport* terbagi menjadi empat bagian, yaitu *Top Viewport* (kiri atas), *Front Viewport* (kanan atas), *Left Viewport* (kiri bawah), dan *Perspective Viewport* (kanan bawah). Untuk mengubah *Viewport*, klik kanan pada label *Viewport* yang bersangkutan dan pilih *Viewport* yang diinginkan. Berikut ini merupakan gambar dari *Viewport*.

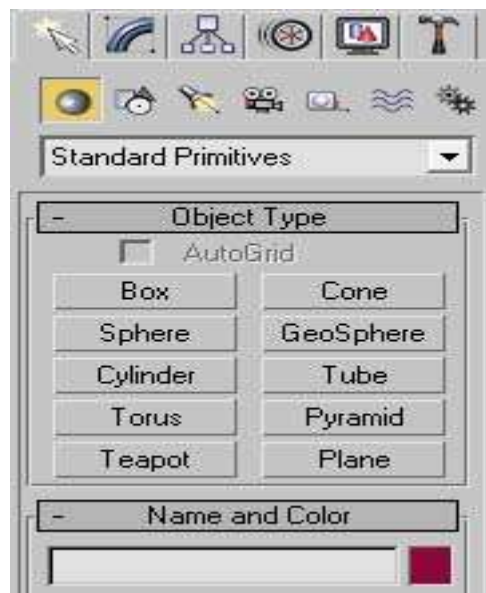


Gambar II.6. Tampilan Viewport

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-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.



Gambar II.7. Tampilan Command Panel

Sumber : (Galih Pramowo; 2010: 8).

II.6.6. Timeline Animation

Timeline Animation merupakan fasilitas yang disediakan 3Ds Max untuk melakukan proses animasi atau sebagai pencatat aktivitas objek kapan harus tampil dan kapan harus menghilang. Selain itu, *Timeline Animation* juga berguna untuk melakukan pengeditan animasi dengan tombol-tombol yang sesuai dengan fungsi masing-masing. Pada bagian ini juga disediakan fasilitas untuk mengontrol animasi, memulai animasi, menghentikan animasi, dan sebagainya. Berikut ini merupakan gambar dari *Timeline Animation*.











Gambar II.8. 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 suatu animasi.

Tabel II.6. Ikon-Ikon Timeline Animation

Ikon	Nama Fasilitas	Keterangan
	<i>Slider Timeline</i>	Merupakan fasilitas untuk memindah <i>frame</i> dari <i>timeline</i> atau untuk menentukan 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 satu titik ke titik yang lain.
	<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 pada <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</i>	Berfungsi untuk mengatur jalannya animasi, kembali ke awal animasi, kembali ke <i>frame</i> sebelumnya, memainkan animasi,

	<i>End</i>	menuju ke frame berikutnya, dan menuju ke akhir animasi.
--	------------	--

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