

**IMPLEMENTASI ALGORITMA *COLLISION DETECTION* PADA GAME  
*SAVE THE RED BIRD* BERBASIS *ANDROID* MENGGUNAKAN *UNITY*  
3D**

**SKRIPSI**

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



Oleh :  
SUTianto RAMADANA  
1611500065

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
INSTITUT SAINS DAN BISNIS ATMA LUHUR  
PANGKALPINANG**

**2020**

## LEMBAR PERYATAAN

Yang bertanda tangan di bawah ini :

NIM : 1611500065

Nama : Sutianto Ramadana

Judul Skripsi : IMPLEMENTASI ALGORITMA *COLLISION DETECTION*  
PADA GAME SAVE THE RED BIRD BERBASIS ANDROID  
MENGUNAKAN UNITY 3D

Menyatakan bahwa Laporan Tugas Akhir saya adalah hasil karya sendiri dan bukan hasil plagiat. Apabila ternyata ditemukan didalam Tugas Akhir saya terdapat unsur plagiat, maka saya siap untuk mendapatkan sanksi akademik yang berkaitan dengan hal tersebut.

Pangkalpinang Juli 2020



(Sutianto Ramadana)

## KATA PENGANTAR

Puji syukur Alhamdulillah kehadirat Allah SWT yang telah melimpahkan segala rahmat dan karuniaNya, sehingga penulis dapat menyelesaikan laporan skripsi yang merupakan salah satu persyaratan untuk menyelesaikan jenjang strata satu (S1) pada Program Studi Teknik Informatika Institut Sains dan Bisnis (ISB) Atma Luhur.

Penulis menyadari bahwa laporan skripsi ini masih jauh dari sempurna. Karena itu, kritik dan saran akan senantiasa penulis terima dengan senang hati.

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 menciptakan dan memberikan kehidupan di dunia.
2. Bapak dan Ibu tercinta yang telah mendukung penulis baik sprit maupun materi.
3. Bapak Drs. Djaetun Hs yang telah mendirikan Atma Luhur.
4. Bapak Dr Husni Teja Sukmana, S.T., M.Sc, selaku Rektor ISB Atma Luhur.
5. Bapak Chandra Kirana, M. Kom Selaku Kaprodi Teknik Informatika.
6. Kepada Ibu Dwi Yuny Sylfania, M.Kom selaku dosen pembimbing.
7. Ayah, Ibu dan Adek yang telah mendukung saya baik moral maupun materi.
8. Teman-teman Angkatan 2016 Elvin yang telah memberikan dukungan untuk menyelesaikan laporan skripsi ini.

Semoga Tuhan Yang Maha Esa membalas kebaikan dan selalu mencurahkan hidayah serta taufik-Nya, Amin.

Pangkalpinang, Juli 2020

Sutianto Ramadana

## **ABSTRACT**

*Currently there are many game engines that can be used to design and create games, one of which is the Unity 3D game engine. For designing or making games not only personal, even some developers also use them for a game production. Collision Detection algorithm is a detection method when two or more spatial objects collide (collide) and will cause a reaction, this method is very widely used in the world of making animation and making games. The author uses this method to design a bird flight game, when the player character crashes into an object obstacle / enemy barrier that has been arranged on a random track (Artificial Intelli-gence) in order to get a reaction on one of the specified objects. This game is applied on Android-based smartphones so that it can be played by anyone in the hope that the game can attract the attention of users when played, and the user's interest to play the game will also increase.*

*Keywords: Unity 3D, Game, Collision Detection, Android.*



## ABSTRAK

Saat ini banyak sekali game engine yang dapat digunakan untuk merancang dan membuat game salah satunya adalah game engine Unity 3D. Untuk perancangan atau pembuatan game tidak hanya personal bahkan beberapa developer pun menggunakannya untuk sebuah produksi game nya. Algoritma *Collision Detection* adalah sebuah metode pendeteksian ketika dua objek spasial atau lebih yang bertabrakan (bertumbukan) dan akan menimbulkan sebuah reaksi, metode ini sangat banyak digunakan dalam dunia pembuatan animasi maupun pembuatan game. Penulis menggunakan metode ini untuk perancangan sebuah game penerbangan burung, ketika karakter pemain menabrak sebuah objek obstacle / penghalang musuh yang sudah disusun pada sebuah lintasan secara acak (*Artificial Intelligence*) agar mendapatkan sebuah reaksi pada salah satu objek yang ditentukan. Game ini diterapkan pada smartphone berbasis android sehingga dapat dimainkan oleh siapa saja dengan harapan game dapat menarik perhatian bagi pengguna saat dimainkan, serta minat pengguna untuk memainkan game juga akan meningkat.

Kata Kunci :Unity 3D, Game, *Collision Detection*, *Android*.



## DAFTAR ISI

LEMBAR PERNYATAAN.....	i
LEMBAR PENGESAHAN SKRIPSI.....	ii
KATA PENGANTAR.....	iii
ABSTRACT .....	iv
ABSTRAK .....	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR .....	vii
DAFTAR TABEL .....	viii
DAFTAR SIMBOL .....	ix
<b>BAB I PENDAHULUAN</b>	
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan dan Manfaat Penelitian.....	4
1.5 Sistematika Penulisan .....	4
<b>BAB II LANDASAN TEORI</b>	
2.1 Model <i>Prototype</i> .....	6
2.2 Metode <i>Object Oriented Programming</i> .....	6
2.3 Tools pengembangan sistem.....	7
2.3.1 <i>Use Case Diagram</i> .....	7
2.3 .2 <i>Activity Diagram</i> .....	8
2.3 .3 <i>Sequence Diagram</i> .....	8
2.4 Teori pendukung.....	9

2.4.1 Pengertian game .....	9
2.4.2 <i>Collision detection</i> .....	9
2.4.3 <i>Finite State Machine</i> .....	11
2.4.4 Kecerdasan buatan ( <i>Artificial Interlligence</i> ).....	12
2.4.5 <i>Android</i> .....	13
2.4.5.1 Pengertian <i>Android</i> .....	13
2.4.5.2 Sejarah <i>Android</i> .....	14
2.4.5.3 <i>Android Studio</i> .....	15
2.4.5.3.1 <i>Android SDK</i> .....	15
2.4.6 <i>Unity 3D</i> .....	16
2.4.7 <i>JAVA (Java Development Kit)</i> .....	16
2.4.8 <i>Visual Studio (Editor code C++/C#)</i> .....	17
2.4.9 <i>Adobe Photoshop</i> .....	18
2.4.10 Tinjauan Studi.....	19

### **BAB III METODOLOGI PENELITIAN**

3.1 Model Prototype.....	21
3.2 Metode Pengembangan Sistem.....	22
3.3 Tools Pengembangan Sistem.....	22
3.4 Algoritma yang digunakan <i>Collision Detection</i> .....	24

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

4.1 Analisis.....	25
4.1.1 Analisis Masalah.....	25
4.1.2 Analisis Kebutuhan Sistem.....	25
4.1.3 Analisis Algoritma <i>Collision Detection</i> .....	26
4.1.4 Analisis Alur Algoritma.....	27
4.5 Rancangan Sistem.....	28
4.6 Use Case Diagram.....	29
4.6.1 Deskripsi Use Case Diagram Play.....	29
4.6.2 Deskripsi Use case Diagram Quit.....	29

4.7 Activity Diagram.....	30
4.7.1 Activity Diagram Play.....	30
4.7.2 Activity Diagram Quit.....	31
4.8 Sequence Diagram.....	32
4.8.1 Sequence Diagram Play .....	32
4.8.2 Sequence Diagram Quit .....	33
4.9 Rancangan Layar.....	33
4.9.1 Perancangan Interface Aplikasi.....	33
4.9.2 Rancangan layar pembuka / splash.....	34
4.9.3 Rancangan layar Menu Utama.....	34
4.9.4 Rancangan layar tampilan permainan.....	35
4.9.5 Rancangan layar tampilan permainan ketika tombol pause.....	36
4.10 Implementasi .....	36
4.11 Tampilan layar.....	37
4.11.1 Tampilan layar Splash .....	37
4.11.2 Tampilan menu utama permainan.....	37
4.11.3 Tampilan layar gameplay permainan.....	38
4.11.4 Tampilan layar permainan ketika klik jeda atau pause.....	38
4.12 Pengujian.....	39
4.13 Membangun Aplikasi .....	39
4.13.1 Setting Resolution dan presentation.....	39
4.13.2 Other setting.....	40
4.13.3 Build APK.....	40
4.14 Pengujian Aplikasi.....	41
4.15 Pengujian Blackbox.....	42

## **BAB V PENUTUP**

5.1 Kesimpulan.....	45
5.2 Saran.....	45



<b>DAFTAR PUSTAKA .....</b>	<b>46</b>
<b>LAMPIRAN.....</b>	<b>49</b>



## DAFTAR GAMBAR

Gambar 2.1 <i>Use Case Diagram</i> .....	7
Gambar 2.2 <i>Activity Diagram</i> .....	8
Gambar 2.3 <i>Sequence Diagram</i> .....	8
Gambar 2.4 Representasi menggunakan segiempat dan lingkaran .....	11
Gambar 2.5 Diagram state sederhana.....	12
Gambar 4.2 <i>Usecase Diagram</i> .....	26
Gambar 4.3 <i>Activity Diagram Play</i> .....	29
Gambar 4.4 <i>Activity Diagram Quit</i> .....	29
Gambar 4.5 <i>Sequence Diagram Play</i> .....	30
Gambar 4.6 <i>Sequence Diagram Quit</i> .....	31
Gambar 4.7 Rancangan layar <i>splash</i> .....	32
Gambar 4.8 Rancangan layar menu utama .....	32
Gambar 4.9 Rancangan tampilan permainan .....	33
Gambar 4.10 Rancangan layar tampilan permainan saat klik pause.....	34
Gambar 4.11 Tampilan Layar <i>Splash</i> .....	35
Gambar 4.12 Tampilan menu utama permainan .....	35
Gambar 4.13 Tampilan <i>gameplay</i> permainan .....	36
Gambar 4.14 Tampilan permainan ketika klik jeda .....	36
Gambar 4.15 Tampilan pengaturan pengembang aplikasi .....	37
Gambar 4.16 Tampilan pengaturan <i>buid</i> Android .....	38
Gambar 4.17 Tampilan <i>buid apk</i> Android .....	39







## DAFTAR TABEL

Tabel 4.1 Deskripsi <i>Use Case Play</i> .....	27
Tabel 4.2 Deskripsi <i>Use Case Quit</i> .....	28
Tabel 4.3 Hasil Uji Aplikasi.....	39
Tabel 4.4 Pengujian Fungsi Aplikasi .....	40
Tabel 4.5 Pengujian Ke Masyarakat .....	41

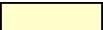



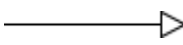
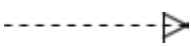
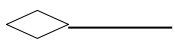







## DAFTAR SIMBOL

### *Usecase Diagram*


Simbol	Nama	Fungsi
	<i>Package</i>	Menambahkan paket Baru dalam diagram
	<i>Actor</i>	menambahkan aktor dalam diagram
	<i>Use case</i>	Menambahkan <i>use case</i> pada Diagram
	<i>Unidirectional association</i>	Menggambarkan relasi antara aktor dengan <i>use case</i>
	<i>Dependencies or instantiates</i>	Menggambarkan ketergantungan ( <i>Dependecies</i> ) atas item didalam diagram
	<i>Generalization</i>	Menggambarkan relasi lanjut antar <i>use case</i> atau menggambarkan struktur pewarisan antar actor

### *Sequance Diagram*







Simbol	Nama	Fungsi
	<i>Object</i>	Menambahkan objek baru
	<i>Association class</i>	Menghubungkan kelas asosiasi

		(association class) pada suatu relasi Asosiasi
	<i>Generalization</i>	Menggambarkan relasi generasi
	<i>Realize</i>	Menggambarkan relasi realisasi
	<i>Aggregation</i>	Menggambarkan relasi Agregasi
	<i>Actor</i>	Menggambarkan aktor pada diagram class
	<i>Use Case</i>	Menggambarkan <i>use case</i> diagram kelas
	<i>Control</i>	Menggambarkan unsur kendali pada diagram
	<i>Boundary</i>	Menambahkan kelas batasan ( <i>Boundary</i> ) pada diagram
	<i>Entity</i>	Menambahkan kelas entitas ( <i>entity</i> ) pada diagram
	<i>Object Message</i>	Menggambarkan pesan ( <i>message</i> ) antara dua objek
	<i>Message To Self</i>	Menggambarkan pesan ( <i>message</i> ) yang menuju dirinya sendiri


	<i>Return Message</i>	Menggambarkan pengembalian dari pemanggil prosedur
---	-----------------------	--



	Destruction Marker	Memperlihatkan saat objek tertentu dihancurkan
---	--------------------	--

### *Activity Diagram*

Simbol	Nama	Fungsi
	State	Menambahkan <i>state</i> untuk suatu objek
	Activity	Menambahkan aktivitas baru pada diagram
	Start State	Memperlihatkan dimana aliran kerja berawal
	End State	Memperlihatkan dimana aliran kerja berakhir
	State transition	Menambahkan transisi dari suatu aktivitas ke aktivitas yang lainnya
	Decisions Point	Menambahkan titik keputusan pada aliran kerja

### *Class Diagram*

Simbol	Nama	Fungsi
	Class	Menambahkan kelas baru pada diagram

	<p>Interface</p>	<p>Menambahkan kelas antarmuka (interface) pada diagram</p>
	<p>Association</p>	<p>Menggambarkan relasi Asosiasi</p>

