

# Rancang Bangun Game Getuk Shooter Menggunakan Algoritma Collision Detection Berbasis Android

*Fauziah<sup>1</sup>, Eka Mutia Putri<sup>2</sup>*

*<sup>1,2</sup>Universitas Kuningan*

*Jl. Cut Nyak Dhien no.36A Kuningan*

*[fauziah@uniku.ac.id](mailto:fauziah@uniku.ac.id)<sup>1</sup>, [eka.mutia.putri@gmail.com](mailto:eka.mutia.putri@gmail.com)<sup>2</sup>*

## ABSTRACT

*Android Based Game has grown rapidly, with the growing user of smartphones and tables in this world. The game maker industry also has many opportunities to produce games. Many advantages of producing games because of many enthusiasts for the game. Making game is an interesting field to learn because of the great potential both in terms of science and in terms of commercial. In this study the author designs and builds an application of the Android based Getuk Shooter game using Collision Detection algorithm for collision attack detection. The method used in this research is RUP method, which is method of software development done repeatedly. The result of this research is to produce Android based mobile game in the form of single player getuk vertical shot that can train the dexterity of player with 2D display and it consists of 3 levels, which can run on android smartphone.*

*Key words : Android, game, collision, shooting*

## 1. PENDAHULUAN

### 1.1. Latar Belakang

Pada era sekarang ini, kebutuhan akan perangkat *mobile* sangat tinggi, yang awalnya sebagai piranti untuk berkomunikasi sekarang sudah banyak perangkat *mobile* yang dapat difungsikan lebih dari sekedar berkomunikasi. Banyak fitur yang disematkan dalam sebuah perangkat *mobile* dengan berbagai macam kegunaan. Salah satu yang paling digemari oleh pengguna *mobile* saat ini adalah game, selain sebagai hiburan game juga dapat digunakan sebagai media pembelajaran, dengan adanya unsur edukasi dalam game tersebut.

*Game* merupakan permainan yang kompleks yang di dalamnya terdapat peraturan, play dan budaya. Di dalam permainan ada sebuah system di dalam pemain terlibat dalam konflik buatan. *Game* bertujuan untuk menghibur, biasanya game banyak disukai oleh anak – anak dan orang dewasa dan *game* juga penting untuk perkembangan otak, untuk meningkatkan konsentrasi dan melatih untuk memecahkan suatu masalah dengan tepat dan cepat.

Salah satu *game* yang paling diminati adalah *game shooting*. *Game shooting* cara bermainnya adalah dengan menembak musuh yang melewati layar yang biasanya menembak secara horizontal. *Game* ini banyak dipilih karena hanya mengandalkan ketangkasan untuk menyelesaikan *game* ini. Apalagi *game* yang sudah

berbasis android yang memang di tujukan untuk smartphone maupun tablet yang sekarang banyak digunakan.

Banyak game-game yang strategi yang disukai oleh anak-anak maupun dewasa, namun kebanyakan dalam game strategi tersebut ada atau bahkan banyak mengandung unsur negatif seperti kekerasan, seperti game yang marak di kalangan anak-anak maupun remaja saat ini adalah game Point Blank, War Rock, Cross Fire, Mortal Kombat dan lain-lain. Sehingga kurang baik bagi anak-anak dan dikhawatirkan hal tersebut akan berpengaruh pada kehidupan mereka dan berdampak buruk terhadap lingkungan sekitar. Oleh karena itu dibutuhkan suatu game yang menarik dan menyenangkan serta sesuai dengan usianya, dapat melatih pola pikir dan ketangkasan.

Penggunaan algoritma dalam game tentu diperlukan sebagai langkah-langkah dalam penyelesaian masalah secara sistematis dan logis. Saat ini sudah ada algoritma yang bisa digunakan untuk mengetahui objek- objek apa saja yang bersentuhan dalam bidang koordinat tertentu. Dari algoritma ini, maka algoritma Collision Detection yang menjadi algoritma yang dapat mendeteksi suatu tubrukan. Algoritma Collision Detection termasuk algoritma untuk deteksi tabrakan biasanya dipakai pada saat pemograman animasi gambar. Maka dari itu penulis ingin menerapkan Algoritma Collision Detection ini ke dalam sebuah game dan. Dengan uraian tersebut maka

penulis mengangkat judul “RANCANG BANGUN GAME GETUK *SHOOTER* MENGGUNAKAN ALGORITMA *COLLISION DETECTION* BERBASIS ANDROID”.

### 1.2. Identifikasi Masalah

Berdasarkan uraian latar belakang masalah yang telah disampaikan adapun identifikasi masalah yang akan dibahas dalam proposal skripsi ini adalah :

- Banyaknya masyarakat menyukai game action yang mengandung unsur kekerasan sehingga dikhawatirkan akan berpengaruh pada sikap masyarakat terutama pada anak-anak dan lingkungan sekitar.
- Adanya kebutuhan algoritma untuk mengetahui objek – objek apa saja yang bersentuhan dalam bidang koordinat tertentu.

## 2. Metodologi Penelitian

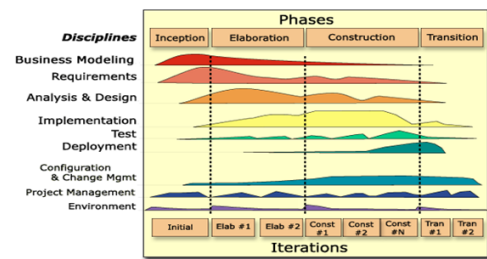
### 2.1. Teknik Pengumpulan Data

Teknik pengumpulan data yang dilakukan oleh penulis menggunakan tiga cara yaitu:

- Metode Kepustakaan**  
Metode Kepustakaan, yaitu proses pengumpulan data melalui buku-buku, tutorial-tutorial dan segala materi yang berkaitan dengan proses penerapan *Algoritma Collision Detection*.
- Metode Observasi**  
Metode Observasi, yaitu memperoleh data dengan cara melakukan pengamatan berbagai *game shooting* terkenal dengan tema yang sama.
- Metode Study Literatur**  
Metode Study Literatur, yaitu mengambil data melalui cara pemanfaatan fasilitas internet, dengan menjelajahi situs-situs yang berhubungan dengan *game shooting* dan *algoritma Collision Detection*.

### 2.2. Metode Pengembangan Sistem

Metodologi penelitian pengembangan perangkat lunak yang digunakan oleh penulis dalam penelitian ini adalah Rational Unified Proses (RUP). Rational Unified Process (RUP) adalah pendekatan perangkat lunak yang dilakukan berulang-ulang (iterative), fokus pada arsitektur (architecture-centric), lebih diarahkan berdasarkan penggunaan kasus (use case driven). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang lebih baik (well defined) dan penstrukturan yang baik (well structured). RUP menyediakan pendefinisian yang baik untuk alur hidup proyek perangkat lunak. RUP adalah sebuah produk proses perangkat lunak yang dikembangkan oleh Rational Software yang diakuisisi oleh IBM di bulan Februari 2003.



Gambar 2.1 Tahap RUP

## 3. LANDASAN TEORI

### 3.1. Game

Pengertian *game* ialah istilah yang digunakan untuk menyebutkan perangkat lunak permainan (Andi : 2009). Sedangkan Menurut **kamus besar bahasa indonesia (KBBI)**, permainan adalah sesuatu yang digunakan untuk bermain, barang atau sesuatu yang dipergunakan. *Game* berasal dari kata bahasa Inggris yang memiliki arti dasar Permainan. Permainan dalam hal ini merujuk pada pengertian kelincahan intelektual (*intellectual playability*). Pengertian *game* pada umumnya berarti aktifitas yang bisa berupa tindakan nyata ataupun tindakan di dalam suatu sistem atau aplikasi yang dapat membawa kesenangan atau hiburan bagi penggunaannya. Dimana hiburan yang didapat tetap mempunyai aturan dan target.

### 3.2. Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat piranti lunak untuk *smartphone*. kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance (OHA)*.

Sejak tahun 2008, Android secara bertahap telah melakukan sejumlah pembaruan untuk meningkatkan kinerja sistem operasi, menambahkan fitur baru, dan memperbaiki bug yang terdapat pada versi sebelumnya. Setiap versi utama yang dirilis dinamakan secara alfabetis berdasarkan nama-nama makanan pencuci mulut atau cemilan bergula; misalnya, versi 1.5 bernama Cupcake, yang kemudian diikuti oleh versi 1.6 Donut. Versi terbaru adalah 5.0 Lollipop, yang dirilis pada 15 Oktober 2014. (Nazruddin Safaat H, 2012).

Penulis akan mengimplementasikan *Game* ini pada versi Android 6.0.1 Marshmallow.

### 3.3. Algoritma Collision Detection

*Algoritma Collision Detection* merupakan proses pengecekan apakah beberapa buah objek spasial saling bertumpuk atau tidak. Jika ternyata ada paling sedikit dua buah objek tersebut dikatakan saling bertumpukan. Pada ruang spasial dua dimensi objek bertumpuk berarti objek spasialnya beririsan.

Penerapan Algoritma Collision Detection: (Nisa, dkk)

Untuk Pemain misal mempunyai nilai (titik koordinat  $X^1$ , titik koordinat  $Y^1$ , Lebar1 dan Panjang1)

Untuk Musuh (Getuk) misal mempunyai nilai (titik koordinat  $X^2$ , titik koordinat  $Y^2$ , Lebar2 dan Panjang2)

Collision Detection =  $((X^1 < X^2 + L2) \ \&\& \ (X^1 + L1 > X^2) \ \&\& \ (Y^2 < Y^2 + P2) \ \&\& \ (Y^1 + P1 > Y^2))$

Misal :

Nilai Pemain (rect1) =

Dik : (titik koordinat  $X^1:5$ , titik koordinat  $Y^1:5$ , L1:50, P1:50)

Nilai Musuh (rect2) =

Dik : (titik koordinat  $X^2:20$ , titik koordinat  $Y^2:10$ , L2:10, P2:10)

Maka :

Jika  $((X^1 < X^2 + L2) \ \&\& \ (X^1 + L1 > X^2) \ \&\& \ (Y^1 < Y^2 + P2) \ \&\& \ (P1 + Y^1 > Y^2))$

Jika  $((5 < 20 + 10) \ \&\& \ (5 + 50 > 20) \ \&\& \ (5 < 10 + 10) \ \&\& \ (50 + 5 > 10))$

Jika  $((5 < 30) \ \&\& \ (55 > 20) \ \&\& \ (5 < 20) \ \&\& \ (55 > 10))$  maka terjadi tumbukan (collision detection).

Berikut adalah pseudocode algoritma Collision Detection pada panah:

```
public bool CheckCollision(GameObject[] temp_G)
{
    foreach (var item in temp_G)
    {
        float x1 = gameObject.transform.position.x - (gameObject.GetComponent<Renderer>().bounds.size.x / 2);
        float y1 = gameObject.transform.position.y - (gameObject.GetComponent<Renderer>().bounds.size.y / 2);
        float lebar1 = gameObject.GetComponent<Renderer>().bounds.size.x;
        float panjang1 = gameObject.GetComponent<Renderer>().bounds.size.y;

        float x2 = item.transform.position.x - (item.GetComponent<Renderer>().bounds.size.x / 2);
        float y2 = item.transform.position.y - (item.GetComponent<Renderer>().bounds.size.y / 2);
        float lebar2 = (item.GetComponent<Renderer>().bounds.size.x);
        float panjang2 = (item.GetComponent<Renderer>().bounds.size.y);
```

```
component<Renderer>().bounds.size.y);

        //Periksa collision
        if (((x1 < x2 + lebar2) && (x1 + lebar1 > x2) && (y2 < y2 + panjang2) && (y1 + panjang1 > y2)))
        {
            //Terjadi collision
            if (item.tag == "Enemy")
            {
                Destroy(item);
                return true;
            }
        }
        //Tidak terjadi collision
        return false;
    }
}
```

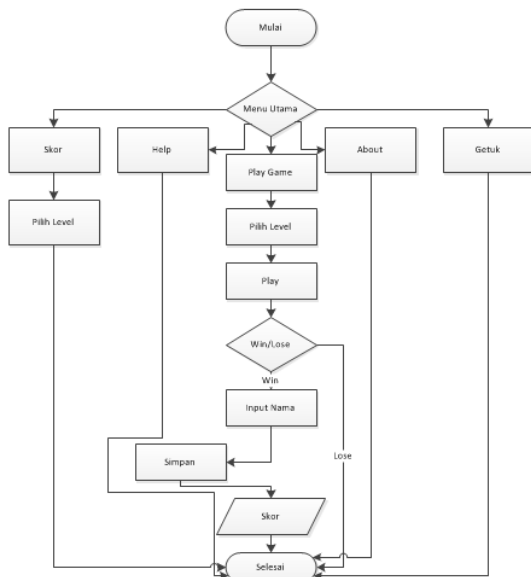
### 3.4. C#

Bahasa pemrograman C# (C Sharp) merupakan gabungan dari kecanggihan bahasa keluarga C (C, C++, Objective-C, Java) yang dirancang oleh Microsoft Corp, sebagai bahasa pemrograman yang sangat berdaya-guna, aman (*secure*), serta mudah digunakan. Sebagai bagian dari platform .NET, bahasa pemrograman C# dirancang sedemikian rupa untuk bekerja dengan sangat baik di atas *framework* .NET yang mampu digunakan untuk menulis perangkat lunak handal demi layanan yang cepat. Bahasa pemrograman C# juga dapat digunakan untuk mengembangkan aplikasi sarana bergerak (*mobile application*), aplikasi berbasis web (*web-based application*), serta aplikasi berskala besar (*enterprise*).

## 4. ANALISIS DAN PERANCANGAN

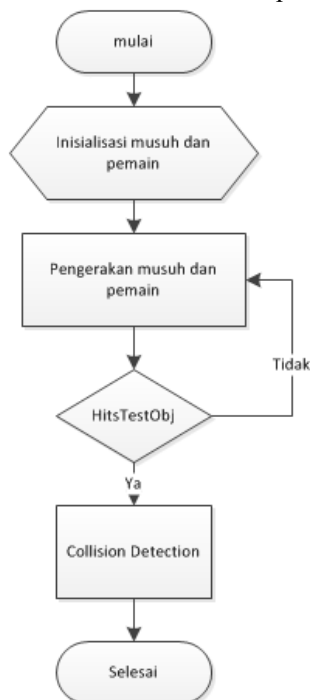
### 4.1. Game Layout Chart

Game layout chart merupakan langkah awal dalam mendesign game. Chart ini menjelaskan atau menggambarkan bagaimana game ini bekerja, dimana pembuatannya dibuat dengan sederhana sehingga mudah dipahami. Berikut game layout chart untuk game getuk shooter:



Gambar 3.1. Game Layout Chart  
**4.2. Flowchart Algoritma Collision Detection**

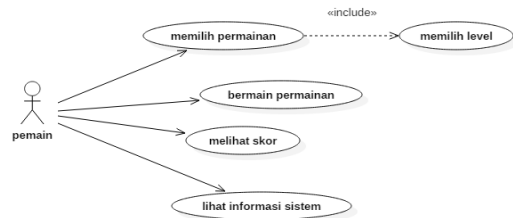
Pada Algoritma Collision detection ini hanya menghitung titik koordinat pada game yang akan di buat atau mendeteksi suatu tubrukan di setiap objek.



Gambar 3.2. Flowchart Algoritma Collision Detection

**4.3. Use Case Diagram**

Usecase diagram menggambarkan aktifitas yang dilakukan oleh aktor pada sistem. Berikut perancangan use case untuk game getuk shooter:



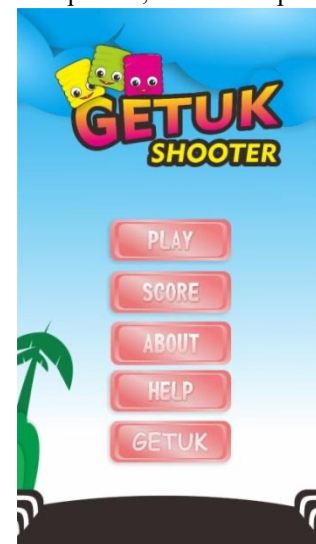
Gambar 3.3. Use Case Diagram

**5. Implementasi Dan Pengujian**

**5.1. Implementasi Desain Interface**

1. Menu Utama

Menu utama, dimana dalam menu ini memiliki beberapa pilihan, berikut tampilannya :



Gambar 5.1. Tampilan Menu Utama

Gambar 5.1 merupakan halaman utama dari game ini, ada 5 button menu dimana tiap button mengarahkan ke halaman yang tertentu sesuai dengan rancangan pada bab III.

2. Menu Mulai Permainan

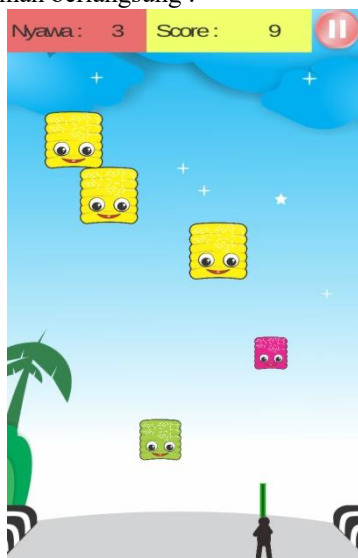
Tampilan menu seperti pada gambar 5.2 dibawah ini :



Gambar 5.2. tampilan mulai permainan  
Terlihat pada gambar 5.2 diatas, menu mulai permainan hanya terdapat fungsi pemilihan level yang akan ditampilkan pada saat permainan berlangsung. Ada 3 tingkatan yaitu mudah, sedang, dan sulit.

3. Menu Start Game

Gambar 5.3 menunjukkan tampilan saat permainan berlangsung :



Gambar 5.3. tampilan scene game  
Setelah pemilihan level maka akan muncul scene game yang di dalam nya terdapat karakter orang dan karakter getuk. Di permainan ini pemain atau karakter orang menembak karakter getuk, pemain menembak vertikal secara otomatis dan berusaha untuk menembak sebanyak – banyaknya untuk mendapatkan skor dan pemain juga harus dapat menghindari dari karakter getuk jika pemain bertubrukan

dengan getuk maka nyawa pemain akan habis dan permainan pun selesai. Jika permainan selesai maka pemain mendapatkan skor yang di capai dan harus mengisi nama untuk di simpan dalam database yang akan dimunculkan pada skor. Dan berikut gambar 5.4 pada saat pemain mendapatkan skor dan mengisi nama.

4. Menu skor game



Gambar 5.4. Tampilan menu skor game  
Pada gambar 5.4 menu skor game, skor di bagi 3 kategori level yaitu : mudah, sedang dan sulit. Pemain akan memilih level yang dimainkan dan skor akan muncul, skor paling atas atau urutan nomor 1 adalah pemain dengan perolehan skor paling tinggi, skor yang ditampilkan hanya sebanyak 10 pemain yang memperoleh skor tinggi. Pada menu skor terdapat informasi nama pemegang skor, perolehan skor atau nilai.

5. Menu bantuan

Gambar dibawah menunjukkan tampilan untuk halaman bantuan seperti gambar 5.5:



Gambar 5.5. Tampilan menu bantuan  
Terlihat pada gambar 4.5 menu bantuan  
hanya berisi penjelasan saat permainan  
berlangsung.

## 6. Kesimpulan Dan Saran

### 1. Kesimpulan

Kesimpulan yang dapat diperoleh dari hasil penulisan skripsi rancang bangun game *Getuk Shooter* dengan mengimplementasikan algoritma *Collision Detection* ini adalah sebagai berikut :

1. Algoritma *Collision Detection* telah berhasil diterapkan pada game *Getuk Shooter* tepatnya di dalam karakter player untuk mengetahui objek – objek apa saja yang bersentuhan dalam bidang koordinat tersebut.
2. Aplikasi game *Getuk Shooter* telah berhasil dibangun sebagai sarana hiburan yang menyenangkan untuk semua kalangan dengan tidak mengandung unsur negatif terutama kekerasan.

### 2. Saran

Berdasarkan perancangan dan hasil implementasi program aplikasi yang dilakukan, maka saran yang perlu diperhatikan dalam mengembangkan aplikasi ini adalah sebagai berikut :

1. Aplikasi ini dapat dikembangkan dengan menambahkan fitur-fitur tambahan dan desain grafis yang lebih baik agar game lebih menarik.
2. Aplikasi ini dapat dikembangkan dengan menambahkan unsur edukasi agar penggunaan game ini lebih bermanfaat.

3. Aplikasi ini dapat dikembangkan ke dalam bentuk 3D agar pemain lebih tertarik untuk memainkannya.

## DAFTAR PUSTAKA

- A.S, Rosa dan M. Salahuddin. 2011. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung : Informatika Bandung
- Asmiatun Siti. 2016. Penerapan Algoritma *Collision Detection* dan *Bayesian* Untuk Strategi Menyerang Jarak Dekat Pada NPC (*Non Player Character*) Menggunakan Unity 3D. Fakultas Teknologi Informasi dan Komunikasi Universitas Semarang.
- Chandra, Cato dan Zakaria, Teddy M. 2008. *Be Smart Be Profesional With Microsoft Office 2007*. Bandung : Informatika.
- H. Darmawan Erico. 2011. Pemrograman Berorientasi Objek C#. Bandung : Informatika Bandung
- Kadir, Abdul. 2003. Perancangan Sistem Informasi. Yogyakarta : Andi
- Komputer Wahana.(2009). Kamus Lengkap Dunia Komputer.Yogyakarta : Andi.
- Ladjamudidin B, Al-Bahra bin. 2006. *Rekayasa Perangkat Lunak*. Yogyakarta : Graha Ilmu
- Musfiroh Lia,dkk. 2014. Penerapan Algoritma *Collision Detection* dan *Boids* Pada Game *Dakkaebi Shooter*. Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muria Kudus.
- Nisa Yulianti Haerun,dkk. Penerapan Metode *Collision Detection* Dalam Permainan Berbasis Android. Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Pakuan.
- Pranata Arya Baskara,dkk. 2015. Mudah Membuat Game dan Potensi Finansialnya dengan Unity 3D. Jakarta, PT. Elex Media Komputindo
- Presman, Roger, S.(1997).Rekayasa Perangkat Lunak: Pendekatan Praktisi (Edisi satu).Yogyakarta.Penerbit:Andi
- Pressman. S. Roger, (2012). *Rekayasa Perangkat Lunak (Buku Satu)*. Yogyakarta : Andi Publisher.
- Safaat H.Nazruddin (2012).Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android.Yogyakarta.Penerbit:Informatika,
- Simarmata Janner.(2010).Rekayasa Perangkat Lunak.Penerbit:Andi, Yogyakarta  
(<https://multimediasmkatarunabhakti.wordpress.com/2013/05/22/pengenalan-staruml/> Diakses tgl 13-02-2018 jam 07:38)