BAB I

PENDAHULUAN

1.1 Latar Belakang

Dunia teknologi pada saat ini berkembang dengan dangat pesat. Perkembangan teknologi mencakup semua bidang kehidupan manusia seperti kesehatan, pangan, industri dll. Begitupun dengan perkembangan teknologi dalam bidang *game*. Perkembangan teknologi dalam bidang *game* saat ini sudah sangat bagus karena *game* seperti kondisi nyatanya.

Pada kemunculan *game* pertama kalinya, *game* masih disajikan secara sederhana dan dan diprakarsai oleh Steven Russel dan proyek yang bernama *Computer Games* pada tahun 1962 dengan produk andalannya bernama Star Wars. Beberapa puluh tahun kemudian, banyak *game* bermunculan dari *game* 2 dimensi dan *game* 3 dimensi.

Game pada dasarnya bersifat hiburan karena jika pengguna memainkan *game* maka akan terasa senang. Dalam era saat ini, *game* disajikan dengan kualitas visualisasi yang sangat canggih karena didukung oleh teknologi sehingga pemain lebih interaktif sesuai kemauannya sendiri dan pemain terasa hidup dalam *game* tersebut. Maka bisa disebutkan bahwa *game* berkembang beriringan dengan teknologi.

1Dalam hal ini penulis tertarik pada *game* bergenre *racing* yaitu *game CTR (Crash Team Racing)*, *game racing* adalah *game* yang menantang pemain untuk beradu kecepatan dan kegesitan dengan komputer atau pemain lain menggunakan

beragam kendaraan seperti mobil, motor, dan sebagainya tergantung *game racing* apa yang dimainkan. Contoh *game racing* yang terkenal adalah seri *Need For Speed* dan seri *Test Drive* (Putra , 2017).

Penulis juga terinspirasi dari *game* bergenre arcade yaitu Hamsterball, *Game* arcade adalah *game* yang tidak terfokus pada alur cerita, melainkan hanya dimainkan untuk bersenang-senang sebagai pengisi waktu senggang atau hanya untuk mencari nilai tertinggi. Permainan dengan genre arcade biasanya merupakan permainan yang membutuhkan refleks pemain, dan biasanya hanya menampilkan sedikit teka-teki atau pemikiran yang kompleks atau keahlian strategi. Kontrol pada permainan arcade biasanya lebih mudah dan lebih sederhana (Soemitha, Pragantha dan Haris, 2020).

Berdasarkan review dari website IGN.com dan Metacritic.com para pemain merasa *game* ini kurang seru untuk dimainkan karena tidak adanya lawan tanding yang membuat permainan semakin menarik, dari segi level *game* ini hanya menambahkan beberapa rintangan dan memperpanjang jarak tempuh
dari lintasan.

Dalam pengembangan *game* tentu diperlukan sebuah algoritma yang berguna untuk langkah sistematis dalam penyelasaian masalah, saat ini sudah ada algoritma yang berguna untuk mendeteksi suatu objek dengan objek lainnya. seperti algoritma *Collision Detection*.

Collision detection merupakan teknik deteksi tabrakan untuk mengetahui objek-objek apa saja yang bersentuhan dalam bidang koordinat tertentu. Objek-objek ini bisa saja memiliki bentuk yang sangat bervariasi.

Objek-objek pada *game* memiliki bentuk yang bervariasi, ada yang berbentuk kotak, segi-n, sampai bentuk pesawat pemain yang sangat mendetail. Untuk mempercepat proses pada collision detection, umumnya objek - objek ini direpresentasikan secara logic dengan bentuk primitif seperti segiempat dan lingkaran (jika pada koordinat dua dimensi), atau kubus dan bola (jika pada koordinat tiga dimensi). Bentuk primitif yang merepresentasikan objek ini biasa disebut sebagai Bounding Box atau Bounding Circle(Nurdiyanto, Winarno , 2018).

Dilihat dari penelitian sebelumnya,yaitu pada jurnal yang berjudul "PENERAPAN METODE *COLLISION DETECTION* PADA *GAME* PETUALANGAN MENGGUNAKAN AKSARA JAWA" dapat disimpulkan bahwa *game* ini dapat memberikan pengetahuan dasar mengenai bentuk serta pengenalan aksara jawa sekaligus sebagai sarana hiburan. Penelitian ini menghasilkan *game* edukasi petualangan untuk mengenal dan memahami aksara jawa yang digunakan sebagai obyeknya. *Game* edukasi menggunakan metode *collision detection* dapat bekerja dengan baik dan menghasilkan sebuah permainan edukasi yang bermanfaat terutama untuk anak-anak dalam mengenal aksara jawa (Nurdiyanto, Winarno , 2018).

Berdasarkan uraian yang telah dipaparkan di atas penulis akan mengimplemetasikan algoritma *Collision Detection* pada *game* yang akan dibuat, maka pada penelitian ini penulis mengambil judul "**Implementasi Algoritma Collision Detection Pada Pengembangan *Game Racing* Arcade Running Ball**".

1.2 Identifikasi Masalah

Identifikasi masalah adalah proses yang paling penting untuk menemukan masalah yang berada diobjek penelitian yang diteliti. Berdasarkan *review* dari beberapa website dan forum, maka dapat dirumuskan beberapa identifikasi masalah sebagai berikut:

1. *Game Hamsterball* kurang menarik dari segi lintasan dan tantangan yang membuat game menjadi *repetitive* (berulang)
2. *Game hamsterball* menjadi membosankan karena pada game tidak ada lawan tanding.

1.3 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah sebagai berikut :

1. Bagaimana cara mengimplementasikan algoritma Collision Detection untuk penambahan *power up* pada *player*?
2. Bagaimana cara untuk menambahkan pengembangan dari *game* Hamsterball sebelumnya dimana *player* hanya melewati rintangan dan sampai *finish* dengan waktu yang ditentukan?

1.4 Batasan Masalah

Agar pembahasan pada penelitian tidak melebar, maka penulis akan membatasi masalah sebagai berikut :

1. *Game* yang dikembangkan akan dibuat hanya pada platform android saja.
2. *Game* ini dikembangkan untuk sistem operasi *Android versi* 4.4 *Kit Kat*

(API level 19), dan RAM 2GB.

3. *Game* hanya bisa dimainkan secara offline.
4. *Game* hanya bisa dimainkan secara single*player*.
5. *Game* dapat menambahkan *user* dan dapat menyimpan *score*.
6. Level yang akan dirancang, yaitu :

a. Level Pertama *player* harus mengalahkan 1 *NPC* dengan menuju garis *finish* melalui lintasan level mudah dengan rintangan berupa batu.

b. Level Kedua *player* harus mengalahkan 2 *NPC* dengan menuju garis *finish* melalui lintasan level sedang dengan rintangan berupa batu dan lubang.

c. Level Ketiga *player* harus mengalahkan 2 *NPC* dengan menuju garis *finish* melalui lintasan level sulit dengan rintangan berupa batu, lubang dan jembatan.

7. Algoritma yang digunakan adalah *Collision Detection* berfungsi untuk Mendeteksi tabrakan antar *player* dan pengambilan *power up*.
8. *Tools* dan bahasa pemrogaman yang digunakan untuk pengembangan *game* yaitu :

a. *Game engine* yang digunakan Unity 2019.4.40f1 (64-bit). dengan bahasa pemerograman C#(*CSharp*).

b. Blender 3.2.2, untuk pembuatan animasi 3D

c. CorelDraw 2019 (64-bit). untuk perancangan antarmuka (*user interface*)

d. Draw.io, untuk membangun perancangan *Flowmap, Game Layout Chart* dan *Flowchart*

e. Visual Studio code, untuk *text editor*.

1.5 Tujuan Penelitian

Adapun tujuan yang hendak dicapai dalam penelitian ini adalah :

1. Membangun *game* Running ball dengan menambahkan item *power up*, rintangan yang harus dilewati dan *NPC* yang harus dikalahkan untuk menyelesaikan stage yang ada.
2. Untuk menerapkan algoritma *Collision Detection* pada *game Running Ball* sebagai pendeteksi tabrakan dan pengambilan *power up*.

1.6 Manfaat Penelitian

Adapaun manfaat yang ingin didapat dari penelitian ini adalah:

1. Bagi Penulis.

a. Dapat menambah pengetahuan dan wawasan penulis mengenai cara pembuatan *game* menggunakan *Unity Engine*..

b. Dapat menambah pengetahuan dan wawasan penulis mengenai penerapan algoritma *Collision Detection* di *game Running Ball*

2. Bagi Pemain.

a. Penambahan fitur dalam *game* diharapkan membuat *game* menjadi lebih menarik lagi.

b. Menggabungkan dua konsep *game* yang berbeda diharapkan membuat *gameplay* menjadi lebih variatif.

c. Merubah *platform* yang tadinya berada di konsol dan dekstop di rubah ke *platform* android yang di harapkan *game* bisa dimainkan kapan saja.

3. Bagi Pembaca.

Penelitian ini diharapkan dapat memberikan dan kontribusi untuk orang lain yang membutuhkan informasi ini mengenai topik penelitian ini dan menambah referensi untuk penelitian berikutnya.

1.7 Pertanyaan Penelitian

Berdasarkan identifikasi masalah yang telah diuraikan sebelumya, maka terdapat pertayaan penelitian sebagai berikut :

1. Apakah algoritma *Collision Detection* dapat digunakan untuk mendeteksi tabrakan dan Pengambilan sistem *power up*?
2. Apakah menggabungkan dua konsep *game* yang berbeda mampu membuat *game* lebih menarik?

1.8 Metodologi Penelitian

Metodologi penelitian adalah proses atau cara ilmiah untuk mendapatkan data yang akan digunakan untuk keperluan penelitian. Metodologi juga merupakan analisis teoretis mengenai suatu cara atau metode. Penelitian merupakan suatu penyelidikan yang sistematis untuk meningkatkan sejumlah pengetahuan, juga merupakan suatu usaha yang sistematis dan terorganisasi untuk menyelidiki masalah tertentu yang memerlukan jawaban.

1.8.1 Teknik Pengumpulan Data

Dalam metode ini membahas tentang cara memperoleh data yang akan dibutuhkan untuk penelitian, maka digunakan metode Studi Pustaka. Mengenai apa yang dimaksud dari metode tersebut dapat dijelaskan sebagai berikut :

Studi Pustaka

Studi pustaka dilakukan dengan menggunakan sumber-sumber seperti buku, jurnal, dan internet. Setudi liteatur ini berguna untuk mengetahui landasan teori, pengetahuan dan informasi pada penelitian ini. Yaitu jurnal referensi yang sesuai dengan judul penelitian. Membaca buku mengenai algoritma *Collision Detection*, jurnal mengenai *game Racing* dan *Unity,* artikel yang berhubungan dengan *game CTR* dan *Hamsterball* , penulis berharap untuk mendapatkan banyak informasi yang bersangkutan dengan penelitian ini.

1.8.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam perancangan *game* ini menggunakan metodologi *Game Development*

Life Cycle(GDLC)

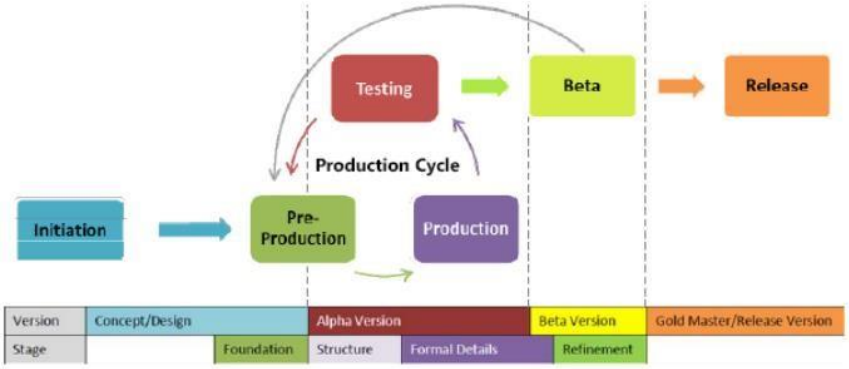
GDLC adalah suatu proses pengembangan sebuag *game* yang menggunakan pendekatan iteratif yang terdiri dari 6 fase pengembangan,dimulai dari fase inisialisasi/pembuatan

konsep,*preproduction,production,testing,beta dan realese*.

Dari 6 fase tersebut dapat di kelompokan menjadi 3 proses utama yaitu: 1. Proses inisialisasi yang terdiri dari konsep dan design,

2. Proses produksi terdiri dari praproduksi,produksi dan pengujian (*Alpha* dan *Beta*),
3. *Realease*.

Fase dan Proses *GDLC Guidelines* dapat dilihat pada gambar dibawah ini :



Gambar 1. 1 Fase dan Proses GDLC

Sumber (Andriyat , 2018)

1. Inisiasi

Inisiasi adalah proses awal yang berupa pembuatan konsep kasar dari *game*. Pada tahap ini penulis membuat *game* dengan konsep balapan yang bertemakan sebuah bola yang melewati berbagai rintangan dan adapun power up untuk mendapatkan tambahan kekuatan.

2. Pra-produksi

Pra-produksi adalah salah satu fase yang penting dalam siklus produksi. Praproduksi melibatkan penciptaan dan revisi desain *game* dan pembuatan prototipe permainan. Desain *game* berfokus pada mendefinisikan genre permainan, *game*play, *game* mekanik/konfensional, alur cerita, karakter, tantangan, faktor kesenangan, aspek teknis, dan dokumentasi elemennya dalam Dokumen Desain

Game (GDD) Pra-produksi berakhir ketika revisi atau perubahan desain *game* telah disetujui dan didokumentasikan di GDD. Pada pengembangan *game* ini penulis membuat sebuah *game* bergenre *racing* yang berfokus pada balapan sebuah bola yang akan melewati beberapa rintangan dan melawan musuh, dan untuk menambah keseruan pada *game* penulis menambahkan power up yang akan ada dalam track, dan untuk leveling nantinya akan mengurut dari yang mudah sampai tersulit mulai dari penambahan NPC dan

penambahan rintangan.

3. Produksi

Produksi adalah proses inti yang berputar di sekitar penciptaan aset, pembuatan kode sumber, dan integrasi kedua elemen. Prototipe terkait dalam fase ini adalah perincian dan penyempurnaan formal. Rincian Formal adalah struktur yang disempurnakan dengan mekanika dan aset yang lebih lengkap. Kegiatan produksi yang terkait dengan penciptaan dan penyempurnaan detail formal adalah menyeimbangkan permainan (terkait dengan kriteria kualitas yang seimbang), menambahkan fitur baru, meningkatkan kinerja secara keseluruhan, dan memperbaiki bug (terkait dengan kriteria kualitas fungsional dan internal yang lengkap). Penyeimbangan permainan yaitu penyesuaian yang terkait dengan kesulitan permainan untuk membuat kesulitan *game* yang tepat (Leveling). Refinement adalah prototipe lengkap yang merupakan subjek dari permainan.

Kriteria kualitas terkait *game* fun dan dapat diakses. Kegiatan selama penyempurnaan diarahkan untuk membuat permainan lebih menyenangkan,

menantang, dan lebih mudah dipahami. Hanya perubahan kecil yang diizinkan dalam fase ini.

4. Pengujian

   Pengujian dalam konteks ini berarti pengujian internal dilakukan untuk menguji kegunaan permainan dan pemutaran. Metode pengujian khusus untuk setiap tahap prototipe. Pengujian yang dilakukan menggunakan metode *black box* dan *white box*. Pengujian *black box* merupakan Pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk mengetahui apakah fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Sedangkan pengujian *White box* merupakan Pengujian perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi masukan dan keluaran yang sesuai dengan spesifikasi kebutuhan.(Wahyu Nur Cholifah, 2018) Output dari pengujian adalah laporan bug, permintaan perubahan, dan keputusan pengembangan. Hasilnya akan memutuskan apakah sudah waktunya untuk maju ke fase berikutnya (Beta) atau mengulangi siklus produksi.

5. Beta

   Beta adalah fase untuk melakukan pengujian pihak ketiga atau eksternal yang disebut pengujian beta. Pengujian beta masih menggunakan metode pengujian yang sama dengan metode pengujian sebelumnya, karena prototipe terkait dalam pengujian beta adalah perincian dan penyempurnaan formal. Untuk metode pengujian menggunakan pengujian UAT. Metode pengujian UAT (User Acceptance Test) merupakan suatu metode pengujian oleh pengguna untuk

menghasilkan sebuah dokumen yang bertujuan sebagai bukti bahwa sistem yang dibuat telah dapat diterima oleh pengguna. Metode pemilihan tester datang dalam dua jenis: beta tettutup dan beta terbuka. Ditahap beta hanya memungkinkan individu yang diundang untuk menjadi peserta, sementara beta terbuka memungkinkan siapa saja yang mendaftar menjadi peserta. Kriteria kualitas dalam beta terkait erat dengan tahap prototipe saat ini. Dalam pengujian detail resmi, penguji diminta untuk menemukan bug (terkait dengan kriteria kualitas fungsional dan internal yang lengkap). Dalam penyempurnaan pengujian, penguji diberi lebih banyak kebebasan untuk menikmati permainan, karena sasaran lebih diarahkan untuk mendapatkan umpan balik (terkait dengan kriteria kualitas aksesibilitas dan menyenangkan). Output dari pengujian beta adalah laporan bug dan masukan pengguna. Sesi Beta ditutup terutama karena 2 alasan, baik jangka beta berakhir atau jumlah penguji beta yang ditentukan telah memberikan laporan uji mereka. Dari sini, dapat menyebabkan siklus produksi lagi untuk memperbaiki produk atau terus merilis *game* jika hasilnya memuaskan.

6. Rilis

   Sudah saatnya build *game* telah mencapai tahap akhir dan siap untuk dirilis ke publik. Rilis melibatkan peluncuran produk, dokumentasi proyek, berbagi pengetahuan, postmortems, dan perencanaan untuk pemeliharaan dan ekspansi permainan. (“Rido Ramadan and Yani Widyani; *Game Development Life Cycle*”)

1.8.3 Metode Penyelesaian Masalah

Berdasarkan tujuan utama dari penyusunan proposal penelitian adalah pengembangan sebuah *game racing* menggunakan *Unity* yang diharapkan dapat menjadi sebuah *game* yang lebih menarik dan banyak di minati khususnya pengguna smartphone android maka dari itu algoritma yang dipakai untuk *game* ini adalah algoritma *Collision detection*.

Collision detection merupakan teknik deteksi tabrakan untuk mengetahui objek-objek apa saja yang bersentuhan dalam bidang koordinat tertentu. Objek-objek ini bisa saja memiliki bentuk yang sangat bervariasi. Objek-objek pada *game* memiliki bentuk yang bervariasi, ada yang berbentuk kotak, segi-n, sampai bentuk pesawat pemain yang sangat mendetail. Untuk mempercepat proses pada collision detection, umumnya objek - objek ini direpresentasikan secara logic dengan bentuk primitif seperti segiempat dan lingkaran (jika pada koordinat dua dimensi), atau kubus dan bola (jika pada koordinat tiga dimensi). Bentuk primitif yang merepresentasikan objek ini biasa disebut sebagai *Bounding Box* atau *Bounding Circle*. Algoritma *collision detection* adalah proses pengecekan apakah beberapa buah objek spasial saling bertumpuk atau tidak. Jika ternyata ada paling sedikit dua buah objek yang bertumpuk, maka kedua objek tersebut dikatakan saling bertumpukan. Objek yang bertumpuk berarti objek spasialnya beririsan.

Collision detection adalah suatu metode yang membahas tentang bagaimana cara mengetahui objek-objek apa saja yang bersentuhan satu sama lain dalam bidang koordinat 2 dimensi ataupun 3 dimensi. Beberapa penelitian pembuatan *game* menggunakan metode *Collision Detection*. Dari beberapa penelitian yang

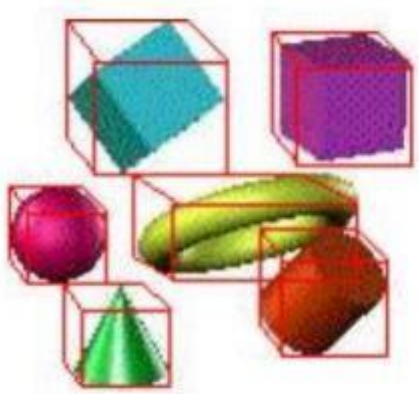
telah dilakukan oeh para peneliti sebelumnya, maka pada penelitian ini juga menggunakan metode *Collision Detection* yang diimplementasikan pada sebuah *game racing Running Ball*.

Adapaun kelebihan dari algoritma *Collision Detection* adalah kemampuannnya dalam penerapan pada skenario tertentu dan Algoritma collision detection juga banyak sekali digunakan hampir pada setiap game,

berdasarkan jurnal "PENERAPAN ALGORITMA *COLLISION DETECTION* DAN *BOIDS* PADA *GAME DOKKAEBI SHOOTER*" yang

disusun oleh Lia Musfiroh, Ahmad Jazuli, dan Anastasya Latubessy. Mereka menggunakan algoritma *Collison Detection* dan *Boids* untuk membangun gamenya.

Untuk memastikan tidak ada objek yang saling menembus dibutuhkan *collision detection* berdasarkan susunan geometri dari objek itu sendiri



Gambar 1. 2 Bounding Boxes

Sumber (Herwanto, Sonjaya 2016)

Dalam gambar di atas tiap bentuk dikelilingi oleh garis batas merah. Jika batas merah tersebut saling menumpuk bisa dikatakan objek di dalamnya sudah bertubrukan dengan objek di dalamnya batas merah yang lain atau belum

menumbuk sama sekali. Pengujian lebih dalam diperlukan untuk mendeteksi tumbukan didalam objeknya. Tetapi jika batasnya tidak saling bertubrukan maka objeknya pun juga belum bertabrakan. Dengan cara ini bisa menghemat sumber daya *CPU* untuk pengujian bentuk yang lebih kompleks. Kekurangannya cara ini terpaku pada sumbu yang sejajar jika sebuah objek berotasi proses akan menjadi semakin panjang karena setiap pendeteksian harus didahului dengan pemeriksaan koordinat yang selalu berubah ketika objek melakukan rotasi.

Untuk penyelesaian masalah penelitian ini, adapun flowchart dari algoritma *Collision Detection* ini adalah sebagai berikut :

Gambar 1. 3 Flowchart Collision Detection

Sumber (Musfiroh, Jazuli, Latubesy, 2014)

1.9 Sistematika Penulisan

Sistematika penulisan adalah sebagai berikut:

Secara garis besar sistematika terbagi menjadi dua, yaitu bagian depan dan bagian

isi:

Bagian Depan

Pada bagian depan terdiri dari Halaman Kulit Luar, Halaman Judul, Halaman Pengesahan, Pernyataan Originalitas, Halaman Motto Dan Atau Persembahan, Abstrak, Kata Pengantar, Daftar Isi, Daftar Gambar dan Daftar Tabel.

Bagian Isi

BAB I Pendahuluan

Bab yang berisi Latar Belakang, Rumusan Masalah, Identifikasi Masalah, Rumusan Masalah, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian, Metodologi Penelitian Dan Sistematika Penulisan.

BAB II Landasan Teori

Bab yang berisikan teori-teori sebagai penunjang yang berhubungan dengan penelitian yang beracukan pada Judul dan Pertanyaan
Penelitian, menemukan solusi dari Penelitian Terdahulu dan Kerangka Teoritis.

BAB III Analisis Dan Perancangan

Bab III berisikan perancangan sistem, analis sistem dan juga perancangan antar muka pada game yang dibuat.

BAB IV Pengujian Dan Implementasi

Pada Bab IV ini berisikan pengujian pada game yang dibuat menggunakan UAT, implementasi algoritma dan hasil akhir dari game.

BAB V Kesimpulan Dan Saran

Bab V berisikan kesimpulan yang didapat dari uji coba pada game yang telah dibuat dan saran untuk pengembangan game yang dibuat berikutnya.