

BAB III

LANDASAN TEORI

3.1 Game

3.1.1. Pengertian Game

Game merupakan kata dalam bahasa Inggris yang berarti permainan. Permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks tidak serius atau dengan tujuan *refreshing*. Suatu cara belajar yang digunakan dalam menganalisa interaksi antara sejumlah pemain maupun perorangan yang menunjukkan strategi-strategi yang rasional.

Teori permainan pertama kali ditemukan oleh sekelompok ahli Matematika pada tahun 1944. Teori itu dikemukakan oleh John von Neumann and Oskar Morgenstern yang berisi: "Permainan terdiri atas sekumpulan peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun untuk memaksimalkan kemenangan sendiri atau pun untuk meminimalkan kemenangan lawan". Peraturan-peraturan menentukan kemungkinan tindakan untuk setiap pemain, sejumlah keterangan diterima setiap pemain sebagai kemajuan bermain, dan sejumlah kemenangan atau kekalahan dalam berbagai situasi.

3.1.2. Sejarah Game

Pada tahun 1952, seorang mahasiswa Universitas *Cambridge* bernama A.S. Gouglas membuat permainan OXO (*tic-tac-toe*) dalam versi grafik. Permainan ini ia kembangkan ketika hendak mendemonstrasikan tesisnya tentang interaksi antara manusia dan komputer. Pada tahun 1958 William Higin Botham mendesain sebuah *Game* dengan judul *Tennis For Two* yang dimainkan dalam *oscilloscope*, dan kemudian ada pula Steve Russel pada tahun 1961 dengan *Game* berjudul *Spacewar* yang dibuat dalam komputer mainframe DEC PDP-1 saat mereka menjalani studi di MIT. Sejarah *video Game* tidak hanya mengenai orang-orang yang berperan di dalamnya, tetapi juga mengenai berbagai perusahaan *Game* yang mempunyai kasus ironis. *Atari* adalah perusahaan Amerika dengan nama Jepang, dan perusahaan Jepang bernama SEGA didirikan oleh orang Amerika. *Magnavox* yang memulai peran perusahaan ini telah berumur 1 abad, dan *Nintendo* sebagai perusahaan yang mempopulerkan kembali *video Game* juga sama tuanya, serta tidak ada yang pernah berpikir bahwa *Sony* yang merupakan perusahaan penemu banyak barang-barang elektronik mulai dari transistor radio hingga alat perekam *video*, akan membuat sebuah *console* yang menjadi produk dengan tingkat penjualan yang tinggi hingga saat ini.

3.1.3. Jenis – jenis Game

Terdapat beberapa jenis *Game* yang dapat dikelompokkan sebagai berikut:

1. Simulasi, Contoh permainan yang termasuk dalam *Game* simulasi adalah simulasi konstruksi dan manajemen, simulasi kendaraan seperti yang diterapkan pada permainan balapan, perang, luar angkasa, dan *mecha*.
2. Edukasi, Contohnya adalah *eduGames* yang dibuat dengan tujuan spesifik sebagai alat pendidikan, baik itu untuk belajar mengenal warna untuk balita, mengenal huruf dan angka, matematika, sampai belajar bahasa asing. *Developer* yang membuatnya, harus memperhitungkan berbagai hal agar *Game* ini benar-benar dapat mendidik, menambah pengetahuan dan meningkatkan ketrampilan yang memainkannya.
3. *Entertainment*
 - a. Aksi – *Shooting*, (tembak-tembakan , atau hajar-hajaran bisa juga tusuk-tusukan, tergantung cerita dan tokoh di dalamnya). *Game* jenis ini sangat memerlukan kecepatan refleks, koordinasi mata-tangan, juga waktu. inti dari *Game* jenis ini adalah tembak-tembakan.
 - b. *Fighting* (pertarungan), ada yang mengelompokkan *Game fighting* di bagian Aksi, namun penulis berpendapat berbeda, jenis ini memang memerlukan kecepatan refleks

dan koordinasi mata-tangan, tetapi inti dari *Game* ini adalah penguasaan jurus (hafal caranya dan lancar mengeksekusinya), pengenalan karakter dan waktu sangatlah penting. Dan berbeda seperti *Game Aksi* pada umumnya yang umumnya hanya melawan *Artificial Intelligence* atau istilah umumnya melawan komputer saja, pemain jenis *fighting Game* ini baru teruji kemampuan sesungguhnya dengan melawan pemain lainnya.

- c. Petualangan, *Game* murni petualangan lebih menekankan pada jalan cerita dan kemampuan berpikir pemain dalam menganalisa tempat secara visual, memecahkan teka-teki maupun menyimpulkan 10 rangkaian peristiwa dan percakapan karakter hingga penggunaan benda-benda tepat pada tempat yang tepat.
- d. *Role Playing, Game* jenis ini sesuai dengan terjemahannya, bermain peran, memiliki penekanan pada tokoh/peran perwakilan pemain di dalam permainan, yang biasanya adalah tokoh utamanya, dimana seiring kita memainkannya, karakter tersebut dapat berubah dan berkembang ke arah yang diinginkan pemain (biasanya menjadi semakin hebat, semakin kuat, semakin

berpengaruh, dll) dalam berbagai parameter yang biasanya ditentukan dengan naiknya *Level*.

- e. *Casual Games*, sesuai namanya, *Game* yang *casual* itu tidak kompleks, mainnya rileks dan sangat mudah untuk dipelajari. Jenis ini biasanya memerlukan spesifikasi komputer yang standar pada jamannya dan ukurannya tidak lebih dari 100 MB karena biasanya dapat di *download* versi demo-nya di *website* resminya. *Genre* permainannya biasanya *puzzle* atau *action* sederhana dan umumnya dapat dimainkan hanya menggunakan.
- f. *Multiplayer Online*, *Game* yang dapat dimainkan secara bersamaan oleh lebih dari 2 orang (bahkan dapat mencapai puluhan ribu orang dalam satu waktu) membuat pemain dapat bermain bersama dalam satu dunia *virtual* dari sekedar *chatting* hingga membunuh naga bersama teman yang entah bermain di mana. Umumnya permainan tipe ini dimainkan di PC dan bertema RPG, walau ada juga yang bertema musik atau *action*.

3.2 Pengertian Edukasi

Pengertian edukasi adalah proses pengajaran secara formal maupun non formal kepada seseorang atau lebih dari satu orang baik secara bersama-sama ataupun individu. Sekolah adalah salah satu sarana untuk edukasi yang memberikan banyak manfaat bagi peserta didik. Edukasi di sekolah mampu mensosialisasikan anak didik untuk mendidik mereka menjadi generasi yang lebih baik. Edukasi adalah hal yang sangat penting bagi bangsa dan kata edukasi ini sangat familiar di masyarakat luas.

3.3 Pengertian Suara

Pengertian suara adalah bunyi yang dikeluarkan dari mulut manusia. Bunyi biasa di keluarkan ketika bercakap-cakap, menyayi, tertawa, dan menangis. Pengertian suara misalnya pada kalimat : penyayi itu merdu suaranya.

Pengertian suara juga bisa berarti bunyi binatang, alat perkakas dan sebagainya. Misalnya, kita bisa menemukan penggunaan istilah suara ini pada kalimat kedengaran suara harimau mengaung atau suara burung bernyanyi.

3.4 Pengertian dan Sejarah Animasi

3.4.1. Sejarah Animasi

Animasi sudah ada sejak pada zaman batu. Sebagai contoh adalah goa Spanyol Utara, *Altamira* di goa ini terdapat banyak lukisan-lukisan di dindingnya. Lukisan itu sudah diteliti sudah ada sejak ribuan taun yang lalu yang menurut catatan adalah zaman manusia purba. Diantara banyak gambar 43 lukisan, ada tampak sebuah gambar yang lainnya, gambar itu adalah gambar seekor babi hutan dengan kepala banyak, dan memiliki 8 buah kaki dan itu adalah film animasi di jaman purba. Di Indonesia wayang kulit diperkirakan merupakan bentuk animasi tertua. Karena wayang memenuhi semua elemen animasi seperti layar, gambar bergerak, dialog dan ilustrasi musik.

Pada zaman ini Jepang pun sudah mulai membuat animasi gaya gambarnya masih mirip dengan animasi barat, akan tetapi cara penyampaiannya berbeda. Dimana orang barat menggunakan cara penyampainya yang kaku, sementara Jepang menggunakan animasi yang bersifat menghibur, seperti menggunakan tokohnya seperti hewan. Satu-satunya manusia dalam animasi propraganda perang buatan Jepang adalah *Momotaro* yang dalam kisah *Momotaro* adalah lelaki kecil yang berjuang melawan setan dengan teman-temannya yaitu hewan-hewan.

Hal ini sedikit menyindir, karena bagi orang-orang Jepang, setan itu memiliki mata berwarna biru (beberapa orang Eropa memiliki mata berwarna biru). *Momotaro* adalah lelaki kecil yang pemberani, melambangkan bahwa Jepang adalah negara kecil yang berani. Dan pada akhirnya, *Momotaro* menang melawan setan. Ini artinya orang Jepang berharap akan menang melawan orang-orang Eropa.

Animasi mulai berkembang sekitar abad ke-18 di Amerika. Pada saat itu teknik *stop motion animation* banyak disenangi. Teknik ini menggunakan serangkaian gambar diam atau *frame* yang dirangkai menjadi satu dan menimbulkan kesan seolah-olah gambar tersebut bergerak. Teknik ini sangat sulit. Membutuhkan waktu, juga biaya banyak. Karena untuk menciptakan animasi selama satu detik, dibutuhkan sebanyak 12-24 *frame* gambar diam. Bayangkan jika film animasi itu berdurasi satu jam bahkan lebih. J. Stuart Blackton mungkin adalah orang Amerika pertama yang menjadi pionir dalam menggunakan teknik *stop motion animation*. Beberapa film telah diciptakannya dengan menggunakan teknik ini adalah *The Enchanted Drawing (1900)* dan *Humorous Phases of Funny Faces (1906)*.

3.4.2. Pengertian Animasi

Animasi diambil dari bahasa Latin '*anima*' yang artinya jiwa, hidup, nyawa, semangat. Animasi itu sendiri memiliki arti yaitu

gambar dua dimensi seolah-olah bergerak karena kelamahan mata yang selalu menyimpang atau mengingat di otak, imajinasi yang terlihat sebelumnya. Dalam animasi kartun diartikan sebagai sekumpulan gambar yang berubah sedikit demi sedikit, apabila di tampilkan secara berurutan dengan kecepatan tertentu ([www.animatorforum.org](http://www animatorforum.org)).

Ketika berbicara tentang *flash*, maka tidak terlepas pula pembicaraan tentang animasi itu sendiri. Pada dasarnya, terdapat dua teknik animasi di dalam program *flash*. Yaitu animasi *Frame by frame* dan animasi *Tween*.

3.4.3. Animasi Frame by Frame

Teknik animasi ini merupakan teknik yang cukup kompleks dan rumit. Hal ini disebabkan pada animasi *frame by frame*. Setiap *frame* pada *timeline* terisi oleh *keyframe-keyframe* yang didalamnya terdapat objek-objek yang berbeda. Sederhananya, teknik animasi ini 45 hampir sama seperti teknik pembuatan film kartun di zaman dulu ketika belum terdapat alat-alat secanggih komputer.

Dimana untuk menghasilkan sebuah animasi, seorang *animator* harus membuat objek yang berjumlah lebih dari satu gambar atau objek yang berbeda, baik itu ukuran ataupun posisinya. Animasi *frame by frame* memiliki kekurangan dikarenakan harus membuat objek-objek dengan bentuk dan posisi yang berbeda pada setiap *keyframe* untuk menghasilkan suatu gerakan yang bagus.

Namun untuk sebagian orang yang cukup mempunyai kesabaran, animasi ini cukup menjadi *favorit*. Hal itu disebabkan ketika mengoptimalkan teknik animasi tersebut, dapat menghasilkan suatu animasi dengan gerakan mulus dan detail.

Berikut contoh animasi menggunakan teknik animasi *Frame by Frame* :



Gambar 3.1 Animasi *Frame by frame* pada *frame 1*



Gambar 3.2 Animasi *Frame by frame* pada *frame 2*



Gambar 3.3 Animasi *Frame by frame* pada *frame 3*

3.4.4. Animasi Tween

Teknik *Animasi tween* dapat dikatakan jauh lebih mudah dibandingkan dengan Animasi *frame by frame*. Pada teknik ini, tidak perlu membuat objek satu per satu pada setiap *framena*. Konsep dasar Animasi *Tween* ini adalah hanya perlu membuat paling sedikitnya menempatkan objek di dua *keyframe*, yaitu pada *keyframe* awal dan *keyframe* terakhir, sedangkan objek-objek yang berbeda antara *keyframe* awal dan *keyframe* akhir tersebut akan diatur secara otomatis. Satu hal yang perlu diperhatikan bahwa satu objek yang dianimasikan membutuhkan satu *layer* untuk menempatkan objek tersebut. 47 Pada program *Flash* terdapat dua jenis Animasi *Tweening*, yaitu *Montion Tween* dan *Shape Tween*. Berikut penjelasannya :

a. Animasi *Motion-Tween*

Motion memiliki arti pergerakan. Pada intinya teknik animasi ini berhubungan dengan pergerakan suatu objek dari

keadaan awal hingga keadaan akhir objek tersebut. Oleh karena jenis objek yang menjadi persyaratan untuk dapat dianimasikan dengan Animasi *Motion-Tween* ini adalah objek yang telah dijadikan *symbol*, baik berupa *Move Clip*, *Button*, dan *Graphic*, maka kata ‘keadaan’ pada pernyataan sebelumnya memiliki arti yang cukup luas, yaitu meliputi *property symbol* yang terdiri atas posisi (X dan Y), skala (W dan H).

b. Animasi *Motion-Tween* dengan Menggunakan *Guide*

Teknik animasi *Motion-Tween Guide* memvariasikan alur gerakan animasi dari suatu objek dengan menggunakan alur yang dibuat sendiri. Dalam hal ini menggunakan teknik modifikasi yang disebut dengan *motion-tween guide*. Arah gerakan suatu animasi disesuaikan dengan alur yang dibuat pada *Motion Guide layer*.

Motion Guide layer adalah suatu *layer* tambahan pada *Timeline* yang keberadaannya berfungsi sebagai tempat membuat alur gerakan animasi yang pada umumnya alur tersebut terbentuk dari sebuah garis. Garis alur yang dibuat pada *Motion Guide Layer* ini akan nampak ketika bekerja di *stage*, kecuali pada *layer* tersebut diklik *Hide Layer*. Namun ketika *Flash movie* dijalankan, garis alur tersebut tidak akan terlihat sama sekali dan yang terlihat hanya animasi objek yang bergerak mengikuti garis alur yang telah dibuat.

c. Animasi *Shape-Tween*

Pada dasarnya, teknik animasi *Shape-Tween* sama seperti pada teknik animasi *Motion-Tween*. Hanya saja yang mejadi perbedaan mendasar yang digunakan, yaitu hanya untuk objek yang berjenis *shape*. Sering kali animator menggunakan teknik animasi ini untuk menggambarkan perubahan bentuk suatu objek dari satu bentuk ke bentuk lainnya.

Sedangkan karakter animasi sendiri dibedakan menjadi :

1. Animasi 2D (2 Dimensi)

Animasi ini lebih dikenal dengan istilah kartun yang berasal dari kata '*cartoon*' artinya gambar yang lucu. Dengan teknik ini, beberapa gambar 2D atau gambar yang mempunyai sudut pandang terbatas ditampilkan berurutan dengan aplikasi pembuat *film*.

2. Animasi 3D (3 Dimensi)

Perkembangan teknologi dan komputer membuat teknik pembuatan animasi 3D semakin berkembang dan maju pesat. Animasi 3D adalah pengembangan dari animasi 2D. Dengan animasi 3D, karakter yang diperlihatkan semakin hidup dan nyata, mendekati wujud manusia aslinya. Dengan animasi 3D karakter dapat dilihat dari banyak sudut pandang (atas, bawah, depan, belakang, kiri, kanan dan *perspektif*).

3. Animasi Tanah Liat (*Clay Animation*)

Animasi dengan teknik seperti ini dilakukan dengan menggunakan tanah liat. Tanah liat yang dipakai bukanlah tanah liat melainkan '*plasticin*', bahkan lentur seperti permen karet yang ditemukan pada tahun 1897. Tokoh-tokoh dalam animasi *clay* dibuat dengan memakai rangka khusus untuk kerangka tubuhnya, lalu kerangka tersebut ditutup dengan *plasticine* sesuai bentuk tubuh tokoh yang dibuat. Bagian-bagian tubuh kerangka ini, seperti kepala, tangan, kaki bisa dilepas dan dipasang lagi. Setelah tokoh-tokohnya siap, lalu difoto gerakan pergerakannya. Foto-foto tersebut lalu digabung menjadi gambar yang bisa bergerak seperti yang ada di film. *Animasi clay* termasuk jenis animasi dari '*Stop Motion Picture*'.

4. Animasi Jepang

Anime adalah sebuah sebutan untuk animasi buatan Jepang yang diperuntukkan tidak hanya anak kecil melainkan orang dewasa juga. Yang membedakan *anime* dengan animasi buatan Eropa atau Amerika adalah formatnya, yaitu serial televisi, OVA, dan *film* bioskop.

3.5. Perangkat Lunak

Perangkat lunak yang digunakan untuk membuat *Game* edukasi mengenal suara dan gambar binatang untuk anak-anak adalah *Adobe Flash CS3* dan *Adobe Audition 1.5*. Berikut akan dijelaskan lebih jauh mengenai *Adobe Flash CS3* dan *Adobe Audition 1.5*.

3.5.1. Adobe Flash CS3

Adobe Flash CS3 adalah program standar profesional yang digunakan untuk membuat animasi interaktif. Program ini berbasis *vector* grafis, jadi aksesnya lebih cepat dan terlihat halus pada skala resolusi layar berapapun. Program ini juga dapat diisi dengan bitmap yang diimpor dari program lain. Salah satu keunggulannya adalah ukurannya yang begitu kecil namun dapat menampilkan animasi yang mengagumkan. *Flash* juga mempunyai kemampuan untuk membuat animasi secara *streaming*, yaitu dapat menampilkan animasi langsung meskipun proses *download* atau *loading* belum selesai seluruhnya. Selain itu, dengan *Adobe Flash CS3* juga dapat dibuat *movie* kartun dan aplikasi interaktif yang memungkinkan pengguna dapat berinteraksi langsung dengan aplikasi yang dibuat.

3.5.2. Memahami Adobe Flash CS3

a. *Work Space*

Work Space atau bisa disebut *Paste Board* adalah bidang berwarna abu abu di layar kerja *Flash*. Bidang ini dimanfaatkan untuk proses pembuatan gambar maupun teks yang akan

digunakan untuk bahan animasi. Objek yang terletak di bidang ini pada saat animasi dijalankan tidak akan ikut ditampilkan dalam animasi karena hanya digunakan untuk persiapan.

b. Stage

Stage dapat diibaratkan layar pada gedung bioskop, yaitu bidang dimana animasi ditampilkan saat *movie* anda dimainkan. *Stage* berupa bidang berwarna putih di layer *Flash*, dikelilingi berwarna abu-abu yang telah dikenalkan di depan yaitu *work space*.

c. Toolbox

Toolbox atau panel adalah kotak yang berisi alat-alat kerja. Dari *tool-tool* yang ada dapat dikelompokkan sesuai kegunaan menjadi 4, yaitu:

1. *Tool* untuk berkreasi yaitu : *Tool* untuk memilih obyek : *Drawing* alat untuk mengubah gambar, dan teks *Tool* untuk membuat tulisan.
2. Untuk mengatur bagaimana gambar ditampilkan terdiri dari *hand* dan *zoom tool*.
3. *Tool* untuk memilih dan membuat warna : untuk mewarnai garis dan gambar.
4. *Options* adalah *tool-tool* yang merupakan pengembangan atau *tool* yang terpilih.

d. *Panel*

Panel pada *file* adalah sarana untuk mengatur, menampilkan dan mengubah elemen yang ada dalam dokumen. Opsi-opsi yang ada dalam *panel* berfungsi untuk mengendalikan warna, *library*, *symbol* dan *instance*, frame serta elemen-elemen lain.

e. *Help*

Fasilitas *search* membantu dalam menemukan dengan cepat tentang masalah cara menuliskan *script* dengan *Action Script*.

f. *Library*

Feature sebagai penampung segala objek seperti *Instance Button*, *Graphic*, *Movie Clip*, dan *Sound*.

g. *Development Panel*

Development ini lebih berguna untuk mengoperasikan *script* dalam *web programming*.

3.5.3. Adobe Audition

Adobe Audition merupakan suatu program yang digunakan untuk merekam, mengedit suara dalam bentuk digital yang berbasis *Windows*. Program ini dilengkapi dengan modul-modul efek suara, seperti *Delay*, *Echo*, Pereduksi *Noise/Hiss*, *Reverb*, Pengatur *Tempo*, *Pitch*, *Graphic* dan *Parametric Equalizer*.

Adobe Audition memberikan fasilitas perekaman suara sampai dengan 128 *track* hanya dengan satu *sound card*, hal ini akan

memberikan kemudahan bagi seorang *sound editor* untuk berekspresi lebih jauh.

Edit suara bisa dilakukan dalam bentuk *.wav*. Dan penyimpanan bisa di *convert* dalam bentuk format seperti *.wma*, *.mp3*, *mp3pro*, dll. Dalam arrangement sebuah musik bisa dilakukan dengan menambahkan beberapa alat musik dan dikoneksikan dengan *line in* atau *michrophone* dari *soundcard*.

Contoh lain dalam proses perekaman yang terjadi dalam suatu radio seperti *News* (berita), umumnya melalui beberapa tahap :

1. Tahap *Take Voice*. Tahapan ini berupa pengambilan suara-suara yang diperlukan sesuai dengan kebutuhan *script*. Pengambilan suara bisa dilakukan diluar (*outdoor*) studio atau di dalam studio (*indoor*). Untuk *outdoor take voice* dilakukan dengan sebuah alat perekam seperti *tape recorder* atau sejenisnya. Sedangkan untuk *indoor* dilakukan didalam studio misalnya *take voice* oleh presenter. Kelebihan dari *take voice indoor* akan dihasilkan kualitas suara yang lebih bagus karena bebas dari suara-suara yang luar yang tidak perlu.
2. *Loading + Editing*. Tahap ini memasukan suara hasil *take voice* kedalam perangkat *editing*. Dari perangkat ini suara akan diedit sesuai dengan kebutuhan, seperti menghilangkan *noise*, atau *equalize* suara sebelum di campur (*mixing*) dengan suara-suara lain.

3. *Mixing* Tahap pencampuran suara-suara sesuai dengan suara yang dibutuhkan berdasarkan kebutuhan *script*. Mengatur suara latar (*back sound* misalnya *music*, *effect*, *smash* dan lain sebagainya), level volume suara dilakukan di tahap ini.
4. *Transferring*. Memindahkan suara hasil *mixing* ke media penyimpanan seperti kaset, harddisk, dan lain sebagainya.

Untuk semua tahapan diatas dapat dilakukan dengan sebuah alat bantu *Digital Sound Editor* (DSE) yaitu *Adobe Audition* yang bekerja di sebuah perangkat *Personal Computer*.

Program DSE banyak sekali macamnya seperti *Cool Edit*, *Soundforge*, *Mix Vibes*, *Adobe Audition*, dan masih banyak yang lainnya. Dari semua DSE mempunyai kekurangan dan kelebihan masing-masing.

Beberapa kelebihan atau fasilitas yang disediakan *Adobe Audition* :

1. *Multitrack* Editing dan *Mixing* sampai 128 track
2. Lebih dari 40 data *sound effect*, *mastering*, perangkat analisa *editing* suara terdapat didalamnya, semua ini termasuk *Echo*, *Reverb*, *Flanging*, *Chorusing*, *Compression*, *Limiting*, *Equalization*, *Noise Reduction*, dan lainnya.
3. Pengorganisasian *track* yang lebih mudah.
4. *Support* untuk banyak format audio.
5. Visualisasi dari gelombang suara analisa dalam bentuk angka.

6. Proses editing dilakukan dengan klik & drag

3.6. Perancangan Storyboard

3.6.1. Pengetian Storyboard

Storyboard adalah sketsa gambar yang disusun berurutan sesuai dengan naskah, dengan *storyboard* kita dapat menyampaikan ide cerita kita kepada orang lain dengan lebih mudah, karena kita dapat menggiring khayalan seseorang mengikuti gambar-gambar yang tersaji, sehingga menghasilkan persepsi yang sama pada ide cerita kita.

Storyboard merupakan langkah yang harus dibuat setelah mendefinisikan elemen-elemen ke dalam objek-objek. Tampilan dalam *storyboard* dilakukan secara berurutan dan diberi penjelasan yang spesifik tentang apa yang ditampilkan pada *layer*. *Storyboard* hampir sama seperti *flowchart* (alur *Storyboard* tidak bisa dipisahkan dengan struktur navigasi Rancangan *storyboard* harus selalu disesuaikan dengan struktur navigasi yang telah dibuat, sehingga *design storyboard* menjadi terarah.

3.6.2. Sejarah Storyboard

Storyboard yang kita kenal saat ini, pertama kali dikembangkan oleh *Walt Disney Studio* sejak tahun 1930-an. Menurut *Diane Disney Miller*, *storyboard* pertama kali diciptakan tahun 1933 yang merupakan evolusi dari buku komik *Walt Disney*

yang berjudul "*Story Sketches*" pada tahun 1920. *Storyboard* menjadi populer semenjak tahun 1940-an ketika memproduksi film "*live-action*" dan berkembang menjadi media standar untuk previsualisasi film.

3.6.3. Penggunaan Storyboard

Karena *storyboard* merupakan alur proses (*flowcart*) dari sebuah cerita, maka *storyboard* sangat banyak digunakan pada kegiatan-kegiatan seperti di bawah ini:

1. Film

Pada dasarnya, sebuah film adalah komik besar dari film atau sebagian film yang diproduksi terlebih dahulu untuk membantu sutradara, sineas dan iklan komersial dalam memvisualisasikan adegan dan menemukan masalah potensial sebelum terjadi.

2. Teater

Praktisi teater Rusia, *Constantin Stanislavski* mengembangkan *storyboard* untuk merencanakan produksi adegan-adegan teater secara terperinci.

3. Animasi

Pada animasi, *storyboard* digunakan untuk mendapatkan ide yang lebih baik dalam pengaturan waktu dan gerakan.

4. Foto-Animasi (*Photomatic*)

Foto-Animasi (*photomatic*) adalah serangkaian foto yang masih diedit bersama dan disajikan dalam layar secara berurutan.

5. Buku komik

Beberapa penulis, telah menggunakan gambar *storyboard* untuk *scripting* buku komik mereka. yaitu dengan menempatkan tokoh, latar belakang gambar dan balon teks.

6. Bisnis

Storyboard saat ini banyak digunakan oleh industri untuk perencanaan promosi iklan dan presentasi bisnis agar lebih memudahkan *audience* dalam memahami maksud dari iklan atau presentasi tersebut.

7. Media interaktif

Storyboard digunakan pada pengembangan *web*, pengembangan *software* dan desain instruksi untuk menjelaskan secara detail disertai dengan audio.

8. Software

Storyboard digunakan dalam pengembangan perangkat lunak sebagai bagian dari mengidentifikasi spesifikasi untuk perangkat lunak tertentu dan membuat petunjuk penggunaan.

3.7. UML (Unified Modeling Language)

3.7.1. Pengenalan UML

UML (*Unified Modeling Language*) merupakan pengganti dari metode analisis berorientasi objek dan *design* berorientasi objek (OOA&D) yang dimunculkan sekitar akhir tahun 80-an dan awal tahun 90-an.

UML merupakan gabungan dari metode *Booch*, *Rumbaugh* (OMT) dan *Jacobson*. Tetapi UML ini akan mencakup lebih luas dari pada OOA&D. Pada pertengahan pengembangan UML dilakukan standarisasi proses dengan OMG (Object Management Group) dengan harapan UML akan menjadi bahasa standar pemodelan pada masa yang akan datang.

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat.

Bahasa pemodelan merupakan bagian terpenting dari metode. Ini merupakan bagian kunci tertentu untuk komunikasi. Jika anda ingin berdiskusi tentang desain dengan seseorang, maka anda hanya membutuhkan bahasa pemodelan bukan proses yang digunakan untuk mendapatkan desain.

UML merupakan bahasa standar untuk penulisan *blueprint software* yang digunakan untuk visualisasi, spesifikasi, pembentukan dan pendokumentasian alat-alat dari sistem software.

3.7.2. Sejarah Singkat UML

Bahasa pemodelan berorientasi objek muncul antara sekitar pertengahan tahun 1970-an dan akhir tahun 1980-an yang dikenal dengan bahasa pemrograman berorientasi objek dan aplikasi kompleks yang berkembang, yang dimulai untuk eksperimen dengan pendekatan alternatif untuk analisis dan desain. Sejumlah metode berorientasi objek bertambah dari kurang lebih 10 sampai lebih dari 50 selama periode 1989 dan 1994. Beberapa user pengguna metode ini menemukan permasalahan dalam bahasa pemodelan ini yang dibutuhkan mereka untuk kelengkapan, sehingga timbul yang dinamakan perang metode. Belajar dari pengalaman, metode generasi baru mulai muncul dengan metode yang terkemuka, seperti *Booch*, *Jacobson's OOSE (Object Oriented Software Engineering)* dan *Rumbaugh's OMT (Object Modelling Technique)*. Metode penting lainnya seperti *Fusion*, *Shler_mellor* dan *Coad-Yourdan*. Setiap metode ini merupakan metode yang lengkap, meskipun setiap metode diakui memiliki kelebihan dan kekurangan. Dalam waktu yang singkat metode *Booch* paling terasa dalam mendesain dan membangun tahapan *project*, OOSE memberikan dukungan yang baik untuk *use cases* seperti cara untuk menjalankan permintaan,

analisis dan desain level tinggi, dan OMT-2 sangat berguna untuk analisis dan sistem informasi data intensif.

Banyak ide-ide yang kritis dimulai dari pertengahan tahun 1990-an ketika *Grady Booch (Relational Software Corporation)*, *Ivar Jacobson (Objectory)* dan *James Rumbaugh (General Electric)* mulai mengadopsi ide-ide dari metode lainnya yang dikumpulkan yang akhirnya diakui sebagai Metode *Object Oriented* yang mudah diseluruh dunia. Kemudian mereka termotivasi untuk membangun UML (*Unified Modelling Language*).

Ada tiga tujuan dibangunnya penyatuan metode tersebut yaitu :

- a. Untuk memodelkan sistem, dari konsep ke bentuk yang cocok dengan menggunakan teknik berorientasi objek.
- b. Untuk menunjukkan skala persoalan yang kompleks.
- c. Untuk membangun bahasa pemodelan yang berguna bagi manusia dan mesin.

Perencanaan bahasa untuk digunakan pada analisa dan desain yang berorientasi objek tidak seperti mendesain bahasa pemograman. Pertama, kita harus mengetahui masalah seperti apakah bahasa mencakup spesifikasi permintaan? Dapatkah bahasa penting untuk pemograman visual? Kedua, kita harus menemukan keseimbangan antara kompleks dan kesederhanaan. Bahasa yang terlalu sederhana akan terbatas untuk problem yang luas yang akan

dipecahkan. Sedangkan untuk bahasa yang kompleks akan berakibat terlalu pengembang pada sistem yang sederhana.

UML dimulai secara resmi pada Oktober 1994, ketika *Rumbaugh* bergabung dengan *Booch* pada *Relational Software Corporation*. Proyek ini memfokuskan pada penyatuan metode *Booch* dan OMT. Versi 0.8 merupakan Metode Penyatuan yang di *release* pada bulan Oktober 1995. Dalam waktu yang sama *Jacobson* bergabung dengan *Relational* dan cakupan dari UML semakin luas sampai diluar perusahaan OOSE. Dokumentasi UML versi 0.9 akhirnya direlease pada bulan Juni 1996. Meskipun pada tahun 1996 ini melihat dan menerima *feedback* dari komunitas *Software Engineering*. Dalam waktu tersebut menjadi lebih jelas bahwa beberapa organisasi software melihat kalau UML merupakan strategi dari bisnisnya. Kemudian dibangunlah UML *Consortium* dengan beberapa organisasi yang akan menyumbangkan sumber dayanya untuk bekerja mengembangkan dan melengkapi UML.

Disini beberapa patner yang berkontribusi pada UML 1.0 diantaranya *Digital Equipment Corporation, Hewlett-packard, I-Logix, Intellicorp, IBM, ICON Computing, MCI Systemhouse, Microsoft, Oracle, Relational, Texas Instruments dan Unisys*. Dari *Colaboration* ini dihasilkan UML 1.0 yang merupakan bahasa pemodelan yang ditetapkan secara baik, *Expressive*, kuat dan cocok untuk lingkungan masalah yang luas. UML 1.0 ditawarkan menjadi

standarisasi dari *Object Management Group* (OMG). Dan pada Januari 1997 sebagai standar bahasa pemodelan.

Antara Januari – Juli 1997 Gabungan group tersebut memperluas kontribusinya sebagai hasil respon dari OMG dengan memasukkan *Adersen Consulting, Ericsson, ObjectTimeLimeted, Platinum Technology, Ptech, Reich Technologies, Softeam, Sterling Software* dan *Taskon*. Revisi dari versi UML(versi 1.1) ditawarkan kepada OMG sebagai standarisasi pada bulan Juli 1997. Dan pada bulan September 1997 versi ini diterima oleh OMG *Analysis and Design Task Force* (ADTF) dan *OMG ArchitectureBoard*. Dan akhirnya pada Juli 1997 UML versi 1.1 menjadi standarisasi.

Pemeliharaan UML terus dipegang oleh *OMG Revision Task Force* (RTF) yang dipimpin oleh Cris Kobryn. RTP merilis editorial dari UML 1.2 pada Juni 1998. Dan pada tahun 1998 RTF juga merilis UML 3.1 dengan disertai dengan *user guide* dan memberikan *technical cleanup*.

3.7.3. Tujuan Penggunaan UML

- a. Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi objek.
- b. Menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.

3.7.4. Diagram dalam UML

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk ilustrasi bagian atau aspek tertentu dari sistem. Sebuah model sistem biasanya mempunyai beberapa diagram untuk setiap jenisnya.

Adapun jenis – jenis diagram antara lain :

a. *Use Case Diagram*

Menggambarkan sejumlah *eksternal actor* dan hubungannya ke *Use Case* yang diberikan oleh sistem. *Use Case* digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat *user*) dan bagaimana fungsi yang ada didalam sistem.

b. *Activity Diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *use case* atau interaksi.