

# PERANCANGAN GAME 3D MR. EGGY SIDE-SCROLLING BERBASIS ANDROID DENGAN MENGGUNAKAN METODE COLLISION DETECTION

Denny<sup>1</sup>, Tony Darmanto<sup>2</sup>, Kartono.<sup>3</sup>

<sup>123</sup>Informatika, Fakultas Teknologi Informatika Universitas Widya Dharma Pontianak  
e-mail: <sup>1</sup>dennyli777@gmail.com, <sup>2</sup>tony.darmanto@gmail.com, <sup>3</sup>kartono@gmail.com

## *Abstract*

*The development of game applications on touch screen Android smartphone devices makes developers have to make interesting games with simple gameplay. Even though games on smartphones currently have simple gameplay, they still have a large number of players due to the supportive and popular genre of games. One genre of game that is quite popular with simple gameplay is the side-scrolling genre. Side-scrolling game which is a game where the main character in the game must pass through obstacles and enemies that are facing. So that in this research Collision Detection method will be applied to enemy programming so that enemy characters can detect collisions on walls or ravines and are made with Godot game engine.*

**Keywords:** Game 3D, Godot Engine, Android

## **Abstrak**

Perkembangan aplikasi *game* pada perangkat *smartphone* Android layar sentuh membuat para pengembang harus membuat *game* yang menarik dengan *gameplay* yang sederhana. Meskipun *game* pada *smartphone* saat ini memiliki *gameplay* yang sederhana tetap saja memiliki jumlah pemain yang banyak dikarenakan *genre game* yang mendukung dan populer. Salah satu *genre game* yang cukup populer dengan *gameplay* yang sederhana adalah *genre side-scrolling*. *Game side-scrolling* yang merupakan *game* dimana karakter utama dalam *game* harus melewati rintangan dan musuh yang menghadang. Sehingga dalam penelitian ini akan diterapkan metode *Collision Detection* pada pemrograman musuh agar karakter musuh dapat mendeteksi tabrakan pada tembok ataupun jurang dan dibuat dengan *game engine* Godot.

**Kata Kunci:** Permainan 3D, Godot Engine, Android

## **1. PENDAHULUAN**

Seiring dengan perkembangan teknologi perangkat *digital*, saat ini hiburan yang paling digemari adalah *video game* yang merupakan salah satu perkembangan teknologi yang sangat pesat. *Game* sendiri saat ini tidak hanya digunakan sebagai media hiburan semata, dilihat dari nilai positif dari sebuah *game*, *game* dapat meningkatkan ketangkasan, melatih pemecahan masalah, dan melatih perkembangan otak.

Industri *video game* terus berkembang seiring dengan perkembangan teknologi. Hal ini ditunjukkan dengan variasi *game* saat ini yang tidak lagi terbatas pada peralatan *game* khusus (*Console*), seperti *Playstation*, *XBOX* dan *Nintendo*. Salah satu perangkat tersebut adalah *smartphone* berbasis *Android*. Perkembangan *smartphone* begitu pesat sehingga mampu menyandingi *Console* dalam industri *game*.

Karena pengguna *smartphone* yang sangat banyak dan dengan minat *game* yang juga sangat beragam maka *game* yang hadir untuk *smartphone* pun sangat beragam mulai dari *game racing*, *RPG*, *shooter*, *puzzle* dan *side-scrolling*. Salah satu *genre* permainan yang masih populer selama puluhan tahun adalah *game* ber-genre *side-scrolling* yang dapat diketahui dari ciri khas dari *game*-nya dimana kamera dari *game* akan mengikuti sang karakter utama berjalan ke kiri ataupun ke kanan.

Untuk membuat sebuah *game* diperlukan sebuah aplikasi yang bertujuan khusus dalam pembuatan *game* yaitu *game engine*. Salah satu *game engine* yang gratis adalah *game engine* Godot. Godot merupakan sebuah *game engine* berbasis *Python*.

## **2. METODOLOGI PENELITIAN**

2.1 Rancangan Penelitian, Metode Pengumpulan Data, Teknik Analisis Sistem, Teknik Perancangan Aplikasi.

2.1.1 Rancangan Penelitian

Rancangan penelitian yang digunakan adalah metode *blackbox*. Pengujian dilakukan dengan cara implementasi aplikasi ke *smartphone Android* dan memastikan setiap *scene* dapat berjalan dengan baik.

### 2.1.2 Metode Pengumpulan Data

Metode pengumpulan data yang dilakukan adalah dengan mengumpulkan informasi dan data dari buku-buku ilmiah, karya ilmiah, jurnal ilmiah, skripsi, dan sumber-sumber tertulis yang dipublikasikan di berbagai media. Informasi dan data dapat berupa teori-teori yang mendasari masalah dan bidang yang akan diteliti oleh penulis.

### 2.1.3 Teknik Analisis Sistem

Teknik analisis sistem *Unified Modeling Language (UML)* digunakan untuk menentukan, memvisualisasikan, membangun dan mendokumentasikan perancangan aplikasi *game*.

### 2.1.4 Teknik Perancangan Aplikasi

Teknik perancangan aplikasi yang digunakan dalam membangun *game* ini adalah menggunakan *game engine Godot*, bahasa pemrograman yang digunakan adalah Bahasa pemrograman *Python* dan aplikasi *Blender* untuk perancangan model 3D.

## 2.2 Landasan Teori

### 2.2.1 Video Game

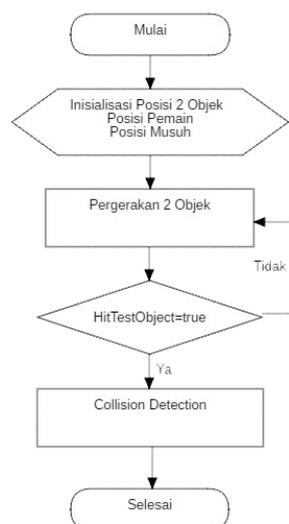
*Video game* hari ini merupakan konvergensi dari tranjectories substansial berbeda dari perkembangan teknologi menyediakan bentuk hiburan yang berbeda dengan audiens dengan kebutuhan yang berbeda. Hasilnya adalah media yang sangat beragam dalam fungsi, konten, dan khalayaknya - begitu beragam, pada kenyataannya, seperti burung dan kelelawar atau lumba-lumba dan ikan, banyak karakteristik sama di antara beberapa karakteristik *video game* yang mungkin hanya mirip saja. <sup>[1]</sup> Semua aktifitas permainan yang terpisah dari kehidupan sehari-hari oleh "batas-batas yang ditetapkan oleh tindakan bermain". <sup>[2]</sup>

### 2.2.2 Side-Scrolling

*Side-scrolling* sebetulnya lebih mengacu pada *format perspektif* ketimbang jenis *genre video game*, namun karena formulanya populer selama puluhan tahun hingga *game action* dapat dihidangkan secara *3D*, istilah ini seringkali dipakai untuk mendeskripsikan *game Metroidvania* dan sejumlah *platformer 2D*. Apakah salah jika ia digunakan sebagai kata ganti kedua *genre* itu? Tidak juga. Syarat masuk ke kategori *side-scrolling* adalah penyajian dua dimensi di mana layar akan mengikuti arah gerakan pemain, baik ke kiri maupun ke kanan atau dari bawah ke atas. Dengan begitu, ada banyak sekali *game* yang bisa dianggap *side-scrolling*, dari mulai *Bomber buatan Sega*, *Super Mario Bros* sampai *Excitebike* buatan *Nintendo*. <sup>[3]</sup> Permainan *side-scrolling* menghadirkan dunia sebagai tampilan tegak lurus terhadap arah yang dihadapi karakter di layar. Dengan fokus yang berat pada gerakan lateral, tujuan seringkali dipenuhi dengan berpindah dari satu ujung panggung ke ujung lainnya. <sup>[4]</sup>

### 2.2.3 Collision Detection

*Collision detection* merupakan teknik deteksi tabrakan untuk mengetahui obyek-obyek apa saja yang bersentuhan dalam bidang koordinat tertentu. <sup>[5]</sup> *Collision Detection* adalah sebuah metode pendeteksian ketika dua objek atau lebih yang bertabrakan dan akan menimbulkan sebuah reaksi, metode ini sangat banyak digunakan dalam dunia pembuatan animasi maupun pembuatan *game*. <sup>[6]</sup>



Gambar 1. Collision Detection

### 2.2.4 Mobile Computing

*Mobile computing* dapat didefinisikan sebagai sebuah lingkungan komputasi tentang mobilitas fisik. *User* dari lingkungan *mobile computing* akan dapat mengakses informasi data atau objek logika lainnya dari perangkat apapun dalam semua jaringan ketika sedang bergerak. Sistem *mobile computing* memungkinkan *user*

melakukan pekerjaan dari mana saja menggunakan perangkat komputasi publik (*web*), perusahaan (informasi bisnis) dan ruang informasi rekam medis pribadi, buku alamat. <sup>[7]</sup> *Mobile computing* berfokus pada mobilitas perangkat dan jaringan yang mempertimbangkan kesadaran konteks dan sumber daya *mobile* atau data akses. Aplikasi-aplikasi *mobile computing* biasanya mengandalkan pembuatan perangkat *mobile*, akses, proses, penyimpanan, dan menyampaikan informasi tanpa dibatasi pada satu lokasi. <sup>[8]</sup>

### 2.2.5 Android

*Android* adalah sistem operasi berbasis *Linux* yang dirancang untuk perangkat selular layer sentuh seperti telepon pintar dan komputer tablet. *Android* awalnya dikembangkan oleh *Android, Inc.*, dengan dukungan finansial dari Google yang kemudian membelinya pada tahun 2005. <sup>[9]</sup> *Android* secara sederhana bisa diartikan sebagai sebuah *software* yang digunakan pada perangkat *mobile* yang mencakup sistem operasi, *middleware*, dan aplikasi kunci yang dirilis oleh Google. <sup>[10]</sup>

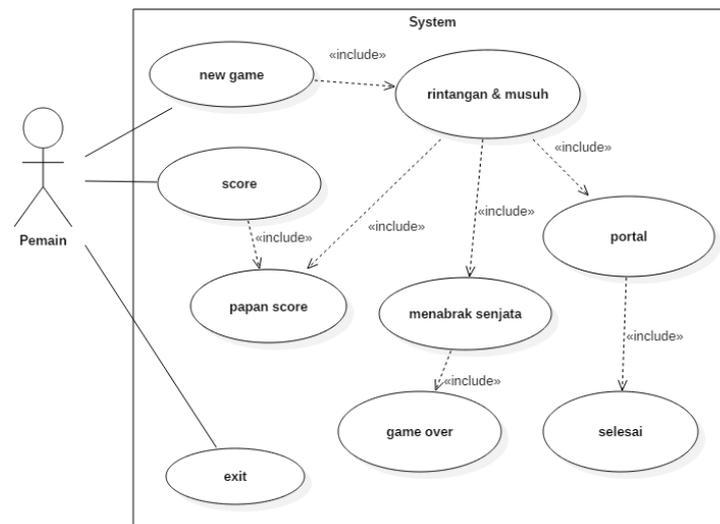
### 2.2.6 Godot Engine

*Godot* adalah *game engine* moderen yang penuh fitur, menyediakan berbagai fitur yang dijelaskan dibagian sebelumnya dan banyak lagi. Ini benar-benar gratis dan *open source*, rilis dibawah lisensi MIT yang sangat permisif. Ini berarti tidak ada biaya, tidak ada biaya yang tersembunyi, dan tidak ada pembayaran *royalty* untuk didapatkan *game* anda. Semua yang anda buat dengan *godot* 100% milik anda, dimana tidak seperti kebanyakan *game engine* komersial yang mensyaratkan hubungan kontrak yang berkelanjutan. Untuk beberapa pengembang, ini sangat menarik. <sup>[11]</sup> *Godot Engine* penuh dengan fitur, *game engine* lintas *platform* untuk membuat *game 2D* dan *3D* dengan antar muka terpadu. Ini menyediakan seperangkat alat umum yang komprehensif, jadi pengguna bisa fokus pada pembuatan *game* tanpa harus mencari alurnya. *Game* dapat di ekspor dalam satu klik ke sejumlah *platform*, termasuk *platform* besar seperti desktop (*Linux, macOS, Windows*) dan juga *mobile* (*Android, iOS*) dan *platform* basis web (*HTML5*). <sup>[12]</sup>

## 3. HASIL DAN PEMBAHASAN

### 3.1 Perancangan Unified Modeling Language (UML)

#### 3.1.1 Diagram Use Case Aplikasi Mr Eggy



Gambar 2. Diagram Use Case

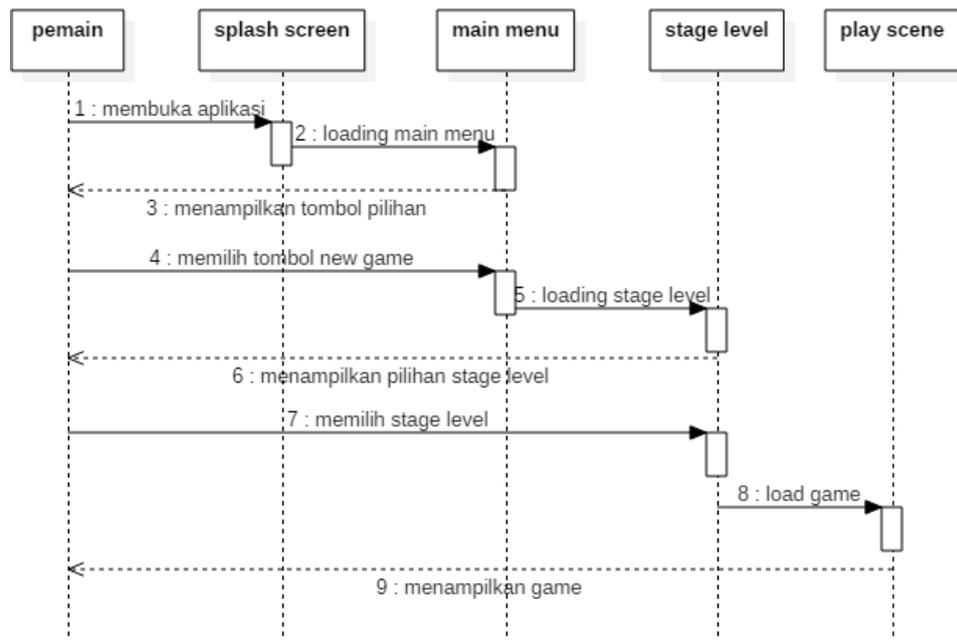
Use Case Aplikasi akan menjelaskan proses interaksi yang terjadi dalam pengoperasian aplikasi. Actor merupakan player yang akan menggunakan aplikasi. Diagram use case aplikasi Mr Eggy dapat dilihat pada Gambar 2.

#### 3.1.2 Gambaran Umum Rancangan Aplikasi Dengan Diagram Sekuensial

Diagram sekuensial bertujuan untuk menggambarkan proses dan interaksi yang terjadi pada aplikasi *game* Mr Eggy serta alur kerja dari aplikasi tersebut dari awal proses sampai akhir dari prosesnya. Berikut ini adalah diagram sekuensial dari aplikasi.

##### 3.1.2.1 Diagram Sekuensial Memulai Permainan

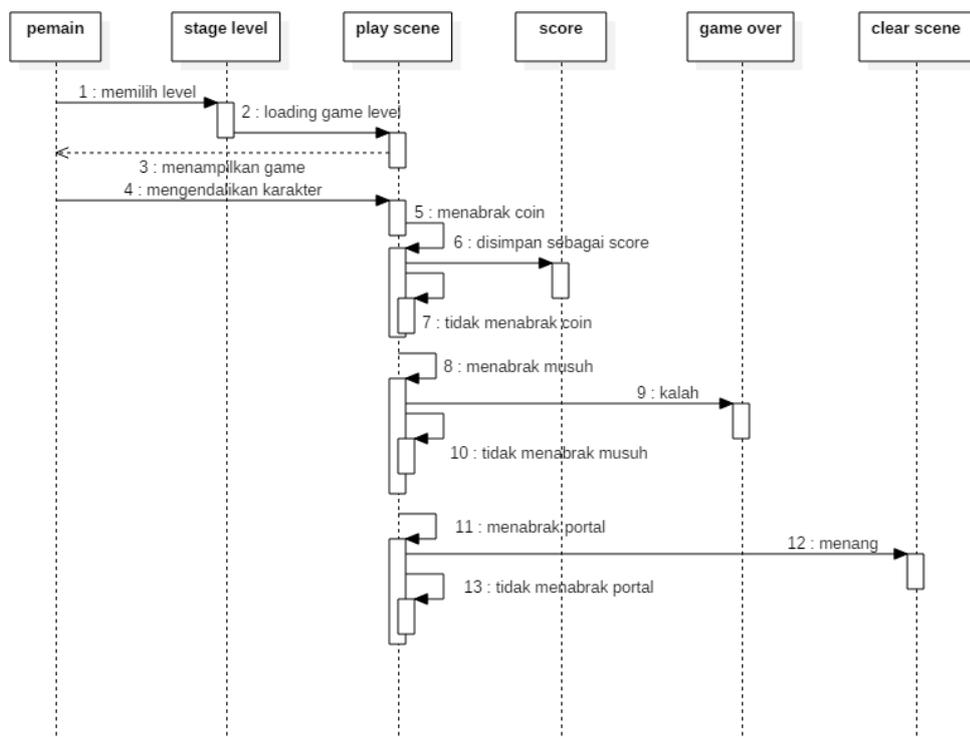
Pada diagram sekuensial di gambar 3 dapat dilihat bahwa ketika pemain membuka aplikasi *game* akan menampilkan *Splash screen* yang kemudian akan menampilkan *Main menu*, *Stage level scene* akan tampil ketika pemain menekan tombol *New game* pada *Main menu*. Pada stage level terdapat pilihan *level* dan *Play scene* akan dipanggil apabila pemain telah memilih *level* yang ingin dimainkan.



Gambar 3. Diagram Sekuensial Memulai Permainan

### 3.1.2.2 Diagram Sekuensial Collision Detection Pada Karakter Utama

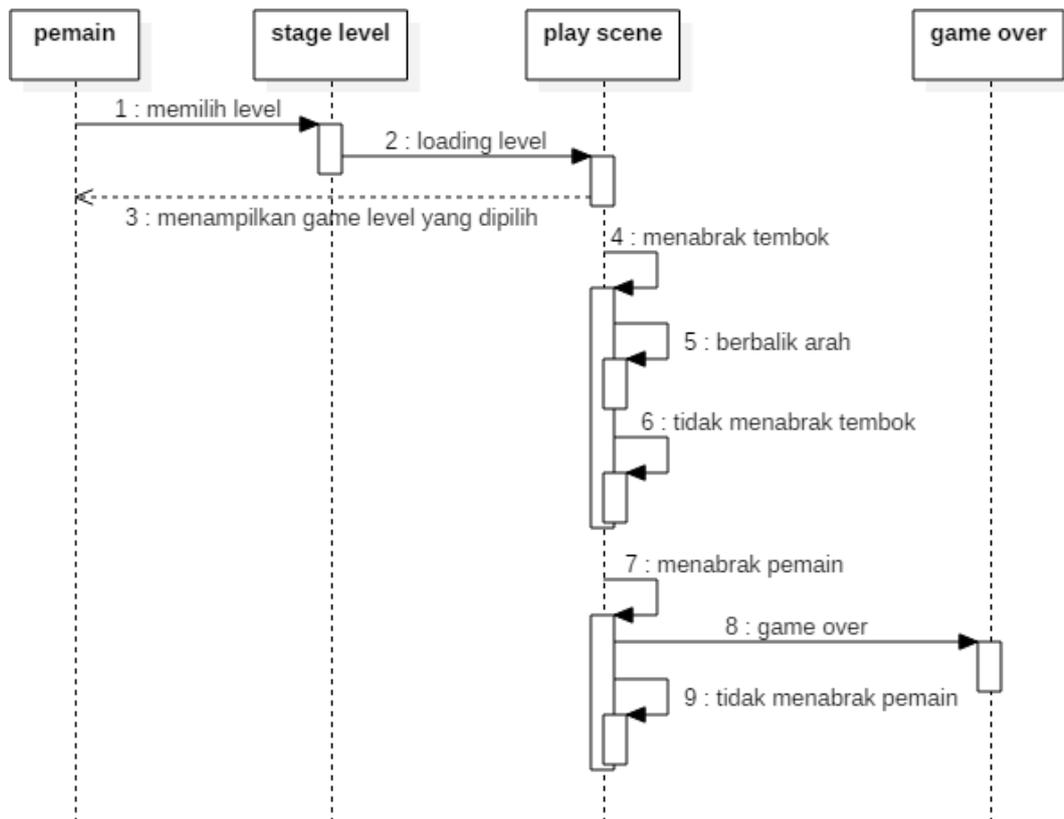
Diagram sekuensial pada gambar 4 menggambarkan bagaimana pendeteksian tabrakan yang terjadi pada karakter utama yang dimainkan, agar sebuah tabrakan dapat terjadi maka pada karakter dengan benda maka pada masing-masing benda harus ditambahkan *type scene collisionshape* yang merupakan bentuk padat dari benda. Jika karakter utama bersentuhan atau menabrak benda seperti *coin* maka nilai *counter* akan bertambah satu, jika karakter menabrak senjata musuh maka scene akan berganti menjadi *Game over scene*, dan jika karakter menabrak *portal* maka scene akan diganti menjadi *Clear stage scene*.



Gambar 4. Diagram Sekuensial Collision Detection Pada Karakter Utama

### 3.1.2.3 Diagram Sekuensial Collision Detection Pada Karakter Musuh

Diagram sekuensial pada gambar 5 terlihat bahwa musuh akan bergerak lurus hingga saat musuh mendeteksi tabrakan dengan tembok ataupun jurang maka musuh akan berbalik arah, untuk pendeteksian tabrakan pada karakter musuh, harus ditambahkan *type scene raycast* pada bagian depan dan bawah sehingga musuh dapat mendeteksi tembok dan jurang. Pada senjata yang dibawa oleh musuh diberikan *type scene collisionshape* sehingga apa bila karakter utama menabrak senjata musuh maka scene akan diganti dengan *Game over scene*.



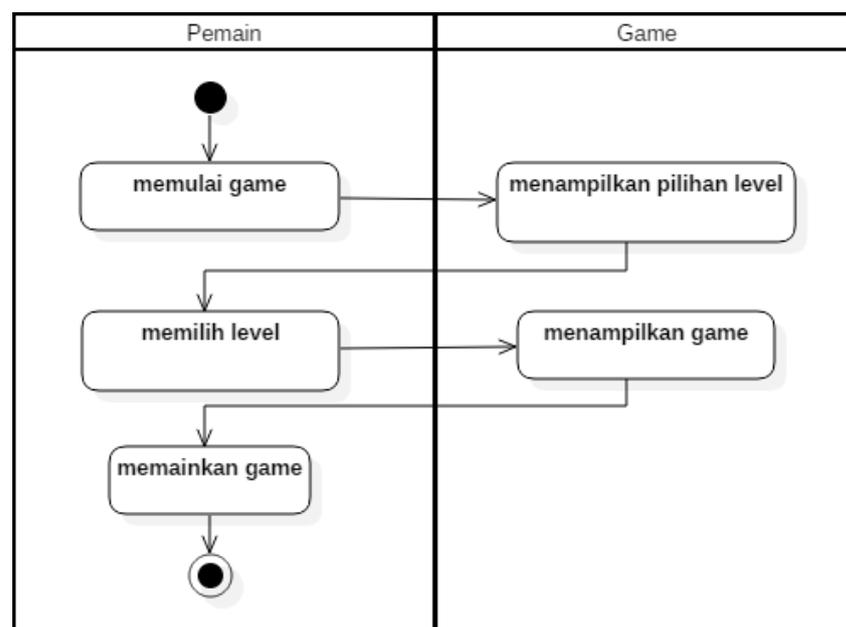
Gambar 5. Diagram Sekuensial Collision Detection Pada Karakter Musuh

### 3.1.3 Gambaran Umum Rancangan Aplikasi Dengan Diagram Aktifitas

Diagram aktifitas bertujuan untuk menggambarkan proses dan interaksi yang terjadi pada aplikasi *game* Mr Eggy serta alur kerja dari aplikasi tersebut dari awal proses sampai akhir dari prosesnya. Berikut ini adalah diagram aktifitas dari aplikasi.

#### 3.1.3.1 Diagram Aktifitas Memulai Permainan

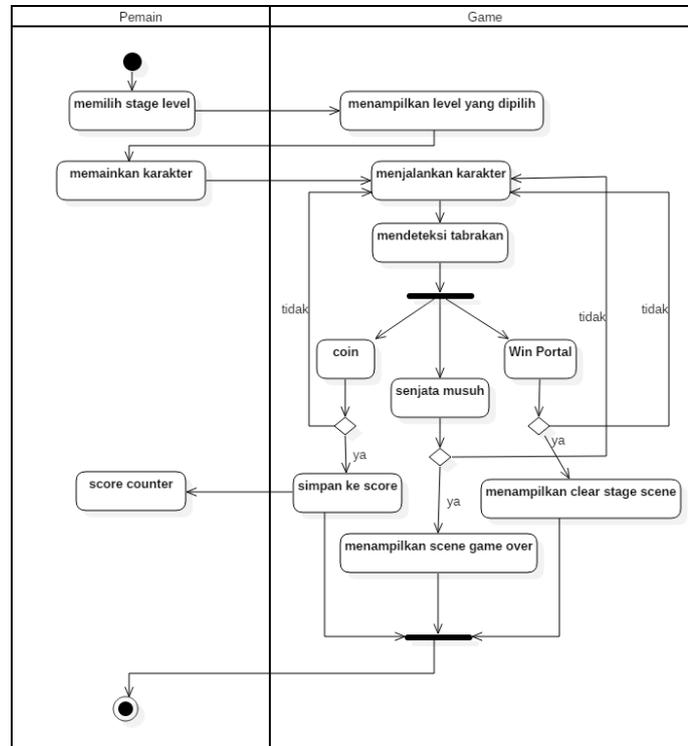
Diagram aktifitas pada gambar 6 menggambarkan bagaimana pendeteksian tabrakan yang terjadi pada karakter utama yang dimainkan, agar sebuah tabrakan dapat terjadi maka pada karakter dengan benda maka pada masing-masing benda harus ditambahkan *type scene collisionshape* yang merupakan bentuk padat dari benda. Jika karakter utama bersentuhan atau menabrak benda seperti *coin* maka nilai *counter* akan bertambah satu, jika karakter menabrak senjata musuh maka *scene* akan berganti menjadi *Game over scene*, dan jika karakter menabrak *portal* maka *scene* akan diganti menjadi *Clear stage scene*.



6. Diagram Aktifitas Memulai Permainan

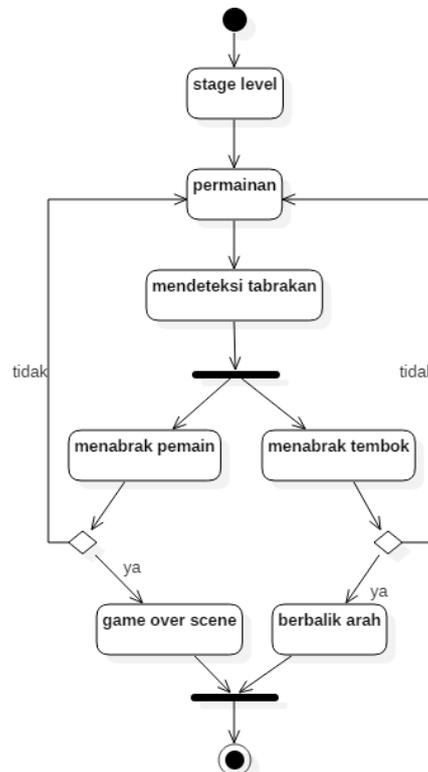
#### 3.1.3.2 Diagram Aktifitas Collision Detection Pada Karakter Utama

Diagram sekuensial pada gambar 7 menggambarkan bagaimana pendeteksian tabrakan yang terjadi pada karakter utama yang dimainkan, agar sebuah tabrakan dapat terjadi maka pada karakter dengan benda maka pada masing-masing benda harus ditambahkan *type scene collisionshape* yang merupakan bentuk padat dari benda. Jika karakter utama bersentuhan atau menabrak benda seperti *coin* maka nilai *counter* akan bertambah satu, jika karakter menabrak senjata musuh maka scene akan berganti menjadi *Game over scene*, dan jika karakter menabrak *portal* maka scene akan diganti menjadi *Clear stage scene*.



7. Diagram Aktifitas Collision Detection Pada Karakter Utama

### 3.1.3.3 Diagram Aktifitas Collision Detection Pada Karakter Musuh



Gambar 8. Diagram Aktifitas Collision Detection Pada Karakter Musuh

Diagram sekuensial pada gambar 8 terlihat bahwa musuh akan bergerak lurus hingga saat musuh mendeteksi tabrakan dengan tembok ataupun jurang maka musuh akan berbalik arah, untuk pendeteksian tabrakan pada karakter musuh, harus ditambahkan *type scene raycast* pada bagian depan dan bawah sehingga musuh dapat mendeteksi tembok dan jurang. Pada senjata yang dibawa oleh musuh diberikan *type scene collisionshape* sehingga apa bila karakter utama menabrak senjata musuh maka *scene* akan diganti dengan *Game over scene*.

### 3.2 Tampilan Aplikasi

Berikut ini adalah tampilan-tampilan halaman yang terdapat pada aplikasi game "Mr Eggy".

#### 3.2.1 Tampilan Menu Utama Pada Aplikasi

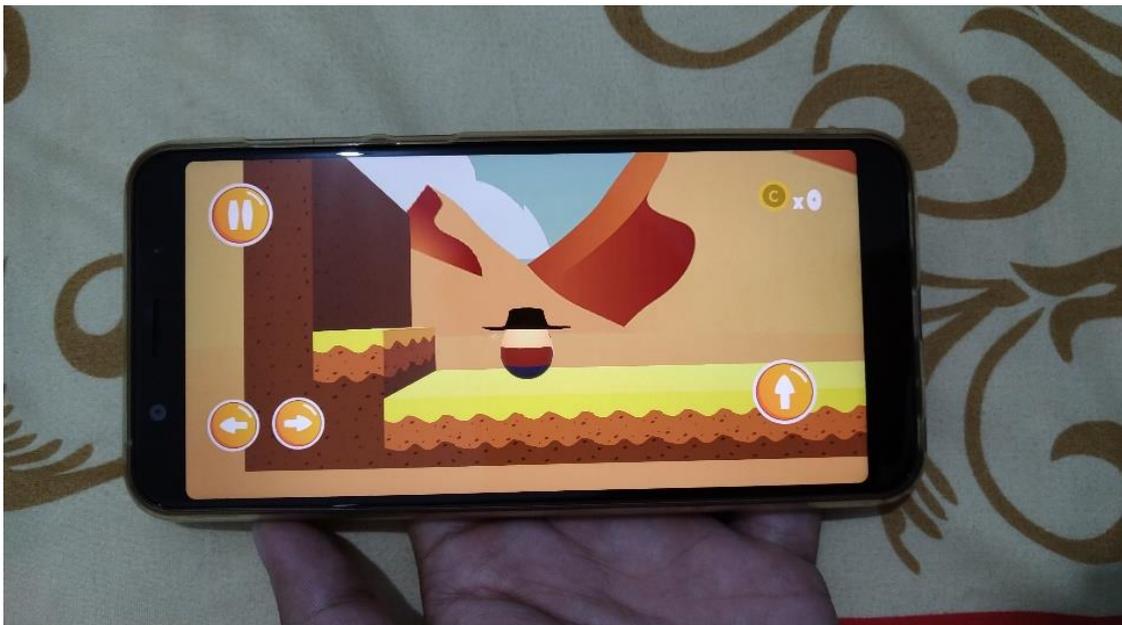
Pada saat pemain membuka aplikasi, maka aplikasi akan menampilkan *Splash screen* yang jika disentuh maka akan berganti *scene* menjadi *Main menu scene* (gambar 9), terdapat tiga tombol pilihan yaitu *New game*, *Score*, dan *Quit*. Untuk memulai permainan, pemain dapat menyentuh tombol *New game* maka pemain akan diberikan empat pilihan *level* untuk dimainkan. Untuk melihat *score* tertinggi yang pernah dicapai, pemain dapat menyentuh tombol *Score*, maka aplikasi akan menampilkan *Score scene*, dan jika pemain ingin keluar dari aplikasi maka pemain tinggal menyentuh tombol *Quit*.



Gambar 9. Tampilan Menu Utama Pada Aplikasi

#### 3.2.2 Tampilan dalam Permainan

Setelah pemain menyentuh tombol *New game* dan memilih *level*, maka yang ditampilkan pada layar *smartphone* pemain adalah *Play scene* (gambar 10), *game* mengambil latar gurun pasir dengan berbagai rintangan dan musuh, pemain harus melewati rintangan hingga sampai ke ujung.



Gambar 10. Tampilan dalam Permainan

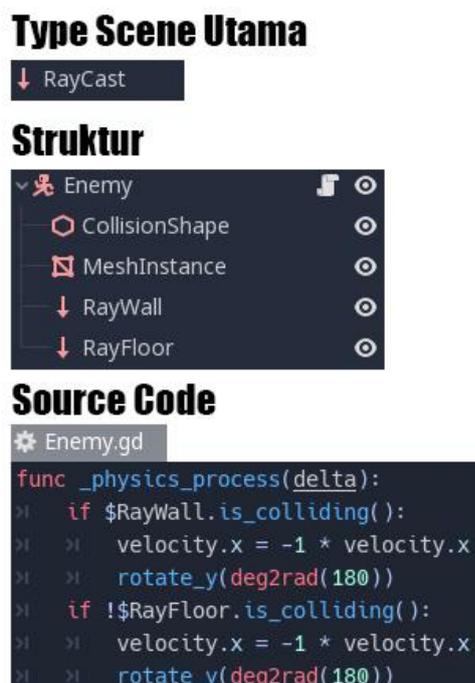
### 3.2.3 Signal dan Source Code dari Collision Detection pada Karakter Utama

Penerapan metode *collision detection* dalam *game* ini memerlukan beberapa fungsi yang tersedia dalam *game engine godot*, yaitu *signal* yang berfungsi untuk memberikan perintah kepada karakter apabila interaksi yang diinginkan terjadi. Untuk karakter utama, agar tabrakan dapat terdeteksi maka diperlukan *signal body\_entered* dan benda-benda yang memberikan *signal* kepada karakter utama, seperti senjata musuh, *coin* dan *portal* harus menggunakan *type scene* utamanya haruslah *type scene area*, agar badan dari karakter utama dapat menembus benda tersebut sehingga akan memicu *signal body\_entered* untuk bekerja sehingga pemain akan dinyatakan *game over*.



Gambar 11. Langkah Penerapan Collision Detection

### 3.2.4 Penerapan Collision Detection pada Musuh



Gambar 12. Struktur dan Source Code dari Collision Detection pada karkater musuh

Pada bagian ini, yang akan dibahas adalah karakter musuh yang melakukan penerapan metode *collision detection* adalah agar karakter musuh dapat mengetahui bentuk dari tempat musuh itu diletakkan dapat dilihat pada gambar 12. Pada struktur dari pembuatan karakter musuh, ditambahkan *type scene raycast* yang ditambahkan pada bagian depan karakter musuh, terdapat dua *raycast* yang dipasangkan pada karakter musuh, satu menghadap ke depan dan satunya lagi menghadap ke bawah. Untuk membedakan *raycast* mana yang dipanggil maka penulis memberi nama pada *raycast* yang menghadap ke depan dengan nama *raywall* dan kondisi yang digunakan adalah *if \$RayWall.is\_colliding*, dimana perintah akan dieksekusi apabila *raywall* mendeteksi tabrakan dengan tembok dan karakter musuh akan berbalik arah. Untuk *raycast* yang menghadap ke bawah penulis memberikan nama *rayfloor* dan kondisi yang digunakan adalah *if !\$RayFloor.is\_colliding*, dimana perintah akan dieksekusi apabila *rayfloor* tidak lagi mendeteksi tabrakan dengan lantai dan karakter musuh akan berbalik arah.

### 3.2.5 Mekanisme Permainan

Pada saat permainan dimulai, pemain hanya bisa menggerakkan karakter utama ke kiri, ke kanan dan ke atas. Musuh yang ada pada setiap *stage level* telah diatur sesuai tempat dan gerakannya sehingga pemain hanya perlu melewati setiap rintangan hingga mencapai *portal*, dan terdapat perbedaan pada *level boss*, dimana karakter *boss* disini akan mendeteksi keberadaan karakter utama sesuai luas *area* yang telah ditentukan oleh penulis.

## 3.3 Pengujian Aplikasi dengan Metode Black Box

Pengujian aplikasi *game "Mr Eggy"* menggunakan metode *Black Box Testing*. Penulis melakukan pengujian setelah aplikasi tersebut setelah aplikasi dipasang ke *smartphone android* penulis. Saat pengujian, penulis memainkan aplikasi *game* tersebut layaknya seorang *user* yang tidak pernah menggunakan aplikasi ini. Dengan begitu, pengujian terhadap aplikasi bisa memperlihatkan kesalahan dan kekurangan dalam rintangan yang akan dihadapi oleh *user*.

### 3.3.1 Pengujian Black Box Menu Utama

Skenario yang digunakan pada pengujian *menu* utama adalah menekan dua tombol bersamaan, sehingga hasil yang didapat yaitu program hanya mengikuti inputan tekan tombol yang paling pertama dan menjalankan perintah sesuai dengan tombol inputan.

### 3.3.2 Pengujian Black Box interaksi pergerakan pemain

Skenario yang digunakan pada pengujian interaksi pergerakan pemain adalah:

- a. Menyentuh kedua tombol kiri dan kanan secara bersamaan, sehingga tombol yang lebih dulu disentuh akan aktif duluan.
- b. Menyentuh tombol arah dan tombol lompat bersamaan, sehingga karakter akan melompat dan bergerak maju atau mundur sesuai arah.
- c. Menyentuh tombol *pause* saat permainan sedang berjalan, sehingga permainan akan berhenti atau membeku.

### 3.3.3 Pengujian Black Box melewati rintangan

Skenario yang digunakan pada pengujian melewati rintangan adalah memainkan semua *level* yang ada dan menguji apakah ada rintangan yang memiliki kesalahan sehingga tidak mungkin diselesaikan, dan hasil dari pengujian ini adalah semua rintangan pada setiap *level* dapat diselesaikan oleh penulis.

### 3.3.4 Pengujian Black Box interaksi pemain dengan objek

Skenario yang digunakan pada pengujian interaksi pemain dengan objek antara lain:

- a. Bergerak mendekati *coin*, sehingga hasil yang didapat yaitu *coin* yang didekati menghilang dan pada *score counter* nilai bertambah satu.
- b. Menabrak *portal*, sehingga hasil yang didapat pemain adalah menyelesaikan *level*.
- c. Terkena tombak musuh, sehingga hasil yang didapat pemain adalah pemain akan langsung kalah.
- d. Terkena peluru musuh, sehingga hasil yang didapat pemain adalah pemain akan langsung kalah.
- e. Mendorong *box* ke tempat atau posisi yang diinginkan agar bisa naik pada lantai yang tinggi, hasilnya *box* dapat bergerak ketika didorong.

### 3.3.5 Pengujian Black Box Collision Detection Musuh

Pada tahap ini, pengujian kali ini adalah mengamati pergerakan musuh, bagaimana musuh dapat mendeteksi tembok dan lantai. Sehingga musuh dapat menyesuaikan tempat diletakkannya musuh dan pengaplikasian metode *collision detection* dapat berjalan.

### 3.3.6 Pengujian Black Box Deteksi Lokasi Pada Karakter Boss

Pada tahap ini, penulis akan menguji pengaplikasian pendeteksian lokasi karakter utama oleh karakter *boss* pada jarak tertentu, sehingga hasil yang didapat adalah karakter *boss* dapat mengikuti atau mengejar karakter utama dan melewati rintangan pada *level* tersebut.

### 3.3.7 Pengujian Black Box Score Counter

Pada pengujian kali ini, penulis akan menguji perhitungan *score* yang didapat oleh pemain, apakah *score* akan terus berlanjut apabila sudah masuk ke *stage level* selanjutnya ataupun memainkan *stage level* sebelumnya dan apakah jika pemain *me-reset* permainan, *score counter*-nya akan kembali pada saat permainan dimulai, sehingga perhitungan untuk menentukan *best score* tidak akan terjadi kesalahan, dan pada perhitungan *best score*.

#### 4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan tentang aplikasi *game* Mr Eggy penulis kesimpulan sebagai berikut:

- a. *Game* Mr Eggy mudah untuk dimainkan karena *genre side-scrolling* yang memiliki sistem permainan sederhana dan hanya memiliki tiga tombol kontrol, sehingga *game* Mr Eggy dapat dimainkan oleh semua kalangan.
- b. *Game* Mr Eggy menggunakan gaya animasi kartun dengan tambahan animasi 3 dimensi pada tampilan *game*-nya sehingga menarik bagi kalangan anak-anak dan remaja yang menyukai tayangan animasi.
- c. Penerapan metode *Collision Detection* menjadikan musuh dalam *game* dapat bergerak sesuai dengan bidang yang terdeteksi oleh karakter musuh tersebut dimana musuh diletakkan.
- d. Pada *game* Mr Eggy, pemain dapat memainkan empat *stage level* yang berbeda.
- e. Berdasarkan hasil pengujian pada *smartphone*, *game* ini dapat dijalankan pada *android* versi *Oreo* dengan RAM 3GB dan tidak dapat dijalankan pada *smartphone* lama yang berarti *game* Mr Eggy memiliki performa tidak optimal untuk *smartphone android*.

#### 5. SARAN

Berikut ini adalah beberapa saran dari penulis bagi pembaca agar pembaca dapat pertimbangan dalam mengembangkan *game* ini agar menjadi lebih baik:

- a. Tambahkan *genre game* seperti *puzzle*, *action* dan *adventure* sehingga *gameplay* dari *game* Mr Eggy menjadi lebih rumit sehingga pemain lebih tertantang.
- b. Pengoptimalan performa dari *game* agar *game* dapat dijalankan pada *smartphone android* yang memiliki versi dan spesifikasi rendah, seperti *smartphone android* versi *ice cream* dan RAM 1GB.
- c. Pembuatan animasi karakter yang lebih baik seperti memiliki tangan dan kaki untuk bergerak sehingga *game* akan semakin menarik untuk dimainkan.
- d. Menambah jumlah *stage level* sehingga pemain memiliki banyak pilihan *stage level*.
- e. Algoritma pada karakter musuh tidak hanya terbatas pada penerapan metode *collision detection*, dapat juga ditambahkan algoritma pergerakan lain sehingga karakter musuh tidak monoton.

#### UCAPAN TERIMA KASIH

Dalam penulisan ini, penulis telah banyak mendapatkan bantuan berupa bimbingan, petunjuk, data, saran, maupun dorongan moral dari berbagai pihak, maka pada kesempatan ini penulis mengucapkan terima kasih kepada civitas Fakultas Teknologi Informatika Universitas Widya Dharma Pontianak, serta kepada keluarga tercinta yang telah banyak memberikan kasih sayang, bantuan, dan dukungan serta doa kepada penulis selama menjalani studi dari awal perkuliahan hingga selesainya penelitian ini.

#### DAFTAR PUSTAKA

- [1] Berger, Kowert dan Quandt (2015). *The Video Game Debate: Unravelling the Physical, Social, and Psychological Effects of Video Games*. [https://books.google.co.id/books?id=z-FzCgAAQBAJ&dq=Rachel+Kowert&hl=id&source=gbs\\_navlinks\\_s](https://books.google.co.id/books?id=z-FzCgAAQBAJ&dq=Rachel+Kowert&hl=id&source=gbs_navlinks_s), 10 Maret 2020, 07:08 WIB
- [2] Fassone, Riccardo (2017). *Every Game is an Island: Endings and Extremities in Video Games*. <https://books.google.co.id/books?id=p43DDQAAQBAJ&printsec=frontcover&dq=video+game+is&hl=id&sa=X&ved=0ahUKEwiHu93d5I7oAhW0guYKHbXWA8MQ6AEIKTAA#v=onepage&q=video%20game%20is&f=false>, 10 Maret 2020, 08:58 WIB
- [3] Baskoro, Wiku (2016). *Daftar Genre Video Game Beserta Penjelasannya*. <https://dailysocial.id/post/jenis-genre-game>. 11 Maret 2020, 00:18 WIB.
- [4] Giantbomb, <https://www.giantbomb.com/side-scrolling/3015-299/>, 11 Maret 2020, 00.33 WIB.
- [5] Lui Haekal Fasha, Fauziah, dan M. Gufroni (2018). *Implementasi Algoritma Collision Detection Pada Game Simulator Driving Car*. Jurusan Teknik Informatika, Fakultas Teknologi Komunikasi dan Informatika. Universitas Nasional.
- [6] Nurdianto, Arif dan Winarno Edy (2018). *Penerapan Metode Collision Detection Pada Game Petualangan Menggunakan Aksara Jawa*. Jurusan Teknik Informatika, Fakultas Teknologi Informasi. Universitas Stikubank.
- [7] Asoke K. Talukdar (2010). *Mobile Computing*. [https://books.google.nl/books?id=4QUiBAAAQBAJ&dq=mobile+computing&source=gbs\\_navlinks\\_s](https://books.google.nl/books?id=4QUiBAAAQBAJ&dq=mobile+computing&source=gbs_navlinks_s), 11 Maret 2020, 09:40 WIB
- [8] Dijiang, Huijun (2017). *Mobile Cloud Computing: Foundations and Service Models*. [https://books.google.co.id/books?id=dupGDgAAQBAJ&dq=0128096446&hl=id&source=gbs\\_navlinks\\_s](https://books.google.co.id/books?id=dupGDgAAQBAJ&dq=0128096446&hl=id&source=gbs_navlinks_s), 11 Maret 2020, 09:45 WIB

- [9] Yudhanto, Yudha dan Wijayanto, Ardhi (2017). *Mudah Membuat dan Berbisnis Aplikasi Android dengan Android Studio*. Penerbit PT Elex Media Komputindo. Jakarta.
- [10] EMS, Tim (2015). *Pemrograman Android dalam Sehari*. Penerbit PT Elex Media Komputindo. Jakarta.
- [11] Bradfield, Chris (2018). *Godot Engine Game Development Projects : Build five cross-platform 2D and 3D with Godot 3.0*. Penerbit Packt Publishing Ltd. Birmingham.
- [12] Linietsky, Juan., Manzur, Ariel dan the Godot community (2014-2019). *Godot Doc*. <https://docs.godotengine.org/en/3.1/about/introduction.html#about-godot-engine>. Diakses pada tanggal 17 Maret 2020, 15:50 WIB.