

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Simulasi**

Simulasi merupakan salah satu cara untuk memecahkan berbagai persoalan yang di hadapi didunia nyata (*real world*). Banyak metode yang dibangun dalam *Operations Research dan system Analist* untuk kepentingan pengambilan keputusan dengan menggunakan berbagai analisis data.

Selain itu Simulasi juga dapat diartikan yaitu suatu program (*software*) komputer yang berfungsi untuk menirukan perilaku sistem nyata (realitas) tertentu. tujuan simulasi antara lain untuk pelatihan (*training*). studi perilaku sistem (*behaviour*) dan hiburan atau permainan (*game*) . beberapa contoh simulasi komputer, antara lain simulasi sistem ekonomi makro, simulasi sistem perbankan, simulasi antrian layanan bank (*service queue*), simulasi game strategis pemasaran (*market game*), simulasi perang (*war game simulation*), simulasi mobil (*car simulation*), simulasi tenaga listrik (*power plan simulation*) simulasi tata kota (*sim city*), simulasi waktu nyata (*real time*), merupakan bagian dari ilmu informatika (teknologi informasi) yang sedang berkembang sangat pesat saat ini.

#### **II.1.1 Jenis Simulasi**

Ada beberapa jenis sistem simulasi, yaitu sebagai berikut :

1. *Identity Simulation* (Simulasi Identitas)

Sistem Simulasi adalah suatu kegiatan yang memberikan pernyataan (*representing*) atas suatu dengan melalui model simbolik yang dapat

dimanipulasi dengan mudah dan dapat menghasilkan angka-angka atau bilangan – bilangan numerik.

*Range* ( jarak ) spektra dari sistem simulasi cukup luas. Yang lebih ekstrem lagi, kita dapat menggunakan sistem sebagai modal untuk mendapatkan pengetahuan atau sifat – sifat maupun tingkah laku didalam sistem itu sendiri.

## 2. *Quasi identity Simulation* (Simulasi Identitas Semu)

Simulasi ini selangkah lebih maju di banding *Identity Simulation*. Simulasi Identitas Semu ini memodelkan berbagai aspek yang terkait dari sistem yang sebenarnya dan dapat mengeluarkan unsur – unsur yang dapat membuat setiap *identity Simulation* tidak berfungsi dengan baik.

## 3. *Laboratory Simulation* (Simulasi Laboratorium)

Simulasi ini lebih mudah dan lebih layak daripada *Identity Simulation* dan *Quasi Identity Simulation* dan akan dapat memberikan jawaban yang lebih esensial pada masa yang akan datang. Biasanya Simulasi laboratorium ini memerlukan berbagai komponen, seperti *operator*, *software* dan *hardware*, komputer, prosedur operasional, fungsi – fungsi matematis, distribusi probabilitas dan lain-lain. ( Thomas J. Kakiay ; 2006 : 2 ).

## II.2. Animasi

Penggunaan animasi sedang berkembang saat ini, misalnya dalam penggunaan animasi pendidikan dan animasi instruksional. Bentuk klasik dari animasi adalah “ film kartun” yang dikembangkan diawal 1900 – an oleh UB

*Iwerks*, *Walt Disney*, dan yang lainnya. diperlukan hingga 24 pekerjaan menggambar yang berbeda untuk tampilan animasi selama satu detik.

Oleh karena pembuatan animasi sangat memakan waktu dan juga biaya, kebanyakan animasi untuk Televisi, dan gambar hidup datang dari studio animasi profesional. bagaimanapun juga, bidang dari animasi mandiri telah hidup sedikitnya sejak tahun 1910-an (yaitu: memelopori *stop-motion animator Ladislas Starevich* di *Russian Empire*). dengan animasi yang telah diproduksi oleh studio mandiri (dan kadang-kadang oleh seseorang secara individu). beberapa produser animasi mandiri sudah melanjutkan ke industri animasi yang profesional *Bill Plympton* adalah salah satu yang paling terkenal dari animator mandiri.

Sekarang, peningkatan program animasi yang mudah seperti *Macromedia Flash* dan bebas saluran distribusi seperti *Newgrounds* dan *deviantART* menyebabkan animator mandiri dan pekerjaan anda terlihat oleh (berpotensi) berjuta – juta orang, sehingga penggunaanya menjadi lebih mudah.

*Limited Animation* adalah suatu cara untuk meningkatkan produksi dan penurunan biaya animasi menggunakan “jalan pintas” didalam proses animasi. metode itu telah di pelopori oleh UPA dan di populerkan oleh *Hanna-Barbera*. dan diadaptasi oleh studio lain sebagai film kartun bergerak dari gedung bioskop ke Televisi.

Prinsip-prinsip dasar animasi yang berlaku, baik pada animasi 2D maupun animasi 3D adalah mengikuti gerakan yang terdapat di alam sekitar kita. Prinsip dasar animasi tersebut pertama kali dibuat oleh *Frank Thomas* dan *Mice Johnston* dalam karyanya yang berupa buku pada tahun 1981 *Toys Story* yang bernama

*John Lasseter* menyempurnakan prinsip-prinsip dasar animasi pada makalahnya yang berjudul *Principles of Traditional Animation Applied To 3D Computer Animation*. (Galih Pranowo ; 2010 : 4)

### **II.3. Dana**

Dana adalah suatu alat yang sangat penting yang di butuhkan oleh setiap perusahaan untuk menjalankan kegiatan operasionalnya, sedangkan menurut Kasmir pengertian dana adalah dana dianggap sebagai kas , sebagai uang yang disimpan di bank dalam bentuk giro atau tabungan , sebagai modal kerja , seluruh aktiva yang dimiliki perusahaan , sebagai aktiva yang dimiliki sifat sama dengan kas.” (2008: 17)

Pada dasarnya dana sangat mempengaruhi jenis usaha setiap perusahaan yang bersangkutan. Berikut adalah jenis-jenis dana :

#### 1. Dana Umum (*General fund*)

Yaitu dana untuk mempertanggungjawabkan sumber-sumber yang tidak dipertanggungjawabkan dalam dana lain.

#### 2 Dana Pendapatan Khusus (*Special Revenue Fund*)

Yaitu dana untuk mempertanggungjawabkan penerimaan sumber-sumber tertentu (selain yang dipertanggungjawabkan dalam dana *trust* belanja dan proyek modal) yang ditujukan untuk aktivitas tertentu.

#### 3 Dana Proyek Modal (*Capital Project Fund*)

Yaitu dana untuk mempertanggungjawabkan sumber-sumber yang digunakan untuk tujuan perolehan atau pembangunan fasilitas modal (selain yang dipertanggungjawabkan melalui dana kepemilikan dan dana *trust*).

4. Dana Pelunasan Utang (*Debt Service Fund*)

Yaitu dana untuk mempertanggungjawabkan pengakumulasian sumber-sumber untuk membayar pokok dan bunga utang jangka panjang umum.

5. Dana Perusahaan (*Enterprise Fund*)

Yaitu dana untuk mempertanggungjawabkan aktivitas bisnis (komersial) yang dilakukan oleh pemerintah.

6. Dana Layanan Internal (*Internal Service Fund*)

Yaitu dana untuk mempertanggungjawabkan barang/jasa yang disediakan oleh suatu unit pemerintah kepada unit pemerintah itu sendiri atau kepada unit pemerintah lain.

7. Dana Trust (*Trust Fund*)

Yaitu dana untuk mempertanggungjawabkan aktiva milik pihak lain yang dikelola oleh pemerintah sebagai pihak yang dipercaya atau *trustee*.

8. Dana Trust Belanja (*Expendable Trust Fund*)

Yaitu dana untuk mempertanggungjawabkan sumber keuangan yang diterima dari unit pemerintah lain, organisasi, dan perseorangan yang dapat dibelanjakan sesuai dengan *trust agreement* atau aturan dari pemberi (donor).

9. Dana Trust Non Belanja (*Non Expendable Trust Fund*) Yaitu dana untuk mempertanggungjawabkan yang diterima suatu unit pemerintah yang tidak bersifat belanja sesuai dengan *agreement* atau aturan dari pemberi (donor).

10. Dana Trust Pensiun (*Pension Trust Fund*)

Yaitu dana untuk mempertanggungjawabkan penerimaan dan pembayaran dari pemerintah, karyawan atau pihak lain untuk tujuan dana pensiun, penggunaan

sumber-sumber dana yang diterima dan perhitungan serta pembayaran kepada para pensiun.

#### 11. Dana Peragenaan (*Agency Fund*)

Yaitu dana untuk mempertanggungjawabkan aktiva milik pihak-pihak lain yang dikelola oleh pemerintah yang bertindak sebagai agen.

### II.4. Tabungan

Tabungan adalah menyimpan sebagian dari pada pendapatan yang diperoleh. Biasanya jumlah yang di cadangkan ialah 30 % dari pada jumlah pendapatan. hal ini bertujuan menggalakan simpanan untuk masa depan mengurangi sikap suka berhutang dan serta memupuk sikap berjiwa cermat di kalangan para pengguna

Walaupun banyak mana pun uang yang di peroleh , jika kita tidak merancang keuangan dengan betul segala uang yang di peroleh akan “hilang” begitu saja. Amat penting untuk kita merancang perbelanjaan harian kita dan bulanan dan simpanan sebagian jaminan masa depan selain untuk kegunaan saat-saat yang tidak di duga.

### II.5. *Unified Modeling Language* (UML)

UML singkatan dari *Unified Modelling Language* yang berarti bahasa pemodelan standart. (Chonoles; 2003 : 6) mengatakan sebagai bahasa, berarti *UML* memiliki sintaks dan *semantic*. Ketika kita membuat model menggunakan konsep *UML* ada aturan –aturan yang harus diikuti. Bagaimana elemen pada

model-model yang kita buat harus berhubungan satu dengan lainnya harus mengikuti standart yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya? Bagaimana sistem mengatasi error yang terjadi? Bagaimana keamanan terhadap sistem yang ada kita buat? Dan sebagainya dapat dijawab dengan *UML*.

*UML* diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

*UML* telah diaplikasikan dalam investasi perbankan, lembaga kesehatan, departemen pertahanan, sistem terdistribusi, sistem pendukung alat kerja, retail, sales, dan supplier.

Blok pembangunan utama *UML* adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasikan objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. *UML* memungkinkan para anggota team untuk bekerja sama dalam mengaplikasikan beragam sistem. Intinya, *UML* merupakan alat komunikasi yang konsisten dalam mensupport para pengembang sistem saat ini. Sebagai perancang sistem mau tidak mau pasti menjumpai *UML*, baik kita sendiri yang membuat

sekedar membaca diagram *UML* buatan orang lain (Prabowo Pudjo Widodo Herlawati ; 2011 : 6).

### **II.5.1.Diagram-Diagram UML**

Beberapa literatur menyebutkan bahwa *UML* menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa yang digabung, misalnya diagram komunikasi, diagram urutan, dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain :

1. Diagram Kelas. Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi, serta relasi-relasi diagram. Diagram ini umu dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas.
2. Diagram paket (*PackageDiagram*) bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas merupakan bagian dari diagram komponen.
3. Diagram *Use Case* bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

4. Diagram interaksi dan *Sequence* (urutan). Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam waktu tertentu.
5. Diagram komunikasi (*Communication Diagram*) bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi *UML* yang menekankan organisasi *structural* dari objek-objek yang menerima serta mengirim pesan.
6. Diagram *Statechart* (*Statechart Diagram*) bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*State*), transisi kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.
7. Diagram aktivitas (*Activity Diagram*) bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan member tekanan pada aliran kendali antar objek.
8. Diagram komponen (*Component Diagram*) bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan diagram kelas dimana komponen dipetakan kedalam satu atau lebih kelas-kelas. Antarmuka-antarmuka serta kolaborasi-kolaborasi.

9. Diagram *Deployment (Deployment Diagram)* bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run time*). Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram *Deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).
- Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai dengan kebutuhan.

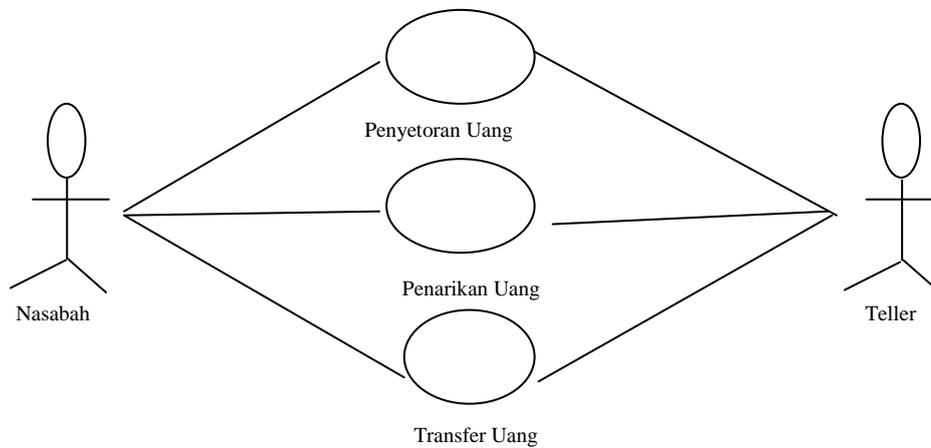
### **II.5.2 Diagram Use Case (*use case diagram*)**

*Use Case* menggambarkan *external view* dari sistem yang akan kita buat modelnya. Menurut Pooley (2005:15) mengatakan bahwa model *use case* dapat dijabarkan dalam diagram, tetapi yang perlu diingat, diagram tidak identik dengan model karena model lebih luas dari diagram.

Komponen pembentuk diagram *use case* adalah :

- a. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
- b. *Use Case*, aktivitas/ sarana yang disiapkan oleh bisnis/sistem.
- c. Hubungan (*Link*), aktor mana saja yang terlibat dalam *use case* ini.

Gambar di bawah ini merupakan salah satu contoh bentuk diagram *use case*

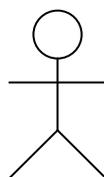


**Gambar II.2. Diagram *Use Case***

**Sumber : Probowo Pudjo Widodo (2011:17)**

### 1. Aktor

Menurut Chonoles (2003 :17) menyarankan sebelum membuat use case dan menentukan aktornya, agar mengidentifikasi siapa saja pihak yang terlibat dalam sistem kita. Pihak yang terlibat biasanya dinamakan *stakeholder*.

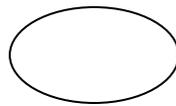


**Gambar II.3. Aktor**

**Sumber : Probowo Pudjo Widodo (2011:17)**

## 2. *Use Case*

Menurut Pilone (2005 : 21) *use case* menggambarkan fungsi tertentu dalam suatu sistem berupa komponen kejadian atau kelas. Sedangkan menurut Whitten (2004 : 258) mengartikan *use case* sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario) baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal. *Use case* digambarkan dalam bentuk *ellips/oval*.



**Gambar II.4. Simbol *Use Case***

**Sumber : Probowo Pudjo Widodo (2011:22)**

*Use case* sangat menentukan karakteristik sistem yang kita buat, oleh karena itu Chonoles (2003:22-23) menawarkan cara untuk menghasilkan *use case* yang baik yakni :

### **a. Pilihlah nama yang baik**

*Use case* adalah sebuah *behaviour* (prilaku), jadi seharusnya dalam frase kata kerja. Untuk membuat namanya lebih detil tambahkan kata benda mengindikasikan dampak aksinya terhadap suatu kelas objek. Oleh karena itu diagram *use case* seharusnya berhubungan dengan diagram kelas.

### **b. Ilustrasikan perilaku dengan lengkap.**

*Use case* dimulai dari inisiasi oleh aktor primer dan berakhir pada aktor dan menghasilkan tujuan. Jangan membuat *use case* kecuali anda mengetahui tujuannya. Sebagai contoh memilih tempat tidur (*King Size*,

*Queen Size*, atau dobel) saat tamu memesan tidak dapat dijadikan *use case* karena merupakan bagian dari *use case* pemesanan kamar dan tidak dapat berdiri sendiri (tidak mungkin tamu memesan kamar tidur jenis king tapi tidak memesan kamar hotel).

**c. Identifikasi perilaku dengan lengkap.**

Untuk mencapai tujuan dan menghasilkan nilai tertentu dari aktor, *use case* harus lengkap. Ketika memberi nama pada *use case*, pilihlah frasa kata kerja yang implikasinya hingga selesai. Misalnya gunakan frasa *reserve a room* (pemesanan kamar) dan jangan *reserving a room* (memesan kamar) karena memesan menggambarkan perilaku yang belum selesai.

**d. Menyediakan *use case* lawan (*inverse*)**

Kita biasanya membutuhkan *use case* yang membatalkan tujuan, misalnya pada *use case* pemesanan kamar, dibutuhkan pula *use case* pembatalan pesanan kamar.

**e. Batasi *use case* hingga satu perilaku saja.**

Kadang kita cenderung membuat *use case* yang lebih dari satu tujuan aktivitas. Guna menghindari kerancuan, jagalah *use case* kita hanya fokus pada satu hal. Misalnya, penggunaan *use case* *check in* dan *check out* dalam satu *use case* menghasilkan ketidakfokusan, karena memiliki dua perilaku yang berbeda.

### II.5.3. Diagram Kelas (*Class Diagram*)

Diagram kelas mempunyai dua jenis yaitu *domain class diagram* dan *design class diagram*. Fokus *domain class diagram* adalah pada sesuatu dalam lingkungan kerja pengguna, bukan pada *class* perangkat lunak yang nantinya akan anda rancang. Sedangkan *design class diagram* tujuannya adalah untuk mendokumentasikan dan menggambarkan kelas-kelas dalam pemrograman yang nantinya akan dibangun.



**Gambar II.5. Notasi *Domain Diagram Class***

**Sumber : E. Triandini dan G. Suardika (2012 : 49 -50)**



**Gambar II.6. Notasi *Design Diagram Class***

**Sumber : E. Triandini dan G. Suardika (2012 : 49 -50)**

#### **II.5.4. Diagram Aktivitas (*Activity Diagram*)**

Diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem dirakit. Diagram ini tidak hanya memodelkan software melainkan memodelkan bisnis juga. Diagram aktivitas menunjukkan aktivitas sistem dalam kumpulan aksi-aksi. Ketika digunakan dalam pemodelan *software*, diagram aktivitas merepresentasikan pemanggilan suatu fungsi tertentu misalnya *call*. Sedangkan bila digunakan dalam pemodelan bisnis, diagram ini menggambarkan aktivitas yang dipicu oleh kejadian-kejadian diluar seperti pemesanan atau kejadian-kejadian internal misalnya penggajian tiap jumat sore (Probowo Pudji Widodo ;2011 : 143-145).

Aktivitas merupakan kumpulan aksi-aksi. Aksi-aksi melakukan langkah sekali saja tidak boleh dipecah menjadi beberapa langkah-langkah lagi. Contoh aksinya yaitu :

- a. Fungsi Matematika
- b. Pemanggilan Perilaku
- c. Pemrosesan Data

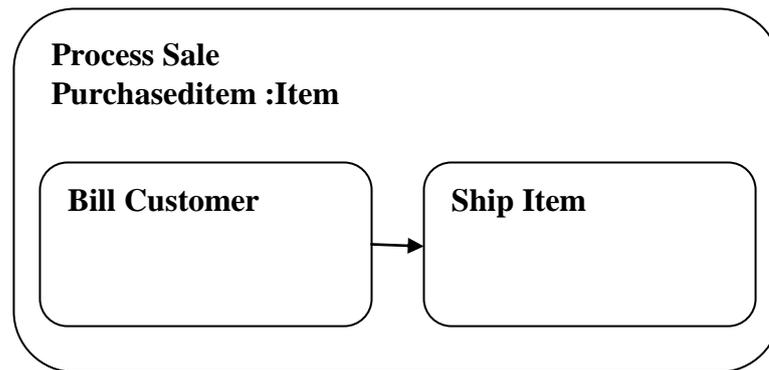
Ketika kita menggunakan diagram aktivitas untuk memodelkan perilaku suatu *classifier* dikatakan kontek dari aktivitas. Aktivitas dapat mengakses atribut dan operasi *classifier*, tiap objek yang terhubung dan parameter-parameter jika aktivitas memiliki hubungan dengan perilaku. Ketika digunakan dengan model proses bisnis, informasi itu biasanya disebut *process-relevant data*. Aktivitas diharapkan dapat digunakan ulang dalam suatu aplikasi, sedangkan aksi biasanya *specific* dan digunakan hanya untuk aktivitas tertentu.



**Gambar II.7. Aktivitas sederhana tanpa rincian**

**Sumber : Probowo Pudjo Widodo (2011:145)**

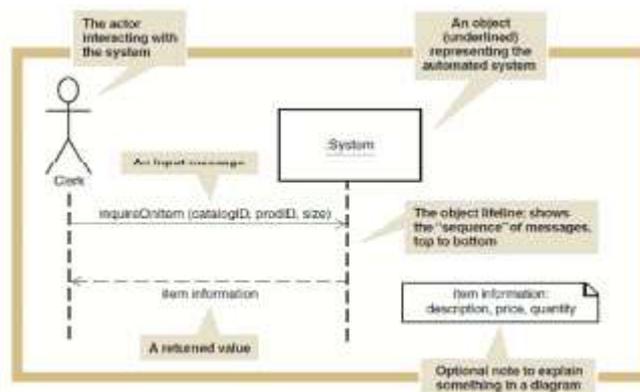
Detail aktivitas dapat dimasukkan di dalam kotak. Aksi diperlihatkan dengan symbol yang sama dengan aktivitas dan namanya diletakkan didalam persegi panjang.



**Gambar II.8. Aktivitas dengan detail rincian**  
**Sumber : Probowo Pudjo Widodo (2011:145)**

### II.5.5 Sequence Diagram

Menurut John Satzinger, 2010, dalam buku *System Analysis and Design in a Changing World*, “System Sequence Diagram (SSD) adalah diagram yang digunakan untuk mendefinisikan input dan output serta urutan interaksi antara pengguna dan sistem untuk sebuah use case.



**Gambar II.9. Notasi Sequence Diagram**  
**Sumber : Evi Triandini dan Gede Suardika (2012 : 71)**

## II.6. Pengertian 3Ds Max

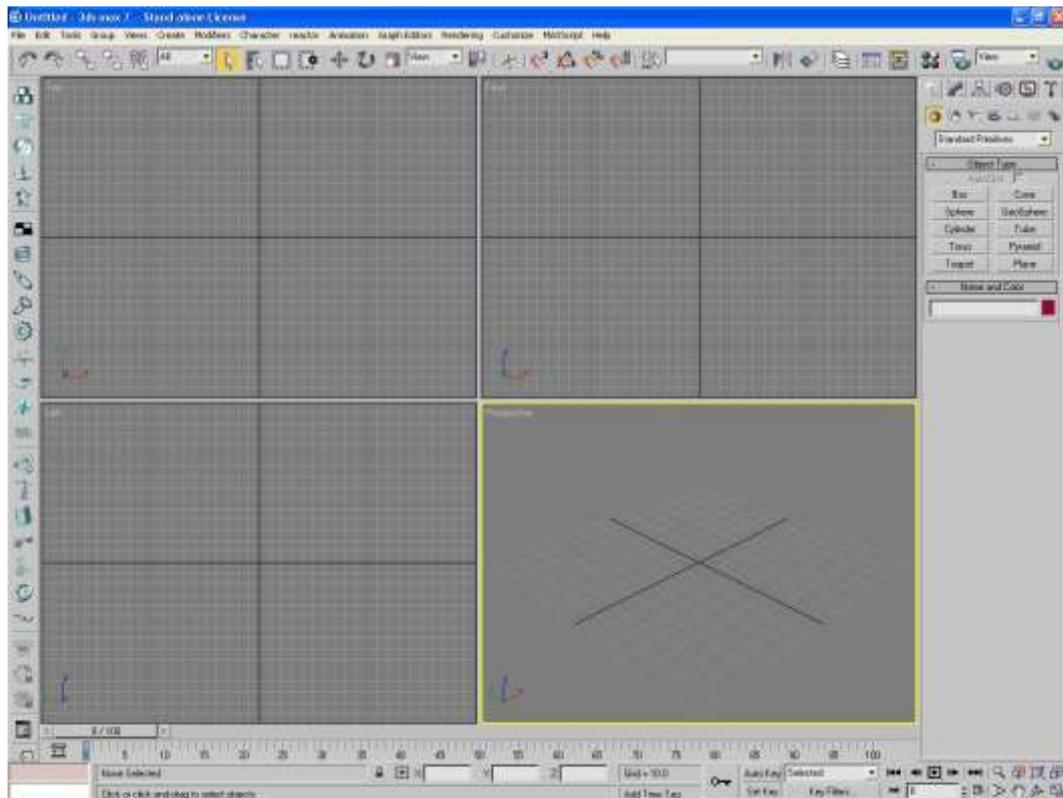
3Ds Max adalah sebuah *software* yang dikhususkan dalam permodelan 3 dimensi ataupun untuk pembuatan animasi 3 dimensi. Selain terbukti andal untuk digunakan dalam pembuatan objek 3 dimensi, 3Ds Max juga banyak digunakan dalam pembuatan desain *furniture*, konstruksi, maupun desain interior. Selain itu, 3Ds Max juga sering digunakan dalam pembuatan animasi atau film kartun.

3Ds Max yang di lengkapi dengan bahasa scripting (*maxScript*) juga terbukti ampuh untuk membuat game 3 dimensi, mulai dari yang sederhana hingga yang rumit sekalipun. Dengan kemampuan tersebut, banyak orang maupun instansi menggunakan 3Ds Max untuk membuat suatu desain atau iklan yang berguna sebagai media publikasi produk atau karya mereka pada public. 3Ds Max memungkinkan pengguna untuk membuat tampilan 3 dimensi yang sangat menarik.

Hingga saat ini, 3Ds Max telah sampai pada versi terbarunya, yaitu 2010 setelah sebelumnya merilis versi 3Ds Max 2009, 2008, 2007, dan beberapa versi lainnya. Dalam hal tampilan atau ragam *tool*, tidak ada perubahan yang signifikan pada versi 3Ds Max 2010 dibandingkan versi sebelumnya (2009). Letak perbedaannya hanyalah pada material dan tools nya. (Galih Pranowo; 2010 : 1)

### II.6.1 Area Kerja Max

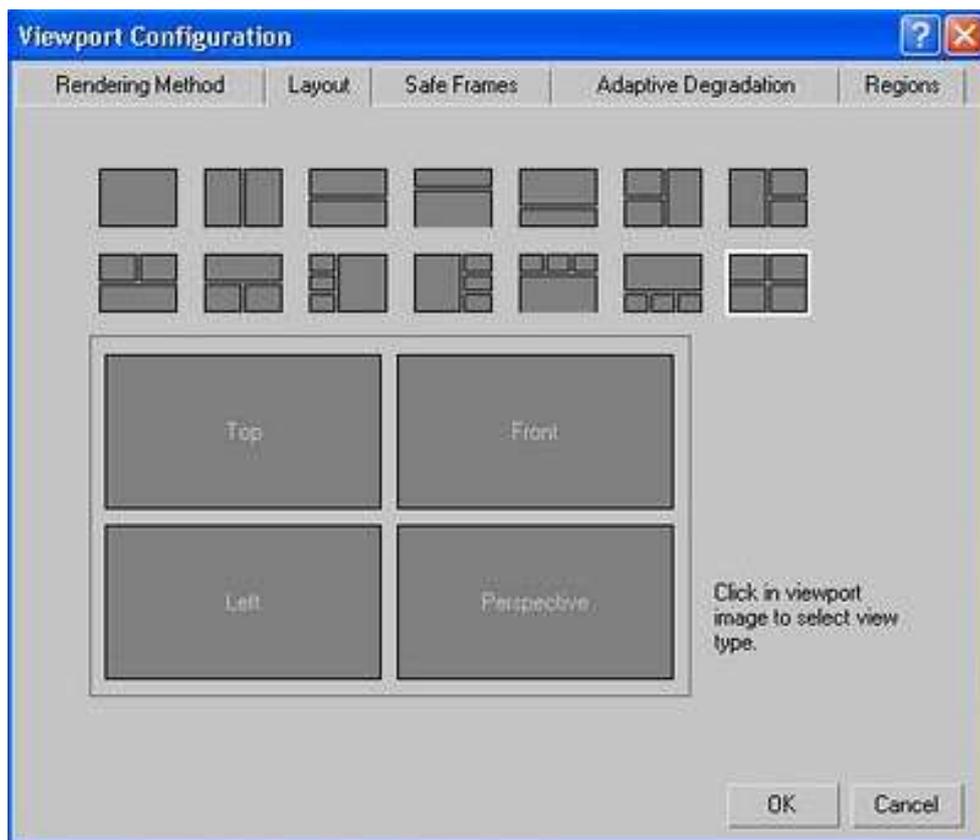
Max adalah program computer truly 3D atau 3 dimensi sepenuhnya. Semua gambar yang kita hasilkan dapat dilihat dari berbagai sudut pandang secara 3D, bahkan dari bawah.



**Gambar II.1. Area Kerja Max**

**Sumber : (Gilang Wiradinata ; 2007 : 1)**

Default area kerja max terdiri atas empat pandangan (*viewport*), yaitu pandangan atas (*Top*), depan (*Front*), kiri (*Left*), dan *Perspective*. Untuk berganti *viewport* aktif, tekan huruf awal *viewport* seperti L untuk *Left*, T untuk *Top*, dan sebagainya. (Gilang Wiradinata ; 2007 : 1)



**Gambar II.2. Pilihan Viewport**

**Sumber : (Gilang Wiradinata ; 2007 : 2)**

### **II.6.2. Title Bar**

Bagian kiri *Title Bar* terdapat *Control* Menu untuk standar aplikasi windows dan keterangan yang menampilkan nama aplikasi dan nama file yang aktif. Pada bagian kanan *Title Bar* terdapat tombol pengaturan tampilan antara lain :

- a. *Minimize*, untukl memperkecilkan ukuran tampilan aplikasi sehingga membentuk ikon aktif pada *Taskbar*.
- b. *Restore*, untuk mengembalikan tampilan aplikasi ke ukuran sebelumnya.

- c. *Maximize*, Untuk memperbesar ukuran tampilan aplikasi menjadi satu layar.
- d. *Close*, untuk menutup aplikasi.

### II.6.3. Menu Bar

*Menu Bar* berisi menu-menu untuk mengakses berbagai perintah pada 3D Studio Max. Menu bar terletak tepat dibawah *Title Bar* dan untuk menggunakannya arahkan *pointer* ke menu yang diinginkan , kemudian klik hingga terbuka menu *drop-down* yang berisi beberapa submenu seperti pada gambar II.3 :



File Edit Tools Group Views Create Modifiers reactor Animation Graph Editors Rendering Customize MAXScript Help

**Gambar II.3. Menu Bar**

**Sumber : (Gilang Wiradinata ; 2007 : 3)**

### II.7. Adobe Premiere Pro Cs4

*Adobe Premiere Pro Cs4* dibuat oleh *Adobe System Inc*, yang merupakan versi pembaharuan dari versi *Adobe Premiere* terdahulu. *Adobe Premiere Pro Cs4* mulai di luncurkan ke pasaran pada bulan Agustus tahun 2003. Mulai versi 6.0 *Adobe Premiere* dapat digunakan untuk mengedit file video dengan format *DV*. *Adobe Premiere* telah menjadi program standar bagi para profesional dalam bidang digital video. Fitur baru *Adobe Premiere Pro Cs4* pada versi baru ini *Adobe Premiere Pro Cs4* hadir dengan tampilan baru, tetapi lebih simple dari para

pendahulunya. Penamaan efek video serta transisinya sedikit mengalami perubahan sehingga membutuhkan waktu bagi para pengguna versi lama (6.0 dan 6.5) untuk menyesuaikan diri dalam tampilan yang baru ini, beberapa fasilitas baru yang ada dalam *Adobe Premiere Pro Cs4* adalah :

1. *Real Time Rendering*, dengan fasilitas ini memungkinkan melihat hasil transisi tanpa melalui proses render.
2. *Adobe Product Integration*, yaitu kita dapat menggunakan berbagai produk *Adobe* (*Adobe Photoshop*, *Adobe After Effect*, *Adobe Encore DVD* dan *Adobe Audition*) untuk bekerja sama dengan *Adobe Premiere Pro Cs4*.
3. *Audio Mixer*, pada versi ini *Adobe* telah memperbaharui fasilitas audio mixer nya, yang mendukung *track-based effects*, *submix tracks*, mendukung proses perekaman *voiceover* langsung (*dubbing*), fasilitas *punning* untuk *surround* dan *stereo* dan *VST filter*.
4. *Color Correction Adobe Premiere Pro Cs4* mulai versi ini melengkapi dirinya dengan fasilitas eksklusif ini. fasilitas ini biasanya terdapat pada *software editing standar broadcast (high-end)*. Dengan adanya fasilitas ini kita dapat dengan mudah mengkoreksi warna clip video kita sesuai dengan keinginan.
5. *Motion Path*, pada versi yang baru ini *Adobe Premiere Pro Cs4* menerapkan prinsip *keyframing* untuk animasi perpindahan posisi pada klip-klip nya.
6. *Visual Effect* yang dilengkapi dengan keyframe, sehingga dapat dianimasikan dengan mudah. Fasilitas ekspor dalam berbagai format. *Adobe Premiere Pro Cs4* menyediakan fasilitas ekspor kedalam berbagai format media seperti

*MPEG-1, MPEG-2, Microsoft DV, AVI* dalam berbagai kompresi, *Windows Media 9 Series dan Real Media 9*.

Kebutuhan komputer untuk menjalankan *Adobe Premiere Pro Cs4* PC (Personal Computer) digunakan untuk mengolah file video baik proses *capture* maupun editing menggunakan *software Adobe Premiere Pro Cs4*, PC yang akan digunakan minimal harus memiliki spesifikasi sebagai berikut :

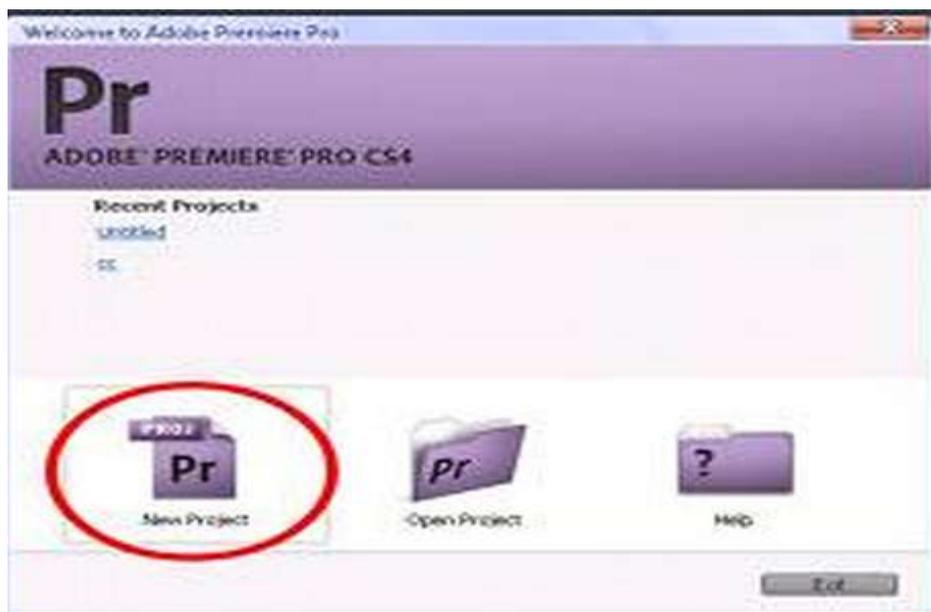
1. *Processor Intel® Pentium® III – 800 Mhz.*
2. *RAM 256 Mb.*
3. *Harddisk* berkecepatan 7200 rpm dengan kapasitas besar (*space* kosong minimal 20 Gb).
4. *VGA Card* 32 Mb (resolusi monitor 1024 x 768 pixel).
5. *Sound Card* yang mendukung *DirectX*.
6. *CD-ROM/RW* dan *DVD-ROM/RW* untuk menulis dalam format *VCD* maupun *DVD*.
7. *DV Capture Card/IEEE 1394 Card.*
8. *Sistem Operasi Windows XP Home/Pro Edition.*

Untuk menggunakan *Adobe Premiere Pro Cs4* maka harus menggunakan *Microsoft Windows XP* sebagai sistem operasinya. Hal ini dikarenakan *Adobe Premiere Pro* hanya dapat berjalan pada sistem operasi tersebut. lain dengan *Adobe Premiere* versi 6.5 yang masih dapat berjalan pada sistem operasi *Windows 98SE* dan *Windows 2000*.

### II.7.1. Interface

Memahami tampilan awal jalankan program *Adobe Premiere Pro Cs4* yang telah kita instal dengan cara pilih *Start > All Programs > Adobe Premiere Pro Cs4*.

Tampilan awal program seperti gambar berikut.



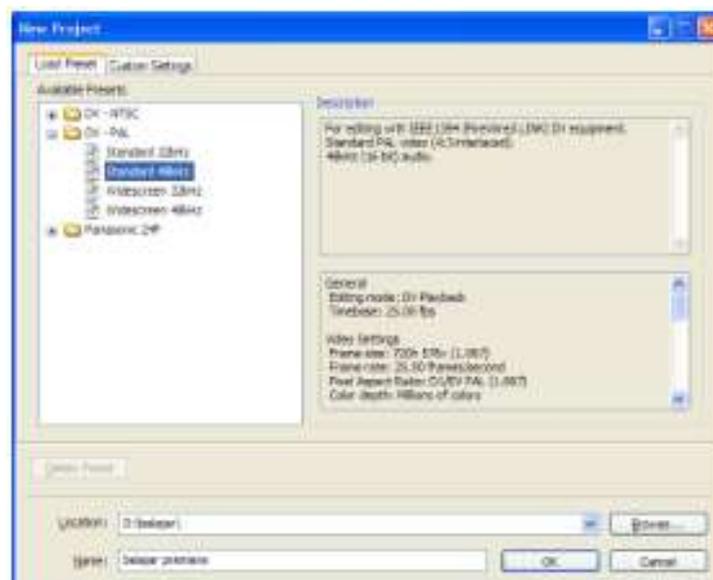
**Gambar II.4. Kotak Dialog Pembuka**

**Sumber : (Fandi ; 2010 : 1)**

Membuat *project* baru dan mengatur seting dasar setelah menjalankan *Adobe Premiere Pro* maka langkah selanjutnya adalah membuat *project* baru dan mensetingnya, langkahnya :

1. Klik tombol *New Project* yang terdapat pada kotak dialog pembuka, maka akan tampil kotak dialog *New Project*.

2. Pada kotak dialog *New Project* atur *Available Presets* dengan pilihan DV-PAL Standard 48 KHz, 48 KHz menyatakan rate audio ketika direkam.
3. Jika DV *Camecorder* menggunakan format video NTSC, pilih DV-NTSC Standard 48 KHz, lalu pilihlah lokasi penyimpanan file dengan klik tombol *Browse*.
4. Isikan nama *Project* pada *text box name* dengan nama baru. Selanjutnya klik OK, untuk melanjutkan , maka akan tampil area kerja *Adobe Premiere Pro Cs4*.



**Gambar II.5. Kotak Dialog New Project**

**Sumber : (Fandi ; 2010 : 2)**

## **II.6.2. Mengenal Area Kerja *Adobe Premiere Pro Cs4***

Garis besar lingkungan kerja *Adobe Premiere Pro Cs4* terdiri dari empat bagian utama yaitu :

1. *Project Window*, yang berada pada sebelah kiri atas.
2. *Monitor Window*, yang berada pada sebelah kanan atas.

3. *Timeline Window*, yang berada pada sebelah kanan bawah.
4. *Tools Window*, yang berada pada sebelah kiri bawah.



**Gambar II.6. Lingkungan Kerja Adobe Premiere Pro**

**Sumber : (Fandi ; 2010 : 3)**