

**IMPLEMENTASI ALGORITMA *FISHER-YATES SHUFFLE*
SEBAGAI PENGACAK POSISI *NON PLAYABLE*
CHARACTER (NPC) PADA GAME
TANTRA BAHARI**

SKRIPSI

Oleh:

BALQIS KAMALIA FIKRIA

NIM. 10650043



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK
IBRAHIM MALANG
2014**

**HALAMAN PENGAJUAN
IMPLEMENTASI ALGORITMA *FISHER-YATES SHUFFLE*
SEBAGAI PENGACAK POSISI *NON PLAYABLE*
CHARACTER (NPC) PADA GAME
TANTRA BAHARI**

SKRIPSI

**Diajukan kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:
BALQIS KAMALIA FIKRIA
NIM. 10650043**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK
IBRAHIM MALANG
2014**

LEMBAR PERSETUJUAN

**IMPLEMENTASI ALGORITMA *FISHER-YATES SHUFFLE*
SEBAGAI PENGACAK POSISI *NON PLAYABLE*
CHARACTER (NPC) PADA GAME
TANTRA BAHARI**

SKRIPSI

Oleh :

Nama : Balqis Kamalia Fikria

NIM : 10650043

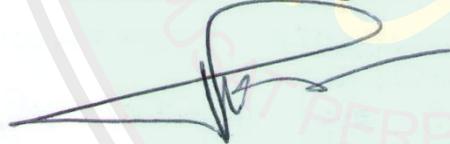
Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Telah diperiksa dan disetujui untuk diuji :

Dosen Pembimbing I

Dosen Pembimbing II



Fachrul Kurniawan, M.MT

Dr. Suhartono, M.Kom.

NIP. 19771020 200901 1 001

NIP. 19680519 200312 1 001

Mengetahui,

Ketua Jurusan Teknik Informatika



Dr. Cahyo Crysdiyan

NIP. 19740424 200901 1 008

LEMBAR PENGESAHAN
IMPLEMENTASI ALGORITMA FISHER-YATES SHUFFLE SEBAGAI
PENGACAK POSISI NON PLAYABLE CHARACTER (NPC) PADA GAME
TANTRA BAHARI

SKRIPSI

Oleh :

BALQIS KAMALIA FIKRIA

NIM. 10650043

Telah dipertahankan di depan dewan penguji skripsi
dan dinyatakan diterima sebagai salah satu persyaratan
untuk memperoleh gelar sarjana komputer (S.Kom)

Tanggal : 26 November 2014

Susunan Dewan Penguji:

1. Penguji Utama : Yunifa Mftachul Aif, M.T
NIP. 19830616 201101 1 004
2. Ketua Penguji : Dr. Muhammad Faisal. M.T
NIP. 19740510 200502 1 007
3. Sekretaris Penguji : Fachrul Kurniawan, M.MT
NIP. 19771020 200901 1 001
4. Anggota Penguji : Dr. Suhartono, M.Kom.
NIP. 19680519 200312 1 001

Tanda Tangan

()
()
()
()

Mengetahui,

Ketua Jurusan Teknik Informatika



Dr. Cahyo Crysdian

NIP. 19740424200901 1 008

PERNYATAAN KEASLIAN PENELITIAN

Saya yang bertandatangan di bawah ini:

Nama : Balqis Kamalia Fikria

NIM : 10650043

Fakultas/Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Penelitian : Implementasi Algoritma *Fisher-Yates Shuffle*
sebagai Pengacak Posisi *Non Playable Character*
(NPC) pada *Game* Tantra Bahari

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 13 November 2014
Yang membuat pernyataan,



Balqis Kamalia Fikria
NIM. 10650043

MOTTO

“Segala aktivitas niatkan ibadah”



HALAMAN PERSEMBAHAN

Dengan rasa syukur seraya mengharap ridho Ilahi. Kupersembahkan karya ini untuk kedua orang tuaku :

Ayahanda H. Ismail Abbas dan Umi Dra. Waridah tercinta yang selalu memberi dukungan semangat, materi dan doa yang tiada henti-hentinya. Terimakasih Jazakumulloh Ahsanal Jaza'.

Dan ucapan terimakasih tak terkira kepada:

- *Adik-adikku, terutama dik Ilma Nurdini MR yang telah banyak membantu, menemani maupun menyemangati dalam menyelesaikan skripsi ini.*
- *Sahabat ikan Vina, Icha, Elis, Diah, Listya, Firoh yang senantiasa mengisi hari-hariku selama di bangku kuliah.*
- *Sahabat Kontrakan Manja. Kiki, Elis, Senyap, Umi. Kebersamaan kita selalu yang terbaik. Terimakasih karena telah saling menyemangati sampai akhir.*
- *Teman - teman seperjuangan skripsi, Vina dan Sari. yang telah memberikan bantuan dan semangat yang luar biasa semoga setelah perjuangan ini kita tetap saling memberikan semangat.*
- *Sahabat-sahabat Teknik Informatika 2010, Sahabat Bzone IT, Rekan kerja di Geomedia yang selalu memberi semangat. Teman-teman asrama Al-Yasini. Semoga kita semua dapat sukses dunia akhirat. Amiin.*

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillah rabbil'Alamin penulis haturkan kehadiran Allah SWT yang telah memberikan rahmat, hidayah, dan ridha-Nya, sehingga penulis dapat menyelesaikan studi di jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus menyelesaikan skripsi berjudul "Implementasi Algoritma *Fisher Yates Shuffle* sebagai Pengacak Posisi *Non Playable Character (NPC)* pada *Game Tantra Bahari*" dengan baik.

Selanjutnya penulis menghaturkan ucapan terima kasih kepada semua pihak yang telah memberikan doa, harapan, dan semangat untuk terselesaikannya skripsi ini. Ucapan terima kasih, penulis sampaikan kepada :

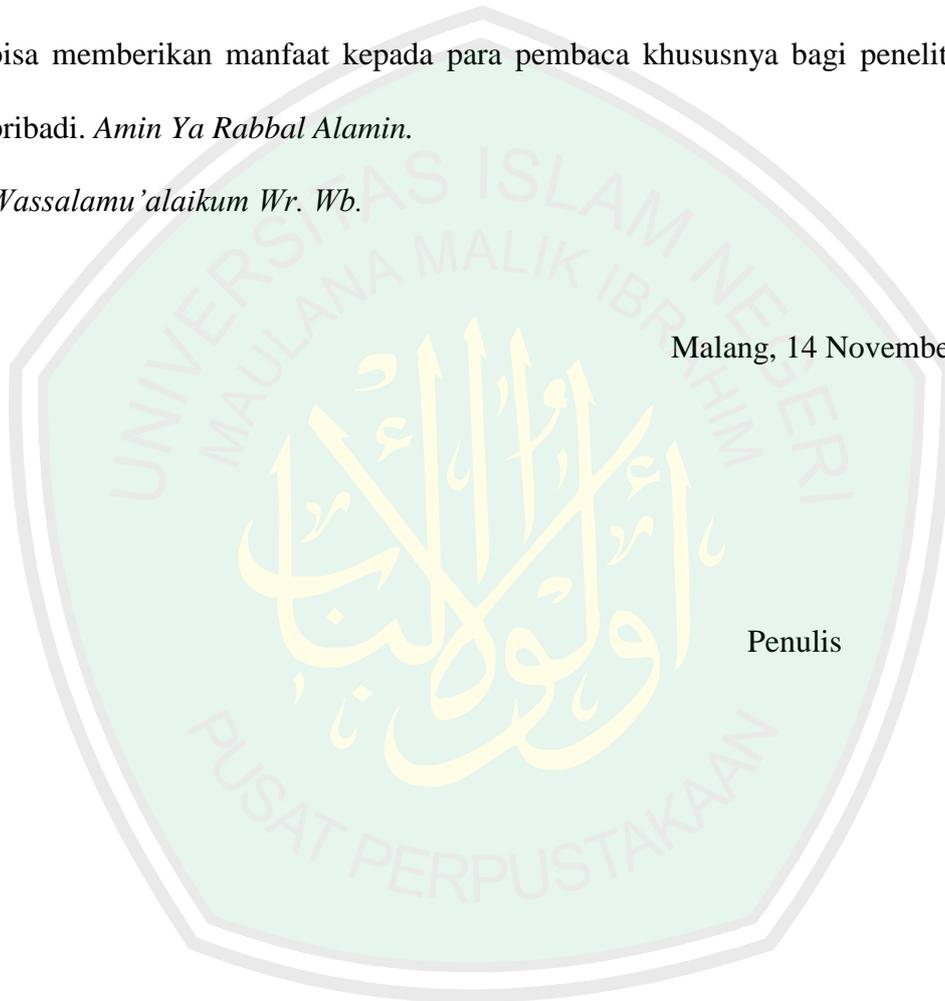
1. Fachrul Kurniawan, M.MT. Selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing, memotivasi, mengarahkan dan memberi masukan dalam pengerjaan skripsi ini.
2. Dr. Suhartono, M.Kom. Selaku dosen pembimbing II, yang telah memberikan masukan, nasehat serta petunjuk dalam penyusunan laporan skripsi ini.
3. Dr. Cahyo Crysdiyan selaku ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Segenap civitas akademika jurusan Teknik Informatika, terutama seluruh dosen, terima kasih atas segenap ilmu dan bimbingannya.
5. Semua pihak yang tidak mungkin penulis sebutkan satu-persatu, atas segala yang telah diberikan kepada penulis dan dapat menjadi pelajaran.

Peneliti menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan dan jauh dari sempurna. Semoga apa yang menjadi kekurangan bisa disempurnakan oleh peneliti selanjutnya. Peneliti juga berharap semoga skripsi ini bisa memberikan manfaat kepada para pembaca khususnya bagi peneliti secara pribadi. *Amin Ya Rabbal Alamin.*

Wassalamu'alaikum Wr. Wb.

Malang, 14 November 2014

Penulis



DAFTAR ISI

HALAMAN JUDUL.....	
HALAMAN PENGANTAR.....	
LEMBAR PERSETUJUAN.....	
LEMBAR PENGESAHAN	
HALAMAN PERNYATAAN	
MOTTO	
HALAMAN PERSEMBAHAN	
KATA PENGANTAR	
DAFTAR ISI.....	
DAFTAR TABEL.....	
DAFTAR GAMBAR	
ABSTRACT.....	
المخلص.....	
BAB I.....	
PENDAHULUAN	
1.1 Latar Belakang.....	
1.2 Identifikasi Masalah.....	
1.3 Batasan Masalah	
1.4 Tujuan Penelitian	
1.5 Manfaat Penelitian	
1.6 Sistematika Penulisan	
1.7 Metodologi Penelitian	
BAB II.....	
TINJAUAN PUSTAKA	
2.1 Permainan (<i>Game</i>)	
2.2 Jenis Game	
2.3 Komponen Permainan (<i>Component Games</i>).....	
2.4 <i>Non Playable Character (NPC)</i>	
2.5 Algoritma <i>Fisher Yates-Shuffle</i>	

2.6	Platform Android
2.7	Sensor <i>Accelerometer</i>
2.8	SQLite.....
2.9	Penelitian Terkait
BAB III	
ANALISA DAN PERANCANGAN SISTEM	
3.1	Analisa dan Perancangan Sistem
3.1.1	Keterangan Umum <i>Game</i>
3.1.2	Penampilan <i>Game</i>
3.1.3	Deskripsi Karakter
3.1.4	Finite State Machine Permainan
3.1.5	Perancangan <i>Fisher Yates Shuffle</i> pada NPC.....
3.2	Perancangan Aplikasi <i>Game</i>
3.2.1	Perancangan Antarmuka <i>Splashscreen</i>
3.2.2	Perancangan Antarmuka Menu <i>Game</i>
3.2.3	Perancangan Antarmuka Help <i>Game</i>
3.2.4	Perancangan Antarmuka <i>Game</i>
3.2.5	Perancangan Alur Permainan
3.2.6	Cara Memainkan <i>Game</i>
3.2.7	Kebutuhan <i>Smartphone</i> Pemain.....
BAB IV	
HASIL DAN PEMBAHASAN.....	
4.1	Implementasi Sistem
4.2	Implementasi Algoritma <i>Fisher Yates Shuffle</i>
4.3	Implementasi Aplikasi <i>Game</i>
4.3.1	Antarmuka <i>Splashscreen</i>
4.3.2	Antarmuka Menu <i>Game</i>
4.3.3	Scene Permainan
4.4	Uji Coba
4.4.1	Uji Coba Algoritma <i>Fisher Yates Shuffle</i>
4.4.2	Uji Coba Aplikasi.....
4.5	Integrasi <i>Game</i> Tantra Bahari dengan Islam.....

BAB V.....

PENUTUP.....

 5.1 Kesimpulan

 5.2 Saran

LAMPIRAN-LAMPIRAN.....



DAFTAR TABEL

Tabel 2.1 Contoh pengerjaan dari <i>Fisher Yates Shuffle</i>	20
Tabel 3.1 Tabel iterasi algoritma <i>Fisher Yates-Shuffle</i>	36
Tabel 3.2 Keterangan Fungsi Tombol Menu	39
Tabel 3.3 Daftar Spesifikasi <i>Smartphone</i>	42
Tabel 4.1 Keterangan Fungsi Tombol Menu Permainan	48
Tabel 4.2 Keterangan Fungsi Tombol Menu	38
Tabel 4.3 Hasil Uji <i>Shuffle Array</i>	53
Tabel 4.4 Hasil Pengujian Aplikasi	56
Tabel 4.5 Tabel pengujian <i>game</i> tantra bahari pada berbagai versi OS android	59



DAFTAR GAMBAR

Gambar 1.1 Metodologi Penelitian	9
Gambar 2.1 Hasil perbandingan algoritma <i>Shuffle</i>	19
Gambar 2.2 Arsitektur Android	22
Gambar 2.3 Sensor <i>accelerometer</i>	23
Gambar 3.1 Karakter Pemain (1)	29
Gambar 3.2 Karakter Pemain (2)	30
Gambar 3.3 Sampah Jaring	31
Gambar 3.4 Sampah Botol	31
Gambar 3.5 Sampah Kaleng	31
Gambar 3.6 Sampah Plastik	31
Gambar 3.7 Obstacle (1)	32
Gambar 3.8 Obstacle (2)	32
Gambar 3.9 Obstacle (3)	32
Gambar 3.10 FSM permainan	33
Gambar 3.11 Flowchart Algoritma <i>Fisher Yates Shuffle</i>	34
Gambar 3.12 Blok Diagram Implementasi <i>Fisher Yates Shuffle</i>	35
Gambar 3.13 Antarmuka <i>Splashscreen</i>	38
Gambar 3.14 Antarmuka Menu Game	38
Gambar 3.15 Antarmuka Help Game	40
Gambar 3.16 Antarmuka Game	41
Gambar 4.1 Tampilan <i>Splashscreen</i>	48
Gambar 4.2 Tampilan Menu utama	49
Gambar 4.3 Tampilan Menu Help	50
Gambar 4.4 Awal Permainan	51
Gambar 4.5 Tampilan Scene pada jarak 1162	51
Gambar 4.6 Halaman <i>input new score</i>	52
Gambar 4.7 Halaman <i>score game</i>	52
Gambar 4.8 Hasil Uji Coba pada <i>distance 515 (1)</i>	54
Gambar 4.9 Hasil Uji pada <i>distance 515 (1)</i>	55
Gambar 4.10 Grafik Bulat Jumlah Pengujian Game Tantra Bahari	61

ABSTRAK

Fikria, Balqis Kamalia. 2014. **Implementasi Algoritma Fisher Yates Shuffle sebagai Pengacak Posisi Non Playable Character (NPC) pada Game Tantra Bahari**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Fachrul Kurniawan, M.MT. (II) Dr.Suhartono, M.Kom.

Kata Kunci: *Non Playable Character, Accelerometer, Fisher Yates Shuffle*

Pesan moral dan ajakan berbuat kebaikan dapat disampaikan melalui banyak cara dan media. Salah satunya adalah media *game*. *Game* ini berbentuk 2D yang dimainkan dengan sensor *accelerometer*, dimana pemain menyelam ke bawah laut untuk membersihkan sampah. Tujuan dari permainan ini adalah sosialisasi gerakan *Reef Clean Up*, yakni gerakan pembersihan sampah bawah laut yang sedang diserukan oleh Yayasan Terumbu Karang Indonesia (TERANGI). Berdasarkan hasil pengujian dapat disimpulkan bahwa implementasi *Fisher Yates Shuffle* untuk menentukan posisi NPC pada *game* Tantra Bahari dapat berjalan dengan baik. Pengujian dilakukan pada perangkat *mobile* yang menggunakan *platform* Android.

ABSTRACT

Fikria, Balqis Kamalia. 2014. **Implementation of Fisher Yates Shuffle Algorithm as Position Scrambler for Non Playable Character (NPC) in Game Tantra Bahari**. Thesis. Informatics Department of Faculty of Science and Technology. Maulana Malik Ibrahim State Islamic University, Malang . Adviser: (I) Fachrul Kurniawan, M.MT. (II) Dr.Suhartono, M.Kom.

Keywords: *Non Playable Character, Accelerometer, Fisher Yates Shuffle*

Moral message and a call to do good can be delivered in many different ways and media. One is the *game* media. This *game* type is 2D, its played with sensor accelerometer where players dive underwater to clean up the trashes. The objective of the *game* is socialization Reef Clean Up movement, Reef Clean Up movement is the movement of the underwater cleanup trash that is being called by the Yayasan Terumbu Karang Indonesia (TERANGI). Based on the test results it can be concluded that the implementation of the *Fisher Yates Shuffle* to determine the position of the NPC in the *game* Tantra Marine can run well. Tests carried out on mobile devices that use the Android platform.



المخلص

فكريا بلقيس, كمليا. ٢٠١٤. تنفيذ الخوارزمية فيشر بيتس سوفيل كما جهاز لتغيير الوظيفة نون فلايبيل خركثير في الالعب التانترا بهاري. أطروحة. قسم المعلوماتية، كلية العلوم والتكنولوجيا. الجامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانج. المشرف: فحروالقورنيون الماجستير و الدكتور سوحرتونو الماجستير

الكلمات البحث: غير القابل للشخصية ، التسارع، فيشر بيتس سوفيل

يمكن أن يتم تسليم رسالة أخلاقية والدعوة لفعل الخير في العديد من الطرق المختلفة ووسائل الإعلام. واحدة هي لعبة سائل الإعلام. يعني بلعب هذه اللعبة مع 2D استتعار شكل التسارع *accelerometer* ، في أي لاعبين الغوص تحت الماء لتنظيف القمامة. والهدف من هذه اللعبة هو حركة الاجتماعية التنشئة *Reef Clean Up* ، أي لحركةتنظيف القمامة تحت الماء التي يتم استدعاء من قبل مؤسسة ترومبو كارانج الإندونيسية (TERANGI) وبناء على نتائج الاختبار يمكن الاستنتاج بأن تنفيذ فيشر بيتس سوفيل لتحديد موقف NPC في اللعبة التانترا بهاري يمكن أن تعمل جيدا. ويتم اختبار على جهاز محمول *mobile* يستخدم منصة *platform* أندرويد.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia adalah Negara kepulauan terbesar didunia yang terdiri dari 17.504 pulau besar dan kecil (Depdagri, 2006), garis pantai sepanjang 81.000 km dan letaknya yang berada di daerah ekuator membuat Indonesia kaya dengan keanekaragaman sumber daya hayati lautnya.

Nilai keanekaragaman hayati laut Indonesia yang dapat dimanfaatkan secara berkelanjutan nilainya setara dengan APBN Indonesia, menurut Rokhmin Dahuri (2006). Artinya betapa berharganya aset tersebut untuk menjadi sumber kekayaan yang membawa kepada kesejahteraan bangsa. Namun, menurut berita yang dilansir oleh media Tempo online (2011) Ekosistem laut di Indonesia sekarang ini mulai memburuk, contoh kecilnya yang terjadi pada kawasan wisata Taman Laut Nasional Bunaken (TLNB) di Manado, Sulawesi Utara, yang keindahannya sudah tersohor di jagad raya, saat ini terancam rusak. Peralnya, tumpukan sampah yang semakin hari semakin menumpuk dari sejumlah muara sungai yang sengaja dibuang oleh orang tidak bertanggung jawab di daerah Kota Manado mengalir ke Teluk Manado.

Salah satu upaya untuk mengatasi masalah sampah bawah laut tersebut adalah dengan melakukan gerakan kegiatan *Reef Clean Up*. Kegiatan *Reef Clean Up* merupakan kegiatan penyelaman yang notabenenya adalah kegiatan hiburan untuk menikmati keindahan alam bawah laut, namun

melihat fakta perairan saat ini banyak tercemar oleh sampah, penyelam juga dihimbau untuk melakukan kontribusi aktif membersihkan sampah bawah laut dengan aturan-aturan yang telah disosialisasikan. Yayasan Terumbu Karang Indonesia (TERANGI) pun mensosialisasikan kegiatan *Reef Clean Up* ini mengingat penyebab terbesar kerusakan pesisir dan menurunnya keanekaragaman ekosistem laut, khususnya di Indonesia diakibatkan oleh pencemaran sampah seperti yang dipaparkan oleh Menteri Kelautan dan Perikanan Syarif Cicip Sutardjo pada sambutan tertulisnya pada acara *Coral Triangle Day* di Nusa Tenggara Barat 9 Juni 2013 lalu.

Sosialisasi Reef Clean Up saat ini bisa dilakukan dengan banyak metode dengan beraneka media. Salah satunya dengan menggunakan media permainan atau *game* yang sekarang ini banyak digemari oleh semua orang dari berbagai kalangan. Alasan mengapa *game* sangat digemari menurut Rahmatsyam Lakoro (2009) adalah karena *game* itu sendiri mempunyai sifat dasar menantang (*challenging*), membuat ketagihan (*addicted*) dan menyenangkan (*fun*) yang mana membuat pemain *game* tersebut seakan-akan larut dalam *game* dan tidak sadar akan keadaan lingkungannya. Berdasarkan hasil penelitian Hanny Haryanto yang berjudul *Pendidikan Moral dengan Menggunakan Sistem Reward Dalam Game Imersif*, menyatakan bahwa ketika seseorang suka dan gemar bermain *game* tertentu, maka isi dan pesan yang terkandung di dalam *game* tersebut dapat mempengaruhi tingkah laku dan mempunyai efek terhadap moral dari pemain *game* tersebut, terutama jika pemain tersebut masih berusia muda,

sehingga mudah untuk menyerap hal-hal yang disampaikan oleh *game*. Oleh sebab itu, yang menjadi penyebab permasalahan menurunnya moral generasi muda terkait dengan *game* adalah bukan *game* itu sendiri, melainkan isi dan pesan yang terkandung di dalam *game* tersebut. Sebagian *game* yang ada sekarang dan yang populer di kalangan generasi muda mempunyai isi dan pesan yang negatif sehingga dapat mempengaruhi moral dan tingkah laku generasi muda sekarang. Pornografi dan kekerasan adalah hal-hal negatif yang dapat disampaikan oleh suatu *game*.

Padahal dalam agama Islam, menyampaikan hal positif dan melarang pada hal-hal negatif dalam berbagai bentuk, metode, dan media termasuk melalui media permainan, merupakan tugas manusia sebagai makhluk Allah yang beriman. Sebagaimana firman Allah swt dalam surat at-Taubah ayat 71:

وَالْمُؤْمِنُونَ وَالْمُؤْمِنَاتُ بَعْضُهُمْ أَوْلِيَاءُ بَعْضٍ يَأْمُرُونَ بِالْمَعْرُوفِ وَيَنْهَوْنَ عَنِ الْمُنْكَرِ وَيُقِيمُونَ الصَّلَاةَ وَيُؤْتُونَ الزَّكَاةَ وَيُطِيعُونَ اللَّهَ وَرَسُولَهُ أُولَئِكَ سَيَرْحَمُهُمُ اللَّهُ إِنَّ اللَّهَ عَزِيزٌ حَكِيمٌ

Artinya

"Dan orang-orang yang beriman, lelaki dan perempuan, sebagian mereka (adalah) menjadi penolong bagi sebagian yang lain. Mereka menyuruh (mengerjakan) yang ma'ruf, mencegah dari yang munkar, mendirikan shalat, menunaikan zakat dan mereka taat pada Allah dan Rasul-Nya.

Mereka itu akan diberi rahmat oleh Allah; sesungguhnya Allah Maha Perkasa lagi Maha Bijaksana" (Attaubah : 71)

Hadist tentang menganjurkan kebaikan dan mencegah perbuatan negatif juga menyebutkan:

عَنْ حُدَيْفَةَ رَضِيَ اللَّهُ عَنِ النَّبِيِّ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ قَالَ: وَالَّذِي نَفْسِي بِيَدِهِ
لَتَأْمُرَنَّ بِالْمَعْرُوفِ وَلَتَنْهَوْنَ عَنِ الْمُنْكَرِ أَوْ لَيُوشِكُنَّ اللَّهُ أَنْ يَبْعَثَ عَلَيْكُمْ عِقَابًا مِنْهُ ثُمَّ
تَدْعُونَهُ فَلَا يَسْتَجَابُ لَكُمْ (رواه الترمذی)

Artinya:

"Hudzaifah berkata bahwa Nabi SAW bersabda," Demi Allah yang jiwaku ada di tangan-Nya, kamu harus menganjurkan kebaikan dan mencegah kemunkaran, atau kalau tidak, pasti Allah akan menurunkan siksa kepadamu, kemudian kamu berdoa, maka tidak diterima doa dari kamu" (HR. Turmudzi).

Untuk itu, dalam penelitian ini penulis bermaksud membuat sebuah *game* yang mengandung nilai positif berupa ajakan berbuat kebaikan dalam upaya sosialisasi gerakan *Reef Clean Up* yang sedang diserukan oleh Yayasan Terumbu Karang Indonesia (TERANGI) dalam bentuk *game* "Tantra Bahari" dengan menggunakan algoritma *Fisher Yates-Shuffle*, yang dapat dimainkan dalam aplikasi *android mobile*.

Game Tantra Bahari ini mengangkat tema tentang misi penyelaman untuk membersihkan sampah-sampah yang mengendap di bawah laut Indonesia. Dalam permainan ini pemain juga dihimbau agar dalam misi pembersihan sampah jangan sampai malah merusak ekosistem bawah laut,

misalnya menyentuh terumbu karang atau melakukan kontak langsung dengan biota laut yang dijaga populasinya. Himbauan tersebut disiratkan dengan *action game* yang unik dan menarik untuk dimainkan. Pemilihan algoritma *Fisher Yates-Shuffle* pada permainan ini berguna untuk melakukan proses pengacakan/ *shuffling* untuk menentukan posisi semua karakter bukan pemain (*Non Playable Character*) yang terdiri atas kelompok NPC sampah, dan *obstacle*.

1.2 Identifikasi Masalah

Berdasarkan uraian latar belakang diatas, penulis akan menerapkan algoritma *Fisher Yates Shuffle* untuk menentukan posisi NPC pada *game* *Tantra Bahari* berbasis android.

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Aplikasi bertipe 2 dimensi
2. *Game* yang dibuat bersifat *single player*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Merancang sebuah *game* yang memadukan *gameplay* dan pesan moral berupa sosialisasi *Reef Clean Up* melalui media permainan petualangan (*Adventure Game*).
2. Mengimplementasikan algoritma *Fisher Yates-Shuffle* sebagai pengacak *Non Playable Character* (NPC) permainan.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah adalah terciptanya permainan yang santai dan menyenangkan sekaligus memberikan pesan moral ntuk menjaga ekosistem laut, melatih sinkronisasi rangsangan visual dan ketepatan gerakan sehingga diharapkan mampu meningkatkan kemampuan otak.

1.6 Sistematika Penulisan

Penulisan skripsi ini tersusun dalam 5 (lima) bab dengan sistematika sebagai berikut :

BAB I PENDAHULUAN

Pendahuluan membahas tentang Latar Belakang, Identifikasi Masalah, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian, Sistematika Penulisan dan Metodologi Penelitian.

BAB II KAJIAN PUSTAKA

Kajian pustaka, berisi tentang teori-teori yang melandasi penyusunan skripsi.

BAB III ANALISA DAN PERANCANGAN SISTEM

Pada bab ini berisi menganalisa kebutuhan *game* yang akan dibangun meliputi kebutuhan *software* dan langkah-langkah pembuatan *game*.

BAB IV HASIL DAN PEMBAHASAN

Menjelaskan tentang pengujian *game* yang telah diterapkan dalam pembuatan *game*.

BAB V PENUTUP

Berisi kesimpulan dan saran.

1.7 Metodologi Penelitian

Pada penelitian ini, peneliti melakukan langkah-langkah yang harus dilalui. Berikut tahapan prosedur penelitian:

1. Merumuskan Hipotesis

Pada tahap ini melakukan anggapan yang sifatnya sementara tentang objek yang diteliti.

2. Menentukan Desain Penelitian

Desain penelitian ini menggunakan model matematika.

3. Mengumpulkan Data

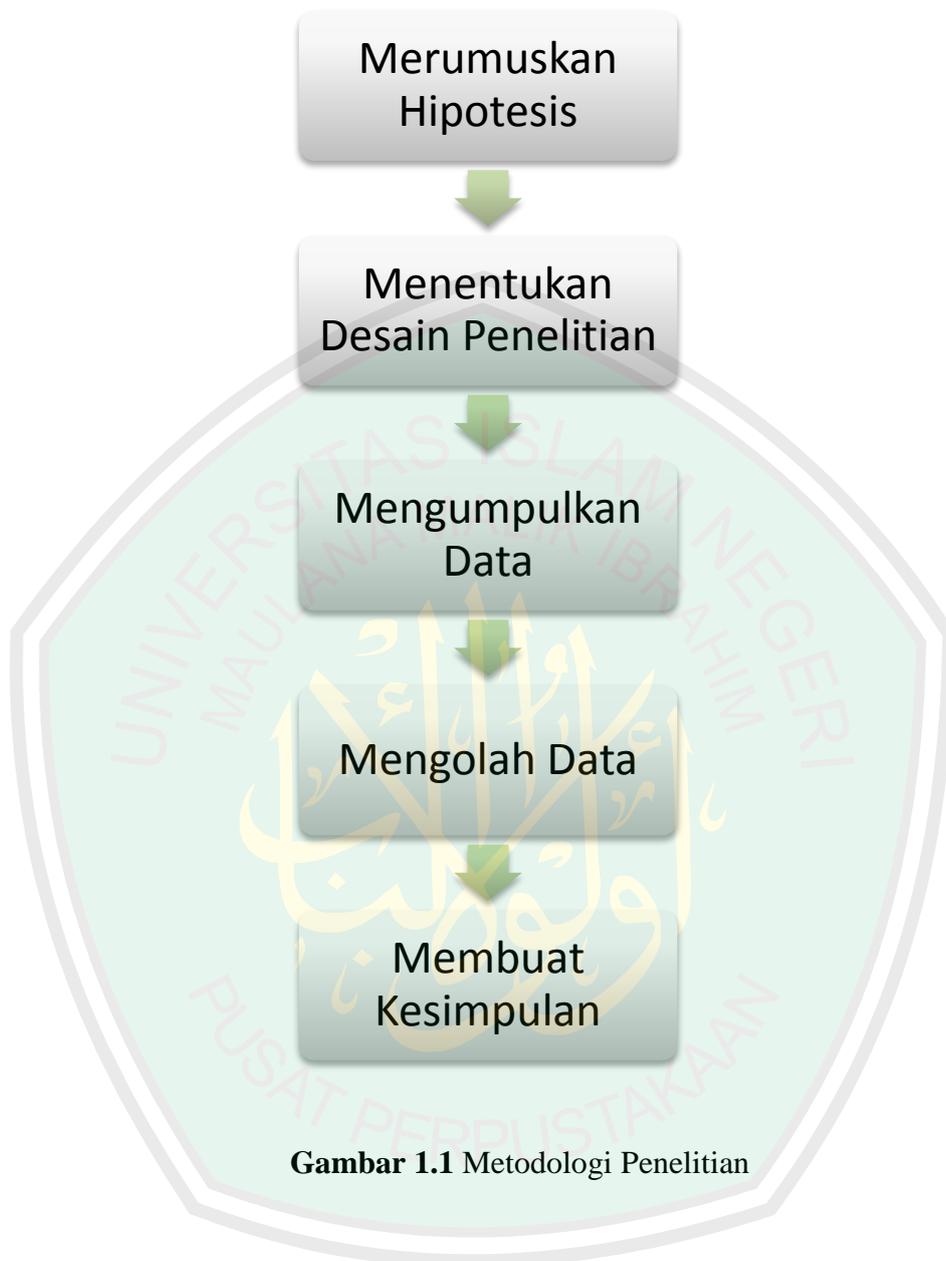
Pada tahap ini akan dilakukan pengumpulan data dari literatur-literatur terkait dengan penelitian ini. Literatur didapatkan dari buku, jurnal, atau skripsi terdahulu.

4. Mengolah Data

Setelah data dikumpulkan akan dilakukan analisis terhadap hasil pengumpulan data dari literatur yang telah didapatkan.

5. Membuat Kesimpulan

Pada tahap ini peneliti membuat kesimpulan yang sesuai dengan hipotesis yang diajukan dan membuat laporan hasil dari penelitian.



Gambar 1.1 Metodologi Penelitian

BAB II

TINJAUAN PUSTAKA

1.1 Permainan (*Game*)

Game adalah media untuk melakukan aktifitas bermain. Aktifitas bermain merupakan suatu aktifitas yang meliputi pemecahan masalah yang menjadi tantangan dari *game* tersebut, dengan mengikuti suatu aturan tertentu.

Game adalah program komputer sangat kompleks yang merangsang otak untuk melakukan serangkaian tugas kognitif dan menghasilkan tingkat pemikiran yang lebih tinggi (Beck dan Wade, 2006).

Saat ini *game* telah berkembang pesat, yang awalnya hanya terdapat pada konsol sekarang telah merambah pada computer dan telepon selular. *Game* yang terdapat pada telepon selular disebut sebagai *mobile game*.

Mobile game adalah *game* yang dimainkan pada perangkat genggam, seperti ponsel dan PDA dengan fungsionalitas komunikasi nirkabel (Jeong dan J.Kim, 2009).

1.2 Jenis *Game*

Menurut Samuel Henry (2010), jenis-jenis *game* menurut format atau gayanya adalah sebagai berikut :

1. *Maze Game*

Jenis *game* ini biasanya menggunakan maze sebagai *setting* atau latar *game*. Jenis *game* maze ini termasuk jenis *game* yang paling awal muncul. Contoh *game* ini adalah *game* Pacman dan Digger.

2. *Board Game*

Jenis *game* ini sama dengan *game* board tradisional seperti monopoli. Hanya saja permainan tradisional ini dimainkan melalui komputer.

3. *Card Game*

Jenis *game* kartu juga tidak jauh berbeda dari *game* tradisional aslinya. Namun, tampilannya lebih bervariasi dari versi tradisional. *Game* ini juga termasuk *game* yang awal muncul. Contoh *game* ini adalah Solitaire dan Hearts.

4. *Battle Card Game*

Contoh *game* ini yang populer yaitu *Battle Card* Pokemon.

5. *Quiz Game*

Game jenis ini merupakan *game* dengan bentuk kuis. Contoh *Quiz Game* yang pernah beredar yaitu *game* kuis *Who Wants to Be Millionaire*.

6. *Puzzle Game*

Jenis *game* ini memberi tantangan dengan cara menjatuhkan atau melenyapkan sesuatu dari sisi atas ke bawah atau dari kiri ke kanan. Contoh *game* ini adalah Tetris.

7. *Shoot Them Up*

Game jenis ini biasanya musuh berbentuk pesawat atau bentuk lain yang datang dari arah kanan, kiri, atau atas yang harus kita tembak sebanyak dan secepat mungkin. Dulu *game* ini berbentuk dua dimensi (2D), tetapi sekarang sudah berkembang dan menggunakan efek tiga dimensi (3D).

8. *Side Scroller Game*

Saat pertama kali muncul *game* ini berbentuk 2D. Sekarang sudah banyak yang dibuat dengan efek 3D. Pada *game* jenis ini pemain diharuskan bergerak searah di alur yang disediakan. Dia diharuskan untuk berjalan, melompat, merunduk serta

menghindari rintangan-rintangan. Contoh *game* ini yang populer yaitu Mario Bros dan Prince of Persia.

9. *Fighting Game*

Jenis *game* ini sesuai dengan namanya berisi tentang pertarungan. Contoh *game* ini yaitu Street Fighter, Samurai Showdown, Virtual Fighter dan Kungfu.

10. *Racing Game*

Adalah *game* tentang balapan. Contoh *game* ini yaitu Need for Speed Underground dan Toca Race Driver.

11. *Turn-Based Strategy*

Game Pemain dalam *game* ini melakukan gerakan setelah pemain lain melakukan gerakan jadi saling bergantian. Contoh *game* yang terkenal adalah Empire dan Civilization.

12. *Real-Time Strategy Game*

Game ini seperti *game* Turn- Based Strategy (RTS), namun pada *game* ini pemain tidak perlu menunggu pemain lain. Pemain tercepatlah yang akan menang. Contoh *game* ini yaitu Warcraft.

13. SIM, *game* genre ini merupakan bentuk permainan simulasi. Di sini pemain membangun sebuah area, kota, negara atau koloni. Contoh *game* ini yaitu Ship Simulator, Train Simulator, dan Crane Simulator.

14. *First Person Shooter*, disebut *first person shooter* karena pandangan pemain adalah pandangan orang pertama. Banyak baku tembak dan *game* ini mengutamakan kecepatan gerakan. Contoh *game* ini yaitu *game* Counterstrike dan Doom.

15. *First Person Shooter 3D Vehicle Based*, game ini sama dengan FPS hanya saja pandangan pemain bukan dari orang pertama, tetapi dari kendaraan atau mesin yang digunakan. Kendaraan itu bisa berupa tank atau kapal.
16. *Third person 3D Games*, game ini juga hampir sama dengan FPS hanya sudut pandang pemain merupakan sudut pandang orang ketiga.
17. *Role Playing Game*, jenis game ini pemainnya memainkan sebuah tokoh atau karakter. Biasanya ada alur cerita yang harus dijalankan. Contoh game ini adalah Legacy of Kain, Blade of Sword, dan Beyond Divinity.
18. *Educational and Edutainment*, game ini lebih mengacu pada isi dan tujuan dari game. Game ini bertujuan memancing minat belajar anak sambil bermain. Contoh game ini adalah game Bobby Bola.
19. *Sports*, jenis game ini memiliki tema olahraga. Game yang menyetengahkan genre olahraga disebut sport game
20. *Adventure Game*, adventure game merupakan genre game petualangan Game jenis ini adalah game dengan mengambil konsep petualangan. Pemain berjalan menuju ke suatu tempat dan disepanjang perjalanan pemain menemukan hal-hal baru untuk dieksplorasi. Dalam game petualangan pemain dituntut kemampuan berfikirnya untuk menganalisa tempat secara visual, memecahkan teka-teki maupun menyimpulkan rangkaian peristiwa dan percakapan karakter, menggunakan benda-benda yang tepat dan diletakan di tempat yang tepat. Gameplay jenis ini adalah keharusan player memecahkan bermacam-macam puzzle melalui interaksi dengan orang lingkungan dalam game tersebut.

Contoh game ini yaitu Sam and Max atau Beyond and Evil, Super Mario.

1.3 Komponen Permainan (*Component Games*)

Komponen-komponen penting yang harus diperhatikan dalam pembuatan *game* (Rogers, 2012) yaitu:

1. *Opening (Splash) Screen* (Pembukaan)

Splash Screen ini opsional dalam pembuatan *game* tapi lebih baik di masukkan kedalam *game* yang dibuat. Hal ini dilakukan agar pengguna mengetahui bahwa *game* ini berjalan seperti seharusnya.

2. *Menu Screen* (Tampilan Menu)

Sejak *game* ini berjalan, biasanya pengguna akan mencari menu untuk mematikan/menyalakan suara atau mencari cara bermain *game* yang dimainkan.

3. *Music* (Musik)

Komponen ini merupakan komponen yang penting karena dapat membangun suasana pengguna memainkan *game* yang dibuat.

4. *Sound Effects* (Efek Suara)

Sound effects dapat membuat *game* menjadi lebih menyenangkan. Ketika terdapat dua objek bertubrukan pengguna dapat mendengar suara seperti *clang*, *thud*, atau *boing*.

5. *Time* (Waktu)

Kebanyakan *game* menggunakan waktu untuk menyelesaikan *game* seperti menyelesaikan *game* teka-teki berdasarkan waktu yang dibutuhkan untuk memecahkan teka-teki.

6. *Lives* (Nyawa)

Game harus mempunyai tantangan agar menjadi menyenangkan, sehingga pemain dapat gagal dalam bermain. Membunuh *player* adalah cara yang pantas dalam memberikan kegagalan. Beberapa *game* memberikan pemain beberapa nyawa per sesi dan juga yang hanya memiliki nyawa.

7. *Obstacles* (Rintangan)

Setiap *game* memiliki *obstacle* yang berbeda-beda. Untuk mencapai tujuan pemain harus bisa melewati *obstacle* yang diberikan oleh *game*.

8. *Levels* (Tingkatan)

Dalam sebuah *game* sangat penting untuk memberikan berbagai tantangan, sehingga pemain dapat mulai bermain dengan tantangan yang mudah dan secara bertahap berjalan menuju tantangan yang lebih tinggi. Tantangan tergantung dengan tingkatan yang dihadapi pemain. Dengan tingkatan dapat memberikan keterampilan pemain dalam bermain.

9. *Adversaries* (Lawan/Musuh)

Lawan dalam permainan kadang-kadang disebut sebagai entitas. Karakter ini adalah penjahat yang harus dihadapi pemain. Lawan berbeda dengan rintangan dalam mengambil tindakan terhadap pemain. Hambatan sedikit lebih pasif ketimbang lawan dalam mengambil tindakan.

10. *Player* (Pemain)

Pemain adalah komponen paling penting dari setiap *game*. Intinya adalah untuk menjaga pemain untuk terlibat dan tertarik sehingga pemain akan terus bermain *game* yang dimainkan.

11. *Scenes* (Adegan)

Setiap adegan memiliki grafis latar belakang yang tidak berubah banyak (meskipun titik pandang pemain mungkin berubah).

1.4 *Non Playable Character (NPC)*

NPC model berarti obyek dinamis yang tidak menjadi subordinat atau dibawah kontrol pemain, obyek ini membuat keputusan secara mandiri dan beroperasi di wilayah virtual dalam *game* (Chong-HanKim dkk, 2007)

Menurut Craig W. Reynolds, *Autonomous character* adalah jenis *Otonomous agent* yang ditujukan untuk penggunaan komputer animasi dan media interaktif seperti *games* dan *virtual reality*. Agen ini mewakili tokoh dalam cerita atau permainan dan memiliki kemampuan untuk improvisasi tindakan mereka. Ini adalah kebalikan dari seorang tokoh dalam sebuah film animasi, yang tindakannya ditulis di muka, dan untuk avatar dalam sebuah permainan atau *virtual reality*. tindakan yang diarahkan secara *real time* oleh pemain. Dalam permainan, karakter otonom biasanya digunakan untuk NPC (*Non-Player Character*).

Autonomous character harus menggabungkan aspek robot otonom dengan beberapa keahlian dari manusia improvisasi aktor dalam teater. Karakter-karakter ini biasanya tidak *real robot*, dan jelas bukan aktor manusia, tetapi berbagi beberapa properti dari masing-masing. Sebuah agen dapat reaktif (naluriah, didorong oleh stimulus) atau dapat deliberatif (intellectual dalam pengertian klasik AI). Agen otonom dapat menangani secara eksklusif dengan abstrak informasi (*softbot*, *knowbot*, atau *information agent*) atau dapat diwujudkan dalam fisik manifestasi (yang khas robot industri atau kendaraan yang

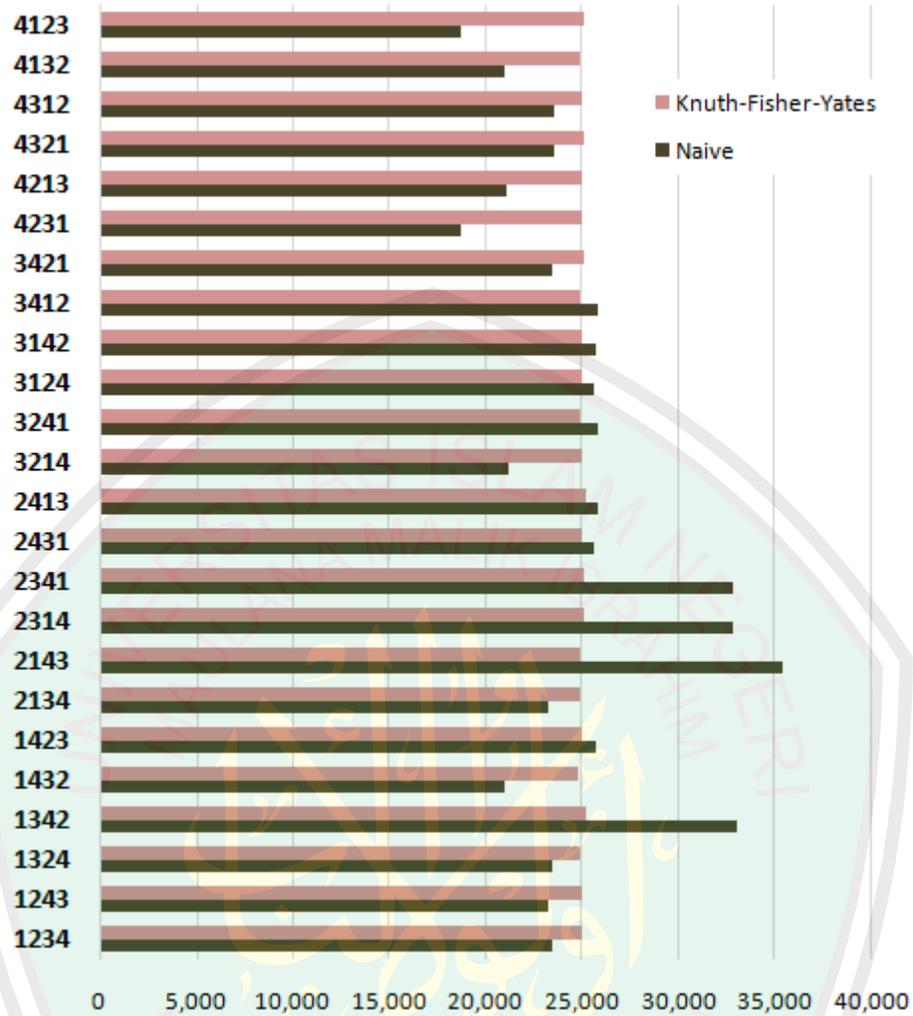
otonom). Kombinasi terletak, reaktif, dan diwujudkan menetapkan beberapa kelas yang berbeda agen otonom.

1.5 Algoritma *Fisher Yates-Shuffle*

Fisher-Yates *Shuffle* (yang dinamakan sesuai dengan pengembangnya yaitu Ronald Fisher and Frank Yates) metode ini digunakan untuk merandom posisi atau input (list). Posisi Permutasi dihasilkan oleh metode atau algoritma ini muncul dengan propabilitas yang sama (En.algoritmy.net).

Metode dasar dan versi asli dari algoritma *Fisher Yates Shuffle* ini di publikasikan pada tahun 1938 yang dideskripsikan pada buku “*Statistical tables for biological, agricultural and medical research*”, didasari pada iteratif elemen dari daftar input dan menuliskannya ke daftar keluaran kedua (pendekatan ini dimaksudkan untuk dapat dilakukan oleh manusia dengan kertas dan pensil).

Algoritma ini dinyatakan bias karena permutasi yang dihasilkan oleh algoritma ini muncul dengan probabilitas yang sama, hal ini dibuktikan dengan percobaan mengacak suatu set kartu yang dilakukan berulang – ulang seperti yang bisa dilihat pada ***gambar 2.1***.



Gambar 2.1 Hasil perbandingan algoritma *Shuffle*

Pada gambar 2.1 dilakukan tes terhadap 4 kartu, yang kemudian diacak sebanyak 600.000 kali. Keterangan pada sumbu Y menunjukkan permutasi/kemungkinan kombinasi kartu yang muncul, sedangkan keterangan pada sumbu X menunjukkan jumlah kemunculan kombinasi itu muncul. Warna pink menunjukkan hasil dari algoritma Fisher-Yates *Shuffle* sedangkan warna hijau tua menunjukkan algoritma Naïve *Shuffle*. Hasilnya algoritma Fisher-Yates menghasilkan nilai yang hampir sama untuk setiap kemungkinan kombinasi kartu, sedangkan pada algoritma Naïve kombinasi kartu tertentu muncul jauh

lebih sering dan sebagian lagi muncul lebih sedikit dibanding kombinasi kartu lain (blog.codinghorror.com).

Metode Fisher-Yates secara umum adalah sebagai berikut :

1. Deklarasi jumlah elemen dalam array
2. Ambil satu elemen secara acak sesuai dengan jumlah elemen dalam array dan range yang diperbolehkan .
3. Lakukan selama kondisi dari elemen yang masih ada terpenuhi
4. Lalu Ganti elemen yang lama dengan elemen yang sudah di acak jika langkah ke-2 masih terpenuhi.

Pada versi modern digunakan sekarang, angka yang terpilih tidak dicoret, tetapi posisinya ditukar dengan angka terakhir dari angka yang belum terpilih.

Pada **Tabel 2.1** berikut ini adalah contoh pengerjaan dari versi modern. Range adalah jumlah angka yang belum terpilih, roll adalah angka acak yang terpilih, scratch adalah daftar angka yang belum terpilih, dan result adalah hasil permutasi yang akan didapatkan.

Tabel 2.1 Contoh pengerjaan dari *Fisher Yates Shuffle*

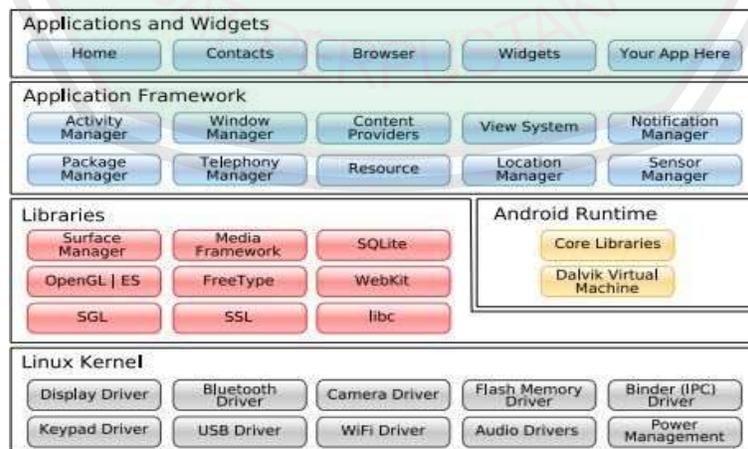
Range	Roll	Stratch	Result
		1 2 3 4 5 6 7 8	
1-8	6	1 2 3 4 5 8 7	6
1-7	2	1 7 3 4 5 8	2 6
1-6	6	1 7 3 4 5	8 2 6
1-5	1	5 7 3 4	1 8 2 6
1-4	3	5 7 4	3 1 8 2 6

1-3	3	5 7	4 3 1 8 2 6
1-2	1	7	5 4 3 1 8 2 6

Permutasi yang didapatkan dari pengerjaan *Fisher Yates Shuffle* pada **Tabel 1** diatas adalah 7 5 4 3 1 8 2 6

1.6 Platform Android

Android adalah sistem operasi untuk ponsel yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel atau *smartphone*. Kemudian untuk mengembangkan android dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile dan Nvidia. Android memiliki arsitektur sistem sebagaimana **gambar 2.2** berikut :



Gambar 2.2 Arsitektur Sistem Android

Versi Android diawali dengan dirilisnya Android beta pada bulan November 2007. Versi komersial pertama, Android 1.0, dirilis pada September 2008. Android dikembangkan secara berkelanjutan oleh Google dan Open Handset Alliance (OHA), yang telah merilis sejumlah pembaruan sistem operasi ini sejak dirilisnya versi awal.

Sejak April 2009, versi Android dikembangkan dengan nama kode yang dinamai berdasarkan makanan pencuci mulut dan penganan manis. Masing-masing versi dirilis sesuai urutan alfabet, yakni Cupcake (1.5), Donut (1.6), Eclair (2.0–2.1), Froyo (2.2–2.2.3), Gingerbread (2.3–2.3.7), Honeycomb (3.0–3.2.6), Ice Cream Sandwich (4.0–4.0.4), Jelly Bean (4.1–4.3), dan KitKat (4.4+). Pada tanggal 3 September 2013, Google mengumumkan bahwa sekitar 1 miliar perangkat seluler aktif di seluruh dunia menggunakan OS Android (phandroid.com). Pembaruan utama terbaru versi Android adalah Lollipop 5.0, yang dirilis pada 3 November 2014 (id.wikipedia.org).

1.7 **Sensor Accelerometer**

Sensor *accelerometer* adalah sensor yang dapat mengukur percepatan, mendeteksi dan mengukur getaran (vibrasi), dan mengukur percepatan akibat gravitasi (Seifert K dan Camacho, 2007).

Pada *smarthphone* android, *accelerometer* dipakai untuk mendeteksi posisi dari handphone apakah dalam mode landscape atau portrait, manfaat utama adalah menu dan aplikasi di *smartphone* akan menyesuaikan posisi dari *smartphone*. Manfaat lain biasanya dipakai untuk permainan *game*, misal *game* mobil dimana kita cukup memiringkan handphone saat kita mau belok kanan atau kiri. Manfaat lain dipakai untuk mengganti lagu atau mengambil screenshoot hanya dengan menggoyangkan *smartphone*.



Gambar 2.3 Sensor *accelerometer*

1.8 SQLite

SQLite adalah *Relational Database Management Server* (RDBMS) alternatif yang bersifat *portable* (tidak memerlukan proses instalasi), cepat, gratis, dan didukung oleh banyak bahasa pemrograman (Didik, 2003)

Tidak seperti pada paradigma client-server umumnya, Inti SQLite bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemrograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi overhead, latency times, dan secara keseluruhan lebih sederhana. Seluruh

elemen basisdata (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah file. Kesederhanaan dari sisi disain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai (Nazruddin, 2007).

Keunggulan SQLite antara lain :

1. *Portable* tidak perlu proses instalasi, cukup menggunakan satu file `sqlite3.exe`;
2. *Flat file* (satu database satu file) ;
3. Mendukung *transaction* dan *view*;
4. Sangat cepat, karena berupa *flat file*;
5. Menggunakan *Query Language* yang mirip dengan RDBMS pada umumnya.

Salah satu *library* yang terdapat dalam database SQLite adalah fungsi *command-line sqlite3* (`sqlite3.exe`) yang memungkinkan user untuk berinteraksi dan menjalankan perintah SQL di SQLite.

1.9 Penelitian Terkait

Berikut ini merupakan beberapa penelitian terdahulu yang berkaitan dengan penelitian yang akan dilakukan :

Yenni Kusumawati (2004) dari Universitas Kristen Duta Wacana tentang “*Metode Pengacakan Fisher Yates-Shuffle untuk Game Puzzle Berbasis J2me*” Penelitian ini menghasilkan sebuah *game puzzle* berbasis *J2ME* yang mengimplementasikan metode *Fisher Yates-Shuffle*, *game* ini menyediakan solusi secara otomatis dalam prosesnya menggunakan metode *Trial and Error*, dan dapat menyimpan *highscore* karena mengimplementasikan *Record Management System* dan *Double Buffering*. Dari penelitian

ini diperoleh kesimpulan yakni algoritma *Fisher Yates-Shuffle* memiliki keunggulan, bahwa dalam proses iterasinya tidak dihasilkan kemungkinan yang terulang, waktu yang dibutuhkan juga lebih sedikit dibanding dengan metode pengacakan biasa. Selain itu metode ini juga bekerja dengan penggunaan memori yang minimal.

Puja Pramudya (2007) dengan judul penelitian “*Game Tebak Kartu dengan Windows Multipoint SDK*”. Permainan pada penelitian ini dibuat pada Microsoft® Multipoint™ SDK dengan konsep *multiplayer*. Algoritma *Fisher Yates-Shuffle* diimplementasikan untuk mengacak kartu yang posisinya akan selalu berubah setiap kali pergantian soal.

Ade-Ibijola, Abejide Olu (2012) pada penelitian dengan judul “*A Simulated Enhancement of Fisher-Yates Algorithm for Shuffling in Virtual Card Games using Domain-Specific Data Structures*”. Permainan ini menggunakan algoritma Fisher Yates-*Shuffle* untuk mengacak kartu yang disempurnakan dengan struktur data. Implementasi algoritma menghasilkan permutasi yang memuaskan, dengan kompleksitas dan kecepatan yang sama bagusnya.

Antony Susanto dan Henky Honggo (2013) dari STMIK GI MDP dengan judul “*Perancangan Ujian Online pada STMIK GI MDP Berbasis Web*”. Aplikasi ujian berbasis web ini mengimplementasikan algoritma *Fisher-Yates Shuffle* yang berfungsi untuk mengacak soal dan algoritma *Levenshtein Distance* yang berfungsi untuk membandingkan jawaban pada saat pengoreksian. Pembuatan aplikasi akan menggunakan bahasa pemrograman PHP dan MySQL untuk pembuatan database dan metodologi yang digunakan adalah pendekatan prototype. Kesimpulan dari penelitian ini adalah (1) Penggunaan algoritma *Fisher-Yates Shuffle* untuk pengacakan soal dan jawaban dapat membuat mahasiswa mendapatkan urutan soal dan jawaban pada soal pilihan ganda yang

berbeda-beda. (2) Penggunaan algoritma Levenshtein Distance dapat membantu mengurangi kesalahan ketik mahasiswa dalam pengetikan jawaban pada format soal jawaban pendek.

Penelitian oleh Mahasiswa dari Universitas Komputer Indonesia yaitu Muhammad Ilham Rizqyawan pada tahun 2013. Penelitiannya ialah membuat aplikasi *Game Edukasi Adventure untuk Pengenalan Teori Musik Berbasis Desktop*. Penelitian ini menggunakan algoritma *Fisher Yates Shuffle* sebagai pengacak soal tentang teori musik yang diimplementasikan pada permainan berbasis desktop.

Dari penelitian tersebut, penulis mengambil referensi bagaimana mengimplementasikan metode *Fisher Yates Shuffle* pada permainan Tantra Bahari.

BAB III

ANALISA DAN PERANCANGAN SISTEM

3.1 Analisa dan Perancangan Sistem

Game yang dibangun adalah *game single player* yang cara memainkannya adalah dengan menggunakan sensor *accelerometer*. Terdapat beberapa karakter pada *game* ini yang di desain agar *game* sangat menarik dimainkan. Karakter tersebut berupa pemain utama dan beberapa karakter NPC (*Non Player Character*). NPC tersebut diacak menggunakan *Fisher Yates Shuffle*, dan diberikan perilaku yang berbeda-beda agar *game* yang dimainkan jadi lebih menarik.

3.1.1 Keterangan Umum *Game*

Game merupakan *game 2D*, dimana untuk memainkannya pemain dapat menggerakkan *gadget/smartphone* ke atas atau ke bawah sesuai posisi *player* yang diinginkan. Dalam *game* ini pemain harus mengumpulkan sampah dan harus menghindari *obstacle*.

Sedangkan untuk objek penelitiannya, yaitu posisi NPC tersebut. Pada *game* ini NPC dibagi dua yaitu NPC sampah dan NPC *obstacle*. Pemain harus mengumpulkan sampah sebanyak-banyaknya untuk menambah *score*, serta harus menghindari *obstacle* agar permainan tidak *berakhir (game over)*.

3.1.2 Penampilan *Game*

Secara umum, *game* ini dibangun dengan grafis 2Dimensi. Semua objek yang ada pada *game* ini dibangun dengan objek 2Dimensi baik *background*, maupun karakternya.

3.1.3 Deskripsi Karakter

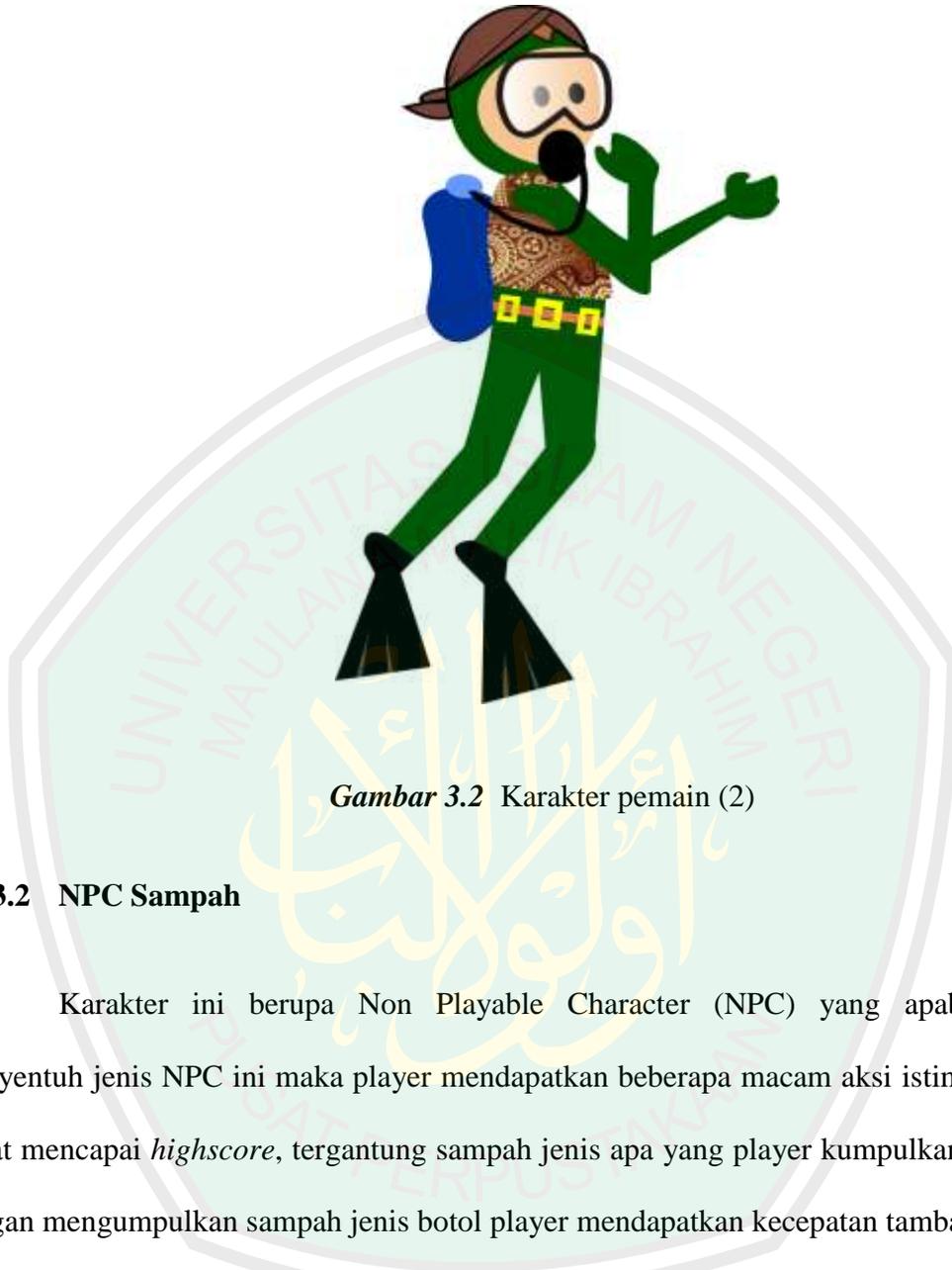
Terdapat beberapa karakter yang dibangun pada *game* ini, yaitu:

3.1.3.1 Karakter Pemain

Karakter ini yang memerankan seorang penyelam bernama Tantra yang diberi misi untuk mengumpulkan sampah bawah laut sebanyak-banyaknya untuk mendapatkan nilai tertinggi dan menempuh jarak terjauh dalam membersihkan sampah.



Gambar 3.1 Karakter pemain (1)

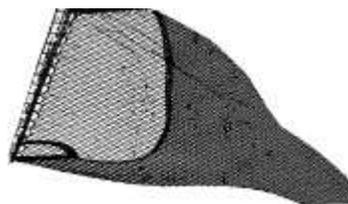


Gambar 3.2 Karakter pemain (2)

3.1.3.2 NPC Sampah

Karakter ini berupa Non Playable Character (NPC) yang apabila player menyentuh jenis NPC ini maka player mendapatkan beberapa macam aksi istimewa untuk dapat mencapai *highscore*, tergantung sampah jenis apa yang player kumpulkan. Misalkan dengan mengumpulkan sampah jenis botol player mendapatkan kecepatan tambahan.

Rancangan NPC sampah dapat dilihat pada gambar 3.1, gambar 3.2, gambar 3.3, gambar 3.4, gambar 3.5, dan gambar 3.6.



Gambar 3.3 Sampah jaring



Gambar 3.4 Sampah botol



Gambar 3.5 Sampah kaleng



Gambar 3.6 Sampah plastik

3.1.3.3 NPC Obstacle

Karakter ini berupa terumbu karang, ikan, dan penyusut hijau. Apabila player mengenai NPC *obstacle* ini maka nyawa (*life*) pemain akan berkurang. Dan ketika nyawa pemain habis maka permainan akan berakhir, kemudian *score* akan di tampilkan.

Rancangan karakter NPC obstacle dapat dilihat pada **gambar 3.7**, **gambar 3.8**, dan **gambar 3.9**.



Gambar 3.7 Obstacle (1)

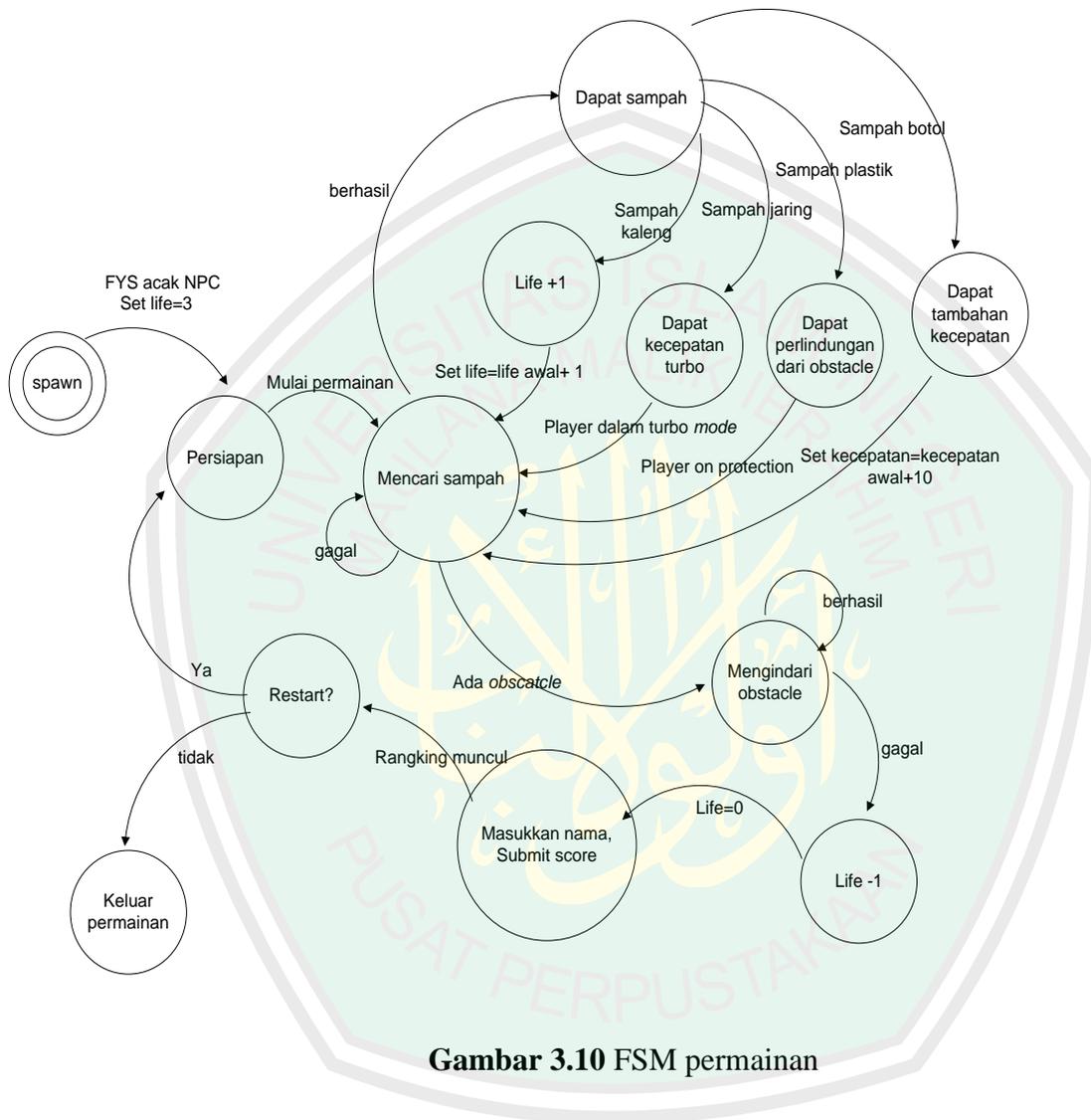


Gambar 3.8 Obstacle (2)



Gambar 3.9 Obstacle (3)

3.1.4 Finite State Machine Permainan

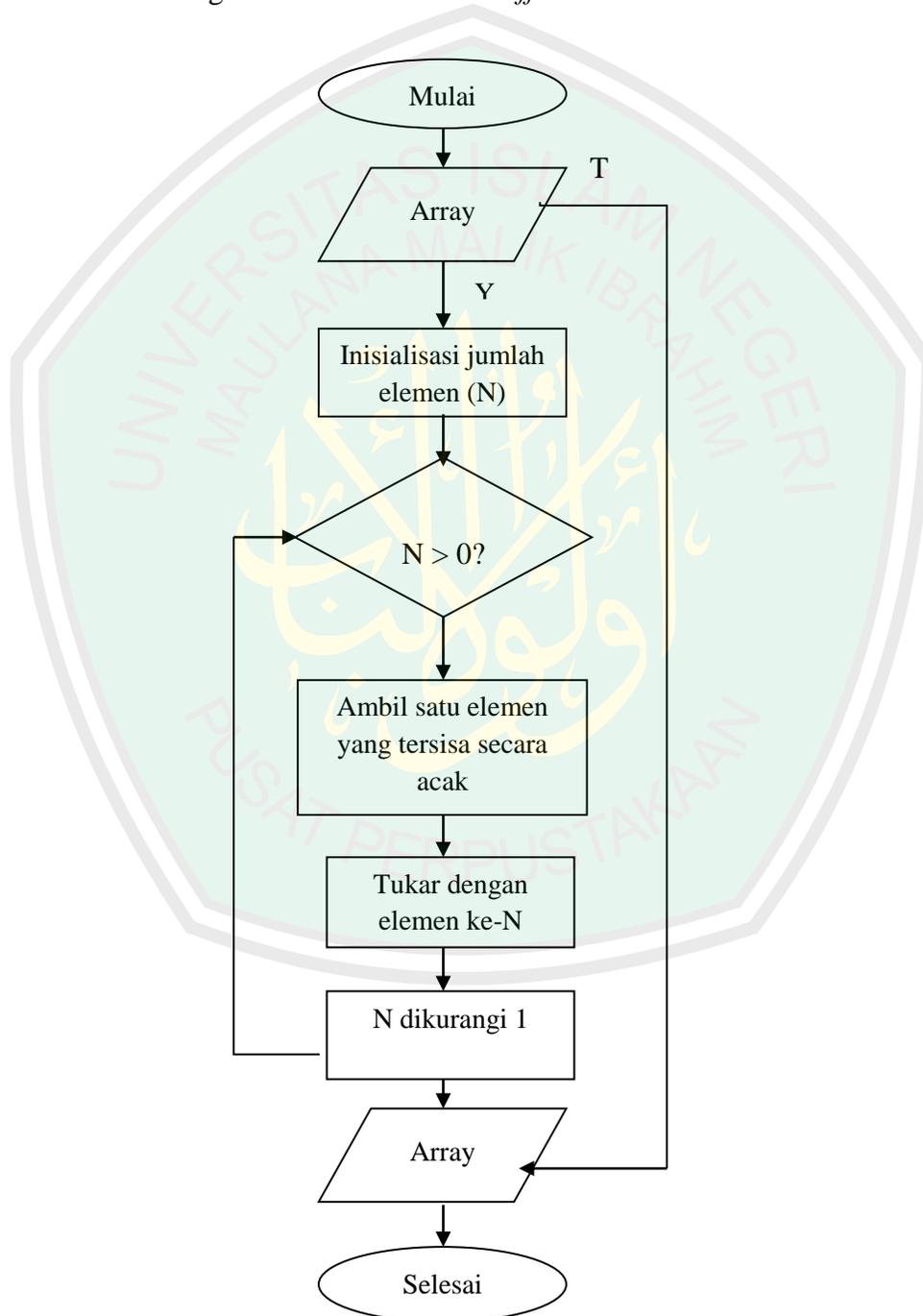


Gambar 3.10 FSM permainan

3.1.5 Perancangan *Fisher Yates Shuffle* pada NPC

Penerapan algoritma terdapat pada NPC. *Fisher Yates Shuffle* diterapkan untuk mengacak/ *Shuffle* posisi NPC.

A. Perencanaan algoritma *Fisher Yates Shuffle*

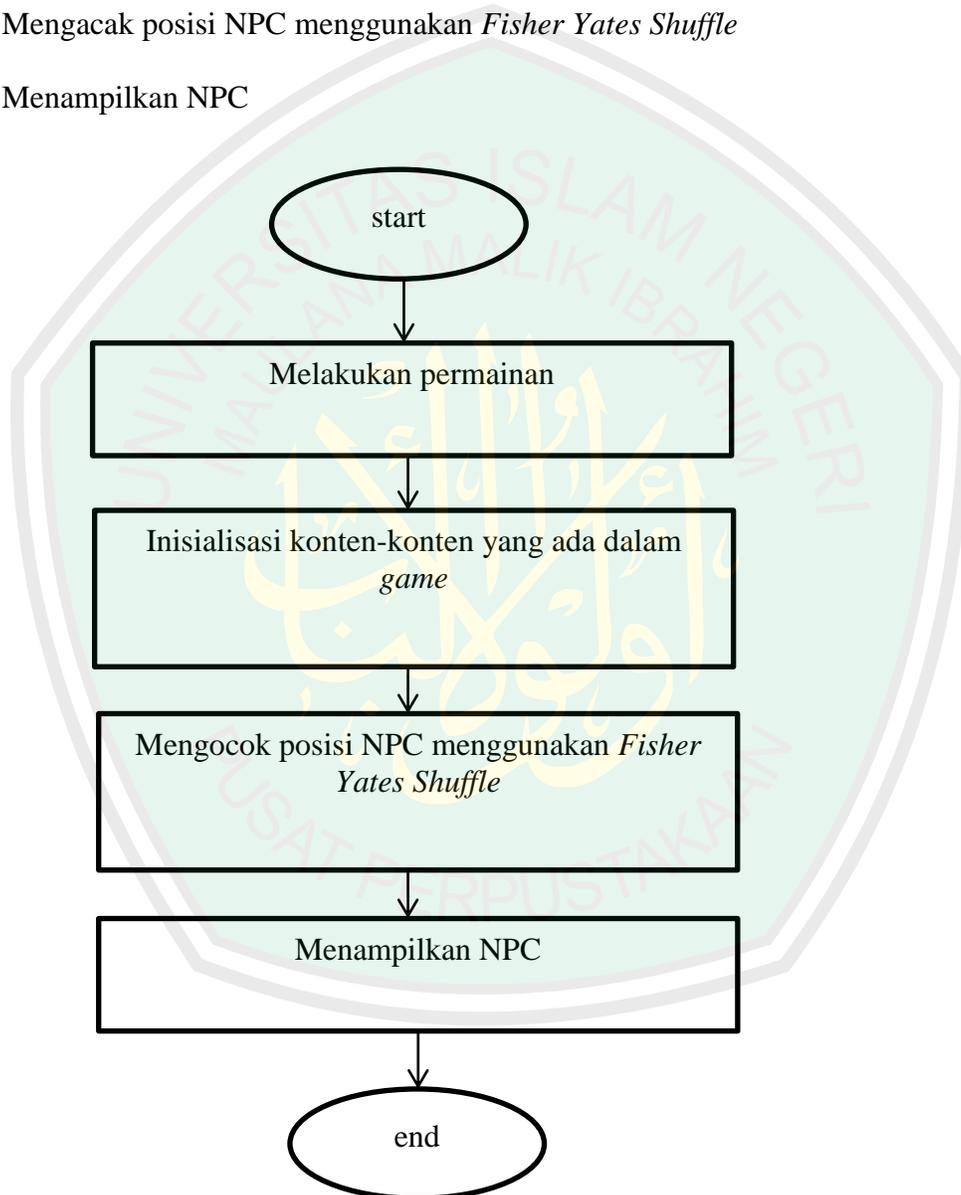


Gambar 3.11 Flowchart Algoritma *Fisher Yates-Shuffle*

B. Implementasi *Fisher Yates Shuffle*

Algoritma *Fisher Yates Shuffle* diimplementasikan dengan alur sebagai berikut :

1. Melakukan permainan
2. Inisialisasi konten-konten yang ada dalam *game*
3. Mengacak posisi NPC menggunakan *Fisher Yates Shuffle*
4. Menampilkan NPC



Gambar 3.12 Blok Diagram Implementasi *Fisher Yates Shuffle*

C. Simulasi Manual Perhitungan *Fisher Yates Shuffle*

Dalam game “Tantra Bahari” misalkan terdapat 8 item NPC yang akan diacak. Maka didapatkan panjang array (N) =8.

Langkah – langkah pengacakan dengan *Fisher-Yates Shuffle* adalah sebagai berikut:

- Ambil satu elemen secara acak (*k*). Nilai *k* yang boleh diambil adalah nilai N (panjang array) yang tersisa dikurangi 1 (N-1)
- Tukarkan array (*k*) yang di ambil dari indeks ke-*k* dengan array *t*/ array ke-N yang diambil dari elemen saat ini (N).
- Ulangi selama masih ada elemen yang tersisa.

Tabel iterasi dari proses manual pengacakan menggunakan algoritma *Fisher Yates-Shuffle* dengan jumlah array N=8 dideskripsikan sebagaimana pada **tabel 2**

Tabel 3.1 Tabel iterasi algoritma *Fisher Yates-Shuffle*

N	k	Array [k]	t/array [N]	Isi array yang telah di <i>Shuffle</i>
8	5	5	8	0, 1, 2, 3, 4, 8, 9, 7, 5
7	1	1	7	0, 7, 2, 3, 4, 8, 9, 1, 5
6	2	2	9	0, 7, 9, 3, 4, 8, 2, 1, 5
5	4	4	8	0, 7, 9, 3, 8, 4, 2, 1, 5
4	1	7	8	0, 8, 9, 3, 7, 4, 2, 1, 5
3	2	9	3	0, 8, 3, 9, 7, 4, 2, 1, 5
2	1	8	3	0, 3, 8, 9, 7, 4, 2, 1, 5
1	0	0	3	3, 0, 8, 9, 7, 4, 2, 1, 5
0	0	3	3	3, 0, 8, 9, 7, 4, 2, 1, 5

Index elemen saat ini ditunjukkan melalui kolom N . Sedangkan kolom k menunjukkan index dari elemen yang akan ditukar, nilai k tersebut diambil secara acak dari *range* yang diperbolehkan (N). Kolom $array[k]$ menunjukkan nilai yang terdapat pada *array* ke- k . Kolom t menunjukkan nilai elemen saat ini yang terdapat pada *array* ke- N .

3.2 Perancangan Aplikasi *Game*

Dalam pembuatan *game* diperlukan sebuah rancangan sebelum membangun *game* yang akan dibangun. Rancangan antarmuka digunakan untuk membuat tampilan dengan tujuan memberikan panduan dalam menggunakan *game* ini. Adapun perancangan antarmuka tersebut adalah sebagai berikut:

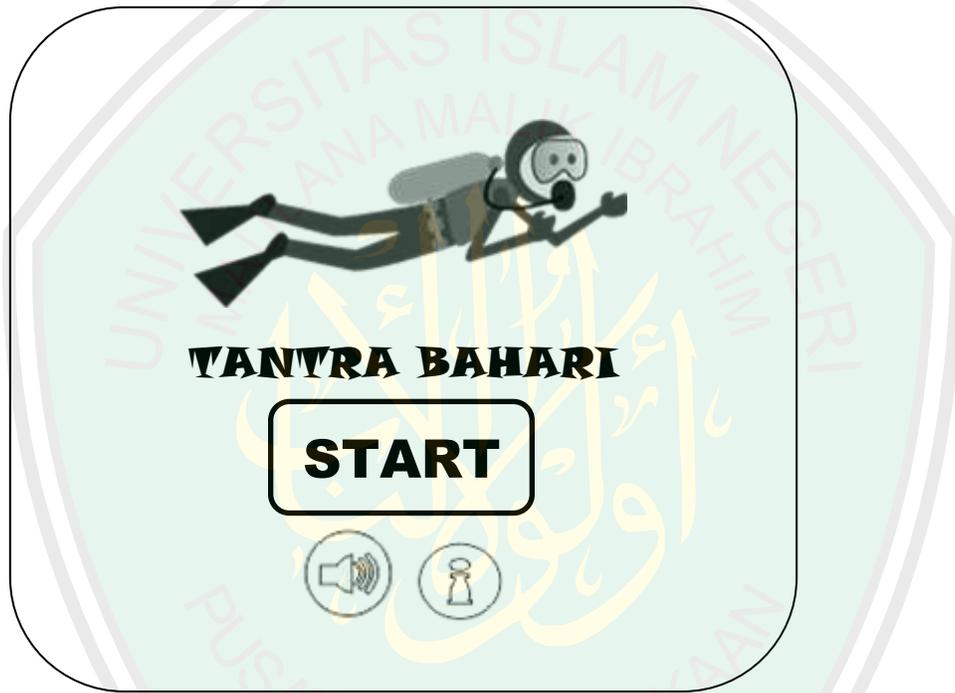
3.2.1 Perancangan Antarmuka *Splashscreen*

Pada saat *game* dijalankan maka akan muncul *splashscreen* yang berupa tampilan nama *game* dan gambar wajah karakter dari *game* tersebut yang terlihat pada **gambar 3.13**. Tampilan ini hanya sekilas setelah itu akan langsung pindah ke antarmuka menu *game* dengan tampilan yang sama, namun ditambah dengan *button-button* menu.



Gambar 3.13 Antarmuka Splashscreen

3.2.2 Perancangan Antarmuka Menu Game



Gambar 3.14 Antarmuka Menu Game

Terdapat beberapa tombol pada menu *game* seperti pada gambar 3.14. Misalkan ketika tombol bergambar *speaker* ditekan maka akan berbunyi suara efek. Keterangan fungsi tombol yang ada pada tabel 3.2 antara lain:

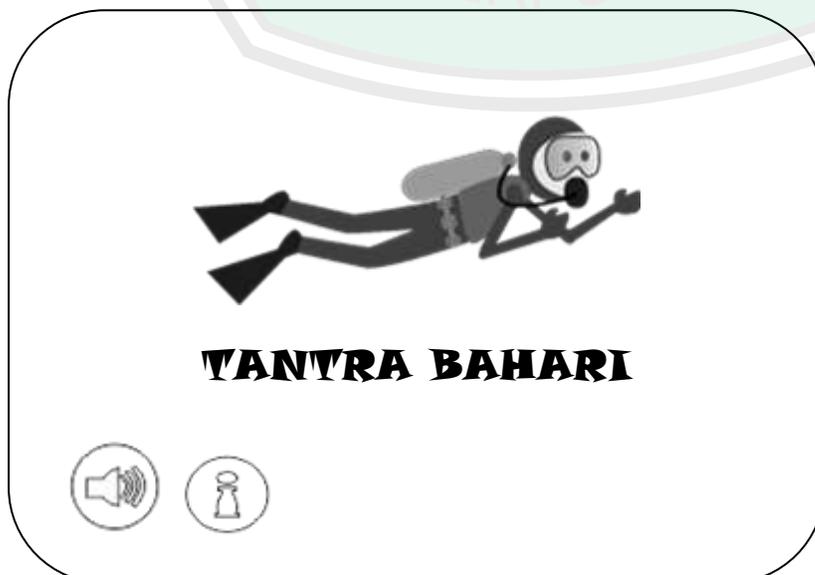
Tabel 3.2 Keterangan Fungsi Tombol Menu

No.	Menu	Keterangan
1.		

		Untuk memulai permainan
2.		Untuk mengaktifkan suara <i>game</i>
3.		Untuk menonaktifkan suara <i>game</i>
4.		Untuk menampilkan keterangan <i>game</i>
5.		Untuk keluar dari permainan

3.2.3 Perancangan Antarmuka Help *Game*

Perancangan antarmuka help *game* dapat dilihat ada **gambar 3.15**. Dimana pada halaman ini dijelaskan bagaimana tata cara memainkan *game* Tantra Bahari, dan keterangan mengenai misi *game* yang harus diikuti oleh pemain.



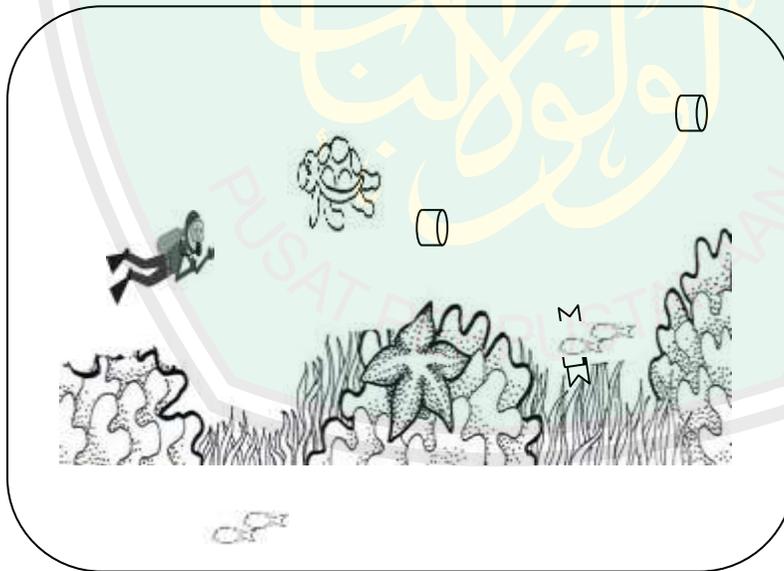
Gambar 3.15 Antarmuka Help

3.2.4 Perancangan Antarmuka *Game*

Perancangan antarmuka *game* ini menggambarkan keadaan layar ketika pemain sudah pada posisi memulai permainan.

Antarmuka *game* pada *game* Tantra Bahari ini mengambil latar bawah laut, sedangkan player diilustrasikan sebagai penyelam bernama Tantra yang bertugas mengumpulkan sampah di bawah laut.

Terdapat beberapa tombol yang ada pada antarmuka ini yang terlihat pada gambar 3.16.



Gambar 3.16 Antarmuka *Game*

3.2.5 Perancangan Alur Permainan

Perancangan ini menjelaskan alur permainan. Dimana pemain harus mengumpulkan sampah sebanyak mungkin, karena ketika pemain mendapatkan sampah. Pemain akan mendapatkan aksi special untuk dapat menjalankan permainan, agar mencapai *score* dan jarak tertinggi.

- Sampah botol, pemain akan mendapatkan tambahan kecepatan
- Sampah plastic, pemain akan mendapatkan perlindungan gelembung agar life pemain tidak berkurang.
- Sampah kaleng, pemain akan mendapat 1 life tambahan.
- Sampah jaring, pemain akan mendapat kecepatan maximum

Agar dapat mencapai *score*/jarak tertinggi maka pemain harus menghindari obstacle yang dapat mengurangi nyawa pemain, apabila nyawa pemain habis maka *game* akan berakhir.

3.2.6 Cara Memainkan *Game*

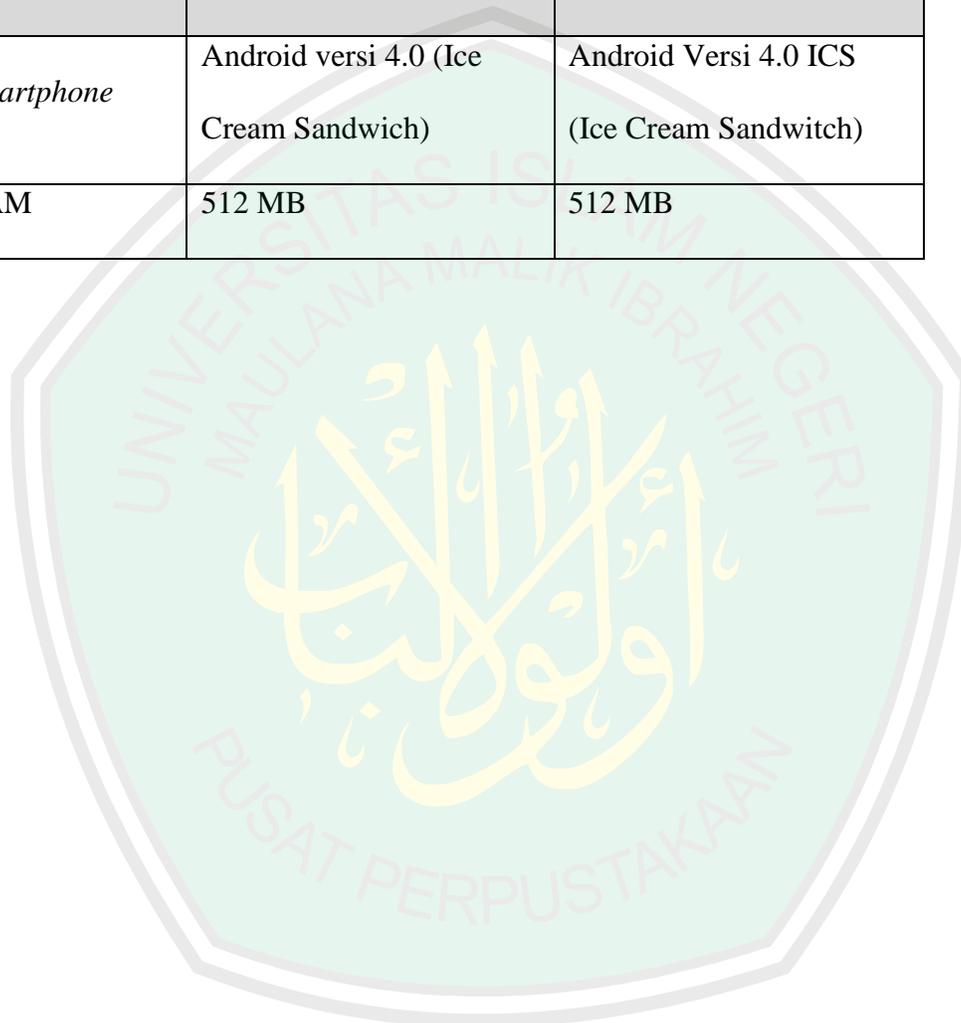
Cara memainkan *game* ini cukup mudah. Pemain dapat menggerak gerakkan gadget/*smartphone* ke arah yang diinginkan, karena permainan ini menggunakan sensor *accelerometer*.

3.2.7 Kebutuhan *Smartphone* Pemain

Berikut ini merupakan tabel daftar spesifikasi *smartphone* yang harus dimiliki pemain untuk memainkan *game* ini.

Tabel 3.3 Daftar Spesifikasi *Smartphone*

Kebutuhan	Spesifikasi Minimum	Spesifikasi Rekomendasi
<i>Smartphone</i>	Android versi 4.0 (Ice Cream Sandwich)	Android Versi 4.0 ICS (Ice Cream Sandwich)
RAM	512 MB	512 MB



BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Pada bab ini membahas tentang implementasi dan perancangan terhadap aplikasi yang sudah dibuat. Serta melakukan pengujian terhadap aplikasi yang dibuat untuk mengetahui apakah aplikasi tersebut telah berjalan sesuai dengan yang diharapkan. Sebelum diimplementasikan, terlebih dahulu dipaparkan spesifikasi sistem perangkat keras (*hardware*) dan perangkat lunak (*software*). Untuk pembuatan aplikasi ini diperlukan perangkat keras (*hardware*) dan lunak (*software*), adapun keperluan tersebut adalah:

1. Perangkat Keras (*Hardware*)

Perangkat keras pendukung digunakan adalah:

a. Laptop Axioo Neon HNM

Spesifikasi komputer pendukung yang digunakan:

- Processor Intel® Celeron CPU B820 @1.70 GHz
- RAM 2 GB DDR3
- Intel® HD Graphics
- 14.0" HD LED LCD
- Hardisk 500 GB
- Sistem Operasi Windows 7 Ultimate

b. *Smartphone* Axioo Picopad 10

Spesifikasi *smartphone* pendukung yang digunakan:

- Processor boxchip A10 cortex A8-1.2 GHz Multi Core 3D. Mali -400 GPU

- RAM 1 GB
- Sistem Operasi Android ICS 4.0.3 (Ice Cream Sandwich)

2. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan untuk pembuatan aplikasi ini adalah:

- a. Software Eclipse. Merupakan software yang dibutuhkan untuk melakukan coding aplikasi Android. Eclipse yang digunakan adalah versi 4.2.1 (Eclipse Juno) yang support dengan Android Development Tools (ADT).
- b. Android Development Tool (ADT), untuk membangun aplikasi.
- c. Android SDK (Software Development Kit), sebagai alat bantu dan API dalam mengembangkan aplikasi Android menggunakan bahasa java. Android SDK yang digunakan adalah versi 20.0.5.
- d. Corel Draw X6, untuk mendesain gambar dan tampilan.
- e. Adobe Photoshop CS6, untuk mendesain gambar dan tampilan.
- f. Microsoft Office Visio 2007, untuk mendesain flowchart dan finite state machine (FSM).
- g. Paint, untuk editing dan resize gambar

4.2 Implementasi Algoritma *Fisher Yates Shuffle*

Algoritma *Fisher Yates Shuffle* diimplementasikan menjadi sebuah fungsi *Shuffle*/kocok dalam sebuah *class* java yang diberi nama “**FisherYates**”. Fungsi *Shuffle* tersebut diberi nama “*Shuffle*”.

```
public void Shuffle(int[] array_position){
```

```

Random r = new Random();

for (int i = array_position.length - 1; i > 0; i--){
    int index = r.nextInt(i + 1);
    int a = array_position[index];
    array_position[index] = array_position[i];
    array_position[i] = a;
}
}

```

Pada bagian ini akan dibahas mengenai implementasi algoritma *Fisher Yates Shuffle* untuk menentukan posisi NPC. Pada saat permainan dimulai *Fisher Yates Shuffle* akan bekerja untuk memunculkan NPC, *class* java yang diberi nama “**FisherYates**” dipanggil melalui fungsi yang dinamai *Shuffle*. Berikut ini adalah *source code* untuk menentukan posisi NPC.

Source code penentuan posisi NPC:

```

int [] array_NPC = Shuffle.nextIntArrayKey(9);
    Shuffle.Shuffle(array_position);

    this.point = 0;
    this.skillPoint = 0;

BitmapTextureAtlasTextureRegionFactory.setAssetBasePath("sprite/"
); assets/sprites

```

```
Npc npc1 = new Npc(360, 60, 4, 1, 200, "sampah1.png", 480, 50,
    "sprite/", array_NPC[0]);
Npc npc2= new Npc(446, 72, 4, 1, 200, "sampah2.png", 480,
    150, "sprite/", array_NPC[1]);
Npc npc3 = new Npc(450, 58, 4, 1, 200, "sampah3.png", 480,
    250, "sprite/", array_NPC[2]);
Npc npc4 = new Npc(586, 64, 4, 1, 200, "enemyFish4.png", 480,
    350, "sprite/", array_NPC[3]);
Npc npc5 = new Npc(495, 64, 4, 1, 200, "enemyFish5.png", 480,
    350, "sprite/", array_NPC[4]);
Npc npc6 = new Npc(460, 88, 4, 1, 200, "enemyFish6.png", 480,
    350, "sprite/", array_NPC[5]);
Npc npc7 = new Npc(549, 103, 4, 1, 200, "enemyFish7.png", 480,
    350, "sprite/", array_NPC[6]);
Npc npc8 = new Npc(115, 74, 2, 1, 200, "terumbu.png", 480, 350,
    "sprite/", array_NPC[7]);
Npc npc9 = new Npc(30, 36, 1, 2, 200, "shell.png", 480, 350,
    "sprite/", array_NPC[8]);

npcList.add(npc1);
npcList.add(npc2);
    npcList.add(npc4);
    npcList.add(npc7);
    npcList.add(npc8);
    npcList.add(npc3);
    npcList.add(npc5);
    npcList.add(npc6);
    npcList.add(npc9);
```

4.3 Implementasi Aplikasi *Game*

Pada pembahasan ini merupakan hasil implementasi aplikasi *game* mencari mufradat.

Berikut ini adalah beberapa hasil implementasi aplikasi *game* tersebut.

4.3.1 Antarmuka *Splashscreen*

Pada saat *game* dijalankan akan menampilkan tampilan seperti pada gambar 4.1.

Splashscreen menampilkan berupa tampilan wajah karakter dan nama aplikasi.



Gambar 4.1 Tampilan *Splashscreen*

4.3.2 Antarmuka Menu *Game*

Menu dalam *game* ini terdapat lima menu yang memiliki fungsi masing-masing, antara lain: mulia, suara aktif, suara nonaktif, informasi dan keluar. Pada tabel 4.1 merupakan keterangan dari fungsi tombol yang ada pada tampilan menu.

Tabel 4.1 Keterangan Fungsi Tombol Menu Permainan

No.	Menu	Keterangan
1.		Untuk memulai permainan
2.		Untuk mengaktifkan suara efek dan musik
3.		Untuk menonaktifkan suara music
4.		Untuk menampilkan informasi tentang <i>game</i>
5.		Untuk keluar dari permainan

Pada halaman menu utama *game* terdapat beberapa tombol yaitu tombol start untuk memulai permainan, tombol bergambar speaker untuk menghidupkan/mematikan efek suara dan tombol help *game* untuk mengetahui keterangan dan tata cara melakukan permainan Tantra Bahari. Halaman utama *game* dapat dilihat pada **gambar 4.2**.



Gambar 4.2 Tampilan Menu Utama

Setelah memasuki menu utama, pemain dapat menekan tombol help untuk mengetahui keterangan *game*. Gambar 4.3 merupakan halaman help *game* yang berisi bagaimana cara memainkan *game* Tantra Bahari dan apa yang harus dilakukan pemain untuk mendapatkan skor tertinggi dalam permainan.

How To Play Tantra Bahari?

Move The phone up and down to make Tantra rise and fall



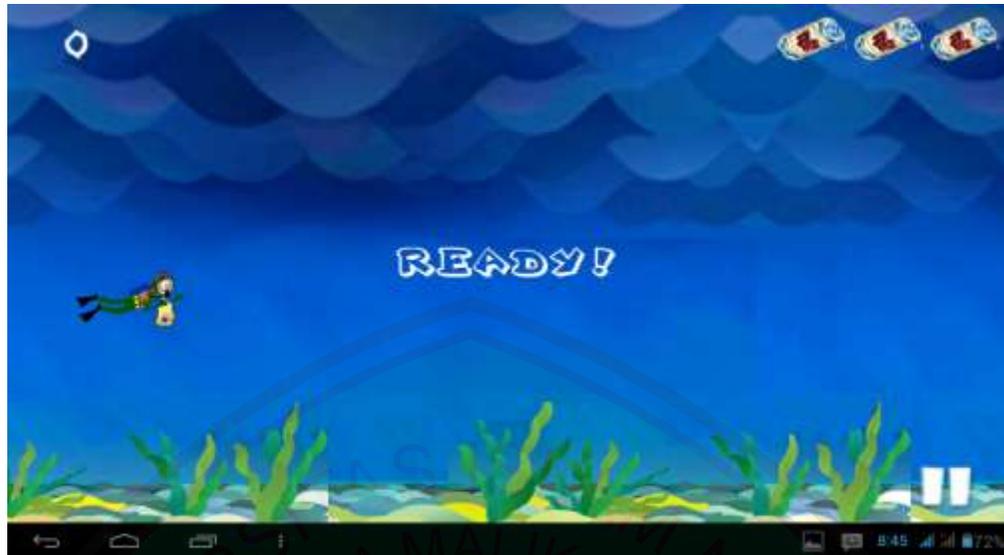
Take the trash, and avoid any sea creatures (obstacle)

Stay alive and get the high score

Gambar 4.3 Tampilan Menu Help

4.3.3 Scene Permainan

Setelah menekan tombol *start* pada tampilan menu utama, maka pemain akan memasuki halaman seperti pada gambar 4.4 dimana life awal dari player adalah 3 bar. Yang ditandai dengan ikon sampah kaleng pada pojok kanan atas layar permainan. Agar life tidak berkurang maka pemain harus menghindari *obstacle*.



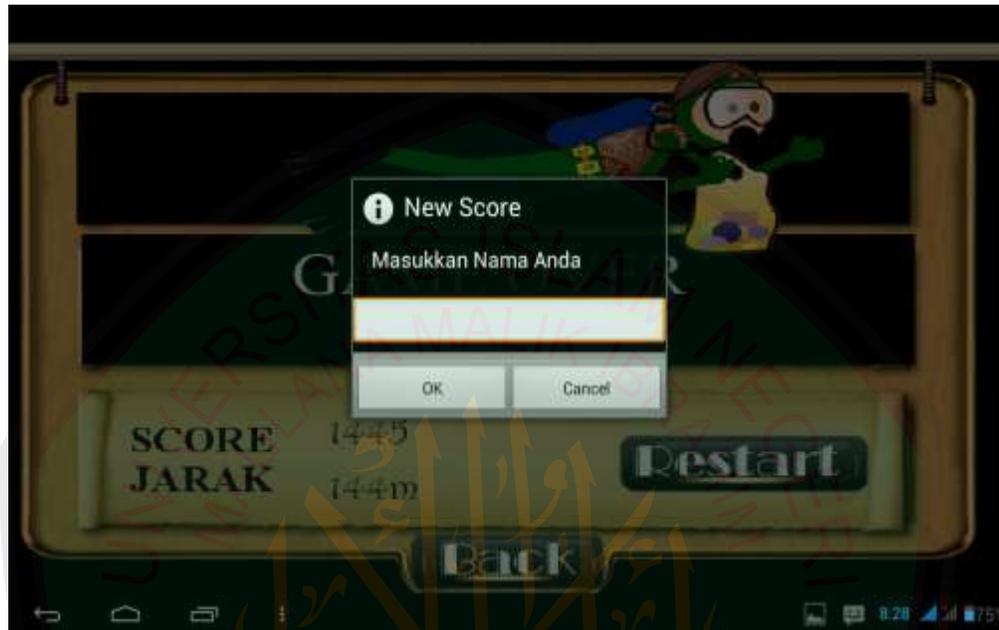
Gambar 4.4 Awal Permainan



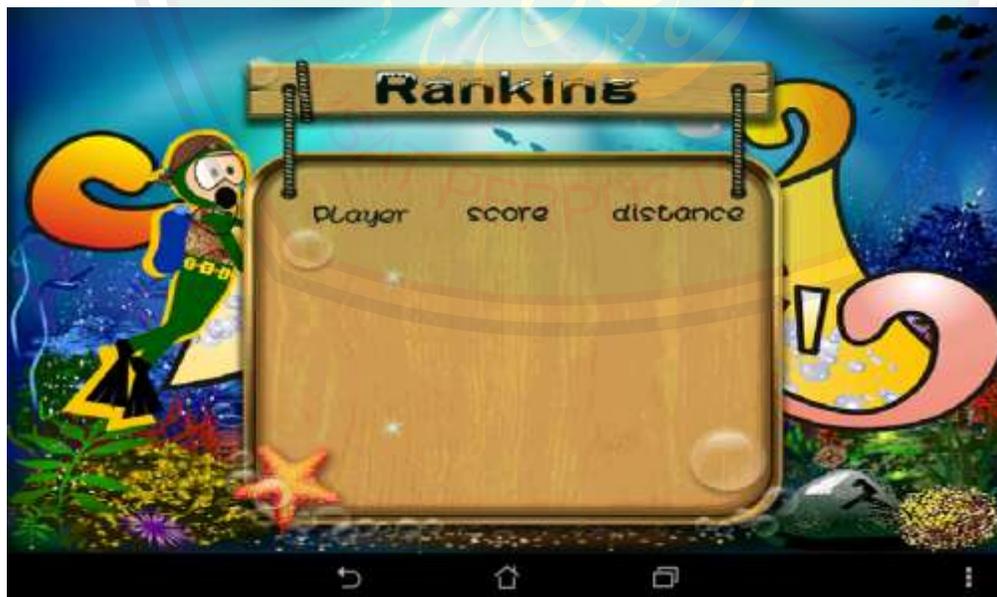
Gambar 4.5 Tampilan Scene pada jarak 1162

Player harus menghindari *obstacle* berupa ikan dan terumbu karang. Serta harus mengambil sampah plastik seperti pada **gambar 4.5** di atas.

Apabila tanda *life* (nyawa) yang berada di pojok kanan atas permainan habis, maka permainan berakhir dan pemain dapat memasukkan *new score* pada tampilan input *new score* seperti pada **gambar 4.6**.



Gambar 4.6 Tampilan *input new score*



Gambar 4.7 Halaman *score game*

4.4 Uji Coba

Pada subbab ini membahas tentang uji coba yang telah dilakukan. Ada dua uji coba yang telah dilakukan yaitu uji coba algoritma *Fisher Yates Shuffle* dan uji coba aplikasi. Berikut pembahasan uji coba tersebut.

4.4.1 Uji Coba Algoritma *Fisher Yates Shuffle*

Uji coba algoritma *Fisher Yates Shuffle* ini dilakukan untuk menentukan posisi NPC. Game ini mempunyai 8 array NPC *obstacle* yang dapat mengaplikasikan 8! (delapan faktorial) dari 8 array NPC tersebut. Oleh karena itu hampir tidak mungkin 2 permutasi memiliki kesamaan.

Tabel 4.3 Hasil Uji *Shuffle* Array

		PERMUTATIONS									
		1	2	3	4	5	6	7	8	9	10
NPC POSITIONS	1	4	8	5	6	5	7	2	5	6	2
	2	2	5	3	5	1	6	7	3	7	8
	3	1	7	2	4	6	3	1	7	3	4
	4	8	3	8	7	7	4	5	4	4	5
	5	3	6	7	3	3	2	8	1	1	3
	6	7	2	6	8	4	1	4	6	5	6
	7	5	4	4	3	8	5	6	8	2	7
	8	8	1	1	1	2	8	3	2	6	1

Tabel 4.3 di atas menunjukkan permutasi 1-10 dari 8 array yang di *Shuffle* dengan algoritma *Fisher Yates Shuffle*, dijalankan menggunakan java compiler.

Pada tabel 4.3 dapat diketahui bahwa algoritma *Fisher Yates Shuffle* dapat mengacak posisi NPC dengan baik.

Hasil pengujian pada *game* dapat terlihat pada **gambar 4.8** dan **gambar 4.9** dibawah ini:



Gambar 4.8 Hasil uji coba pertama pada *distance* 515 (1)



Gambar 4.9 Hasil uji coba kedua pada *distance* 515 (2)

Gambar 4.8 dan **gambar 4.9** merupakan hasil screenshot yang di ambil pada dua kali permainan di jarak yang sama, yakni jarak 515. Terlihat bahwa NPC tidak berulang dan dapat terkocok dengan baik.

4.4.2 Uji Coba Aplikasi

Uji coba ini dilakukan untuk mengetahui apakah aplikasi yang telah dibuat dapat diimplementasikan terhadap *device* atau *smartphone* yang lain. Berikut hasil pengujian dari aplikasi pada beberapa *smartphone* dalam bentuk **tabel 4.6**.

Tabel 4.4 Hasil Pengujian Aplikasi

No.	Vendor	Versi OS	Layar	CPU	RAM	Keterangan
1.	Samsung GT- S5360	v2.3.5 (Ginger bread)	3"	832 MHz Processor	290 MB	<i>Game</i> tidak dapat di jalankan.
2.	Axioo picopad 10 GJT	v 4.0.4 (Ice Cream Sandwi ch)	10"	Processor boxchip A10 cortex A8-1.2 GHz Multi Core 3D. Mali - 400 GPU	1 GB	Seluruh tampilan berjalan dengan baik. Efek musik dan sensor berfungsi dengan baik. <i>Game</i> berjalan dengan baik.
3.	Samsung GT- s5282	v4.1.2 (Ice Cream Sandwi ch)	3.0"	1 GHz Cortex-A5 Sensors Accelerome ter	512 MB	Seluruh tampilan berjalan dengan baik. Efek musik

						dan sensor berfungsi dengan baik. <i>Game</i> berjalan dengan baik.
4.	Samsung Galaxy Note 3 N9000	v4.3 (Jelly Bean)	5.7"	Exynos 5 Octa 5420 Chipset Octa Core (Quad-core 1.9 GHz Cortex-A15 + Quad-core 1.3 GHz Cortex-A7) Mali-T628 MP6 GPU	1 GB	Seluruh tampilan berjalan dengan baik. Efek musik dan sensor berfungsi dengan baik. <i>Game</i> berjalan dengan baik.
5.	ASUS K012	v4.3 (Jelly Bean)	7"	Intel® Atom™ Multicore Processor	1 GB	Sensor accelerometer tidak berjalan

				Z2520 (1.2 GHz)		dengan baik
6.	Oppo r1001	v4.2.2 (Jelly Bean)	4.0"	dual core 1.3GHz, GPU Mali-400	512M B	Seluruh tampilan berjalan dengan baik. Efek musik dan sensor berfungsi dengan baik. <i>Game</i> berjalan dengan baik.

Tabel 4.5 Tabel pengujian *game* tantra bahari pada berbagai versi OS android

No	Versi Android	Sudah diuji	Belum diuji	Keterangan
1	Android 1.0 (API level 1)		√	Belum dilakukan uji coba
2	Android 1.1 (API level 2)		√	Belum dilakukan uji coba
3	Android 1.5 Cupcake (API level 3)		√	Belum dilakukan uji coba
4	Android 1.6 Donut (API level 4)		√	Belum dilakukan uji coba
5	Android 2.0 Eclair (API level 5)		√	Belum dilakukan uji coba
6	Android 2.0.1 Eclair (API level 6)		√	Belum dilakukan uji coba
7	Android 2.1 Eclair (API level 7)		√	Belum dilakukan uji coba
8	Android 2.2–2.2.3 Froyo (API level 8)		√	Belum dilakukan uji coba
9	Android 2.3–2.3.2 Gingerbread (API level 9)		√	Belum dilakukan uji coba

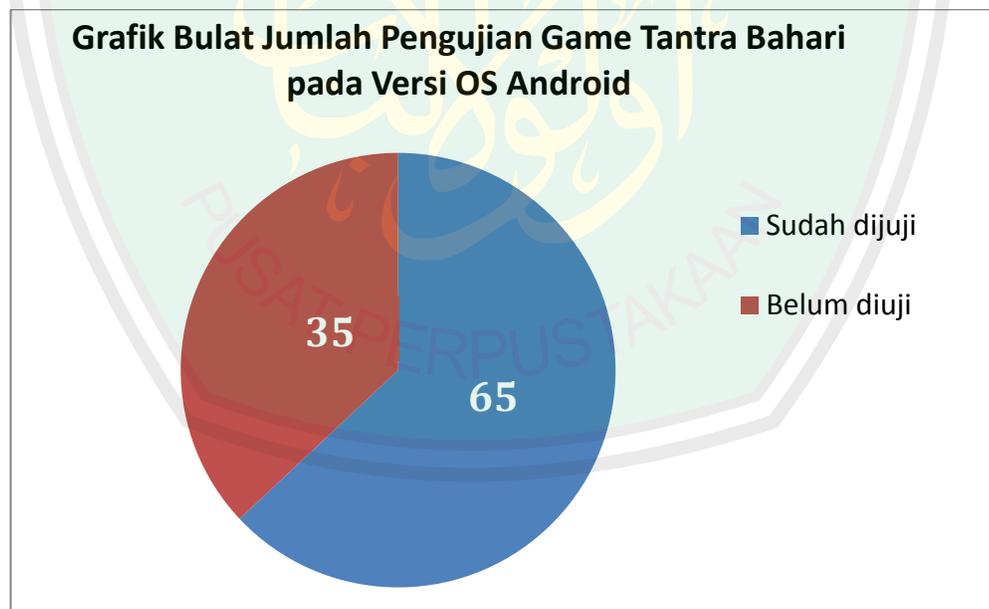
10	Android 2.3.3–2.3.7 Gingerbread (API level 10)	√		Game tidak berjalan
11	Android 3.0 Honeycomb (API level 11)		√	Belum dilakukan uji coba
12	Android 3.1 Honeycomb (API level 12)		√	Belum dilakukan uji coba
13	Android 3.2 Honeycomb (API level 13)		√	Belum dilakukan uji coba
14	Android 4.0–4.0.2 Ice Cream Sandwich (API level 14)	√		Game berjalan dengan baik
15	Android 4.0.3–4.0.4 Ice Cream Sandwich (API level 15)	√		Game berjalan dengan baik
16	Android 4.1 Jelly Bean (API level 16)	√		Game berjalan dengan baik
17	Android 4.2 Jelly Bean (API level 17)	√		Game berjalan dengan baik
18	Android 4.3 Jelly Bean (API level 18)	√		Game berjalan dengan baik
19	Android 4.4 KitKat (API level 19)	√		Game berjalan dengan baik

20	Android 5.0 Lollipop		√	Belum dilakukan uji coba
----	----------------------	--	---	--------------------------

Spesifikasi minimum untuk *game* tantra bahari adalah android OS versi 4.0 (Ice Cream Sandwich), sehingga ketika dilakukan uji coba pada android versi 2.3.5 API level 10 (Gingerbread) maka *game* tidak dapat dijalankan.

Berdasarkan **tabel 4.5**, dari 20 versi android, *game* tantra bahari telah di uji coba pada 7 macam versi android dan belum di uji coba pada 13 versi OS android.

Grafik uji coba *game* tantra bahari yang dilakukan pada berbagai versi android dapat dilihat pada **gambar 4.10** berikut:



Gambar 4.10 Grafik Bulat Jumlah Pengujian Game Tantra Bahari pada Versi OS Android

4.5 Integrasi Game Tantra Bahari dengan Islam

Dalam agama Islam, menyampaikan hal positif dan melarang pada hal-hal negatif dalam berbagai bentuk, metode, dan media merupakan tugas manusia sebagai makhluk Allah yang beriman. Dalam surat Ali Imron ayat 104 Allah berfirman:

وَلْتَكُنْ مِنْكُمْ أُمَّةٌ يَدْعُونَ إِلَى الْخَيْرِ وَيَأْمُرُونَ بِالْمَعْرُوفِ وَيَنْهَوْنَ عَنِ الْمُنْكَرِ وَأُولَئِكَ هُمُ الْمُفْلِحُونَ



Artinya:

“Dan hendaklah ada di antara kamu segolongan umat yang menyeru kepada kebajikan, menyuruh kepada yang ma'ruf dan mencegah dari yang munkar. Merekalah orang-orang yang beruntung.”

Dalam tafsir Ibnu katsir, Abu Ja'far Al Baqir berkata, Rasulullah SAW pernah membaca ayat *وَلْتَكُنْ مِنْكُمْ أُمَّةٌ يَدْعُونَ إِلَى الْخَيْرِ* “*dan hendaklah ada di antara kamu golongan umat yang menyeru kepada kebajikan.*” Lalu beliau bersabda: *“(الخير اتباع القرآن وسنتي) “kebaikan itu adalah mengikuti sunnah Al Qur'an dan sunnahku”*.

Dalam hal ini dalam tafsiran lain kata “al khoiru” adalah kebaikan manusia pada agama dan dunia “*ما فيه صلاح الدنيا و الدين*”. Dengan demikian kata yad'uuna ilal khoiri dapat diambil kesimpulan Allah memerintahkan umat-Nya agar menyeru atau mengajak kepada kebaikan dan berbuat baik, yaitu dengan cara mengikuti Al Qur'an dan sunnah nabi SAW.

Salah satu seruan dan ajakan dalam kebaikan adalah ajakan untuk menjaga kebersihan, dalam sebuah hadist disampaikan bahwa *kebersihan merupakan sebagian dari iman*”, hal ini menjelaskan bahwa seseorang yang beriman pasti mencintai kebersihan.

Sebab Al quran yang merupakan sumber tuntunan manusia dalam menjalani kehidupan juga menuntun manusia dalam menjaga kebersihan lingkungan dan tidak merusaknya, seperti yang terkandung dalam QS. Ar Rum ayat 41 – 42:

ظَهَرَ الْفَسَادُ فِي الْبَرِّ وَالْبَحْرِ بِمَا كَسَبَتْ أَيْدِي النَّاسِ لِيُذِيقَهُمْ بَعْضَ الَّذِي عَمِلُوا لَعَلَّهُمْ يَرْجِعُونَ ﴿٤١﴾

Artinya:

“Telah nampak kerusakan di darat dan di laut disebabkan Karena perbuatan tangan manusi, supaya Allah merasakan kepada mereka sebahagian dari (akibat) perbuatan mereka, agar mereka kembali (ke jalan yang benar).”

Ada banyak cara untuk mengajak berbuat baik, ajakan untuk beribadah, menolong sesama, ajakan untuk menuntut ilmu dari berbagai aspek, dan ada juga ajakan untuk menjaga kebersihan, misalnya ajakan untuk menjaga kebersihan laut.

Ajakan menjaga kebersihan tersebut dapat dilakukan dengan berbagai cara salah satunya adalah melalui media permainan. *Game* Tantra Bahari dibangun untuk membangun pesan moral pentingnya menjaga kebersihan laut atau bahari Indonesia.

BAB V

PENUTUP

5.1 Kesimpulan

Game Tantra Bahari ini merupakan *game* yang cara bermainnya menggunakan *sensor accelerometer*, didalamnya terdapat pesan moral yakni ajakan untuk membersihkan bawah laut.

Algoritma *Fisher Yates Shuffle* diimplementasikan untuk mengacak array berisi gambar NPC. Dimana pada permainan ini array berisi gambar diacak untuk mendapatkan posisi NPC yang tidak berulang sesuai panjang array NPC (N). NPC sampah dengan panjang array 4 (N=4) atau (array[0..3]) dan NPC *obstacle* dengan panjang array 8 (N=8) atau [0..7]). Posisi NPC diinisialisasikan dengan nilai koordinat float x dan float y dari deklarasi gambar NPC pada layar 720x480.

Dari hasil implementasi dan uji coba, algoritma *Fisher Yates Shuffle* dapat digunakan untuk mengacak posisi NPC pada *game* Tantra Bahari agar posisinya tidak berulang pada permainan berikutnya. *Game* Tantra Bahari ini dipasang pada jenis *smartphone* android dengan versi android yang berbeda sehingga diketahui pada android OS versi 2.3.5 *game* tidak berhasil diinstall, dan pada versi OS 4.3 dari *device* ASUS K012, *game* dapat diinstall namun *sensor accelerometer* tidak berfungsi sehingga permainan tidak bisa berjalan dengan baik, selebihnya pada *device smartphone* dengan android OS versi 4.0.4 *device* Axioo pico 10, versi 4.1.2 dari Samsung GT s5282, versi 4.2.2 dari Oppo R1001, dan dari android OS versi 4.3 Samsung galaxy note 3 *game* dapat berjalan dengan baik.

5.2 Saran

Dalam pembuatan *game* ini tentu masih banyak kekurangan yang masih perlu dilakukan perbaikan dan pengembangan untuk menjadikan aplikasi ini semakin bagus dan diminati banyak orang. Oleh karena itu penulis menyarankan beberapa hal untuk bahan pengembangan selanjutnya, diantaranya:

1. Mengembangkan *game* ini agar memiliki tampilan yang lebih menarik lagi.
2. Adanya pengembangan *multiplatform* agar dapat dimainkan diberbagai sistem operasi *mobile* tidak hanya di android.



DAFTAR PUSTAKA

- Antony Susanto dan Henky Honggo. 2013. *“Perancangan Ujian Online pada STMIK GI MDP Berbasis Web”*. STMIK GI MDP.
- Beck, John C, dan Mitchell Wade. 2006. *Gamers Juga Bisa Sukses*. Diterjemahkan oleh: Isman H. Suryaman. Jakarta: PT. Grasindo
- Black, Paul E. 2005. *“Fisher–Yates Shuffle” Dictionary of Algorithms and Data Structures*. National Institute of Standards and Technology.
- Coding Horror. *The Danger of Naivete*. [Online]. Diakses 11 Januari 2014.
<http://blog.codinghorror.com/the-danger-of-naivete/>
- Departemen Agama RI. 2005. *Al-Jumanatul ‘Ali: Al Quran dan Terjemahannya*. Bandung: CV Penerbit J-Art.
- Didik Setiawan. 2003. *Mengenal SQLite Command Line*. ilmu komputer.
- Durstenfeld, Richard. 1964. "Algorithm 235: Random permutation". Communications of the ACM 7 (7): 420. doi:10.1145/364520.364540.
- HanKim, Chong dkk. 2006. *Verification of FSM using Attributes Definition of NPCs Models*. JCSNS International Journal of Computer Science and Network Society.
- Haryanto, Hanny. 2010. *Pendidikan Moral dengan Menggunakan Sistem Reward Dalam Game Imersif*. Semarang: Majalah Ilmiah DIAN.
- Henry, Samuel. 2010. *Cerdas dengan Game: Panduan Praktis Bagi Orangtua dalam Mendampingi Anak Bermain Game*. Jakarta: PT. Gramedia Pustaka Utama.
- Ibijola, Ade dan Abejide Olu. 2012, *A Simulated Enhancement of Fisher-Yates Algorithm for Shuffling in Virtual Card Games using Domain-Specific Data Structures*. International Journal of Computer Applications, 54:24-28 Diakses tanggal 2 Januari 2014 dari http://www.academia.edu/2985786/A_Simulated_Enhancement_of_FisherYates_Algorithm_for_Shuffling_in_Virtual_Card_Games_using_DomainSpecific_Data_Structures
- Jeong, Eui Jun, dan J. Kim. 2009. *Definitions, Key Characteristics, and Generations of Mobile Games*. USA: IGI Global.
- Katsir, Ibnu. 2003. *Tafsir Ibnu Katsir Jilid 2*. Diterjemahkan oleh: M. Abdul Ghofal E.M. Bogor: Pustaka Imam Asy-syafi'i.

- Kusumawati, Yenni. 2004. *Metode Pengacakan Fisher Yates-Shuffle untuk Game Puzzle Berbasis J2me*. Universitas Kristen Duta Wacana.
- Lakoro, Rahmatsyam. 2009. *Mempertimbangkan Peran Permainan Edukasi dalam Pendidikan di Indonesia*. Institut Teknologi Surabaya.
- Mulyadi. 2010. *Membuat Aplikasi untuk Android I*. Yogyakarta: Multimedia Center Publishing.
- Pavel, Micka. *Fisher-Yates-Shuffle Algorithm* : Founder and administrator of web encyclopedia Algoritmy.net.[Online]. Diakses 10 Januari 2014.
<http://en.algoritmy.net/article/43676/Fisher-Yates-Shuffle>
- Pramudya. Puja. 2007. *Game Tebak Kartu dengan Windows Multipoint SDK*. Ilmu Komputer.
- Reynolds, Craig W. *Steering Behaviors For Autonomous Characters*. Sony Computer Entertainment America.
- Rizqyawan, Muhammad Ilham. 2013. *Game Edukasi Adventure untuk Pengenalan Teori Musik Berbasis Dekstop*. Universitas Komputer Indonesia.
- Rogers, Rick. 2012. *Learning Android Game Programming*. United States: Pearson Education, Inc.
- Safaat, Nazruddin. 2011. *Android : Pemrograman Aplikasi Mobile Smartphone dan Tablet PC*. Bandung: Informatika.
- Seifert, K. dan Camacho O. 2007. *Implementing Positioning Algorithms Using Accelerometers*. Freescale Semiconductor.
- Wikipedia Indonesia. *Daftar Versi Android*. [Online]. Diakses 11 Januari 2014.
http://id.wikipedia.org/wiki/Daftar_versi_Android
- Yayasan Terumbu Karang Indonesia. [Online]. Diakses 11 Januari 2014.
<http://www.terangi.or.id/>

LAMPIRAN-LAMPIRAN





1. Tampilan menu pada layar 10"



2. Tampilan menu pada layar 5.7"



3. Tampilan *game* yang tidak berhasil pada layar 3"



4. Tampilan *game* pada layar 3"