

BAB II

LANDASAN TEORI

2.1 Game

Game berasal dari kata bahasa Inggris yang berarti permainan. Permainan adalah suatu kegiatan yang melibatkan *user* secara langsung dalam permainan dimana *user* tersebut harus memecahkan masalah pada permainan tersebut. Tujuan utama dari permainan adalah menghibur *user*. *Game* secara naluri merupakan bagian dari kehidupan manusia. Makna sekilas dari *game* memberikan pengertian bahwa *game* merupakan suatu aktifitas yang tidak dilakukan dengan sungguh-sungguh. Untuk mengetahui apa yang sesungguhnya disebut dengan *game*, maka paling tidak dapat memahami dari adanya sejumlah pengertian *game* yang biasa dialami dalam kehidupan (Zamroni, et. al. 2013:489)

2.1.1 Klasifikasi Game

Menurut Kurniawan (2015:23) ada tiga macam klasifikasi *game* diantaranya adalah : a) *Game as Game*, *game* yang dimaksud adalah *game* untuk kesenangan atau *fun*, b) *Game as Media*, tujuan utama dari *game* ini adalah untuk menyampaikan pesan tertentu, menyampaikan pesan dari pembuat *game* tersebut, c) *Game Beyond Game*, bisa disebut juga dengan istilah *gamification*. *Gamification* adalah penerapan konsep atau cara berpikir *game design* ke dalam lingkup *non-game*

2.1.2 GameEngine

GameEngine adalah sistem perangkat lunak yang dirancang untuk menciptakan dan mengembangkan video game. Ada banyak mesin permainan yang dirancang untuk bekerja pada konsol permainan video dan sistem operasi desktop seperti Microsoft Windows, Linux, dan Mac OS X. Fungsionalitas inti disediakan oleh mesin permainan mencakup mesin render (“renderer”) untuk 2D atau 3D grafis, mesin fisika atau tabrakan (tumbukan), suara, script, animasi, kecerdasan buatan, jaringan, streaming, manajemen memori, threading, dukungan lokalisasi, dan adegan grafik (Saputra, 2013:16-17).

2.1.3 Tahapan Pembuatan Game

Pengembangan sebuah game mengikuti sebuah metode yang sudah menjadi standar dalam pengembangannya. Metode pengembangan game merupakan sebuah cara yang digunakan dalam membantu proses perancangan, pembuatan program dan pengujian game agar memenuhi aspek yang diinginkan. Metode perancangan game diantaranya adalah sebagai berikut : a) *Concept*, merupakan tahap awal yang digunakan oleh pengembangan yang beris mengenai tujuan dalam pembuatan atau pengembangan, menentukan siapa yang akan memainkan game, dan manfaat dari game. b) *Design*, yang beris mengenai alur cerita, kode program, spesifikasi perangkat keras, tampilan dan kebutuhan material atau bahan dalam pengembangan aplikasi. Pada tahap ini juga dikenal suatu proses pembuatan *storyboard*. *Storyboard* merupakan gambar skenario yang

dibuat secara bertahap yakni setiap *scene* dalam *game*. *Storyboard* ini yang nantinya akan menerangkan susunan dari materi *game* yang dibuat. c) *Material Collecting*, merupakan tahap ketiga yang dilakukan dalam pengembangan *game*. Pada tahap ini dilakukan proses pengumpulan bahan-bahan yang telah disebutkan pada tahap sebelumnya. d) *Assembly*, merupakan tahap keempat yang dilakukan dalam proses pengembangan *game*. Pada tahap ini dilakukan proses pembuatan semua obyek yang sudah didefinisikan pada tahap *design*. e) *Testing*, tahap ini merupakan pengujian hasil dari proses *assembly*. f) *Distribution*, pada tahap ini dilakukan proses penyebarluasan *game* yang telah diuji. Proses penyebarluasan dapat dilakukan dengan menggunakan sistem *standalone* yaitu dengan menggunakan *Compact Disc* secara online (Martono dan Teguh, 2015:25-26).

2.2 Game Casual

Menurut Adams (2009), *Game casual* adalah *game* yang lebih menekankan dipergunakan yang mudah untuk dimainkan dan biasanya tidak membutuhkan perhatian yang serius dari segi waktu dan keseringan bermain, selain itu *game casual* memiliki ciri-ciri grafik yang penuh warna dan tanpa kekerasan. (Arifin et.al, 2013:58)

2.3 Corona SDK

Corona SDK adalah SDK yang dibuat oleh Walter Luh yang merupakan founder dari Corona Labs Inc. Programmer software dapat menggunakan corona

SDK untuk membangun sebuah aplikasi *mobile* untuk perangkat Android, iPhone, dan iPad. Walter Luh memulai CoronaLabs setelah keluar dari Adobe pada tahun 2007. Kemudian pada Juni 2009, Luh mengeluarkan Corona SDK pertama kali versi beta. Corona memakai bahasa pemrograman Lua yang sangat mudah sekali dipelajari bahkan untuk pemula dalam hal pemrograman. CoronaLabs mempunyai 3 buah produk yakni Corona SDK, Corona Enterprise, dan CoronaCloud. (Fauzi et.al, 2014:116).

2.3.1 Platform

Platform merupakan kombinasi antara sebuah arsitektur perangkat keras dengan sebuah kerangka kerja perangkat lunak. Kombinasi tersebut memungkinkan sebuah perangkat lunak dapat berjalan. Adapun jenis-jenis platform diantaranya adalah: a) Arcade Games atau yang sering disebut ding-dong di Indonesia, biasanya berada di daerah/tempat khusus dan memiliki box atau mesin yang memang khusus di design untuk jenis video games tertentu dan tidak jarang bahkan memiliki fitur yang dapat membuat pemainnya lebih merasa “masuk” dan “menikmati”, seperti pistol, kursi khusus, sensor gerakan, sensor injakkan dan stir mobil. b) PC Games, yaitu video game yang dimainkan menggunakan Personal Computers. c) Console games, yaitu video games yang dimainkan menggunakan console tertentu, seperti Playstation 2, Playstation 3, XBOX 360, dan NintendoWii. d) Handheld games, yaitu yang dimainkan di console khusus video game yang dapat dibawa kemana-mana, contoh Nintendo DS dan Sony PSP. e) Mobile games, yaitu yang dapat dimainkan atau khusus untuk mobilephone atau PDA. (Martono dan Teguh, 2015:24)

2.4 Bahasa Pemrograman Lua

Lua berasal dari bahasa Portugis yang berarti Bulan. Lua merupakan bahasa pemrograman ringkas dan dirancang sebagai bahasa pemrograman dinamis berbasis skrip dengan semantik yang dapat dikembangkan atau ditambahkan. Sebagai bahasa skrip, Lua memiliki API dalam bahasa C yang relatif lebih sederhana dibandingkan bahasa skrip lainnya. Lua ditulis pertama kali oleh Roberto Ierusalimsky, LuizHenriquedeFigueiredo, dan WaldemarCeles yang merupakan anggota ComputerGraphics Technology Group (Tecgraf) pada universitas PontificalCatholic, Rio deJaneiro, Brazil, pada tahun 1993. Secara historis, bahasa pemrograman Lua berawal dari bahasa pemrograman yang digunakan untuk pendeskripsian serta entri data, serta pengkonfigurasiannya suatu sistem yang masing-masing dikembangkan secara terpisah oleh tecgraf dari tahun 1992 hingga tahun 1993. Secara umum Lua digambarkan sebagai bahasa pemrograman multi paradigma yang menyediakan seperangkat kecil atas fitur-fitur umum yang dapat dikembangkan lebih lanjut untuk memenuhi ragam kebutuhan yang berbeda-beda, karenanya Lua tidak menyediakan fitur yang lengkap dan kompleks yang hanya berfokus pada satu paradigma pemrograman. Secara umum, Lua berusaha untuk menyediakan fitur-fitur abstrak/meta yang lebih fleksibel dan dapat dikembangkan jika dibutuhkan dibandingkan menyediakan seperangkat pustaka yang lengkap untuk memenuhi satu kebutuhan tertentu. Hal tersebut menjadikan Lua sebagai bahasa pemrograman yang ringkas

dan dapat secara mudah diadaptasikan untuk memenuhi beragam jenis kebutuhan. (Setiasih et.al, 2014:115-116)

2.4.1 *ObjectOrientedProgramming* Lua

Lua menyediakan fasilitas *metatable* yang memungkinkan dapat diimplementasikan relatif cukup mudah. *Table* dalam Lua memiliki *object* yang memiliki lebih dari satu arti. Pada Lua *table* memiliki identitas independent dan nilai-nilai pada *table*, sedangkan *object* dapat memiliki nilai-nilai yang berbeda-beda dengan *object* yang sama. Dalam Lua *object* memiliki operasi sendiri, begitu juga dengan *table* yang dapat memiliki operasi. Seperti contoh pembuatan sebuah *function* baru dan *object* “Account” dibawah ini :

```
Account = {balance = 0}
function Account.withdraw (v)
Account.balance = Account.balance - v
end

Account.withdraw(100.00)
```

Baris terakhir berfungsi untuk memanggil *object*. (Ierusalimschy, 2004:16)

2.4.2 Contoh Program LUA

Berikut adalah contoh klasik sebuah program “helloworld” yang dapat ditulis dengan pemrograman Lua :

```
print("Hello World!")
```

Gambar 2.1 Contoh Program Lua

2.4.3 Contoh Game Menggunakan LUA

- EmpireOfMath

Empireofmath merupakan game edukasi untuk anak Sekolah Dasar khususnya kelas empat. Game ini menginputkan materi-materi kelas empat sekolah dasar seperti angka romawi, bangun ruang dan hitung campurankedalamgame.



Gambar 2.2 Menu Utama EmpireOfMath



Gambar 2.3 GamePlayEmpireOfMath

- FlappyBat

Gameflappybat merupakan salah satu yang dibuat dengan menggunakan bahasa Lua. Cara main *gameflappybat* sama dengan *flappybird* yang membedakan hanya pada karakter yang digunakan, pada *flappybat* menggunakan lalat sebagai karakternya.



Gambar 2.4 Flappybat

- **Protect The Planet**

Protectthe planet merupakan game yang menggunakan bahasa pemrograman Lua. Dalam *game* ini *user* akan disuruh melindungi planet dari ancaman dari monster-monster luar angkasa. Cara mainnya dengan cara *tap* ke objek musuh yang akan menyerang planet. Jika *user* berhasil

menghancurkan musuh maka poin akan bertambah sepuluh dan jika musuh berhasil sampai ke planet maka status bar akan berkurang.



Gambar 2.5 Menu Utama Protect The Planet



Gambar 2.6 GamePlayProtectthe Planet

2.5 Pemodelan dan Elemen Pada Game

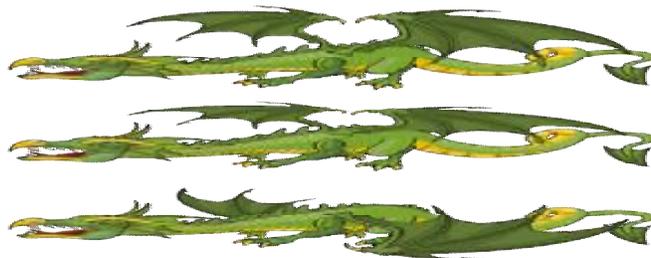
2.5.1 Adobe Photoshop

Adobe Photoshop merupakan program yang sangat handal, program ini juga merupakan program standar yang digunakan dalam dunia *imaging* dan *retouching* foto (Adobe photoshop untuk desainer web, hal.1). Adobe Photoshop juga salah satu perangkat lunak canggih yang dapat digunakan untuk membuat,

menyunting dan memanipulasi tampilan termasuk mengoreksi warna dan memberikan efek tampilan atas sebuah gambar atau photo. Hasil dari program Adobe Photoshop merupakan sebuah gambar atau image yang di dalam program komputer grafis terbagi menjadi dua kelompok, yaitu gambar bitmap dan vektor. Adobe Photoshop merupakan software grafis berbasis bitmap (pixel), yang biasa dipakai untuk mengedit foto, membuat ilustrasi bahkan desain web. Sehingga banyak digunakan di studio foto, percetakan, productionhouse, biro arsitektur, pabrik tekstil, dan bidang yang berkaitan dengan Teknologi Informasi (Prihantari, 2013:51).

2.5.2 Sprite

Sprite merupakan gambar animasi yang nantinya menjadi assets dalam sebuah *game*. Sprite dapat berupa benda seperti peluru, monster, karakter utama dalam game, musuh, kekuatan spesial, kunci atau pintu dan semua assets yang akan digunakan dalam pembuatan game. Pada umumnya Sprite adalah grafis animasi. Grafis animasi dibuat dari sprite yang sama namun berbeda penampakannya. Gambar 2.7 menunjukkan sebuah sprite karakter naga terbang (Martono dan Kurniawan, 2015:27).



Gambar 2.7 Contoh Sprite Naga

2.6 Android

2.6.1 Sejarah Android

Perjalanan Android dimulai sejak Oktober 2003 ketika 4 orang pakar IT, Andi Rubin, Rich Miner, NickSears dan Chris White mendirikan Android.Inc, di California US. Visi Android untuk mewujudkan mobiledevice yang lebih peka dan mengerti pemiliknya, kemudian menarik raksasa dunia maya Google. Google kemudian mengakuisisi Android pada Agustus 2005. OS Android dibangun berbasis platform Linux yang bersifat open source, senada dengan Linux, Android juga bersifat Open Source. Dengan nama besar Google dan konsep open source pada OS Android, tidak membutuhkan waktu lama bagi Android untuk bersaing dan menyisihkan Mobile OS lainnya seperti Symbian, Windos Mobile, Blackberry dan iOS. Kini siapa yang tak kenal Android yang telah menjelma menjadi penguasa *Operating System* bagi *smartphone* (Lengkong et.al, 2015:20).

2.6.2 ArsitekturAndroid

Menurut Murtiwiyati dan Gleen (2013:2-3) arsitektur Androiddiantaranya adalah sebagai berikut: a) Applications dan Widgets, Applications dan Widgets ini adalah layer dimana berhubungan dengan aplikasi saja, di mana biasanya

download aplikasi dijalankan kemudian dilakukan instalasi dan jalankan aplikasi tersebut. b) ApplicationsFrameworks, adalah layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti contentproviders yang berupa sms dan panggilan telepon. c)Libraries, adalah layer di mana fitur-fiturAndroid berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas kernel, Layer ini meliputi berbagailibrary C/C++ inti seperti Libc dan SSL. d) Android Run Time Layerberfungsi yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan Implementasi Linux. e) LinuxKernel, adalah layer di mana inti dari *operatingsystem* dari Android itu berada. Berisi file-filesystem yang mengatur sistem *processing, memory, resource, drivers* dan sistem-sistem operasi Android lainnya. Linuxkernel yang digunakan Android adalah linuxkernelrelease 2.6.

2.6.3 Software Development Kit (SDK)

Menurut Abdul Kadir (2013:5) Software Development Kit merupakan salah tools API (ApplicationInterfaceProgramming) yang digunakan mengembangkan aplikasi platform Android dengan bahasa pemrograman JAVA.Android SDK adalah kumpulan *software* yang berisi mengenai pustaka, *debugger*(alat pencari kesalahan), emulator (peniru perangkat bergerak), dokumentasi, kode contoh dan panduan. Keberadaan Android SDK dapat membantu *developer* untuk membuat dan menguji aplikasi Android, tanpa harus

mempunyai perangkat keras berbasis Android. Selain itu Android SDK dapat kompetibel dengan Windows, Mac dan Linux.

2.7 SublimeText Editor

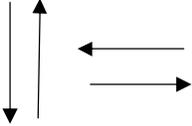
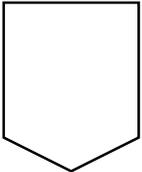
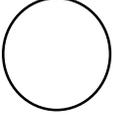
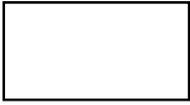
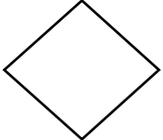
SublimeText adalah teks editor berbasis Python, sebuah teks editor yang elegan, kaya akan fitur, cross-platform, mudah dan simpel yang cukup terkenal di kalangan pengembang, penulis, dan desainer. Para programmer biasanya menggunakan sublimeText untuk menyunting *sourcecode* yang sedang dikerjakan(Yusti, 2015).

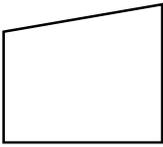
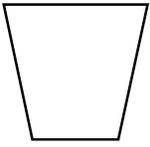
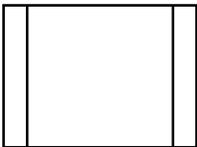
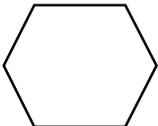
2.8 Flowchart

Flowchart adalah suatu simbol yang digunakan untuk menggambarkan alur atau proses secara detail jalannya aplikasi dari awal hingga akhir. *Flowchart* umumnya digunakan untuk mendiskripsikan dan medokumentasikan urutan dari prosedur-prosedur yang sudah dibuat secara keseluruhan dalam bentuk grafik. Adapun simbol-simbol yang digunakan dalam *flowchart* adalah sebagai berikut :

Tabel 2.1 Flowchart

Simbol	Nama	Fungsi
--------	------	--------

Simbol	Nama	Fungsi
	<p><i>FlowDirection</i></p>	<p>Berfungsi untuk menghubungkan antara simbol yang satu dengan simbol yang lain</p>
	<p><i>ConnectorSymbol</i></p>	<p>Simbol untuk penyambungan proses pada lembar halaman yang berbeda</p>
	<p><i>ConnectorSymbol</i></p>	<p>Simbol untuk penyambungan proses pada lembar halaman yang sama</p>
	<p><i>ProcessingSymbol</i></p>	<p>Simbol yang menunjukkan pengolahan yang dilakukan komputer</p>
	<p><i>DecisionSymbol</i></p>	<p>Simbol pemilihan proses berdasarkan kondisi yang ada</p>
	<p><i>Input-OutputSymbol</i></p>	<p>Simbol yang menyatakan proses input dan output</p>

Simbol	Nama	Fungsi
	<i>Manual InputSymbol</i>	Simbol untuk inputan data manual
	<i>Manual OperationSymbol</i>	Simbol yang menunjukkan pengolahan yang tidak dilakukan komputer
	<i>PredifineProcessSymbol</i>	Simbol yang menunjukkan proses yang sudah dikenal
	<i>PreparationSymbol</i>	Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage
	<i>TerminatorSymbol</i>	Simbol untuk permulaan (<i>start</i>) dan akhir (<i>stop</i>) dari suatu kegiatan

2.9 Penelitian Sebelumnya

Penelitian sebelumnya merupakan bagian pembahasan mengenai literatur-literatur yang digunakan sebagai acuan apakah permainan tersebut layak untuk diciptakan atau tidak. Berikut adalah beberapa tinjauan penelitian sebelumnya

yang dibuat menggunakan Corona SDK, bahasa pemrograman Lua dan berbasis Android sebagai acuan dalam perancangan *game* “The Djo” :

1. Tracing Aksara Jawa adalah *game* yang dibuat menggunakan *engine* Corona SDK dengan bahasa pemograman Lua. Sesuai dengan judul pada tampilan awal *game* “Tracing Aksara Jawa” pada gambar 2.8, *game* ini dibuat dengan tujuan sebagai media pembelajaran mengenal dan menulis aksara Jawa, dalam *game* ini *user* akan disuruh menulis aksara dengan cara mengikuti pola yang sudah disediakan, selain itu pada gambar 2.9 terdapat menu latihan dimana jika menu latihan tersebut dipilih maka *user* akan dialihkan ke tampilan latihan soal mengenai *game* “Aksara Jawa”. (Dewi et.al, 2014).



Gambar 2.8 Tampilan Awal Game Tracing Aksara Jawa



Gambar 2.9 Tampilan Menu Game Tracing Aksara Jawa

2. *TrashGrabber* adalah game edukasi yang dibuat untuk mengenal jenis-jenis sampah, game ini dibuat dengan menggunakan *engine* Corona SDK dengan bahasa pemrograman Lua. Pada gambar 2.10 menunjukkan tampilan awal dari game “*TrashGrabber*” terdapat menu-menu pilihan seperti adventure, survival, highscore dan *exit*. Pada gambar 2.11 menunjukkan *gameplay* dari permainan *TrashGrabber*, untuk level pertama menggunakan latar *background* perpustakaan. Pada game *TrashGrabber* terdapat banyak level dengan berbagai *background* antara lain taman, ruang tamu dan ruang kelas. Pada gambar 2.12 merupakan tampilan score, tampilan tersebut akan muncul jika waktu yang disediakan sudah habis dan *user* selesai menyelesaikan misi dalam level tersebut (I Dewa et.al, 2014).



Gambar 2.10 Menu TrashGrabber



Gambar 2.11 GamePlayTrashGrabber



Gambar 2.12 Timeout Level Success

3. Game Edukasi Rambu Lalu Lintas berbasis Android adalah *game* yang bertujuan untuk mengenalkan rambu-rambu lalu lintas yang sering ditemui di jalan. Pada gambar 2.13 merupakan tampilan *gameplay* dari game ini, sekilas dari tampilannya, user akan disuruh memilih jawaban yang sesuai soal dan gambar rambu yang ditampilkan, jika benar maka *score* akan bertambah dan jika salah maka *score* tidak akan bertambah atau berkurang, selain itu terdapat batas waktu yang disediakan untuk menjawab pertanyaan yang ditampilkan. Jika *user* selesai menjawab semua pertanyaan atau waktu berkurang maka akan muncul tampilan *score*, jumlah jawaban benar, jumlah jawaban salah seperti pada gambar 2.14. Pada *game* ini disediakan satu menu galeri seperti pada gambar 2.15, menu galeri ini berfungsi referensi tentang rambu-rambu lalu lintas lengkap dengan fungsinya (Jaya et.al, 2015).



Gambar 2.13 GamePlay Rambu Lalu Lintas



Gambar 2.14 Completed Level



Gambar 2.15 Scene Galeri