

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Animasi

Animasi yang berasal dari kata dasar *to animate*, yang dalam artian menghidupkan. Perwujudan manusia untuk menghidupkan gambar-gambar yang terlihat bergerak sebagai perantara dari pengungkapan mereka. Gambar-gambar dinding gua pada zaman purba dapat menceritakan kegiatan kehidupan dan berburu yang di lakukan menjadi satu kesatuan cerita yang utuh, menandakan bahwa sejak zaman dahulu baik disadari atau pun tidak, manusia zaman purba dapat menggunakan gambar untuk membuat sebuah gambar hidup yang sekarang disebut sebagai animasi.

Menurut Utami (2011:44) mengatakan bahwa “animasi adalah rangkaian gambar yang membentuk sebuah gerakan”. Secara garis besar menurut Syahfitri (2011:215) jenis animasi dapat di kategorikan kedalam tiga bagian yaitu :

1. Animasi 2D (Dua Dimensi),

Animasi ini yang paling akrab dengan keseharian kita. Biasa disebut juga dengan film kartun. Kartun sendiri berasal dari kata Cartoon, yang berarti gambar yang lucu. Memang, film kartun ini kebanyakan film yang lucu.

2. Animasi 3D (Tiga Dimensi)

Perkembangan teknologi dan dunia komputer membuat teknik pembuatan animasi 3D semakin berkembang dan maju pesat. Animasi 3D adalah perkembangan dari animasi 2D. Dengan animasi 3D, karakter yang diperlihatkan semakin hidup dan nyata, mendekati wujud aslinya.

3. Animasi Tanah Liat (*Clay Animation*)

Meski namanya *clay* (tanah liat), namun yang dipakai bukanlah tanah liat biasa. Animasi ini menggunakan *palsticin*, bahan lentur seperti permen karet yang ditemukan pada tahun 1897. Tokoh-tokoh pada animasi *clay* dibuat dengan menggunakan rangka yang khusus untuk kerangka tubuhnya. Film animasi clay pertama kali dirilis bulan Februari 1908 berjudul, *A Sculptor's Web Rarebit Nigthmare*. Untuk beberapa waktu yang lalu juga, beredar film *clay* yang berjudul *Chicken Run*.

Menurut Adryanto (2010:6) dilihat dari teknik pembuatannya animasi juga dapat di bedakan kedalam tiga bagian, yaitu:

1. *Stop-motion animation*

Stop-motion animation sering pula disebut *claymation* karena dalam perkembangannya, jenis ini sering menggunakan *clay* (tanah liat) sebagai objek yang digerakkan. Teknik *stop-motion animation* merupakan animasi yang dihasilkan dari pengambilan gambar berupa objek (boneka atau yang lainnya) yang digerakkan setahap demi setahap. Dalam pengerjaannya teknik ini memiliki kesulitan dan memerlukan kesabaran yang tinggi. Wallace and Gromit dan Chicken Run , karya Nick Parks, merupakan salah satu contoh karya *stop motion animation*.

2. *Cell animation*

Cell animation sering disebut juga animasi tradisional karena teknik pengerjaannya dilakukan pada *celluloid transparent* yang sekilas mirip sekali dengan transparansi *OHP* yang sering kita gunakan. Pada pembuatan *cell animation*, setiap tahap gerakan digambar satu persatu

diatas *cell*. Contoh penggunaan teknik *cell animation* dalam industri film antara lain seperti Naruto, Shinchon dan lain-lain.

3. Animasi Komputer

Sesuai dengan namanya, animasi ini secara keseluruhan dikerjakan dengan menggunakan komputer. Dari pembuatan karakter, mengatur gerakan pemain dan kamera, pemberian suara, serta *special* efeknya semuanya dikerjakan dengan komputer. Dengan animasi komputer, hal-hal yang awalnya tidak mungkin digambarkan dengan animasi menjadi mungkin dan lebih mudah. Sebagai contoh perjalanan wahana ruang angkasa ke suatu planet dapat digambarkan secara jelas, atau proses terjadinya tsunami.

2.2 Teori Pendukung

Teori Pendukung merupakan alat yang digunakan untuk menggambarkan logika model dari suatu sistem dengan menggunakan simbol-simbol, lambing-lambang, ataupun diagram yang menunjukkan secara tepat arti dari fungsinya. Fungsi dari teori pendukung adalah untuk menjelaskan kepada pengguna bagaimana fungsi dari sistem informasi dapat bekerja dengan suatu bentuk logika model.

2.2.1. *Game*

Game memiliki makna yang berbeda-beda pada setiap kalangan. Secara garis besar dalam kamus bahasa Indonesia *game* berarti permainan suatu sarana ataupun arena hiburan baik dikalangan anak-anak maupun orang dewasa. Sedangkan pengertian *game* menurut Kimpraswil (Muhammad, 2009:26)

mendefinisikan bahwa “permainan adalah usaha olah diri (olah pikir dan fisik) yang sangat bermanfaat bagi peningkatan dan pengembangan motivasi, kinerja dan prestasi dalam melaksanakan tugas dan kepentingan organisasi dengan lebih baik”.

Lain halnya dengan Daeng (Ismail, 2009:17) mengungkapkan bahwa “permainan adalah bagian mutlak dari kehidupan anak dan permainan merupakan bagian integral dari proses pembentukan kepribadian anak”.

Ismail (2009:26) mengatakan bahwa :“pengertian permainan dapat di kelompokkan menjadi dua bagian. Pertama, permainan merupakan aktifitas bermain yang murni mencari kesenangan tanpa mementingkan menang atau kalah. Kedua, permainan merupakan arena bermain untuk mencari kesenangan dan kepuasan, namun ditandai dengan pencapaian menang-kalah”.

Menurut Agustinus dalam bukunya Pemrograman Animasi dan Game Profesional mengatakan “*game* merupakan permainan komputer yang dibuat dengan teknik dan metode animasi”. Jika ingin mendalami penggunaan animasi haruslah memahami pembuatan *game*. Atau, jika ingin membuat *game* haruslah memahami teknik dan metode animasi, sebab keduanya saling berkaitan.

Dalam perancangan *game* ini tidak lepas dari tools-tools pendukung yang digunakan yaitu antara lain:

1. *Adobe Flash Professional*

Adobe Flash Professional merupakan software utama yang digunakan dalam perancangan *game* ini. *Adobe Flash Professional* merupakan perkembangan dari macromedia flash player yang tidak hanya digunakan untuk pembuatan animasi saja, namun sekarang juga dapat digunakan dalam pembuatan *game* dan film.

Gerantabee dkk (2012:1) mengemukakan bahwa “*Adobe Flash Professional CS6 is the authoring environment for creating rich, interactive content and advertisements for digital, web delivery.*” Sedangkan menurut Chun (2010:1) “*Adobe Flash Professional CS5 provides a comprehensive authoring environment for creating digital animation and interactive Web sites. Flash is widelyused to create engaging applications rich in video, sound, graphics, and animation.*”Dapat disimpulkan bahwa *Adobe Flash Professional* merupakan suatu aplikasi untuk menciptakan animasi interaktif ataupun suatu desain demi kebutuhan tertentu dengan fitur yang mencakup *audio visual* yang mengesankan.

2. *Action Script*

Action script merupakan salah satu bagian dari *Adobe Flash Professional* yang dapat digunakan sebagai bahasa pemrograman untuk menciptakan animasi yang diinginkan. Menurut Radion (2012:6) “*Action Script* adalah bahasa yang dikenali dan digunakandalam *Flash* untuk melakukan berbagai hal dalam olah *movie*”.Dapat disimpulkan bahwa *Actionscript* merupakan bahasa pemrograman yang digunakan pada *Adobe Flash* dalam pengolahan *movie* yang memberikan perintah dengan memperoleh pengendalian terhadap *movie* itu sendiri.Dalam perancangan *game* ini juga menggunakan bahasa pemrograman *action script* yang bertujuan untuk memberikan visualisasi dan mengontrol pergerakan pada *game*. *Action script* yang digunakan dalam perancangan *game* ini menggunakan *action script 3.0*.

3. *Adobe Air*

Adobe air singkatan dari *Adobe Integrated Runtime* juga merupakan salah satu bagian didalam *adobe flash professional*.*Adobe air* merupakan perangkat

tambahan yang digunakan untuk menjadikan aplikasi *game* ini dapat di terapkan kedalam perangkat *mobile phone*. Menurut Chamber (2008) “Adobe Air adalah sebuah *cross operation system runtime* yang dikembangkan oleh Adobe untuk mengijinkan para *developer web* untuk memanfaatkan keahlian pengembangan *web*”. *Adobe air* memungkinkan pengembang untuk paket *code* yang sama kedalam aplikasi asli untuk Windows dan Mac OS desktop serta iPhone, iPad, Kindle Fire, Nook Tablet dan perangkat lain android (<http://www.adobe.com/>).

4. *Adobe Photoshop*

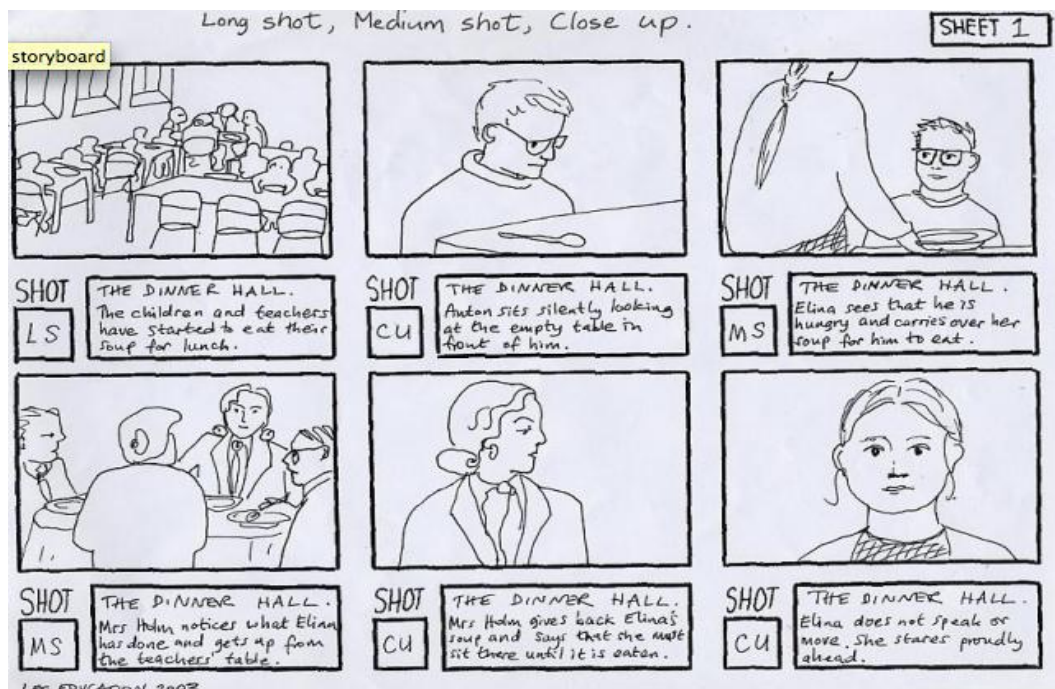
Menurut Hidayat (2011:9) *Adobe Photoshop*, atau biasa disebut *Photoshop*, adalah perangkat lunak *editor* citra buatan *Adobe Systems* yang di khususkan untuk pengeditan foto/gambar dan pembuatan efek.

5. *State Transition Diagram*

Menurut Indrajani (2011:p17) *State Transition Diagram* (STD) adalah suatu kondisi yang menunjukkan keadaan tertentu, dimana suatu sistem dapat ada dan transisi menghasilkan keadaan tertentu yang baru. Biasa digunakan dalam sistem yang *real time*.

6. Storyboard

Storyboard biasanya digunakan para perancang film, animasi, web, iklan maupun *game* untuk mendapatkan ide dari sebuah gambaran alur cerita yang akan diterapkan. Dalam perancangan *game* ini *storyboard* digunakan untuk mendapatkan gambaran visual dalam perancangan *game* yang penulis buat.



Sumber : www.mywebpresenters.com

Gambar II.1 Storyboard

Menurut Nurhasanah dan Destyany (2011:3) *Storyboard* adalah area berseri dari sebuah gambar sketsa yang digunakan sebagai alat perencanaan untuk menunjukkan secara visual bagaimana aksi dari sebuah cerita berlangsung.

7. Metode *Waterfall*

Metode *waterfall* merupakan salah satu metode yang digunakan dalam perancangan *game* ini. Menurut Rosa dan Shalahuddin (2013:28), model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini awalnya “Linier Sequential Model”. Model ini sering disebut dengan “Classic Life Cycle” atau model *waterfall*. Model *waterfall* biasanya yang paling sering digunakan dalam *software engineering*. Disebut sebagai model *waterfall* karena semua proses tahap demi tahap di lalui harus menunggu selesai tahapan sebelumnya dan berjalan berurutan.

Dalam metode *waterfall* Rosa dan Shalahuddin juga menjelaskan (2013:29) tahapan-tahapan yang dilalui antara lain:

a. Analisa Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk di dokumentasikan. Mengumpulkan kebutuhan secara lengkap kemudian dianalisa dan didefinisikan kebutuhan yang harus di penuhi oleh *software* yang akan dibangun, mengingat bahwa *software* harus dapat berinteraksi dengan elemen-elemen yang lain seperti *hardware*, *database* dan lain-lain. Tahapan ini sering juga disebut dengan *Project Defenition*.

b. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data,

arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisa kebutuhan ke representasi desain agar dapat di implementasikan menjadi program pada tahap selajutnya. Desain perangkat lunak yang di hasilkan pada tahap ini juga perlu di dokumentasikan.

c. Pembuatan Kode Program

Desain harus di terjemahkan kedalam bentuk mesin yang dapat dibaca. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat dilakukan dengan cara mekanis. Untuk dapat dimengerti oleh mesin (komputer), maka desain yang sebelumnya harus diubah kedalam bentuk yang dapat dimengeti oleh mesin, yaitu kedalam bahasa pemrograman melalui proses *coding*. Tahapan ini lah yang akan secara teknis nantinya yang akan dikerjakan oleh programmer.

d. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah di uji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diiginkan.

e. *Support*

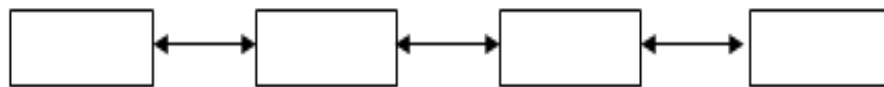
Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bias terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru.

8. Struktur Navigasi

Menurut Binanto (2010:268) mengemukakan bahwa “Setiap rencana akan dibuat desainnya dan kemudian diproduksi menjadi produk jadi yang bersifat sementara. Di samping itu tahap ini mencakup perencanaan struktur navigasi yang baik untuk antarmuka pengguna. Ada 4 struktur dasar yang digunakan yaitu linier, hirarki, non linier dan komposit”.

a. Linier (Satu Alur)

Linier merupakan struktur navigasi yang memiliki suatu rangkaian yang berurutan, dari frame atau bite informasi satu ke yang lainnya.

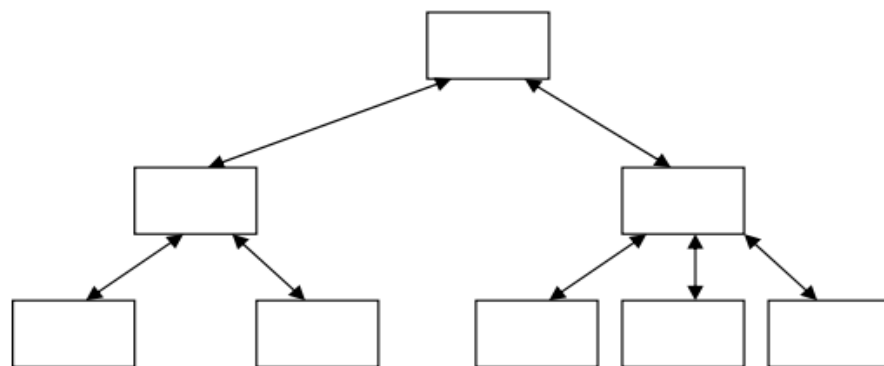


Sumber : Binanto (2010:269)

Gambar II.2 Struktur Navigasi *Linier*

b. *Hierarkis* (Bercabang)

Struktur *Hierarkis* (bercabang) karena pengguna menelusuri program di sepanjang cabang pohon dari struktur dari isi.

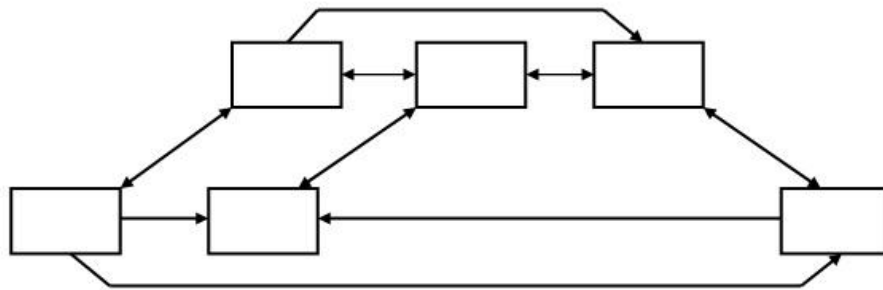


Sumber : Binanto (2010:269)

Gambar II.3 Struktur Navigasi Bercabang

c. *Non-linier* (Tidak Berurut)

Struktur navigasi *Nonlinier* (tidak berurut) adalah navigasi dengan bebas melalui isi proyek dan tidak terikat dengan rute yang telah ditetapkan sebelumnya.

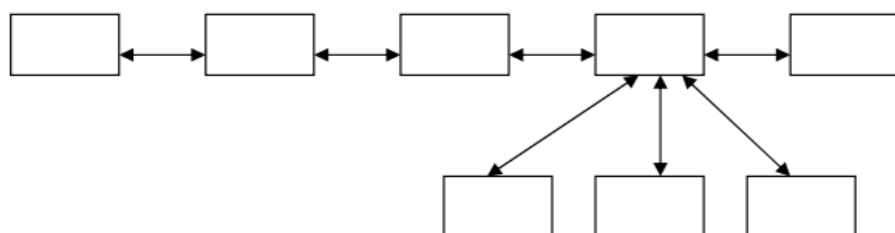


Sumber : Binanto (2010:269)

Gambar II.4 Struktur Navigasi *Nonlinier*

d. *Composit* (Campuran)

Struktur navigasi dengan bebas, tetapi terkadang dibatasi oleh informasi kritis atau pada data yang paling terorganisasi secara logis dalam suatu



percabangan.

Sumber : Binanto (2010:270)

Gambar II.5 Struktur Navigasi *Composite*

9. Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan suatu aktifitas yang dapat menjamin kualitas dari perangkat lunak dan mempresentasikan spesifikasi, desain dan pengkodean. Pada dasarnya pengujian merupakan proses eksekusi suatu program yang dilakukan dengan maksud menemukan kesalahan dari suatu program yang akan di kembangkan.

Menurut Rosa dan Shalahuddin (2013:272) “Pengujian adalah suatu set aktifitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan. Aktifitas pengujian terdiri dari satu set atau sekumpulan langkah dimana dapat menempatkan desain kasus uji yang spesifik dan metode pengujian”.

Secara umum, pola pengujian pada perangkat lunak menurut Rosa dan shalahuddin (2013:272) diantaranya adalah:

- a. Pengujian dimulai dari level komponen hingga integrasi antarkomponen menjadi sebuah sistem.
- b. Teknik pengujian berbeda-beda sesuai dengan berbagai sisi atau unit uji dalam waktu yang berbeda-beda pula bergantung pada pengujian pada bagian mana yang dibutuhkan.
- c. Pengujian dilakukan oleh pengembang perangkat lunak, dan jika untuk proyek besar, pengujian bisa dilakukan oleh tim uji yang tidak terkait dengan tim pengembang perangkat lunak (*independet test group(ITG)*).
- d. Pengujian dan penirkutuan (*debugging*) merupakan akrigitas yang berbeda, tapi penirkutuan (*debugging*) harus diakomodasi pada

berbagai strategi pengujian. Pengujian lebih fokus untuk mencari adanya kesalahan (*error*) baik dari sudut pandang orang secara umum atau dari sudut pandang pengembang tanpa harus menemukan lokasi kesalahan pada kode program. Penirkutan (*debugging*) adalah proses mencari lokasi kesalahan (*error*) pada kode program sehingga dapat segera diperbaiki oleh pembuat program (*programmer*).

Pengujian untuk validasi menurut Rosa dan Shalahuddin (2013:275) memiliki beberapa pendekatan sebagai berikut:

a. *Black-Box Testing*

Yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji yang dibuat adalah:

- 1) Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
- 2) Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.

