

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam penulisan tugas akhir ini penulis melakukan pencarian informasi dari penelitian-penelitian yang terdahulu untuk melihat kelebihan dan kekurangan dari penelitian-penelitian yang sudah dilakukan sebelumnya. Pencarian informasi ini digunakan sebagai bahan tinjauan pustaka dalam penulisan tugas akhir ini. Tinjauan pustaka ini membahas tentang perkembangan *Bord Game* Tradisional dan pemilihan *game engine* yang digunakan sebagai dasar penelitian.

2.1.1 Perkembangan *Board Game* Tradisional

Dengan perkembangan teknologi yang sangat pesat seperti saat ini, *board game* mulai beralih dari bentuk fisik menjadi bentuk digital. Banyak sekali *board game* versi digital saat ini. Seperti *board game* Ludo dengan versi digitalnya yaitu *Ludo King*, *Ludo Online*, *Ludo Master*. *Board game* catur dengan *catur*, *chess*, dan *Chess master*. Serta masih banyak *board game* yang dijadikan digital.

Perkembangan *board game* tidak hanya sebatas beralih ke bentuk digital saja, Line Corporation bersama Modo Marble melakukan perubahan aturan serta komponen pada permainan *monopoly* tradisional. Line Corporation dan Modo Marble mengubah permainan *monopoly* tradisional menjadi permainan *monopoly* yang lebih strategis, dengan nama *LINE Let's Get Rich* [9]. Permainan ini mampu menarik perhatian para pengguna android. Berdasarkan hasil pencarian penulis pada *Play Store*, pengunduh *LINE Let's Get Rich* mencapai 50 juta lebih dengan nilai rating 4,3 [10].

Tidak hanya Line Corporation, JOYCITY Corp juga telah membuat *monopoly* menjadi sebuah permainan yang lebih strategis. Berbeda dengan *Line Let's Get Rich*, *Game of Dice* [11] buatan JOYCITY Corp merubah elemen kartu dan merubah *bord*-nya menjadi lebih kompleks agar menambah nuansa strategi di dalam permainannya. Dari *Game of Dice*, JOYCITY Corp. mampu menarik

perhatian para pengguna *Play Store*. Berdasarkan hasil penelusuran penulis pada *Play Store*, *Game of Dice* memperoleh rating dari penggunanya hingga 4,3 [12].

Sama seperti yang dilakukan perusahaan Line Corporation dan JOYCITY Corp., Muhammad Ilham Nur Isra' [8] dalam penelitiannya melakukan pengembangan permainan ular tangga tradisional ke permainan ular tangga yang lebih strategis dengan cara memperbaharui aturan dan menambahkan unsur-unsur strategi pada permainan ular tangga yang lama, sehingga mengurangi faktor keberuntungan pada permainan ular tangga yang lama. Penelitian ini menghasilkan GDD (*Game Design Document*) dan SRS (*Software Requirement Specifications*) dari permainan ular tangga yang lebih strategis.

Disini penulis mengembangkan penelitian sebelumnya yang dilakukan oleh Muhammad Ilham Nur Isra' dengan cara GDD dan SRS yang sudah dibuat dijadikan permainan digital.

2.1.2 Game Engine yang digunakan

Menurut Bryan [13], *game engine* adalah *software* yang dirancang untuk pembuatan dan pengembangan suatu *video game*. *Game engine* memberikan kemudahan dalam menciptakan konsep sebuah permainan yang akan dibuat. Mulai dari sistem *rendering*, *physics*, *scripting*, *A.I*, dan bahkan sistem *networking* dalam permainan. *Game engine* dapat dikatakan sebagai *arsitekturnya* para *developer game* dalam merancang sebuah permainan. Ada banyak sekali *game engine* diantaranya adalah Kodu, Unity 3D, Construct 2, dan lain-lain.

Bagi *developer* permainan, *game engine* memiliki peran penting dalam membuat sebuah permainan, karena *game engine* memiliki fungsionalitas masing-masing. Setiap *game engine* memiliki kompleksitasnya berbeda-beda. Penggunaan *game engine* yang tepat akan mempermudah dan mempercepat proses produksi suatu permainan [14].

Disisi lain, ada 2 jenis permainan digital, yaitu permainan 2D dan permainan 3D. Ada *game engine* yang dikhususkan untuk membuat permainan 2D atau permainan 3D. Tetapi, ada juga *game engine* yang bisa membuat

permainan 2D maupun permainan 3D. Dalam permainan *board game* digital, menurut William Herkewitz [15] ada 11 dari 15 *top board game* digital yang menggunakan animasi 2D dan 4 sisanya menggunakan animasi 3D.

Dari penelitian yang dilakukan oleh Sudarmilah, Ferdiana, Nugroho, Susanto dan Ramadhani [16] yang membandingkan ketiga *game engine* yaitu Kodu, Unity 3D, dan Construct 2 dengan tema yang sama menyimpulkan bahwa kodu dan unity 3D lebih cocok untuk permainan yang menekankan pada 3D. Meskipun Kodu lebih cepat untuk proses prototyping, Unity 3D memiliki kelebihan dalam menambahkan karakter dan properti. Sedangkan untuk Construct 2 lebih cocok untuk membuat permainan 2D dengan proses *scripting* yang lebih sederhana hanya dengan "*drag and drop*" dan menggunakan logika sederhana di Lembar *EvenSheet*.

Oleh karena itu permainan yang dibuat oleh penulis berbasis pada 2D dan menggunakan *game engine* construct 2 untuk membuat permainan ular tangga bergenre strategi ini.

2.2 Landasan Teori

Dalam penulisan tugas akhir ini, penulis menggunakan teori-teori yang berasal dari jurnal maupun buku. Teori-teori tersebut digunakan penulis sebagai landasan dalam penulisan skripsi ini.

2.2.1 Permainan

Menurut KBBI kata “permainan” berasal dari “main” yang artinya melakukan perbuatan untuk bersenang-senang. permainan adalah suatu kegiatan yang sangat menyenangkan bagi semua orang. Bahkan bukan hanya untuk anak-anak saja melainkan juga untuk remaja, pemuda, bahkan orang tua sekalipun [17].

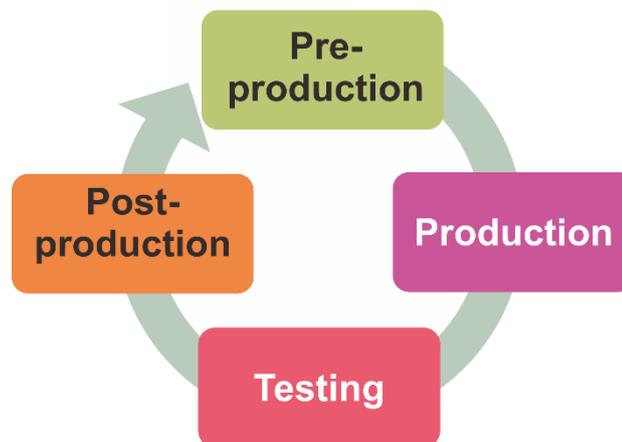
2.2.2 Permainan Strategi

Menurut Tracy Fullerton [18], permainan strategi merupakan sebuah permainan yang berfokus pada penggunaan taktik dan perencanaan dalam

memainkannya, termasuk juga dalam mengelola unit serta sumber daya yang ada pada permainan tersebut. Ada dua tipe *game* strategi, *RTS* atau *Real-Time Strategy* dan *TBS* atau *Turn-Based Strategy*. *RTS* adalah tipe permainan strategi dimana player akan beradu strategi dalam satu waktu, sedangkan *TBS* adalah tipe *game* strategi yang dimana *player* akan saling bergantian dalam menentukan masing-masing strateginya, tipe *game* strategi ini biasanya digunakan dalam sebuah permainan *board game*. Contoh untuk permainan *RTS* adalah *Dota 2*, *Clash Royale* dan *Age of Empire* dan contoh untuk permainan *TBS* adalah *Line Let's Get Rich*, *Yu-Gi-oh! Duel Link* dan *Ludo*.

2.2.3 Tahap-tahap Pengembangan Game

Game Development Life Cycle atau *GDLC* adalah panduan yang digunakan untuk menentukan langkah dan kriteria yang perlu dipertimbangkan dalam membangun permainan yang berkualitas baik [19]. Menurut Heather Maxwell Chandler [20], terdapat empat tahap dalam melakukan pengembangan permainan yaitu *pre-production*, *production*, *testing*, dan *post-production*. Seperti yang terlihat pada gambar 2.1.



Gambar 2.1 Model GDLC Heather Chandler [20]

a. *Pre-Production*

Tahap ini adalah tahap untuk mendefinisikan dan melakukan perencanaan terhadap permainan yang akan dibangun. Hal ini terkait dengan

konsep dasar permainan, identifikasi pengguna, tujuan permainan dan menganalisis kebutuhan dari permainan yang akan dirancang.

b. *Production*

Tahap ini merupakan proses inti dari perancangan permainan. Tahap ini merupakan tahap pengimplementasian dari rancangan yang telah dibangun, tahap ini berkaitan dengan penciptaan teknis-teknis dari permainan yang akan dibangun dan yang berkaitan dengan keindahan.

c. *Testing*

Tahap ini adalah tahanan untuk melakukan pengujian. Pada tahap ini permainan yang telah dibangun akan dijalankan dan diperiksa apakah terdapat *bug* atau tidak.

d. *Post Production*

Pada tahap ini, permainan yang telah melakukan pengujian akan didokumentasikan dan dilakukan evaluasi. Hasil evaluasi ini dapat digunakan sebagai masukan untuk tahap *Pre-Production* pada produk selanjutnya.

2.2.4 Testing Program

Testing adalah proses menganalisa suatu *software* untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan (yaitu *defects/error/bugs*) dan mengevaluasi fitur-fitur dari *software* itu sendiri (standar ANSI/IEEE1059).

2.2.5 Black box Testing

Menurut Mustaqbal, Firdaus, & Rahmadi [21], *Black box Testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari *software*. *Tester* dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program. Pengujian berusaha menemukan kesalahan dengan kategori sebagai berikut:

- a. Fungsi–fungsi yang tidak benar atau tidak ada.
- b. Kesalahan antarmuka (*interface error*).
- c. Kesalahan struktur data atau akses basis data.
- d. Kesalahan kinerja (*performance error*).

e. Kesalahan inisialisasi dan terminasi.

2.2.6 Construct 2

Menurut perusahaan Scirra Ltd. selaku pembuat *game engine* construct 2 [22], Construct 2 adalah *game engine* berbasis HTML 5 yang dirancang khusus untuk permainan 2D. Construct 2 dikembangkan untuk dapat membuat aplikasi dengan metode *visual programming*, yaitu *drag & drop* dengan kebutuhan *coding* yang minimal. Pada umumnya, Construct 2 dapat digunakan di Windows, namun *game engine* ini juga dapat dijalankan di Mac, Linux, atau iPad. Construct 2 mendukung berbagai platform untuk *export*-nya seperti PC, Android, iPhone, Mac, dan browser. Dalam construct 2, perintah yang digunakan pada permainan diatur dalam *EvenSheet*. Didalam *EvenSheet* terdiri dari *variable*, *condition*, dan *Action*. *Variable* adalah tempat menyimpan sebuah *value*. *Condition* adalah kondisi/keadaan yang diperlukan untuk menjalankan *action*. Sedangkan *action* adalah perintah yang harus dijalankan/dilakukan. Secara singkat hubungan *condition* dan *action* adalah jika *condition* terpenuhi, maka lakukan *action*.

Ada beberapa *Condition Commands* dasar yang sering digunakan dalam pembuatan permainan menggunakan Construct 2 yaitu:

- Compare two values* : kondisi dimana 2 *value/variable* bernilai “true” atau benar.
- Compare Variables* : sangat mirip dengan *Compare two values*, tetapi *compare variable* hanya bisa membandingkan suatu *variable* dengan *value*.
- clicked* : kondisi ketika objek di-*click*.
- On start of layout* : kondisi ketika pertama kali halaman ditampilkan.
- Else* : kondisi ketika berlawanan dengan kondisi yang sudah ditentukan.
- Is Boolean* : kondisi ketika *variable* boolean bernilai *true*.

- Repeat* : kondisi untuk *looping*/perulangan yang ditentukan.
- Every X second* : kondisi ketika setiap x (x bernilai 0.1,0.2,0.3, dst.) detik maka akan menjalankan *action* yang ditentukan.
- Is Visible* : kondisi ketika objek terlihat di layar/halaman.
- Is Empty* : kondisi ketika bagian yang ditentukan pada objek *board* kosong. (hanya bisa digunakan pada *objek board*).
- Is Overlapping* : kondisi ketika objek bertumpukan dengan objek lain.
- Is On Layer* : kondisi ketika objek berada di *layer* tertentu.
- On Collision with* : kondisi ketika *objek* bertabrakan dengan objek lain.

Ada beberapa *Action commands* dasar yang sering digunakan dalam pembuatan permainan menggunakan Construct 2 yaitu:

- Wait* : *action* “Menunggu”, dimana *action* selanjutnya dijalankan setelah beberapa detik yang sudah ditentukan.
- Add value* : menambahkan isi *variable*.
- Substract From* : mengurangi isi *variable* (berlaku untuk *variable* integer).
- set value* : menghapus isi *variable* kemudian diganti dengan *value* yang baru.
- Set animation frame* : mengubah tampilan *frame* ke *frame* yang lain dari satu objek.
- Set animation speed* : mengubah kecepatan pindah *frame* ke *frame* yang lain dari suatu objek. Misalkan Set animation speed sama dengan 5 pada objek A, maka dalam 1 detik sistem akan menampilkan 5 *frame* dari objek A.

<i>Play</i>	: untuk menjalankan objek suara. (berlaku untuk objek suara).
<i>Set text</i>	: mengganti teks yang lama dengan teks baru yang ditentukan (berlaku untuk objek teks).
<i>Go to layout</i>	: untuk berpindah ke halaman yang ditentukan.
<i>Set boolean</i>	: mengubah isi <i>value variable boolean</i> .
<i>Delete</i>	: menghapus <i>index</i> pada objek <i>array</i> (berlaku pada objek <i>array</i>).
<i>Create chess</i>	: membuat objek didalam objek <i>board</i> (berlaku objek <i>board</i>).
<i>GridMove</i>	: memindahkan objek dari suatu tempat ke tempat yang ditentukan pada objek <i>board</i> (berlaku untuk objek yang berada didalam objek <i>board</i>).
<i>Set visible</i>	: menghilangkan atau menampilkan tampilan dari objek.
<i>Create object</i>	: membuat objek baru pada halaman yang ditampilkan.
<i>Destroy</i>	: menghapus/menghilangkan objek.

2.2.7 CorelDraw X7

CorelDraw adalah adalah perangkat lunak editor grafik vektor(gambar yang dibuat dari unsur garis dan kurva yang membentuk suatu obyek atau gambar) berbasis Sistem Operasi windows [23]. Seri terbarunya adalah CorelDRAW X8 yang dirilis pada 15 Maret 2016. Sedangkan CorelDRAW X7 dirilis pada 27 Maret 2014, penggunaan CorelDRAW X7 bisa dijalankan dengan sistem operasi windows dengan minimal RAM 1 GB dan resolusi layar minimal 1280 x 768 pixe.