

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Animasi

Menurut Munir (2013:317) "Animasi adalah rangkaian gambar yang disusun berurutan atau dikenal dengan istilah *frame*, satu *frame* terdiri dari satu gambar. Jika susunan gambar tersebut ditampilkan bergantian dengan waktu tertentu maka akan terlihat bergerak".

2.1.1. Pengertian *Game*

Menurut Novaliendry (2013: 111) "*Game* berasal dari kata bahasa Inggris yang memiliki arti dasar permainan. Permainan dalam hal ini merujuk pada pengertian kelincahan intelektual (*intellectual playability*)". *Game* juga bisa diartikan sebagai arena keputusan dan aksi pemainnya, ada target-target yang ingin dicapai pemainnya.

Menurut Jasson (2009:2) "*Game* adalah suatu *system* atau program dimana satu atau lebih pemain mengambil keputusan melalui kendali pada obyek di dalam *game* untuk suatu tujuan tertentu". Sedangkan, menurut Wardhani, dkk (2013:473)

"*Game* (permainan) secara umum adalah sebuah aktivitas rekreasi dengan tujuan bersenang-senang, mengisi waktu luang, atau berolahraga ringan. Permainan biasanya dilakukan sendiri atau bersama-sama".

Dari beberapa pengertian *Game* diatas, maka dapat disimpulkan bahwa *game* adalah aktivitas yang bertujuan untuk mengasah otak pemain dan membuat pemain menjadi senang.

2.1.2. Jenis-jenis *Game*

Jenis *game* yang lebih dikenal dengan *genre*. *Genre* juga berarti format atau gaya dari sebuah *game*. Format sebuah *game* bisa murni sebuah *genre* atau bisa merupakan campuran (*hybrid*) dari beberapa *genre* lain. (Jasson, 2009:6-14).

Berikut ini akan dijelaskan macam-macam jenis *game* tersebut:

1. *Quiz Game*

Quiz Game adalah bentuk permainan atau pikiran di mana pemain (sebagai individu atau dalam tim) berusaha untuk menjawab pertanyaan dengan benar.

2. *Puzzle Game*

Game jenis ini memberikan tantangan pada pemainnya dengan cara menjatuhkan sesuatu dari sisi sebelah atas kebawah. Semakin lama akan semakin cepat dan semakin banyak objek yang jatuh. Contoh *game Tetris*, *Magic Inlay*, *Roket Mania* dan *Chip Challenge*.

3. *Shooting Game*

Shooting game adalah *game* aksi tembak-menembak merupakan tema utamanya, karena tujuan dalam *game* jenis ini hanya untuk membunuh lawan ataupun musuh dengan senjata yang telah disediakan.

4. *Adventure Game* (Petualangan)

Adventure menggabungkan unsur-unsur jenis komponen antara *game action* dan *game adventure*, biasanya menampilkan rintangan yang berjangka panjang yang harus diatasi menggunakan alat atau item sebagai alat bantu dalam mengatasi rintangan, serta rintangan yang lebih kecil yang hampir terus-menerus ada.

5. *Slide Scrolling Game*

Pada jenis *game* ini karakter dapat bergerak ke samping diikuti dengan gerakan *background*. Contoh *game* tipe 2D seperti Super Mario, *Metal Slug*, dan sebagainya.

6. *Fighting Game*

Game ini biasanya mempunyai ciri pertarungan satu lawan satu antara dua karakter, salah satu dari karakter di kendalikan oleh *computer*.

7. *Sport Game*

Merupakan jenis *game* yang memiliki unsur olahraga di dalamnya. Banyak sekali olahraga di dunia nyata yang dimasukkan ke dalam *game*, sehingga tidak hanya dapat berolahraga seperti biasa, namun juga dapat dilakukan dengan dalam *video game*.

8. *Racing Game*

Game jenis ini memberikan permainan lomba kecepatan dari kendaraan yang dimainkan oleh pemain. Contoh *game Driver, Test Drive, dan Ridge Racer*.

2.1.4. Elemen-Elemen Dasar *Game*

Fungsi tampilan dari berbagai *game* ada yang serupa atau mirip satu sama lain, baik fungsi maupun bentuknya. Walaupun *genre* satu sama lain berbeda, namun elemen itu tetap memiliki fungsi dan maksud yang sama. (Jasson, 2009:20-29).

Berikut adalah keterangan berbagai elemen-elemen yang ada dalam sebuah *game* yang terdiri atas:

1. *Title* (Judul)

Judul adalah elemen penting dalam pengembangan sebuah *game*. Membuat judul sebuah *game* tidaklah mudah. Judul harus singkat namun menyiratkan isi dari maksud *game* tersebut.

2. *Title Screen* (Layar Judul)

Judul memang merupakan hal yang penting, namun untuk membuat judul lebih menarik adalah tampilan grafis yang sesuai dengan judul *game* tersebut. Pada *game* masa kini, sebelum *title screen* muncul terkadang sudah ada beberapa animasi pembuka yang muncul.

3. *Cutscene/Intro* (Pengenalan Cerita)

Tujuan dari *cutscene/intro* adalah untuk membuka awal pengenalan sebelum pemain memulai *game*, suatu penting di dalam alur cerita *game* maupun suatu transisi antara tingkat satu dengan tingkat selanjutnya. Adanya *cutscene* atau *intro* pada *game* ini, membuat pemain secara perlahan akan masuk ke alur permainan atau kepada bagian selanjutnya dari *game*.

4. *User Interface* (Antarmuka)

Berbeda dengan *control panel* yang hanya aktif ketika dibutuhkan. Sementara *user interface* tetap aktif selama pemain memainkan *game* dan tidak hanya tampilan pada layar, *keyboard* dan *mouse* yang digunakan adalah *user interface*. Untuk layar maka *user interface* adalah simbol atau tampilan khusus yang membantu pemain memainkan *game*. Contoh pada *game* jenis *First Person Shooter* (FPS) *user interface*-nya adalah jumlah peluru dan nyawa yang ada pada tampilan layar.

5. *Help* (Bantuan)

Tidak semua *game* dapat dimengerti dengan cepat oleh pemain *game* tersebut. Untuk itulah elemen *help* ini menjadi salah elemen yang cukup penting. Di dalam *help* ini semua instruksi yang bisa membantu pemakai ditampilkan dengan cepat dan sederhana sehingga mudah dipahami dan bisa langsung digunakan.

6. *Mouse Pointer* (Ikon Penunjuk *Mouse*)

Bentuk *pointer mouse* bisa bervariasi sesuai dengan tema dari *game* yang dibuat. Terkadang bentuk *pointer* bisa berubah jika menyentuh area tertentu di dalam *game* (disebut *hotspot*).

7. *Music and Sound* (Musik dan Efek Suara)

Semua jenis *game* saat ini sewajarnya dilengkapi musik dan suara yang cocok dengan tema dari *game* yang akan dikembangkan. Namun jika *game* yang dikembangkan tidak memiliki suara dan musik akan terkesan tidak *profesional* dan kurang bermutu.

8. *Art* (Seni)

Pada dasarnya *game* yang baik dihasilkan dari gambar yang memang memiliki seni yang baik pula. Gambar juga memegang peranan penting dalam elemen *game* yang akan dikembangkan. Itu sebabnya kenapa *game* masa kini khususnya di bidang *art* (seni) dipegang oleh beberapa orang yang memiliki keahlian tersendiri.

9. *Storyline* (Cerita)

Alur cerita tidak digunakan pada semua *game*, hanya jenis *game* tertentu yang menggunakan alur cerita misalnya *adventure game*, *RPG game*, *Real Time Strategi (RTS) game* dan *Action game*.

10. Level (Tingkatan)

Hampir setiap *game* memiliki beberapa level untuk dimainkan. Maksudnya supaya pada tahap awal pemain bisa menguasai permainan dengan baik. Pada tingkat selanjutnya pemain tidak akan merasa bosan karena tantangan akan meningkat.

11. Demo

Pada dasarnya *game* muncul dengan mode demo terlebih dahulu sehingga bisa dimainkan sebelum *software* aslinya muncul di pasaran. Memang tidak semua *game* menyediakan demo *game*, tetapi keuntungan promosi bisa didapatkan dengan menyediakan demo *game*.

12. *Exit Screen* (Layar Keluar)

Layar keluar ini merupakan elemen penting dalam pengembangan *game*. *Game* yang baik biasanya memiliki *exit screen* pada *game* yang dikembangkan.

13. *Copyright* (Hak Cipta)

Hal paling penting dalam pengembangan sebuah perangkat lunak *game* ini adalah hak cipta yang diberikan oleh pembuat *game*. Hal ini untuk mencegah masalah yang berkaitan dengan hak cipta dimasa depan.

14. *Setup* (Pemasangan atau Instal)

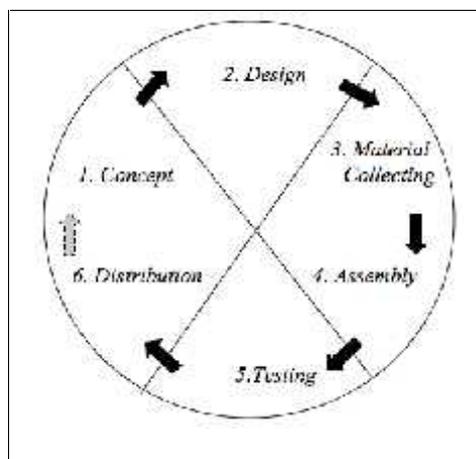
Sudah menjadi kewajiban bagi pembuat perangkat lunak untuk membuat *game* dalam bentuk *setup*. Program *setup* pada *game* akan dikompilasi terlebih dahulu menjadi paket yang padat.

2.2. Teori Pendukung

2.2.1. Metodologi Pengembangan Multimedia

Menurut Munir (2013:97) “Metodologi dalam pengembangan *software* selalu dikaitkan dengan kerangka kerja atau *framework* karena menggunakan pendekatan sistem informasi”. Tujuan mewujudkan kerangka kerja adalah untuk membimbing peneliti dalam mengembangkan *software* tersebut.

Menurut Sutopo dalam Munir (2013:104) “Metodologi pengembangan multimedia terdiri dari 6 tahapan, yaitu *concept*, *design*, *material collecting*, *assembly*, *testing* dan *distribution*”. Seperti gambar berikut ini:



Sumber: Sutopo dalam Munir (2013:104)

Gambar II. 1.

Metodologi Pengembangan Multimedia

1. *Concept*

Tahapan *concept* (pengonsepan) adalah tahapan untuk menentukan tujuan dan siapa pengguna program (identifikasi audiens).

2. *Design*

Design (perancangan) adalah tahap pembuatan spesifikasi mengenai arsitektur program, gaya, tampilan dan kebutuhan materil/bahan untuk program.

3. *Material Collecting*

Material collecting adalah tahap pengumpulan bahan yang sesuai dengan kebutuhan yang dikerjakan.

4. *Assembly*

Tahap *assembly* adalah tahap pembuatan semua objek atau bahan multimedia.

5. *Testing*

Tahap *testing* (pengujian) dilakukan setelah menyelesaikan tahap pembuatan (*assembly*) dengan menjalankan aplikasi/program dan melihatnya apakah ada kesalahan atau tidak.

6. *Distribution*

Pada tahap ini, aplikasi akan disimpan dalam suatu media penyimpanan.

2.2.2. *HTML5*

Menurut Prasetya (2013:13-14) “HTML (*HyperText Markup Language*) telah mencapai versi 5 (atau dikenal *HTML5*) yang merupakan standar *web* masa depan”. *HTML5* merupakan revisi baru yang merujuk pada spesifikasi HTML

4.01. Proyek ini digagas oleh *Web Hypertext Application Technology Working Group* (WHATWG) karena melihat HTML 4.01 sudah cukup lama tidak diperbaharui oleh Konsorsium *word wide web* (*World Wide Web Consortium/W3C*). Saat itu W3C dan WHATWG berinisiatif untuk berkerja sama mengembangkan *HTML5*.

HTML5 sudah menyediakan fitur *built-in* (tanpa tambahan *plugin* atau API) untuk mengimplementasikan animasi, *canvas*, *event-handler*, validasi *form*, *drag & drop*, audio & video, *geolocation*, dan komunikasi *socket*.

2.2.3. Android

Menurut Safaat (2014:1) “*Android* adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *Linux* yang mencakup sistem operasi, *middleware* dan aplikasi”. Pada masa saat ini kebanyakan vendor-vendor *smartphone* sudah memproduksi *smartphone* berbasis *Android*, vendor-vendor itu antara lain *HTC*, *Motorola*, *Samsung*, *LG*, *HKC*, *Huawei*, *Archos*, *Webstation Camangi*, *Dell*, *Nexus*, *SciPhone*, *WayteQ*, *Sony Ericsson*, *Acer*, *Philips*, *T-Mobile*, *Nexian*, *IMO*, *Asus* dan masih banyak lagi vendor *smartphone* di dunia yang memproduksi *Android*.

Menurut Safaat (2014:3) “*Android* dipuji sebagai *platform mobile* pertama yang lengkap, Terbuka, dan Bebas”. Berikut ini beberapa kelebihan dari sistem operasi *Android* adalah sebagai berikut”,

Dari beberapa pengertian *android* diatas, maka dapat disimpulkan bahwa *Android* adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi ini yang dirilis oleh *Google*.

1. Lengkap (*Complete Platform*)

Sistem operasi *Android* merupakan sistem operasi yang menyediakan banyak *tools* dalam mengembangkan aplikasi yang dapat digunakan kembali oleh *developer* untuk mengembangkan aplikasinya.

2. Terbuka (*Open Source Platform*)

Platform Android yang bersifat *open source* (terbuka), hal ini yang membuat sistem operasi *Android* mudah dikembangkan oleh *developer*.

3. Bebas (*Free Platform*)

Developer tidak perlu membayar royalti untuk memperoleh *lisence* sehingga dapat dengan bebas mengembangkan, mendistribusikan dan memperdagangkan sistem operasi *Android*.



Sumber: Safaat (2014:1)

Gambar II. 2.

Logo Android

2.2.4. PhoneGap

Menurut Prasetya (2013:117) “*PhoneGap* adalah sebuah kerangka kerja (*framework*) *open source* yang digunakan untuk membangun aplikasi-aplikasi *mobile* lintas *platform* dengan memanfaatkan teknologi-teknologi *web* standart, mencakup HTML, CSS, dan *JavaScript*. Secara khusus, *PhoneGap* didasarkan pada spesifikasi HTML5 yang mampu menghasilkan aplikasi-aplikasi *mobile* modern.

2.2.5. Construct 2

Menurut Permana (2015:114) “*Construct 2* adalah sebuah *tool* berbasis *HTML5* untuk menciptakan sebuah permainan”. Dengan *tool Construct 2* memungkinkan siapa saja membuat *game* tanpa harus memiliki pengalaman pemrograman. Dikembangkan oleh *Scirra Ltd*, hal ini ditujukan terutama untuk para *non-programmer* yang ingin menciptakan sebuah *game* secara *drag and drop* menggunakan editor visual dan berbasis sistem logika perilaku. *Construct 2* ini didesain khusus untuk pengembang *game* yang *non-programmer* dimana hanya dibutuhkan logika dan teori *game* yang diciptakan. Untuk menciptakan sebuah *game*, pengembang hanya melakukan *drag and drop* menggunakan editor visual.

Dari beberapa pengertian *Construct 2* diatas, maka dapat disimpulkan bahwa *construct 2* adalah sebuah *tool* berbasis *HTML5* untuk menciptakan sebuah permainan hal ini memungkinkan orang untuk *Construct game* tanpa *coding* yang diperlukan



Sumber: Permana (2015:114)

Gambar II. 4.

Tampilan *Construct 2*

2.2.6. *Storyboard*

Menurut Binanto (2010:255) “*Storyboard* merupakan pengorganisasi grafik, contohnya adalah sederetan ilustrasi atau gambar yang ditampilkan berurutan untuk keperluan visualisasi awal dari suatu *file*, animasi, atau urutan media interaktif, termasuk interaktivitas di *web*”.

Sedangkan menurut Nurhasanah, dkk (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”.

Dari beberapa pengertian *Storyboard* diatas, maka dapat disimpulkan bahwa *storyboard* adalah terjemahan berupa gambar sketsa dari naskah yang sudah dibuat, dan digunakan dalam proses perancangan sebuah produk dan untuk menunjukkan secara *visual* bagaimana aksi dari sebuah cerita berlangsung.

2.2.7. Pengujian *Black Box*

Menurut Rizky (2011:264) “*Black-Box Testing* adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya”.

Sedangkan menurut Rosa dan Shalahuddin (2013:275) “*Black-box Testing* (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan”

Dari beberapa pengertian *Black Box* diatas, maka dapat disimpulkan bahwa *Black Box* adalah menguji perangkat lunak dari segi spesifikasi fungsional untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan, sangat penting untuk dilakukan agar tidak terjadi kesalahan alur program yang telah dibuat.