

**PENERAPAN ALGORITMA *FISHER YATES SHUFFLE*
DAN ALGORITMA *FORWARD CHAINING*
PADA GAME EDUKASI *MR GARBAGE***

SKRIPSI

Oleh:

**RISTI NURARIFAH
NIM. 13650033**



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2018**

LEMBAR PENGAJUAN

**PENERAPAN ALGORITMA *FISHER YATES SHUFFLE*
DAN ALGORITMA *FORWARD CHAINING*
PADA GAME EDUKASI *MR GARBAGE***

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:**

**RISTI NURARIFAH
NIM. 13650033**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2018**

LEMBAR PERSETUJUAN

PENERAPAN ALGORITMA *FISHER YATES SHUFFLE*
DAN ALGORITMA *FORWARD CHAINING*
PADA GAME EDUKASI MR GARBAGE

SKRIPSI

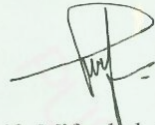
Oleh:

RISTI NURARIFAH
NIM. 13650033

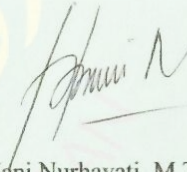
Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal, Mei 2018

Dosen Pembimbing I

Dosen Pembimbing II



Yunifa Miftachul Arif, M.T
NIP. 19830616 201101 1 004



Hani Nurhayati, M.T
NIP. 19780625 200801 2 006

Mengesahkan,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Yanyu Crysdian
19740424 200901 1 008

LEMBAR PENGESAHAN

**PENERAPAN ALGORITMA FISHER YATES SHUFFLE
DAN ALGORITMA FORWARD CHAINING
PADA GAME EDUKASI MR GARBAGE**

SKRIPSI

Oleh:

**RISTI NURARIFAH
NIM. 13650033**

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: Mei 2018

Susunan Dewan Penguji

Tanda Tangan

Penguji Utama : Fachrul Kurniawan, M.MT
NIP. 19771020 200901 1 001

Ketua Penguji : Fresy Nugroho, M.T
NIP. 19710722 201101 1 001

Sekretaris Penguji : Yunifa Miftachul Arif, M.T
NIP. 19830616 201101 1 004

Anggota Penguji : Hani Nurhayati, M.T
NIP. 19780625 200801 2 006

(.....)

(.....)

(.....)

(.....)

Mengesahkan,

Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Maulana Malik Ibrahim Malang



Prasanto Crysdiyan
NIP. 19740424 200901 1 008

PERNYATAAN ORISINALITAS PENELITIAN

Saya yang bertanda tangan di bawah ini :

Nama : RISTI NURARIFAH

NIM : 13650033

Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Penelitian : **PENERAPAN ALGORITMA FISHER YATES
SHUFFLE DAN ALGORITMA FORWARD CHAINING PADA GAME
EDUKASI MR GARBAGE**

Menyatakan dengan sebenar-benarnya bahwa hasil penelitian saya ini tidak terdapat unsur-unsur penjiplakan karya penelitian atau karya ilmiah yang pernah dilakukan atau dibuat oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata hasil penelitian ini terbukti terdapat unsur-unsur jiplakan, maka saya bersedia untuk mempertanggung jawabkan, serta diproses sesuai peraturan yang berlaku.

Malang, Mei 2018

Penulis



Risti Nurarifah
NIM. 13650033

MOTTO

“Do the best and pray. God will take care of the rest”

Lakukan yang terbaik, kemudian berdoa. Tuhan yang akan mengurus sisanya.



HALAMAN PERSEMBAHAN

Alhamdulillah puji syukur kehadiran Allah SWT yang memberikan kekuatan kepada saya hingga bisa sampai menyelesaikan kuliah S1 di kampus hijau tercinta. Sholawat serta salam kepada Nabi Muhammad SAW, yang membawa petunjuk terbaik kepada seluruh umat manusia.

Terima kasih kepada kedua orang tua saya, Ayah saya, Bapak Bejo yang mendidik saya dari kecil hingga sekarang bisa menyelesaikan kuliah saya, Ibu saya tercinta, Ibu Tani yang tiap hari mendo'akan saya, mendukung saya dalam melangkah, menemani saya setiap saat, mendidik saya dari lahir hingga mampu menyelesaikan segala kewajiban saya dibangku pendidikan. Hormat dan rasa terimakasih yang tiada terhingga kupersembahkan karya kecil ini kepada bapak dan ibu yang telah memberikan kasih sayang, segala dukungan dan cinta kasih yang tiada terhingga yang tiada mungkin dapat dibalas hanya dengan selembur kertas yang bertuliskan kata cinta dan persembahan.

Terima kasih kepada dosen-dosen yang telah sabar dan ikhlas dalam mendidik saya hingga mampu melewati seluruh ujian dari semua mata kuliah yang saya tempuh, terutama kepada Bapak Yunifa Miftachul Arif, M.T dan Ibu Hani Nurhayati, M.T, semoga ilmu yang beliau amalkan berguna bagi seluruh mahasiswa dan semoga beliau diberikan kekuatan oleh Allah dalam berjihad didunia pendidikan hingga melahirkan anak didik yang mampu mengamalkan segala ilmu yang telah diberikan.

Kakak-kakakku tersayang, Neng anis dan Neng iwin. Tiada yang paling mengharukan saat kumpul bersama kalian, walaupun sering bertengkar tapi hal itu selalu menjadi warna yang tak akan bisa tergantikan. Terimakasih untuk segala macam dukungan yang diberikan.

Mas Anang dan Mas Arif, yang tak kenal lelah menanyakan “Kapan Lulus ris?”. Akhirnya setelah sekian lama aku lulus mas! Terima kasih atas dukungan moril dan materil.

Seluruh keluarga besar saya tercinta, Terima kasih selalu memberi dukungan yang tak terhingga serta doa yang senantiasa mengiringi setiap langkah saya.

Mas Arief Muhammad Angkasa, Terima kasih atas kesabarannya yang telah memberikan semangat dan inspirasi dalam menyelesaikan tugas akhir ini. Semoga semakin sukses karirnya.

Sahabatku, Ainul Widad dan Maya terima kasih atas doa, nasehat, hiburan, tumpangan rumah, transferan dan canda tawanya selama 8 tahun ini. Semoga hubungan ini tetap harmonis sampai kita tua.

Terimakasih untuk Larry Page dan Sergey Brin yang telah menciptakan google. Good job dan terimakasih telah mempermudah pengerjaan skripsi saya.

Para pejuang tandatangan, Pipit, Rahayu, Mbak Permata. See u on top gaes ☺

Untuk orang-orang diluar sana yang pernah mempertanyakan kapan skripsi ini selesai.

Last but not least, untuk teman-teman Teknik Informatika angkatan 2013 yang belum pada selesai, tetap semangat gaes tidak apa terlambat wisuda dari pada tidak pernah wisuda sama sekali.



KATA PENGANTAR

Assalamu'alaikum Wr.Wb.

Segala puji bagi Allah SWT tuhan semesta alam, karena atas segala rahmat dan karunia-Nya sehingga penulis mampu menyelesaikan skripsi dengan baik dan lancar. Shalawat serta salam selalu tercurah kepada tauladan terbaik Nabi Muhammad SAW yang telah membimbing umatnya dari zaman kebodohan menuju islam yang *rahmatan lil alamiin*.

Dalam menyelesaikan skripsi ini, banyak pihak yang telah memberikan bantuan baik secara moril, nasihat dan semangat maupun materil. Atas segala bantuan yang telah diberikan, penulis ingin menyampaikan doa dan ucapan terimakasih yang sedalam-dalamnya kepada:

1. Bapak Dr. Cahyo Crysdian, selaku ketua jurusan Teknik Informatika yang telah memberikan motivasi untuk terus berjuang.
2. Bapak Yunifa Miftachul Arif, M.T, selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing, mengarahkan dan memberi masukan kepada penulis dalam pengerjaan skripsi ini hingga akhir.
3. Ibu Hani Nurhayati, M.T, selaku dosen pembimbing II yang senantiasa memberi masukan dan nasihat serta petunjuk dalam penyusunan skripsi ini.
4. Segenap dosen Teknik Informatika yang telah memberikan bimbingan keilmuan kepada penulis selama masa studi.
5. Teman-teman seperjuangan Teknik Informatika angkatan 2013.

6. Semua pihak yang tidak mungkin penulis sebutkan satu-persatu, atas segala yang telah diberikan kepada penulis dan dapat menjadi pelajaran.

Berbagai kekurangan dan kesalahan mungkin pembaca temukan dalam penulisan skripsi ini, untuk itu penulis menerima segala kritik dan saran yang membangun dari pembaca sekalian. Semoga apa yang menjadi kekurangan bisa disempurnakan oleh peneliti selanjutnya dan semoga karya ini senantiasa dapat memberi manfaat. Amim. *Wassalamualaikum Wr. Wb*

Malang,

Penulis

DAFTAR ISI

| | |
|---|--------------|
| HALAMAN JUDUL | i |
| LEMBAR PENGAJUAN..... | ii |
| LEMBAR PERSETUJUAN | iii |
| LEMBAR PENGESAHAN | iv |
| PERNYATAAN ORISINALITAS PENELITIAN..... | v |
| MOTTO | vi |
| HALAMAN PERSEMBAHAN | vii |
| KATA PENGANTAR..... | x |
| DAFTAR ISI..... | xii |
| DAFTAR GAMBAR..... | xv |
| DAFTAR TABEL | xvii |
| ABSTRAK | xviii |
| ABSTRACT | xix |
| ملخص البحث | xx |
| BAB 1 PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Identifikasi Masalah | 6 |
| 1.3 Tujuan Penelitian..... | 6 |
| 1.4 Batasan Masalah..... | 7 |
| 1.5 Manfaat Penelitian..... | 7 |
| 1.6 Metode Penelitian..... | 7 |
| 1.7 Sistematika Penulisan..... | 9 |
| BAB II KAJIAN PUSTAKA | 11 |
| 2.1 Landasan Teori | 11 |
| 2.1.1 <i>Game</i> | 11 |
| 2.1.2 <i>Genre Game</i> | 12 |
| 2.1.3 <i>Game</i> Sebagai Edukasi..... | 18 |
| 2.2 Sampah | 19 |
| 2.2.1 Pengertian Sampah..... | 19 |
| 2.2.2 Kondisi Sampah di Indonesia..... | 21 |
| 2.2.3 Sumber Sampah | 23 |
| 2.2.4 Jenis Sampah..... | 25 |
| 2.2.5 Pengelolaan Persampahan..... | 26 |
| 2.3 Algoritma <i>Fisher Yates Shuffle</i> | 30 |
| 2.4 Algoritma <i>Forward Chaining</i> | 34 |
| 2.5 <i>Game Engine</i> | 38 |

| | | |
|--|---|-----------|
| 2.5.1 | Unity..... | 38 |
| 2.6 | Blender | 45 |
| 2.7 | Hasil Penelitian Yang Relevan..... | 46 |
| BAB III PERANCANGAN GAME | | 49 |
| 3.1 | Analisis dan Perancangan <i>Game</i> | 49 |
| 3.1.1 | Deskripsi <i>Game</i> | 49 |
| 3.1.2 | Keterangan Umum <i>Game</i> | 49 |
| 3.1.3 | <i>Storyboard</i> | 50 |
| 3.1.4 | Deskripsi Karakter | 54 |
| 3.1.5 | Deskripsi Item | 55 |
| 3.1.6 | Macam-macam Sampah | 56 |
| 3.2 | <i>Finite State Machine</i> | 57 |
| 3.3 | <i>Levelling</i> | 58 |
| 3.4 | <i>Scoring</i> | 59 |
| 3.5 | Pertanyaan | 59 |
| 3.6 | Metode Algoritma <i>Fisher Yates Shuffle</i> Pada Pengacakan Pertanyaan . | 59 |
| 3.7 | Metode <i>Forward Chaining</i> Untuk Menentukan Kenaikan <i>Level</i> | 63 |
| 3.8 | Proses Representasi Cara Kerja <i>Game</i> | 65 |
| BAB IV HASIL DAN PEMBAHASAN | | 69 |
| 4.1 | Implementasi | 69 |
| 4.1.1 | Perangkat Yang Digunakan..... | 69 |
| 4.2 | Implementasi Sistem | 70 |
| 4.2.1 | Implementasi Algoritma <i>Fisher Yates Shuffle</i> Pada Pengacakan Pertanyaan..... | 70 |
| 4.2.2 | Implementasi Algoritma <i>Forward Chaining</i> Pada Kenaikan <i>Level</i> | 72 |
| 4.2.3 | <i>Cek Rules</i> dalam <i>Knowledge Base</i> | 73 |
| 4.3 | Implementasi <i>Game</i> | 75 |
| 4.3.1 | Tampilan <i>Splashscrene</i> | 75 |
| 4.3.2 | Tampilan Menu <i>Game</i> | 75 |
| 4.3.3 | Tampilan <i>Scene Introduction</i> | 76 |
| 4.3.4 | Tampilan <i>Game</i> | 78 |
| 4.4 | Uji Coba | 87 |
| 4.4.1 | Uji Coba <i>Game</i> Menggunakan Algoritma <i>Fisher Yates Shuffle</i> | 87 |
| 4.4.2 | Uji Coba <i>Game</i> Menggunakan Algoritma <i>Forward Chaining</i> | 94 |
| 4.4.3 | Pengujian <i>Level</i> | 95 |
| 4.4.4 | Analisa Hasil Uji Coba Algoritma <i>Forward Chaining</i> Pada <i>Game</i> | 104 |

| | | |
|-------------------------------|-----------------------------|------------|
| 4.5 | Uji Coba <i>Game</i> | 106 |
| 4.6 | Integrasi Dalam Islam | 107 |
| BAB V KESIMPULAN | | 111 |
| 5.1 | Kesimpulan..... | 111 |
| 5.2 | Saran | 112 |
| DAFTAR PUSTAKA | | |



DAFTAR GAMBAR

| | |
|--|----|
| Gambar 1.1 Alur Metode Penelitian | 8 |
| Gambar 2.1 <i>Game</i> the elder scrolls v : skyrim..... | 13 |
| Gambar 2.2 <i>Game</i> Ninja Blade | 13 |
| Gambar 2.3 <i>Game</i> Teka Teki Silang..... | 14 |
| Gambar 2.4 <i>Game</i> Ragnarok..... | 15 |
| Gambar 2.5 <i>Game</i> Champhionship Manager..... | 16 |
| Gambar 2.6 <i>Game</i> Comand & Conquer 3 : Kane Wrath | 17 |
| Gambar 2.7 <i>Game</i> Tajwid Petualangan | 18 |
| Gambar 2. 8 Sampah Jakarta..... | 22 |
| Gambar 2. 9 Sampah Gunung Semeru, Jawa Timur | 22 |
| Gambar 2.10 Hirarki Pengelolaan Sampah..... | 26 |
| Gambar 2.11 Teknik Operasional Pengelolaan Persampahan | 29 |
| Gambar 2.12 Hasil Perbandingan Algoritma <i>Shuffle</i> | 31 |
| Gambar 2.13 <i>Pseudo Code Fisher Yates Shuffle</i> | 32 |
| Gambar 2.14 Gambaran Kerja <i>Forward Chaining / Bottom Up Reasoning</i> | 37 |
| Gambar 2.15 Logo Unity | 39 |
| Gambar 2.16 Asset Store..... | 42 |
| Gambar 2.17 Platform yang didukung Unity..... | 43 |
| Gambar 2. 18 Penggunaan Script pada Unity | 45 |
| Gambar 2.19 Tampilan Blender 2.75..... | 46 |
| Gambar 3.1 Mr Garbage | 54 |
| Gambar 3.2 Karakter Enemy..... | 55 |
| Gambar 3.3 Item Senjata..... | 55 |
| Gambar 3.4 Item Sapu Ajaib..... | 56 |
| Gambar 3.5 Tempat Sampah..... | 56 |
| Gambar 3.6 Sampah botol..... | 57 |
| Gambar 3.7 Sampah kaleng | 57 |
| Gambar 3.8 Sampah daun kering | 57 |
| Gambar 3.9 Sampah kertas | 57 |
| Gambar 3.10 <i>Finite State Machine</i> | 58 |
| Gambar 3.11 Flowchart Algoritma <i>Fisher Yates Shuffle</i> | 60 |
| Gambar 3.12 Blok Diagram Implementasi <i>Fisher Yates Shuffle</i> | 61 |
| Gambar 3.13 Kriteria kenaikan <i>level</i> | 64 |
| Gambar 3.14 Representasi Cara Kerja <i>Game</i> | 65 |
| Gambar 4.1 Tampilan <i>Splashscreen</i> | 75 |
| Gambar 4.2 Tampilan Menu <i>Introduction</i> (1)..... | 76 |
| Gambar 4.3 Tampilan Menu <i>Introduction</i> (2)..... | 77 |
| Gambar 4.4 Tampilan Menu <i>Introduction</i> (3)..... | 77 |
| Gambar 4.5 Tampilan Menu <i>Introduction</i> (4)..... | 78 |

| | |
|--|-----|
| Gambar 4.6 Tampilan <i>gameplay</i> dan posisi awal permainan | 78 |
| Gambar 4.7 Mulai permainan pada <i>level 1</i> | 79 |
| Gambar 4.8 Tampilan <i>gameplay</i> dan posisi awal | 79 |
| Gambar 4.9 Mulai permainan pada <i>level 2</i> | 80 |
| Gambar 4. 10 Tampilan <i>gameplay</i> dan posisi awal | 80 |
| Gambar 4. 11 Mulai permainan pada <i>level 3</i> | 81 |
| Gambar 4.12 Tampilan <i>gameplay</i> dan posisi awal | 81 |
| Gambar 4.13 Mulai permainan pada <i>level 4</i> | 82 |
| Gambar 4.14 Tampilan <i>gameplay</i> dan posisi awal | 82 |
| Gambar 4.15 Mulai permainan pada <i>level 5</i> | 83 |
| Gambar 4.16 Pemain menemukan sapu ajaib | 83 |
| Gambar 4.17 Tampilan pertanyaan | 84 |
| Gambar 4.18 Pemain mendapatkan senjata sapu | 84 |
| Gambar 4.19 Pemain bertemu kucing pengganggu | 85 |
| Gambar 4.20 Tampilan <i>interface pause</i> | 85 |
| Gambar 4.21 Tampilan <i>game over</i> | 86 |
| Gambar 4.22 Tampilan <i>level up</i> | 86 |
| Gambar 4.23 Tampilan <i>mission completed</i> | 87 |
| Gambar 4.24 Hasil pengacakan soal | 88 |
| Gambar 4.25 <i>Pseudo code</i> generator java | 93 |
| Gambar 4.26 Hasil <i>running</i> generator java | 93 |
| Gambar 4.27 Uji coba algoritma <i>Forward Chaining</i> jika nyawa penuh dan <i>score</i> sama dengan 100 | 105 |
| Gambar 4.28 Uji coba algoritma <i>Forward Chaining</i> jika nyawa habis dan <i>score</i> kurang dari 100 | 105 |

DAFTAR TABEL

| | |
|--|-----|
| Tabel 2. 1 Sumber Sampah di Masyarakat | 24 |
| Tabel 2.2 Pengerjaan metode <i>Fisher Yates Shuffle</i> | 33 |
| Tabel 2.3 Karakter <i>Forward Chaining</i> | 38 |
| Tabel 3.1 <i>Storyboard</i> | 50 |
| Tabel 3. 2 Ketentuan <i>Level</i> | 59 |
| Tabel 3.3 Data <i>Array</i> | 62 |
| Tabel 3.4 Iterasi Algoritma <i>Fisher Yates Shuffle</i> | 63 |
| Tabel 3.5 Variabel Aturan <i>Game Mr Garbage</i> | 64 |
| Tabel 3.6 Variabel Hasil <i>Game Mr Garbage</i> | 64 |
| Tabel 3.7 Variabel <i>Level Game Mr Garbage</i> | 64 |
| Tabel 3.8 Representasi Kondisi | 66 |
| Tabel 3.9 Representasi <i>Review</i> | 66 |
| Tabel 3. 10 Representasi <i>Level</i> | 66 |
| Tabel 3.11 Aturan <i>Game Mr Garbage</i> Berdasarkan <i>Levelling</i> | 67 |
| Tabel 3.12 Aturan <i>Game Mr Garbage</i> Berdasarkan <i>Review</i> | 67 |
| Tabel 4.1 Perangkat Keras Yang Digunakan Untuk Pembuatan <i>Game</i> | 69 |
| Tabel 4.2 Perangkat Lunak Yang Digunakan Untuk Pembuatan <i>Game</i> | 70 |
| Tabel 4.3 <i>Rule</i> pada <i>Knowledge Base</i> | 74 |
| Tabel 4.4 Keterangan Fungsi Tombol Menu <i>Game</i> | 76 |
| Tabel 4.5 Hasil Uji Coba Urutan Baru yang dihasilkan Algoritma <i>Fisher Yates Shuffle</i> | 89 |
| Tabel 4.6 Hasil Uji Coba Algoritma <i>Fisher Yates Shuffle</i> | 90 |
| Tabel 4.7 Frekuensi kemunculan angka..... | 91 |
| Tabel 4.8 Presentase Frekuensi perulangan kemunculan angka | 92 |
| Tabel 4.9 Hasil Pengujian Algoritma <i>Fisher Yates Shuffle</i> Pada <i>Game</i> | 94 |
| Tabel 4.10 Aturan <i>Level 1</i> | 95 |
| Tabel 4.11 Hasil Pengujian Algoritma <i>Forward Chaining Level 1</i> | 97 |
| Tabel 4.12 Aturan <i>Level 2</i> | 97 |
| Tabel 4.13 Hasil Pengujian Algoritma <i>Forward Chaining Level 2</i> | 98 |
| Tabel 4.14 Aturan <i>Level 3</i> | 99 |
| Tabel 4.15 Hasil Pengujian Algoritma <i>Forward Chaining Level 3</i> | 100 |
| Tabel 4.16 Aturan <i>Level 4</i> | 101 |
| Tabel 4.17 Hasil Pengujian Algoritma <i>Forward Chaining Level 4</i> | 102 |
| Tabel 4.18 Aturan <i>Level 5</i> | 102 |
| Tabel 4.19 Hasil Pengujian Algoritma <i>Forward Chaining Level 5</i> | 104 |
| Tabel 4.20 Kesimpulan Hasil Uji Coba <i>Forward Chaining</i> Pada <i>Game</i> | 104 |
| Tabel 4.21 Tabel uji coba <i>game</i> | 106 |

ABSTRAK

Nurarifah, Risti. 2018. **Penerapan Algoritma Fisher Yates Shuffle dan Algoritma Forward Chaining pada Game Edukasi Mr Garbage**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing (I) Yunifa Miftachul Arif, M.T (II) Hani Nurhayati, M.T

Kata Kunci: *Fisher Yates Shuffle, Forward Chaining, Pengacakan Soal, Levelling, Game Edukasi.*

Sampah merupakan material sisa yang tidak diinginkan setelah berakhirnya suatu proses dan erat kaitannya dengan kesehatan masyarakat. Sampai saat ini permasalahan sampah semakin hari tidak bisa dihindarkan. Akan semakin kompleks jika terus dibiarkan. Mulai dari berseraknya sampah yang mengganggu pemandangan sampai tersumbatnya aliran sungai dan akhirnya menyebabkan banjir. Permasalahan ini terjadi karena kurangnya kepedulian setiap orang untuk membuang sampah pada tempatnya. Demi mendukung menjaga lingkungan dari sampah maka butuh sarana untuk mawadahi pembelajaran lingkungan menjadi konten yang menarik. Konten permainan adalah salah satu media hiburan yang banyak digemari masyarakat. Konten yang unik merupakan salah satu daya tarik untuk mampu mendukung iklim belajar.

Penelitian ini menjelaskan bagaimana merancang kenaikan *level* pemain dan pengacakan pertanyaan pada suatu *game* dan juga bertujuan memberikan pengetahuan dan menciptakan kesadaran kepada anak-anak akan pentingnya menjaga lingkungan dari sampah. *Game Mr Garbage* adalah permainan berjenis edukasi berbasis *desktop* yang dibangun dengan memanfaatkan Unity3d. Pemain bertugas untuk menyelesaikan misi yang sarat akan nilai pengenalan terhadap lingkungan.

Implementasi kecerdasan buatan pada penelitian ini diterapkan pada pertanyaan dan *levelling* dengan memanfaatkan metode *Fisher Yates Shuffle* dan *Forward Chaining*. Metode *Fisher Yates Shuffle* digunakan sebagai pengacak pertanyaan dan metode *Forward Chaining* digunakan untuk menentukan kenaikan *level* pemain. Penelitian ini difokuskan pada *platform desktop*.

ABSTRACT

Nurarifah, Risti. 2018. **An Application of Fisher Yates Shuffle Algorithm and Forward Chaining Algorithm on Mr. Garbage's Education Game**. Thesis. Department of Informatics, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim of Malang.
Supervisor: (I) Yunifa Miftachul Arif, M.T (II) Hani Nurhayati, M.T

Keywords: Fisher Yates Shuffle, Forward Chaining, Task Randomization, Leveling, Educational Game

Waste is an unwanted waste material after the end of a process and is closely related to the public health. Until today, the problem of waste is increasingly inevitable. It will become more complex if it is not handled soon. Start from the scattering of garbage that disturbs the landscape and the clogging of the river, and eventually causes flooding. This problem occurs because of the lack of awareness of each person to dispose waste in its place. For supporting the environment from the waste, it needs a means to accommodate environmental learning into interesting content. Game content is one of the popular entertainment media. Unique content is one of the attractions to be able to support the learning climate.

The research explains the design level up player and randomizes questions on a game and also aims at providing knowledge and creates awareness to children about the importance in keeping the environment from waste. Game Mr. Garbage is a game education based on desktop that is built by Unity3d. Players are tasked with completing missions that are full of environmental recognition values.

The Implementation of artificial intelligence in this research was applied to the question and leveling by utilizing the methods Fisher Yates Shuffle and Forward Chaining. The Fisher Yates Shuffle method was used as a randomizer of questions and the Forward Chaining method was used to determine the level increase of the player. The research focused on desktop platforms.

ملخص البحث

نور عارفة، ريبستي. 2018. تطبيق خوارزمية فيشر ياتيس يوفيل (*Fisher Yates Shuffle*) وخوارزمية *Forward Chaining* في لعبة التعليمية مستير غارامب. البحث. قسم المعلوماتية كلية العلوم والتكنولوجيا الجامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانج.

الإشراف: يونيفا مفتاح العارف، الماجستير، وهاني نور حياتي، الماجستير

الكلمات الرئيسية: *Forward Chaining*، *Fisher Yates Shuffle*، العشوائية، المشكلة، الإستواء، لعبة تعليمية.

النفائيات هي مواد النفائيات غير مرغوبة بعد انتهاء العملية وترتبط ارتباطاً وثيقاً بالصحة العامة. حتى الآن أصبحت مشكلة النفائيات حتمية بشكل متزايد. سوف تصبح أكثر تعقيداً إذا تركت وحدها. بدءاً من تناثر النفائيات التي تزعج المناظر الطبيعية حتى انسداد النهر وتؤدي في نهاية المطاف إلى الفيضانات. تحدث هذه المشكلة بسبب نقص الوعي لدى كل شخص للتخلص من النفائيات في مكانها. لدعم البيئة من النفائيات، فإنها تأخذ وسيلة لاستيعاب التعلم البيئي في محتوى مثير للاهتمام. محتوى اللعبة هو واحد من وسائل الترفيه بكثير المحبوب. المحتوى الفريد هو أحد من العوامل الجذب لدعم مناخ التعلم.

يشرح هذا البحث عن تصميم زيادة المستوى اللاعب و العشوائية الأسئلة في لعبة ، ويهدف أيضاً إلى توفير المعرفة وخلق الوعي للأطفال حول أهمية الحفاظ على البيئة من النفائيات. اللعبة مستير غارامب هي لعبة تعليمية تعتمد على سطح المكتب الذي يبنى بواسطة Unity3d ، يتكلف اللاعب بإكمال المهام المليئة بقيم التعرف على البيئة.

تطبيق الذكاء الاصطناعي في هذا البحث ينطبق على السؤال والتسوية من خلال استخدام . تستخدم طريقة فيشر ياتيس يوفيل *Forward Chaining* و الأسلوب فيشر ياتيس يوفيل لتحديد زيادة مستوى *Forward Chaining* كمعالج عشوائي للأسئلة وتستخدم طريقة *platform desktop* اللاعب. ركز هذا البحث على سطح المكتب المنصة)

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Sampah merupakan material sisa yang tidak diinginkan setelah berakhirnya suatu proses dan erat kaitannya dengan kesehatan masyarakat. Di dalam sampah akan hidup berbagai macam mikroorganisme penyebab penyakit dan juga binatang serangga pemindah atau penyebar penyakit. Sampai saat ini permasalahan sampah semakin hari tidak bisa dihindarkan. Akan semakin kompleks jika terus dibiarkan. Mulai dari berseraknya sampah yang mengganggu pemandangan sampai tersumbatnya aliran sungai dan akhirnya menyebabkan banjir. Permasalahan ini terjadi karena kurangnya kepedulian setiap orang untuk membuang sampah pada tempatnya.

Di Indonesia jumlah sampah mengalami peningkatan seiring dengan pertumbuhan penduduk Indonesia. Kementerian Lingkungan Hidup mencatat selama 10 tahun terakhir, jumlah sampah terutama di perkotaan menunjukkan kenaikan. "Pada tahun 2005 timbunan sampah sebanyak 11%, naik menjadi 15% pada tahun 2015". Dalam perkembangannya, ternyata permasalahan sampah di Indonesia tersebut menjadi lebih kompleks dan meluas terutama terkait isu pencemaran sampah di laut. Isu pencemaran sampah di laut ini mulai menjadi perhatian publik dan pemerintah Indonesia terutama sejak dipublikasikannya hasil penelitian Jambeck, Jena r.,et.al, 2015 yang berjudul "*plastic waste inputs from land into the ocean*" (www.sciencemag.org, February 12, 2015) yang menyatakan potensi sampah plastik yang ada di lautan Indonesia mencapai 187,2 juta

ton/tahun. Hasil penelitian ini pun menyatakan bahwa Indonesia menjadi Negara kedua terbesar di dunia yang menyumbang sampah ke laut setelah Cina. Peningkatan jumlah sampah yang tidak diikuti oleh perbaikan dan peningkatan sarana dan prasarana pengelolaan sampah mengakibatkan permasalahan sampah menjadi kompleks. Dalam kurun waktu dua puluh tahun yang akan datang, mungkin dunia akan dipenuhi sampah dan terjadi banyak kerusakan yang diakibatkan dari sampah.

Pendidikan membuang sampah harus diberikan kepada setiap orang sedini mungkin, agar anak-anak semakin peduli terhadap lingkungan. Kita dapat melihat bagaimana keadaan sekarang, dimana banyak anak-anak yang kurang peduli terhadap apa yang dia lakukan yaitu membuang sampah dengan sembarangan. Selain itu peran orang tua dalam mendidik anaknya untuk membuang sampah pada tempatnya juga sangat kurang.

Budaya membuang sampah pada tempatnya merupakan salah satu hal kecil yang dapat dilakukan oleh manusia untuk mengatasi permasalahan lingkungan. Melalui budaya membuang sampah pada tempatnya, secara tidak langsung akan memberikan dampak positif, baik bagi lingkungan maupun bagi manusia. Sehingga membudayakan membuang sampah pada tempatnya merupakan hal wajib yang harus dibiasakan sejak dini.

Dalam agama Islam, telah mengajarkan kepada para pemeluknya agar mereka senantiasa menjaga kebersihan, baik kebersihan lahir maupun kebersihan bathin. Selain itu Allah SWT juga menyukai hal-hal yang suci dan bersih sebagaimana dalam hadist :

عَنْ سَعْدِ بْنِ أَبِي وَقَّاصٍ عَنْ أَبِيهِ عَنِ النَّبِيِّ مُحَمَّدٍ : إِنَّ اللَّهَ طَيِّبٌ يُحِبُّ
الطَّيِّبَ نَظِيفٌ يُحِبُّ النَّظَافَةَ كَرِيمٌ يُحِبُّ الْكِرَامَ جَوَادٌ يُحِبُّ الْجُودَ فَتَنَظَّفُوا
أَفْنِيَّتَكُمْ (رواه الترمذي)

Artinya :

“Diriwayatkan dari Sa’ad bin Abi Waqas dari ayahnya, dari Rasulullah saw : Sesungguhnya Allah SWT itu suci yang menyukai hal-hal yang suci, Dia Maha Bersih yang menyukai Kebersihan, Dia Mahamulia yang menyukai Kemuliaan, Dia Maha Indah yang menyukai Keindahan, karena itu bersihkanlah tempat-tempatmu. (HR.Tirmizi)

Dari hadist tersebut, dinyatakan bahwa “Kebersihan, Kesucian, dan Keindahan merupakan sesuatu yang disukai oleh Allah SWT. Jika kita melakukan sesuatu yang disukai oleh Allah SWT, tentu mendapatkan nilai di hadapan-Nya, yakni berpahala. Dengan kata lain, kotor, jorok, sampah berserakan, lingkungan yang semrawut dan tidak indah itu tidak disukai oleh Allah SWT. Sebagai hamba yang taat, tentu kita terdorong untuk melakukan hal-hal yang disukai oleh Allah SWT. Untuk mewujudkan kebersihan dan keindahan tersebut dapat dimulai dari diri kita sendiri, di lingkungan keluarga, masyarakat, maupun di lingkungan sekolah. Bentuknya juga sangat bermacam-macam, mulai dari membersihkan diri setiap hari, membersihkan kelas, menata ruang kelas sehingga tampak indah dan nyaman. Bila kita dapat mewujudkan kebersihan dan keindahan, maka kehidupan kita pasti terasa lebih nyaman. Hadist tentang menjaga kebersihan juga menyebutkan :

عَنْ أَبِي مَالِكٍ الْأَشْعَرِيِّ قَالَ قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ :
الطُّهُورُ شَطْرُ الْإِيمَانِ وَالْحَمْدُ لِلَّهِ الْمِيزَانُ وَسُبْحَانَ اللَّهِ وَالْحَمْدُ لِلَّهِ تَمْلَانِ
أَوْ تَمْلَانِ مَا بَيْنَ السَّمَاوَاتِ وَالْأَرْضِ وَالصَّلَاةُ نُورٌ وَالصَّدَقَةُ بُرْهَانٌ
وَالصَّبْرُ ضِيَاءٌ وَالْقُرْآنُ حُجَّةٌ لَكَ (رواه مسلم)

Artinya :

“Diriwayatkan dari Malik Al Asy’ari dia berkata, Rasulullah saw bersabda : Kebersihan adalah sebagian dari iman dan bacaan hamdallah dapat memenuhi mizan (timbangan), dan bacaan subhanallahi walhamdulillah memenuhi kolong langit dan bumi, dan shalat adalah cahaya dan shadaqah adalah pelita, dan sabar adalah sinar, dan al quran adalah pedoman bagimu. (HR.Muslim)

Dari hadist yang kedua dijelaskan bahwa kebersihan merupakan sebagian dari iman. Maksudnya adalah, keimanan seseorang akan menjadi lengkap kalau dia dapat menjaga kebersihan. Dengan kata lain, orang yang tidak dapat menjaga kebersihan berarti keimanannya masih belum sempurna. Secara tidak langsung hadist ini menandakan bahwa kebersihan bagi umat Islam merupakan sesuatu yang sangat penting untuk diterapkan. Dalam hadist mengenai kebersihan ini juga dirangkai dengan pernyataan Rasulullah sebagai berikut :

- a. Kebersihan sebagian dari iman.
- b. Berzikir dengan membaca “Alhamdulillah” itu memenuhi mizan (timbangan) amal baik kelak di hari kiamat.
- c. Berzikir “Subhanallah Walhamdulillah” pahalanya memenuhi kolong langit dan bumi.
- d. Shalat itu cahaya bagi umat Islam.
- e. Shadaqah itu pelita bagi umat Islam.
- f. Sabar itu sinar bagi umat Islam.
- g. Dan Al Quran merupakan pedoman hidup umat Islam.

Rangkaian hadist semacam ini secara tidak langsung juga sebagai isyarat bahwa menjaga kebersihan adalah sangat penting dan utama sebagaimana keutamaan dari zikir, shalat, shadaqah, dan sabar.

Sementara yang terjadi saat ini adalah media pendidikan tentang pengelolaan sampah yang ada masih kurang jumlahnya dan tidak menarik.

Menurut Daryanto (2014) menyebutkan bahwa materi dan metode pelaksanaan pendidikan lingkungan hidup yang selama ini digunakan dirasakan belum memadai sehingga pemahaman kelompok sasaran mengenai pelestarian lingkungan hidup menjadi tidak utuh. Disamping itu materi dan metode pelaksanaan pendidikan lingkungan hidup yang tidak aplikatif kurang mendukung penyelesaian permasalahan lingkungan hidup yang dihadapi di daerah masing-masing. Mengingat berkembangnya teknologi saat ini yang cukup sangat pesat peneliti mencoba memecahkan persoalan ini melalui rekayasa perangkat lunak yang dirasa cukup murah, populer dan digemari anak-anak.

Game merupakan sebuah bentuk hiburan yang banyak digemari oleh kalangan anak-anak, remaja bahkan dewasa. Bagi sebagian orang, *game* dijadikan sebagai penyegar pikiran dan untuk menghilangkan kejenuhan. Banyak sekali *game* yang menggambarkan dunia di dalamnya sedekat mungkin dengan kehidupan nyata. Untuk membuat *game* yang menarik mendekati kehidupan nyata perlu ditunjang oleh kemampuan untuk menciptakan karakter-karakter dalam *game* yang mampu merespon lingkungannya (Widiastuti, 2012).

Pembuatan *game* atau sebuah aplikasi pada umumnya diperlukan satu atau beberapa metode. Diantaranya yaitu metode *Fisher Yates Shuffle* dan *Forward Chaining*. Metode *Fisher Yates Shuffle* digunakan untuk pengacakan pertanyaan agar pertanyaan yang ditampilkan tidak tampil secara berulang-ulang. Sedangkan metode *Forward Chaining* digunakan untuk menentukan kenaikan *level* pemain berdasarkan kriteria yang telah ditetapkan.

Berdasarkan latar belakang tersebut, maka peneliti akan membuat *game* edukasi *Mr Garbage* menggunakan algoritma *Fisher Yates Shuffle* dan algoritma

Forward Chaining. *Game* yang dibuat ini nantinya akan dikemas dalam bentuk *game* edukasi interaktif yang memiliki unsur pembelajaran lingkungan. *Game* ini akan dibangun menggunakan Unity 3D.

Game yang diciptakan ini harapannya akan memberikan pengetahuan dan menciptakan kesadaran kepada anak-anak akan pentingnya menjaga lingkungan dari sampah. Pola hidup cinta lingkungan yang ditanamkan ini diawali dari hal-hal kecil dalam kehidupan sehari-hari yaitu membuang sampah secara benar.

1.2 Identifikasi Masalah

Berdasarkan latar belakang diatas, berikut adalah masalah yang dapat diidentifikasi:

1. Bagaimana menerapkan metode *Fisher Yates Shuffle* sebagai pengacak pertanyaan pada *game* edukasi *Mr Garbage*?
2. Bagaimana menerapkan metode *Forward Chaining* untuk mengatur penentuan kenaikan *level* pemain pada *game* edukasi *Mr Garbage*?

1.3 Tujuan Penelitian

Tujuan dari dibuatnya *game* ini adalah:

1. Merancang sebuah *game* yang memadukan *gameplay* dan pesan moral berupa sosialisasi *Reef Clean Up* melalui media permainan edukasi (*Education Game*).
2. Memberi manfaat sebagai sarana pembelajaran melalui *game* yang menyenangkan.
3. Mengimplementasikan algoritma *Forward Chaining* sebagai pengatur penentuan kenaikan *level* pemain.

4. Mengimplementasikan algoritma *Fisher Yates Shuffle* sebagai pengacak pertanyaan permainan.

1.4 Batasan Masalah

Untuk menjaga fokus dari penelitian ini, maka beberapa batasan yang diberikan adalah sebagai berikut :

1. *Game* ini bersifat *single player*.
2. *Game* ini berbasis *desktop*.
3. Konten yang disajikan merupakan pembelajaran Lingkungan.
4. *Game* ini dikhususkan untuk anak-anak tingkat pendidikan Sekolah Dasar umur 8-11 tahun.

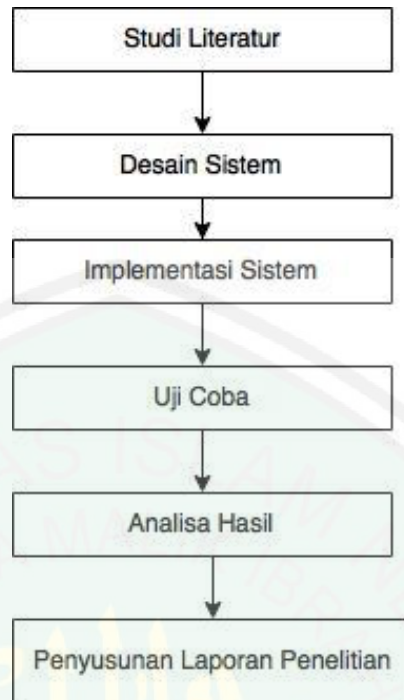
1.5 Manfaat Penelitian

Beberapa manfaat yang bisa diambil dari penelitian ini adalah :

1. Pemain bisa bermain *game* sekaligus belajar mengenai kebersihan lingkungan yang disajikan penulis dengan cara yang menyenangkan.
2. Pemain diharapkan mampu mengimplementasikan dalam kehidupan sehari-hari.

1.6 Metode Penelitian

Metode yang digunakan dalam penelitian ini adalah tentang *game* edukasi *Mr Garbage* untuk pembelajaran lingkungan dapat dilihat pada **Gambar 1.1**



Gambar 1.1 Alur Metode Penelitian

Berikut penjelasan dari **Gambar 1.1** alur metode penelitian:

1. Studi Literatur

Pada tahapan ini dilakukan pencarian dan pemahaman literature yang berhubungan dengan permasalahan pembuatan *game* edukasi *Mr Garbage* untuk pembelajaran lingkungan yang dapat diperoleh dalam bentuk *paper*, buku rujukan dan hasil *browsing* di internet yang terkait dengan penelitian ini.

2. Desain Sistem

Proses ini meliputi perancangan system pada *game*, *audio*, *desain interface*, bahasa pemrograman yang digunakan beserta seluruh bahan pembelajaran berupa data ataupun item yang terkait dengan *game*.

3. Implementasi Sistem

Tahap ini merupakan penerapan metode yang digunakan yaitu *Fisher Yates Shuffle* dan *Forward Chaining* dalam *game* edukasi *Mr Garbage* untuk pengacakan pertanyaan dan sebagai penentu kenaikan *level* pemain.

4. Uji Coba

Dalam proses ini akan dilakukan serangkaian uji coba baik sebelum *game* rampung maupun setelahnya, untuk mencegah terjadinya kesalahan dalam *game* ini.

5. Analisa Hasil

Pada tahap ini akan dianalisa ketepatan hasil dari metode yang digunakan yaitu *Fisher Yates Shuffle* dan *Forward Chaining* terhadap *output* yang diinginkan yaitu penentu kenaikan *level game* edukasi *Mr Garbage*.

6. Penyusunan Laporan Penelitian

Pada tahapan ini dilakukan penulisan laporan penelitian yang merupakan dokumentasi dari konsep atau teori penunjang, perancangan *game*, pembuatan *game*, dan dokumentasi dari uji coba dan analisis, serta kesimpulan dan saran.

1.7 Sistematika Penulisan

1. BAB I PENDAHULUAN

Bab ini berisi tentang penjelasan penulis menyusun laporan penelitian ini beserta manfaat dan tujuan.

2. BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan tentang teori dasar dan data-data yang berhubungan dengan pembuatan *game*.

3. BAB III DESAIN DAN RANCANGAN GAME

Bab ini berisi desain rancangan *game* maupun implementasi algoritma *Fisher Yates Shuffle* dan algoritma *Forward Chaining* untuk pengacak pertanyaan dan sebagai penentu kenaikan *level* pemain.

4. BAB IV IMPLEMENTASI

Bab ini berisi tentang implementasi algoritma *Fisher Yates Shuffle* dan algoritma *Forward Chaining* terhadap *game* beserta hasil uji coba pada *game*.

5. BAB V PENUTUP

Bab ini berisi tentang kesimpulan secara keseluruhan pada penelitian terhadap algoritma *Fisher Yates Shuffle* dan algoritma *Forward Chaining* untuk pengacakan pertanyaan dan sebagai penentu kenaikan *level* pemain.

BAB II

KAJIAN PUSTAKA

2.1 Landasan Teori

2.1.1 *Game*

Game adalah sebuah sistem di mana pemain terlibat dalam konflik buatan, didefinisikan oleh aturan, yang menghasilkan hasil kuantitatif. Permainan muncul dari hubungan antara tindakan pemain dan sistem, itu adalah proses dimana seorang pemain mengambil tindakan dalam sistem yang dirancang dari sebuah permainan dan sistem merespon tindakan. Artinya dari suatu tindakan dalam permainan ada hubungan antara tindakan dan hasil (Salen dan Zimmerman, 2004:80).

Agustinus Nilwan (1998) dalam bukunya yang berjudul “Pemrograman Animasi dan Game Profesional 4” mendefinisikan *game* sebagai:

“Permainan komputer yang dibuat dengan teknik dan metode animasi. Jika ingin mendalami penggunaan animasi haruslah memahami pembuatan *game*, sebaliknya juga berlaku demikian sebab keduanya saling berkaitan.”

Menurut Semiawan (2007: 19-20), “berbagai *game* (permainan) dapat dibuat secara sengaja (*intentionality*) dengan maksud agar anak meningkatkan beberapa kemampuan tertentu berdasarkan pengalaman belajar. Melalui aktivitas bermain, berbagai pekerjaanya terwujud. Bermain adalah aktivitas yang dipilih sendiri oleh anak karena menyenangkan, bukan karena akan memperoleh hadiah atau pujian”.

Teori permainan (*game*) pertama kali ditemukan oleh sekelompok ahli matematika pada tahun 1944. Teori itu dikemukakan oleh John Von Neumann and Oskar Morgenstern yang berisi “permainan terdiri atas sekumpulan peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun untuk memaksimalkan kemenangan sendiri atau pun untuk meminimalkan kemenangan lawan. Peraturan–peraturan menentukan kemungkinan tindakan untuk setiap pemain, sejumlah keterangan diterima setiap pemain sebagai kemajuan bermain, dan sejumlah kemenangan atau kekalahan dalam berbagai situasi.

2.1.2 *Genre Game*

Jenis-jenis *game* dikenal dengan istilah *genre game*. *Genre* sebuah *game* bisa murni ataupun campuran dari beberapa *genre* lain. Berikut beberapa *genre game* (Henry Samuel, 2005:44):

1. *Action Game*

Game jenis ini adalah *game* yang menawarkan aksi sebagai daya tarik utama. Kecepatan refleks, koordinasi mata-tangan juga *timing* merupakan keterampilan utama yang dibutuhkan dalam memainkan *game* ini. Contoh dari *game* ini adalah the elder scrolls v : skyrim seperti pada **Gambar 2.1**.



Gambar 2.1 *Game the elder scrolls v : skyrim*

2. *Adventure Game*

Adventure game bernuansa petualangan. Memasuki gua bawah tanah, melompati bebatuan di antara lahar, bergelayutan dari pohon satu ke pohon lain, bergulat dengan ular sambil mencari kunci untuk membuka pintu kuil legendaris, atau sekedar mencari telepon umum untuk mendapatkan misi berikutnya. Itulah beberapa dari banyak hal yang karakter pemain harus lakukan dan lalui dalam *game* jenis ini. Contoh dari *game* ini adalah *ninja blade* seperti pada

Gambar 2.2.



Gambar 2.2 *Game Ninja Blade*

3. *Puzzle Game*

Game jenis ini sesuai dengan namanya berintikan mengenai pemecahan teka-teki, baik itu menyusun balok, menyamakan warna bola, memecahkan perhitungan matematika, melewati labirin, sampai mendorong-dorong kota masuk ke tempat yang seharusnya, itu semua termasuk dalam jenis ini. Sering pula permainan jenis ini adalah juga unsur permainan dalam *video game* petualangan maupun *game* edukasi. Contoh *game* ini dapat dilihat pada **Gambar 2.3**.



Gambar 2.3 *Game* Teka Teki Silang

4. *Role Playing Game*

Game jenis ini sesuai dengan terjemahannya, bermain peran, memiliki penekanan pada tokoh atau peran perwakilan permainan di dalam permainan yang biasanya adalah toko utamanya dimana seiring kita memainkannya. Karakter tersebut dapat berubah dan berkembang ke arah yang diinginkan pemain (biasanya menjadi semakin hebat, semakin kuat, semakin berpengaruh dll). Dalam

berbagai parameter yang biasanya ditentukan dengan naiknya *level*, baik dari status kepintaran, kecepatan dan kekuatan karakter, senjata yang semakin sakti, ataupun jumlah teman maupun makhluk peliharaan. Salah satu *game* yang terkenal dengan RPG adalah ragnarok seperti pada **Gambar 2.4**.



Gambar 2.4 *Game* Ragnarok

5. *Simulation Game*

Game jenis ini seringkali menggambarkan dunia di dalamnya sedekat mungkin dengan dunia nyata dan memperhatikan dengan detail berbagai faktor. Dari membangun rumah gedung hingga kota, mengatur pajak dan dana kota hingga keputusan memecat atau menambah karyawan. *Game* jenis ini membuat pemain harus berpikir untuk mendirikan, membangun dan mengatasi masalah dengan menggunakan dana yang terbatas. Contoh *game* yang populer dari jenis ini adalah championship manager seperti pada **Gambar 2.5**.



Gambar 2.5 *Game Champhionship Manager*

6. *Strategy Game*

Game jenis strategi, layaknya bermain catur justru lebih memerlukan keahlian berpikir dan memutuskan setiap gerakan secara hati-hati dan terencana. *Game* strategi biasanya memberikan pemain atas kendali tidak hanya satu orang tapi minimal sekelompok orang dengan berbagai jenis tipe kemampuan sampai kendaraan bahkan hingga pembangunan berbagai pembangunan, pabrik dan pusat pelatihan tempur tergantung dari tema ceritanya. Unsur-unsur permainannya biasanya berkisar sekitar prioritas pembangunan, peletakan pasukan, mencari dan memanfaatkan sumberdaya (uang, besi, kayu dll). Contoh dari *game* ini adalah comand & conquer 3 : kane wrath seperti pada **Gambar 2.6**.



Gambar 2.6 *Game Comand & Conquer 3 : Kane Wrath*

7. *Education Game*

Game edukasi merupakan paket *software* yang menciptakan kemampuan pada lingkungan *game* yang diberikan sebagai alat bantu untuk memotivasi atau membantu siswa untuk melalui prosedur *game* secara teliti untuk mengembangkan kemampuannya. *Developer* yang membuatnya harus memperhitungkan berbagai hal agar *game* ini benar-benar dapat mendidik, menambah pengetahuan dan meningkatkan keterampilan yang memainkannya. Target segmentasi pemain harus pula disesuaikan dengan tingkat kesulitan dan *design visual* ataupun animasinya. Contoh *education game* dapat dilihat seperti **Gambar 2.7**.



Gambar 2.7 *Game Tajwid Petualangan*

2.1.3 *Game* Sebagai Edukasi

Game sebagai edukasi adalah merupakan salah satu jenis media yang digunakan untuk memberikan pengajaran, menambah pengetahuan penggunanya melalui suatu media unik dan menarik. Jenis ini biasanya ditujukan untuk anak-anak, maka permainan warna sangat diperlukan disini bukan tingkat kesulitan yang dipentingkan.

Berdasarkan uraian diatas dapat disimpulkan *game* edukasi adalah salah satu bentuk *game* yang dapat berguna untuk menunjang proses belajar-mengajar secara lebih menyenangkan dan lebih kreatif, dan digunakan untuk memberikan pengajaran atau menambah pengetahuan penggunanya melalui suatu media yang menarik. Karakteristik *game* yang menyenangkan, memotivasi, membuat kecanduan dan membuat aktifitas ini digemari oleh banyak orang.

Bisnis *game* juga sudah merambah kemana-mana, namun ironisnya konten dari *game* sebagian besar berisi hiburan dan sangat sedikit yang berkonten pendidikan (edukasi). Sebenarnya tanpa disadari *game* dapat mengajarkan banyak

keterampilan dan *game* dapat dijadikan sebagai salah satu alternatif pendidikan (Buckingham dan Scalon, 2002).

Manfaat yang bisa didapat dari penggunaan media pembelajaran diantaranya :

1. Membawa kesegaran dan variasi bagi pengalaman belajar siswa.
2. Membuat hasil belajar lebih bermakna bagi berbagai kemampuan siswa.
3. Mendorong pemanfaatan yang bermakna dari mata pelajaran dengan melibatkan imajinasi dan partisipasi aktif yang mengakibatkan meningkatnya hasil belajar.
4. Memperluas wawasan dan pengalaman siswa yang mencerminkan pembelajaran nonverbalistik dan membuat generalisasi yang tepat.
5. Pembelajaran dapat dilakukan secara mantap karena meningkatnya kemampuan manusia untuk memanfaatkan media komunikasi, informasi dan data secara lebih konkrit dan rasional.
6. Mengatasi keterbatasan ruang, waktu dan daya indera

Dengan demikian media pembelajaran mempunyai peranan yang sangat besar dalam penyampaian pesan dalam proses pembelajaran yang diharapkan dapat meningkatkan hasil belajar siswa.

2.2 Sampah

2.2.1 Pengertian Sampah

Permasalahan lingkungan saat ini ada di berbagai tempat. Permasalahan itu menyangkut berbagai pencemaran, baik pencemaran tanah, air, udara dan suara. Pencemaran tersebut diakibatkan oleh aktivitas manusia. Pencemaran tanah

misalnya, banyaknya sampah yang tertimbun di tempat sampah apabila tidak ditangani dengan baik akan menurunkan tingkat kesehatan masyarakat.

Berdasarkan SK SNI Tahun 1990, sampah adalah limbah yang bersifat padat terdiri dari zat organik dan zat anorganik yang dianggap tidak berguna lagi dan harus dikelola agar tidak membahayakan lingkungan dan melindungi investasi pembangunan.

Sampah adalah istilah umum yang sering digunakan untuk menyatakan limbah padat. Sampah adalah sisa-sisa bahan yang mengalami perlakuan-perlakuan, baik karena telah sudah diambil bagian utamanya, atau karena pengolahan, atau karena sudah tidak ada manfaatnya yang ditinjau dari segi social ekonomis tidak ada harganya dan dari segi lingkungan dapat menyebabkan pencemaran atau gangguan terhadap lingkungan hidup. (S. Hadiwiyoto, *Penanganan dan Pemanfaatan Sampah*, Jakarta: Yayasan Idayu, 1983)

Sampah adalah bahan yang terbuang atau dibuang dari hasil aktifitas manusia maupun proses alam yang belum memiliki nilai ekonomi. Menurut kamus istilah lingkungan hidup, sampah mempunyai definisi sebagai bahan yang tidak mempunyai nilai, bahan yang tidak berharga untuk maksud biasa, pemakaian bahan rusak, barang yang cacat dalam pembuatan manufaktur, materi berlebihan, atau bahan yang ditolak. (E. Colink, *Istilah Lingkungan Untuk Manajemen*, 1996)

Dari pengertian diatas dapat disimpulkan bahwa sampah sesuatu yang tidak berguna lagi, dibuang oleh pemiliknya dari pemakai semula, atau sampah adalah sumberdaya yang tidak siap pakai.

2.2.2 Kondisi Sampah di Indonesia

Kementerian Lingkungan Hidup dan Kehutanan (KLHK) menilai persoalan sampah sudah meresahkan. Indonesia bahkan masuk dalam peringkat kedua di dunia sebagai penghasil sampah plastik ke Laut setelah Tiongkok. Hal itu berkaitan dengan data dari KLHK yang menyebut plastik hasil dari 100 toko atau anggota Asosiasi Pengusaha Ritel Indonesia (APRINDO) dalam waktu satu tahun sajam sudah mencapai 10,95 juta lembar sampah kantong plastik. Jumlah itu ternyata setara dengan luasan 65,7 hektare kantong plastik atau sekitar 60 kali luas lapangan sepak bola. Padahal, KLHK menargetkan pengurangan sampah plastik lebih dari 1,9 juta ton hingga 2019.

Dirjen Pengelolaan Sampah, Limbah, dan B3 KLHK Tuti Hendrawati Mintarsih menyebut total jumlah sampah Indonesia pada tahun 2019 akan mencapai 68 juta ton, dan sampah plastik diperkirakan akan mencapai 9,52 juta ton atau 14 persen dari total sampah yang ada.

Berdasarkan data Jambeck (2015), Indonesia berada di peringkat kedua dunia penghasil sampah plastik ke laut yang mencapai sebesar 187,2 juta ton setelah Cina yang mencapai 14,6 juta ton per tahun. Berada di urutan ketiga adalah Filipina yang menghasilkan sampah plastik ke laut mencapai 83,4 juta ton, diikuti Vietnam yang mencapai 55,9 juta ton, dan Sri Lanka yang mencapai 14,6 juta ton per tahun.

Permasalahan sampah sudah meresahkan tiap tahunnya kota-kota di dunia menghasilkan sampah hingga 1,3 miliar ton. Diperkirakan oleh Bank Dunia pada tahun 2025 jumlah ini bertambah hingga 2,2 miliar ton.



Gambar 2. 8 Sampah Jakarta
Sumber : Waste4Change, Dinas Kebersihan Jakarta dan KLH.

Sampah Jakarta menghasilkan 6.270 ton Per hari, sama dengan berat 30 Paus biru dewasa. Sekitar 79% sampah di angkut ke TPST Bantar Gebang, Bekasi. Dan sekitar 21% sampah tidak terangkut. Total 19% sampah yang dipilah. Dan selebihnya dibuang dan dibakar. Kondisi sampah di jakarta dapat dilihat pada **Gambar 2.8** diatas.



Gambar 2. 9 Sampah Gunung Semeru, Jawa Timur
Sumber : BBC Indonesia

Data Balai Besar Taman Nasional Bromo Tengger Semeru menunjukkan setiap pengunjung membuang sekitar 0,5 kilogram sampah di Gunung Semeru. Padahal, setiap hari gunung tersebut disambangi 200 hingga 500 pendaki. Sampah ditinggalkan begitu saja oleh para pendaki dan pengunjung gunung, tanpa ada yang berinisiatif membawanya ke tempat pembuangan di bagian bawah gunung. Kondisi sampah yang berada di gunung semeru dapat dilihat seperti **Gambar 2.9**.

2.2.3 Sumber Sampah

Sampah berasal dari kegiatan penghasil sampah seperti pasar, rumah tangga, perkotaan (kegiatan komersil/ perdagangan), penyapuan jalan, taman, atau tempat umum lainnya, dan kegiatan lain seperti dari industri dengan limbah yang sejenis sampah. Sumber dari sampah di masyarakat pada umumnya, berkaitan erat dengan penggunaan lahan dan penempatan. Di Indonesia sekitar 60-70% dari total volume sampah yang dihasilkan merupakan sampah basah dengan kadar air antara 65-75%. Sumber sampah terbanyak berasal dari pasar tradisional dan pemukiman. Sampah pasar tradisional, seperti pasar lauk-pauk dan sayur-mayur membuang hampir 95% sampah organik. Di dalam kehidupan manusia, sebagian besar jumlah sampah berasal dari beberapa aktivitas. Beberapa sumber sampah dapat diklasifikasikan menjadi antara lain: Perumahan, Institusi, Konstruksi dan pembongkaran, Pelayanan perkotaan, Unit pengolahan, Industri, Pertanian. (E. Damanhuri, Bandung: Teknik Lingkungan ITB, 1999). Klasifikasi di atas dapat dilihat lebih jelas pada **Tabel 2.1** berikut:

Tabel 2. 1 Sumber Sampah di Masyarakat

| Sumber | Fasilitas, aktifitas, lokasi sampah dihasilkan | Tipe sampah |
|---|--|--|
| Perumahan | Keluarga kecil atau beberapa keluarga tinggal bersama, apartemen kecil, menengah, dan tingkat tinggi | Sampah makanan, kertas, kardus, plastik, tekstil, kulit, sampah kebun, kayu, kaca, kaleng timah, aluminium |
| Komersil | Toko, restoran, pasar, bangunan kantor, hotel, motel, percekatan unit pelayanan, bengkel, dan lain-lain. | Kertas, kardus, plastik, kayu, sampah makanan, kaca, logam, sampah khusus, sampah berbahaya dan lain-lain. |
| Institusi | Sekolah, rumah sakit, penjara, pusat pemerintah | Kertas, kardus, plastik, kayu, sampah makanan, kaca, logam, sampah khusus, sampah berbahaya dan lain-lain. |
| konstruksi pembongkaran | Area konstruksi baru, area renovasi/ perbaikan jalan, peruntuhan bangunan, perkerasan yang rusak | Kayu, baja, beton, tanah |
| Pelayanan perkotaan (tidak termasuk fasilitas pengolahan) | Pembersihan jalan, pertamanan, pembersihan cekungan, area parkir dan pantai, tempat rekreasi lainnya. | Sampah khusus, kotoran, hasil penyapuan jalan, sisa penghiasan pohon dan pertamanan, sampah umum dari area parkir, pantai dan tempat rekreasi. |
| Unit pengolahan; insinerator kota | Proses pengolahan air, air limbah, dan lain-lain | Limbah unit pengolahan, pada dasarnya terdiri dari residu lumpur. |
| Sampah perkotaan | (seluruh sampah di atas) | (seluruh sampah di atas) |
| Industri | Konstruksi, fabrikasi, produksi ringan dan berat, unit kimia, pembangkit energi, pembongkaran dan lain-lain. | Limbah proses industri, potongan material, dan lain-lain. |
| Pertanian | Tanaman baris, kebun buah-buahan, kebun anggur, produksi susu, peternakan dan lain-lain. | Sampah makanan yang rusak, sampah pertanian, kotoran, sampah berbahaya. |

2.2.4 Jenis Sampah

Berdasarkan bahan asalnya sampah dibagi menjadi dua jenis yaitu sampah organik dan anorganik. (Cecep Dani Sucipto, Teknologi Pengolahan Daur Ulang Sampah, Jakarta: Goysen Publishing, 2009 hlm 2-3)

a. Sampah Organik

Sampah organik yaitu buangan sisa makanan misalnya daging, buah sayuran dan sebagainya. Contoh sampah organik adalah: potongan-potongan/ pelat-pelat dari logam, berbagai jenis batu-batuan, pecahan-pecahan gelas, tulang-belulang, dan lain-lain. Sampah jenis ini, melihat fisiknya keras maka baik untuk peninggian tanah rendah atau dapat pula untuk memperluas jalan setapak. Tetapi bila rajin mengusahakannya sampah dari logam dapat kembali dilebur untuk dijadikan barang yang berguna, batu-batuan untuk mengurung tanah yang rendah atau memperkeras jalan setapak, pecahan gelas dapat dilebur kembali dan dijadikan barang-barang berguna, dan tulang-belulang bila dihaluskan (dan diproses) dapat untuk pupuk dan lain-lain. (Alex S, sukses mengolah sampah organik menjadi pupuk organik, hlm 9-10).

b. Sampah Anorganik

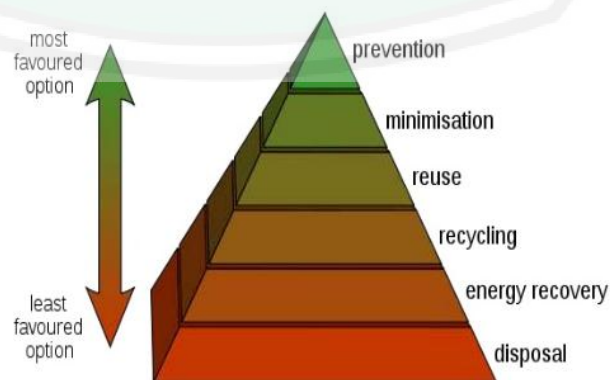
Sampah anorganik yaitu sisa material sintesis misalnya plastik, kertas, logam, kaca, keramik dan sebagainya. Melihat proses penghancurannya oleh jasad-jasad mikroba, maka sampah zat anorganik terdiri atas:

- 1) Zat anorganik dari bahan plastik.

2) Zat anorganik non-plastik.

2.2.5 Pengelolaan Persampahan

Pengelolaan sampah merupakan kegiatan yang sistematis, menyeluruh, dan berkesinambungan yang meliputi pengurangan dan penanganan sampah. Pengelolaan sampah adalah pengumpulan, pengangkutan, pemrosesan, pendaur-ulangan, atau pembuangan dari material sampah. Kalimat ini biasanya mengacu pada material sampah yang dihasilkan dari kegiatan manusia, dan biasanya dikelola untuk mengurangi dampaknya terhadap kesehatan, lingkungan atau keindahan. Pengelolaan sampah juga dilakukan untuk memulihkan sumber daya alam. Pengelolaan sampah bisa melibatkan zat padat, cair, gas, atau radioaktif dengan metode dan keahlian khusus untuk masing-masing jenis zat. Metode pengelolaan sampah berbeda beda tergantung banyak hal, diantaranya tipe zat sampah, tanah yang digunakan untuk mengolah dan ketersediaan area. (Yudhi Kartikawan, Pengelolaan persampahan, Yogyakarta: Jurnal Lingkungan Hidup, 200). Adapun beberapa konsep tentang pengelolaan sampah yang berbeda dalam penggunaannya dapat dilihat pada **Gambar 2.10** hirarki pengelolaan sampah dibawah berikut ini :



Gambar 2.10 Hirarki Pengelolaan Sampah

Hirarki-hirarki sampah merujuk kepada “3 M” mengurangi sampah, menggunakan kembali sampah dan daur ulang, yang mengklasifikasikan strategi pengelolaan sampah sesuai dengan keinginan dari segi minimisasi sampah. Hirarki sampah tetap menjadi dasar dari sebagian besar strategi minimalisasi sampah. Tujuan hirarki ini adalah untuk mengambil keuntungan maksimum dari produk-produk praktis dan untuk menghasilkan limbah dalam jumlah maksimum.

Pengacuan sistem pengelolaan sampah terdiri dari aspek manajemen, pengelolaan persampahan, aspek pembiayaan, aspek pengaturan, aspek peran serta masyarakat dan aspek teknik operasional. Dapat dilihat penjelasannya sebagai berikut :

1. Aspek Organisasi dan Manajemen

Aspek ini mempunyai peranan pokok: menggerakkan, mengaktifkan dan mengarahkan sistem manajemen persampahan kota. Sub sistem ini meliputi bentuk serta pola organisasi dan komponen pelengkapannya, yakni persoalan serta sistem manajemen. Struktur manajemen meliputi perencanaan, pelaksanaan dan pengendalian untuk jenjang strategis, teknik maupun operasional.

2. Aspek Pembiayaan

Aspek ini merupakan komponen sumber dalam arti supaya sistem mempunyai kinerja yang baik. Sub sistem ini diatur dengan struktur pembiayaan dalam bentuk anggaran serta alternatif sumber pendanaan.

3. Aspek Pengaturan

Aspek ini merupakan komponen yang menjaga pola/dinamika sistem agar dapat mencapai sasaran secara efektif. Umumnya kompleksitas permasalahan justru diredam oleh penerbitan peraturan yang mengatur seluruh komponen yang secara umum dibagi menjadi tiga kelompok, yaitu:

- a. Sebagai landasan pendirian instansi pengelola (dinas perusahaan daerah dan lainnya).
- b. Sebagai landasan pemberlakuan struktur tarif.
- c. Sebagai landasan ketertiban umum (masyarakat) dalam pengelolaan persampahan.

4. Aspek Peran Serta Masyarakat

Dalam kondisi keterbatasan kemampuan sistem, yakni penyediaan kapasitas kerja maupun pendanaan, maka salah satu alternatif adalah peran serta masyarakat. Untuk itu diperlukan upaya untuk menumbuhkan peran masyarakat dalam pengelolaan sampah dengan membentuk program yang dilaksanakan secara terarah, intensif, dan berorientasi kepada penyebaran pengetahuan penanaman kesadaran, peneguhan sikap dan pembentukan perilaku.

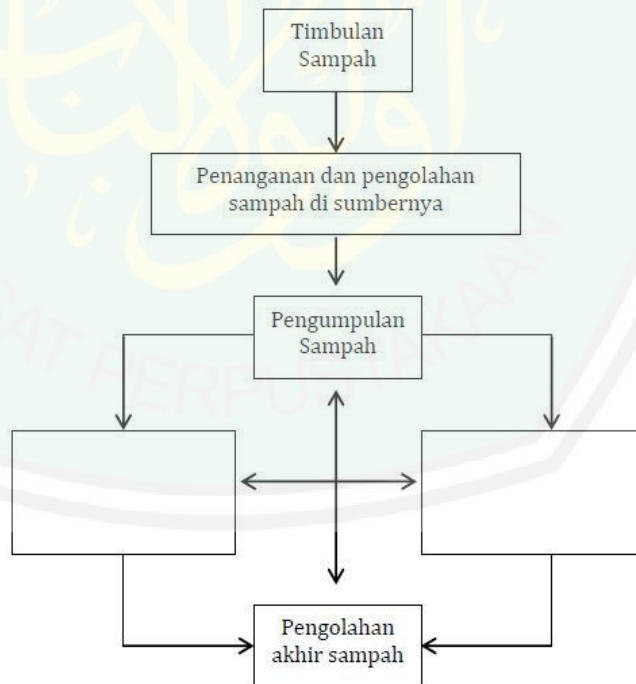
5. Aspek Teknik Operasional

Aspek ini merupakan komponen yang paling dekat dengan obyek pengelolaan sampah. Aspek ini terdiri dari perangkat keras, misalnya: sarana pewadahan, pengumpulan, pengangkutan dan pembuangan akhir. Disini permasalahan yang timbul pada umumnya berkisar pada

perbedaan yang jauh antara kebutuhan dan kapasitas operasi yang dapat disediakan oleh sistem.

Tchobanolous, dkk (1993), berdasarkan 6 komponen utama dalam sistem pengelolaan sampah terpadu, maka teknik operasional persampahan terdiri dari :

- a. Penentuan/perhitungan jumlah timbulan sampah.
- b. Penanganan dan pengolahan sampah di sumbernya.
- c. Pengumpulan sampah.
- d. Pemisahan, proses pengolahan dan perubahan sampah.
- e. Pemandahan dan pengangkutan sampah.
- f. Pengolahan akhir sampah.



Gambar 2.11 Teknik Operasional Pengelolaan Persampahan

2.3 Algoritma *Fisher Yates Shuffle*

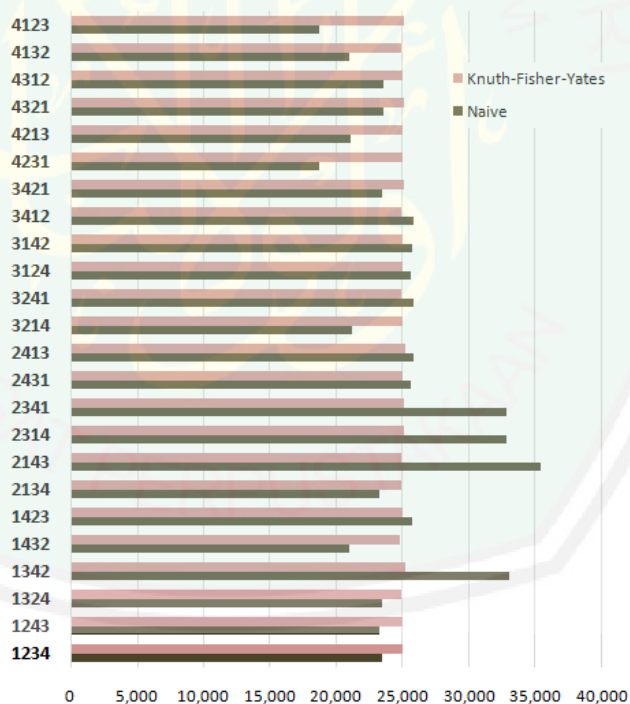
Fisher Yates Shuffle (dinamai berdasarkan penemunya, Ronald Fisher dan Frank Yates), juga dikenal sebagai *Knuth Shuffle* (diambil dari nama Donald Knuth) adalah sebuah algoritma untuk menghasilkan permutasi acak dari suatu himpunan terhingga, dengan kata lain untuk mengacak suatu himpunan tersebut.

Pemakaian *Fisher Yates Shuffle* bisa melalui dua cara yaitu: *Original Method* dan *Modern Method*. Menurut Pavel Micka (2011) *Original Method* dipublikasikan pada tahun 1938, pada metode ini dilakukan dengan cara penarikan secara berulang dari unsur daftar masukan kemudian menuliskannya ke daftar keluaran kedua. Pendekatan ini dilakukan oleh manusia dengan secarik kertas dan sebuah pensil.

Pada *Modern Method* dijabarkan untuk penggunaan komputerisasi yang dikenalkan oleh Richard Durstenfield pada tahun 1964. *Modern Method* dikenalkan karena lebih optimal dibandingkan dengan *Original Method*. Algoritma yang modern berbeda dari yang sebelumnya, sangat komputasi dan matematis. Prosesnya angka terakhir akan dipindahkan ke angka yang ditarik keluar dan mengubah angka yang ditarik keluar menjadi angka akhir yang tidak ditarik lagi untuk setiap kali penarikan dan berlanjut untuk iterasi berikutnya. Hal ini dilakukan dalam $O(1)$ waktu dan ruang. Dengan demikian, waktu dan ruang kompleksitas algoritmanya $O(n)$ yang optimal (O'Connor dan Derek, 2014)

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 2.12**

Pada **Gambar 2.12** dibawah ini, 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 *Naive Shuffle*. Hasilnya algoritma *Fisher Yates Shuffle* menghasilkan nilai yang hampir sama untuk setiap kemungkinan kombinasi kartu, sedangkan pada algoritma *Naive Shuffle* kombinasi kartu tertentu muncul jauh lebih sering dan sebagian lagi muncul lebih sedikit dibanding kombinasi kartu lain. (blog.codinghorror.com)



Gambar 2.12 Hasil Perbandingan Algoritma *Shuffle*

Langkah-langkah pengacakan menggunakan algoritma *Fisher Yates Shuffle* adalah sebagai berikut:

1. Siapkan masukan (dalam bentuk angka ataupun karakter) sebanyak N.

2. Ambil sebuah nomor acak k diantara satu sampai i jumlah angka yang belum dianggap teracak (dicoret).
3. Hitung dari bawah, coret angka k yang belum dicoret dan tuliskan angka tersebut di lain tempat.
4. Ulangi langkah 2 dan langkah 3 sampai semua angka sudah tercoret.
5. Urutan angka yang dituliskan pada langkah 3 adalah permutasi acak dari angka awal.

Pseudo code algoritma *Fisher Yates Shuffle* adalah sebagai berikut:

```

function algoFisherYate (A)
  for  $i \leftarrow A.length-1$  down to 1 do
     $s = \text{random number from } 0 \text{ to } i$ 
    swap(A[i],A[s])
  endfor

```

Gambar 2.13 *Pseudo Code Fisher Yates Shuffle*

Sebagai contoh, apabila ada angka dengan urutan 1 2 3 4 5 6 7 8. Maka proses pengacakannya dengan algoritma *Fisher Yates Shuffle* adalah sebagai berikut:

| <i>Range</i> | <i>Roll</i> | <i>Scratch</i> | <i>Result</i> |
|--------------|-------------|-----------------|---------------|
| | | 1 2 3 4 5 6 7 8 | |

Pertama pilih angka acak 1-8, pilih 4 kemudian menukar angka urutan keempat yaitu angka 4 dengan angka urutan kedelapan atau angka terakhir yaitu angka 8.

| <i>Range</i> | <i>Roll</i> | <i>Scratch</i> | <i>Result</i> |
|--------------|-------------|----------------|---------------|
| 1-8 | 4 | 1 2 3 8 5 6 7 | 4 |

Angka acak selanjutnya dari 1-7, pilih 6 kemudian tukar angka urutan keenam yaitu angka 6 dengan angka urutan ketujuh atau angka terakhir yaitu angka 7.

| <i>Range</i> | <i>Roll</i> | <i>Scratch</i> | <i>Result</i> |
|--------------|-------------|----------------|---------------|
| 1-7 | 6 | 1 2 3 8 5 7 | 6 4 |

Angka acak berikutnya dari 1-6, 1-5 dan seterusnya, sehingga dengan mengulangi langkah-langkah seperti diatas akan didapatkan hasil seperti **Tabel 2.2** berikut:

Tabel 2.2 Pengerjaan metode *Fisher Yates Shuffle*

| <i>Range</i> | <i>Roll</i> | <i>Scratch</i> | <i>Result</i> |
|--------------|-------------|----------------|-----------------|
| 1-6 | 2 | 1 7 3 8 5 | 2 6 4 |
| 1-5 | 7 | 1 5 3 8 | 7 2 6 4 |
| 1-4 | 1 | 8 5 3 | 1 7 2 6 4 |
| 1-3 | 3 | 8 5 | 3 1 7 2 6 4 |
| 1-2 | 5 | 8 | 5 3 1 7 2 6 4 |
| | | | 8 5 3 1 7 2 6 4 |

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

Menurut Vinay Signh (2014) penggunaan algoritma *Fisher Yates Shuffle* yang *modern* oleh Richard Durstenfeld dapat mengurangi kompleksitas algoritma menjadi $O(n)$, dibandingkan dengan mengacak menggunakan metode yang lain seperti menggunakan sorting yang sangat tidak efisien karena adanya *loop* bersarang.

Algoritma *Fisher Yates Shuffle* dipilih karena algoritma ini merupakan metode pengacakan yang lebih baik atau dapat dikatakan sesuai untuk pengacakan angka, dengan waktu eksekusi yang cepat serta tidak memerlukan waktu yang lama untuk melakukan suatu pengacakan.

Pengacakan suatu hal yang sangat penting dalam pembuatan banyak aplikasi. Meskipun terlihat mudah namun pada dasarnya jika tidak dilakukan dengan baik maka pengacakan itu dapat berdampak buruk untuk suatu aplikasi. Dalam hal ini pengacakan menggunakan algoritma *Fisher Yates Shuffle* dapat dijadikan referensi untuk diterapkan dalam sebuah aplikasi yang menggunakan metode pengacakan. (Bendersky dan Eli, 2010)

2.4 Algoritma *Forward Chaining*

Forward Chaining adalah suatu rantai yang dicari atau dilewati dari suatu permasalahan untuk memperoleh solusinya disebut dengan *Forward Chaining*. Cara lain menggambarkan *Forward Chaining* adalah dengan penalaran dari fakta menu konklusi yang terdapat dari fakta. Didalam mengambil keputusan dari fakta-

fakta yang telah diketahui dengan menggunakan inferensi, yang dimana terdapat beberapa kaidah inferensi yaitu:

1. Modus Ponens (*Law Of Detachment*)

Kaidah ini didasari pada tautologi $(p \wedge (p \rightarrow q)) \rightarrow q$, yang dalam hal ini, p dan $p \rightarrow q$ adalah hipotesis, sedangkan q adalah konklusi. Kaidah ini dapat juga ditulis dengan cara :

$$\begin{array}{r} p \rightarrow q \\ p \\ \hline \end{array}$$

Jadi konklusi : q

2. Modus Tollens

Kaidah ini didasari pada tautologi $[\sim q \wedge (p \rightarrow q)] \rightarrow \sim p$. Kaidah ini dapat juga ditulis dengan cara :

$$\begin{array}{r} p \rightarrow q \\ \sim q \\ \hline \end{array}$$

Jadi konklusi : $\sim p$

3. Silogisme Hipotesis

Kaidah ini didasari pada tautologi $[(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$.

Kaidah ini dapat juga ditulis dengan cara :

$$\begin{array}{r} p \rightarrow q \\ q \rightarrow r \\ \hline \end{array}$$

Jadi konklusi : $p \rightarrow r$

4. Silogisme Disjungtif

Kaidah ini didasari pada tautologi $[(p \wedge q) \wedge \sim p] \rightarrow q$. Kaidah ini dapat juga ditulis dengan cara :

$$p \vee q$$

$$\frac{q \rightarrow r}{\quad}$$

Jadi konklusi : $p \rightarrow r$

5. Simplifikasi

Kaidah ini didasarkan pada tautologi $(p \wedge q) \rightarrow p$. Kaidah ini dapat juga ditulis dengan cara :

$$p \wedge q$$

Jadi konklusi : p

6. Penjumlahan

Kaidah ini didasarkan pada tautologi $p \rightarrow (p \vee q)$. Kaidah ini dapat juga ditulis dengan cara :

$$\frac{p}{\quad}$$

Jadi konklusi : $p \vee r$

7. Konjungsi

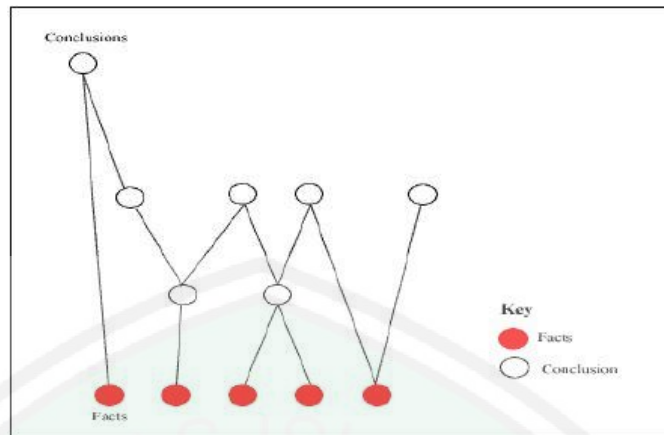
Kaidah ini didasarkan tautologi $((p) \wedge (q)) \rightarrow (p \wedge q)$. Kaidah ini dapat juga ditulis dengan cara :

$$p$$

$$q$$

Jadi konklusi : $p \wedge q$

Forward Chaining melakukan pencarian dari suatu masalah kepada solusinya (seperti diilustrasikan pada **Gambar 2.14**). Jika klausa premis sesuai dengan situasi, maka proses akan memberikan kesimpulan. (Sri Kusumadewi, 2003)



Gambar 2.14 Gambaran Kerja *Forward Chaining / Bottom Up Reasoning*

Kelebihan metode *Forward Chaining* adalah data baru dapat dimasukkan ke dalam *table database* inferensi dan kemungkinan untuk melakukan inferensi *rules*, sehingga dalam implementasinya. *Forward Chaining* sangat membantu *developer* aplikasi dalam membangun sistem. Jika *developer* ingin menambah beberapa kondisi dan aturan, *developer* tidak perlu membongkar lagi kode program dari awal. Metode ini juga mampu menyediakan banyak sekali informasi dari hanya jumlah kecil data. Dan kelebihan utama dari *Forward Chaining* yaitu metode ini akan bekerja dengan baik ketika problem bermula dari mengumpulkan atau menyatukan informasi lalu kemudian mencari kesimpulan apa yang dapat diambil dari informasi tersebut. **Tabel 2.3** dibawah adalah karakteristik *Forward Chaining*.

Tabel 2.3 Karakter *Forward Chaining*

| <i>Forward Chaining</i> |
|---|
| Perencanaan, Monitoring, Control Disajikan untuk masa depan Data memandu, penalaran dari bawah ke atas Bekerja kedepan untuk mendapatkan solusi apa yang mengikuti fakta <i>Breadth first search</i> dimudahkan |

Metode *Forward Chaining* disebut juga data driven karena mesin inferensi menggunakan informasi yang ditentukan oleh user untuk memindahkan keseluruhan jaringan dari logika 'AND' dan 'OR' sampai sebuah terminal ditentukan sebagai obyek. Bila mesin inferensi tidak dapat menentukan obyek maka mesin inferensi akan meminta informasi lain.

Langkah-langkah yang harus dilakukan dalam membuat sistem *Forward Chaining* berbasis aturan, yaitu :

a. Pendefinisian Masalah

Tahap ini meliputi pemilihan domain masalah dan akuisisi pengetahuan.

b. Pendefinisian Data Input

Sistem *Forward Chaining* memerlukan data awal untuk memulai inferensi.

c. Pendefinisian Struktur Pengendalian Data

Aplikasi yang kompleks memerlukan premis tambahan untuk membantu mengendalikan.

2.5 *Game Engine*

2.5.1 *Unity*

Unity merupakan salah satu *game engine* paling terkemuka dewasa ini. Unity adalah *software development* yang terintegrasi untuk menciptakan video

game atau konten lainnya seperti visualisasi arsitektur atau *real-time* animasi baik bernuansa 2D maupun 3D. Unity dapat digunakan pada Microsoft Windows dan Mac Os X. Permainan yang dihasilkan dapat dijalankan secara *multiplatform*. Unity juga dapat menghasilkan permainan untuk *browser* dengan menggunakan Plugin Unity Web Player.



Gambar 2.15 Logo Unity

a. Sejarah Unity 3D dan Perkembangannya

Unity 3D adalah salah satu *software* yang bagus untuk mengembangkan *game* 3D dan selain itu juga merupakan *software* atau aplikasi yang interaktif atau dapat juga digunakan untuk membuat animasi 3 dimensi. Unity lebih tepat dijelaskan sebagai salah satu *software* untuk mengembangkan *video game* atau disebut juga *game engine*, yang sebanding dengan *game engine* yang lain contohnya saja : Director dan Torque *Game engine*. Unity sebanding dengan mereka (Director dan Torque) dikarenakan mereka sama-sama menggunakan grafis yang digunakan untuk pengembangan aplikasi 3D.

Dalam beberapa tahun perkembangannya sebelum dirilis, Unity telah diluncurkan pertama kali sebagai versi pra-rilis dengan

gooball sebuah *video game* yang didesain khusus untuk Apple Macintosh. Gooball dengan Unity pra-rilis telah diluncurkan atau diumumkan pada bulan Maret 2005, sementara itu Unity diluncurkan secara resmi sebagai aplikasi yang bersifat komersial pada dua bulan setelahnya yaitu bulan Juni 2005. Satu tahun kemudian yaitu tahun 2006, aplikasi pengembang *game* ini telah menjadi nominasi untuk Apple Design Awards dalam kategori “Best Os x Graphics”. Unity juga disebut sebagai aplikasi pengembang *multiplatform*, yang mana artinya Unity mendukung untuk mengembangkan aplikasi *game* dan aplikasi lain untuk beberapa Platforms seperti *game console*, Mobile Phone Platforms, Windows dan Os x.

Sejak Unity resmi dirilis sebagai Unity versi 1.0.1 banyak pembaruan (*update*), *upgrades* dan fitur yang telah ditambahkan selama tahun perilisannya tersebut dan Unity terus berkembang secara terus-menerus. Sekarang ini Unity sudah ada pada versi 3.0 yang telah diumumkan pada bulan Maret 2010.

Unity datang dengan beberapa pilihan perijinan (*license*) berkisar dari yang gratis untuk Unity paket utama (dasar) untuk Unity pro, untuk satu *license* dijual dengan harga \$1200 baik Unity yang versi gratis dan Unity yang versi pro menawarkan banyak fitur yang dapat digunakan, masih untuk yang versi gratis memperlihatkan sebuah halaman *splash* pada *game* yang tetap (*regular*) dan desain untuk *game online* menggunakan Unity watermark.

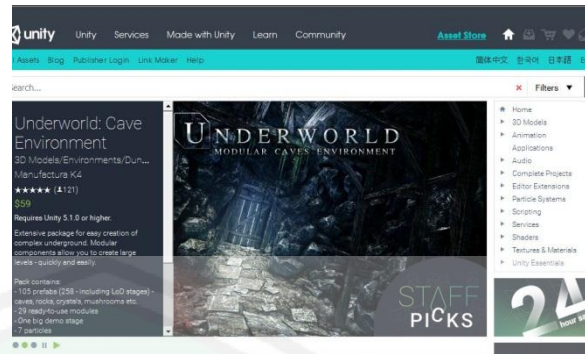
Lebih dari beberapa tahun banyak *game* yang dikembangkan dan dibuat berjalan pada Unity. Beberapa lebih ketetapan berada dalam satu bungkus atau masukkan butuh kecepatan dunia, yang mana sekarang ini dalam perkembangan dan waktunya rilis berikutnya pada tahun ini, *wolfquest* yang mana rilis pada tahun 2007, *Tiger Woods pga tour online*, yang mana telah dibuat pada April pada tahun 2007 dan *Atmosphir* yang mana banyak *game* baru yang dapat berjalan di Unity.

b. Fitur-Fitur Unity

1) Asset store

Asset merupakan aspek dari permainan yang akan direferensikan oleh beberapa komponen, asset itu sendiri, atau kelengkapan penunjang pembuat *game*. *Asset Store* merupakan tempat untuk mendapatkan *asset* yang digunakan untuk menunjang pembuatan *game*.

Asset yang ada pada Unity 3D dibagi menjadi dua, yaitu *eksternal* dan *internal*. *Asset eksternal* merupakan *asset* yang ditambahkan dari sumber di luar Unity 3D, seperti 3D Model, *Texture*, *Sound Effect* dan *Texture*. *Asset internal* merupakan *asset* yang sudah ada di dalam Unity, seperti *Materials*, *Shaders*, *Cube maps*, *Physics materials*, dan *Prefabs*.



Gambar 2.16 Asset Store

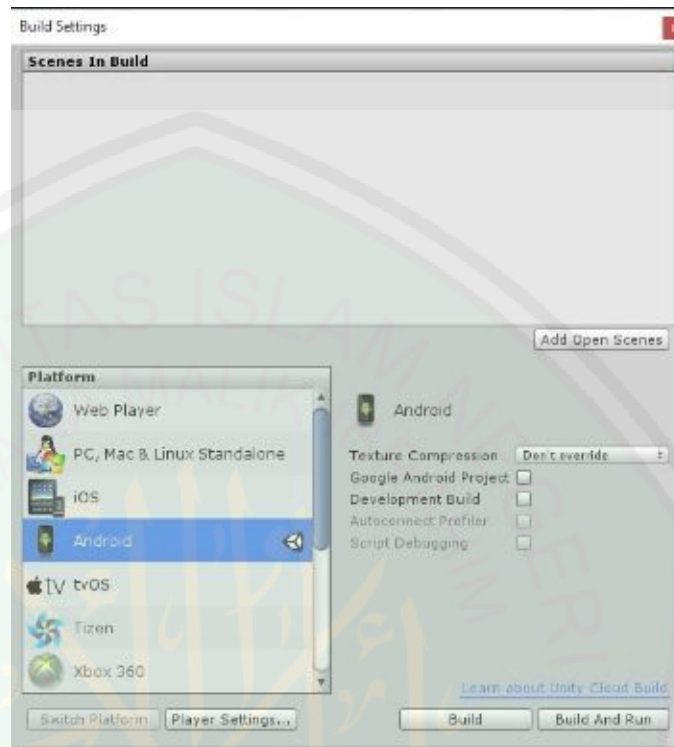
2) Asset Tracking

Unity juga menyertakan *server Unity Asset*—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*), *video playback* menggunakan *theora codec*, *engine* daratan dan vegetasi (dimana mensupport *tree billboard*, *occlusion culling* dengan *umbra*), *built – in lightmapping* dan *global illumination* dengan *beast*, *multiplayer networking* menggunakan *raknet*, dan navigasi *mesh pencari jalur built-in*.

3) Platforms

Seperti yang dibahas sebelumnya, *Unity 3D* dapat dijalankan secara *cross platform*. *Platform* yang didukung antara lain *xbox one*, *blackberry 10*, *Windows 8*, *Windows Phone 8*, *Windows*, *Mac*, *Linux*, *Android*, *Ios*, *Unity Web Player*, *Adobe Flash*, *Playstation 3*, *Xbox 360*, *wii u*, dan *wii*. Selain itu, *Unity 3D* juga mendukung *playstation vita*, meskipun belum ada

informasi resmi. Rencananya, Unity 3D versi berikutnya akan mendukung playstation 4.



Gambar 2.17 Platform yang didukung Unity

4) *Physics*

Unity juga memiliki support built-in untuk 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.

5) *Rendering*

Unity telah mendukung penggunaan *graphic engine*, seperti Direct3D (Windows, Xbox 360), OpenGL (Mac, Windows, Linux, PS3), OpenGL ES (Android, Ios), dan APIs (wii). Selain itu, Unity 3D juga mendukung penggunaan Bump

Mapping, Reflection Mapping, Parallax Mapping, Screen Space Ambient Occlusion (SSAO), Dynamic Shadows menggunakan Shadow Maps, Render-To-Texture dan Full-Screen Post-Processing Effects.

Untuk meningkatkan kualitas pemetaan atau tokoh dalam *game*, Unity 3D mendukung penggunaan software pengolahan gambar lain, seperti 3ds Max, Maya, Softimage, Blender, Modo, Zbrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks, dan Allegorithmic Substance.

6) *Scripting*

Bahasa pemrograman merupakan hal yang umum ditemui dalam pembuatan suatu *game*. Melalui bahasa pemrograman, anda dapat memberikan kecerdasan buatan (*Artificial Intelligence*) pada *game*. Perbedaan *scripting* pada *game engine* dengan membuat *game* menggunakan bahasa pemrograman langsung adalah, pada *game engine* anda tidak perlu membuat *game* dari nol. Alasannya karena *game engine* sudah menyediakan bahan-bahan dasar pembuatan *game*, seperti karakter, peta, tleset, dan lain sebagainya.

Scripting yang digunakan pada Unity dibangun menggunakan Mono 2.6. Mono 2.6 merupakan implementasi open source dari .net Framework. Bahasa pemrograman yang didukung oleh Unity 3D, antara lain Javascript, c#, dan boo

(menggunakan sintaks phyton). Mulai dari Unity 3D versi 3.0, digunakan Mono develop untuk Debugging Script.

```

11 function Update ()
12 {
13     if(Input.GetKeyDown(KeyCode.RightArrow))
14         transform.Rotate(Vector3(0.0, 10.0, 0.0));
15
16     if(Input.GetKeyDown(KeyCode.LeftArrow))
17         transform.Rotate(Vector3(0.0, -10.0, 0.0));
18 }

```

Gambar 2. 18 Penggunaan Script pada Unity

c. Bahasa Pemrograman C#

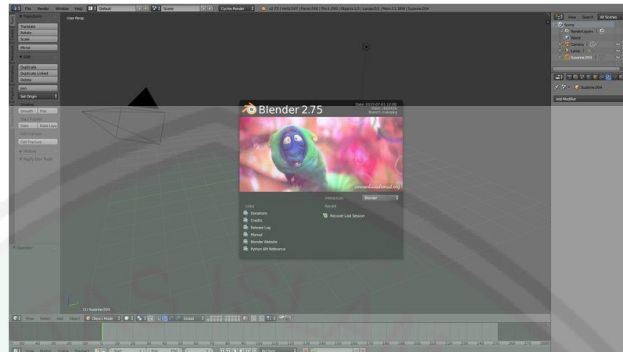
C# (disebut juga sebagai “c sharp”) adalah sebuah bahasa pemrograman *object-oriented* seperti java dan merupakan bahasa *component-oriented* yang pertama. Bahasa pemrograman ini dibuat berdasarkan bahasa C++ yang telah dipengaruhi oleh aspek–aspek ataupun fitur bahasa yang terdapat pada bahasa–bahasa pemrograman lainnya seperti Java, Visual Basic, dan lain–lain dengan beberapa penyederhanaan. Kelebihan dari C# yaitu:

1. Kode pada C# lebih sederhana dan lebih *modern*.
2. Kombinasi fitur–fitur terbaik dari beberapa bahasa pemrograman.
3. Memiliki sintaks yang konsisten.

2.6 Blender

Blender adalah perangkat lunak sumber terbuka grafika komputer 3D. Perangkat lunak ini digunakan untuk membuat film animasi, efek visual, model cetak 3D, aplikasi 3D interaktif dan permainan video. Blender memiliki fitur termasuk pemodelan 3D, penteksturan, penyunting gambar bitmap, penulangan,

simulasi cairan dan asap, simulasi partikel, animasi, penyunting video, pemahat digital, dan rendering.



Gambar 2.19 Tampilan Blender 2.75

2.7 Hasil Penelitian Yang Relevan

Pada tahun 2012, ade-ibijola dan abejide olu, merancang aplikasi a *Simulated Enhancement of Fisher Yates Algorithm for Shuffling in Virtual Card Game Using Domain Specific Data Structures*. Aplikasi ini membuat suatu aplikasi untuk simulasi permainan kartu secara virtual dengan algoritma *Fisher Yates* dan *Domain specific data structures*. Dari 54 buah kartu yang ditarik ke tempat secara acak. Pengacakan kartu dapat berjalan sesuai alur. Aplikasi ini diterapkan dalam *desktop* dengan desain dan kecepatan yang dapat disesuaikan.

Pada penelitian yang berjudul “Aplikasi sistem pakar untuk mendiagnosa penyakit THT menggunakan teknik *Forward Chaining*”. Pada penelitian tersebut menggunakan algoritma *Forward Chaining* untuk mendiagnosa gejala-gejala penyakit THT. Dengan menggunakan algoritma *Forward Chaining* dapat disimpulkan bahwa aplikasi sistem pakar ini dapat mempercepat proses diagnosa penyakit, dapat membantu pekerjaan dokter, memberikan informasi yang dibutuhkan, dan dapat memberikan hasil yang sesuai dengan diagnosa yang dilakukan dokter. Berdasarkan hasil wawancara pada sebuah puskesmas atau

rumah sakit, aplikasi ini dapat membantu perawat atau paramedis untuk memberikan diagnosa awal dan penanganan sementara untuk pemeriksaan di POSYANDU ketika dokter tidak bisa hadir (Siagia, Luckman.2009)

Pada tahun 2014, supriyanto, dkk (supriyanto et al, 2014) membuat aplikasi *edugame guess calculation* berbasis android menggunakan algoritma *Fisher Yates* untuk pengacakan puzzle. *Game* ini berlatar belakang untuk membantu minat anak-anak dalam pembelajaran matematika dengan perhitungan logika sederhana, yang meliputi penambahan, pengurangan, perkalian, dan pembagian dimana perhitungan yang telah dilakukan akan ditempatkan pada beberapa kotak tersusun yang disebut puzzle. Pengujian yang dilakukan terhadap *game* ini yaitu dengan melalui kuesioner, aplikasi ini dapat membantu anak-anak dalam pembelajaran matematika dengan presentase tingkat kepuasan pemakainya mencapai 98%.

Pada tahun yang sama, Ryan Nugraha, dkk (Ryan et al, 2014) membuat aplikasi *The Lost Insect* untuk pengenalan jenis serangga berbasis Unity 3D menggunakan Algoritma *Fisher Yates Shuffle* untuk pengacakan soal. *Game* ini berlatar belakang sebagai alternatif untuk pengguna mengenal pembelajaran dan pemberitaan informasi tentang serangga menjadi sumber pengetahuan yang langsung dapat diakses dan digunakan. Manfaat dari penggunaan *game* ini adalah dapat mengenal jenis-jenis serangga yang ada di lingkungan sekitar.

Tahun 2017 Teguh Arifianto, dkk (Teguh et al, 2017) membuat *Game* Balap Karung menggunakan metode *Forward Chaining* khususnya untuk menentukan penghargaan yang akan diberikan untuk player. *Game* ini merupakan *game* yang bertipe *levelling*, dimana kekuatan *game* terletak pada tingkat kesulitan yang berbeda pada tiap *level*. *Game* Balap Karung berjalan pada

platform dekstop (windows). Dan permainan ini hanya dapat dimainkan oleh seorang *user*. Hasil dari *game* ini menunjukkan nilai validitas sebesar 100%.

Dari penelitian tersebut, penulis mengambil referensi bagaimana mengimplementasikan metode *Fisher Yates Shuffle* untuk pengacakan soal pertanyaan dan metode *Forward Chaining* untuk menentukan kenaikan *level* pada *game Mr Garbage*.



BAB III

PERANCANGAN GAME

3.1 Analisis dan Perancangan *Game*

3.1.1 Deskripsi *Game*

Game ini adalah *game* yang bergenre edukasi *game* dan dimainkan secara *single player* dan *game* ini dikhususkan untuk anak Sekolah Dasar umur 8-11 tahun. Pada *game* ini terdapat karakter *Mr Garbage* sebagai karakter utama yang akan dijalankan oleh pengguna. Karakter utama (*Mr Garbage*) harus mengambil sampah yang berserakan dan membuangnya ke tempat sampah. Karakter musuh yaitu kucing pengganggu merupakan karakter lawan yang akan dijalankan secara otomatis oleh komputer. Di dalam permainan ini terdapat 5 *level* yang berbeda dengan karakteristik dan tingkat kesulitan yang berbeda pula. Dan di dalam permainan ini juga terdapat pertanyaan tentang kebersihan lingkungan agar pemain dapat belajar sambil bermain *game*.


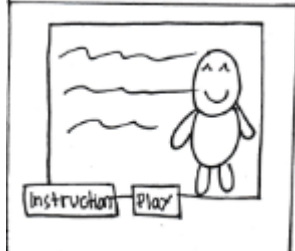
3.1.2 Keterangan Umum *Game*

Mr Garbage game merupakan permainan edukasi berbasis *desktop* yang dijadikan media untuk pengenalan dan kesadaran terhadap kebersihan lingkungan. *Game Mr Garbage* di dalamnya terdapat karakter *Mr Garbage* yang menjadi karakter utama. *Game* ini terdiri dari 5 *level* dengan tingkat kesulitan yang berbeda sesuai dengan ketentuan *stage game*. Pada *game* ini akan ada banyak sampah yang berserakan dimana-mana. Karakter utama (*Mr Garbage*) harus membuang sampah pada tempatnya. Jika semua sampah berhasil dimasukkan ke tempat sampah maka akan lanjut ke *level 2*. Dan pada *level 2* nanti *Mr Garbage*

juga harus memasukkan sampah ke tempat sampah tapi harus sesuai dengan jenis kategorinya. Pada *level 3 Mr Garbage* masih harus membersihkan sampah yang berserakan dan harus dimasukkan ke tempat sampah sesuai dengan jenis kategori sampah masing-masing. Setelah menyelesaikan *level 3* maka dapat lanjut ke *level 4*. Didalam *level 4 Mr Garbage* masih harus membersihkan sampah yang berserakan dan harus dimasukkan ke tempat sampah sesuai dengan jenis kategori sampah masing-masing. Pada *level 4* juga *player* harus menjawab pertanyaan yang ada dan akan menghadapi musuh yaitu kucing pengganggu. Sedangkan pada *level 5* juga terdapat karakter musuh (kucing pengganggu) yang akan menghalangi *player* mencapai *finish* karena jika *player* menabrak kucing pengganggu maka *player* akan mati. Disini juga *player* harus menjawab pertanyaan yang ada tetapi dengan tingkat kesulitan soal yang lebih sulit.

3.1.3 Storyboard

Tabel 3.1 Storyboard

| Scene | Board | Audio | Naskah |
|-------|---|-------------------------------------|---|
| 1 |  | Backsound musik "the happy mood" | Pada tampilan awal ini muncul <i>splash screen game Mr Garbage</i> |
| 2 |  | Backsound musik "the happy mood" | Animasi pembuka, pada <i>scene</i> ini terdapat dua menu yaitu <i>instructions</i> dan menu <i>play</i> |

3



Backsound musik
“the happy mood”

Jika kita klik menu *instructions* maka kita akan menuju halaman *instructions* yang berisi tentang tata cara bermain *game Mr Garbage*

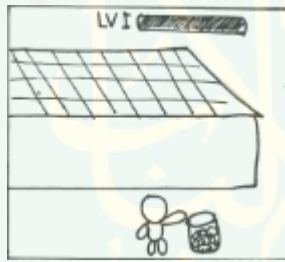
4



Backsound musik
“bike_rides”

Mr Garbage berada dihalaman rumah dan melihat banyak sekali sampah berserakan

5



Backsound musik
“bike_rides”

Mr Garbage membersihkan sampah dan memasukkannya ke tempat sampah

6



Backsound musik
“bike_rides”

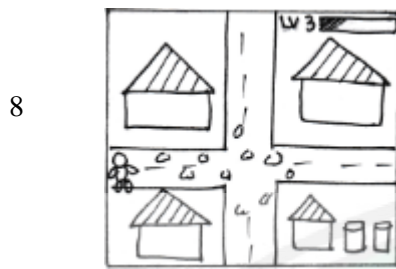
Mr Garbage berada ditengah – tengah perkotaan dan melihat banyak sekali sampah berserakan

7



Backsound musik
“bike_rides”

Mr Garbage harus memasukkan sampah ke tempat sampah sesuai kategori jenis sampahnya



Backsound musik
“bike_rides”

Mr Garbage berada di kompleks perumahan yang dipenuhi dengan sampah



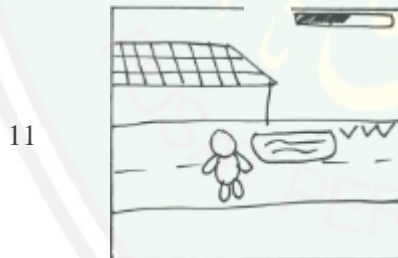
Backsound musik
“bike_rides”

Mr Garbage harus membersihkan sampah yang ada di lingkungan kompleks dan memasukkannya sesuai kategori jenis sampahnya



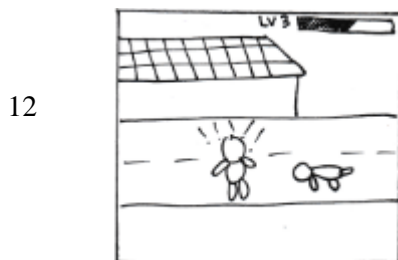
Backsound musik
“bike_rides”

Disisi lain membersihkan sampah *Mr Garbage* juga harus menemukan sapu ajaib yang berisi pertanyaan




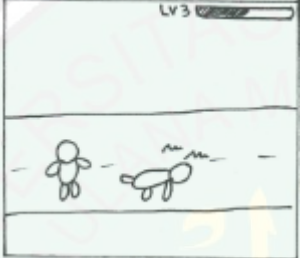



Backsound musik
“bike_rides”

Jika jawaban *Mr Garbage* benar maka akan mendapat item senjata untuk mengusir musuh “kucing pengganggu”



Backsound musik
“bike_rides & meow”

Ketika membersihkan sampah *Mr Garbage* juga akan di ganggu oleh kucing

- 13  *Backsound musik* “bike_rides & meow” *Mr Garbage* harus menghindari kucing pengganggu
- 14  *Backsound musik* “bike_rides & meow 2” Jika *Mr Garbage* mendapatkan item senjata maka bisa digunakan untuk mengusir musuh
- 15  *Backsound musik* “bike_rides” *Mr Garbage* berhasil menyelesaikan semua misi
- 16  *Backsound musik* “boxcat_games_25_victory” Tampilan jika *Mr Garbage* berhasil menyelesaikan misi
- 17  *Backsound musik* “boxcat_games_25_victory” Tampilan jika *Mr Garbage* kalah

3.1.4 Deskripsi Karakter

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

1. Karakter Utama

Karakter ini memerankan sebagai *Mr Garbage* yang diberi misi untuk membersihkan lingkungan dari sampah. Untuk menyelesaikan misi tersebut, pemain harus membersihkan semua sampah dan membuangnya pada tempat sampah jika sudah selesai maka dapat pindah ke *scene* atau *level* selanjutnya. Karakter *Mr Garbage* dapat dilihat pada **Gambar 3.1**.



Gambar 3.1 Mr Garbage

2. Karakter Musuh Pengganggu (NPC)

Karakter musuh dalam *game* ini yaitu kucing pengganggu, pemain harus menghindari karakter ini karena dapat merugikan pemain yaitu dapat membuat pemain mati. Karakter musuh dapat dilihat pada **Gambar 3.2**.



Gambar 3.2 Karakter Enemy

3.1.5 Deskripsi Item

Berikut akan dijelaskan berbagai komponen item yang berada dalam *game*:

a. Item Senjata

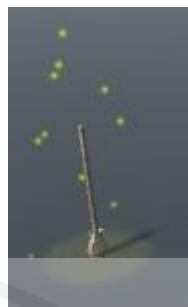
Jika pemain benar menjawab pertanyaan maka pemain akan mendapat hadiah senjata sapu. Senjata ini dapat berfungsi untuk mengusir musuh pengganggu. Desain item senjata dapat dilihat pada **Gambar 3.3**.



Gambar 3.3 Item Senjata

b. Item Sapu ajaib

Ketika pemain membersihkan sampah pemain juga harus menemukan item sapu ajaib. Sapu ajaib ajaib berisi pertanyaan jika pemain benar menjawab pertanyaan maka pemain akan mendapat hadiah item senjata. Desain item sapu ajaib dapat dilihat pada **Gambar 3.4**.



Gambar 3.4 Item Sapu Ajaib

c. Item Tempat Sampah

Ketika pemain mendapatkan sampah maka pemain harus memasukkannya atau membuangnya ke tempat sampah yang telah disediakan di setiap *scene game*. Pada *game* ini tempat sampah dikategorikan dalam dua jenis, yaitu: tempat sampah organik dan tempat sampah anorganik. Desain item tempat sampah dapat dilihat pada **Gambar 3.5**.



Gambar 3.5 Tempat Sampah

3.1.6 Macam-macam Sampah

Apabila *player* menyentuh jenis sampah ini dan membuangnya ke tempat sampah maka *player* akan mendapatkan *score* +10 dan seterusnya sampai point pengalaman penuh. Rancangan sampah dapat dilihat pada **Gambar 3.6, Gambar 3.7, Gambar 3.8, dan Gambar 3.9**.



Gambar 3.6 Sampah botol



Gambar 3.7 Sampah kaleng



Gambar 3.8 Sampah daun kering

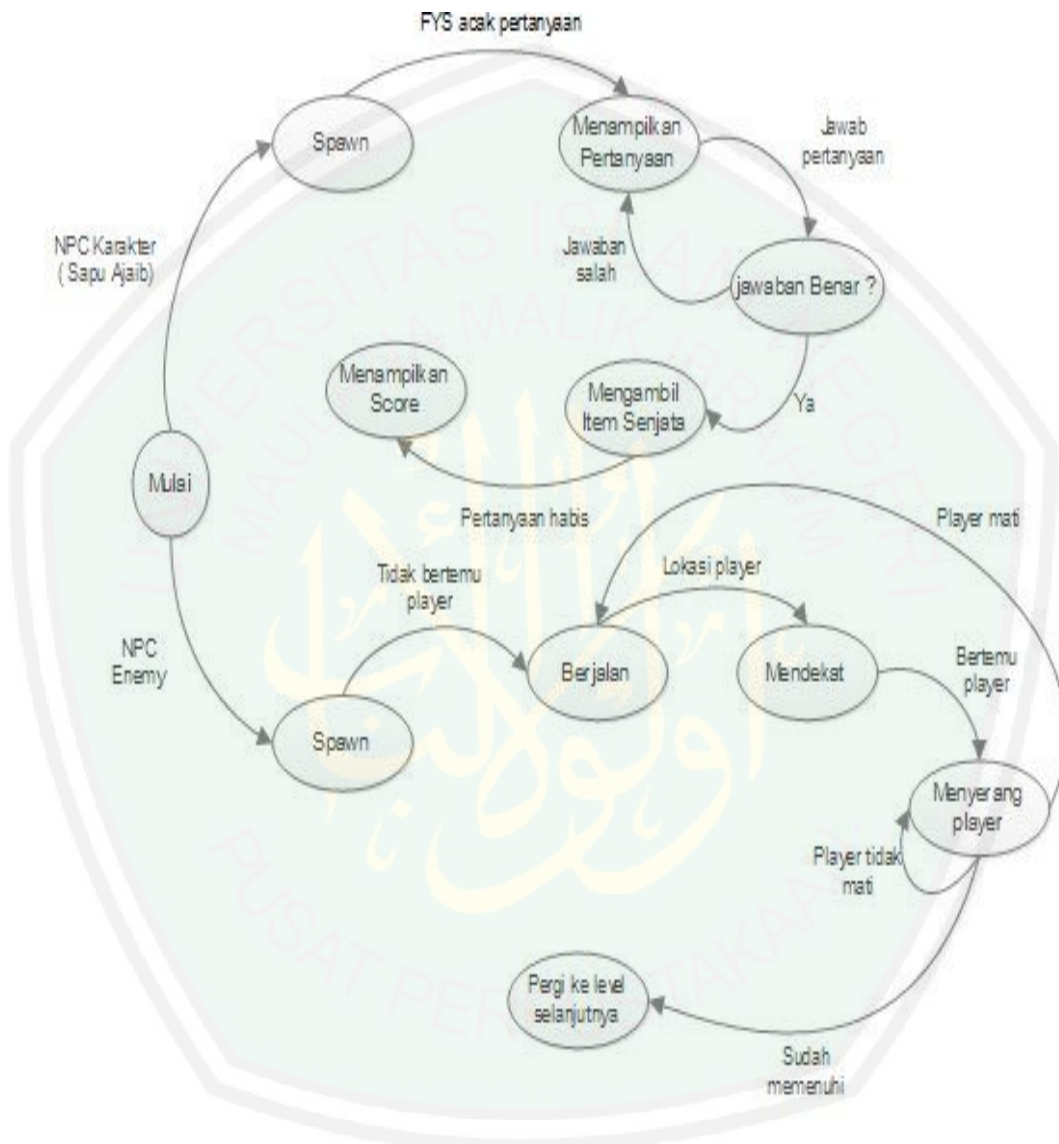


Gambar 3.9 Sampah kertas

3.2 *Finite State Machine*

Implementasi *Finite State Machine* di *game* ini adalah untuk mengatur perilaku NPC. Sedangkan perilaku karakter utama mengikuti perintah orang yang bermain *game*. Pada permainan ini terdapat dua jenis NPC, yaitu NPC *Enemy* dan

NPC Karakter (Sapu Ajaib). Pergerakan NPC otomatis sesuai dengan yang diprogramkan, FSM berikut ini menggambarkan perilaku NPC *Enemy* dan NPC Karakter (Sapu Ajaib).



Gambar 3.10 *Finite State Machine*

3.3 Levelling

Dalam *game* ini terdapat 5 *level* yang setiap *level*nya mempunyai rintangan yang berbeda pada setiap *level*nya :

Tabel 3. 2 Ketentuan Level

| No | Level | Tingkat kesulitan |
|----|---------|-------------------|
| 1 | Level 1 | Mudah |
| 2 | Level 2 | Sedang |
| 3 | Level 3 | Sedang |
| 4 | Level 4 | Sulit |
| 5 | Level 5 | Sulit |

3.4 Scoring

Sistem *scoring* pada *game* edukasi *Mr Garbage* ini berdasarkan ketepatan dalam membuang sampah secara benar. Apabila pemain berhasil membuang sampah dengan benar maka dengan otomatis *score* akan bertambah 10 dan seterusnya.

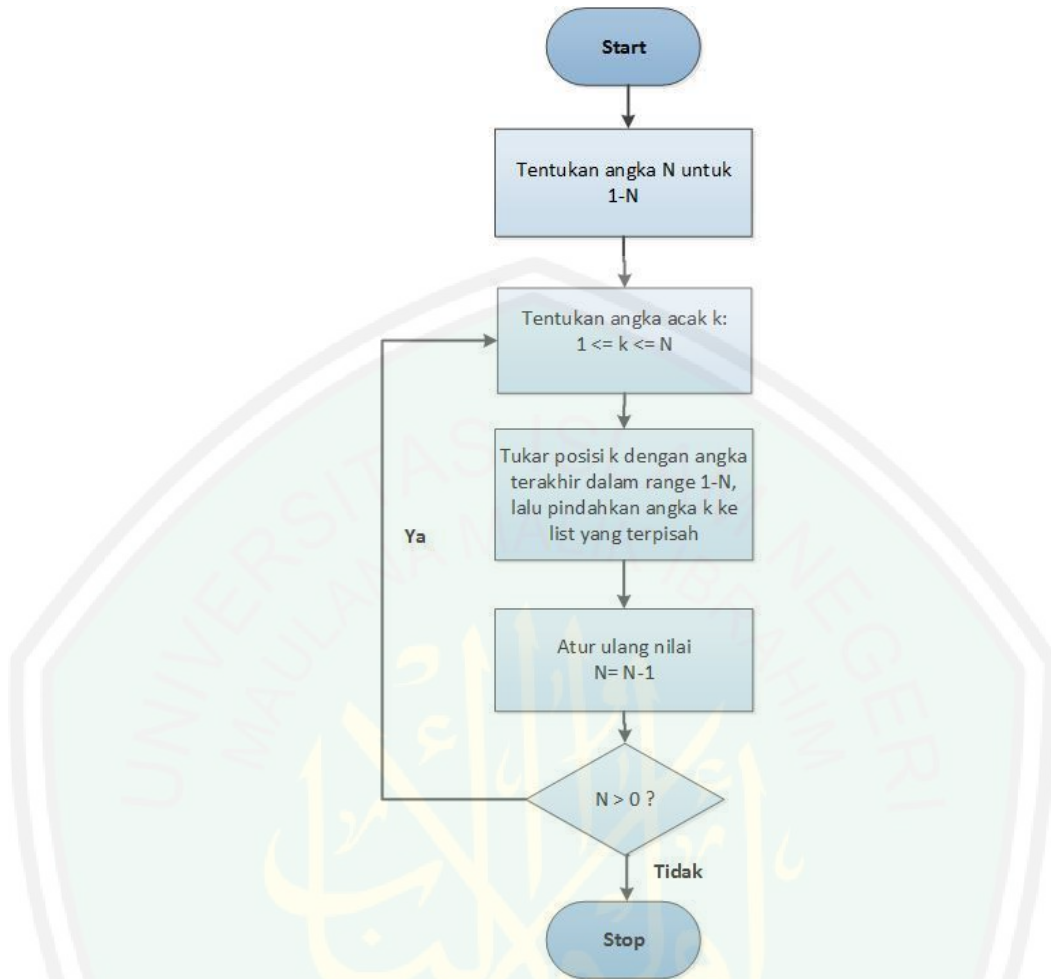
3.5 Pertanyaan

Pertanyaan yang diberikan dikhususkan untuk anak Sekolah Dasar umur 8-11 tahun dengan tema yang diberikan yaitu kebersihan lingkungan. Pertanyaan akan diacak menggunakan metode *Fisher Yates Shuffle*, sehingga urutan pertanyaan yang akan muncul tidak mudah ditebak dan tidak muncul secara berulang-ulang.

3.6 Metode Algoritma *Fisher Yates Shuffle* Pada Pengacakan Pertanyaan

Penerapan algoritma *Fisher Yates Shuffle* terdapat pada pertanyaan yang berada di dalam item sapu. *Fisher Yates Shuffle* diterapkan untuk mengacak pertanyaan di setiap item sapu yang ditemukan.

- A. Flowchart algoritma *Fisher Yates Shuffle* dapat dilihat pada Gambar 3.11 dibawah ini.**



Gambar 3.11 Flowchart Algoritma *Fisher Yates Shuffle*

Berdasarkan **Gambar 3.11** diketahui bahwa algoritma *Fisher Yates Shuffle* dalam mengacak urutan data diawali dengan menentukan angka N untuk *range* $1-N$. Setelah itu tentukan angka acak k dari *range* $1-N$, tidak boleh 0 dan melebihi N . Tukar posisi k dengan angka terakhir *range* $1-N$ ditempat terpisah. Atur ulang N , $N=N-1$. Jika N bernilai 0 , maka hentikan proses pengacakan.

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 pertanyaan menggunakan *Fisher Yates Shuffle*
4. Menampilkan pertanyaan



Gambar 3.12 Blok Diagram Implementasi *Fisher Yates Shuffle*

C. Simulasi Manual Perhitungan *Fisher Yates Shuffle*

Dalam *game Mr Garbage* : dimisalkan terdapat 9 pertanyaan yang akan diacak. Maka didapatkan panjang array $(N)= 9$

Tabel 3.3 Data Array

| Array ke | Pertanyaan |
|----------|--|
| 1 | Sampah yang bersifat mudah membusuk adalah sampah ? |
| 2 | Lingkungan yang sehat membantu kita terhindar dari ? |
| 3 | Plastik merupakan jenis sampah ? |
| 4 | Yang termasuk jenis sampah organik adalah ? |
| 5 | Aktivitas manusia menghasilkan sisa-sisa yang disebut ? |
| 6 | Lalat dapat menjadi media ? |
| 7 | Dibawah ini yang termasuk lingkungan yang tidak sehat adalah ? |
| 8 | Jika masuk kelas agar tidak banyak sampah, sebaiknya ? |
| 9 | Yang tidak termasuk contoh sampah anorganik adalah ? |

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

- a. Ambil satu elemen secara acak (k). Nilai k yang boleh diambil adalah elemen yang belum pernah diambil.
- b. Tukarkan nilai (k) yang diambil dengan elemen terakhir (n) yang belum diambil.
- c. Ulangi selama masih ada elemen yang belum diambil.

Tabel iterasi dari proses manual pengacakan menggunakan Algoritma *Fisher Yates Shuffle* dengan jumlah array $N=9$ dideskripsikan seperti pada **Tabel**

3.4

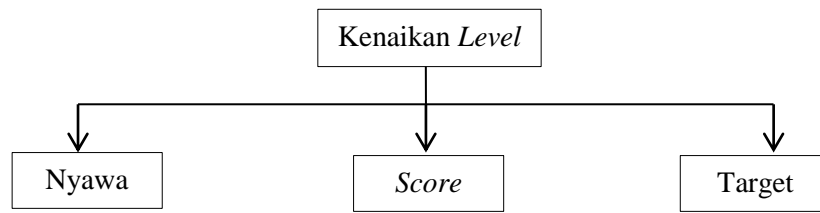
Tabel 3.4 Iterasi Algoritma *Fisher Yates Shuffle*

| <i>Range</i> | <i>Roll</i> | <i>Scratch</i> | <i>Result</i> |
|------------------|-------------|------------------------------------|-------------------------------|
| | | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 | |
| 1-10 | 5 | 1, 2, 3, 4, 10 , 6, 7, 8, 9 | 5 |
| 1-9 | 3 | 1, 2, 9 , 4, 10, 6, 7, 8 | 3, 5 |
| 1-8 | 4 | 1, 2, 9, 8 , 10, 6, 7 | 4, 3, 5 |
| 1-7 | 5 | 1, 2, 9, 8, 7 , 6 | 10, 4, 3, 5 |
| 1-6 | 2 | 1, 6 , 9, 8, 7 | 2, 10, 4, 3, 5 |
| 1-5 | 3 | 1, 6, 7 , 8 | 9, 2, 10, 4, 3, 5 |
| 1-4 | 1 | 8 , 6, 7 | 1, 9, 2, 10, 4, 3, 5 |
| 1-3 | 7 | 8, 6 | 7, 1, 9, 2, 10, 4, 3, 5 |
| 1-2 | 8 | 6 | 8, 7, 1, 9, 2, 10, 4, 3, 5 |
| Hasil Pengacakan | | | 6, 8, 7, 1, 9, 2, 10, 4, 3, 5 |

Pada **Tabel 3.3** pertanyaan yang akan ditampilkan telah diinisialisasikan dalam bentuk *array* secara berurutan. *Array* tersebut diacak menggunakan metode *Fisher Yates Shuffle* pada **Tabel 3.4** sehingga menghasilkan urutan sebagai berikut : 6, 8, 7, 1, 9, 2, 10, 4, 3, 5

3.7 Metode *Forward Chaining* Untuk Menentukan Kenaikan *Level*

Pada *game* ini metode *Forward Chaining* digunakan untuk menentukan kondisi kenaikan *level*. Ada beberapa komponen yang mempengaruhi *level up player* berdasarkan Algoritma *Forward Chaining*, yaitu nyawa, *score*, dan target.



Gambar 3.13 Kriteria kenaikan level

Berikut ini adalah variabel yang digunakan sebagai parameter dalam *Forward Chaining* untuk pengembangan game *Mr Garbage* :

Tabel 3.5 Variabel Aturan Game Mr Garbage

| No | Kode Rule | Kondisi | Keterangan |
|----|-----------|-----------|------------|
| 1 | A01 | [0,1,2,3] | Nyawa |
| 2 | AXX | ... | ... |

Tabel 3.6 Variabel Hasil Game Mr Garbage

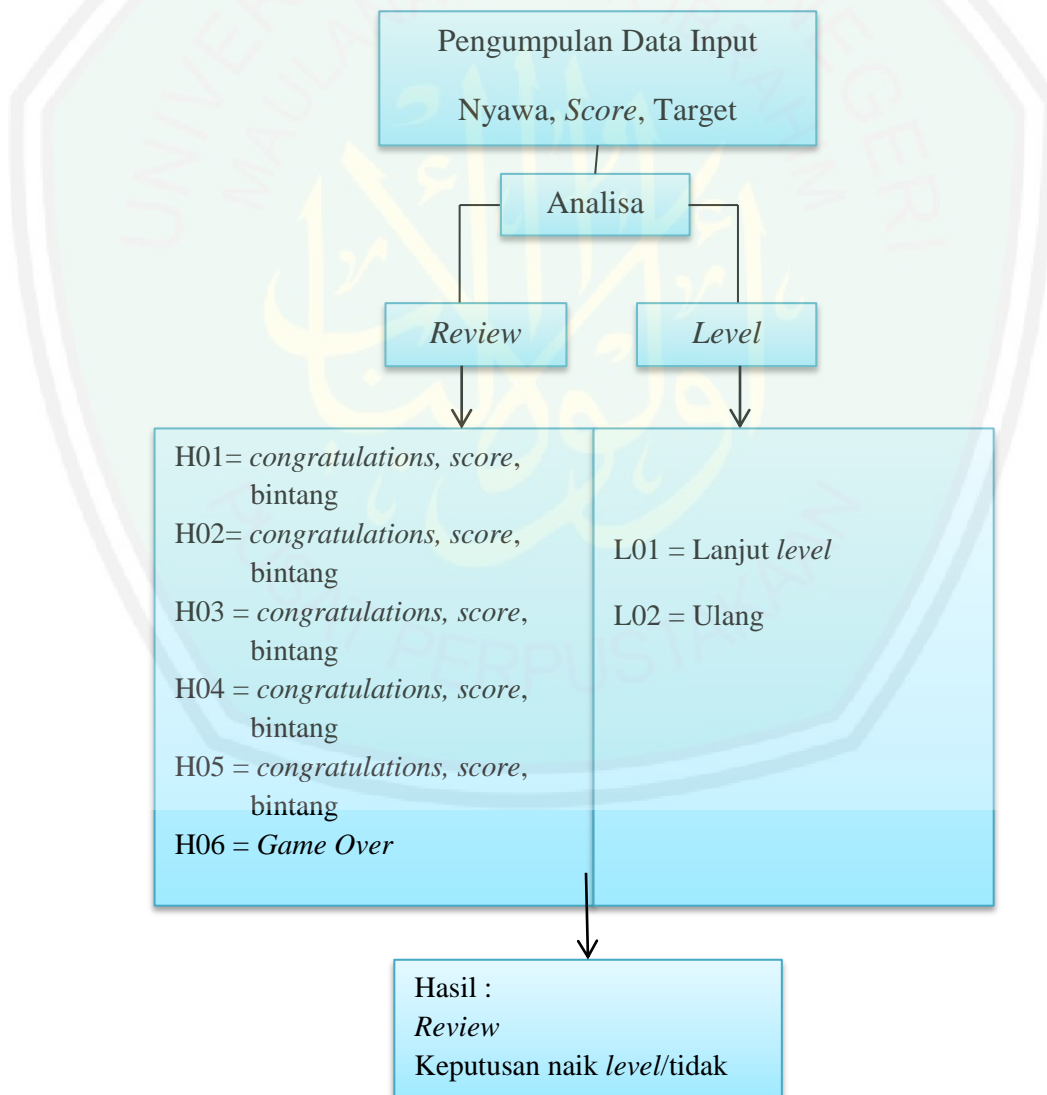
| No | Kode Rule | Kondisi |
|----|-----------|---|
| 1 | H01 | <i>Congratulations, Score,</i> bintang |
| 2 | HXX | ... |

Tabel 3.7 Variabel Level Game Mr Garbage

| No | Kode Rule | Kondisi |
|----|-----------|--------------|
| 1 | L01 | Lanjut Level |
| 2 | LXX | ... |

3.8 Proses Representasi Cara Kerja *Game*

Proses representasi pengetahuan dilakukan dengan cara mengumpulkan kondisi untuk memperoleh hasil. Berdasarkan data *input* yang diperoleh maka dilakukan analisis untuk menentukan *review* apa yang didapat dalam setiap *levelnya* serta layak atau tidaknya seorang *player* untuk naik ke *level* berikutnya dalam *game*. **Gambar 3.14** dibawah ini merupakan proses representasi cara kerja *game*.



Gambar 3.14 Representasi Cara Kerja *Game*

Hasil representasi pengetahuan kondisi aturan disajikan pada **Tabel 3.8**, representasi pengetahuan *review* disajikan pada **Tabel 3.9**, sedangkan representasi pengetahuan kenaikan *level* pada **Tabel 3.10**.

Tabel 3.8 Representasi Kondisi

| No | Kondisi | Kode Rule | Keterangan |
|----|--------------------------------|-----------|------------|
| 1 | [0, 1, 2, 3] | A01 | Nyawa |
| 2 | [>=50, >=60, >=70, >=80, >=90] | A03 | Score |
| 3 | [T,TT] | A04 | Target |

Tabel 3.9 Representasi Review

| No | Kondisi | Kode |
|----|--|------|
| 1 | <i>Congratulations, score, bintang</i> | H01 |
| 2 | <i>Congratulations, score, bintang</i> | H02 |
| 3 | <i>Congratulations, score, bintang</i> | H03 |
| 4 | <i>Congratulations, score, bintang</i> | H04 |
| 5 | <i>Congratulations, score, bintang</i> | H05 |
| 6 | <i>Game Over</i> | H06 |

Tabel 3. 10 Representasi Level

| No | Kode Level | Keterangan |
|----|------------|---------------------|
| 1 | L01 | Lanjut <i>level</i> |
| 2 | L02 | Mengulang |

Tabel 3.11 Aturan *Game Mr Garbage* Berdasarkan *Levelling*

| No | Kode Level | Keterangan |
|----|----------------------------|-------------------------|
| 1 | H01, H02, H03, H04, H05 | L01 Lanjut Level |
| 2 | H06 | L02 Ulang |

Tabel 3.12 Aturan *Game Mr Garbage* Berdasarkan *Review*

| No | IF | THEN |
|----|---------------------------------------|--|
| 1 | A01='3' AND A03='>=50' AND A04='T' | H01 <i>Congratulations, score,</i> Nyawa sisa 3, <i>Score</i> >=50, terpenuhi bintang |
| 2 | A01='3' AND A03='>=60' AND A04='T' | H02 <i>Congratulations, score,</i> Nyawa sisa 3, <i>Score</i> >=60, terpenuhi bintang |
| 3 | A01='3' AND A03='>=70' AND A04='T' | H03 <i>Congratulations, score,</i> Nyawa sisa 3, <i>Score</i> >=70, terpenuhi bintang |
| 4 | A01='3' AND A03='>=80' AND A04='T' | H04 |

Nyawa sisa 3, *Score* >=80, terpenuhi *Congratulations, score,*
bintang

A01='3' AND A03='>=90' AND H05
A04='T'

5
Nyawa sisa 3, *Score* >=90, terpenuhi *Congratulations, score,*
bintang

6 *Game Over/Ulang* H06



BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi

Bab ini membahas mengenai implementasi dari perencanaan yang telah diajukan. Selain itu pada bab ini dilakukan pengujian terhadap *game* untuk mengetahui apakah *game* tersebut telah berjalan sesuai dengan tujuan penelitian yang ingin dicapai. Sebelum diimplementasikan terlebih dahulu dipaparkan spesifikasi sistem perangkat keras (*hardware*) dan perangkat lunak (*software*), adapun keperluan tersebut adalah:

4.1.1 Perangkat Yang Digunakan

Perangkat yang diperlukan dalam pembuatan *game Mr Garbage* ini ada dua perangkat, yaitu perangkat keras dan perangkat lunak. Berikut penjelasan perangkat keras dan perangkat lunak yang digunakan dalam pembuatan *game*.

Perangkat keras yang digunakan untuk pembuatan *game* ini dijelaskan pada **Tabel 4.1** berikut:

Tabel 4.1 Perangkat Keras Yang Digunakan Untuk Pembuatan *Game*

| No | Perangkat Keras | Spesifikasi |
|----|------------------|--|
| 1 | Processor | AMD Dual-core Processor C-60 (1.0 GHz, Cache 1 MB) |
| 2 | RAM | 2 GB DDR3 PC-10600 |
| 3 | HDD | 320 GB Serial ATA 5400 RPM |
| 4 | Monitor | 14" |
| 5 | Speaker | On |
| 6 | Mouse & Keyboard | On |

Perangkat lunak yang diperlukan untuk pembuatan *game* ini seperti pada

Tabel 4.2 sebagai berikut:

Tabel 4.2 Perangkat Lunak Yang Digunakan Untuk Pembuatan *Game*

| No | Perangkat Lunak | Spesifikasi |
|----|------------------|--------------------------|
| 1 | Sistem Operasi | Windows 7 64 Bit |
| 2 | Game Engine | Unity3D 5.5.5p1 (64-bit) |
| 3 | Konsep desain 2D | Photoshop |
| 4 | Desain 3D | Blender 2.7 |
| 5 | Script Writer | Mono Develop |

4.2 Implementasi Sistem

Bab ini menjelaskan tentang implementasi algoritma *Fisher Yates Shuffle* dan algoritma *Forward Chaining* pada *game* yang dibuat. Berikut penjelasan implementasi sistem yang telah dibuat:

4.2.1 Implementasi Algoritma *Fisher Yates Shuffle* Pada Pengacakan

Pertanyaan

Pada bagian ini akan dibahas mengenai implementasi algoritma *Fisher Yates Shuffle* untuk mengacak pertanyaan yang akan muncul. Pada saat *game* dimulai di *level 4 Fisher Yates Shuffle* akan bekerja untuk memunculkan dan mengacak pertanyaan. Berikut ini adalah *source code* untuk mengacak pertanyaan.


```

//Fisher yates
for(int i=0;i<urutan.Length;i++){
    int random = Random.Range (0, urutan.Length - 1);

    int randomElemnt = urutan [random];
    urutan [random] = urutan [i];
    urutan [i] = randomElemnt;
}

//for(int i=0;i<urutan.Length;i++){
Debug.Log (" urutan = "+urutan [0]+"-"+urutan [1]
+"-"+urutan [2]+"-"+urutan [3]+"-"+urutan [4]
+"-"+urutan [5]+"-"+urutan [6]+"-"+urutan [7]
+"-"+urutan [8]+"-"+urutan [9]+"-"+urutan [10]
+"-"+urutan [11]+"-"+urutan [12]+"-"+urutan [13]
+"-"+urutan [14]+"-"+urutan [15]+"-"+urutan [16]
+"-"+urutan [17]+"-"+urutan [18]+"-"+urutan [19]
+"-"+urutan [20]+"-"+urutan [21]+"-"+urutan [22]
+"-"+urutan [23]+"-"+urutan [24]+"-"+urutan [25]
+"-"+urutan [26]+"-"+urutan [27]+"-"+urutan [28]
+"-"+urutan [29]+"-"+urutan [30]+"-"+urutan [31]
+"-"+urutan [32]+"-"+urutan [33]+"-"+urutan [34]
+"-"+urutan [35]+"-"+urutan [36]+"-"+urutan [37]
+"-"+urutan [38]+"-"+urutan [39]+"-"+urutan [40]
+"-"+urutan [41]+"-"+urutan [42]+"-"+urutan [43]
+"-"+urutan [44]+"-"+urutan [45]+"-"+urutan [46]
+"-"+urutan [47]+"-"+urutan [48]+"-"+urutan [49]);
//}
//
}

```

Unsur pendidikan pada *game* ini merupakan pembelajaran lingkungan yang dimana *game* ini akan mengajukan pertanyaan kepada pemain. Dari semua soal pada kumpulan pertanyaan yang digunakan pada *game* ini adalah tipe soal pilihan ganda yang terdapat 50 soal. Pertanyaan ini akan muncul pada *level* 4 dan hanya ada 10 soal yang akan muncul. Berikut ini *source code* untuk batasan soal yang akan muncul:

```

//pembatasan soal yg akan muncul
int pos = 0;
public void StartQuiz(){
    uiQuiz.SetActive (true);
    Cursor.visible = true;
    //posisi urutan
    if (pos == 10) {
        pos = 0;
    }
    int acak = urutan[pos];
    TQuiz.text = question [acak].pertanyaan;
    TpsiA.text = question [acak].opsiA;
    TpsiB.text = question [acak].opsiB;
    TpsiC.text = question [acak].opsiC;
    TpsiD.text = question [acak].opsiD;
    Jbenar = question [acak].benar;
    Time.timeScale = 0;
    //posisi selanjutnya
    pos = pos + 1;
}

```

Berdasarkan *source code* diatas, batasan soal yang akan ditampilkan adalah 10 soal dari 50 soal yang ada. Pada *level 4* semua jumlah soal akan diacak semua tapi yang akan muncul hanya 10 soal. Jika 10 soal sudah muncul semua maka akan kembali ke pertanyaan pertama lagi.

4.2.2 Implementasi Algoritma *Forward Chaining* Pada Kenaikan *Level*

Pada *game Mr Garbage* ini implementasi metode *Forward Chaining* digunakan untuk menentukan kenaikan *level* pemain berdasarkan aturan-aturan atau *rule* yang telah diterapkan dengan tahapan sebagai berikut:

- 1 Mulai *level 1*, jika pada *level 1* pemain bisa mengumpulkan *score* ≥ 50 , nyawa ≥ 0 dan target terpenuhi maka akan lanjut *level* selanjutnya.
- 2 Mulai *level 2*, jika pada *level 2* pemain bisa mengumpulkan *score* ≥ 60 , nyawa ≥ 0 dan target terpenuhi maka akan lanjut *level* selanjutnya.

- 3 Mulai *level 3*, jika pada *level 3* pemain bisa mengumpulkan *score* ≥ 70 , nyawa ≥ 0 dan target terpenuhi maka akan lanjut *level* selanjutnya.
- 4 Mulai *level 4*, jika pada *level 4* pemain bisa mengumpulkan *score* ≥ 80 , nyawa ≥ 0 dan target terpenuhi maka akan lanjut *level* selanjutnya.
- 5 Mulai *level 5*, jika pada *level 5* pemain bisa mengumpulkan *score* ≥ 90 , nyawa ≥ 0 dan target terpenuhi maka pemain berhasil menyelesaikan misi.
- 6 Permainan selesai.
- 7 Jika pemain tidak bisa mengumpulkan *score*, nyawa habis dan tidak dapat menyelesaikan misi yang telah ditentukan, maka pemain harus mengulang sesuai *level* yang dimainkan.

4.2.3 Cek Rules dalam Knowledge Base

Di dalam *knowledge base* telah dibuat beberapa *rule* yang diterapkan pada *game* edukasi ini. Pada tahapan ini akan dicek kesamaan *rule-rule* yang ada dengan semua premis yang telah tersimpan. Dan jika telah menemukan *rule* dengan kondisi-kondisi yang sama dengan premis yang tersimpan, maka akan diambil kesimpulan yang telah ditentukan dalam tiap *rule*.

Di dalam *game* ini akan mendapatkan kesimpulan kenaikan *level* untuk *user* yang telah ditentukan tiap *rule* dari kesamaan kondisi pada *rule* dengan kondisi tiap premis. Berikut *rule* pada *Knowledge Base*:

Tabel 4.3 Rule pada Knowledge Base

| No | Kondisi | Hasil |
|----|---|--------------|
| 1 | A01='3' AND A02='100' AND A03='T' | L01 |
| | Nyawa sisa 3, Score 100, terpenuhi | Lanjut level |
| 2 | A01='2' AND A02='100' AND A03='T' | L01 |
| | Nyawa sisa 2, Score 100, terpenuhi | Lanjut level |
| 3 | A01='0' AND A02='70' AND A03='TT' | L02 |
| | Nyawa sisa 0, Score 70, Tidak terpenuhi | Ulang |
| 4 | A01='0' AND A02='50' AND A03='TT' | L02 |
| | Nyawa sisa 3, Score 50, Tidak terpenuhi | Ulang |
| 5 | A01='1' AND A02='100' AND A04='T' | L01 |
| | Nyawa sisa 3, Score 70, terpenuhi | Lanjut level |

Pengujian ini diterapkan pada *level* satu sebagai contoh. *Player* diminta untuk menyelesaikan permainan pada *level* 1 ini. Ketika *player* selesai memainkan *level* 1 ini maka algoritma *Forward Chaining* akan bekerja sesuai dengan data input atau hasil akhir yang masuk. Permainan ini membahas sampai *level* 5 dengan ketentuan tiap *level* berbeda.

4.3 Implementasi *Game*

Pada bagian ini akan dipaparkan tentang implementasi dari rancangan *game Mr Garbage* dan penjelasan tiap-tiap *scene*. Berikut adalah tampilan *game Mr Garbage* yang telah selesai dibuat.

4.3.1 Tampilan *Splashscene*

Tampilan pembuka halaman akan tampil pada saat pertama kali *game* dijalankan sebelum ke tampilan utama *game*. Pada tampilan ini akan tampil 3 menu utama.



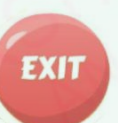


Gambar 4.1 Tampilan *Splashscreen*

4.3.2 Tampilan Menu *Game*

Pada bagian ini menampilkan menu-menu pilihan. Pada tampilan menu terdapat 3 menu yaitu *Play*, *Introduction* dan *Exit* yang memiliki fungsi masing-masing yaitu:

Tabel 4.4 Keterangan Fungsi Tombol Menu *Game*

| No. | Menu | Keterangan |
|-----|---|--|
| 1 |  | Berfungsi untuk memulai <i>game</i> pada saat pertama kali memainkan <i>game</i> |
| 2 |  | Berfungsi untuk menampilkan informasi tentang <i>game</i> |
| 3 |  | Berfungsi untuk keluar dari permainan |

4.3.3 Tampilan *Scene Introduction*

Bagian ini menampilkan penjelasan misi yang harus diselesaikan pemain untuk membersihkan lingkungan dari sampah serta menjelaskan jenis-jenis sampah sesuai dengan kategorinya.

Gambar 4.2 Tampilan Menu *Introduction* (1)



Gambar 4.3 Tampilan Menu *Introduction* (2)



Gambar 4.4 Tampilan Menu *Introduction* (3)



Gambar 4.5 Tampilan Menu *Introduction* (4)

4.3.4 Tampilan *Game*

Setelah memilih menu *Play* maka pemain akan memasuki arena permainan seperti pada **Gambar 4.6**. Dimana nyawa awal dari *player* adalah 3 bar yang ditandai dengan ikon hati pada pojok kiri atas layar *game*.



Gambar 4.6 Tampilan *gameplay* dan posisi awal permainan

Pada **Gambar 4.7** merupakan tampilan *gameplay* dan posisi awal pemain pada *level 1*. *Scene* pada tampilan *gameplay level 1* adalah halaman depan rumah.



Gambar 4.7 Mulai permainan pada *level 1*

Pada **Gambar 4.8** merupakan tampilan *gameplay* dan posisi awal pemain pada *level 2*. *Scene* pada tampilan *gameplay level 2* adalah tengah pergudangan.



Gambar 4.8 Tampilan *gameplay* dan posisi awal permainan pada *level 2*



Gambar 4.9 Mulai permainan pada *level 2*

Pada **Gambar 4.10** merupakan tampilan *gameplay* dan posisi awal pemain pada *level 3*. *Scene* pada tampilan *gameplay level 3* adalah komplek perumahan.

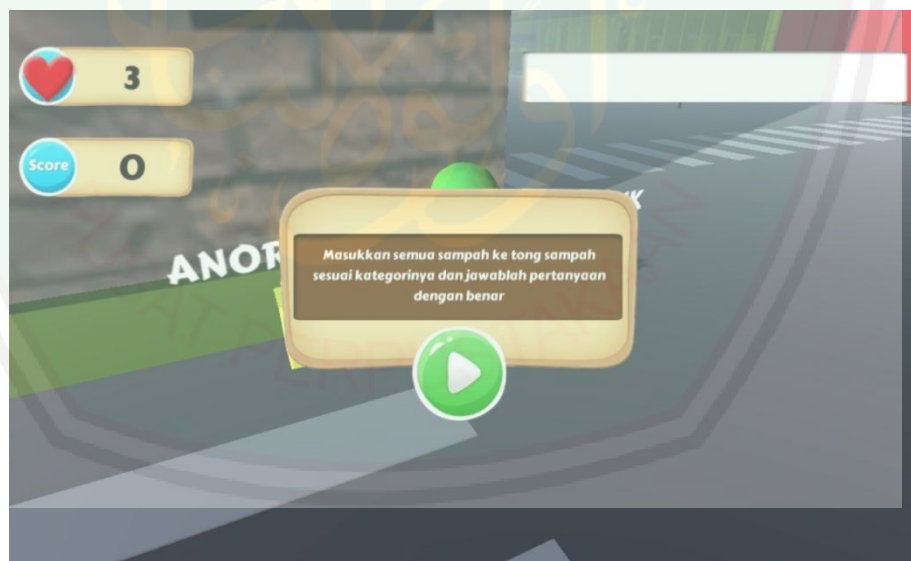


Gambar 4. 10 Tampilan *gameplay* dan posisi awal permainan pada *level 3*



Gambar 4. 11 Mulai permainan pada *level 3*

Pada **Gambar 4.12** merupakan tampilan *gameplay* dan posisi awal pemain pada *level 4*. *Scene* pada tampilan *gameplay level 4* adalah tengah perkotaan.

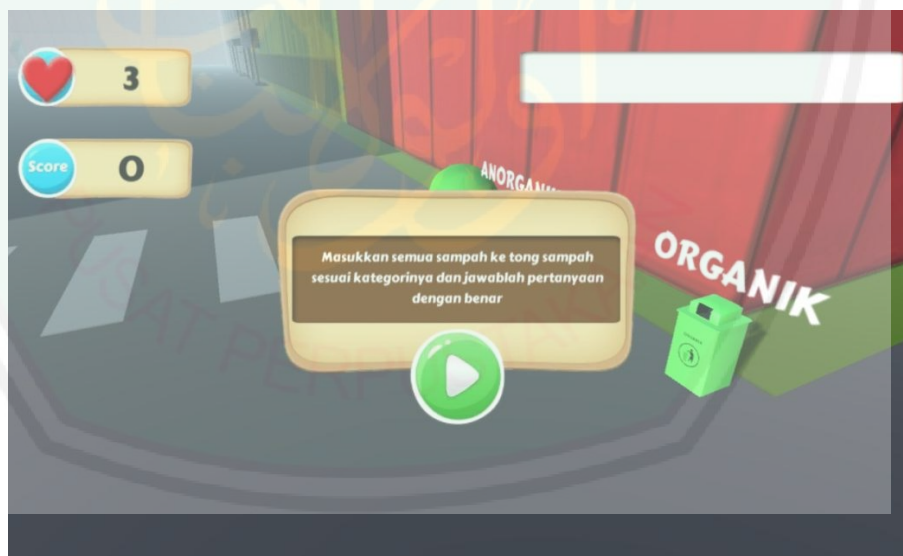


Gambar 4.12 Tampilan *gameplay* dan posisi awal permainan pada *level 4*



Gambar 4.13 Mulai permainan pada *level 4*

Pada **Gambar 4.14** merupakan tampilan *gameplay* dan posisi awal pemain pada *level 5*. *Scene* pada tampilan *gameplay level 5* adalah gedung perkantoran.



Gambar 4.14 Tampilan *gameplay* dan posisi awal permainan pada *level 5*



Gambar 4.15 Mulai permainan pada *level 5*

Sapu ajaib pada **Gambar 4.16** akan menampilkan pertanyaan. Sapu ajaib ini juga akan hilang ketika kita sudah menjawab pertanyaan yang ada.



Gambar 4.16 Pemain menemukan sapu ajaib

Jika pemain menjawab pertanyaan dengan benar maka pemain akan mendapatkan senjata sapu ajaib yang bisa digunakan untuk mengusir kucing pengganggu seperti pada **Gambar 4.18**.



Gambar 4.17 Tampilan pertanyaan



Gambar 4.18 Pemain mendapatkan senjata sapu

Agar nyawa tidak berkurang maka pemain harus menghindari *enemy* yaitu berupa seekor kucing seperti pada **Gambar 4.19**. Kucing ini diletakkan di beberapa sudut pada setiap *level* dan setiap 1 kucing akan mengurangi 1 nyawa pemain. Jika nyawa habis sebelum menyelesaikan misi, maka permainan akan berakhir seperti pada **Gambar 4.21**.



Gambar 4.19 Pemain bertemu kucing pengganggu

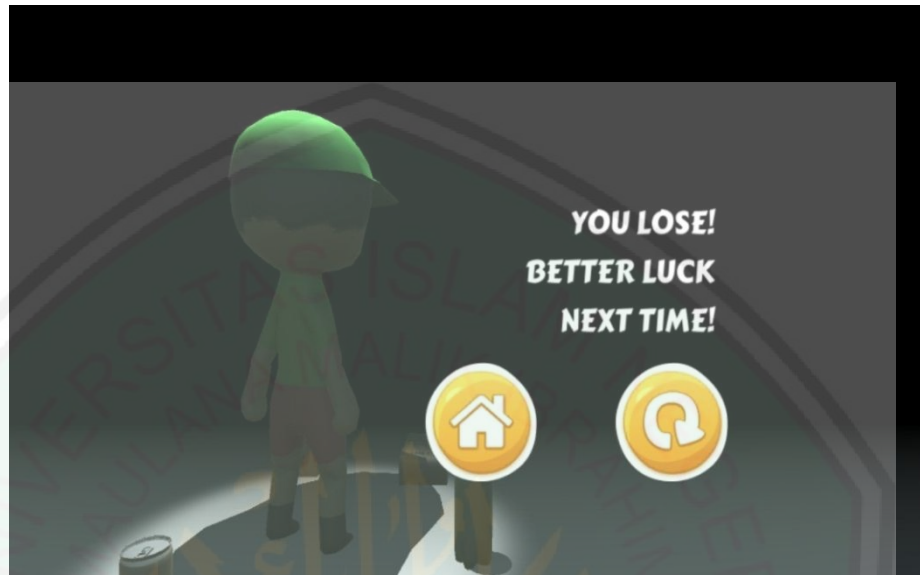
Pada **Gambar 4.20** merupakan tampilan *interface pause*. Pada halaman ini akan tampil saat pengguna ingin menghentikan sementara permainan. Pengguna dapat menekan tombol 'ESC' pada *keyboard*.



Gambar 4.20 Tampilan *interface pause*

Pada **Gambar 4.21** merupakan tampilan *gameover*. Pada halaman ini terdapat pertanyaan apakah pemain ingin mencoba lagi permainan atau tidak. Jika

pemain ingin memilih *yes*, maka pemain akan mengulang permainan. Tetapi jika, maka pemain akan kembali ke halaman menu utama.



Gambar 4.21 Tampilan *game over*

Pada **Gambar 4.22** merupakan halaman tampilan ketika pemain berhasil melakukan menyelesaikan misinya. Dan menuju halaman misi untuk *level* selanjutnya.



Gambar 4.22 Tampilan *level up*



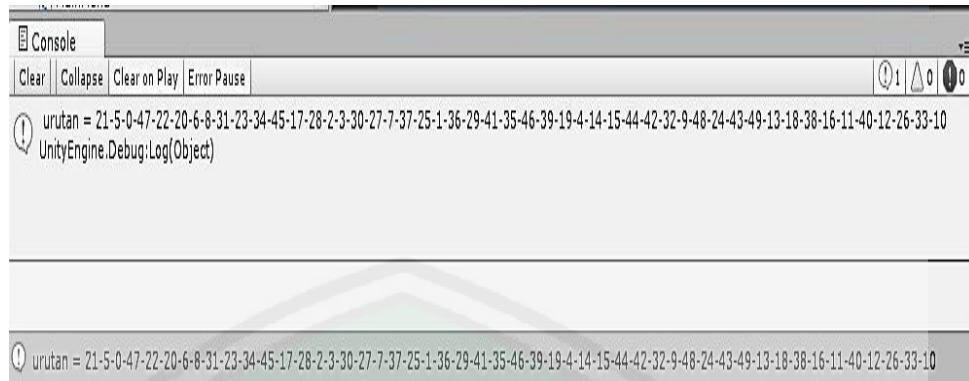
Gambar 4.23 Tampilan *mission completed*

4.4 Uji Coba

Pada sub bab ini membahas tentang uji coba yang telah dilakukan. Terdapat tiga uji coba yakni uji coba *game* menggunakan algoritma *Fisher Yates Shuffle*, uji coba *game* menggunakan algoritma *Forward Chaining* dan uji coba pada *game*. Berikut pembahasan uji coba tersebut:

4.4.1 Uji Coba *Game* Menggunakan Algoritma *Fisher Yates Shuffle*

Uji coba algoritma *Fisher Yates Shuffle* pada *game* ini dilakukan untuk mengacak pertanyaan yang akan muncul pada *level* 4 dan 5. Perulangan dilakukan sebanyak data yang ada dan data yang sudah diambil tidak akan diambil kembali. Oleh karena itu hampir tidak mungkin 2 permutasi memiliki kesamaan. Untuk menguji metode *Fisher Yates Shuffle* pengujian hasil metode tersebut ditampilkan pada console unity sehingga dapat diketahui apakah *array* sudah diacak atau belum. Berikut adalah hasil pengacakan soal yang telah dilakukan yang ditunjukkan pada **Gambar 4.24**.



Gambar 4.24 Hasil pengacakan soal

Pada **Gambar 4.24** di atas adalah tampilan tab console di Unity ketika *game* dimainkan. Gambar diatas menunjukkan hasil *array* yang telah diacak urutannya dengan algoritma *Fisher Yates Shuffle*. Hasil tersebut langsung muncul ketika permainan dimulai pada *level 4* dan *level 5*. Hasil urutan terbaru tersebut akan digunakan sebagai urutan konten kebersihan lingkungan yang akan ditampilkan setiap kali *player* mendapat satu sapu ajaib. Misalkan pada percobaan ini hasil urutannya mulai dari urutan pertama adalah 21, 5, 0, 47, 22 dan seterusnya sampai urutan ke 50. Maka pada sapu ajaib pertama yang di dapat *player* akan ditampilkan pertanyaan pengetahuan lingkungan nomor 21, sapu ajaib kedua akan ditampilkan pertanyaan pengetahuan lingkungan nomor 5, sapu ajaib ketiga akan ditampilkan pertanyaan pengetahuan lingkungan nomor 0 dan begitu juga seterusnya setiap kali *player* mendapatkan sapu ajaib.

Tabel 4.5 Hasil Uji Coba Urutan Baru yang dihasilkan Algoritma Fisher Yates Shuffle

| PK | Hasil Urutan Baru |
|----|--|
| 1 | 21, 5, 0, 47, 22, 20, 6, 8, 31, 23, 34, 45, 17, 28, 2, 3, 30, 27, 7, 37, 25, 1, 36, 29, 41, 35, 46, 39, 19, 4, 14, 15, 44, 42, 32, 9, 48, 24, 43, 49, 13, 18, 38, 16, 11, 40, 12, 26, 33, 10 |
| 2 | 27, 37, 10, 26, 25, 2, 49, 18, 23, 19, 11, 24, 42, 43, 5, 31, 0, 47, 12, 15, 28, 17, 30, 35, 44, 33, 4, 45, 8, 3, 34, 9, 32, 22, 36, 48, 1, 21, 7, 16, 13, 6, 41, 39, 14, 20, 46, 29, 40, 38 |
| 3 | 4, 15, 16, 24, 2, 32, 34, 48, 44, 6, 38, 7, 14, 49, 12, 39, 20, 36, 40, 3, 33, 25, 23, 26, 17, 42, 22, 35, 10, 46, 11, 9, 45, 41, 31, 5, 19, 29, 30, 0, 43, 47, 8, 27, 21, 1, 13, 28, 37, 18 |
| 4 | 1, 30, 35, 46, 8, 16, 11, 44, 49, 47, 23, 6, 5, 14, 0, 45, 3, 48, 9, 7, 21, 19, 39, 26, 17, 10, 43, 28, 4, 15, 38, 25, 18, 24, 34, 27, 31, 32, 22, 20, 37, 42, 40, 29, 12, 41, 2, 36, 33, 13 |
| 5 | 41, 21, 37, 14, 27, 31, 28, 6, 40, 25, 15, 29, 32, 0, 19, 3, 42, 33, 5, 20, 35, 17, 26, 30, 47, 38, 4, 39, 1, 13, 49, 2, 34, 23, 18, 11, 8, 45, 48, 36, 12, 43, 44, 46, 10, 22, 16, 24, 7, 9 |
| 6 | 31, 3, 11, 9, 49, 37, 17, 10, 4, 48, 18, 2, 16, 46, 14, 38, 8, 12, 5, 13, 22, 34, 26, 28, 25, 47, 42, 41, 30, 39, 32, 40, 45, 1, 35, 29, 23, 33, 43, 0, 27, 19, 36, 24, 44, 15, 21, 20, 6, 7 |
| 7 | 1, 12, 17, 15, 32, 20, 44, 14, 13, 4, 8, 48, 10, 33, 7, 26, 45, 23, 11, 40, 46, 34, 38, 42, 21, 29, 28, 6, 2, 0, 3, 39, 37, 27, 47, 24, 22, 36, 43, 5, 41, 49, 18, 9, 25, 30, 19, 31, 35, 16 |
| 8 | 11, 30, 0, 49, 1, 17, 4, 37, 20, 18, 12, 42, 21, 7, 36, 38, 43, 46, 44, 25, 13, 8, 26, 41, 31, 27, 22, 39, 28, 16, 23, 15, 33, 2, 48, 9, 32, 40, 3, 5, 6, 10, 14, 19, 24, 29, 34, 35, 45, 47 |
| 9 | 17, 12, 36, 39, 41, 18, 5, 48, 7, 1, 30, 14, 49, 11, 44, 0, 42, 4, 3, 27, 29, 28, 40, 47, 19, 32, 13, 23, 6, 16, 2, 26, 31, 33, 34, 24, 22, 46, 45, 37, 15, 10, 20, 43, 21, 35, 8, 25, 9, 38 |
| 10 | 3, 37, 31, 25, 15, 19, 42, 2, 10, 5, 24, 40, 35, 6, 23, 8, 44, 45, 16, 36, 11, 18, 13, 29, 39, 43, 38, 0, 17, 12, 32, 22, 7, 34, 9, 20, 21, 49, 27, 14, 41, 48, 33, 47, 4, 1, 30, 26, 28, 46 |

Tabel 4.5 diatas merupakan hasil uji coba urutan baru yang dihasilkan oleh algoritma *Fisher Yates Shuffle*. Uji coba ini dilakukan untuk menunjukkan bahwa algoritma ini selalu menghasilkan urutan baru yang berbeda setiap kali dijalankan. Pada tabel tersebut menunjukkan hasil uji coba algoritma tersebut sebanyak 10 kali. Kolom pertama berjudul PK atau percobaan ke-x, dan pada kolom kedua hasil urutan baru setelah program dijalankan. Hasil uji coba pada **Tabel 4.5** menunjukkan bahwa tidak ada satupun urutan baru yang dihasilkan oleh algoritma tersebut sama dengan urutan yang lainnya.

Dalam setiap kali permainan tidak semua konten pertanyaan yang berjumlah 50 itu akan ditampilkan. Pada *game* yang telah dibuat terdapat 10 sapu ajaib, jadi pertanyaan pengetahuan lingkungan yang akan ditampilkan adalah pertanyaan pengetahuan dengan nomor-nomor yang sesuai dengan 10 angka pertama pada urutan baru yang dihasilkan. Berikut 10 pertanyaan dalam sapu ajaib hasil 10 kali uji coba yang akan keluar.

Tabel 4.6 Hasil Uji Coba Algoritma *Fisher Yates Shuffle*

| PK | Hasil |
|----|---------------------------------------|
| 1 | 21, 5, 0, 47, 22, 20, 6, 8, 31, 23 |
| 2 | 27, 37, 10, 26, 25, 2, 49, 18, 23, 19 |
| 3 | 4, 15, 16, 24, 2, 32, 34, 48, 44, 6 |
| 4 | 1, 30, 35, 46, 18, 16, 11, 44, 49, 47 |
| 5 | 41, 20, 37, 14, 27, 31, 28, 6, 40, 25 |
| 6 | 31, 3, 11, 9, 49, 37, 17, 11, 4, 48 |
| 7 | 1, 12, 17, 15, 32, 20, 44, 24, 13, 4 |
| 8 | 11, 30, 44, 49, 1, 17, 4, 37, 20, 18 |
| 9 | 17, 19, 36, 39, 41, 18, 5, 48, 7, 1 |
| 10 | 3, 37, 31, 25, 15, 19, 42, 2, 16, 5 |

Tabel 4.6 diatas merupakan 10 konten soal hasil uji coba pengacakan dengan algoritma *Fisher Yates Shuffle*. Dari tabel tersebut dapat dicari berapa nomor soal yang sering muncul dalam 10 kali percobaan. Dan didapatkan pada **Tabel 4.7**.

Tabel 4.7 Frekuensi kemunculan angka

| No Soal | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---------|---|---|---|---|---|---|---|---|---|---|
| Iterasi | 1 | 4 | 3 | 2 | 3 | 3 | 3 | 1 | 1 | 1 |

Lanjutan Tabel 4.7

| No Soal | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|---------|----|----|----|----|----|----|----|----|----|----|
| Iterasi | 1 | 3 | 1 | 1 | 1 | 3 | 2 | 4 | 3 | 2 |

Lanjutan Tabel 4.7

| No Soal | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
|---------|----|----|----|----|----|----|----|----|----|----|
| Iterasi | 3 | 1 | 1 | 2 | 1 | 3 | 1 | 2 | 1 | 0 |

Lanjutan Tabel 4.7

| No Soal | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
|---------|----|----|----|----|----|----|----|----|----|----|
| Iterasi | 2 | 4 | 2 | 0 | 1 | 1 | 1 | 5 | 0 | 1 |

Lanjutan Tabel 4.7

| No Soal | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 |
|---------|----|----|----|----|----|----|----|----|----|----|
| Iterasi | 1 | 1 | 1 | 0 | 3 | 0 | 1 | 2 | 3 | 4 |

Dari **Tabel 4.7** dapat diperoleh presentase Iterasi kemunculan dari pengacakan menggunakan algoritma *Fisher Yates Shuffle*. Hasilnya dapat dilihat pada **Tabel 4.8**.

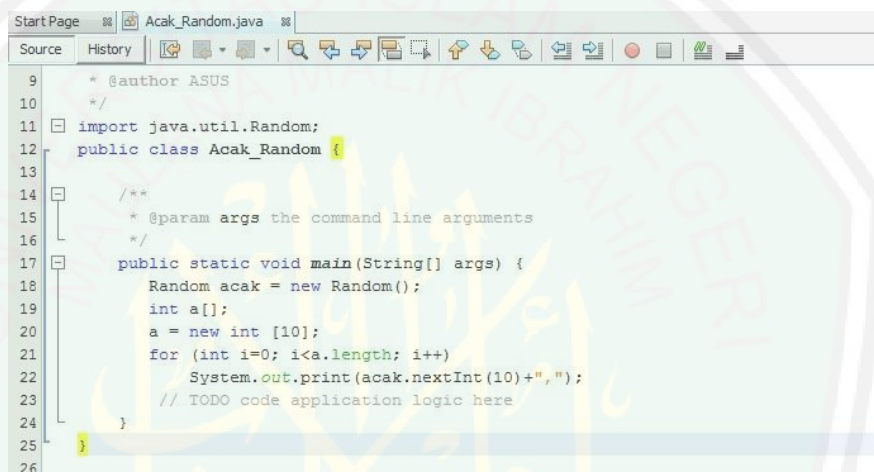
Tabel 4.8 Presentase Frekuensi perulangan kemunculan angka

| Iterasi kemunculan | Jumlah nomor soal | Presentase iterasi kemunculan |
|--------------------|-------------------|-------------------------------|
| 0 | 5 | 5% |
| 1 | 21 | 21% |
| 2 | 8 | 8% |
| 3 | 10 | 10% |
| 4 | 4 | 4% |
| 5 | 1 | 1% |

Dari **Tabel 4.8** dapat dilihat presentase frekuensi kemunculan angka dapat diketahui dari 50 nomor soal, terdapat 21 nomor soal hanya muncul sekali dan 5 nomor soal yang tidak muncul. Berarti terdapat 26 soal yang tidak mengalami perulangan sehingga presentase soal yang tidak mengalami perulangan sebesar 26%. Sedangkan terdapat 8 nomor soal muncul sebanyak 2 kali, 10 nomor soal

muncul sebanyak 3 kali, 4 nomor soal muncul sebanyak 4 kali, dan 1 nomor soal muncul sebanyak 5 kali. Berarti terdapat 23 soal yang mengalami perulangan sehingga presentase soal yang mengalami perulangan sebesar 23%.

Pada pengacakan menggunakan random generator java dilakukan hal yang sama yaitu menggunakan 10 data, dimana data tersebut diambil secara acak. Berikut *pseudo code* pengacakan menggunakan random biasa:



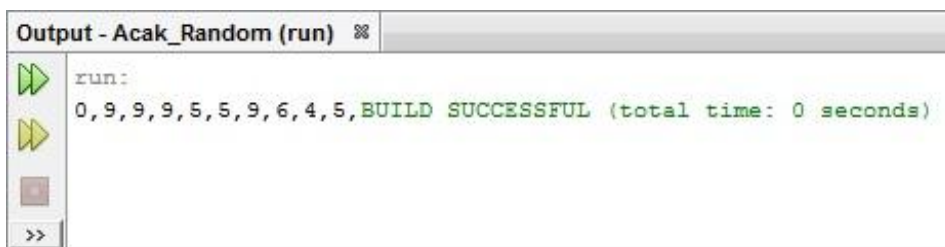
```

9      * @author ASUS
10     */
11     import java.util.Random;
12     public class Acak_Random {
13
14         /**
15          * @param args the command line arguments
16          */
17         public static void main(String[] args) {
18             Random acak = new Random();
19             int a[];
20             a = new int [10];
21             for (int i=0; i<a.length; i++)
22                 System.out.print(acak.nextInt(10)+" ");
23             // TODO code application logic here
24         }
25     }
26

```

Gambar 4.25 Pseudo code generator java

Dari hasil *pseudo code* diatas menampilkan 10 data yang diambil. Dibandingkan dengan algoritma *Fisher Yates Shuffle* , *random* biasa menghasilkan beberapa data yang sama seperti 9 sebanyak 4 kali dan 5 sebanyak 3 kali.



```

Output - Acak_Random (run)
run:
0, 9, 9, 9, 5, 5, 9, 6, 4, 5, BUILD SUCCESSFUL (total time: 0 seconds)

```

Gambar 4.26 Hasil running generator java

Tabel 4.9 Hasil Pengujian Algoritma Fisher Yates Shuffle Pada Game

| Uji Coba ke | Level | Masukkan | Hasil yang diharapkan | Berhasil | Tidak berhasil |
|-------------|-------|----------------|----------------------------------|----------|----------------|
| 4 | 1-10 | Kali game play | Pengacakan dalam game Mr Garbage | soal | √ |
| 5 | 1-10 | Kali game play | Pengacakan dalam game Mr Garbage | soal | √ |

Dari hasil **Tabel 4.9** diatas telah dilakukan pengujian terhadap Algoritma *Fisher Yates Shuffle* pada *game* yang dilakukan dengan cara memainkan *game* sebanyak 10x secara terus menerus agar dapat dilihat apakah permutasi yang diterapkan dalam *game* berhasil dijalankan.

4.4.2 Uji Coba Game Menggunakan Algoritma *Forward Chaining*

Pada pengujian *Forward Chaining* digunakan untuk menentukan kenaikan *level* seorang *player* dengan suatu kondisi dimana hasil aturan atau *rule – rule* yang diasumsikan. Proses pengujian dilakukan pada desain setiap tampilan *scene* pada *game*.

Berikut adalah pengujian yang dilakukan terhadap sistem terutama pengujian terhadap kerja metode *Forward Chaining*. Dari hasil pengujian pada **Tabel 4.3** jika dilakukan *review* terhadap *rules* yang sudah diterapkan pada

metode *Forward Chaining* dalam *game* ini maka akan didapatkan perhitungan sebagai berikut:

Rules:

```
A01='3' AND A02='100' AND A03='T' THEN L01
A01='2' AND A02='100' AND A03='T' THEN L01
A01='0' AND A02='70' AND A03='TT' THEN L02
A01='0' AND A02='50' AND A03='TT' THEN L02
A01='1' AND A02='100' AND A03='T' THEN L01
```

Jumlah valid = 5

$$\begin{aligned} \text{Presentase validitas} &= \text{jumlah valid/jumlah pengujian} * 100\% \\ &= 5/5 * 100\% \\ &= 100\% \end{aligned}$$

Perhitungan nilai validitas diatas didasarkan pada perhitungan pengujian terhadap *rules Forward Chaining* pada *game*. Dari hasil perhitungan didapatkan nilai validitas sebesar 100%. Hal ini membuktikan bahwa hasil dan *review* yang didapat dalam *game* ketika bermain sama dengan *rules* yang sudah dimasukkan dalam metode *Forward Chaining*.

4.4.3 Pengujian Level

Hasil uji coba masing-masing *level* di jelaskan pada tabel di bawah ini:

1. Pengujian *level* 1

Tabel 4.10 Aturan Level 1

| Score | Target | Nyawa |
|-------|-----------|-------|
| >=50 | Terpenuhi | >=3 |

- Target yang harus diperoleh adalah *score* ≥ 50 , dan target harus terpenuhi

| No | Score | Target | Nyawa | Hasil |
|----|-------|-----------------|-------|-------|
| 1 | 100 | Terpenuhi | 3 | Lolos |
| 2 | 100 | Terpenuhi | 2 | Lolos |
| 3 | 70 | Tidak Terpenuhi | 0 | Gagal |
| 4 | 50 | Tidak Terpenuhi | 0 | Gagal |
| 5 | 100 | Terpenuhi | 1 | Lolos |

Penjelasan:

1. Pemain mendapatkan *score* 100, target terpenuhi, nyawa sama dengan 3 dan hasil lolos karena semua kriteria terpenuhi.
2. Pemain mendapatkan *score* 100, target terpenuhi, nyawa sama dengan 2 dan hasil lolos karena semua kriteria terpenuhi.
3. Pemain mendapatkan *score* 70, target tidak terpenuhi, nyawa sama dengan 0 dan hasil gagal karena kriteria tidak terpenuhi.
4. Pemain mendapatkan *score* 50, target tidak terpenuhi, nyawa sama dengan 0 dan hasil gagal karena kriteria tidak terpenuhi.
5. Pemain mendapatkan *score* 100, target terpenuhi, nyawa sama dengan 1 dan hasil lolos karena semua target terpenuhi.

Berikut penjelasan dari **Tabel 4.11** adalah hasil yang diharapkan merupakan hasil harus dicapai sesuai dengan ketentuan kenaikan *level* minimal pada *level* 1 yang telah dijelaskan diatas.

Tabel 4.11 Hasil Pengujian Algoritma *Forward Chaining Level 1*

| Perco baan ke- | Score | Target | Nyaw a | Hasil Level Forward Chaining | Hasil Level Yang Diharapkan | Kesimpulan |
|----------------------|-------|--------|-----------|------------------------------------|-----------------------------------|------------|
| 1 | 100 | T | 3 | Lolos | Lolos | Sesuai |
| 2 | 100 | T | 2 | Lolos | Lolos | Sesuai |
| 3 | 70 | TT | 0 | Gagal | Gagal | Sesuai |
| 4 | 50 | TT | 0 | Gagal | Gagal | Sesuai |
| 5 | 100 | T | 1 | Lolos | Lolos | Sesuai |

Pada uji coba *level 1* data menunjukkan kesesuaian antara hasil uji kenaikan *level Forward Chaining* dan hasil uji kenaikan *level* yang diharapkan dapat bekerja dengan baik.

2. Pengujian *Level 2*

Tabel 4.12 Aturan *Level 2*

| Score | Target | Nyawa |
|-----------|-----------|----------|
| ≥ 60 | Terpenuhi | ≥ 3 |

- Target yang harus diperoleh adalah *score* ≥ 60 , dan target harus terpenuhi

| No | Score | Target | Nyawa | Hasil |
|----|-------|-----------------|-------|-------|
| 1 | 60 | Terpenuhi | 3 | Lolos |
| 2 | 70 | Terpenuhi | 2 | Lolos |
| 3 | 60 | Terpenuhi | 1 | Lolos |
| 4 | 40 | Tidak Terpenuhi | 0 | Gagal |
| 5 | 30 | Tidak Terpenuhi | 0 | Gagal |

Penjelasan:

1. Pemain mendapatkan *score* 60, target terpenuhi, nyawa sama dengan 3 dan hasil lolos karena semua target terpenuhi.
2. Pemain mendapatkan *score* 70, target terpenuhi, nyawa sama dengan 2 dan hasil lolos karena semua target terpenuhi.
3. Pemain mendapatkan *score* 60, target terpenuhi, nyawa sama dengan 1 dan hasil lolos karena semua target terpenuhi.
4. Pemain mendapatkan *score* 40, target tidak terpenuhi, nyawa sama dengan 0 dan hasil gagal karena semua kriteria tidak memenuhi target.
5. Pemain mendapatkan *score* 30, target tidak terpenuhi, nyawa sama dengan 0 dan hasil gagal karena semua kriteria tidak memenuhi target.

Berikut penjelasan dari **Tabel 4.13** adalah hasil yang diharapkan merupakan hasil harus dicapai sesuai dengan ketentuan kenaikan *level* minimal pada *level* 2 yang telah dijelaskan diatas.

Tabel 4.13 Hasil Pengujian Algoritma *Forward Chaining Level 2*

| Perco baan ke- | <i>Score</i> | Target | Nyaw a | Hasil <i>Level</i> <i>Forward</i> <i>Chaining</i> | Hasil <i>Level</i> Yang Diharapkan | Kesimpulan |
|----------------------|--------------|--------|-----------|---|--|------------|
| 1 | 60 | T | 3 | Lolos | Lolos | Sesuai |
| 2 | 70 | T | 2 | Lolos | Lolos | Sesuai |
| 3 | 60 | T | 1 | Lolos | Lolos | Sesuai |
| 4 | 40 | TT | 0 | Gagal | Gagal | Sesuai |
| 5 | 30 | TT | 0 | Gagal | Gagal | Sesuai |

Pada uji coba *level 2* data menunjukkan kesesuaian antara hasil uji kenaikan *level Forward Chaining* dan hasil uji kenaikan *level* yang diharapkan dapat bekerja dengan baik.

3. Pengujian *Level 3*

Tabel 4.14 Aturan *Level 3*

| Score | Target | Nyawa |
|-----------|-----------|----------|
| ≥ 70 | Terpenuhi | ≥ 3 |

- Target yang harus diperoleh adalah skor ≥ 70 , dan target harus terpenuhi

| No | Score | Target | Nyawa | Hasil |
|----|-------|-----------------|-------|-------|
| 1 | 50 | Tidak Terpenuhi | 3 | Gagal |
| 2 | 80 | Terpenuhi | 3 | Lolos |
| 3 | 60 | Tidak Terpenuhi | 0 | Gagal |
| 4 | 70 | Terpenuhi | 2 | Lolos |
| 5 | 60 | Tidak Terpenuhi | 3 | Gagal |

Penjelasan:

1. Pemain mendapatkan *score* 50, target tidak terpenuhi, nyawa sama dengan 3 dan hasil gagal karena semua kriteria tidak memenuhi target.
2. Pemain mendapatkan *score* 80, target terpenuhi, nyawa sama dengan 3 dan hasil lolos karena semua target terpenuhi.

3. Pemain mendapatkan *score* 60, target tidak terpenuhi, nyawa sama dengan 0 dan hasil gagal karena semua kriteria tidak memenuhi target.
4. Pemain mendapatkan *score* 70, target terpenuhi, nyawa sama dengan 2 dan hasil lolos karena semua target terpenuhi.
5. Pemain mendapatkan *score* 60, target tidak terpenuhi, nyawa sama dengan 3 dan hasil gagal karena *score* tidak memenuhi target.

Berikut penjelasan dari **Tabel 4.15** adalah hasil yang diharapkan merupakan hasil harus dicapai sesuai dengan ketentuan kenaikan *level* minimal pada *level* 3 yang telah dijelaskan diatas.

Tabel 4.15 Hasil Pengujian Algoritma *Forward Chaining* Level 3

| Perco baan ke- | <i>Score</i> | Target | Nyawa | Hasil <i>Level Forward Chaining</i> | Hasil <i>Level Yang Diharapkan</i> | Kesimpulan |
|----------------------|--------------|--------|-------|---|--|------------|
| 1 | 50 | TT | 3 | Gagal | Gagal | Sesuai |
| 2 | 80 | T | 3 | Lolos | Lolos | Sesuai |
| 3 | 60 | TT | 0 | Gagal | Gagal | Sesuai |
| 4 | 70 | T | 2 | Lolos | Lolos | Sesuai |
| 5 | 60 | TT | 3 | Gagal | Gagal | Sesuai |

Pada uji coba *level* 3 data menunjukkan kesesuaian antara hasil uji kenaikan *level Forward Chaining* dan hasil uji kenaikan *level* yang diharapkan dapat bekerja dengan baik.

4. Pengujian *level 4***Tabel 4.16 Aturan Level 4**

| Score | Target | Nyawa |
|-----------|-----------|-------|
| ≥ 80 | Terpenuhi | 3 |

- Target yang harus diperoleh adalah *score* ≥ 80 , target harus terpenuhi dan nyawa ≤ 0

| No | Score | Target | Nyawa | Hasil |
|----|-------|-----------------|-------|-------|
| 1 | 40 | Tidak Terpenuhi | 3 | Gagal |
| 2 | 80 | Terpenuhi | 0 | Gagal |
| 3 | 50 | Tidak Terpenuhi | 3 | Gagal |
| 4 | 80 | Terpenuhi | 2 | Lolos |
| 5 | 90 | Terpenuhi | 1 | Lolos |

Penjelasan:

1. Pemain mendapatkan *score* 40, target tidak terpenuhi, nyawa sama dengan 3 dan hasil gagal karena *score* tidak memenuhi target.
2. Pemain mendapatkan *score* 80, target terpenuhi, nyawa sama dengan 0 dan hasil gagal karena nyawa tidak memenuhi target.
3. Pemain mendapatkan *score* 50, target tidak terpenuhi, nyawa sama dengan 3 dan hasil gagal karena *score* tidak memenuhi target.

4. Pemain mendapatkan score 80, target terpenuhi, nyawa sama dengan 2 dan hasil lolos karena semua target terpenuhi.
5. Pemain mendapatkan score 90, target terpenuhi, nyawa sama dengan 1 dan hasil lolos karena semua target terpenuhi.

Berikut penjelasan dari **Tabel 4.17** adalah hasil yang diharapkan merupakan hasil harus dicapai sesuai dengan ketentuan kenaikan *level* minimal pada *level* 4 yang telah dijelaskan diatas.

Tabel 4.17 Hasil Pengujian Algoritma *Forward Chaining* Level 4

| Perco baan ke- | Score | Target | Nyawa | Hasil Level <i>Forward Chaining</i> | Hasil Level Yang Diharapkan | Kesimpulan |
|----------------------|-------|--------|-------|--|-----------------------------------|------------|
| 1 | 40 | TT | 3 | Gagal | Gagal | Sesuai |
| 2 | 80 | T | 0 | Gagal | Gagal | Sesuai |
| 3 | 50 | TT | 3 | Gagal | Gagal | Sesuai |
| 4 | 80 | T | 2 | Lolos | Lolos | Sesuai |
| 5 | 90 | T | 1 | Lolos | Lolos | Sesuai |

Pada uji coba *level* 4 data menunjukkan kesesuaian antara hasil uji kenaikan *level Forward Chaining* dan hasil uji kenaikan *level* yang diharapkan dapat bekerja dengan baik.

5. Pengujian *level* 5

Tabel 4.18 Aturan *Level* 5

| Score | Target | Nyawa |
|-----------|-----------|-------|
| ≥ 90 | Terpenuhi | 3 |

- Target yang harus diperoleh adalah $score \geq 90$, target terpenuhi dan nyawa ≤ 0

| No | Score | Target | Nyawa | Hasil |
|----|-------|-----------------|-------|-------|
| 1 | 40 | Tidak Terpenuhi | 0 | Gagal |
| 2 | 90 | Terpenuhi | 3 | Lolos |
| 3 | 90 | Terpenuhi | 2 | Lolos |
| 4 | 80 | Tidak Terpenuhi | 1 | Gagal |
| 5 | 50 | Tidak Terpenuhi | 3 | Gagal |

Penjelasan:

1. Pemain mendapatkan $score$ 40, target tidak terpenuhi, nyawa sama dengan 0 dan hasil gagal karena kriteria tidak terpenuhi.
2. Pemain mendapatkan $score$ 90, target terpenuhi, nyawa sama dengan 3 dan hasil lolos karena semua target terpenuhi.
3. Pemain mendapatkan $score$ 90, target terpenuhi, nyawa sama dengan 2 dan hasil lolos karena semua target terpenuhi.
4. Pemain mendapatkan $score$ 80, target tidak terpenuhi, nyawa sama dengan 1 dan hasil gagal karena $score$ tidak memenuhi target.
5. Pemain mendapatkan $score$ 50, target tidak terpenuhi, nyawa sama dengan 3 dan hasil gagal karena $score$ tidak memenuhi target. dan target harus terpenuhi.

Berikut penjelasan dari **Tabel 4.19** adalah hasil yang diharapkan merupakan hasil harus dicapai sesuai dengan ketentuan kenaikan *level* minimal pada *level* 5 yang telah dijelaskan diatas.

Tabel 4.19 Hasil Pengujian Algoritma *Forward Chaining* Level 5

| Perco- baan ke- | Score | Target | Nyawa | Hasil Level <i>Forward Chaining</i> | Hasil Level Yang Diharapkan | Kesimpulan |
|--------------------|-------|--------|-------|--|-----------------------------------|------------|
| 1 | 40 | TT | 0 | Gagal | Gagal | Sesuai |
| 2 | 90 | T | 3 | Lolos | Lolos | Sesuai |
| 3 | 90 | T | 2 | Lolos | Lolos | Sesuai |
| 4 | 80 | TT | 1 | Gagal | Gagal | Sesuai |
| 5 | 50 | TT | 3 | Gagal | Gagal | Sesuai |

Pada uji coba *level* 5 data menunjukkan kesesuaian antara hasil uji kenaikan *level Forward Chaining* dan hasil uji kenaikan *level* yang diharapkan dapat bekerja dengan baik.

4.4.4 Analisa Hasil Uji Coba Algoritma *Forward Chaining* Pada Game

Berdasarkan pengujian algoritma *Forward Chaining* untuk masing-masing *level* didapatkan kesimpulan seperti pada **Tabel 4.20** berikut:

Tabel 4.20 Kesimpulan Hasil Uji Coba *Forward Chaining* Pada Game

| Uji Coba Level ke- | Jumlah Data Sesuai | Jumlah Data Tidak Sesuai |
|--------------------|--------------------|--------------------------|
| 1 | 5 | - |
| 2 | 5 | - |
| 3 | 5 | - |
| 4 | 5 | - |
| 5 | 5 | - |

Tabel 4.20 dapat dihitung presentasi akurasi keberhasilan *Forward Chaining* dalam menentukan kenaikan *level* pemain sebagai berikut:

$$\text{Akurasi} = \frac{\text{banyaknya prediksi yang benar}}{\text{total banyaknya prediksi}} \times 100\%$$

$$\text{Akurasi} = \frac{25}{25} \times 100\% = 100\%$$

Berdasarkan dari data-data diatas dapat dilihat bahwa akurasi dari perhitungan sistem menunjukkan hasil yang akurat. Dan menunjukkan bahwa *game* ini dapat berfungsi dengan baik. Hasil pengujian pada *game* dapat terlihat pada **Gambar 4.27** dan **Gambar 4.28** dibawah ini:



Gambar 4.27 Uji coba algoritma *Forward Chaining* jika nyawa penuh dan *score* sama dengan 100



Gambar 4.28 Uji coba algoritma *Forward Chaining* jika nyawa habis dan *score* kurang dari 100

4.5 Uji Coba *Game*

Uji coba ini dilakukan untuk mengetahui apakah *game* yang telah dibuat dapat diimplementasikan di PC. Berikut hasil pengujian yang disajikan dalam

Tabel 4.21:

Tabel 4.21 Tabel uji coba *game*

| No | Versi OS | Layar | Spesifikasi | RAM | Keterangan |
|----|-------------|-------|---|------|--|
| 1 | Windows 7 | 14" | Acer Aspire 4740 Processor Intel (R) Core(TM) 2,31Ghz (4CPU) VGA Intel HD Graphic Media Accelerator | 4 GB | Tampilan menu berjalan dengan baik. Tombol berfungsi dengan baik. Tampilan <i>game</i> berjalan dengan baik. |
| 2 | Windows 8 | 14" | Asus A43SA Processor Intel Core i3-2330M VGA ATI 6730 (2GB) | 2 GB | Tampilan menu berjalan dengan baik. Tombol berfungsi dengan baik. Tampilan <i>game</i> berjalan dengan baik. |
| 3 | Windows 8.1 | 14" | HP Processor Quadcore 2.1 Ghz VGA Radeon Dual Graphics (4 GB) | 4 GB | Tampilan menu berjalan dengan baik. Tombol berfungsi dengan baik. Tampilan <i>game</i> berjalan dengan baik. |
| 4 | Windows 10 | 14" | Axioo Neon HNM Prosesor Intel (R) Celeron (R) 1,70 GHz VGA Intel (R) HD Graphics | 2 GB | Tampilan menu berjalan dengan baik. Tombol berfungsi dengan baik. Tampilan <i>game</i> berjalan dengan baik. |

4.6 Integrasi Dalam 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 وَأُولَٰئِكَ 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* yang artinya:

“Dan hendaklah ada diantara kamu golongan umat yang menyeru kepada kebajikan.” Lalu beliau bersabda: *الخير اتباع القرآن وسنتي* artinya:

“Kebaikan itu adalah mengikuti sunnah Al Qur’an dan sunnahku”.

Dalam hal ini tafsiran lain kata “al khoiru” adalah kebaikan manusia pada agama dan dunia “*ما فيه صلاح الدنيا والدين* “. 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 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 Qur’an yang merupakan sumber tuntutan 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 الْبَحْرِ بِمَا كَسَبَتْ أَيْدِي النَّاسِ لِيُذِيقَهُمْ بَعْضَ الَّذِي
عَمِلُوا لَعَلَّهُمْ يَرْجِعُونَ

Artinya:

“Telah nampak kerusakan di darat dan di laut disebabkan karena perbuatan tangan manusia, supaya Allah merasakan kepada mereka sebagian 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 lingkungan dari sampah.

Hadist Anas bin Malik tentang Membuat Mudah, Gembira dan Kompak

عَنْ أَنَسِ بْنِ مَالِكٍ عَنِ النَّبِيِّ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ قَالَ يَسِّرُوا وَلَا
تُعَسِّرُوا وَأَوْسِرُوا وَلَا تَتَنَفَّرُوا

Artinya:

“Dari Anas bin Malik dari Nabi SAW “mudahkanlah dan jangan kamu persulit. Gembirakanlah dan jangan kamu membuat lari”. (HR. Abu Abdillah Muhammad bin Ismail al-Bukhori al-Ju’fi)”.

Hadist diatas menjelaskan bahwa proses pembelajaran harus dibuat dengan mudah sekaligus menyenangkan agar siswa atau anak kecil tidak tertekan secara psikologis dan tidak merasa bosan. Dan suatu pembelajaran juga harus menggunakan metode yang tepat disesuaikan dengan situasi dan kondisi, terutama dengan mempertimbangkan keadaan orang yang akan belajar. Dari ayat tersebut bisa juga menegaskan bahwa dalam suatu pembelajaran itu adalah untuk menguji kualitas diri siapa yang paling baik maka itulah yang akan menjadi pemenang.

Dalam hal inilah dimaksud adalah *game* atau permainan yang baik dapat menjadikan nilai tambah kualitas diri yang baik yang dapat menambah wawasan dan ilmu pengetahuan bukan hanya sekedar sebagai hiburan kesenangan saja. Dalam visi tertentu, sebuah program *game* bisa digunakan untuk hal-hal yang bermanfaat seperti simulasi dari sebuah pelajaran, latihan kecepatan dan ketepatan menembak dan sebagainya. Para calon pilot tempur pun menggunakan *game* komputer untuk berlatih sebelum mereka bertempur secara sesungguhnya di medan laga. Semua itu tergantung dari bagaimana cara menggunakan fasilitas modern itu. Namun kita harus mengakui bahwa ada sekian banyak orang yang telah menjadikan tempat-tempat *game* sebagai sarana buang waktu dan juga buang uang.

Dengan melihat ayat yang telah dipaparkan, peneliti mengambil kesimpulan bahwa untuk menyampaikan ilmu pendidikan, dapat melalui apa saja. Maka peneliti memilih media berupa *game* edukasi untuk pendalaman kebersihan

lingkungan. Tentu saja keinginan peneliti agar kian banyaknya peminat *game* edukasi selaras dengan kian bertambahnya *developer game* untuk memperkaya *game* edukasi sebagai media pengajaran yang inovatif bagi kita semua.



BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil implementasi dan pengujian yang dilakukan peneliti, maka dapat ditarik kesimpulan bahwa:

1. Penelitian ini berhasil membuat *game* edukasi kebersihan lingkungan pada Unity 3D sebagai *game engine* berbasis *desktop*.
2. Dari hasil implementasi dan uji coba, algoritma *Fisher Yates Shuffle* dapat digunakan untuk mengacak pertanyaan pada *game* edukasi *Mr Garbage* agar pertanyaan yang ditampilkan tidak berulang pada permainan berikutnya.
3. Metode *Forward Chaining* telah diterapkan pada pembuatan *game Mr Garbage*. Konsep metode *Forward Chaining* pada *game* ini digunakan untuk menentukan kenaikan *level* pemain. *Rules* atau aturan yang digunakan pada metode *forward chaining* ini sebanyak 3 *rules*, yaitu: nyawa, *score*, target.
4. Berdasarkan hasil pengujian proses *review* dan perpindahan *level*, *rules* yang digunakan pada metode *Forward Chaining* sebanyak 3 *rules*, dan hasil pengujian menunjukkan hasil nilai validitas sebesar 100%.
5. Berdasarkan hasil pengujian terhadap *game Mr Garbage*, dapat diketahui hasil pengujian sistem dapat berjalan dengan baik sebanyak 100%, hasil pengujian tombol berfungsi dengan baik sebanyak 100% dan hasil pengujian tampilan berjalan dengan baik sebanyak 100%.

5.2 Saran

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

1. Mengingat *genre* dari *game* ini adalah *game* edukasi, maka diharapkan banyak *game* dengan *genre* ini bermunculan dan juga peminat *game* edukasi kian meningkat.
2. Mengembangkan *game* ini agar memiliki tampilan yang lebih menarik.
3. Nantinya permainan ini diharapkan mampu untuk dikembangkan pada *platform smartphone* agar lebih menarik dan variatif.
4. Bisa mempublikasikan lewat *Playstore* atau *Appstore* agar dapat dimanfaatkan oleh khalayak umum.

DAFTAR PUSTAKA

- Adams, E. 2010. *Fundamental of Game Design 2nd Edition*. California: Barkley
- Ade-ibijola, abejideolu. 2012. A Simulated Enhancement of Fisher-Yates Algorithm for Shuffling in Virtual Card Games Using Domain-Specific Data Structures, *International Journal of Computer Applications*, Diakses 10 Juli 2017, dari <http://www.academia.edu/>,
- Agustinus Nilwan. 1998. *Pemrograman Animasi dan Game Profesional 4*. Jakarta : Elex Media Komputindo
- Alex S. Sukses Mengolah Sampah Organik Menjadi Pupuk Organik, hlm 9-10
- Al-Qur'an dan Terjemahannya. Kitab Suci Al-Qur'an Departemen Agama Republik Indonesia Surabaya: Al-Hidayah.
- Apperley, Thomas H. 2008. *Genre and Game Studies : Toward a Critical Approach to Video Game Genres*. University of Melbourne
- Bendersky, Eli. 2010. *The Intuition Behind Fisher Yates Shuffling*, <http://eli.thegreenplace.net>
- Black, Paul E. 2005. "Fisher-Yates Shuffle" *Dictionary of Algorithms and Data Structures*. National Institute of Standards and Technology.
- Cecep Dani Sucipto, *Teknologi Pengolahan Daur Ulang Sampah*, (Jakarta: Goysen Publishing, 2009), hlm 2-3
- Coding Horror, *The Danger of Naivete*. [Online]. Diakses 5 Juli 2017
- Daryanto, Agus Suprihatin. 2013. *Pengantar Pendidikan Lingkungan Hidup*. Yogyakarta. Penerbit Graha Media
- E. Colink, *Istilah Lingkungan Untuk Manajemen*, 1996
- E. Damanhuri dan Tri Padi, *Probleme de Dechets Urban en Indonesie, TFE ENTPE (Perancis), 1982 E. Damanhuri (Editor): Teknik Pengelolaan Persampahan – Modul A dan Modul B, Disiapkan untuk PT. Freeport Indoensia*, (Bandung: Teknik Lingkungan ITB, 1999).
- Henry, Samuel. 2005. *Panduan Praktis Membuat Game 3D*. Yogyakarta: Graha Ilmu
- Jambeck, Jena r.,et.al, 2015. "*plastic waste inputs from land into the ocean*" (www.sciencemag.org, February 12, 2015)
- Kartono. 1994. *Teori Permainan (Game Theory)*. Andi Offset : Yogyakarta)

- Katsir, Ibnu. 2003. Tafsir Ibnu Katsir Jilid 2. Diterjemahkan oleh: M. Abdul Ghofal E.M. Bogor: Pustaka Imam Asy-syafi'i
- Kusumadewi, Sri. 2003. Artificial Intelligence (Teknik dan Aplikasinya). Graha Ilmu
- M. Gelbert, dkk. Konsep Pendidikan Lingkungan Hidup dan "Wall Chart". Buku Panduan Pendidikan Lingkungan Hidup (Malang: PPPGT/VEDC. 1996)
- Neumann, J.Von dan O. Morgenstern. 1994. Theory of Games and Economic Behaviour. Princeton New Jersey: Princeton University Press
- Nugraha, R., Exridores, E dan Sopriyadi, H 2015, Penerapan Algoritma Fisher-Yates pada Aplikasi The Lost Insect untuk Pengenalan Jenis Serangga Berbasis Unity 3D, STMIK Global Informatika MDP, Palembang
- O'Connor, Derek. 2014. A Historical Note on Shuffle Algorithms, <https://www.academia.edu>, Diakses 5 Juli 2017
- Pavel, Micka. Fisher-Yates-Shuffle Algorithm: Founder and administrator of web encyclopedia Algoritmy.net [Online]. Diakses 5 Juli 2017
- Priangga, B., Supriyanto dan Yoannita 2014, Penerapan Algoritme Fisher-Yates pada Edugame Guess Calculation Berbasis Android, STMIK Global Informatika MDP, Palembang
- Putra, Y.S, M. Aziz Muslim, Agus Naba. "Game Chicken Roll dengan menggunakan Metode Forward Chaining". Jurnal EECCIS Vol. 7 No. 1, 2013
- Riskiawati, Hendrik A. 2005. Penerapan Sistem Pakar Forward Chaining Berbasis Aturan Pada Pengawasan Status Penerbangan. Jurnal Integral. Vol.10.no 3
- Rolling, Andrew dan Ernest Adams. 2003. Game Design. USA: New Riders Publishing
- S. semiwiyoto, Penanganan dan Pemanfaatan Sampah, (Jakarta: Yayasan Idayu, 1983)
- Salen, Katie dan Zimmerman, Eric. 2005. The Game Design Reader: A Rules of Play Anthology. The MIT Press. <http://mitpress.mit.edu/books/game-design-reader>, Diakses 5 Juli 2017
- Saputra Ari Nopian, Alifullah, Willy. 2016. Penerapan Algoritma Fisher Yates pada Edugame Gess Animals Food Groups
- Sri Subekti, Pengelolaan Sampah Rumah Tangga 3R Berbasis Masyarakat Pendahuluan, Available at: <http://www.scribd.com/doc/19229978/tulisan-bektihadini> Diakses 5 Juli 2017
- Vinaysingh, Shuffle an array by modern Fisher-Yates method <http://www.vinaysingh.info/fisher-yates-shuffle/> diakses 5 Juli 2017

Yudhi Kartikawan, Pengelolaan Persampahan, (Yogyakarta: Jurnal Lingkungan Hidup, 200)

Zainuddin. 2011. Merancang dan Membuat Program Permainan Quad dengan Metode Forward Chaining. Skripsi STIKOM Surabaya

