

## BAB 2

### LANDASAN TEORI

Pada bab ini akan diuraikan beberapa hal mengenai teori yang dijadikan landasan dalam pengembangan aplikasi game rangku alu pada *platform* Android. Landasanteori ini diuraikan meliputi penjelasan mengenai Rangku Alu, pengembangan *mobile game*, *platform* Android, Algoritma *Fisher-Yates*, *Unity3D*, Android SDK.

#### 2.1 Rangku Alu

Rangku Alu adalah sebuah permainan tradisional Indonesia dari Manggarai Nusa Tenggara Timur (NTT) dengan menggunakan dua pasang bambu atau tongkat yang di gerak-gerakkan hingga kaki pemain lawan terjepit bambu. Bagi masyarakat Manggarairangku alu dilakukan untuk merayakan hasil panen perkebunan dan pertanian (Nanang, 2015).

Dalam permainan rangku alu ini, bambu disusun dan dimainkan dengan cara diayunkan seperti menjepit oleh beberapa orang pemain. Salah satu atau dua pemain melompat-lompat menghindari jepitan dari bambu tersebut. Saat melompat-lompat menghindari jepitan, pemain seakan melakukan gerakan tari. Dari situlah awal terbentuknya gerakan dasar tari rangku alu ini. Gerakan penari dan pemain bambu tersebut kemudian dipadukan dengan irama musik dan lagu daerah sehingga menghasilkan seni yang khas, yaitu tari rangkualu. Dulunya, tarian ini sering ditampilkan saat usai panen raya, pada saat bulan purnama. Pada saat itulah para remaja berkumpul dan meramaikan acara ini.

Dalam pertunjukannya, tarian rangku alu ini dimainkan oleh para remaja, baik laki-laki maupun perempuan dengan menggunakan pakaian adat seperti ikat kepala, baju bero, dan kain songket khas daerah Manggarai. Tari rangku alu ini biasanya dimainkan oleh 4 orang pemegang bambu dan beberapa orang menari secara

bergantian. Untuk memainkan bambu tersebut dipadukan dengan irama musik dan lagu, sehingga gerakan para penari pun bisa terlihat seirama.

Untuk gerakan tarian ini, sebenarnya berasal dari gerakan para penari saat menghindari jepitan bambu, sehingga didominasi oleh gerakan kaki. Dalam tarian ini tentu membutuhkan kelincahan dan ketepatan untuk menghindari jepitan bambu. Apabila penari kurang lincah maka akan terjatuh karena terjepit bambu. Namun dari situlah keseruan dari tari rangkuk alu ini. Selain mendebarkan juga mengundang gelak tawa para penonton.

Dalam pertunjukan tari rangkuk alu ini biasanya diiringi oleh alunan musik tradisional seperti gong dan gendang. Selain diiringi dengan musik, tari rangku alu juga diiringi dengan nyanyian lagu daerah. Seperti yang dikatakan sebelumnya, irama musik dan nyanyian ini juga di sesuaikan dengan pemegang bambu, sehingga gerakan penari yang melompatpun terlihat seirama. Kemenangan dalam permainan ini adalah pelompat harus berhasil menghindari jepitan bambu selama mungkin sesuai dengan tingkatan *ritme* lompatan.

Dalam perkembangannya, tari rangku alu telah menjadi salah satu tarian tradisional yang cukup terkenal di Manggarai, Flores, NTT. Tarian ini sering di tampilkan di berbagai acara seperti acara perayaan, acara budaya, penyambutan tamu penting dan berbagai acara lainnya. Selain sebagai bentuk warisan budaya, tari rangkualu juga telah menjadi daya tarik bagi para wisatawan yang datang kesana.

## **2.2 Mobile Game**

*Mobile game* pada umumnya diartikan sebagai permainan yang dimainkan pada perangkat *mobile* seperti telepon genggam, PDA, *smartphone*, ataupun permainan portabel seperti Playstation Portable dan Nintendo DS. Penelitian, pengembangan dan pengujian yang dilakukan pada *mobile games* telah membawa masalah dan pertanyaan yang tidak dapat diselesaikan oleh model desain lama. *Mobile games* mengubah hubungan antara manusia dan komputer yang berakibat tidak hanya dari cara bermain tetapi desain *interface* dan interaksi struktur juga berubah. Kegiatan pemain berubah, pemain tidak lagi duduk didepan perangkat mereka tapi benar-benar dapat bergerak bebas sambil bermain (Grueter *et al*, 2005).

### 2.2.1 *Designing Mobile Game*

Sekarang ini perangkat-perangkat *portable* sering digunakan dalam segala aspek kehidupan kita sehari-hari seperti bermain, bekerja, belajar bahkan sambil berkendara. Pendesainan aplikasi khususnya aplikasi *game* berbasis *mobile* juga dipengaruhi aspek-aspek ini. Berikut adalah hasil studi yang dilakukan oleh Weedon(2009) mengenai *user requirement* dalam pembuatan *mobile games*.

#### 1. *Gaming on the go*

- a. Pentingnya akses cepat dalam menjalankan suatu game.
- b. Cepat dan mudah saat mengakses *save point/pause* ketika berada dalam kendaraan.
- c. *Auto saving* saat perangkat mati atau *sleep mode* sangat penting.

#### 2. *Visibility*

- a. Pertimbangan untuk menggunakan warna *high contrast* antara *background* dan *foreground* untuk membantu visibilitas saat berada di tempat bercahaya terang.
- b. Indikator penting harus sangat tampak sehingga dapat terlihat dalam kondisi pencahayaan yang buruk pada layar kecil.

#### 3. *Touch Screen Issue*

- a. *OnScreen gesture* sering bekerja lebih baik ketimbang tombol.
- b. Pastikan pada *OnScreen control* tidak terlalu kecil atau terlalu berdekatan.

#### 4. *Phone Issues*

- a. Idealnya *mobile game* pada *handphone* dapat dimainkan dengan satu tangan.
- b. Apa yang terjadi dengan *game* saat ada panggilan, sms, atau *email* masuk.

#### 5. *Audio*

- a. Pertimbangan bagaimana suara notifikasi *handphone* saat game juga mengeluarkan suara.
- b. Idealnya game dapat dimainkan tanpa suara, ketika di tempat umum pemain tidak membawa *headphones*.
- c. Akan lebih berguna jika mengembangkan pengatur suara.

d. Sediakan pilihan bagi pengguna untuk memilih musiknya sendiri.

#### 6. *Inputs*

- a. Pertimbangkan ketersediaan perangkat *input/output* pada perangkat mempengaruhi permainan. (*microphones, motion sensors, accelerometer, cameras, touch screens, WiFi, GPS, dan lain-lain*).
- b. Jika permainan sangat bergantung dengan koneksi jaringan, maka kekuatan jaringan harus ditampilkan setiap waktu.

#### 7. Lokasi umum saat bermain

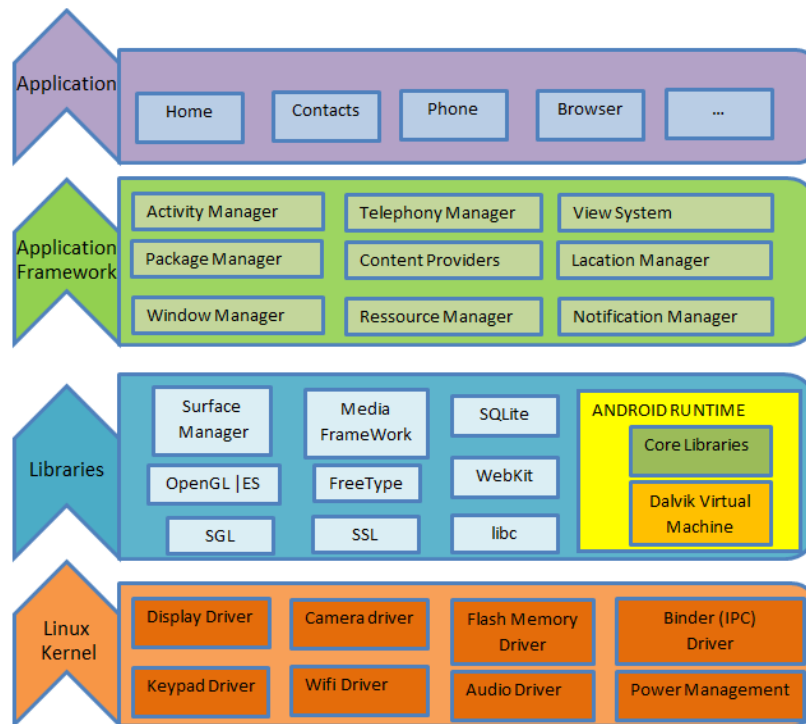
- a. Di kendaraan umum saat beraktifitas sehari-hari.
- b. Di kendaraan umum saat melakukan perjalanan panjang.
- c. Di bandara atau dalam pesawat.
- d. Di sekolah.
- e. Menunggu seseorang
- f. Di rumah.

### 2.3 Android

Android adalah sistem operasi *mobile* dan *platform* yang didasari oleh linux kernel versi 2.6 dan tersedia secara bebas untuk penggunaan *commercial* ataupun *noncommercial*. dan bersifat *open source*.

Saat kita ingin membuat *game* menggunakan android, *platform* pada android memiliki beberapa kemudahan (Derek James, 2013), yaitu:

1. Android adalah *openplatform*, yang artinya android tidak membatasi apa yang kita bisa akses atau apa yang bisa kita lakukan.
2. Android adalah *mobile platform* yang paling cepat berkembang, yang artinya lebih banyak orang yang akan mengunduh dan memainkan *game* kita.



**Gambar 2.1. Arsitektur Android (Hasina et al., 2014)**

Pada bagian paling bawah, ada linux kernel yang menyediakan *drivers* dasar untuk komponen *hardware*. Kemudian di atasnya dibangun android *runtime* dan *libraries*. Android *runtime* bertanggung jawab untuk menjalankan aplikasi android. Setiap aplikasi android berjalan pada proses mereka masing-masing menggunakan *Dalvik Virtual Machine* mereka masing-masing. *Dalvik Virtual Machine* dihubungkan dengan *core libraries* yang menyediakan fungsi dasar program java. Disamping *core libraries*, juga ada kumpulan dari C/C++ *libraries* yang membangun dasar untuk aplikasi *framework*. *Library* ini bertanggung jawab untuk melakukan proses yang berat seperti *graphics rendering*, *audio playback*, dan *database access*. Kemudian *application framework* menghubungkan *system libraries* dan *runtime*. *Framework* ini mengatur aplikasi dan menyediakan *framework* yang rinci dalam setiap operasi aplikasi. Kemudian bagian paling atas adalah aplikasi, apakah aplikasinya itu *User Interfaces* atau *background services* dapat mengkomunikasikan kemampuan mereka pada aplikasi lainnya. Komunikasi ini memungkinkan aplikasi untuk menggunakan kembali komponen dari aplikasi lainnya.

## 2.4 Algoritma *Fisher-Yates Shuffle*

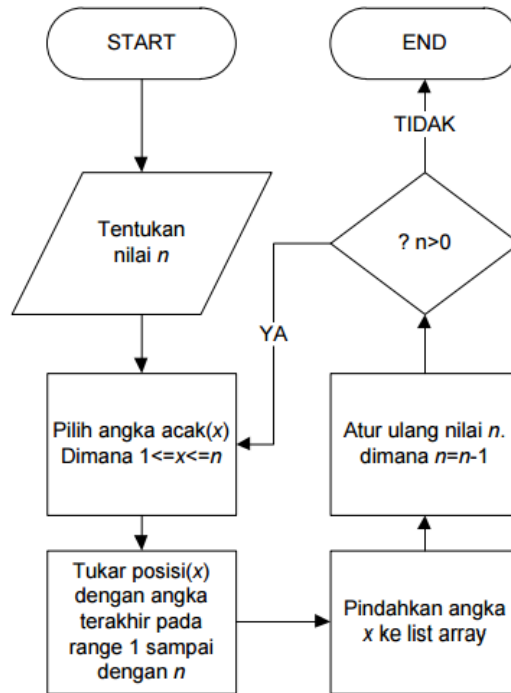
*Fisher-Yates shuffle* (diambil dari nama Ronald Fisher dan Frank Yates) atau juga dikenal dengan nama Knuth shuffle (diambil dari nama Donald Knuth), adalah sebuah algoritma untuk menghasilkan suatu permutasi acak dari suatu himpunan terhingga, dengan kata lain untuk mengacak suatu himpunan tersebut. *Fisher-Yates shuffle* jika diimplementasikan dengan benar, maka hasil dari algoritma ini tidak akan berat sebelah, sehingga setiap permutasi memiliki kemungkinan yang sama (Ade-Ibijola, 2012). Misalkan dalam permainan kartu Blok Bakarada 10 kartu yang akan diacak, maka *array*-nya adalah urutan Kartu = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]. *Array* tersebut dimasukkan kedalam prosedur *shuffle* dimana proses pengacakan terjadi. *Array* tersebut didapat panjang *array* yang kemudian dimasukkan ke dalam variabel *m* ( $m=10$ ) (Ade-Ibijola, 2012). Langkah-langkah pengacakan soal dengan *Fisher-Yates Shuffle* adalah sebagai berikut :

1. Ambil satu elemen secara acak dari elemen yang tersisa. Pengambilan elemen acak adalah berdasarkan elemen yang tersisa. Misalkan jika  $m = 10$ , maka elemen acak yang boleh diambil adalah  $10(array[0..9])$ .
2. Tukar dengan elemen saat ini. Penukaran dilakukan dengan memasukkan elemen saat ini ke dalam suatu variabel baru bernama *t*. Elemen saat ini ( $array[m]$ ) diisi nilai dari elemen acak tadi ( $array[i]$ ). Dan elemen acak ( $array[i]$ ) diisi nilai dari variabel *t*.
3. Ulangi selama masih ada elemen yang tersisa Tahap ketiga dilakukan pengulangan sebanyak 10 kali untuk mengacak urutan soal tersebut. Iterasinya bisa dilihat pada tabel 2.1.

**Tabel 2.1** Contoh perhitungan dengan algoritma Fisher-Yates

<b>m</b>	<b>i</b>	<b>array[i]</b>	<b>t /array[m]</b>	<b>Array yang sudah fix array[0,1,2,3,4,5,6,7,8,9]</b>	<b>isi array setelah di swap array[0,1,2,3,4,5,6,7,8,9]</b>
10	6	6	9	6	0, 1, 2, 3, 4, 5, 9, 7, 8, 6
9	0	0	8	0, 6	8, 1, 2, 3, 4, 5, 9, 7, 0, 6
8	1	1	7	1, 0, 6	8, 7, 2, 3, 4, 5, 9, 1, 0, 6
7	2	2	9	2, 1, 0, 6	8, 7, 9, 3, 4, 5, 2, 1, 0, 6
6	2	9	5	9, 2, 1, 0, 6	8, 7, 5, 3, 4, 9, 2, 1, 0, 6
5	1	7	4	7, 9, 2, 1, 0, 6	8, 4, 5, 3, 7, 9, 2, 1, 0, 6
4	1	4	3	4, 7, 9, 2, 1, 0, 6	8, 3, 5, 4, 7, 9, 2, 1, 0, 6
3	1	3	5	3, 4, 7, 9, 2, 1, 0, 6	8, 5, 3, 4, 7, 9, 2, 1, 0, 6
2	1	5	5	5, 3, 4, 7, 9, 2, 1, 0, 6	8, 5, 3, 4, 7, 9, 2, 1, 0, 6
1	0	8	8	8, 5, 3, 4, 7, 9, 2, 1, 0, 6	8, 5, 3, 4, 7, 9, 2, 1, 0, 6

Permutasi yang didapatkan adalah 8 5 3 4 7 9 2 1 0 6. Kolom m menunjukkan index elemen saat ini. Kolom i menunjukkan index dari elemen yang akan ditukar, nilai i tersebut diambil secara acak dari range yang diperbolehkan (m). Kolom *array[i]* menunjukkan nilai yang terdapat pada *array* ke-i. Kolom t menunjukkan nilai yang terdapat pada *array* ke-m (elemen saat ini). *array* yang sudah fix menunjukkan *array* yang sudah tidak boleh ditukar kembali nilainya, jumlahnya bertambah seiring bertambahnya iterasi. Kemudian kolom *array* yang sudah di swap menunjukkan isi *array* setelah elemennya ditukar/swap ada setiap iterasi. Untuk *flowchart* dari algoritma *Fisher-Yates shuffle* dapat dilihat pada Gambar 2.2.



Gambar 2.2 Flowchart Fisher-Yates Shuffle

Dari flowchart pada Gambar 2.2 terlihat bahwa pengacakan akan selesai jika seluruh array telah diacak. Fisher-Yates shuffle dihasilkan urutan array yang diacak. Untuk pseudocode-nya dapat dilihat pada Tabel 2.2

Tabel 2.2 Pseudocode dengan algoritma Fisher-Yates Shuffle

No	Algoritma	Pseudocode
1	Ketikan masih ada elemen tersisa untuk diacak	While (m) {
2	Ambil elemen secara acak dari elemen yang tersisa	<code>i=Math.floor(Math.random()*m-</code> <code>-)</code> ;
3	Kemudian tukar dengan elemen saat ini	<code>t = array [m];</code> <code>array[m] = array[i];</code> <code>array[i] = t;</code> }



## 2.5 Unity

Unity atau Unity3D merupakan suatu aplikasi yang digunakan untuk mengembangkan *game multi platform* yang didesain untuk mudah digunakan. Editor pada Unity dibuat dengan user interface yang sederhana. Unity secara rinci dapat digunakan untuk membuat *video game 3D*, *real time animasi 3D* dan visualisasi arsitektur dan isi serupa yang interaktif lainnya. Editor Unity dapat menggunakan *plugin* untuk *web player* dan menghasilkan *game browser* yang didukung oleh Windows dan Mac (Rosikhan & Aristiawan, 2015).

Unity memiliki fitur yang bernama *Asset Server*. *Asset Server* adalah asset dan juga sekaligus *version control system* yang dilengkapi dengan antarmuka pengguna grafis yang terintegrasi ke dalam Unity. *Asset Server* sangat berguna ketika orang-orang dalam satu tim mengerjakan satu proyek yang sama di komputer yang berbeda-beda atau ketika akses *remote* dibutuhkan. *Asset Server* Unity dapat digunakan oleh semua *scripts* dan *game assets* sebagai solusi dari versi kontrol dan dapat mendukung proyek yang berukuran raksasa yang memiliki folder-folder berukuran *multi gigabytes*. *Server aset* Unity juga cocok pada Mac, Windows dan Linux dan juga berjalan pada PostgreSQL.

Pada Unity versi 4 ke bawah, perizinan atau *licensing* dari Unity terdiri dari dua bentuk yaitu Unity *Free* dan Unity *Pro*. Versi Unity *Free* tersedia dalam bentuk gratis, sedangkan versi Unity *Pro* hanya dapat dibeli. Versi Unity *Pro* memiliki fitur ekstra seperti efek *post processing* dan *render effect texture*. Sedangkan Versi Unity yang gratis, selain memiliki keterbatasan fitur juga memiliki batasan saat *building application*. Aplikasi yang dibangun menggunakan Unity *Free Edition* akan menampilkan *splash screen* berlogo Unity saat dijalankan. Unity dan Unity *Pro* menyediakan tutorial, isi, contoh project, wiki, dukungan melalui forum dan perbaruan kedepannya. Unity dapat digunakan pada iPhone, iPod dan iPad *operating system* yang mana iOS ada sebagai *add-ons* pada Unity editor yang telah ada lisensinya, dengan cara yang sama juga pada Android.

Namun, pada Unity versi 5 dan 6 yang terbagi atas 4 lisensi yaitu Unity *Personal Edition*, *Plus*, *Pro* dan *Enterprise* tidak ada lagi diskriminasi dan pembatasan fitur. Unity *Personal Edition* meskipun gratis, sudah bisa mengakses semua fitur yang sebelumnya hanya bisa diakses oleh Unity versi berbayar. Satu-

satunya perbedaan yang mencolok antara Unity *Personal Edition* dengan Unity versi lainnya adalah Unity *Personal Edition* masih menampilkan *splash screen* berlogokan Unity pada aplikasi yang dibangun, sementara Unity versi lainnya yang berbayar mengizinkan *developer* untuk mengkustomisasi *splash screen*-nya sendiri.

Berikut adalah fitur-fitur yang terdapat pada Unity 3D :

### 1. *Rendering*

*Graphics engine* yang digunakan adalah Direct3D (Windows, Xbox 360), OpenGL (Mac, Windows, Linux, PS3), OpenGL ES (Android, iOS), dan proprietary APIs (Wii). Ada pula kemampuan untuk *bump mapping*, *reflection mapping*, *parallax mapping*, *screen space ambient occlusion (SSAO)*, *dynamic shadows using shadow maps*, *render-to-texture* dan *full-screen post-processing effects*.

Unity dapat mengambil format desain dari 3ds Max, Maya, Softimage, Blender, modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks and Allegorithmic Substance. *Asset* tersebut dapat ditambahkan ke *game project* dan diatur melalui *graphical user interface* Unity.

*ShaderLab* adalah bahasa yang digunakan untuk *shaders*, dimana mampu memberikan deklaratif “*programming*” dari *fixed-function pipeline* dan program *shader* ditulis dalam GLSL atau Cg. Sebuah *shader* dapat menyertakan banyak varian dan sebuah spesifikasi *fallback declarative*, dimana membuat Unity dapat mendeteksi berbagai macam *video card* terbaik saat ini, dan jika tidak ada yang kompatibel, maka akan dilempar menggunakan *shader* alternatif yang mungkin dapat menurunkan fitur dan performa.

Pada 3 Agustus 2013, seiring dengan peluncuran versi 4.2, Unity mengijinkan *developer indie* menggunakan *realtime shadows* hanya untuk *directional lights*, dan juga menambahkan kemampuan dari DirectX11 yang memberikan *shadows* dengan resolusi pixel yang lebih sempurna, *texture* untuk membuat objek 3d dari *grayscale* dengan lebih grafik *facial*, animasi yang lebih halus dan mempercepat FPS.

## 2. Scripting

*Script game engine* dibuat dengan Mono 2.6, merupakan sebuah implementasi *open-source* dari *.NET Framework*. Programmer dapat menggunakan UnityScript (bahasa terkustomisasi yang terinspirasi dari syntax ECMAScript, dalam bentuk JavaScript), C#, atau Boo (terinspirasi dari syntax bahasa pemrograman python). Dimulai dengan dirilisnya versi 3.0, Unity menyertakan versi MonoDevelop yang terkustomisasi untuk *debug script*.

## 3. Asset Tracking

Unity juga menyertakan *Server UnityAsset* – sebuah solusi terkontrol untuk *developer game asset* dan *script*. Server tersebut menggunakan PostgreSQL sebagai *backend*, sistem audio dibuat menggunakan FMOD *library* (dengan kemampuan untuk memutar Ogg Vorbis compressed audio), *videoplayback* menggunakan Theora codec, *engine* daratan dan *vegetasi* (dimana mensupport *tree billboarding*, *Occlusion Culling* dengan Umbra), *built-in lightmapping* dan *global illumination* dengan *Beast*, *multiplayer networking* menggunakan RakNet, dan *navigasi mesh* pencari jalur *built-in*.

## 4. Platforms

Unity mendukung pengembangan ke berbagai *platform*. Didalam project, developer memiliki kontrol untuk mengirim perangkat *mobile*, *web browser*, *desktop*, dan *console*. Unity juga mengizinkan spesifikasi kompresi *textur* dan pengaturan resolusi di setiap *platform* yang didukung.

Saat ini *platform* yang didukung adalah BlackBerry 10, Windows 8, Windows Phone 8, Windows, Mac, Linux, Android, iOS, Unity Web Player, Adobe Flash, PlayStation 3, Xbox 360, Wii U and Wii. Meskipun tidak semua terkonfirmasi secara resmi, Unity juga mendukung PlayStation Vita yang dapat dilihat pada *game* Escape Plan dan Oddworld: New 'n' Tasty.

Rencana *platform* berikutnya adalah PlayStation 4 dan Xbox One. Dan juga rumor untuk kedepanya mengatakan HTML akan menjadi *platformnya*, dan *plug-in* Adobe baru dimana akan disubtitusikan ke Flash Player, juga akan menjadi *platform* berikutnya.

#### 5. *Asset store*

Diluncurkan November 2010, *unityasset store* adalah sebuah *resource* yang hadir di Unity editor. *asset store* terdiri dari koleksi lebih dari 4,400 *asset packages*, beserta *3D models*, *textures* dan *materials*, sistem *particle*, musik dan efek suara, *tutorial* dan *project*, *scripting package*, *editor extensions* dan servis *online*.

#### 6. *Physics*

Unity juga memiliki *suport built-in* untuk *PhysX physics engine* (sejak Unity 3.0) dari Nvidia (sebelumnya Ageia) dengan penambahan kemampuan untuk simulasi *real-time cloth* pada *arbitrary* dan *skinned meshes*, *thick ray cast*, dan *collision layers*.

### **2.6 Android SDK**

Android-SDK merupakan *tools* yang biasa digunakan seorang *programmer* yang ingin mengembangkan aplikasi berbasis google android. Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, contoh kode, dan *tutorial*. Saat ini Android sudah mendukung arsitektur x86 pada Linux (distribusi Linux apapun untuk *desktop* modern), Mac OS X 10.4.8 atau lebih, Windows XP atau Vista. Persyaratan mencakup JDK, Apache Ant dan Python 2.2 atau yang lebih baru. IDE yang didukung secara resmi adalah Eclipse 3.2 atau lebih dengan menggunakan *plugin Android Development Tools (ADT)*, dengan ini pengembang dapat menggunakan teks editor untuk mengedit file Java dan XML serta menggunakan peralatan *command line* untuk menciptakan, membangun, melakukan *debug* aplikasi Android dan pengendalian perangkat Android (misalnya, *reboot*, menginstal paket perangkat lunak dengan jarak jauh).

## 2.7 Penelitian Terdahulu

Adapun penelitian-penelitian terdahulu yang menjadi sumber referensi penelitian ini adalah penelitian yang dilakukan oleh Juardi, Tommy dkk pada tahun 2013 yang membangun game karapan sapi menggunakan metodologi *waterfall model* yaitu *Requirement Analysis and Definition, System and Software Design, Implementation and Unit Testing, Integration and and System Testing, Operation and Maintenance*. Hasil yang dicapai adalah terciptanya game edukasi “Balapan Karapan Sapi” dengan gameplay yang menarik, menghibur dan mudah untuk dimainkan.

Pada tahun 2013 , Farizi et al pada penelitiannya mengembangkan sebuah game Simulator angklung pada *smartphone* berbasis Android dengan tahapan *Game Design* dan *Technical Design*. Tahapan tersebut kemudian diimplementasikan dan dikembangkan menggunakan *AndEngine* dengan bahasa Java. Berdasarkan hasil pengujian unit dan integrasi dapat disimpulkan bahwa unit modul dari program *game* sudah memenuhi kebutuhan fungsional perancangan. Hasil pengujian performa menunjukkan performa yang optimal ketika *game* berjalan pada *smartphone* yang memiliki prosesor minimal sebesar 600 Mhz.

Pada tahun 2015, Ovy Rizki melakukan pengujian pada gerak bola untuk mengetahui tingkat sensitifitas objek serta pengujian fungsi random untuk mengetahui persentase kemungkinan terjadinya keadaan *random event* pada game adaptasi bekel.

Penerapan algoritma fisher-yates telah direalisasikan oleh Ryan et al (2015) untuk pengenalan jenis serangga berbasis Unity 3D pada aplikasi *the lost insect* dengan persentasi tingkat kepuasan pemakai mencapai 80% dan aplikasinya dengan mudah digunakan dengan perolehan persentase mencapai 66,25% .

Untuk keterangan lain mengenai penelitian terdahulu yang berkaitan dengan penelitian ini dapat dilihat pada Tabel 2.3 berikut :

**Tabel 2.3 Penelitian Terdahulu**

No	Nama Penelitian	Penulis	Keterangan
1.	Aplikasi <i>Game Racing</i> “Balapan Karapan Sapi” pada <i>smartphone</i> berbasis Android menggunakan LibGDX Engine.	Juardi, A., Tommy, Salim, W (2013)	<i>Game</i> terbukti menarik, menghibur dan mudah digunakan.
2.	Rancang bangun permainan <i>mobile</i> Simulator Angklung pada system operasi Android.	Farizi, H.R., Aryadita, H. & Adam J.E.M (2013)	Performa <i>game</i> optimal dengan syarat prosesor minimal 600Mhz
3	Game Adaptasi Bekel Berbasis Android	Ovy Rizki Fahrani (2015)	Pengujian pada gerak bola untuk mengetahui tingkat sensitifitas objek serta pengujian fungsi random untuk mengetahui persentase kemungkinan terjadinya keadaan random event.
4	Penerapan Algoritma Fisher-Yates Pada Aplikasi The Lost Insect Untuk Pengenalan Jenis Serangga Berbasis Unity 3D	Ryan et al (2015)	Aplikasi dapat membantu anak-anak dalam mengenali serangga dengan persentase tingkat kepuasan pengguna mencapai 80%.