

**IMPLEMENTASI ALGORITMA COLLISION DETECTION PADA  
GAME TPS BERBASIS ANDROID MENGGUNAKAN UNITY 3D**

**SKRIPSI**



Andrian  
1511500108

**PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
ATMA LUHUR  
PANGKALPINANG  
2019**

**IMPLEMENTASI ALGORITMA COLLISION DETECTION PADA  
GAME TPS BERBASIS ANDROID MENGGUNAKAN UNITY 3D**

**SKRIPSI**

**Diajukan Untuk Melengkapi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer**



Andrian  
1511500108

**PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
ATMA LUHUR  
PANGKALPINANG  
2019**

LEMBAR PENGESAHAN SKRIPSI

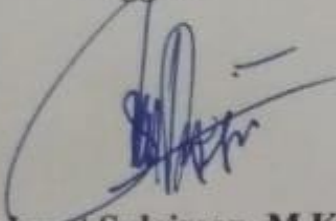
IMPLEMENTASI ALGORITMA COLLISION DETECTION PADA GAME TPS  
BERBASIS ANDROID MENGGUNAKAN UNITY 3D

Yang dipersiapkan dan disusun oleh

ANDRIAN  
1511500108

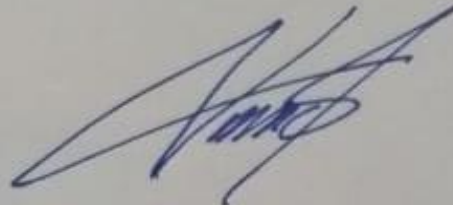
Telah dipertahankan di depan Dewan Penguji  
Pada Tanggal 01 Juli 2019

Susunan Dewan Penguji  
Anggota



Rahmat Sulaiman, M.Kom.  
NIDN.0208019401

Dosen Pembimbing



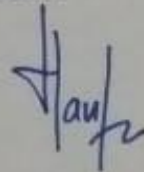
Lukas Tommy, M.Kom.  
NIDN.0215099201

Kaprodi Teknik Informatika



R. Burham Isnanto F., S.Si, M.Kom  
NIDN.0224048003

Ketua



Laurentinus, M.Kom  
NIDN.0201079201

Skripsi ini telah diterima dan sebagai salah satu persyaratan  
Untuk memperoleh gelar Sarjana Komputer  
Tanggal 01 Juli 2019

KETUA STMIK ATMA LUHUR PANGKALPINANG



Dr. Husni Teja Sukmana, S.T., M.Sc.

## LEMBAR PERNYATAAN

Yang bertanda tangan di bawah ini :

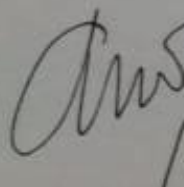
NIM : 1511500108

Nama : Andrian

Judul Skripsi : IMPLEMENTASI ALGORITMA COLLISION  
DETECTION PADA GAME TPS BERBASIS  
ANDROID MENGGUNAKAN UNITY 3D

Menyatakan bahwa laporan skripsi saya adalah hasil karya sendiri, tidak membeli, tidak membayar pihak lain untuk membuat, dan bukan plagiat. Apabila ternyata ditemukan didalam laporan skripsi saya terdapat unsur diatas, maka saya siap untuk mendapatkan sanksi akademik yang terkait dengan hal tersebut.

Pangkalpinang,.....

  
(Andrian)



## KATA PENGANTAR

Segala puji bagi Allah SWT yang Maha Pengasih lagi Maha Penyayang atas Rahmat dan Hidayah-Nya sehingga laporan skripsi yang merupakan salah satu persyaratan untuk menyelesaikan jenjang strata satu (S1) pada Program Studi Teknik Informatika STMIK Atma Luhur dapat diselesaikan.

Dengan segala keterbatasan, penulis menyadari pula bahwa laporan skripsi ini takkan terwujud tanpa bantuan, bimbingan, dan dorongan dari berbagai pihak. Untuk itu, dengan segala kerendahan hati, penulis menyampaikan ucapan terima kasih kepada:

1. Allah SWT yang telah memberikan kekuatan dan ketekunan untuk menyelesaikan laporan skripsi ini.
2. Orang tua tercinta yang senantiasa memberikan dukungan dan doa sehingga laporan skripsi ini dapat diselesaikan dengan baik.
3. Bapak Drs.Djaetun HS yang telah mendirikan Atma Luhur
4. Bapak Dr Husni Teja Sukama, S.T., M.Sc, selaku Ketua STMIK Atma Luhur.
5. Bapak R.Burham Isnanto Farid, S.Si, M.Kom selaku Kaprodi Teknik Informatika.
6. Lukas Tommy,M.Kom selaku dosen pembimbing.
7. Saudara dan saudari, serta sahabat-sahabatku terutama kawan-kawan angkatan 2015 yang telah memberikan dukungan moral untuk terus menyelesaikan skripsi ini.

Akhir kata penulis maaf apabila dalam penyusunan laporan skripsi ini masih terdapat kekurangan dan masih jauh dari sempurna. Karena itu, kritik dan saran yang membangun akan penulis terima dengan senang hati. Semoga karya yang sederhana ini dapat memberikan manfaat bagi pihak lain.

Pangkalpinang, 01 Juli 2019

Andrian



## ABSTRACT

*TPS Game (Third Person Shooter) is a genre of shooter games, where the player's characters are visible on the screen and the main gameplay consists of shooting, and this game is designed to run on the Android platform. The movement of enemy characters in a game is closely related to Artificial Intelligence (AI) artificial intelligence which makes the enemy as if alive. The collision detection algorithm is a method of detecting when two or more objects collide with each other and will cause a reaction. This method for designing a FPS game (First Person Shooter), when a character attacks another character will get a reaction to the character. The model, method, and software development tool used in this study are prototype models, respectively. From the results of this study it can be concluded that collision detection is very important in a game because without collision detection one object with another object will cross each other and collide with collision detection, the game looks more realistic.*

*Keywords : TPS, Android, Artificial Intelligence, collision detection.*



## ABSTRAK

*Game TPS (Third Person Shooter)* adalah *genre game shooter*, dimana karakter pemain yang terlihat di layar dan *gameplay* utamanya terdiri dari pengambilan gambar, dan *game* ini dirancang agar bisa berjalan di *platform* Android. Pergerakan karakter musuh di suatu *game* sangat erat kaitannya dengan kecerdasan buatan *Artificial Intelligence (AI)* yang membuat musuh seolah-olah hidup. Algoritma *collision detection* adalah sebuah metode pendeteksian ketika dua objek atau lebih yang saling bertabrakan dan akan menimbulkan sebuah reaksi. Metode ini untuk perancangan sebuah *game FPS (First Person Shooter)*, ketika karakter melakukan serangan ke karakter lain maka akan mendapatkan sebuah reaksi pada karakter tersebut model, metode, dan *tool* pengembangan perangkat lunak yang digunakan dalam penelitian ini berturut-turut adalah model *prototype*. Dari hasil penelitian ini dapat disimpulkan bahwa *collision detection* sangat penting dalam sebuah *game* karena tanpa adanya *collision detection* (deteksi tubrukan) maka satu objek dengan objek lainnya akan saling berpapasan dan bertubrukan dengan adanya *collision detection* maka *game* tampak lebih realistis.

Kata Kunci: TPS, Android, *Artificial Intelligence*, *collision detection*



## DAFTAR ISI

|   |     |
|---|-----|
| <b>LEMBAR PENGESAHAN</b> .....          | i   |
| <b>LEMBAR PERNYATAAN</b> .....          | ii  |
| <b>KATA PENGANTAR</b> .....             | iii |
| <b>ABSTRACT</b> .....                   | iv  |
| <b>ABSTRAK</b> .....                    | v   |
| <b>DAFTAR ISI</b> .....                 | vi  |
| <b>DAFTAR GAMBAR</b> .....              | ix  |
| <b>DAFTAR TABEL</b> .....               | x   |
| <b>DAFTAR SIMBOL</b> .....              | xi  |
| <br>                                    |     |
| <b>BAB I PENDAHULUAN</b>                |     |
| 1.1 Latar Belakang.....                 | 1   |
| 1.2 Rumusan Masalah .....               | 2   |
| 1.3 Batasan Masalah.....                | 2   |
| 1.4 Tujuan dan Manfaat Penelitian.....  | 3   |
| 1.4.1 Tujuan Penelitian.....            | 3   |
| 1.4.2 Manfaat Penelitian.....           | 3   |
| 1.5 Sistematika Penulisan.....          | 4   |
| <br>                                    |     |
| <b>BAB II LANDASAN TEORI</b>            |     |
| 2.1 Model <i>Prototype</i> .....        | 6   |
| 2.1.1 Keunggulan <i>Prototype</i> ..... | 7   |
| 2.1.2 Kelemahan <i>Prototype</i> .....  | 7   |
| 2.2 Metodologi .....                    | 8   |
| 2.3 UML .....                           | 9   |
| 2.3.1 Jenis-jenis UML .....             | 9   |
| 2.3.1.1 <i>Use Case Diagram</i> .....   | 9   |
| 2.3.1.2 <i>Activity Diagram</i> .....   | 10  |



|         |  |    |
|---------|--|----|
| 2.3.1.3 | <i>Sequence Diagram</i> .....              | 11 |
| 2.4     | Algoritma <i>Collision Detection</i> ..... | 12 |
| 2.5     | Aplikasi <i>Mobile</i> .....               | 13 |
| 2.6     | TPS( <i>Third Person Shooter</i> ).....    | 13 |
| 2.7     | Android.....                               | 13 |
| 2.9     | Unity 3D .....                             | 14 |
| 2.10    | <i>Blackbox Testing</i> .....              | 15 |

### **BAB III METODOLOGI PENELITIAN**

|       |  |    |
|-------|--|----|
| 3.1   | Model Pengembangan Perangkat Lunak .....   | 20 |
| 3.1.1 | Membangun <i>Prototyping</i> .....         | 20 |
| 3.1.2 | Evaluasi <i>Prototyping</i> .....          | 20 |
| 3.1.3 | Mengkodekan Perangkat Lunak.....           | 21 |
| 3.1.4 | Menguji Perangkat Lunak .....              | 21 |
| 3.1.5 | Evaluasi Perangkat Lunak .....             | 21 |
| 3.1.6 | Penggunaan Perangkat Lunak .....           | 21 |
| 3.2   | Metode Pengembangan Perangkat Lunak .....  | 21 |
| 3.3   | Alat Bantu Pengembangan Sistem .....       | 22 |
| 3.4   | Algoritma <i>Collision Detection</i> ..... | 22 |

### **BAB IV HASIL DAN PEMBAHASAN**

|         |                                |    |
|---------|--------------------------------|----|
| 4.1     | Analisa Masalah .....          | 25 |
| 4.2     | Analisa Kebutuhan.....         | 25 |
| 4.2.1   | Kebutuhan Fungsional.....      | 25 |
| 4.2.2   | Kebutuhan Non-Fungsional.....  | 26 |
| 4.2.3   | Rancang Sistem .....           | 26 |
| 4.2.3.1 | <i>Use Case Diagram</i> .....  | 27 |
| 4.2.3.2 | <i>Use Case Scenario</i> ..... | 28 |
| 4.2.3.3 | <i>Activity Diagram</i> .....  | 31 |
| 4.2.3.4 | <i>Sequence Diagram</i> .....  | 37 |

|                             |                                  |    |
|-----------------------------|----------------------------------|----|
| 4.3                         | Rancang Algoritma.....           | 43 |
| 4.4                         | Rancang Layar.....               | 44 |
| 4.4.1                       | Rancang Layar <i>User</i> .....  | 44 |
| 4.5                         | <i>Gameplay</i> .....            | 45 |
| 4.6                         | <i>MDA frame mekanika</i> .....  | 46 |
| 4.6.1                       | <i>Mechanics</i> .....           | 46 |
| 4.6.2                       | <i>Dynamic</i> .....             | 47 |
| 4.6.3                       | <i>Aesthetic</i> .....           | 47 |
| 4.7                         | Tampilan Layar.....              | 48 |
| 4.7.1                       | Tampilan Layar <i>Game</i> ..... | 48 |
| 4.7.2                       | Tampilan Karakter Bergerak.....  | 48 |
| 4.7.3                       | Tampilan Karakter Melompat.....  | 49 |
| 4.7.4                       | Tampilan Karakter Menembak.....  | 49 |
| 4.7.5                       | Tampilan Musuh Menyerang.....    | 50 |
| 4.8                         | Pengujian <i>Blackbox</i> .....  | 50 |
| <br><b>BAB V PENUTUP</b>    |                                  |    |
| 5.1                         | Kesimpulan.....                  | 52 |
| 5.2                         | Saran.....                       | 52 |
| <b>DAFTAR PUSTAKA</b> ..... |                                  | 54 |
| <b>LAMPIRAN</b> .....       |                                  | 56 |

## DAFTAR GAMBAR

|  |    |
|--|----|
| Gambar 2.1 Model <i>Prototype</i> .....                                | 6  |
| Gambar 2.2 <i>Use Case Diagram</i> .....                               | 10 |
| Gambar 2.3 <i>Activity Diagram</i> .....                               | 11 |
| Gambar 2.4 <i>Squence Diagram</i> .....                                | 12 |
| Gambar 2.5 <i>Flowchart</i> Algoritma <i>Collision Detection</i> ..... | 13 |
| Gambar 2.6 <i>Black box</i> .....                                      | 15 |
| Gambar 3.2 <i>Bounding Circle</i> .....                                | 23 |
| Gambar 3.3 <i>Bounding Circle</i> Pada Aktor <i>Game</i> .....         | 23 |
| Gambar 4.1 <i>Use Case Diagram</i> Pemain .....                        | 27 |
| Gambar 4.2 <i>Use Case Diagram</i> Bot.....                            | 27 |
| Gambar 4.3 <i>Activity Diagram</i> Gerak Karakter.....                 | 31 |
| Gambar 4.4 <i>Activity Diagram</i> Melompat.....                       | 32 |
| Gambar 4.5 <i>Activity Diagram</i> Menerima Serangan.....              | 32 |
| Gambar 4.6 <i>Activity Diagram</i> Menerima Serangan Musuh.....        | 33 |
| Gambar 4.7 <i>Activity Diagram</i> Mengejar.....                       | 34 |
| Gambar 4.8 <i>Activity Diagram</i> Menyerang Musuh.....                | 35 |
| Gambar 4.9 <i>Activity Diagram</i> Menyerang Pemain.....               | 36 |
| Gambar 4.10 <i>Activity Diagram</i> Patroli.....                       | 37 |
| Gambar 4.11 <i>Squence Diagram</i> Melompat.....                       | 38 |
| Gambar 4.12 <i>Squence Diagram</i> Menembak Musuh.....                 | 39 |
| Gambar 4.13 <i>Squence Diagram</i> Menerima Kerusakan.....             | 40 |
| Gambar 4.14 <i>Squence Diagram</i> Mengejar Musuh.....                 | 41 |
| Gambar 4.15 <i>Squence Diagram</i> Menggerakkan Pemain.....            | 42 |
| Gambar 4.16 <i>Squence Diagram</i> Patroli.....                        | 43 |
| Gambar 4.17 Rancang Layar <i>User</i> .....                            | 44 |
| Gambar 4.18 Tampilan Layar <i>Game</i> .....                           | 48 |
| Gambar 4.19 Tampilan Karakter Bergerak.....                            | 49 |
| Gambar 4.20 Tampilan Karakter Melompat.....                            | 49 |
| Gambar 4.21 Tampilan Karakter Menembak.....                            | 50 |
| Gambar 4.22 Tampilan Musuh Menyerang.....                              | 50 |

## DAFTAR TABEL

|   |    |
|---|----|
| Tabel 2.1 Ringkasan Penelitian Terdahulu .....                      | 18 |
| Tabel 4.1 Skenario <i>Use Case</i> Gerak Karakter.....              | 28 |
| Tabel 4.2 Skenario <i>Use Case</i> Menembak Musuh .....             | 28 |
| Tabel 4.3 Skenario <i>Use Case</i> Melompat.....                    | 29 |
| Tabel 4.4 Skenario <i>Use Case</i> Menerima Serangan .....          | 29 |
| Tabel 4.5 Skenario <i>Use Case</i> Berpatroli .....                 | 29 |
| Tabel 4.6 Skenario <i>Use Case</i> Menyerang Pemain .....           | 30 |
| Tabel 4.7 Skenario <i>Use Case</i> Mengejar Pemain.....             | 30 |
| Tabel 4.8 Skenario <i>Use Case</i> Menerima Kerusakan.....          | 30 |
| Tabel 4.9 Penjelasan Kemampuan <i>Enemy</i> dan <i>Player</i> ..... | 45 |
| Tabel 4.10 Hasil Pengujian <i>Blackbox</i> .....                    | 51 |



## DAFTAR SIMBOL

### 1. Activity Diagram



#### *Start Point*

Mengambarkan awal dari suatu aktivitas yang berjalan pada sistem.



#### *End Point*

Mengambarkan akhir dari suatu aktivitas yang berjalan pada sistem.



#### *Activity State*

Mengambarkan suatu proses / kegiatan bisnis.

NewSwimlane



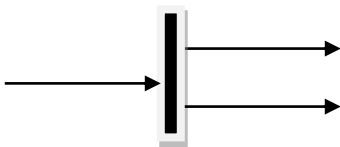
#### *Swimlane*

Mengambarkan pembagian / pengelompokkan berdasarkan tugas dan fungsi sendiri.



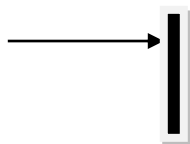
#### *Decision Points*

Mengambarkan pilihan untuk pengambilan keputusan, true atau false.



#### *Fork*

Mengambarkan aktivitas yang dimulai dengan sebuah aktivitas dan diikuti oleh dua atau lebih aktivitas yang harus dikerjakan.



#### *Join*

→ → Menggambarkan aktivitas yang dimulai dengan dua atau lebih aktivitas yang sudah dilakukan dan menghasilkan sebuah aktivitas.

[ ... ]

*Guards*

Sebuah kondisi benar sewaktu melewati sebuah transisi, harus konsisten dan tidak overlap.

*Transition*

→

Menggambarkan aliran perpindahan control antara state.

## 2. Use Case Diagram



*Actor*

Abstraksi dari orang atau sistem yang mengaktifkan fungsi dari use case.



*Use Case*

Menggambarkan proses sistem dari perpektif pengguna (user).

—————

*Relasi/Asosiasi*

Menggambarkan hubungan antara actor dengan use case.

<< include >>

----->

Asosiasi yang termasuk didalam *use case* lain, yang bersifat harus dilakukan bila *use case* lain



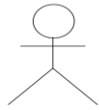
tersebut dilakukan.

<<extend>>

----->

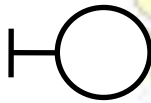
Perluasan dari *use case* lain jika kondisi atau syarat terpenuhi dan tidak harus dilakukan.

### 3. Sequence Diagram



*Actor*

Menggambarkan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem.



*Boundary*

Sebuah obyek yang menjadi penghubung antara user dengan sistem. Contohnya window, dialogue box atau screen (tampilan layar).



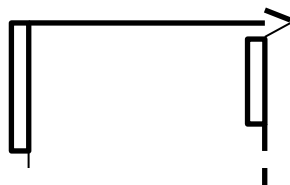
*Control*

Suatu obyek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas.



*Entity*

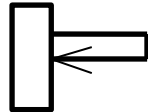
Menggambarkan suatu objek yang berisi informasi kegiatan yang terkait yang tetap dan disimpan ke dalam suatu database.



*Object Message*

Menggambarkan pengiriman pesan dari sebuah

objek ke objek lain.



*Recursive*

Sebuah obyek yang mempunyai sebuah operation kepada dirinya sendiri.



*Return Message*

Menggambarkan pesan/hubungan antar objek, yang menunjukkan urutan kejadian yang terjadi.



*Lifeline*

Garis titiktitik yang terhubung dengan obyek, sepanjang lifeline terdapat activation.



*Activation*

Activation mewakili sebuah eksekusi operasi dari obyek, panjang kotak ini berbanding dengan durasi aktivasi sebuah operasi.