

BAB II

LANDASAN TEORI

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.

Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Perancangan sistem dapat dirancang dalam *flowchart*, yang merupakan alat bentuk grafik yang dapat digunakan untuk menunjukkan urutan-urutan proses. (Yeni Agusti; 2013:48).

II.1.1. Multimedia

Multi-banyak, Media sarana berkomunikasi untuk melewati informasi. Suatu sistem yang terdiri dari perangkat keras, perangkat lunak dan 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* dan *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 interaktif yang mengkombinasi teks grafik, animasi, audio dan gambar video.

Multimedia adalah kombinasi dari komputer dan video atau secara umum merupakan kombinasi tiga elemen yaitu suara, gambar, dan teks atau kombinasi dari yang sedikit dua media *input* atau *output* dari data yang berupa audio (suara dan musik). Istilah multimedia berawal dari teater, bukan komputer. Pertunjukan yang memanfaatkan lebih dari satu medium seringkali disebut pertunjukan multimedia. (Chrisna Atmadji; 2010:59).

II.1.2. Siklus Hidup Pengembangan Sistem

Metode perangkat lunak yang digunakan dalam penelitian adalah dengan pendekatan siklus hidup pengembangan system yang selanjutnya disebut SHPS. SHPS adalah pendekatan melalui beberapa tahap untuk menganalisis dan merancang sistem yang dimana sistem tersebut telah dikembangkan dengan baik melalui penggunaan siklus penganalisisan dan pemakai secara spesifik.’

Tahap-tahap SHPS adalah sebagai berikut:

1. Mengidentifikasi masalah, peluang dan tujuan
2. Menentukan Syarat-syarat
3. Menganalisis kebutuhan sistem
4. Merancang sistem
5. Mengembangkan dan mendokumentasikan sistem
6. Menguji dan mempertahankan sistem
7. Mengimplemetasikan dan mengepaluasi

Tahap ini adalah tahap perancangan suatu sistem untuk nantinya diimplementasikan menjadi sebuah perangkat lunak. Pada pengajaran Kinematika Gerak Lurus, Gerak Melingkar dan Gerak Parabola berbasis komputer menggunakan *Data Flow Diagram (DFD)* sebagai pemodelan sistem yang telah dijabarkan sebelumnya. Pada tahap ini akan dilalui tahap merancang sistem yang direkomendasikan sebelumnya sesuai dengan siklus hidup pengembang sistem (SHPS). Dalam tahap desain dari (SHPS). Penulis menggunakan informasi yang telah terkumpul sebelumnya untuk mencapai desain sistem informasi yang logik. Penulis merancang prosedur data-entry sedemikian rupa sehingga data yang dimasukkan kedalam sistem benar benar akurat selain itu penulis menggunakan teknik-teknik bentuk perancangan layar tertentu untuk hasil yang lebih baik. (Novri Hadinata; 2010:20).

II.2. Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. (Memahami Penggunaan UML (*Unified Modelling Language*) ; Mulawarman ; 2011:1).

Berikut tujuan utama dalam desain UML adalah :

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.

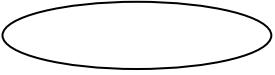
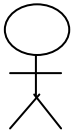
Memiliki integrasi praktik terbaik. (Memahami Penggunaan UML (*Unified Modelling Language*) ; Mulawarman ; 2011 : 2).

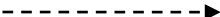
II.2.1. Use Case

Use Case adalah deskripsi fungsi dari sebuah sistem dari *perspektif* pengguna. *Use case* bekerja dengan cara mendeeskripsikan tipikal intraksi 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 systemboundary*. *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:

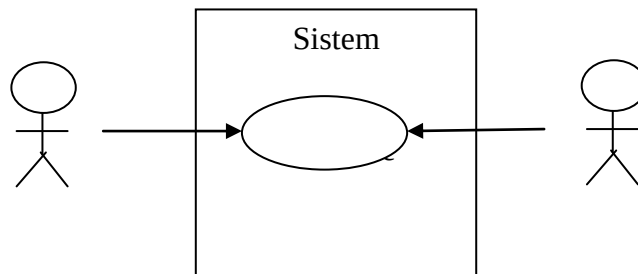
Tabel II.1. Simbol-simbol dari Use Case Diagram

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja awal nama <i>use case</i>
	<i>Actor</i> adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi <i>actor</i> , 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 <i>actor</i> berintraksi dengan <i>use case</i> , tetapi tidak memiliki <i>control</i> terhadap <i>use case</i>

	Asosiasi antara <i>actor</i> dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta intraksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara <i>aktor</i> dan <i>use case</i> yang menggunakan panah terbuka untuk mengidikasikan bila <i>actor</i> berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (required) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata ;2013 :4)

Berikut contoh dari *Use case Diagram*:

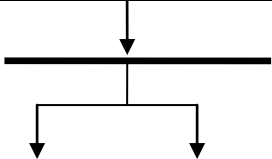
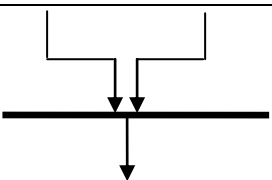
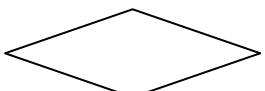
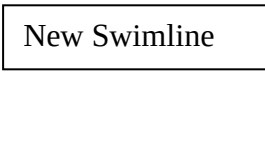


Gambar II.1. Contoh *Use case Diagram*
(Sumber : Adi Nugroho ;2010:86)

II.2.2. *Activity Diagram*

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

Tabel II.2. Simbol Pada Activity Diagram

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

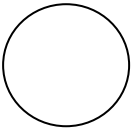
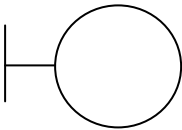
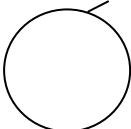
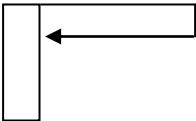
(Sumber : Windu Gata ;2013 :6)

II.2.3. 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>form entry</i> dan <i>form cetak</i> .
	<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.2.4. Class Diagram

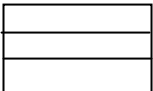
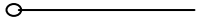
Class adalah hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

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
	<i>Class</i> , menambahkan kelas baru pada diagram
	<i>Interface</i> , 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

<i>Multiplicity</i>	<i>Penjelasan</i>
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.3. Simulasi

Model simulasi pada dasarnya salah satu strategi pembelajaran yang bertujuan memberikan pengamalan belajar yang lebih konkrit melalui penciptaan tiruan-tiruan bentuk pengalaman yang mendekati suasana yang tanpa resiko. Dalam model ini menampilkan materi pelajaran dikemas dalam bentuk animasi, gambar, dan paduan warna yang serasi dan harmonis. (Yeni Agusti; 2013:48-49).

Simulasi digunakan untuk memperagakan sesuatu sehingga siswa merasa seperti berada dalam keadaan yang sebenarnya. Simulasi banyak digunakan pada pembelajaran materi yang membahayakan, ataupun materi yang sulit dimengerti. Hanya ada 1 model simulasi pada pembelajaran sains fisika ini, yaitu gabungan dari beberapa topik pembahasan. Simulasi ini juga disajikan setelah materi Gerak Lurus Beraturan dikarenakan pada simulasi ini telah

mencakup beberapa topik pembahasan pada penggunaan rumusnya. Simulasi yang ditawarkan dalam aplikasi pembelajaran sains fisika ini berupa simulasi sebuah mobil bergerak dengan tambahan suasana multimedia dan suara didalamnya agar pembelajaran sains fisika lebih menarik dan dapat dimengerti melalui model simulasi ini. (Junita Verayanthi Manurung; 2014:121).

II.4. Animasi

Penyimpanan data terhadap perubahan tempat atau bentuk pada setiap gerakan obyek, dan terdapat beberapa teknik untuk menggerakkan suatu obyek yang setiap pergerakannya disimpan dalam *frame* dengan jangka waktu yang telah ditentukan. (Mikael Sugianto 2011:151).

II.4.1. 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.4.2. Pengertian Animasi

Penyimpanan data terhadap perubahan tempat atau bentuk pada setiap gerakan obyek, dan terdapat beberapa teknik untuk menggerakkan suatu obyek

yang setiap pergerakannya disimpan dalam *frame* dengan jangka waktu yang telah ditentukan. (Mikael Sugianto 2011:151).

II.4.3. 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*, air akan menjadi aktor utamanya sehingga animasi jenis ini membutuhkan banyak animasi air. Animasi jenis ini dibuat dengan teknik *frame by frame*, yaitu setiap gerakan air di-*capture* satu per satu dengan *camera*.

II.5. Gerak Lurus Beraturan

Sebuah mobil sedang melewati orang di tepi jalan. Mobil tersebut bergerak terhadap orang yang berada di tepi jalan, namun diam terhadap sopir mobil. Jadi untuk mengatakan benda itu bergerak atau diam, harus melihat dahulu apa yang dilakukan sebagai acuan atau pedoman. Oleh karena itu, *bergerak* atau *diam* adalah *relatif*, bergantung pada benda lain yang digunakan sebagai acuan.

Benda dikatakan bergerak, jika kedudukan benda itu mengalami perubahan terhadap acuan. Lintasan benda yang bergerak sangat beraneka ragam, ada yang berupa garis lurus, parabola, melingkar dan sebagainya. Gerak benda dapat diberi nama sesuai dengan lintasan yang dilaluinya. jadi, benda bergerak lurus jika lintasannya berupa garis lurus, gerak parabola, karena lintasan benda berupa parabola dan seterusnya. Ada beberapa besaran gerak lurus yang akan dibahas diantaranya jarak, perpindahan, kecepatan, dan percepatan.

II.5.1. Jarak dan Perpindahan

Sebuah benda yang melakukan gerak lurus, pembahasan geraknya dapat menggunakan sumbu x sebagai kerangka acuan. Perpindahan bergantung pada kedudukan awal dan akhir saja, tidak tergantung pada lintasannya. Perpindahan dinyatakan positif jika benda bergerak dari kiri ke kanan searah sumbu x positif (+). Sebaliknya perpindahan dinyatakan negatif (-) jika benda bergerak dari kanan ke kiri searah sumbu x negatif.

II.5.2. Kelajuan dan Kecepatan

1. Kelajuan dan Kecepatan Rata-Rata

Laju rata-rata adalah jarak yang di tempuh benda sepanjang lintasannya dibagi waktu yang diperlukan untuk menempuh jarak tersebut.

Secara matematis ditulis :

$$\bar{v} = \frac{s}{t}$$

Kecepatan rata-rata adalah perpindahan dibagi dengan waktu tempuh.

2. Laju Sesaat dan Kecepatan Sesaat

Besaran kecepatan sesaat pada saat tertentu adalah kemiringan garis yang menyinggungkurva posisi x terhadap waktu t pada saat itu

II.5.3. Percepatan dan Perlambatan

1. Percepatan

Percepatan (*acceleration*= a) adalah perubahan kecepatan tiap satu satuan waktu.

a. Percepatan rata-rata

Amati gambar 3.11. Diagram v dan t menggambarkan hubungan kecepatan dengan selang waktu gerak benda yang tidak beraturan. Karena gerakanya tidak beraturan maka percepatan untuk selang waktu (Δt) dinyatakan dengan percepatan rata-rata. jika ditulis dengan persamaan matematika, bentuknya adalah sebagai berikut.

$$\bar{a} = \frac{v_2 - v_1}{t_2 - t_1} = \frac{\Delta v}{\Delta t}$$

b. percepatan sesaat

Perhatikan kembali gambar 3.11. berdasarkan diagram $v-t$ tersebut, jika selang waktu Δt mendekati nol ($\Delta t \rightarrow 0$) maka titik A dan B hampir berimpit.

2. Perlambatan

Pernahkah anda melakukan perjalanan dengan api kereta api atau bus ? bagaimana kecepatan kereta api atau bus saat sudah mendekati stasiun atau terminal tujuannya? Sudah tentu kecepatan kereta api atau bus tersebut akan dikurangi dan sesampainya di tempat tujuan akan berhenti.

Gerak kereta api dan bus dalam mengurangi kecepatan ini disebut perlambatan. perlambatan sama dengan percepatan yang bertanda negatif (-) persamaan – persamaan yang digunakan pun sama dengan persamaan percepatan.

II.5.4. Gerak Lurus Beraturan(GLB)

Mungkin anda pernah memerhatikan speedometer pada mobil yang sedang bergerak hanya menunjuk pada angka tertentu. dalam hal ini, mobil yang anda tumpangi dapat dikatakan sedang melakukan gerak lurus beraturan.

Dari kegiatan fisika lab 3.1, mobil mainan berbaterai melakukan gerak lurus beraturan. sebuah benda dikatakan melakukan gerak lurus beraturan jika kecepatan setiap saat sama besar (tetap) sehingga untuk waktu yang sama, jarak tempuhnya sama pula. jadi,

Gerak lurus beraturan adalah gerak suatu benda yang lintasannya berupa garis lurus dengan kecepatan tetap.

$$v = \frac{s}{t}$$

$$v = \frac{\Delta s}{t} = \frac{s - s_0}{t}$$

$$s = s_0 + vt$$

Jarak yang di tempuh benda (s) pada grafik v - t merupakan luas bidang yang dibatasi oleh garis grafik v dan sumbu t dalam selang waktu tertentu. Jadi, semakin besar sudutnya, semakin besar pula kecepatan gerak lurus beraturan tersebut.

II.6. Pengenalan 3Ds Max

3D Studio Max (3Ds Max) adalah *software visualisasi (modelling 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, 2012).

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*, *vektor* 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)*. (Mikael Sugianto 2011: 1-3).

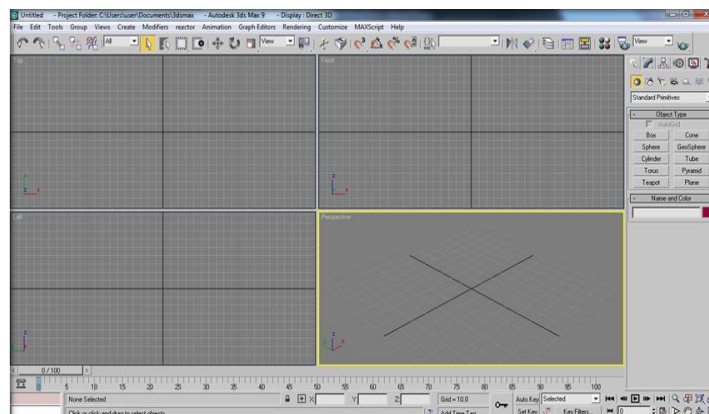
II.6.1. Area Kerja 3Ds Max 9

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.2. di bawah ini :



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

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.3.

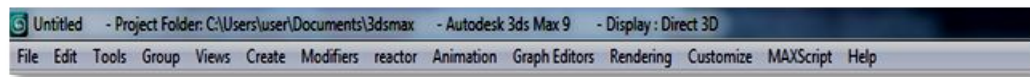


Gambar II.3. Tampilan Area Kerja 3Ds Max
 Sumber : (Galih Pranowo: 2010; 2)

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 Pranowo: 2010; 2)

II.6.3. Main Toolbar

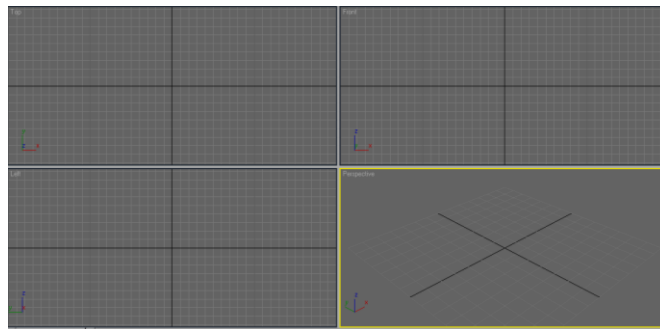
Main Toolbar terletak tepat di bawah *Menu Bar*. *Main Toolbar* menyediakan *shortcut* instruksi-instruksi praktis mulai dari penyelesaian objek, *Material Editor*, hingga *Rendering*. Adapun tombol-tombol dari *Main Toolbar* adalah sebagai berikut : *Select and Link, Unlink Selection, Bind to Space 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, Rendered Frame Window, Render Production, Render Iterative, ActiveShade*. Berikut ini merupakan gambar dari *Main Toolbar*.



Gambar II.5. Tampilan Main Toolbar
 Sumber : (Galih Pranowo: 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*.



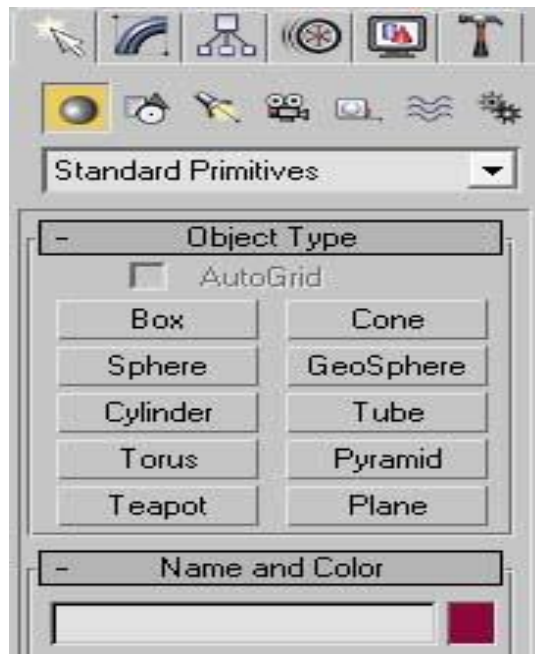
Gambar II.6. Tampilan Viewport

Sumber : (Galih Pranowo: 2010)

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*.

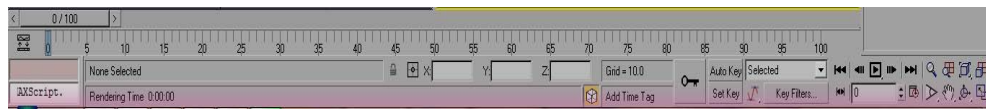


Gambar II.7. Tampilan Command Panel

Sumber : (Galih Pranowo: 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*.

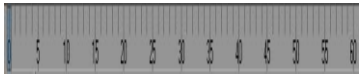


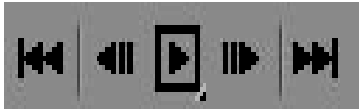
Gambar II.8. Tampilan *Timeline Animation*

Sumber : (Galih Pranowo: 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.1. Icon-Icon *Timeline Animation*

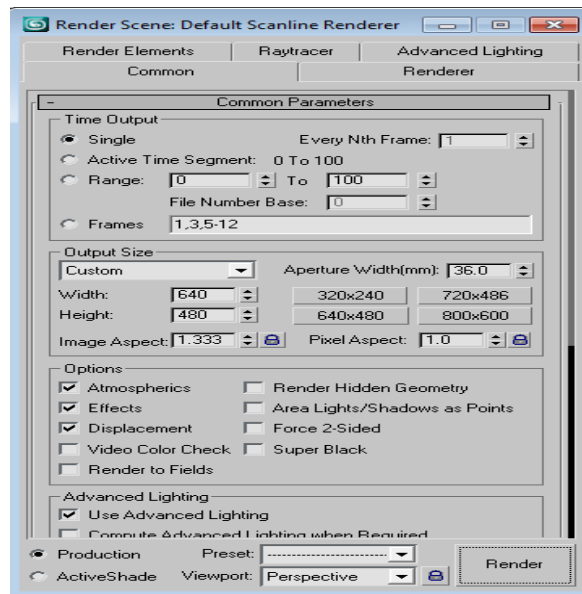
Ikons	Nama Fasilitas	Keterangan
	<i>Slider Timeline</i>	Merupakan fasilitas untuk memindah <i>frame</i> dari <i>timeline</i> atau untuk menentukan objek berada pada urutan <i>frame</i> tertentu.
	<i>Frame</i>	<i>Frame</i> adalah sebuah kolom yang berada pada <i>timeline</i> yang berfungsi untuk membuat 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 End</i>	Berfungsi untuk mengatur jalannya animasi, kembali ke awal animasi, kembali ke <i>frame</i> sebelumnya, memainkan animasi, menuju ke <i>frame</i> berikutnya, dan menuju ke akhir animasi.

Sumber : (Galih Pranowo: 2010; 14)

II.6.7. 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*.



Gambar II.9. Tampilan Kotak Dialog Rendering
Sumber : (Galih Pranowo: 2010)