

BAB III

METODE PENELITIAN DAN PERANCANGAN

3.1 Konsep *Side Scrolling Endless Run*

Endless running game adalah *platform game* dimana karakter dari *game* tersebut akan terus menerus berlari di atas jalur yang dikeluarkan oleh sistem *game* tersebut. Dalam *game* ini pergerakan karakter terbatas hanya untuk meloncat, menyerang rintangan, atau melakukan gerakan khusus yang ada di dalam *game* tersebut. Objektif atau misi yang berada di dalam *game* ini adalah berlari sejauh mungkin sebelum karakter mati.

Konsep *side scrolling endless run* ini dipilih karena peminat genre *game* ini berasal dari segala macam golongan masyarakat, dari mulai anak-anak, remaja, sampai orang dewasa sekalipun, selain mudah dimainkan oleh para pengguna, tujuan dari genre ini mudah untuk dimengerti, karena genre *game* ini hanya menuntut penggunanya untuk mengumpulkan skor sebanyak-banyaknya.

3.2 Metodologi Pembuatan *Game*

Dalam pembuatan *game raccoon run* ini ada beberapa metodologi penelitian seperti *iterative*, *waterfall*, dan *agile*. Pada kesempatan kali ini penulis menggunakan metodologi *iterative*, karena jika menggunakan metode *waterfall*, proses yang telah terjadi tidak akan dapat diulangi kembali, dan jika menggunakan metode *agile*, banyak proses awal yang tidak dapat dilakukan karena tidak memiliki data data yang cukup. Berikut adalah penjelasan dari metode *iterative*:

3.2.1 Metode *Iterative Game Development*

Metode *Iterative* ini akan memproses *game* secara terus menerus karena sebuah *game* akan selalu mengalami perubahan disetiap bagiannya. *Game* akan selalu memiliki usulan, purwarupa, *running test*, dan akan selalu ada evaluasi saat *game* sudah rilis ataupun belum rilis. Metode *iterative* memiliki prinsip kerja yaitu tidak realistik jika membuat sebuah *game* yang ideal dan diinginkan hanya dalam satu kali percobaan. Dengan membuat dan mengetes inti dari kriteria *game* (misalnya kesenangan pengguna saat bermain), desainer *game* akan mampu meningkatkan potensi pasaran *game* yang telah dibangun.

Metode *iterative* dapat diterapkan dibidang mana saja, tetapi metode ini sangat berpengaruh besar dalam pembuatan *game*. *Game* desainer membuat purwarupa dan menjalankan *testing* untuk mengukur *game element functionality* dan memperluas dan menemukan konsep *game* yang baru untuk meningkatkan kepuasan pengguna.



Gambar 3.2.1 Metode *Iterative*

Pada metode ini dibagi menjadi beberapa bagian yaitu membuat konsep *game*, membuat desain *game*, *development*, meluncurkan *game*, mengumpulkan data, menganalisis data *game*.

Membuat konsep *game* adalah proses untuk membuat dan memikirkan konsep *game* yang akan dibuat, contohnya *game* FPS (*First Person Shooter*), MOBA (*Multiplayer Online Battle Arena*), *board game*, atau jenis lainnya.

Desain *game*, proses ini adalah untuk membuat desain dari *game* yang sudah ditentukan konsepnya. Pada tahap ini, *developer* membuat purwarupa dari *game* yang ingin dikembangkan lebih lanjut.

Development, pada proses ini *developer* akan membuat *game* yang sudah memiliki konsep dan purwarupa yang pada proses sebelumnya dilakukan, *developer* akan membuat *game* sampai *game* tersebut selesai dan siap diluncurkan ke pasaran.

Release, tahap ini adalah tahap dimana *developer* siap meluncurkan *game* yang telah selesai dibuat ke pasaran agar dapat digunakan oleh pengguna.

Capture data, pada tahap ini *developer* akan mencari umpan balik atau komentar-komentar dari para pengguna *game* yang sudah dirilis.

Analisis data *game*, pada tahap ini pihak *developer* akan menggunakan umpan balik, komentar, kritik dan saran dari para pengguna agar *developer* dapat mencari, memperbaiki, dan mengembangkan *game* agar dapat diterima oleh pengguna.

Keuntungan dan kerugian menggunakan metode *iterative* adalah:

Tabel 3.2.1 keuntungan dan kerugian menggunakan metode *iterative*

Keuntungan	Kerugian
Dalam metode <i>iterative</i> dapat membuat desain dari aplikasi yang ingin dibuat sebelum membuat produk.	Setiap fase dari metode <i>iterative</i> itu bersifat kaku, karena tidak dapat melangkahi fase yang belum selesai.
Dalam metode <i>iterative</i> developer dapat membuat dan mengembangkan produk langkah demi langkah, dan developer dapat melacak kesalahan pada fase awal, sehingga dapat mencegah penurunan kualitas dari produk.	Sistem arsitektur yang mahal dan beban desain yang tinggi karena tidak semua persyaratan untuk membuat produk dikumpulkan sebelum tahap pembuatan produk.
Dalam metode <i>iterative</i> developer lebih mengutamakan untuk mendesain produk dari pada mengumpulkan dokumen	

3.3 Metode Penelitian

Metode *iterative* yang digunakan pada penelitian ini adalah sebagai berikut:

1. **Membuat Konsep *Game***
 Penelitian ini diawali dengan membuat konsep *game* yang akan dikembangkan, memilih tema dan genre *game* dan peneliti menentukan untuk membuat *side scrolling endless run*. Konsep ini mengadaptasi dari *game "cookie run"* dan "*flappy bird*"
2. **Desain *Game***

Selanjutnya peneliti membuat desain *game* yang akan dibuat karakter, *terrain*, dan musuh dari karakter mulai dibuat purwarupanya.

3. *Development*

Pada tahap ini *game* akan didevelop menggunakan *construct 2* dengan menggunakan konsep dan desain *game* yang telah dibuat sebelumnya.

4. *Release*

Game akan dipublikasikan dengan menggunakan *.apk* yang dapat dimainkan di android.

5. *Capture Data*

Tahap ini peneliti akan mengumpulkan umpan balik dari pengguna, baik saran maupun kritik yang berguna untuk mengembangkan *game* ini ke depannya.

6. *Analyze Game Data*

Pada proses ini, umpan balik yang telah didapatkan dari pengguna digunakan untuk mengembangkan *game* sesuai dengan kritik dan saran yang baik dan membangun, agar *game* ini dapat lebih diterima oleh pengguna.

3.4 Perbandingan *Construct 2* dengan *Godot Tools*

Penelitian ini dibuat menggunakan program *construct 2*, tapi masih banyak program untuk membuat *game* contoh nya seperti *unity*, *unreal engine*, *godot tools*, dan masih banyak lagi. Peneliti membandingkan *construct 2* dengan *godot tools* karena dua tools ini memiliki kesamaan yaitu sama sama unggul dalam *game 2D development*.. Berikut ini adalah perbandingan kelebihan dan kekurangan antara *Construct 2* dengan *Godot Tools*:

Kelebihan

Tabel 3.1.1 Perbandingan kelebihan menggunakan Construct 2 dan Godot Tools

<i>Construct 2</i>	<i>Godot Tools</i>
Hanya memerlukan pengetahuan programming yang minim	Mudah dipelajari karena menggunakan bahasa pemrograman GD Script hampir menyerupai <i>python</i>
<i>Developer</i> construct 2 selalu memperbaharui program construct 2, biasanya 2 minggu sekali	<i>Built in physics</i> untuk 2D dan 3D
Dapat membuat <i>game</i> dengan waktu yang singkat karena hanya memerlukan sedikit pengetahuan pemrograman	<i>Multi platform</i> (Windows, OS X, Linux, iOS, Android, BlackBerry, dan HTML 5)
<i>Built in physics</i> untuk 2D	Dapat digunakan di iMac
Tersedia versi gratis tetapi memiliki fitur yang terbatas	Dapat di develop di windows dan dijalankan menggunakan Linux
<i>Multi platform</i> (memiliki 15 dukungan platform termasuk Windows, Linux, HTML 5, iOS, Android, Chrome Store, PhoneGap, dan Scirra)	Dapat mengganti bahasa yang digunakan didalam program godot
Mudah untuk membuat animasi	Tampilan tatap muka mudah digunakan oleh yang bukan programmer sekalipun
Tersedia di Steam	Memiliki ukuran kecil, hanya 30mb
<i>Developer</i> tersedia untuk membantu mengembangkan <i>game</i>	

Tabel 3.1.1 Perbandingan kelebihan menggunakan Construct 2 dan Godot Tools lanjutan

Memiliki performa yang baik saat dijalankan ditelpon pintar	
Tersedia fitur kamera, mikrofon, pengenalan suara	
Fitur animasi dan edit gambar	
Dapat menggunakan intel cordova <i>plugins</i>	

Kekurangan

Tabel 3.1.2 Perbandingan kekurangan menggunakan Construct 2 dan Godot Tools

<i>Construct 2</i>	<i>Godot Tools</i>
Versi gratis memiliki banyak batasan fitur fitur penting dalam membuat game yang baik	Manajemen <i>tileset</i> (kumpulan dari data vector) bisa lebih efisien
Construct 2 hanya ada di windows	Sulit digunakan untuk pengguna program Unity
HTML 5 hanya dapat dijalankan di <i>browser</i> tertentu	<i>2D physics</i> lemah jika dibandingkan dengan Box2d
HTML 5 bergantung dengan performa <i>browser</i>	

3.5 Spesifikasi Perangkat

Pada penelitian ini dibutuhkan perangkat pendukung untuk membangun sistem. Perangkat yang digunakan adalah perangkat keras (*Hardware*) dan perangkat lunak (*Software*) dengan spesifikasi sebagai berikut:

1. Perangkat keras (*laptop*)

Prosesor : Intel Core i7-2630QM 2.0GHz

Memori : 12 GB

Harddisk : 1.5TB

2. Perangkat lunak

Sistem operasi : Windows 10 Home(64-bit)

IDE : Construct 2

Pengolah kata : Microsoft word 2013

Browser : Google Chrome

Language : Construct 2

3.6 Karakter

Dalam *game* karakter adalah elemen terpenting karena para pemain *game* akan memilih karakter yang disukai untuk dimainkan didalam *game*. Dalam pembuatan proyek ini karakter *racoon* menjadi tokoh utama dalam *game racoon run*.

3.7 Sistem Poin

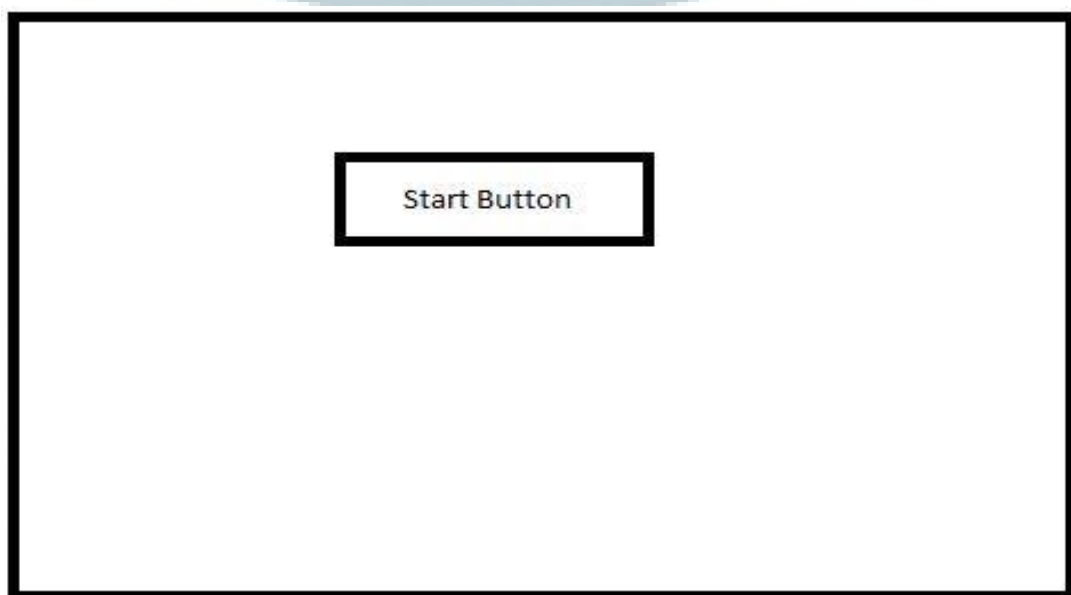
Sistem poin pada *game side scrolling endless running* ini adalah menggunakan sistem dengan mengumpulkan bintang yang berada di dalam *game* dengan cara menabrakkan karakter ke bintang untuk memperbanyak skor yang didapatkan. Skor akan berhenti saat karakter terjatuh, menabrak musuh, atau tersangkut oleh map dan hilang dari layar.

3.8 Misi

Misi dari *game* ini adalah mengumpulkan skor dengan menabrakkan karakter *raccoon* ke bintang sebanyak-banyaknya agar dapat menambah point dan bersaing skor tertinggi dengan teman-teman yang memainkan *game* ini juga.

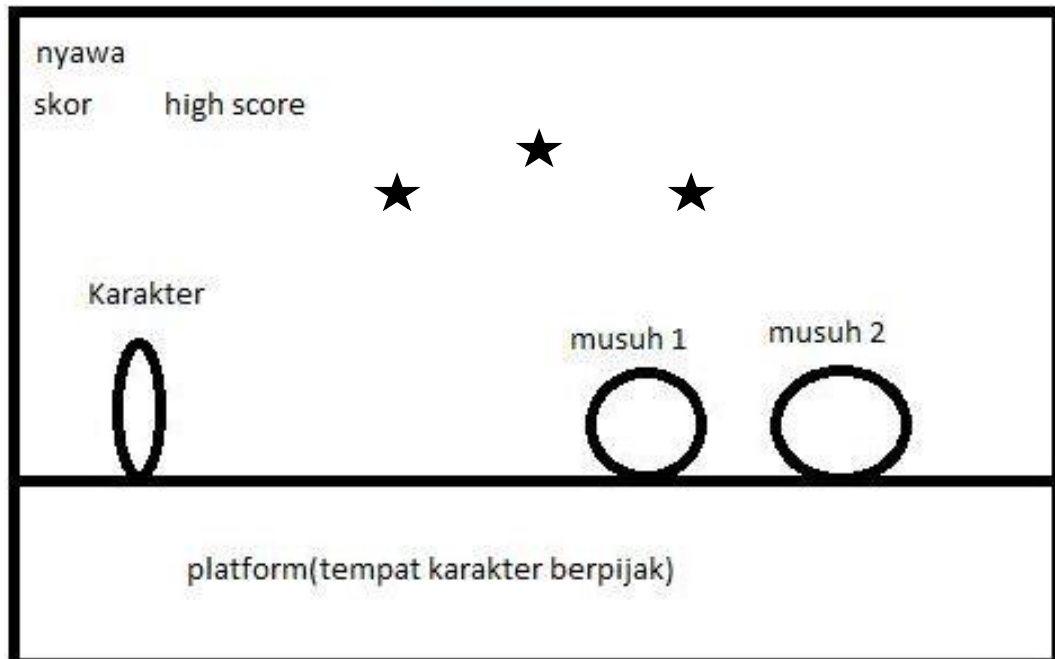
3.9 Perancangan Antarmuka Pengguna

Pada perancangan antarmuka terdapat 3 bagian antarmuka yang akan dapat digunakan oleh pengguna (*user*) yaitu:



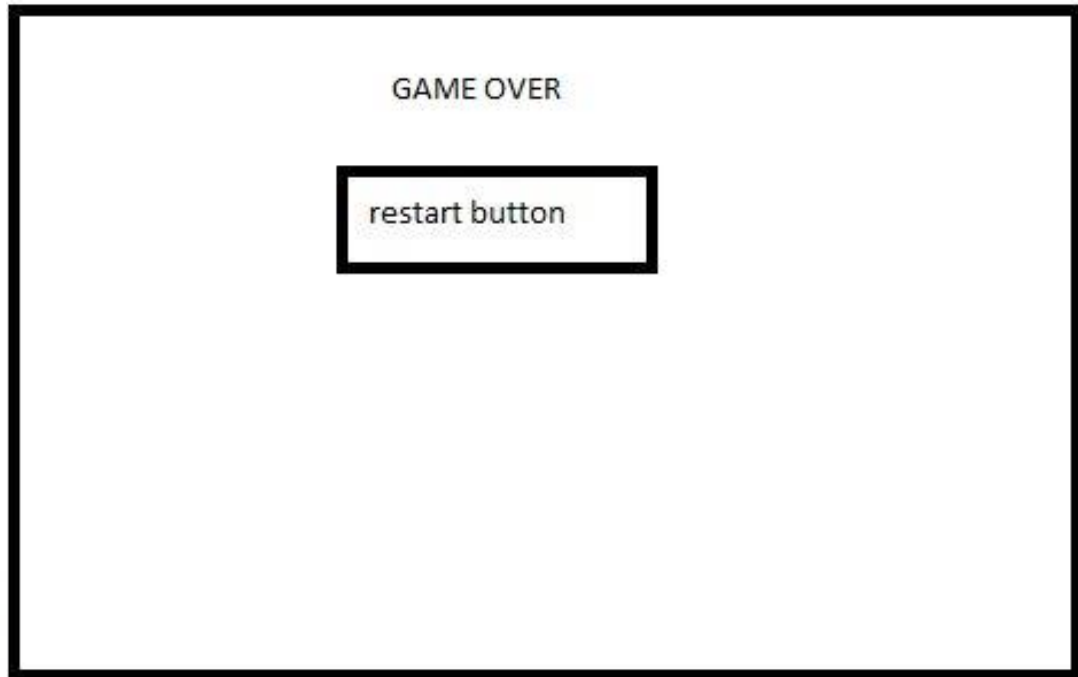
Gambar 3.8.1 Tatapan antarmuka *Home*.

Pada antarmuka halaman *Home*, pengguna akan melihat *start button*, saat pengguna menekan tombol *start*, maka tampilan akan langsung berubah ke tampilan antar muka bagian *game*.



Gambar 3.8.2 Tatapan antarmuka *Game*.

Pada antarmuka halaman *game* pengguna akan melihat karakter yang akan digunakan oleh pengguna, nyawa dari karakter yang digunakan oleh pengguna, skor yang telah dikumpulkan oleh pengguna saat *game* tersebut masih berjalan (karakter belum mati), *high score* untuk mencatat dan menampilkan skor tertinggi yang pernah diperoleh selama aplikasi belum ditutup, musuh 1 dan musuh 2 yang akan mengganggu pengguna dalam mengumpulkan bintang jika karakter menabrak atau mengenai musuh 1 dan 2 akan mengurangi nyawa dari karakter, bintang yang akan muncul di tempat random dengan waktu yang di tentukan agar dapat dikumpulkan oleh pengguna dan platform lantai atau dasar yang digunakan oleh karakter untuk berpijak.



Gambar 3.8.3 Tatapan antarmuka *GameOver*.

Saat karakter yang digunakan oleh pengguna mati (dengan cara nyawa habis atau terjatuh dari platform), pengguna akan secara otomatis berpindah halaman ke halaman antarmuka *GameOver*, lalu tersedia *restart button* yang jika dipilih oleh pengguna akan mengulang kehalaman *game* dari awal.

U M N