

RANCANG BANGUN GAME EDUKASI PENGENALAN FAUNA DI INDONESIA

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Noor Aziza Arifani
NIM: 115060801111058



INFORMATIKA / ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

RANCANG BANGUN GAME EDUKASI PENGENALAN FAUNA DI INDONESIA

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Noor Aziza Arifani
NIM: 115060801111058

Skripsi ini telah diuji dan dinyatakan lulus pada
25 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Erig M. Adams J, S.T., M.Kom
NIP: 198504102012121001

Wibisono Sukmo W, S.T., M.T.
NIK: 2010088204041001

Mengetahui
Ketua Program Studi Informatika/Ilmu Komputer

Drs. Marji, M.T.
NIP: 196708011992031001

PERNYATAAN ORISINALITAS

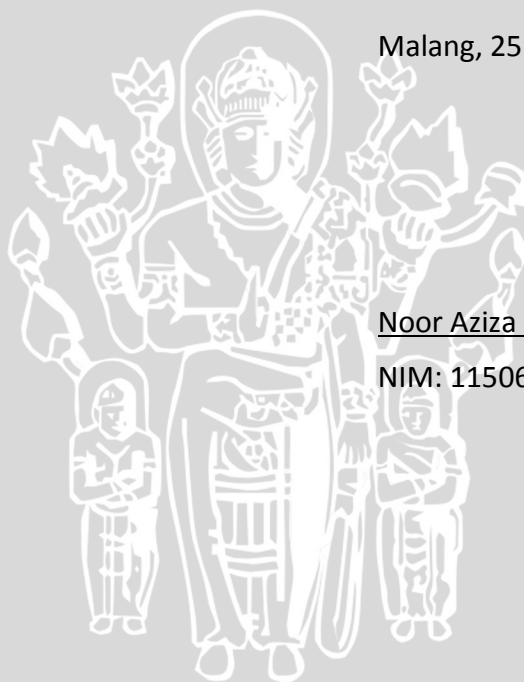
Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 25 Januari 2016

Noor Aziza Arifani

NIM: 115060801111058



KATA PENGANTAR

Puji syukur kehadiran Allah SWT, karena atas limpahan rahmat, dan hidayah-Nya, penulis dapat menyelesaikan Skripsi yang berjudul “Rancang Bangun *Game* Edukasi Pengenalan Fauna di Indonesia”. Skripsi ini diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana S-1 Fakultas Ilmu Komputer Universitas Brawijaya Malang.

Penyusunan skripsi ini tidak lepas dari bantuan semua pihak yang terus memberikan semangat, kritik, saran dan bimbingan serta doa. Penulis menyampaikan penghargaan dan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Eriq M. Adams J., S.T., M.Kom. selaku dosen pembimbing I dan Bapak Wibisono Sukmo W., S.T., M.T. selaku dosen pembimbing II.
2. Bapak Drs. Marji, M.T. dan Bapak Issa Arwani, S.Kom., M.Sc. selaku Ketua dan Sekretaris Program Studi Informatika.
3. Bapak Drs. Mohammad Hari, M.Si. dan Ibu Sri Sutjihandajani, S.Pd. selaku orang tua penulis, Hardani Sucireswari dan Kamila Qurrota A’yun selaku saudara kandung penulis, yang selalu memberikan doa dan dukungan dalam pengerjaan skripsi.
4. Teman-teman seperjuangan dalam mengerjakan skripsi, Ayu Leonitami, Eureka Kurnia Sari, S.Kom., Indrahidayati Nurrohmah, Fadel Trivandi Dipantara, S.Kom.
5. Kresna Selvian yang memberikan dukungan dan membantu dalam pengerjaan skripsi ini.
6. Teman-teman seperjuangan masa kuliah SCANF yang selalu bersama baik suka maupun duka.
7. Dan semua pihak yang tidak dapat disebutkan satu persatu, yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya skripsi ini.

Semoga skripsi ini dapat bermanfaat bagi pembaca. Penulis sadar bahwa masih banyak kekurangan dalam penyusunan skripsi ini. Dengan segala kerendahan hati, penulis mengharapkan kritik dan saran yang bersifat membangun dari pembaca.

Malang, 25 Januari 2016

Penulis

noorazizaarifani@gmail.com

ABSTRAK

Noor Aziza Arifani, 2016: Rancang Bangun Game Edukasi Pengenalan Fauna di Indonesia.

Dosen Pembimbing: Eriq M. Adams J., S.T., M.Kom. dan Wibisono Sukmo W., S.T., M.T.

Indonesia memiliki keanekaragaman fauna yang sangat beragam. Saat ini, Daftar Merah IUCN mencatat sekitar 23% spesies mamalia dan 12% jenis burung di Indonesia termasuk kategori terancam punah. Hal tersebut menandakan pentingnya proses pengenalan fauna di Indonesia untuk masyarakat Indonesia. Permainan adalah salah satu cara untuk dapat menemukan solusi dari suatu masalah dengan pendekatan yang menyenangkan. *Game* yang dibuat berfungsi untuk mengenalkan pengguna tentang persebaran fauna di Indonesia. Target pengguna adalah semua umur. Metode yang digunakan untuk pengembangan game ini adalah *iterative with rapid prototyping*. Proses perancangan menghasilkan *game* dengan *gameplay turn-based strategy*. Proses implementasi dilakukan menggunakan *game engine* Unity dan bahasa pemrograman C#. Pengujian menunjukkan hasil implementasi telah memenuhi kebutuhan fungsional *game* ini. Pengujian pada pengguna menunjukkan bahwa *game* dapat diterima dengan baik. *Game* yang dibangun terbilang menyenangkan dan dapat mengenalkan pengguna pada persebaran fauna di Indonesia.

Kata kunci: *game*, edukasi, persebaran fauna di Indonesia

ABSTRACT

Noor Aziza Arifani, 2016: Rancang Bangun Game Edukasi Pengenalan Fauna di Indonesia.

Dosen Pembimbing: Eriq M. Adams J., S.T., M.Kom. dan Wibisono Sukmo W., S.T., M.T.

Indonesia has a very diverse fauna diversity. Currently, the IUCN Red List recorded approximately 23% of mammal species and 12% of bird species in Indonesia are included in the endangered category. This indicates the importance of the introduction of fauna in Indonesia for Indonesian society. Game is a way to find the solution to a problem with a fun approach. The game created is used to introduce the user about the distribution of fauna in Indonesia. Target users are of all ages. The method used for the development of this game is iterative with rapid prototyping. The designing process produces games with turn-based strategy gameplay. The implementation process is done using Unity game engine and C# programming language. Tests showed the results of the implementation have met the functional requirement of this game. Tests on the users indicates that the game can be received well. The gameplay is built fairly fun and can introduce users to the distribution of fauna in Indonesia.

Keyword: game, education, distribution of fauna in Indonesia

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS.....	iii
KATA PENGANTAR	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
DAFTAR LAMPIRAN.....	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah	2
1.3 Tujuan.....	2
1.4 Manfaat	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan.....	2
BAB 2 LANDASAN KEPUSTAKAAN.....	4
2.1 <i>Game</i>	4
2.1.1 <i>Genre Game</i>	4
2.1.2 <i>Game</i> Edukasi	4
2.1.3 Elemen Formal <i>Game</i>	5
2.2 Persebaran Fauna Indonesia	6
2.3 <i>MDA Framework</i>	7
2.4 <i>Paper Prototyping</i>	7
2.5 Unity	7
2.6 Bahasa Pemograman C#.....	8
2.7 Pengujian Perangkat Lunak	8
BAB 3 METODOLOGI	13
3.1 Studi Literatur	13
3.2 Perancangan	13

3.3 Implementasi	15
3.4 Pengujian	15
BAB 4 PERANCANGAN DAN IMPLEMENTASI.....	17
4.1 Perancangan	17
4.1.1 <i>Paper Prototyping</i>	17
4.1.2 Deskripsi <i>Game</i>	27
4.2 Implementasi	31
4.2.1 Batasan Implementasi.....	31
4.2.2 Implementasi <i>Gameplay</i>	31
4.2.3 <i>Quality Assurance</i> dan <i>Playtesting</i>	48
4.3 Pengujian	48
BAB 5 PENGUJIAN	49
5.1 Hasil Pengujian dengan <i>White Box Testing</i>	49
5.1.1 Pengujian Pemilihan Pemain	49
5.1.2 Pengujian Putar Dadu	50
5.1.3 Pengujian Menangkap Hewan.....	52
5.1.4 Pengujian Melepaskan Hewan	53
5.2 Hasil Pengujian dengan <i>Black Box Testing</i>	54
5.2.1 <i>Test Flow Diagram</i>	54
5.2.2 <i>Focus Testing</i>	56
5.3 Analisis Pengujian.....	57
5.3.1 Hasil Analisis <i>White Box Testing</i>	57
5.3.2 Hasil Analisis <i>Black Box Testing</i>	57
5.3.3 Hasil Analisis <i>Focus Testing</i>	57
BAB 6 PENUTUP	59
6.1 Kesimpulan	59
6.2 Saran.....	59
DAFTAR PUSTAKA	60

DAFTAR TABEL

Tabel 4.1 Peraturan <i>paper prototyping</i> iterasi pertama.....	17
Tabel 4.2 Peraturan <i>paper prototyping</i> iterasi kedua	21
Tabel 4.3 Peraturan <i>paper prototyping</i> iterasi ketiga	24
Tabel 4.4 Gambaran <i>game</i> “Nusantara Rescue Zoo”	27
Tabel 4.5 <i>Method</i> Start pada pemilihan pemain.....	31
Tabel 4.6 <i>Method</i> endSelection pada pemilihan pemain	31
Tabel 4.7 <i>Method</i> Start pada awal permainan.....	32
Tabel 4.8 <i>Method</i> rollButton pada putar dadu	33
Tabel 4.9 <i>Method</i> rolled pada putar dadu	34
Tabel 4.10 <i>Method</i> confirmRolled pada putar dadu	36
Tabel 4.11 <i>Method</i> setBait pada menangkap hewan	36
Tabel 4.12 Metode untuk menentukan keberhasilan menangkap hewan.....	37
Tabel 4.13 <i>Method</i> kandangButton pada kandang	38
Tabel 4.14 <i>Method</i> buyKandang pada kandang	38
Tabel 4.15 <i>Method</i> setKandang pada kandang	38
Tabel 4.16 <i>Method</i> confirmBreed pada mengembangbiakkan hewan	40
Tabel 4.17 <i>Method</i> checkBreed pada mengembangbiakkan hewan	40
Tabel 4.18 <i>Method</i> release pada melepaskan hewan	41
Tabel 4.19 <i>Method</i> trade pada menjual kandang	42
Tabel 4.20 <i>Method</i> confirmTrade pada menjual kandang.....	43
Tabel 4.21 Laporan <i>quality assurance</i>	48
Tabel 5.1 Hasil pengujian unit pemilihan pemain	50
Tabel 5.2 Hasil pengujian unit putar dadu	51
Tabel 5.3 Hasil pengujian unit menangkap hewan.....	53
Tabel 5.4 Hasil pengujian unit melepaskan hewan	54
Tabel 5.5 Hasil pengujian dengan TFD	56
Tabel 5.6 Hasil <i>focus testing</i>	56
Tabel 5.7 Analisis hasil <i>focus testing</i>	58

DAFTAR GAMBAR

Gambar 2.1 Notasi <i>Flow Graph</i>	9
Gambar 2.2 <i>Flowchart</i> (A) dan <i>Flow Graph</i> (B)	10
Gambar 2.3 Contoh Flow	11
Gambar 3.1 Diagram alir metode perancangan <i>Iterative</i>	14
Gambar 3.2 Diagram alir metode perancangan <i>Iterative with Rapid Prototyping</i>	14
Gambar 4.1 <i>Paper prototype</i> perancangan iterasi pertama babak 1	18
Gambar 4.2 <i>Paper prototype</i> perancangan iterasi pertama babak 2	19
Gambar 4.3 <i>Playtest</i> perancangan iterasi pertama babak 1	20
Gambar 4.4 <i>Playtest</i> perancangan iterasi pertama babak 2	21
Gambar 4.5 <i>Paper prototype</i> perancangan iterasi kedua	23
Gambar 4.6 <i>Playtest</i> perancangan iterasi kedua	24
Gambar 4.7 <i>Paper prototype</i> perancangan iterasi ketiga	26
Gambar 4.8 <i>Playtest</i> perancangan iterasi ketiga	27
Gambar 4.9 Tampilan halaman awal	44
Gambar 4.10 Tampilan halaman pemilihan pemain	44
Gambar 4.11 Tampilan halaman <i>board</i>	45
Gambar 4.12 Tampilan halaman kandang	46
Gambar 4.13 Tampilan halaman menangkap hewan	46
Gambar 4.14 Tampilan halaman melepaskan hewan	47
Gambar 4.15 Tampilan halaman hasil akhir	47
Gambar 5.1 Pengujian unit pemilihan pemain	49
Gambar 5.2 <i>Flow graph</i> pemilihan pemain	49
Gambar 5.3 Pengujian unit putar dadu	50
Gambar 5.4 <i>Flow graph</i> putar dadu	51
Gambar 5.5 Pengujian unit menangkap hewan	52
Gambar 5.6 <i>Flow graph</i> menangkap hewan	52
Gambar 5.7 Pengujian unit melepaskan hewan	53
Gambar 5.8 <i>Flow graph</i> melepaskan hewan	53
Gambar 5.9 Pengujian <i>gameplay</i> menggunakan TFD	55

DAFTAR LAMPIRAN

LAMPIRAN A KUESIONER PRA PENELITIAN	61
LAMPIRAN B KUESIONER PENGUJIAN <i>FOCUS TESTING</i>	62



BAB 1 PENDAHULUAN

1.1 Latar belakang

Indonesia memiliki keanekaragaman fauna yang sangat beragam meskipun daratan Indonesia hanya 1,3% dari permukaan Bumi. Indonesia merupakan daerah asal dari 12% mamalia, 16% reptil dan amfibi, dan 17% spesies burung di seluruh dunia. Spesies-spesies fauna tersebut memiliki peran penting untuk menjaga rantai makanan serta menjaga keseimbangan ekosistem dan regenerasi habitat. Selain itu, terdapat juga spesies yang memiliki nilai budaya yang berarti penting untuk masyarakat Indonesia. Saat ini, Daftar Merah IUCN mencatat sekitar 23% spesies mamalia dan 12% jenis burung di Indonesia termasuk kategori terancam punah. Secara umum ancaman utama kepunahan spesies disebabkan oleh aktivitas manusia (WWF Indonesia, 2015). Hal tersebut menandakan pentingnya proses pengenalan fauna di Indonesia untuk masyarakat Indonesia.

Proses pengenalan fauna merupakan hal yang sangat penting dilakukan, khususnya di Indonesia. Saat ini kesadaran masyarakat akan pentingnya pelestarian fauna masih terbilang rendah. Hal ini terbukti dengan masih adanya kerusakan habitat dan eksploitasi secara berlebihan terhadap habitat asli fauna (ProFauna, 2012). Pengenalan fauna dapat berfungsi untuk meningkatkan kepedulian masyarakat Indonesia tentang keanekaragaman fauna yang ada di Indonesia. Pengenalan ini dapat dilakukan dengan proses pembelajaran. Proses pembelajaran dengan melakukan metode konvensional cenderung membosankan. Selain itu, materi kurang dapat dipahami (Hidayat & Sriyanto, 2015).

Metode-metode inovatif telah banyak dikembangkan untuk meningkatkan minat belajar. Teknologi pembelajaran, khususnya yang berbasis digital, diharapkan dapat meningkatkan kualitas belajar sehingga hasil belajar juga semakin baik (Darmawan, 2012). Permainan adalah salah satu cara untuk dapat menemukan solusi dari suatu masalah dengan pendekatan yang menyenangkan (Keese, 2011). Jika metode pembelajaran yang digunakan menarik, maka isi dari materi akan lebih mudah dipahami.

Berdasarkan penjelasan tersebut, penelitian akan dilakukan untuk mengembangkan *game* edukasi pengenalan fauna di Indonesia yang ditujukan untuk masyarakat umum. Hasil dari implementasi *game* diharapkan menyenangkan untuk dimainkan serta memiliki fungsi edukatif. *Game* ini diharapkan dapat mengenalkan pengguna tentang persebaran fauna di Indonesia.

1.2 Rumusan masalah

Berdasarkan uraian pada latar belakang masalah, maka rumusan masalah yang akan diselesaikan di dalam penelitian ini adalah:

1. Bagaimana *gameplay* dari *game* edukasi pengenalan fauna di Indonesia?
2. Bagaimana implementasi *game* edukasi pengenalan fauna di Indonesia berdasarkan *gameplay* yang telah dirancang?
3. Apakah seluruh fungsionalitas dari *game* edukasi pengenalan fauna di Indonesia sudah terpenuhi?
4. Apakah *game* edukasi pengenalan fauna di Indonesia menarik dan dapat mengenalkan pengguna tentang fauna yang ada di Indonesia?

1.3 Tujuan

Tujuan dari penelitian ini adalah merancang, mengimplementasi, dan memenuhi kebutuhan fungsionalitas *game* edukasi pengenalan fauna di Indonesia, serta mengetahui penerimaan pengguna terhadap *game* edukasi pengenalan fauna di Indonesia.

1.4 Manfaat

Manfaat yang diperoleh dari penelitian ini adalah:

1. Bagi pengembang *game*, permainan ini diharapkan dapat menambah referensi tentang *game* edukasi.
2. Bagi pemain, permainan ini diharapkan dapat menghibur sekaligus mengenalkan tentang fauna di Indonesia.
3. Bagi penulis, permainan ini diharapkan dapat menambah wawasan tentang aspek-aspek pengembangan *game*, khususnya *game* edukasi.

1.5 Batasan masalah

Untuk mendapatkan hasil pembahasan yang sesuai dengan yang diharapkan, pembatasan masalah yang diperlukan pada penelitian ini yaitu:

1. Terdapat 25 jenis fauna yang diperkenalkan pada permainan ini, yaitu elang jawa, kancil, babirusa, kasuari, banteng, harimau, bekantan, jalak bali, rangkong, kijang, anoa, badak, cenderawasih, walabi, tapir, nuri, tarsius, kuskus, merak, komodo, orang utan, kakatua, gajah, maleo, dan buaya.
2. Pengenalan fauna meliputi nama hewan, bentuk hewan, dan area asal hewan.

1.6 Sistematika pembahasan

Sistematika pembahasan secara garis besar meliputi beberapa bab, yaitu sebagai berikut:

Bab 1 : Pendahuluan

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan.

Bab 2 : Landasan Kepustakaan

Bab ini berisi tentang teori-teori dan referensi yang mendukung dalam perancangan, implementasi, dan pengujian dari *game* yang dibangun.

Bab 3 : Metodologi

Bab ini membahas tentang metode yang digunakan dalam penelitian yang meliputi studi literatur, metode perancangan, metode implementasi, metode pengujian serta pengambilan kesimpulan dan saran.

Bab 4 : Perancangan dan Implementasi

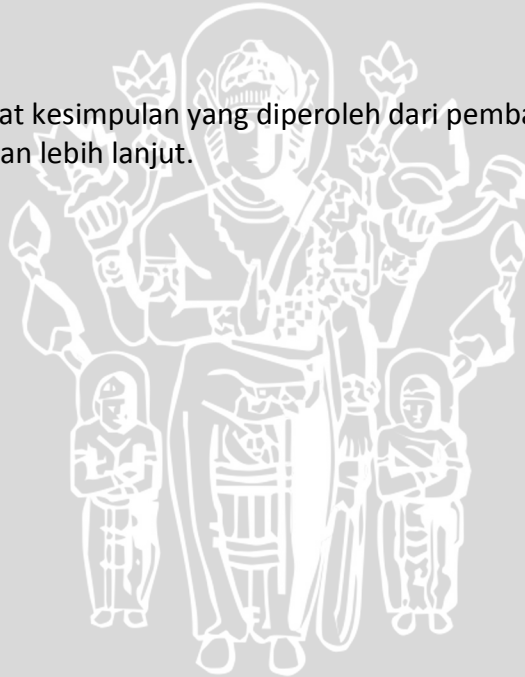
Bab ini memuat hasil perancangan dan implementasi dari *game* edukasi pengenalan fauna di Indonesia.

Bab 5 : Pengujian

Bab ini memuat hasil pengujian dan analisis dari hasil implementasi yang telah dibuat.

Bab 6 : Penutup

Bab ini memuat kesimpulan yang diperoleh dari pembahasan serta saran untuk penelitian lebih lanjut.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Game

Game adalah aktivitas yang memiliki aturan. *Game* sering melibatkan konflik, baik dengan sesama pemain, dengan sistem *game* itu sendiri, maupun dengan keberuntungan. Sebagian besar *game* memiliki tujuan, tetapi tidak semuanya. Sebagian besar *game* memiliki kondisi awal dan kondisi akhir. Sebagian besar *game* mengharuskan pemain untuk mengambil keputusan. *Video game* adalah *game* yang menggunakan layar digital untuk menampilkan video atau animasi (Brathwaite & Schreiber, 2009).

2.1.1 Genre Game

Game dibagi menjadi beberapa *genre*. *Genre game* mendeskripsikan tentang gaya *gameplay* dari suatu *game* (Rogers, 2010).

1. *Action*, adalah *game* yang memerlukan koordinasi tangan atau mata dalam permainan.
2. *Shooter*, pemain bertindak sebagai penembak dan bertugas untuk menembakkan proyektil ke musuh.
3. *Adventure*, berfokus pada penyelesaian teka-teki, serta mengumpulkan barang dan manajemen inventori.
4. *Construction/management*, *genre* ini mengharuskan pemain membangun dan mengembangkan lokasi dengan sumber daya yang terbatas.
5. *Life simulation*, mirip seperti *genre management*, tetapi berfokus pada membangun dan menjaga hubungan pada simulasi kehidupan.
6. *Music/rhythm*, pemain berusaha untuk mencocokkan ritme atau irama untuk mendapatkan nilai.
7. *Party*, dirancang untuk beberapa pemain dan berbasis kompetisi. *Genre* ini kebanyakan berformat *minigame*.
8. *Puzzle*, menggunakan logika untuk menyelesaikan suatu pola.
9. *Sports*, berdasarkan pada kompetisi atletik, baik secara tradisional (mirip dengan olahraga asli), maupun secara ekstrim (menggunakan modifikasi yang tidak masuk akal).
10. *Strategy*, mengharuskan pemain untuk berpikir dan merencanakan sesuatu.
11. *Vehicle simulation*, pemain menjalankan simulasi mengendarai suatu kendaraan, bisa saja berupa mobil atau bahkan pesawat ulang alik. *Game* ber-*genre* ini dibuat untuk mewujudkan lingkungan yang semirip mungkin dengan aslinya.

2.1.2 Game Edukasi

Game edukasi adalah *game* yang dirancang untuk mengajarkan manusia tentang suatu materi sehingga menjadi keterampilan manusia tersebut. Alat pendidikan ini menjadi populer karena banyak pihak yang menganggap pembelajaran dengan permainan memiliki banyak keuntungan. *Game* adalah permainan interaktif yang mengajarkan tujuan, aturan, adaptasi, penyelesaian

masalah, interaksi, dan semua itu dikemas dalam suatu cerita. Pendekatan ini memenuhi kebutuhan pokok dalam pembelajaran, yaitu dengan menyediakan kegembiraan, keterlibatan dalam *game*, struktur, motivasi, kepuasan diri, adrenalin, kreativitas, interaksi sosial dan emosi (Keesee, 2011).

2.1.3 Elemen Formal *Game*

Elemen formal *game* adalah bagian terkecil dari *game* yang dapat dikaji secara individu. Berikut adalah beberapa bagian dari *game* yang dapat diperhitungkan perancang dalam merancang sebuah *game* (Schreiber, 2009).

1. *Players*

Elemen ini menjelaskan tentang struktur pemain dalam *game*. Struktur pemain meliputi jumlah pemain, apakah pemain dapat meninggalkan *game* saat masih berlangsung dan bagaimana hal tersebut mempengaruhi jalannya *game*, serta bagaimana hubungan antar pemain, apakah individu atau dalam tim.

2. *Objectives*

Elemen ini menjelaskan tentang tujuan *game* yang harus dicapai oleh pemain. Hal ini adalah hal dasar yang penting dalam merancang suatu *game*. Setelah *objectives* ditentukan, elemen formal lain akan lebih mudah didefinisikan.

3. *Rules*

Elemen ini memuat aturan dari *game*. Terdapat tiga kategori *rules* yaitu *setup* (aturan yang ada saat permainan baru dimulai), *progression of play* (aturan tentang apa yang terjadi saat *game* dimainkan), dan *resolution* (hal yang menyebabkan *game* berakhir dan hasil akhir dari *game* tersebut). Aturan tersebut dapat bergantung dari pengambilan keputusan pemain atau dapat juga ditentukan secara otomatis dari sistem *game*.

4. *Resources and Resource Management*

Elemen ini mencakup sumber daya (*resource*) dan manajemen sumber daya. *Resource* adalah hal-hal yang dimiliki seorang pemain. *Resource* dapat berupa barang-barang, nyawa pemain, uang, wilayah kekuasaan, senjata, waktu, informasi, dan sebagainya. Perancang *game* harus mendefinisikan bagaimana cara pemain dapat memanfaatkan *resource* tersebut.

5. *Game State*

Elemen ini mencakup keadaan objek-objek permainan dalam suatu waktu tertentu. Dalam *video game*, *game state* diperlukan untuk mencatat data apa saja yang diperlukan komputer untuk menjalankan permainan.

6. *Information*

Elemen ini mendefinisikan tentang seberapa banyak informasi *game state* yang dapat diketahui oleh pemain. Perubahan kuantitas informasi yang

tersedia untuk pemain dapat memberi efek yang drastis pada *game*, meskipun elemen formal lain tetap sama.

7. Sequencing

Elemen ini menjelaskan cara pemain melakukan aksi. Elemen ini mencakup urutan pemain dalam melakukan aksi dan urutan aksi-aksi yang dapat dilakukan seorang pemain. Aksi tersebut juga dapat diberikan batasan waktu sehingga menambah tekanan untuk pemain.

8. Player Interaction

Elemen ini menjelaskan interaksi antar sesama pemain. Hal ini merupakan hal yang penting namun sering diabaikan. Perancang *game* dapat mendefinisikan cara pemain berinteraksi dengan pemain lain dan bagaimana seorang pemain dapat mempengaruhi pemain lain.

9. Theme

Elemen ini mencakup tema cerita dari *game* dan latar. Elemen ini merupakan bagian dari *game* tetapi tidak mempengaruhi *gameplay* secara langsung. *Theme* dapat menimbulkan koneksi emosional pemain terhadap *game* dan juga dapat membuat *game* lebih mudah untuk dimainkan.

10. Games as Systems

Jika salah satu elemen formal diubah, maka *game* bisa menjadi sangat berbeda. Tiap elemen formal memberikan dampak yang mendalam pada pengalaman pemain. Elemen-elemen tersebut berhubungan satu dengan yang lain, mengubah suatu elemen dapat mempengaruhi elemen lain. Karena elemen-elemen tersebut saling berhubungan, suatu *game* dapat disebut sebagai sistem.

2.2 Persebaran Fauna Indonesia

Indonesia merupakan negara yang kaya akan keanekaragaman fauna. Hal ini disebabkan karena letak geologis Indonesia. Pembagian fauna Indonesia menurut Alfred Weber dan Wallace ada tiga, yaitu fauna Asiatis, fauna Australis, dan fauna Peralihan (Tim Abdi Guru, 2007).

1. Fauna Asiatis

Fauna Asiatis merupakan fauna yang berasal dari Indonesia bagian barat. Wilayah barat ini meliputi Pulau Jawa, Sumatera, Kalimantan, Makassar, dan sebelah barat Selat Lombok. Beberapa hewan yang termasuk fauna Asiatis yaitu harimau, gajah, badak, orang utan, beruang, tapir, kijang, banteng, siamang, trenggiling, kancil, burung jalak bali, burung merak, buaya, dan ikan pesut.

2. Fauna Australis

Fauna Australis merupakan fauna yang berasal dari Indonesia bagian timur. Wilayah timur ini meliputi Papua, Kepulauan Aru, serta pulau-pulau kecil di

sekitarnya. Beberapa hewan yang termasuk fauna Australis yaitu kuskus, burung kasuari, burung cenderawasih, dan hewan berkantung.

3. Fauna Peralihan

Fauna Peralihan adalah fauna yang berasal dari daerah peralihan di antara bagian barat dan bagian timur. Wilayah peralihan meliputi Pulau Sulawesi, Kepulauan Maluku, dan Kepulauan Nusa Tenggara. Beberapa hewan yang termasuk fauna Peralihan yaitu burung kakaktua, burung maleo, anoa, biawak, babirusa, dan komodo.

2.3 MDA Framework

Terdapat tiga komponen dalam *game*, yaitu *Mechanics*, *Dynamics*, dan *Aesthetics* (Schreiber, 2009).

1. *Mechanics* adalah kata lain dari aturan (*rules*) pada sebuah *game*. Ini adalah batasan-batasan dalam pengoperasian *game*. *Mechanics* mendefinisikan tentang kondisi awal *game* (*setup*), aksi pemain, *game state*, dan kondisi akhir *game* (*resolution*).
2. *Dynamics* mendefinisikan tentang jalannya permainan saat aturan telah ditetapkan. *Dynamics* mencakup strategi yang timbul berdasarkan aturan dan interaksi antara sesama pemain.
3. *Aesthetics* bukan merujuk pada elemen visual dari *game*, melainkan mengenai pengalaman pemain saat memainkan *game*. Apakah pemain menikmati *game* yang dimainkan? Apakah *game* melibatkan pemain secara emosional atau intelektual? *Aesthetics* adalah efek dari *dynamics* yang dirasakan oleh pemain.

2.4 Paper Prototyping

Paper prototyping adalah metode membuat *prototype* sebuah *game* secara fisik sehingga dapat dimainkan tanpa menggunakan komputer. *Prototype* digital lebih memakan waktu dan mahal dibandingkan *paper prototype* (Schreiber, 2009). Berikut adalah keunggulan *paper prototyping*:

1. Biayanya rendah dan hampir semua sistem dapat disimulasikan pada secarik kertas dengan menggunakan pensil.
2. Pembuatannya cepat tidak rumit seperti melibatkan logika pemrograman.
3. *Prototype* mudah diubah atau dihapus.
4. Kertas dapat digunakan untuk memodelkan hampir semua sistem *gameplay*.
5. Dengan membuat sesuatu yang dapat dimainkan, maka secara otomatis perancangan sistem juga akan dilakukan terlebih dahulu.

2.5 Unity

Game Engine adalah perangkat lunak yang digunakan sebagai sarana pembuatan *game*. Dalam *game engine* biasanya terdapat fungsi-fungsi yang sering digunakan dalam pembuatan *game*, misalnya fungsi *rendering*, pemanggilan

suara, *network*, dan pembuatan efek. Dengan adanya *game engine*, *developer* dapat lebih mudah membangun sebuah *game*.

Unity adalah salah satu *game engine*. Unity dibangun oleh Unity Technologies yang berfokus untuk membuat perangkat lunak yang dapat digunakan oleh semua orang, khususnya dalam pembuatan *game*. Unity memungkinkan *developer* untuk membuat *game* dengan mudah dan cepat secara gratis. Hasil *game* yang dibuat dengan Unity dapat dipasang ke berbagai *platform* seperti Windows, Mac, Android, iOS, PS3, Wii, dan sebagainya (Roedavan, 2014).

2.6 Bahasa Pemrograman C#

Bahasa pemrograman C# adalah salah satu bahasa resmi untuk pemrograman *framework* Microsoft .NET. Tujuan bahasa C# adalah sebuah bahasa pemrograman yang sederhana, modern, *general purpose*, dan berorientasi objek. C# disebutkan sebagai pengembangan dari bahasa C++, yang berasal dari rumpun bahasa C. Bahasa C# dianggap sebagai bahasa modern yang berkonsep OOP, tetapi tidak menggunakan konsep yang susah dipahami seperti *pointer*, *multiple inheritance*, dan alokasi sumber daya. Pengelolaan sumber daya pada bahasa C# diatur secara otomatis.

Bahasa C# merupakan bahasa pemrograman berbasis komponen, artinya objek-objek dikembangkan dari sekumpulan komponen dan setiap aksinya berdasarkan objek dari komponen tersebut. Komponen adalah dasar dari pengembangan sistem. Komponen tersebut dapat berupa properti, metode, dan *event*. Komponen-komponen tersebut berada pada *runtime* khusus pada .NET Framework (Ferdiana, 2006).

2.7 Pengujian Perangkat Lunak

1. Pengujian *White Box*

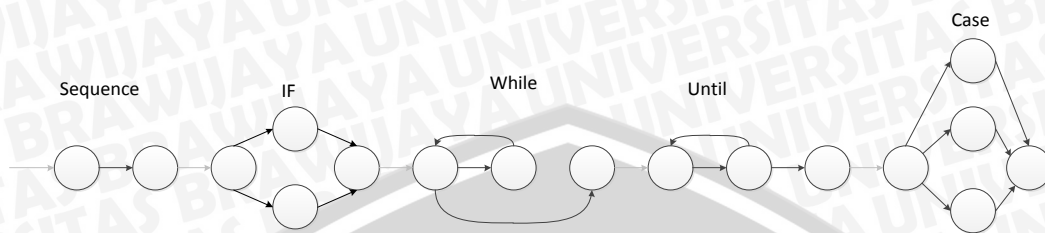
Dalam pengujian *white box*, penguji dapat mengakses *source code* dari program. Penguji dapat mencoba tiap segmen dari *source code* dan mengamati hasil yang didapat dari kombinasi input yang berbeda. Pengujian dengan metode ini terbilang sulit karena hampir tidak mungkin untuk memperhitungkan kompleksitas *feedback loop* dari pemain.

Pada pengujian *white box*, penguji mengeksekusi suatu modul dengan mencoba semua kemungkinan yang dapat diterima modul tersebut. *Input* dalam pengujian ditentukan oleh tipe dan nilai data yang dapat diterima oleh *source code*. Hasil diamati dengan melihat hasil *return* dari modul tersebut, variabel global yang dipengaruhi oleh modul tersebut, dan juga variabel lokal yang ada di dalam modul (Schultz, et al., 2005).

Basis path testing adalah metode pengujian *white box* yang diusulkan pertama kali oleh Tom McCabe. Metode *basis path* memungkinkan perancang *test case* mengambil ukuran kompleksitas logika dari desain prosedural dan menggunakan ukuran ini sebagai acuan dalam mendefinisikan basis set. *Test case* diperoleh

untuk mencoba apakah basis set menjalankan setiap *statement* dalam program paling tidak sekali (Pressman, 2001).

1. Notasi Flow Graph



Gambar 2.1 Notasi Flow Graph

Sumber: (Pressman, 2001)

Flow graph menggambarkan *logical control flow* menggunakan notasi yang diilustrasikan pada Gambar 2.1. Setiap struktur memiliki simbol *flow graph* tersendiri.

2. Cyclomatic Complexity

Cyclomatic complexity atau kompleksitas siklomatik ada satuan ukuran perangkat lunak yang menunjukkan ukuran kuantitatif kompleksitas logika dari sebuah program. Dalam konteks metode *basis path testing*, nilai untuk *cyclomatic complexity* menunjukkan nomor dari jalur independen dalam *basis set* sebuah program. Jalur independen adalah semua jalur yang ada dalam program yang paling tidak berupa satu set pemrosesan *statement* atau sebuah kondisi baru.

Perbedaan antara *flowchart* dan *flow graph* bisa dilihat pada Gambar 2.2. *Cyclomatic complexity* memiliki fondasi yang kuat pada teori *graph* dan menyediakan satuan ukuran perangkat lunak yang sangat berguna. Tingkat kompleksitas bisa dihitung dengan salah satu dari tiga cara yang ada.

1. Jumlah *region* dari *flow graph*.
2. *Cyclomatic complexity*, $V(G)$, untuk sebuah *flow graph* ditentukan dengan:

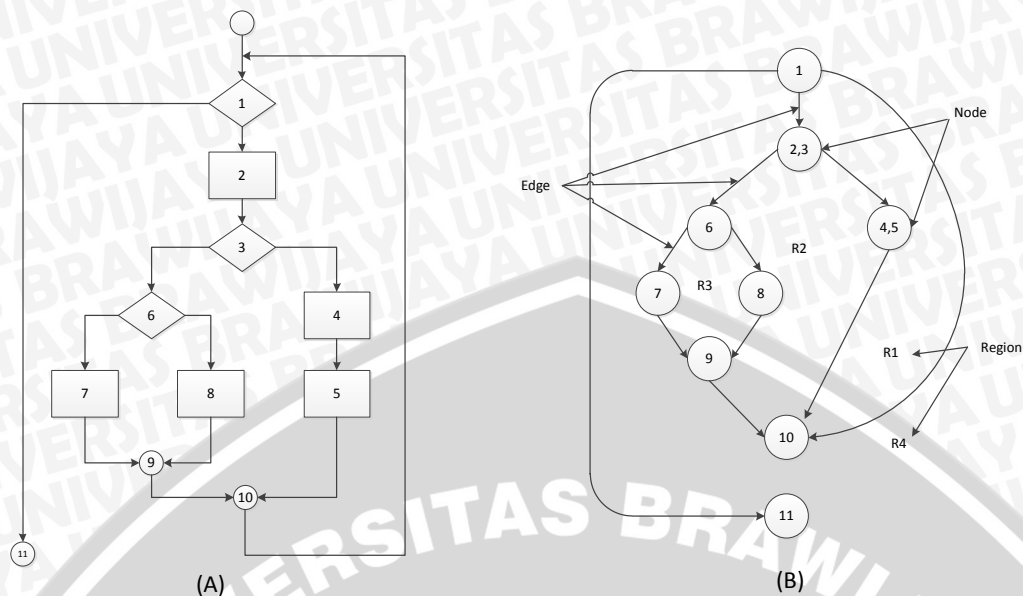
$$V(G) = E - N + 2 \dots\dots\dots(1)$$

Dimana E adalah jumlah *edge*, dan N adalah jumlah *node* dalam sebuah *flow graph*.

3. *Cyclomatic complexity*, $V(G)$, untuk sebuah *flow graph* juga dapat ditentukan dengan cara:

$$V(G) = P + 1 \dots\dots\dots(2)$$

Dimana P adalah *predicate nodes* yang ada pada *flow graph*.



Gambar 2.2 Flowchart (A) dan Flow Graph (B)

Sumber: (Pressman, 2001)

2. Pengujian Black Box

Pengujian *black box* dilakukan di luar aplikasi dan tidak memperhatikan *source code* dari aplikasi. Penguji *game* menguji kelemahan *game* dari *input* yang dapat digunakan pemain pada umumnya. Pengujian *black box* ini dianggap sebagai pengujian yang efektif untuk menguji sistem atau modul yang sangat kompleks.

Input dalam permainan dapat berupa tombol, audio, video, paket dari jaringan, atau data yang tersimpan. Setelah menerima *input*, sistem *game* akan memprosesnya dan mengeluarkan *output*. *Output* tersebut dapat berupa video, audio, getaran, dan data yang akan disimpan.

Input dalam *game* bersifat *feedback loop*, di mana setelah menerima *input* *game* akan memberikan *output*, kemudian pemain memberikan *input* lagi dan begitu seterusnya. Elemen menyenangkan dalam *game* akan terjadi jika *feedback* dari pemain tidak dapat ditebak. Penguji *game* harus bertindak layaknya pemain dan memberikan *feedback* yang tidak terduga sehingga dapat menemukan celah kelemahan *game* (Schultz, et al., 2005).

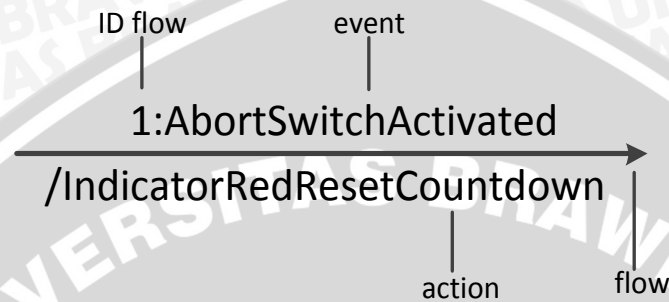
Test flow diagram (TFD) adalah sebuah model grafis yang mewakili perilaku permainan dari perspektif pemain (Schultz, et al., 2005). Proses pengujian dilakukan dengan menelusuri diagram untuk menguji permainan baik secara akrab maupun dengan cara yang tidak terduga. Karena ditampilkan secara grafis, *test flow diagram* membuat penguji, pengembang, dan produser dapat meninjau, menganalisis, dan memberi umpan balik dengan mudah pada rancangan pengujian.

TFD memberikan pendekatan formal untuk desain pengujian yang mempromosikan modularitas dan kelengkapan. Sebuah TFD dibuat dari susunan berbagai komponen gambar. Komponen tersebut digambar, diberi tulisan, dan

dihubungkan sesuai dengan aturan tertentu. Dengan mengikuti aturan tersebut membuat pengujian akan lebih mudah dimengerti dan digunakan kembali untuk proyek berikutnya. Berikut ini adalah elemen-elemen dari TFD.

1. Flow

Flow adalah sebuah garis yang menghubungkan suatu *state* ke *state* yang lain, *flow* dilengkapi dengan sebuah panah yang menunjukkan arah *flow*, dan juga sebuah nomor ID. Contoh *flow* dapat dilihat pada gambar 2.3 dibawah.



Gambar 2.3 Contoh Flow

Sumber: (Schultz, et al., 2005)

2. Event

Event adalah sebuah operasi yang dilakukan oleh pemain, alat permainan, jaringan pada permainan *multiplayer*, atau mekanisme internal yang ada dalam permainan.

3. Action

Action merupakan suatu kejadian dalam permainan yang terjadi ketika ada transisi dari sebuah *event*. Pada umumnya *action* pada permainan dapat dirasakan oleh pemain, seperti misalnya efek suara, efek visual, *controller* permainan yang bergetar, informasi teks pada permainan, dan lain-lain.

4. State

State mewakili perilaku permainan yang tetap dan terus berjalan. Selama *state* tidak dihentikan, maka perilaku yang sama akan terus dijalankan. Setiap kali kembali ke sebuah *state*, perilaku yang ada dalam *state* akan terdeteksi. Sebuah *state* digambarkan dengan bentuk lingkaran dengan nama yang unik di dalamnya.

5. Primitive

Primitive adalah sebutan untuk *Event*, *action*, dan *state* juga. *Primitive* dapat didefinisikan sebagai rincian perilaku yang ditampilkan pada diagram TFD. Penjabaran *primitive* dapat membentuk sebuah “kamus data” pada TFD. Penjabaran tersebut dapat berbentuk teks (misalnya bahasa Indonesia), atau bahasa pemrograman (misalnya C#).

6. Terminator

Terminator kotak khusus ditempatkan pada TFD untuk menunjukkan di mana pengujian akan dimulai dan di mana pengujian akan berakhir. Hanya terdapat dua *terminator* dalam sebuah TFD, yaitu *IN* yang pada umumnya hanya memiliki sebuah *flow* yang mengarah ke sebuah *state*, dan *OUT*, yang dapat memiliki lebih dari satu *flow* dan berasal dari satu *state* atau lebih.

3. Focus Testing

Pada *focus testing*, desainer mengumpulkan pemain yang termasuk dalam target pengguna untuk mengetahui seberapa berhasilkah *game* memenuhi kebutuhannya. Pengujian jenis ini biasanya digunakan untuk tujuan pemasaran (Schreiber, 2009).

Aspek yang dapat dievaluasi untuk menguji elemen *fun* pada *game* yaitu *game interface*, *game mechanics*, dan *gameplay*. *Game interface* atau antarmuka adalah perangkat yang memungkinkan pemain berinteraksi dengan *game*. *Game mechanics* adalah aturan-aturan yang menyebabkan *game* terbentuk. *Gameplay* adalah proses permainan yang dapat menyebabkan pemain mencapai tujuannya (Federoff, 2002).



BAB 3 METODOLOGI

Bab ini akan menjelaskan tentang langkah-langkah metode penelitian yang digunakan untuk membangun *game* edukasi pengenalan fauna di Indonesia.

3.1 Studi Literatur

Pada tahapan ini penulis melakukan kajian terhadap dasar teori yang mendukung dalam pembuatan skripsi ini. Dasar teori yang digunakan yaitu:

1. *Game*
2. Persebaran Fauna di Indonesia
3. *MDA Framework*
4. *Paper Prototyping*
5. Unity
6. Bahasa Pemrograman C#
7. Pengujian Perangkat Lunak

3.2 Perancangan

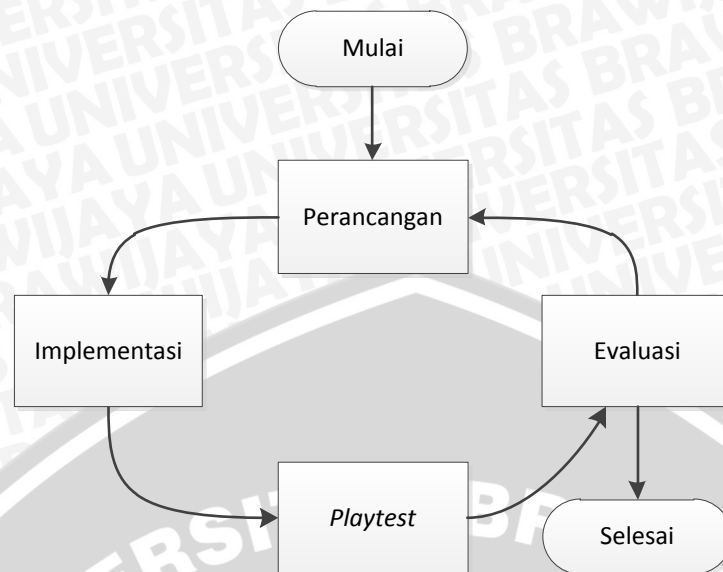
Kerangka kerja yang digunakan untuk membangun *game* edukasi pengenalan fauna di Indonesia ini adalah *MDA Framework*. Sedangkan metode yang digunakan adalah *iterative with rapid prototyping*.

1. MDA Framework

MDA Framework terdiri dari *Mechanics*, *Dynamics*, dan *Aesthetics*. Seorang perancang *game* menciptakan aturan (*Mechanics*) dari game. Aturan yang dibuat tersebut akan menyebabkan permainan berjalan (*Dynamics*). Jalannya permainan tersebut akan menimbulkan pengalaman pemain (*Aesthetics*). Tugas seorang perancang *game* adalah merancang komponen *mechanics* dari game sehingga pemain dapat merasakan pengalaman yang menyenangkan.

2. Iterative with Rapid Prototyping

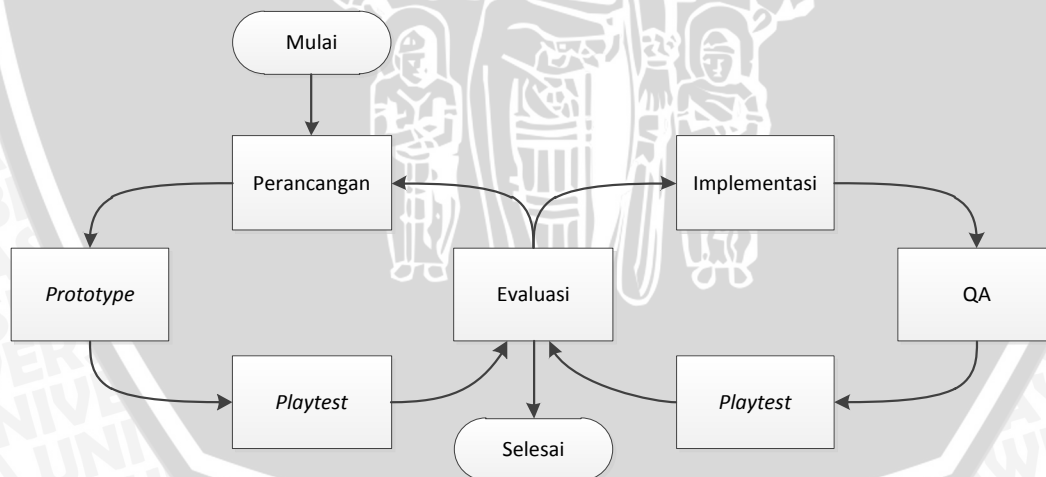
Gambar 3.1 merupakan diagram alir metode *iterative*. Pada metode ini terdapat siklus yang terdiri dari proses perancangan, implementasi, *playtest*, dan evaluasi. Pada proses evaluasi, jika hasilnya sudah cukup baik, maka proses pembuatan selesai. Jika hasilnya belum cukup baik, maka proses bisa diulang kembali ke perancangan. Iterasi dapat dilakukan sampai hasil akhir yang didapatkan sudah baik.



Gambar 3.1 Diagram alir metode perancangan *Iterative*

Sumber: (Schreiber, 2009)

Pada *game* non-digital, metode ini dapat berjalan dengan baik karena prosesnya cepat. Tetapi pada *video game*, proses implementasi memakan waktu cukup lama. Semakin banyak iterasi yang dilakukan, semakin baik pula hasil akhir *game* yang didapatkan. Karena membutuhkan iterasi yang banyak, maka perancang *game* sering melakukan *paper prototyping* terlebih dahulu sebelum implementasi. Hal ini dilakukan untuk menghemat waktu dan biaya. Metode ini dikenal sebagai *iterative with rapid prototyping*.



Gambar 3.2 Diagram alir metode perancangan *Iterative with Rapid Prototyping*

Sumber: (Schreiber, 2009)

Gambar 3.2 menunjukkan proses perancangan menggunakan metode *iterative with rapid prototyping*. Iterasi dapat terjadi pada proses *prototyping* dan proses implementasi. Pada proses evaluasi, apabila hasil yang didapat sudah memuaskan, maka iterasi dapat dihentikan.

3.3 Implementasi

Sasaran lingkungan pengguna untuk *game* yang dihasilkan adalah perangkat Android dengan level API minimal Android 2.3.1 dan resolusi layar WSVGA Landscape (1024x600). Berikut ini adalah kebutuhan perangkat yang diperlukan untuk proses implementasi.

1. Spesifikasi Perangkat Keras

Berikut adalah spesifikasi perangkat keras yang digunakan untuk implementasi *game* pengenalan fauna di Indonesia.

- | | |
|-----------------------|--|
| 1. Prosesor | : Intel(R) Core(TM) i5-3337U CPU @ 1.80GHz |
| 2. RAM | : 4,00 GB |
| 3. Graphic Card | : NVIDIA GeForce 720M |
| 4. Sound Card | : Realtek High Definition Audio |
| 5. Monitor | : 14" HD LED LCD |
| 6. Kapasitas Harddisk | : 500 GB |

2. Spesifikasi Perangkat Lunak

Berikut adalah spesifikasi perangkat lunak yang digunakan untuk implementasi *game* pengenalan fauna di Indonesia.

1. Sistem Operasi Windows 8 Pro 64-bit
2. Unity 5.2.3f1 (64-bit) Personal Edition
3. MonoDevelop
4. CorelDRAW X7 (64-bit)
5. Tiled

3.4 Pengujian

Pengujian dilakukan dengan dua metode yaitu *white box* dan *black box*. Pengujian *white box* dilakukan dengan pengujian unit menggunakan metode *basic path testing*. Pada pengujian *black box* metode yang akan digunakan adalah *test flow diagram* dan *focus testing*.

1. Pengujian White Box

Metode *basic path* digunakan untuk pengujian *white box* karena kompleksitas logika dari suatu rancangan dapat diukur dan digunakan sebagai *basic set*. Pada tiap kasus dilakukan pengujian untuk mengetahui apakah *basic set* dijalankan atau tidak.

2. Pengujian Black Box

Metode yang digunakan pada pengujian *black box* adalah *test flow diagram*. Metode *test flow diagram* dapat menguji *game* pada kasus yang familiar maupun yang tidak terduga. Penguji dapat dengan mudah mengamati dan menganalisis rancangan pengujian karena ditampilkan secara grafis.

Metode *focus testing* atau pengujian terhadap pengguna pada *black box testing* dapat menilai apakah permainan sudah sesuai dengan tujuan pembuatan permainan, yaitu apakah permainan menarik dan dapat mengenalkan pengguna mengenai persebaran fauna di Indonesia.

Objek yang menjadi sasaran dari penelitian ini secara umum adalah semua kalangan, namun secara khusus adalah pengguna berumur tujuh tahun ke atas. Pengujian pada penelitian ini akan dilakukan dengan metode pengumpulan data berupa kuesioner. Hasil data yang diperoleh dari kuesioner *focus testing* akan dianalisis dengan metode statistik deskriptif.



BAB 4 PERANCANGAN DAN IMPLEMENTASI

4.1 Perancangan

Perancangan dilakukan dengan melakukan siklus *paper prototyping* yang terdiri dari perancangan, *prototype*, *playtest*, dan evaluasi. Setelah mendapat hasil *prototype* yang baik, langkah selanjutnya adalah menentukan deskripsi *game*.

4.1.1 Paper Prototyping

Proses *paper prototyping* untuk mendapatkan hasil *gameplay* yang baik pada penelitian ini dilakukan sebanyak tiga kali iterasi. Tiap iterasi terdiri dari perancangan, *prototype*, *playtest*, dan evaluasi.

4.1.1.1 Iterasi Pertama

1. Perancangan

Tabel 4.1 menunjukkan peraturan untuk *paper prototyping* iterasi pertama.

Tabel 4.1 Peraturan *paper prototyping* iterasi pertama

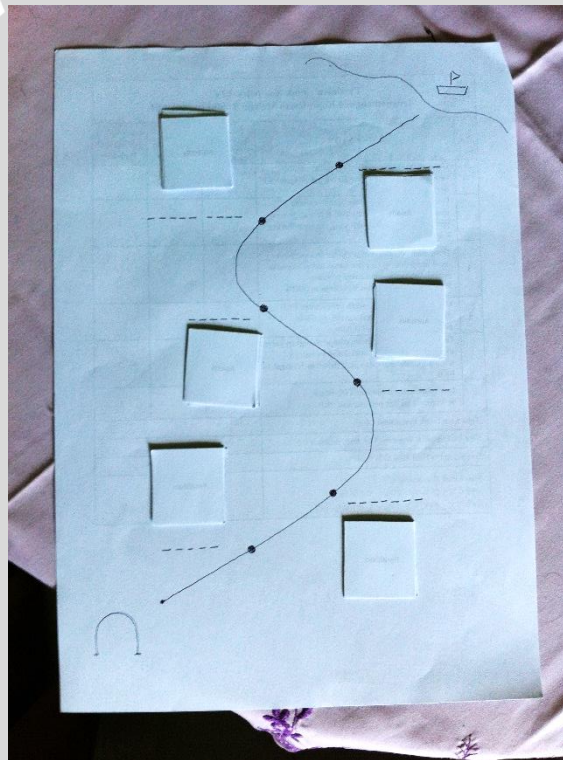
Peraturan	Keterangan
Jumlah Pemain	1 orang
<i>Objectives</i>	Mengumpulkan potongan tiket untuk naik perahu dan menemukan teman-teman
<i>Setup</i>	Pemain berada di pintu masuk kebun binatang. Memiliki 0 potongan tiket dan 1 <i>hint</i>
<i>Progression of Play</i>	<i>Game</i> dibagi menjadi 2 babak. Tujuan tiap babak adalah mengumpulkan potongan tiket untuk naik perahu dan menyeberangi sungai. Babak I: Terdapat 25 kandang hewan yang belum teridentifikasi. Pemain harus menebak hewan yang ada di tiap kandang untuk mendapatkan potongan tiket. Di tiap kandang akan diberikan petunjuk yang akan ditampilkan satu per satu. Pemain dapat menjawab dengan menggunakan bantuan "<i>hint</i>" atau tanpa "<i>hint</i>". Jika pemain dapat menjawab, pemain akan mendapatkan 1 potongan tiket dan pada kandang akan muncul gambar binatang yang ditebak dan ditandai dengan bintang: 3 bintang jika pemain menebak dengan ≤ 3 petunjuk, pemain mendapat <i>hint</i> untuk digunakan selanjutnya 2 bintang jika pemain menebak dengan ≤ 5 petunjuk 1 bintang jika pemain menebak dengan > 5 petunjuk Setelah mendapatkan 25 potongan tiket, pemain dapat menaiki perahu untuk menuju babak selanjutnya. Babak II: Terdapat <i>grid</i> kotak 5x5. Tiap kotak terdapat pertanyaan. Pemain harus mendapatkan 3 potongan tiket untuk menaiki perahu. 1 potongan tiket didapatkan dengan membentuk 1 garis terdiri dari 5

Tabel 4.1 Peraturan *paper prototyping* iterasi pertama (lanjutan)

Peraturan	Keterangan
	kotak secara horizontal, vertikal, atau diagonal. Untuk membentuk garis tersebut, pemain harus menjawab pertanyaan dari kotak tersebut. Setelah mendapatkan 3 potongan tiket, pemain dapat menaiki perahu untuk menyeberangi sungai dan mencari teman-teman.
<i>Resolution</i>	Ketika pemain telah 2x menyeberangi sungai dan dapat menemukan teman-teman, pemain itu menang.

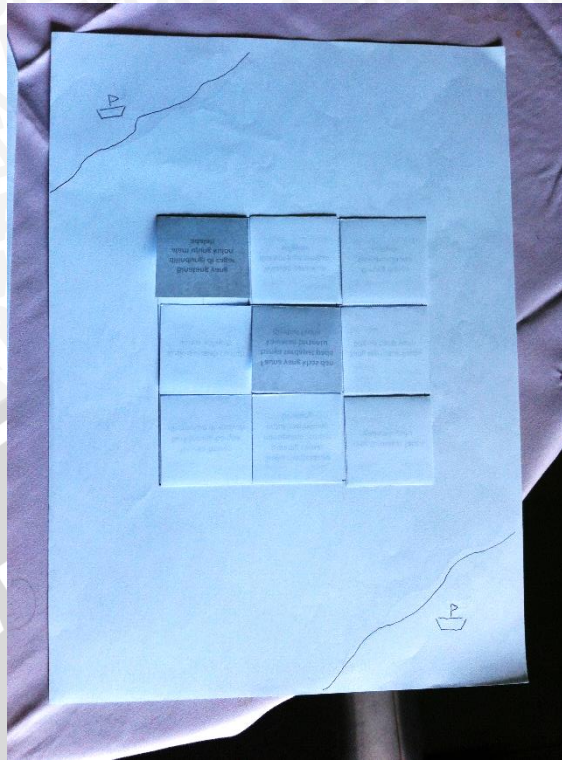
2. *Prototype*

Gambar 4.1 merupakan *paper prototype* yang dihasilkan dari iterasi pertama pada babak pertama. Di babak pertama terdapat enam *stage* yang harus dilalui oleh pemain. Pada tiap *stage* pemain harus menebak hewan dari petunjuk yang disediakan.



Gambar 4.1 *Paper prototype* perancangan iterasi pertama babak 1

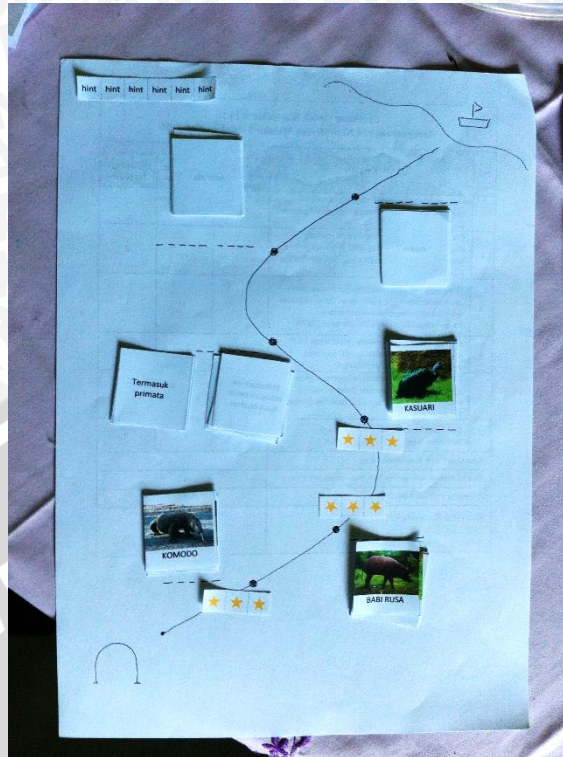
Gambar 4.2 merupakan *paper prototype* sesuai perancangan untuk babak kedua. Pada babak kedua terdapat 3x3 *grid* yang berisi pertanyaan. Setelah melewati babak kedua ini, pemain dapat memenangkan permainan.



Gambar 4.2 *Paper prototype* perancangan iterasi pertama babak 2

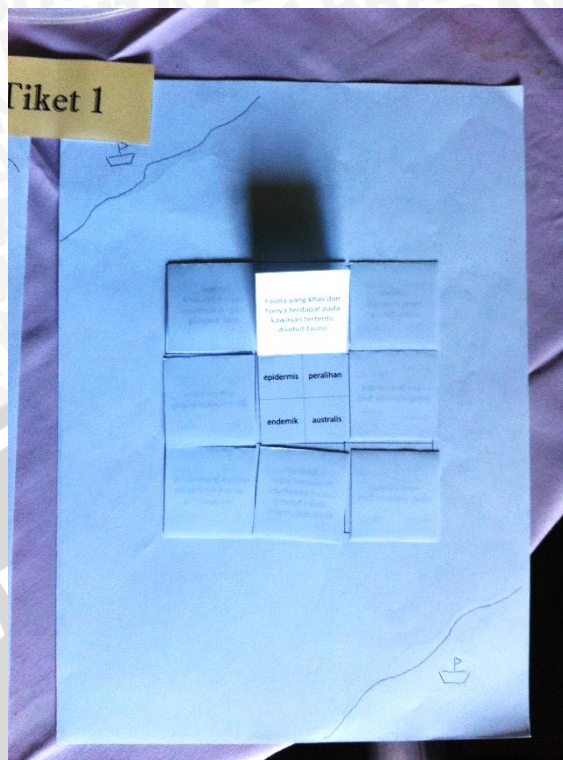
3. *Playtest*

Gambar 4.3 adalah pelaksanaan *playtest* dengan peraturan yang dibuat pada perancangan iterasi pertama. Jika pemain berhasil menebak hewan sesuai petunjuk yang disediakan, maka pemain dapat menyelesaikan satu *stage*. Pada *stage* yang telah diselesaikan tersebut akan diberi bintang sesuai petunjuk yang dibutuhkan pemain.



Gambar 4.3 Playtest perancangan iterasi pertama babak 1

Gambar 4.4 adalah pelaksanaan playtest iterasi pertama pada babak kedua. Pada babak ini pemain memilih salah satu petak dan menjawab pertanyaan yang ada di petak tersebut. Pemain harus membentuk garis yang terdiri dari tiga petak untuk memenangkan permainan.



Gambar 4.4 Playtest perancangan iterasi pertama babak 2

4. Evaluasi

Berdasarkan *playtest* yang telah dilakukan, *game* yang dibangun terbilang kurang menyenangkan. Hal ini disebabkan karena *gameplay* yang ada masih sangat monoton, yaitu seperti menjawab kuis biasa. Selain itu, *game* yang dibangun kurang menantang karena tidak adanya pengambilan keputusan dari pemain yang melibatkan resiko.

4.1.1.2 Iterasi Kedua

1. Perancangan

Tabel 4.2 menunjukkan peraturan untuk *paper prototyping* iterasi kedua.

Tabel 4.2 Peraturan *paper prototyping* iterasi kedua

Peraturan	Keterangan
Jumlah Pemain	1 orang
<i>Objectives</i>	Mengumpulkan 25 medali emas dari 25 hewan
<i>Setup</i>	Pemain mempunyai uang 50000, 1 kandang gajah dan 1 gajah di dalamnya
<i>Progression of Play</i>	Tiap giliran, pemain melempar dadu. Jika angka yang keluar: - 1&2: Pemain mendapat pesan dari penjaga hutan bahwa ada hewan (<i>random</i> sesuai kandang yang ada) yang sakit dan perlu diselamatkan. Jika semua kandang terisi penuh, maka pesan ini dapat diabaikan dan pemain dapat melanjutkan gilirannya. Jika

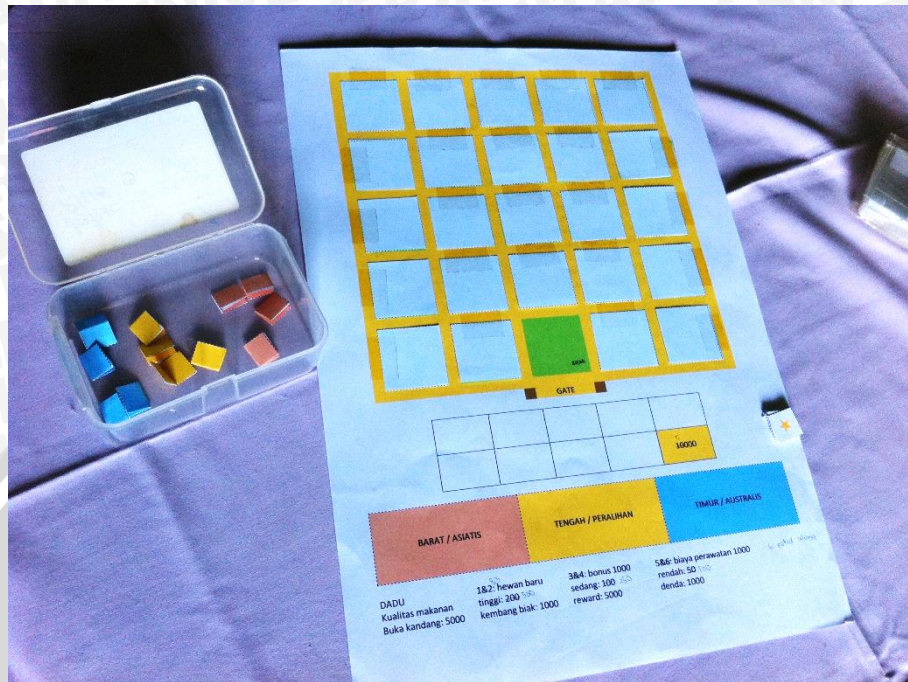
Tabel 4.2 Peraturan *paper prototyping* iterasi kedua (lanjutan)

Peraturan	Keterangan
	tidak, pemain dapat memilih untuk menyelamatkan hewan tersebut atau membiarkannya. Jika pemain memilih untuk membiarkan, pemain tidak perlu melakukan apa-apa dan melanjutkan gilirannya. Jika pemain memilih untuk menyelamatkan, pemain harus memilih salah satu dari 3 makanan untuk dijadikan umpan untuk menangkap hewan. • Makanan kualitas tinggi. Harga: 200. Kesempatan: 75% • Makanan kualitas sedang. Harga: 100. Kesempatan: 50% • Makanan kualitas rendah. Harga: 50. Kesempatan: 25% Jika hewan berhasil ditangkap, maka hewan akan diletakkan di kandang yang sesuai hewan tersebut. Jika hewan kabur, maka pemain dapat melanjutkan gilirannya. - 3&4: Pemain mendapatkan bonus dari pengunjung kebun binatang sebesar 1000. - 5&6: Pemain membayar biaya perawatan kebun binatang sebesar 1000. Selain melempar dadu, pemain juga dapat melakukan: - membuka kandang hewan baru dengan biaya sebesar 5000. - mengembangbiakkan hewan untuk menghasilkan anak, biaya: 1000. - melepaskan hewan kembali ke habitatnya. Hewan dapat dilepaskan jika sudah ada 4 hewan di dalam kandang (2 induk & 2 anak). Pemain dapat memilih habitat untuk hewan, yaitu wilayah barat, timur, dan peralihan. Jika habitat benar (misalnya gajah dilepaskan di habitat wilayah barat), maka pemain mendapat hadiah sebesar 5000. Jika habitat salah, maka pemain harus membayar 1000 untuk biaya pemindahan hewan. Setiap melepaskan hewan pada habitat yang benar, pemain mendapat medali sesuai hewan tersebut (misalnya melepaskan gajah 1x mendapatkan medali gajah perunggu). • Medali perunggu: 1x melepaskan hewan • Medali perak: 2x melepaskan hewan • Medali emas: 3x melepaskan hewan Setiap 10 giliran/lemparan dadu, pemain mendapatkan uang dari pendapatan kebun binatang sebesar 10000.
Resolution	Ketika pemain telah mendapatkan 25 medali emas, pemain tersebut menang

2. *Prototype*

Gambar 4.5 adalah *paper prototype* yang dihasilkan dari perancangan iterasi kedua. Di bagian atas terdapat petak-petak untuk menandakan 25 kandang hewan. Di bawah kandang hewan terdapat petak untuk menandakan giliran pemain. Bagian paling bawah adalah area melepaskan hewan yang terdiri dari area

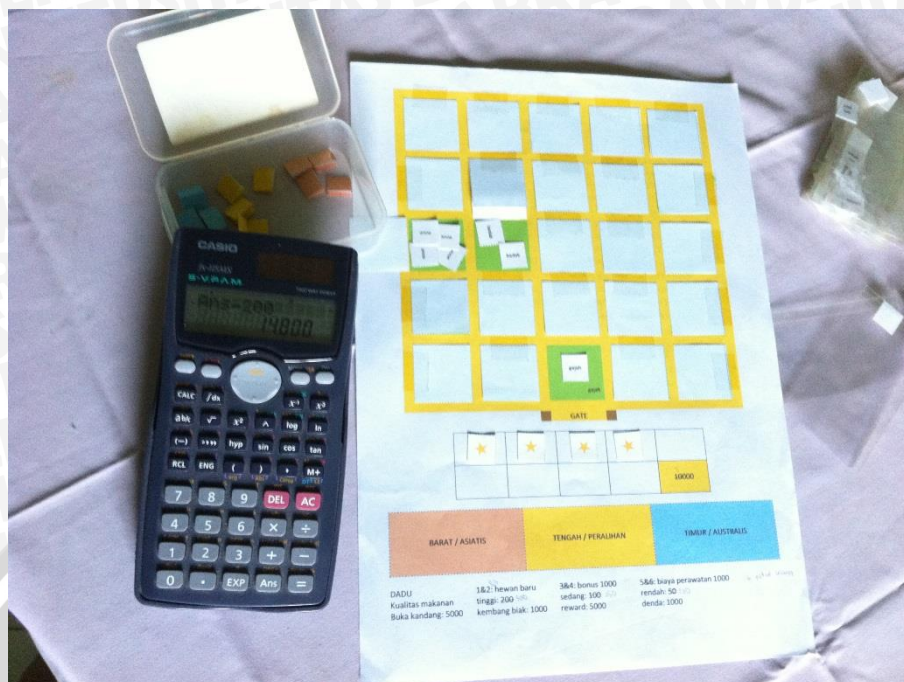
asiatis, peralihan, dan australis. Kotak berisi kertas di samping kiri adalah undian hasil umpan yang digunakan saat menangkap hewan.



Gambar 4.5 Paper prototype perancangan iterasi kedua

3. Playtest

Gambar 4.6 adalah pelaksanaan *playtest* dengan peraturan yang dibuat pada perancangan iterasi kedua. Pemain dapat memutar dadu, membeli kandang, dan mengontrol kandang yang dimiliki. Kalkulator yang ada di sebelah kiri gambar dilakukan untuk menghitung uang yang dimiliki pemain. Kandang yang terbuka berwarna hijau, dan kertas yang berada di kandang tersebut menandakan hewan yang ada di kandang tersebut.



Gambar 4.6 Playtest perancangan iterasi kedua

4. Evaluasi

Sesuai hasil *playtest* yang telah dilakukan, *gameplay* dinilai telah cukup baik. Kekurangan dari iterasi kedua ini adalah penggunaan uang dan harga-harga yang belum sesuai. Berikut adalah saran-saran untuk *prototype* selanjutnya.

- *Game* diubah menjadi *turn-based*.
- Penentuan waktu untuk mengembangbiakkan hewan.
- Kemungkinan dadu untuk mendapatkan hewan baru ditambah.

4.1.1.3 Iterasi Ketiga

1. Perancangan

Tabel 4.3 menunjukkan peraturan untuk *paper prototyping* iterasi ketiga.

Tabel 4.3 Peraturan *paper prototyping* iterasi ketiga

Peraturan	Keterangan
Jumlah Pemain	2 sampai 5 orang
<i>Objectives</i>	Mengumpulkan 5 buah medali
<i>Setup</i>	Semua kandang masih tertutup. Setiap pemain memilih warna yang melambangkan dirinya untuk menandai kandang. Setiap pemain mempunyai uang 20000.
<i>Progression of Play</i>	Tiap giliran, pemain dapat melakukan: - melempar dadu. (wajib)

Tabel 4.3 Peraturan *paper prototyping* iterasi ketiga (lanjutan)

Peraturan	Keterangan
	- membuka kandang hewan baru dengan biaya sebesar 5000. (opsional) - mengembangkan hewan. (opsional) - melepaskan hewan kembali ke habitatnya. (opsional) - menjual/membeli kandang. (opsional) Setiap 5 giliran/lemparan dadu, pemain mendapatkan uang dari pendapatan kebun binatang sebesar 2000. MELEMPAR DADU Jika angka yang keluar: - 1&2&3: Pemain mendapat pesan dari penjaga hutan bahwa ada hewan (<i>random</i> sesuai kandang yang dibuka) yang sakit dan perlu diselamatkan. Jika semua kandang terisi penuh, maka pesan ini dapat diabaikan dan pemain dapat melanjutkan gilirannya. Jika tidak, pemain dapat memilih untuk menyelamatkan hewan tersebut atau membiarkannya. Jika pemain memilih untuk membiarkan, pemain tidak perlu melakukan apa-apa dan melanjutkan gilirannya. Jika pemain memilih untuk menyelamatkan, pemain harus memilih salah satu dari 3 makanan untuk dijadikan umpan untuk menangkap hewan. • Makanan kualitas tinggi. Harga: 800. Kesempatan: 75% • Makanan kualitas sedang. Harga: 400. Kesempatan: 50% • Makanan kualitas rendah. Harga: 200. Kesempatan: 25% Jika hewan berhasil ditangkap, maka hewan akan diletakkan di kandang yang sesuai hewan tersebut. Jika hewan kabur, maka pemain dapat melanjutkan gilirannya. - 4: Pemain mendapatkan bonus dari pengunjung kebun binatang sebesar 1000. - 5: Pemain membayar biaya perawatan kebun binatang sebesar 1000. - 6: Putar ulang (tidak dipakai) MENGEMBANGBIAKKAN HEWAN Hewan dapat dikembangkan dengan harga 1000. Hewan dapat dikembangkan jika sudah ada 2 hewan atau lebih di dalam kandang. Hewan baru akan lahir setelah 3 giliran. MELEPASKAN HEWAN Hewan dapat dilepaskan jika sudah ada 4 hewan di dalam kandang (2 induk & 2 anak) dan keempat hewan dalam kandang tersebut harus dilepaskan semua. Pemain dapat memilih habitat untuk hewan, yaitu wilayah barat, timur, dan peralihan. Jika habitat benar (misalnya gajah dilepaskan di habitat wilayah barat), maka pemain mendapat hadiah sebesar 7000. Jika habitat salah, maka pemain harus membayar 2000 untuk biaya pemindahan hewan. Setiap melepaskan hewan pada habitat yang benar, pemain mendapat 1 medali.

Tabel 4.3 Peraturan *paper prototyping* iterasi ketiga (lanjutan)

Peraturan	Keterangan
	MENJUAL/MEMBELI KANDANG Pemain dapat menjual kandang dengan kisaran harga sebagai berikut. - Kandang saja. Min:2500. Max: 7000 - Kandang + 1 hewan. Min:3000. Max: 8000 - Kandang + 2 hewan. Min:3500. Max: 9000 - Kandang + 3 hewan. Min:4000. Max: 10000 - Kandang + 4 hewan. Min:4500. Max: 11000 Pemain lain dapat membeli kandang tersebut.
<i>Resolution</i>	Pemain menang jika telah mendapat 5 medali. Pemain kalah jika uangnya habis.

2. *Prototype*

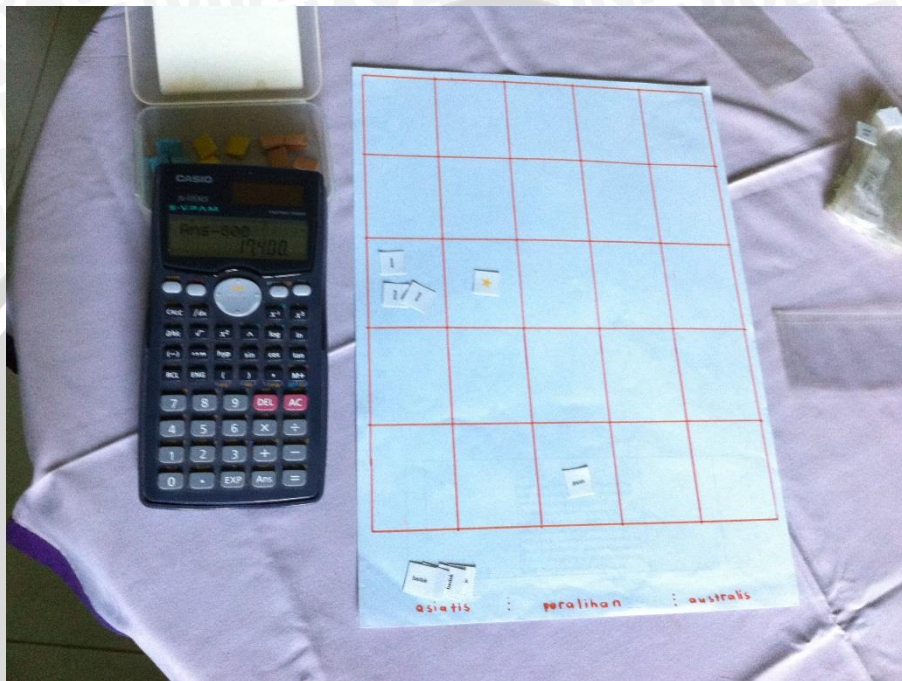
Gambar 4.7 adalah *paper prototype* yang dihasilkan dari perancangan iterasi ketiga. Terdapat 25 petak yang menandakan 25 kandang hewan. Pada bagian bawah terdapat area untuk melepaskan hewan yang terdiri dari area asiatis, peralihan, dan australis. Kotak berisi kertas di sebelah kiri adalah undian untuk hasil umpan pada saat menangkap hewan.



Gambar 4.7 *Paper prototype* perancangan iterasi ketiga

3. Playtest

Gambar 4.8 adalah pelaksanaan *playtest* dengan peraturan yang dibuat pada perancangan iterasi ketiga. Pemain dapat memutar dadu, membeli kandang, dan mengontrol kandang yang dimiliki. Petak yang berisi kertas menandakan kandang yang telah terbuka dan berisi hewan. Kalkulator di sebelah kiri gambar digunakan untuk menghitung uang yang dimiliki pemain.



Gambar 4.8 *Playtest* perancangan iterasi ketiga

4. Evaluasi

Sesuai *playtest* yang telah dilakukan, *prototype* yang dihasilkan pada iterasi ketiga ini sudah baik. *Prototype* yang dihasilkan dapat diimplementasikan pada program.

4.1.2 Deskripsi Game

Tabel 4.4 menunjukkan gambaran umum tentang *game* yang dibangun dengan judul "Nusantara Rescue Zoo".

Tabel 4.4 Gambaran *game* "Nusantara Rescue Zoo"

No.	Elemen	Deskripsi
1.	Judul Game	Nusantara Rescue Zoo
2.	Platform	Android
3.	Target Usia	Semua umur
4.	Rating ESRB	E (Everyone)
5.	Genre	Strategy, Edukasi
6.	Gameplay	Turn-based

Tabel 4.4 Gambaran *game* “Nusantara Rescue Zoo” (lanjutan)

No.	Elemen	Deskripsi
7.	Unique Selling Point (USP)	1. Sebagai media untuk mengenal persebaran fauna di Indonesia 2. Desain antarmuka yang menarik

Berikut adalah penjelasan tentang elemen formal *game* yang terbentuk.

1. *Players*

Game ini dirancang untuk dua sampai lima pemain. *Game* ini berbasis *turn-based*, di mana pemain bergiliran saat memainkan *game*.

2. *Objectives*

Tujuan dari *game* ini adalah mengumpulkan lima buah medali. Medali didapatkan setelah pemain melepaskan hewan ke habitat aslinya. Pemain yang terlebih dahulu mengumpulkan lima buah medali akan memenangkan permainan.

3. *Rules*

Aturan memiliki tiga kategori yaitu *setup*, *progression of play*, dan *resolution*.

- *Setup*
Setiap pemain memilih karakter yang melambangkan dirinya. Setiap pemain memiliki uang sebesar 20000 dan 0 medali. Semua kandang masih tertutup.
- *Progression of Play*
Tiap giliran, pemain dapat melakukan:
 - Menekan tombol putar. (wajib)
 - Membuka kandang hewan baru dengan biaya sebesar 5000. (opsional)
 - Mengembangbiakkan hewan. (opsional)
 - Melepaskan hewan kembali ke habitatnya. (opsional)
 - Menjual/membeli kandang. (opsional)

Setiap 5 giliran/lemparan dadu, pemain mendapatkan uang dari pendapatan kebun binatang sebesar 2000.

MENEKAN TOMBOL PUTAR

Berikut adalah kemungkinan yang dapat muncul saat menekan tombol putar:

- Pemain mendapat pesan dari penjaga hutan bahwa ada hewan (*random* sesuai kandang yang dibuka) yang sakit dan perlu diselamatkan. Jika semua kandang terisi penuh, maka pesan ini dapat diabaikan dan pemain dapat melanjutkan gilirannya. Jika tidak, pemain dapat memilih untuk menyelamatkan hewan tersebut atau membiarkannya. Jika pemain memilih untuk membiarkan, pemain tidak perlu melakukan apa-apa dan

melanjutkan gilirannya. Jika pemain memilih untuk menyelamatkan, pemain harus memilih salah satu dari 3 makanan untuk dijadikan umpan untuk menangkap hewan.

- Makanan kualitas tinggi. Harga: 800. Kesempatan: 75%
- Makanan kualitas sedang. Harga: 400. Kesempatan: 50%
- Makanan kualitas rendah. Harga: 200. Kesempatan: 25%

Jika hewan berhasil ditangkap, maka hewan akan diletakkan di kandang yang sesuai hewan tersebut. Jika hewan tidak tertangkap, maka pemain dapat melanjutkan gilirannya.

- Pemain mendapatkan bonus dari pengunjung kebun binatang sebesar 1000.
- Pemain membayar biaya perawatan kebun binatang sebesar 1000.

MENGEMBANGBIAKKAN HEWAN

Hewan dapat dikembangkan dengan harga 1000. Hewan dapat dikembangkan jika sudah ada 2 hewan atau lebih di dalam kandang. Hewan baru akan lahir setelah 3 giliran.

MELEPASKAN HEWAN

Hewan dapat dilepaskan jika sudah ada 4 hewan di dalam kandang (2 induk & 2 anak) dan keempat hewan dalam kandang tersebut harus dilepaskan semua. Pemain dapat memilih habitat untuk hewan, yaitu wilayah barat, timur, dan peralihan. Jika habitat benar (misalnya gajah dilepaskan di habitat wilayah barat), maka pemain mendapat hadiah sebesar 7000. Jika habitat salah, maka pemain harus membayar 2000 untuk biaya pemindahan hewan. Setiap melepaskan hewan pada habitat yang benar, pemain mendapat 1 medali.

MENJUAL/MEMBELI KANDANG

Pemain dapat menjual kandang dengan kisaran harga sebagai berikut.

- Kandang saja. Minimum: 2500. Maksimum: 7000
- Kandang dengan 1 hewan. Minimum: 3000. Maksimum: 8000
- Kandang dengan 2 hewan. Minimum: 3500. Maksimum: 9000
- Kandang dengan 3 hewan. Minimum: 4000. Maksimum: 10000
- Kandang dengan 4 hewan. Minimum: 4500. Maksimum: 11000

Pemain lain dapat membeli kandang tersebut.

- *Resolution*
Pemain menang jika telah mendapat 5 medali. Pemain kalah jika uangnya habis.

4. *Resources and Resources Management*

Kondisi sumber daya (*resources*) tiap pemain bergantung pada keputusan yang diambil pemain tersebut. Berikut ini adalah sumber daya yang dimiliki oleh tiap pemain.

1. Uang; dapat digunakan pemain untuk membeli *item-item* dalam *game*.
2. Medali; didapatkan setelah pemain melepaskan hewan ke habitat aslinya.
3. Kandang hewan; dapat diisi dengan cara menangkap dan mengembangbiakkan hewan.

5. *Game State*

Game state menyimpan informasi tentang uang dan medali yang dimiliki oleh tiap pemain, pemain yang sedang aktif, serta status kandang. Status kandang dimiliki oleh tiap kandang dan berisi tentang terbuka atau tertutupnya kandang, pemilik kandang, dan jumlah hewan yang ada di kandang.

6. *Information*

Informasi yang bersifat terbuka untuk tiap pemain yaitu jumlah uang, medali, dan keadaan kandang yang dimiliki pemain tersebut, serta kandang mana saja yang masih tertutup dan dimiliki oleh pemain lain. Sedangkan informasi yang bersifat tertutup untuk pemain adalah hewan yang ada di kandang yang masih tertutup.

7. *Sequencing*

Pemain memainkan *game* secara bergiliran mulai dari pemain pertama hingga pemain terakhir. Pada gilirannya, pemain dapat memutar dadu, membeli kandang, menjual kandang, mengembangbiakkan hewan, dan melepaskan hewan dengan urutan sesuai keinginan pemain.

8. *Player Interaction*

Interaksi yang bisa terjadi antar sesama pemain pada *game* ini adalah jual-beli (*trading*). Seorang pemain dapat menjual kandang yang dimilikinya dan pemain lain dapat membeli kandang tersebut.

9. *Theme*

Pemain bertindak sebagai karakter yang memiliki kandang hewan di kebun binatang. Misi pemain adalah menyelamatkan hewan yang mengalami cedera di hutan. Pemain harus mengembangbiakkan hewan tersebut, lalu melepaskannya ke habitat asli hewan tersebut.

10. *Game as Systems*

Elemen-elemen formal yang telah disebutkan membentuk suatu sistem *game* di mana elemen-elemen tersebut saling berkaitan dan mempengaruhi antara satu dengan yang lain.

4.2 Implementasi

Sebelum melakukan implementasi, batasan implementasi harus ditentukan terlebih dahulu. Implementasi dilakukan dengan melakukan proses *digital prototyping* yang terdiri dari implementasi, *quality assurance*, *playtest*, dan evaluasi.

4.2.1 Batasan Implementasi

Berikut adalah batasan-batasan yang digunakan dalam proses implementasi.

1. Bahasa pemrograman yang digunakan adalah C#
2. Permainan dikembangkan untuk perangkat Android dengan resolusi layar 1024×600.

Sedangkan spesifikasi perangkat yang digunakan untuk implementasi telah dijelaskan pada Bab 3.

4.2.2 Implementasi *Gameplay*

Tahap implementasi *gameplay* meliputi implementasi prosedur program dan implementasi antarmuka.

4.2.2.1 Implementasi Prosedur Program

Bagian ini akan menjelaskan tentang *source code* program yang diimplementasikan pada *game* "Nusantara Rescue Zoo". Penjelasan berfokus pada fungsi-fungsi yang menjadi inti dari prosedur program.

1. Implementasi Pemilihan Pemain

Berikut ini akan dijelaskan *method-method* yang digunakan saat pemilihan pemain.

Tabel 4.5 *Method* Start pada pemilihan pemain

1	<code>void Start () {</code>
2	<code> setP1 (1);</code>
3	<code> setP2 (2);</code>
4	<code> setP3 (3);</code>
5	<code> setP4 (4);</code>
6	<code> setP5 (5);</code>
7	<code> setPlayers (2);</code>
8	<code>}</code>

Tabel 4.5 berisi *method* Start yang dijalankan saat awal *scene* dimuat. *Method* tersebut berfungsi untuk menentukan jumlah pemain dan karakter untuk masing-masing pemain secara *default*.

Tabel 4.6 *Method* endSelection pada pemilihan pemain

1	<code>public void endSelection()</code>
2	<code>{</code>
3	<code> int players = PlayerPrefs.GetInt("players");</code>
4	<code> bool duplicate = false;</code>
5	<code> for(int i = 1; i <= players; i++)</code>

Tabel 4.6 Method endSelection pada pemilihan pemain (lanjutan)

6	{
7	for(int j = i - 1; j >= 1; j--)
8	{
9	if(PlayerPrefs.GetInt("playerchar"+i) ==
10	PlayerPrefs.GetInt("playerchar"+j))
11	{
12	duplicate = true;
13	break;
14	}
15	}
16	if(duplicate == true)
17	break;
18	}
19	
20	if(duplicate == true)
21	warning.SetActive(true);
22	else
23	Application.LoadLevel("Game");
24	}

Tabel 4.6 menunjukkan *method* endSelection yang dijalankan saat pemilihan pemain berakhir. *Method* ini digunakan untuk menyimpan jumlah pemain dan mengecek apakah ada karakter yang sama untuk pemain yang berbeda.

2. Implementasi Awal Permainan

Berikut ini akan dijelaskan *method* yang digunakan saat awal permainan.

Tabel 4.7 Method Start pada awal permainan

1	void Start () {
2	int i;
3	
4	players = PlayerPrefs.GetInt("players");
5	playerchar = new int[players];
6	money = new int[players];
7	medal = new int[players];
8	rollcount = new int[players];
9	for(i = 0; i < players; i++)
10	{
11	playerchar[i] = PlayerPrefs.GetInt("playerchar"
12	+ (i+1)) - 1;
13	money[i] = 20000;
14	medal[i] = 0;
15	rollcount[i] = 0;
16	}
17	activePlayer = 0;
18	activePlayerImage.GetComponent<Image>().sprite =
19	characters[playerchar[0]];
20	breeding = new List<breedInfo>();
21	
22	for(i = 0; i < 25; i++)
23	{
24	if(kandang[i].isOpen == true)
25	{
26	kandang[i].button.GetComponent
27	<Image>().sprite = kandang[i].logo;
28	if(kandang[i].ownerID == activePlayer)
29	

Tabel 4.7 Method Start pada awal permainan (lanjutan)

30		kandang[i].button.GetComponent
32	<Button>().interactable = true;	
33	else	
34		kandang[i].button.GetComponent
35	<Button>().interactable = false;	
36	}	
37	else	
38		
39	kandang[i].button.GetComponent<Image>().sprite = unopened;	
40	}	
41	activeKandang = 0;	
	}	

Tabel 4.7 menunjukkan baris program yang dijalankan saat permainan akan dimulai. *Method* ini mengimplementasikan aturan *setup* yang sudah dirancang. Hasil dari *method* ini adalah keadaan awal permainan seperti uang dan medali yang dimiliki masing-masing pemain, serta kandang-kandang yang masih tertutup.

3. Implementasi Putar Dadu

Berikut ini akan dijelaskan *method-method* yang digunakan saat memutar dadu.

Tabel 4.8 Method rollButton pada putar dadu

1	public void rollButton()
2	{
3	checkWin();
4	checkLose();
5	
6	if(isRolled == false)
7	{
8	rolled();
9	isRolled = true;
10	rollcount[activePlayer]++;
11	if(rollcount[activePlayer] % 5 == 0)
12	{
13	incomePanel.SetActive(true);
14	boardPanel.SetActive(true);
15	}
16	rollButtonGO.GetComponent<Image>().sprite =
17	rollButtonImage[1];
18	}
19	else
20	{
21	if(activePlayer == players - 1)
22	activePlayer = 0;
23	else
24	activePlayer++;
25	
26	isRolled = false;
27	rollButtonGO.GetComponent<Image>().sprite =
28	rollButtonImage[0];
29	activePlayerText.GetComponent<Text>().text =
30	"Giliran Pemain " + (activePlayer + 1);
32	activePlayerImage.GetComponent<Image>().sprite
33	= characters[playerchar[activePlayer]];
34	}

Tabel 4.8 Method rollButton pada putar dadu (lanjutan)

35	activePlayerMoney.GetComponent<Text>(). text =
36	money[activePlayer].ToString();
37	activePlayerMedal.GetComponent<Text>().text =
38	medal[activePlayer] + " / 5";
39	
40	for(int i = 0; i < 25; i++)
41	{
42	if(kandangs[i].isOpen == true)
43	{
44	kandangs[i].button.GetComponent
45	<Image>().sprite = kandangs[i].logo;
46	if(kandangs[i].ownerID ==
47	activePlayer)
48	kandangs[i].button.
49	GetComponent<Button>().interactable = true;
50	else
51	kandangs[i].button.
52	GetComponent<Button>().interactable = false;
53	}
54	else
55	kandangs[i].button.
56	GetComponent<Image>().sprite = unopened;
57	}
58	checkBreed();
59	}

Pada Tabel 4.8 ditunjukkan baris program *method* rollButton yang dipanggil saat tombol putar atau tombol akhiri giliran ditekan. Pada awal baris selalu dilakukan pengecekan apakah permainan sudah memenuhi kondisi menang atau kondisi kalah. Saat tombol putar ditekan, program akan memanggil fungsi rolled() yang terlihat pada baris 8. Setelah itu akan dilakukan pengecekan apakah pemain sudah lima kali memutar dadu. Tiap lima kali giliran, pemain akan mendapat uang. Saat tombol akhiri giliran ditekan, dilakukan pergantian pemain seperti terlihat pada baris 21 sampai 24. Setelah itu, tombol kandang akan ditampilkan sesuai dengan kandang yang dimiliki pemain yang sedang aktif. Pada baris 57, fungsi checkBreed() dipanggil untuk mengetahui apakah ada hewan yang baru lahir.

Tabel 4.9 Method rolled pada putar dadu

1	public void rolled()
2	{
3	rolledNumber = Random.Range (1, 6);
4	Debug.Log ("Angka dadu : " + rolledNumber);
5	dialogWindow.SetActive(true);
6	boardPanel.SetActive(true);
7	
8	if(rolledNumber == 4)
9	{
10	characterName.GetComponent<InputField>().text =
11	"Zookeeper";
12	characterImage.GetComponent<Image>().sprite =
13	zookeeper;
14	dialog.GetComponent<Text>().text =
15	"\"Selamat!!\nAnda mendapatkan bonus dari pengujung " +
16	"kebun binatang sebesar 1000\"";

Tabel 4.9 *Method* rolled pada putar dadu (lanjutan)

17	
18	resultSound.GetComponent<ResultSound>().playWon();
19	}
20	else if(rolledNumber == 5)
21	{
22	characterName.GetComponent<InputField>().text =
23	"Zookeeper";
24	characterImage.GetComponent<Image>().sprite =
25	zookeeper;
26	dialog.GetComponent<Text>().text = "\"Anda
27	wajib membayar biaya perawatan kebun " +
28	"binatang sebesar 1000\"";
29	resultSound.GetComponent<ResultSound>().
30	playLose();
31	}
32	}
33	else
34	{
35	characterName.GetComponent<InputField>().text =
36	"Ranger";
37	characterImage.GetComponent<Image>().sprite =
38	ranger;
39	
40	totalKandang = 0;
41	int[] idKandang = new int[25];
42	for(int i = 0; i < 25; i++)
43	{
44	if(kandang[i].isOpen == true &&
45	kandang[i].ownerID == activePlayer && kandang[i].jumlahHewan < 2)
46	{
47	idKandang[totalKandang] = i;
48	totalKandang++;
49	}
50	}
51	if(totalKandang == 0)
52	{
53	dialog.GetComponent<Text>().text =
54	"\"Anda belum membeli kandang atau semua " +
55	"kandang telah terisi penuh\"";
56	}
57	else
58	{
59	randomHewan = idKandang[Random.Range(0,
60	totalKandang)];
61	dialog.GetComponent<Text>().text =
62	"\"Keadaan darurat!!\nAda hewan " +
63	kandang[randomHewan].namaHewan +
64	" yang cedera di hutan.\n" +
65	"Ayo segera selamatkan!\n\"";
66	}
67	}
68	}

Tabel 4.9 menunjukkan *method* rolled yang menunjukkan mekanisme pemutaran dadu. Pada baris 3 program *me-random* angka antara 1 sampai 5. Jika angka yang didapat adalah 4, tampilan program akan menunjukkan bahwa pemain mendapat bonus. Jika angka yang didapat adalah 5, program akan menunjukkan bahwa pemain wajib membayar biaya. Selain dua angka tersebut, maka program

memberikan informasi pada pemain bahwa ada hewan yang cedera. Hewan yang cedera di-random sesuai kandang yang dimiliki pemain.

Tabel 4.10 Method confirmRolled pada putar dadu

1	<code>public void confirmRolled()</code>
2	<code>{</code>
3	<code>boardPanel.SetActive(false);</code>
4	<code>if(rolledNumber == 4)</code>
5	<code>{</code>
6	<code>money[activePlayer] += 1000;</code>
7	<code>activePlayerMoney.GetComponent<Text>(). text =</code>
8	<code>money[activePlayer].ToString();</code>
9	<code>}</code>
10	
11	<code>else if(rolledNumber == 5)</code>
12	<code>{</code>
13	<code>money[activePlayer] -= 1000;</code>
14	<code>activePlayerMoney.GetComponent<Text>(). text =</code>
15	<code>money[activePlayer].ToString();</code>
16	<code>}</code>
17	<code>else if(totalkandang > 0)</code>
18	<code>{</code>
19	<code>catchCanvas.SetActive(true);</code>
20	<code>boardCanvas.SetActive(false);</code>
21	<code>setCatch();</code>
22	<code>}</code>
23	<code>}</code>

Method confirmRolled() dapat dilihat pada Tabel 4.10. Method ini memberi efek setelah dadu diputar. Pada baris 4 jika angka yang didapat adalah 4, uang pemain akan bertambah. Jika angka yang didapat 5, uang pemain akan berkurang. Selain dua angka itu, program akan menuju ke area menangkap hewan.

4. Implementasi Menangkap Hewan

Berikut ini akan dijelaskan *method* yang digunakan saat menangkap hewan.

Tabel 4.11 Method setBait pada menangkap hewan

1	<code>public void setBait(int baitID)</code>
2	<code>{</code>
3	<code>int limit = 0;</code>
4	<code>if(baitID == 1)</code>
5	<code>{</code>
6	<code>money[activePlayer] -= 800;</code>
7	<code>limit = 3;</code>
8	<code>}</code>
9	<code>else if(baitID == 2)</code>
10	<code>{</code>
11	<code>money[activePlayer] -= 400;</code>
12	<code>limit = 2;</code>
13	<code>}</code>
14	<code>else</code>
15	<code>{</code>
16	<code>money[activePlayer] -= 200;</code>
17	<code>limit = 1;</code>
18	<code>}</code>
19	<code>catchAPMoney.GetComponent<Text>(). text =</code>
20	<code>money[activePlayer].ToString();</code>

Tabel 4.11 Method setBait pada menangkap hewan (lanjutan)

21	activePlayerMoney.GetComponent<Text>().text =
22	money[activePlayer].ToString();
23	
24	int rndval = Random.Range(0, 4);
25	Debug.Log("Angka umpan : " + rndval);
26	catchResult.SetActive(true);
27	catchPanel.SetActive(true);
28	if(rndval < limit)
29	{
30	catchResultText.GetComponent<Text>().text =
32	"Selamat!\nHewan " +
33	kandang[randomHewan].namaHewan + "
34	berhasil tertangkap!";
35	resultSound.GetComponent<ResultSound>().
36	playWon();
37	kandang[randomHewan].jumlahHewan++;
38	}
39	else
40	{
41	catchResultText.GetComponent<Text>().text =
42	"Hewan " +
43	kandang[randomHewan].namaHewan + "
44	tidak berhasil tertangkap!";
45	resultSound.GetComponent<ResultSound>().
46	playLose();
47	}
48	}

Method setBait pada Tabel 4.11 menunjukkan hal yang terjadi saat pemain memilih umpan untuk menangkap hewan. Baris 4 sampai 18 menunjukkan uang yang diambil dari pemain dan limit untuk angka *random* sesuai umpan yang dipilih. Baris 28 menunjukkan jika angka *random* kurang dari limit, maka hewan berhasil tertangkap.

Tabel 4.12 Metode untuk menentukan keberhasilan menangkap hewan

No	Umpan	Tingkat Keberhasilan	Limit	Angka Random	Keberhasilan
1	Makanan kualitas tinggi	75%	3	0	Tertangkap
				1	Tertangkap
				2	Tertangkap
				3	Tidak tertangkap
2	Makanan kualitas sedang	50%	2	0	Tertangkap
				1	Tertangkap
				2	Tidak tertangkap
				3	Tidak tertangkap
3	Makanan kualitas rendah	25%	1	0	Tertangkap
				1	Tidak tertangkap
				2	Tidak tertangkap
				3	Tidak tertangkap

Tabel 4.12 adalah metode yang digunakan untuk menentukan keberhasilan menangkap hewan. Jika umpan yang dipilih adalah makanan kualitas tinggi dan angka random yang dihasilkan adalah 0, maka hewan berhasil tertangkap. Hewan berhasil tertangkap apabila angka random yang dihasilkan kurang dari limit yang ditentukan.

5. Implementasi Kandang

Berikut ini akan dijelaskan *method-method* yang digunakan saat tombol kandang ditekan.

Tabel 4.13 Method kandangButton pada kandang

1	<code>public void kandangButton(int kandangID)</code>
2	<code>{</code>
3	<code> activeKandang = kandangID;</code>
4	<code> if(kandangs[kandangID].isOpen == false)</code>
5	<code> {</code>
6	<code> confirmBuyWindow.SetActive(true);</code>
7	<code> boardPanel.SetActive(true);</code>
8	<code> }</code>
9	<code> else</code>
10	<code> {</code>
11	<code> kandangCanvas.SetActive(true);</code>
12	<code> boardCanvas.SetActive(false);</code>
13	<code> setKandang();</code>
14	<code> }</code>
15	<code>}</code>

Method kandangButton pada Tabel 4.12 menunjukkan baris program yang dijalankan saat tombol kandang ditekan. Jika kandang masih tertutup, pemain dapat membeli kandang. Jika kandang terbuka, program akan menuju ke jendela kandang.

Tabel 4.14 Method buyKandang pada kandang

1	<code>public void buyKandang()</code>
2	<code>{</code>
3	<code> kandangs[activeKandang].isOpen = true;</code>
4	<code> kandangs[activeKandang].ownerID = activePlayer;</code>
5	<code> kandangs[activeKandang].button.GetComponent<Image>().</code>
6	<code> sprite = kandangs[activeKandang].logo;</code>
7	<code> kandangs[activeKandang].button.GetComponent<Button>().</code>
8	<code> interactable = true;</code>
9	<code> money[activePlayer] -= 5000;</code>
10	<code> activePlayerMoney.GetComponent<Text>(). text =</code>
11	<code> money[activePlayer].ToString();</code>
12	<code>}</code>

Method buyKandang pada Tabel 4.13 menunjukkan hal yang terjadi saat pemain membeli kandang. Informasi pada kandang diubah sesuai pemain yang membeli kandang. Selain itu uang pemain dikurangi 5000.

Tabel 4.15 Method setKandang pada kandang

1	<code>public void setKandang()</code>
2	<code>{</code>

Tabel 4.15 *Method* setKandang pada kandang (lanjutan)

```

3      kandangHewan.GetComponent<Text>().text = "Kandang " +
4      kandang[activeKandang].namaHewan;
5      for(int i = 0; i < 4; i++)
6      {
7          if(i < kandang[activeKandang].jumlahHewan)
8          {
9              hewan[i].GetComponent<Image>().sprite =
10             kandang[activeKandang].hewan;
11             hewan[i].SetActive(true);
12         }
13         else
14         {
15             hewan[i].SetActive(false);
16         }
17     }
18
19     KandangAPText.GetComponent<Text>().text = "Giliran
20 Pemain " + (activePlayer + 1);
21     KandangAPIImage.GetComponent<Image>().sprite =
22     characters[playerchar[activePlayer]];
23     KandangAPMoney.GetComponent<Text>().text =
24     money[activePlayer].ToString();
25     KandangAPMedal.GetComponent<Text>().text =
26     medal[activePlayer] + " / 5";
27
28     bool isBreeding = false;
29     foreach(breedInfo a in breeding)
30     {
31         if(a.animalID == activeKandang)
32             isBreeding = true;
33     }
34
35     if((kandang[activeKandang].jumlahHewan == 2 ||
36     kandang[activeKandang].jumlahHewan == 3)
37     && isBreeding == false)
38         kembangBiak.GetComponent<Button>().interactable
39     = true;
40     else
41         kembangBiak.GetComponent<Button>().interactable
42     = false;
43
44     if(kandang[activeKandang].jumlahHewan == 4)
45         lepas.GetComponent<Button>().interactable =
46     true;
47     else
48         lepas.GetComponent<Button>().interactable =
49     false;
50
51     if(isBreeding == false)
52         jual.GetComponent<Button>().interactable =
53     true;
54     else
55         jual.GetComponent<Button>().interactable =
56     false;
57 }

```

Method setKandang pada Tabel 4.14 dijalankan saat jendela kandang ditampilkan. Baris 5 sampai 17 berfungsi untuk menampilkan hewan sesuai jumlah

hewan yang ada di kandang tersebut. Jika jumlah hewan adalah 2 atau 3, maka tombol kembang biak aktif. Jika jumlah hewan adalah 4, maka tombol lepaskan aktif. Jika status kandang sedang berkembangbiak, maka tombol kembang biak, tombol lepaskan, dan tombol jual tidak aktif.

6. Implementasi Mengembangbiakkan Hewan

Berikut ini akan dijelaskan *method-method* yang digunakan saat mengembangbiakkan hewan.

Tabel 4.16 Method confirmBreed pada mengembangbiakkan hewan

1	<code>public void confirmBreed()</code>
2	<code>{</code>
3	<code> breedInfo current = new breedInfo();</code>
4	<code> current.animalID = activeKandang;</code>
5	<code> current.ownerID = activePlayer;</code>
6	<code> if(isRolled == true)</code>
7	<code> current.startCount = rollcount[activePlayer] -</code>
8	<code>1;</code>
9	<code> else</code>
10	<code> current.startCount = rollcount[activePlayer];</code>
11	<code> breeding.Add(current);</code>
12	<code> money[activePlayer] -= 1000;</code>
13	<code> confirmBreedWindow.SetActive(false);</code>
14	<code> activePlayerMoney.GetComponent<Text>().text =</code>
15	<code>money[activePlayer].ToString();</code>
16	<code> setKandang();</code>
17	<code>}</code>
18	

Method *confirmBreed* pada Tabel 4.15 digunakan saat pemain mengembangbiakkan hewan. Informasi untuk kembang biak ditambahkan pada *list breeding* terlihat pada baris 12. Baris 13 menunjukkan uang yang diambil dari pemain untuk mengembangbiakkan hewan.

Tabel 4.17 Method checkBreed pada mengembangbiakkan hewan

1	<code>public void checkBreed()</code>
2	<code>{</code>
3	<code> foreach(breedInfo a in breeding)</code>
4	<code> {</code>
5	<code> int diff = rollcount[activePlayer] -</code>
6	<code>a.startCount;</code>
7	<code> if(a.ownerID == activePlayer && diff >= 3)</code>
8	<code> {</code>
9	<code> kandangs[a.animalID].jumlahHewan++;</code>
10	<code> activeKandang = a.animalID;</code>
11	<code> newBreedWindow.SetActive(true);</code>
12	<code> boardPanel.SetActive(true);</code>
13	<code> newBreedText.GetComponent<Text>().text =</code>
14	<code>"Hewan " +</code>
15	<code> kandangs[a.animalID].namaHewan +</code>
16	<code>" yang dikembangbiakkan sudah lahir!";</code>
17	<code> breeding.Remove(a);</code>
18	<code> break;</code>
19	<code> }</code>
20	<code> }</code>
21	<code>}</code>

Method checkBreed pada Tabel 4.16 digunakan untuk mengecek apakah hewan yang dikembangbiakkan sudah lahir. Pada baris 7 ditunjukkan jika pemain sudah melewati 3 giliran atau lebih, maka jumlah hewan pada kandang akan ditambahkan. Pada baris 17 ditunjukkan info kembang biak akan dihapus dari *list breeding*.

7. Implementasi Melepaskan Hewan

Berikut ini akan dijelaskan tentang *method* yang digunakan saat melepaskan hewan.

Tabel 4.18 Method release pada melepaskan hewan

```

1      public void release(string destination)
2      {
3          releaseAreaText.GetComponent<Text>().text = "Area " +
4      destination;
5          for(int i = 0; i < 4; i++)
6          {
7              releaseAnimals[i].GetComponent<Image>().sprite
8      = kandang[activeKandang].hewan;
9          }
10         kandang[activeKandang].jumlahHewan = 0;
11
12         if(destination == kandang[activeKandang].habitat)
13         {
14             releaseDialog.GetComponent<Text>().text =
15             "\"Terima kasih!!\n\" +
16                 "Anda telah merawat hewan " +
17             kandang[activeKandang].namaHewan +
18                 " dengan baik dan mengembalikan
19             ke habitat yang benar yaitu area " +
20                 destination + ". Selamat, Anda
21             berhak mendapatkan 1 medali dan uang " +
22                 "sebesar 7000.\n";
23             resultSound.GetComponent<ResultSound>().
24             playWon();
25             money[activePlayer] += 7000;
26             medal[activePlayer]++;
27         }
28         else
29         {
30             releaseDialog.GetComponent<Text>().text =
31             "\"Maaf, Anda melepaskan hewan ke area " +
32                 "yang salah.\nHabitat yang benar untuk
33             hewan " + kandang[activeKandang].namaHewan +
34                 " adalah area " +
35             kandang[activeKandang].habitat + ". Anda wajib membayar " +
36                 "biaya pemindahan hewan sebesar
37             2000.\n";
38             resultSound.GetComponent<ResultSound>().
39             playLose();
40             money[activePlayer] -= 2000;
41         }
42         releaseAPMoney.GetComponent<Text>(). text =
43         money[activePlayer].ToString();
44         releaseAPMedal.GetComponent<Text>().text =
45         medal[activePlayer] + " / 5";
46
47     }

```


Tabel 4.18 Method release pada melepaskan hewan (lanjutan)

48	activePlayerMoney.GetComponent<Text>().text =
49	money[activePlayer].ToString();
50	activePlayerMedal.GetComponent<Text>().text =
51	medal[activePlayer] + " / 5";
52	}

Method release pada Tabel 4.17 dijalankan pada saat melepaskan hewan. Pada baris 12 terlihat jika area tujuan sesuai, maka dialog yang ditampilkan menginformasikan bahwa habitat telah benar serta pemain mendapat uang dan medali. Jika tidak sesuai, maka dialog akan menunjukkan habitat yang sesuai untuk hewan serta uang pemain akan berkurang.

8. Implementasi Menjual Kandang

Berikut ini akan dijelaskan tentang *method-method* yang digunakan saat menjual kandang.

Tabel 4.19 Method trade pada menjual kandang

1	public void trade()
2	{
3	kandangPanel.SetActive(true);
4	int minharga = 0;
5	int maxharga = 0;
6	
7	switch(kandangs[activeKandang].jumlahHewan)
8	{
9	case 0: minharga = 2500; maxharga = 7000; break;
10	case 1: minharga = 3000; maxharga = 8000; break;
11	case 2: minharga = 3500; maxharga = 9000; break;
12	case 3: minharga = 4000; maxharga = 10000; break;
13	case 4: minharga = 4500; maxharga = 11000; break;
14	}
15	
16	hargaLimit.GetComponent<Text>().text = "Harga Kandang
17	: \n(min : " +
18	minharga + "; max : " + maxharga + ")";
19	hargaSlider.GetComponent<Slider>().minValue =
20	minharga;
21	hargaSlider.GetComponent<Slider>().maxValue =
22	maxharga;
23	hargaSlider.GetComponent<Slider>().value = minharga;
24	hargaText.GetComponent<InputField>().text =
25	minharga.ToString();
26	
27	for(int i = 0; i < 5; i++)
28	{
29	if(i < players)
30	{
31	buyerButton[i].button.SetActive(true);
32	
33	buyerButton[i].image.GetComponent<Image>().sprite =
34	characters[playerchar[i]];
35	if(i == activePlayer)
36	
37	buyerButton[i].button.GetComponent<Button>().interactable =
38	false;
39	
40	else

Tabel 4.19 Method trade pada menjual kandang (lanjutan)

41	
42	buyerButton[i].button.GetComponent<Button>().interactable =
43	true;
44	}
45	else
46	{
47	buyerButton[i].button.SetActive(false);
48	}
49	}
50	}
51	

Method trade pada Tabel 4.18 dijalankan saat tombol jual ditekan. Baris 7 sampai 14 menunjukkan penentuan harga minimal dan maksimal untuk menjual kandang sesuai jumlah hewan yang ada dalam kandang.

Tabel 4.20 Method confirmTrade pada menjual kandang

1	public void confirmTrade()
2	{
3	money[activePlayer] += hargajual;
4	money[activeBuyer] -= hargajual;
5	kandangs[activeKandang].ownerID = activeBuyer;
6	confirmTradeWindow.SetActive(false);
7	
8	activePlayerMoney.GetComponent<Text>(). text =
9	money[activePlayer].ToString();
10	if(kandangs[activeKandang].ownerID == activePlayer)
11	kandangs[activeKandang].button.
12	GetComponent<Button>().interactable = true;
13	else
14	kandangs[activeKandang].button.
15	GetComponent<Button>().interactable = false;
16	}

Method confirmTrade pada Tabel 4.19 dijalankan saat pemain menjual kandang. Pada baris 3 dan 4 uang penjual akan bertambah, sedangkan uang pembeli akan berkurang. Pemilik kandang akan diubah sesuai pemain yang membeli kandang.

4.2.2.2 Implementasi Antarmuka

Bagian ini menjelaskan antarmuka yang diimplementasikan pada *game* "Nusantara Rescue Zoo". Antarmuka yang akan dijelaskan adalah antarmuka yang berperan inti pada *game*.

1. Implementasi Halaman Awal

Implementasi menu utama atau halaman awal dapat dilihat pada Gambar 4.9. Tombol mulai akan memunculkan halaman pemilihan pemain. Tombol pengaturan akan memunculkan halaman pengaturan. Tombol tentang akan memunculkan halaman tentang. Tombol keluar akan mengakhiri permainan dan keluar dari aplikasi.



Gambar 4.9 Tampilan halaman awal

2. Implementasi Halaman Pemilihan Pemain

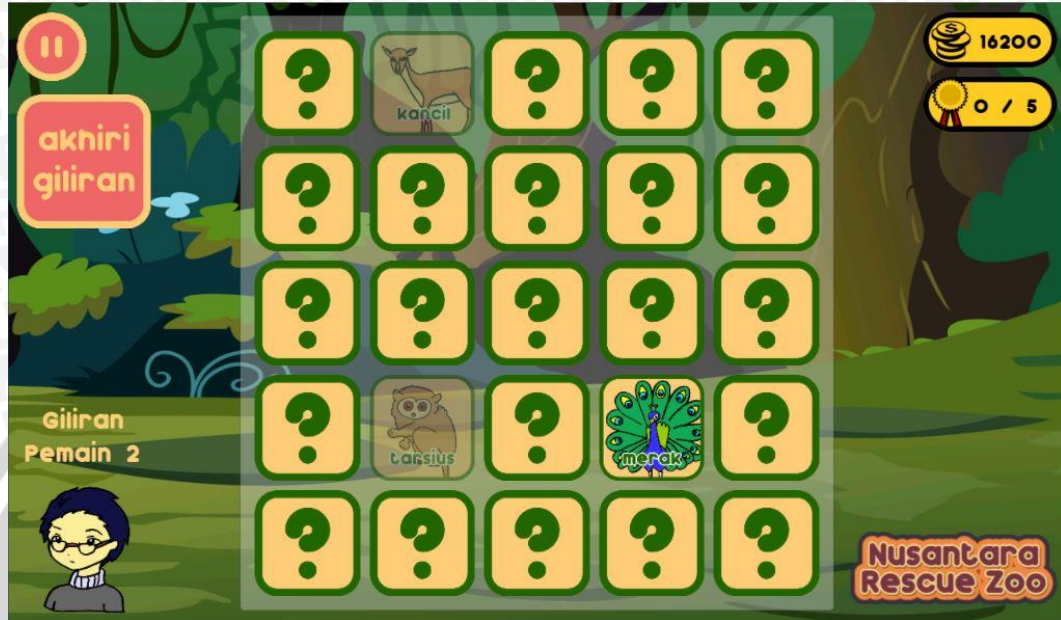
Halaman pemilihan pemain dapat dilihat pada Gambar 4.10. Pada halaman ini pemain dapat menentukan jumlah pemain dan karakter untuk masing-masing pemain.



Gambar 4.10 Tampilan halaman pemilihan pemain

3. Implementasi Halaman *Board*

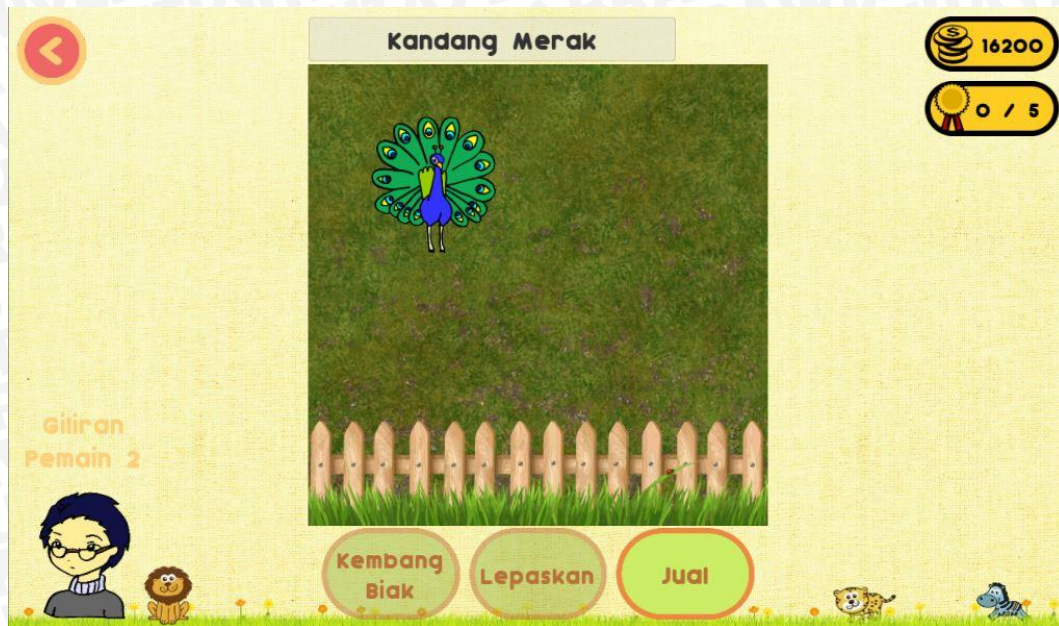
Gambar 4.11 menunjukkan halaman *board* yang berisi tombol-tombol kandang yang dimiliki pemain yang sedang aktif, kandang yang dimiliki pemain lain, dan kandang yang masih belum terbuka.



Gambar 4.11 Tampilan halaman *board*

4. Implementasi Halaman Kandang

Tampilan kandang setelah terbuka terlihat pada Gambar 4.12. Pemain dapat mengembangkan hewan, melepaskan hewan, dan menjual kandang pada halaman ini.



Gambar 4.12 Tampilan halaman kandang

5. Implementasi Halaman Menangkap Hewan

Gambar 4.13 menunjukkan tentang tampilan pada area menangkap hewan. Pemain dapat memilih dari tiga umpan yang ada untuk menangkap hewan.



Gambar 4.13 Tampilan halaman menangkap hewan

6. Implementasi Halaman Melepaskan Hewan

Gambar 4.14 menunjukkan tampilan pada halaman melepaskan hewan. Dialog yang ada menunjukkan apakah pemain berhasil melepaskan hewan ke area yang benar atau tidak.



Gambar 4.14 Tampilan halaman melepaskan hewan

7. Implementasi Halaman Hasil Akhir

Pada Gambar 4.15 terlihat tampilan hasil akhir. Pada halaman ini terlihat hasil akhir permainan berupa *ranking* pemain.



Gambar 4.15 Tampilan halaman hasil akhir

4.2.3 Quality Assurance dan Playtesting

Setelah melaksanakan proses *quality assurance*, ditemukan adanya *bug* pada program. *Bug* yang ada segera diperbaiki lalu dilaksanakan *playtest*. Berikut adalah laporan *bug* pada program.

Tabel 4.21 Laporan *quality assurance*

No	Keterangan <i>Bug</i>	Perbaikan	Status Perbaikan
1	Saat dialog muncul, jendela di belakang dialog masih dapat dioperasikan	Program diperbaiki sehingga halaman di belakang dialog tidak dapat dioperasikan sampai dialog tertutup	Selesai
2	Saat menangkap hewan, hasil hewan tertangkap atau tidak tertangkap seharusnya diberi efek suara	Saat hewan tertangkap diberi efek suara menang dan saat hewan tidak tertangkap diberi efek suara kalah	Selesai

Playtest dilaksanakan setelah proses *quality assurance*. Dari hasil *playtest*, *game* berbentuk digital sudah terbilang baik dan fungsi-fungsi dari *game* berjalan dengan lancar dari awal sampai akhir.

4.3 Pengujian

Ada dua jenis pengujian yang dilakukan, yaitu pengujian *white box* dan *black box*. Pengujian *white box* dilakukan dengan pengujian unit menggunakan metode *basic path testing*. Sedangkan untuk metode *black box* dilakukan dengan metode *test flow diagram*, dan pengujian terhadap pengguna (*focus testing*).

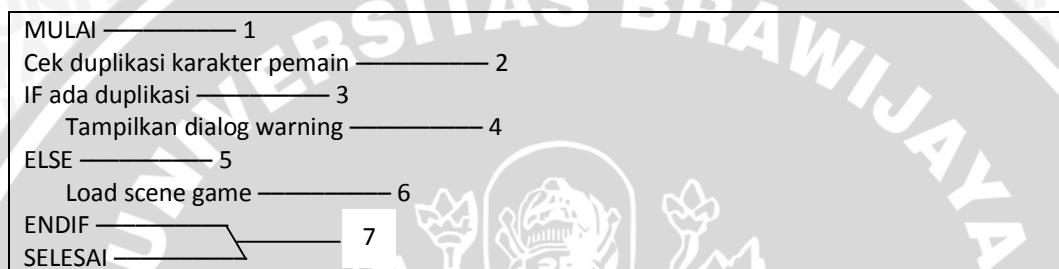
BAB 5 PENGUJIAN

Bab ini akan menjelaskan hasil dan analisis pengujian dengan metode *white box* dan *black box*.

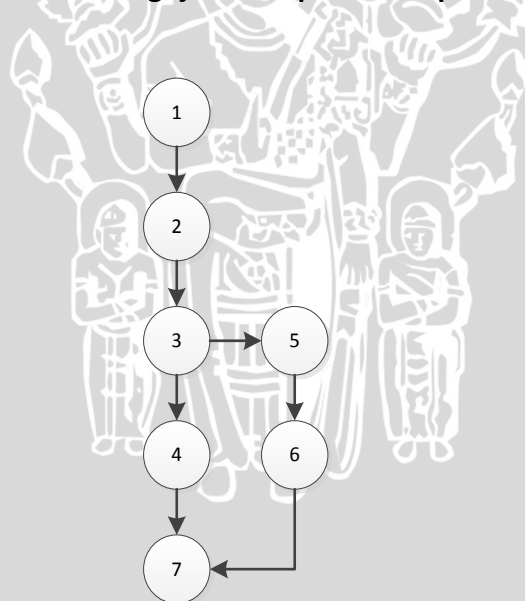
5.1 Hasil Pengujian dengan *White Box Testing*

Pengujian unit termasuk salah satu metode *white box testing* menggunakan teknik *basic path testing*. Pengujian dilakukan berdasarkan tingkat kompleksitas dari algoritma perancangan. Pengujian unit yang akan dilakukan meliputi prosedur-prosedur yang berperan inti dalam program.

5.1.1 Pengujian Pemilihan Pemain



Gambar 5.1 Pengujian unit pemilihan pemain



Gambar 5.2 *Flow graph* pemilihan pemain

Gambar 5.2 menunjukkan *flow graph* untuk unit pemilihan pemain. Untuk menghitung kompleksitas siklotomatik (*Cyclomatic complexity*), digunakan persamaan $V(G)=E-N+2$. E adalah *edge* atau jumlah garis yang ada pada *flow graph*, lalu N adalah jumlah *node*.

$$V(G)=E-N+2$$

$$V(G)=7-7+2$$

$V(G)=2$

Jalur Independen:

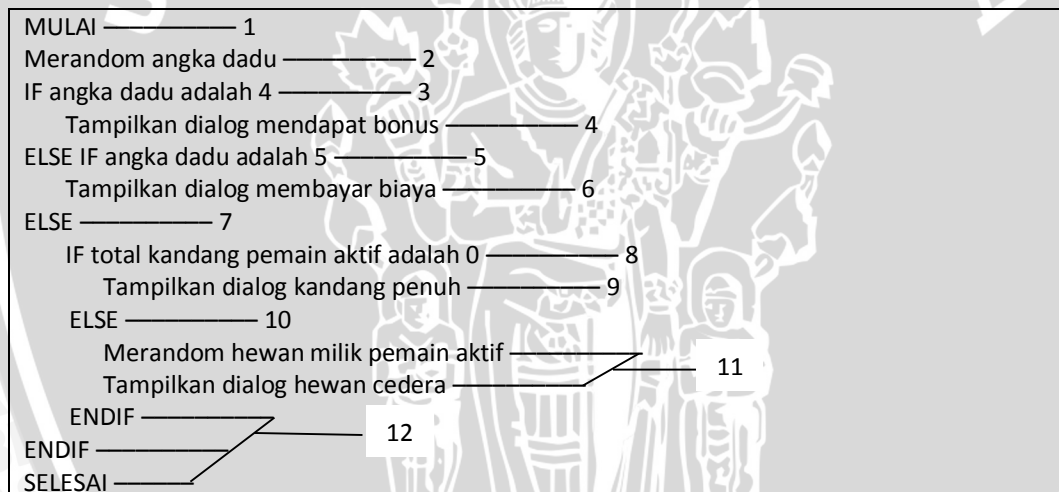
1. 1-2-3-4-7
2. 1-2-3-5-6-7

Tabel 5.1 adalah hasil pengujian pada unit pemilihan pemain dengan jalur yang didapatkan dari perhitungan kompleksitas siklomatik. Terdapat 2 jalur yang dapat diuji.

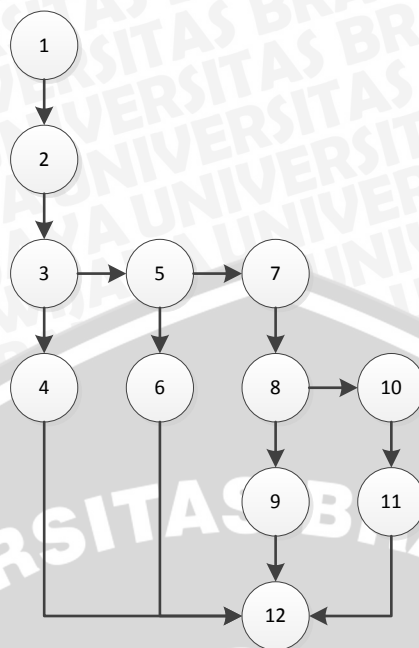
Tabel 5.1 Hasil pengujian unit pemilihan pemain

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Terdapat duplikasi pada karakter pemain yang berbeda	Muncul dialog <i>warning</i>	Muncul dialog <i>warning</i>
2	Tidak terdapat duplikasi pada karakter pemain yang berbeda	Memuat <i>scene game</i>	Memuat <i>scene game</i>

5.1.2 Pengujian Putar Dadu



Gambar 5.3 Pengujian unit putar dadu



Gambar 5.4 Flow graph *putar dadu*

$$V(G)=E-N+2$$

$$V(G)=14-12+2$$

$$V(G)=4$$

Jalur Independen:

1. 1-2-3-4-12
2. 1-2-3-5-6-12
3. 1-2-3-5-7-8-9-12
4. 1-2-3-5-7-8-10-11-12

Tabel 5.2 adalah hasil pengujian pada unit *putar dadu* dengan jalur yang didapatkan dari perhitungan kompleksitas siklomatik. Terdapat 4 jalur yang dapat diuji.

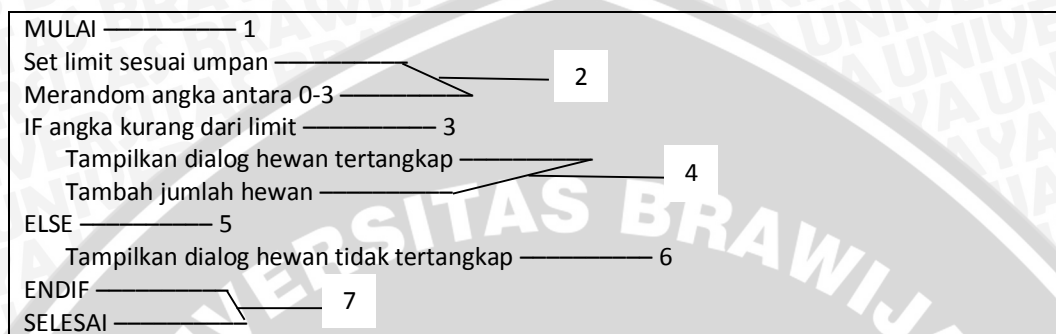
Tabel 5.2 Hasil pengujian unit *putar dadu*

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Angka dadu yang muncul adalah 4	Muncul dialog mendapat bonus	Muncul dialog mendapat bonus
2	Angka dadu yang muncul adalah 5	Muncul dialog membayar biaya perawatan	Muncul dialog membayar biaya perawatan
3	Angka dadu yang muncul adalah 1, 2, atau 3 dan total kandang pemain aktif adalah 0	Muncul dialog kandang telah terisi penuh	Muncul dialog kandang telah terisi penuh

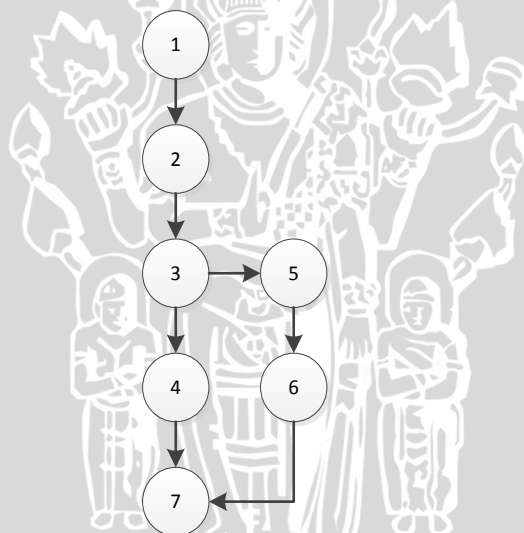
Tabel 5.2 Hasil pengujian unit putar dadu (lanjutan)

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
4	Angka dadu yang muncul adalah 1, 2, atau 3 dan total kandang pemain aktif lebih dari 0	Muncul dialog selamatkan hewan cedera	Muncul dialog selamatkan hewan cedera

5.1.3 Pengujian Menangkap Hewan



Gambar 5.5 Pengujian unit menangkap hewan



Gambar 5.6 Flow graph menangkap hewan

$$V(G)=E-N+2$$

$$V(G)=7-7+2$$

$$V(G)=2$$

Jalur Independen:

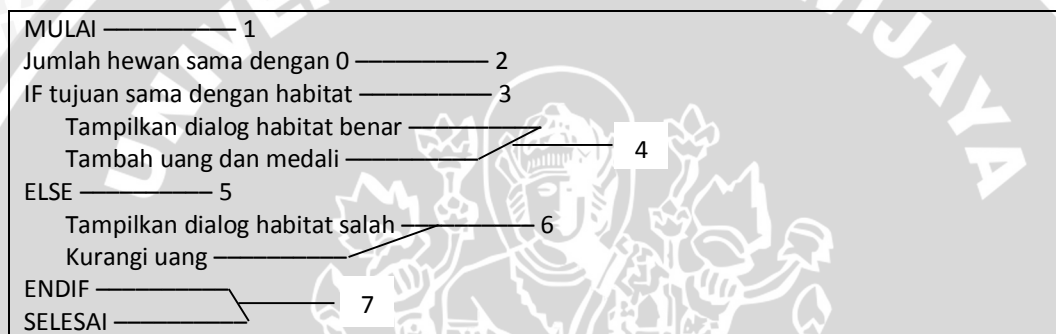
- 1-2-3-4-7
- 1-2-3-5-6-7

Tabel 5.3 adalah hasil pengujian pada unit menangkap hewan dengan jalur yang didapatkan dari perhitungan kompleksitas siklomatik. Terdapat 2 jalur yang dapat diuji.

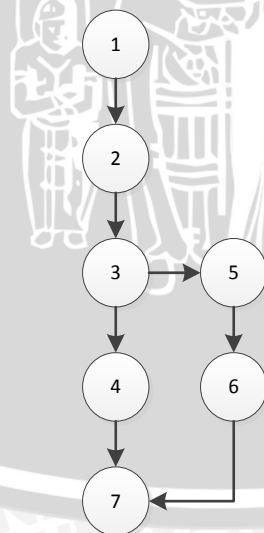
Tabel 5.3 Hasil pengujian unit menangkap hewan

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Angka yang di-random kurang dari limit	Muncul dialog hewan berhasil tertangkap, jumlah hewan bertambah	Muncul dialog hewan berhasil tertangkap, jumlah hewan bertambah
2	Angka yang di-random lebih dari limit	Muncul dialog hewan tidak berhasil tertangkap	Muncul dialog hewan tidak berhasil tertangkap

5.1.4 Pengujian Melepaskan Hewan



Gambar 5.7 Pengujian unit melepaskan hewan



Gambar 5.8 Flow graph melepaskan hewan

$$V(G)=E-N+2$$

$$V(G)=7-7+2$$

$$V(G)=2$$

Jalur Independen:

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

Tabel 5.3 adalah hasil pengujian pada unit melepaskan hewan dengan jalur yang didapatkan dari perhitungan kompleksitas siklomatik. Terdapat 2 jalur yang dapat diuji.

Tabel 5.4 Hasil pengujian unit melepaskan hewan

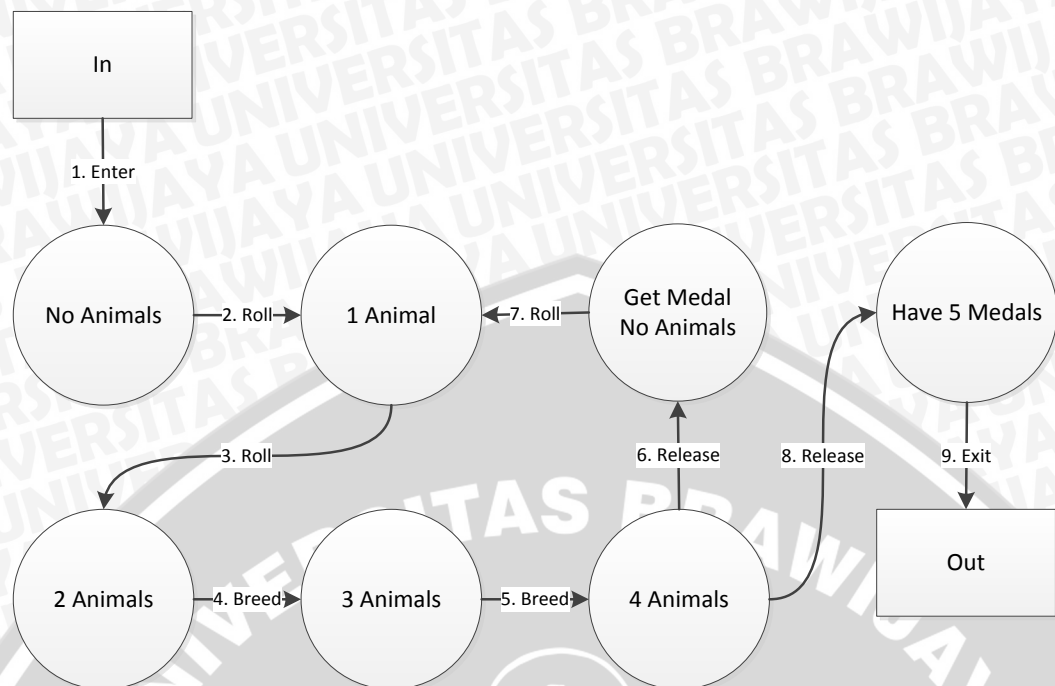
Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Tujuan pelepasan hewan sesuai dengan habitat hewan	Muncul dialog habitat benar, uang dan medali bertambah	Muncul dialog habitat benar, uang dan medali bertambah
2	Tujuan pelepasan hewan tidak sesuai dengan habitat hewan	Muncul dialog habitat salah, uang berkurang	Muncul dialog habitat salah, uang berkurang

5.2 Hasil Pengujian dengan *Black Box Testing*

Black box testing dilakukan dengan metode *test flow diagram* dan *focus testing*. *Test flow diagram* dilakukan untuk mengetahui hasil implementasi sudah memenuhi harapan atau belum. *Focus testing* dilakukan untuk mengetahui penerimaan pengguna terhadap *game* yang dibangun.

5.2.1 Test Flow Diagram

TFD adalah model grafis yang menunjukkan perilaku program dari perspektif pemain. TFD menyediakan pendekatan formal dalam mengetes desain.



Gambar 5.9 Pengujian *gameplay* menggunakan TFD

Jalur yang ada dihitung dengan menggunakan *minimum path generation*. Dengan cara ini pengujian dapat dilakukan dengan jumlah sedikit dan lebih menyeluruh. Dari Gambar 5.9, jalur yang terbentuk yaitu:

1-2-3-4-5-6-7-3-4-5-8-9

Test case yang dihasilkan yaitu:

1. Terminator "In", Event "Enter"
 - a. *Game* dimulai
 - b. Pemain memiliki kandang hewan
2. Event "Roll"
 - a. Pemain memutar dadu dan menangkap hewan
3. Event "Roll"
 - a. Pemain memutar dadu dan menangkap hewan
4. Event "Breed"
 - a. Pemain mengembangkan hewan dan hewan baru lahir
5. Event "Breed"
 - a. Pemain mengembangkan hewan dan hewan baru lahir
6. Event "Release"
 - a. Pemain melepaskan hewan ke habitat aslinya dan mendapatkan *reward*

7. Event “Roll”
 - a. Pemain memutar dadu dan menangkap hewan
8. Event “Release”
 - a. Pemain melepaskan hewan ke habitat aslinya dan mendapatkan *reward*
9. Terminator “Out”, Event “Exit”
 - a. Menentukan pemenang
 - b. *Game* berakhir

Hasil pengujian dengan TFD dapat dilihat pada Tabel 5.5.

Tabel 5.5 Hasil pengujian dengan TFD

Jalur	Kasus Uji	Hasil yang Diharapkan	Hasil yang Didapatkan
1	Pengujian prosedur pada <i>gameplay</i>	<i>Game</i> berjalan lancar sesuai fungsinya tanpa adanya <i>bug</i> pada tiap prosedur	<i>Game</i> berjalan lancar sesuai fungsinya tanpa adanya <i>bug</i> pada tiap prosedur

5.2.2 Focus Testing

Focus testing adalah pengujian kepada pengguna. *Focus testing* dilakukan untuk mengetahui penerimaan pengguna terhadap *game*, yaitu tentang ketertarikan pengguna dan apakah *game* dapat mengenalkan kepada persebaran fauna di Indonesia. Pengujian dilakukan dengan pengumpulan data menggunakan angket pada target pengguna berumur di atas tujuh tahun. Hasil angket yang diisi oleh 25 responden terlihat pada Tabel 5.6.

Tabel 5.6 Hasil *focus testing*

No	Pernyataan	Jawaban				
		1	2	3	4	5
1.	Game ini menyenangkan	0	0	0	15	10
2.	Tampilan grafis game ini menarik	0	0	2	4	9
3.	Audio game sudah sesuai	0	0	0	9	15
4.	Anda tidak mengalami kesulitan mengoperasikan game	0	0	1	10	14
5.	Harga-harga item pada game ini sudah sesuai	0	0	2	15	8
6.	Alur permainan game ini menarik	0	0	2	11	12
7.	Anda ingin melanjutkan game ini hingga selesai	0	0	1	13	11
8.	Game mengenalkan Anda tentang fauna di Indonesia	0	0	0	3	22

Tabel 5.6 Hasil *focus testing* (lanjutan)

No	Pernyataan	Jawaban				
		1	2	3	4	5
9.	Informasi fauna pada game dapat dipahami	0	0	0	9	16
Keterangan: 1: Sangat Tidak Setuju 2: Tidak Setuju 3: Sedang 4: Setuju 5: Sangat Setuju						

Pertanyaan angket terdiri dari pertanyaan yang menguji apakah *game* ini menyenangkan, yaitu pertanyaan nomor 1 sampai 7, dan pertanyaan yang menguji apakah *game* ini mengenalkan pada materi, yaitu pertanyaan nomor 8 dan 9. Pertanyaan nomor 1 sampai 7 menguji *game interface*, *game mechanics*, dan *gameplay*.

5.3 Analisis Pengujian

Berikut ini akan dijelaskan mengenai analisis pengujian yang telah dilakukan meliputi *white box testing*, *black box testing* menggunakan TFD, dan *focus testing*.

5.3.1 Hasil Analisis *White Box Testing*

Berdasarkan hasil pengujian unit yang dilakukan, jumlah jalur pada logika setiap *method* sesuai dengan perhitungan *cyclomatic complexity*. Kasus uji yang didapatkan sesuai dengan jumlah jalur dan menampilkan hasil sesuai dengan harapan. Dari hasil tersebut dapat ditarik kesimpulan bahwa tiap unit modul dari program sudah memenuhi fungsi program yang diharapkan.

5.3.2 Hasil Analisis *Black Box Testing*

Berdasarkan pengujian terhadap prosedur secara *black box* pada *game* “Nusantara Rescue Zoo”, output yang dihasilkan pada program telah sesuai dengan yang diharapkan. Dari hasil tersebut dapat disimpulkan bahwa hasil implementasi dari *game* telah memenuhi kebutuhan fungsionalitas yang diharapkan.

5.3.3 Hasil Analisis *Focus Testing*

Pengujian dilaksanakan dengan mengambil data dari angket pengguna. Hasil analisis dapat dilihat pada Tabel 5.7. Mayoritas responden sebanyak 60% menganggap *game* ini menyenangkan. Mayoritas responden sebanyak 76% menyatakan tampilan grafis *game* sangat menarik. Mayoritas responden sebanyak 60% menyatakan audio sangat sesuai dengan *game*. Mayoritas responden sebanyak 56% tidak mengalami kesulitan mengoperasikan *game*. Mayoritas

responden sebanyak 60% menyatakan harga *item* dalam *game* sudah sesuai. Mayoritas responden sebanyak 48% menyatakan alur permainan *game* ini sangat menarik. Mayoritas responden sebanyak 52% ingin melanjutkan *game* ini hingga selesai.

Dari hasil angket nomor 2 dan 3 dapat terlihat bahwa *game interface* dapat diterima dengan baik. Hasil angket nomor 4 dan 5 menunjukkan bahwa *game mechanics* sudah berjalan dengan baik. Hasil angket nomor 6 dan 7 menunjukkan bahwa *gameplay* dari *game* ini menarik. Secara keseluruhan dapat disimpulkan bahwa *game* yang dibangun menarik dan menyenangkan.

Sedangkan untuk pengujian mengenai materi yang ada dalam *game*, mayoritas responden sebanyak 88% menyatakan *game* ini sangat mengenalkan tentang fauna di Indonesia. Mayoritas responden sebanyak 64% menyatakan informasi fauna pada *game* sangat dapat dipahami. Dari hasil ini dapat disimpulkan bahwa *game* yang dibangun dapat mengenalkan pengguna tentang persebaran fauna di Indonesia.

Tabel 5.7 Analisis hasil *focus testing*

No	Pernyataan	Jawaban				
		1	2	3	4	5
1.	Game ini menyenangkan	0	0	0	60%	40%
2.	Tampilan grafis game ini menarik	0	0	8%	16%	76%
3.	Audio game sudah sesuai	0	0	0	36%	60%
4.	Anda tidak mengalami kesulitan mengoperasikan game	0	0	4%	40%	56%
5.	Harga-harga item pada game ini sudah sesuai	0	0	8%	60%	32%
6.	Alur permainan game ini menarik	0	0	8%	44%	48%
7.	Anda ingin melanjutkan game ini hingga selesai	0	0	4%	52%	44%
8.	Game mengenalkan Anda tentang fauna di Indonesia	0	0	0	12%	88%
9.	Informasi fauna pada game dapat dipahami	0	0	0	36%	64%
Keterangan: 1: Sangat Tidak Setuju 2: Tidak Setuju 3: Sedang 4: Setuju 5: Sangat Setuju						

BAB 6 PENUTUP

6.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, pengujian, dan analisis, dapat ditarik kesimpulan sebagai berikut.

1. Berdasarkan proses perancangan, *gameplay* yang dipilih untuk mengembangkan *game* edukasi pengenalan fauna di Indonesia adalah *turn-based strategy*. Misi pemain adalah melepaskan hewan ke habitat aslinya dan mendapatkan medali.
2. *Game* edukasi pengenalan fauna di Indonesia berjudul “Nusantara Rescue Zoo” berhasil diimplemetasikan menggunakan *game engine* Unity dengan bahasa pemrograman C#.
3. Berdasarkan hasil pengujian terhadap program, seluruh kebutuhan fungsionalitas pada *game* edukasi pengenalan fauna di Indonesia sudah terpenuhi
4. Berdasarkan hasil pengujian penerimaan pengguna, *game* edukasi pengenalan fauna di Indonesia menarik dan dapat mengenalkan pengguna tentang persebaran fauna di Indonesia.

6.2 Saran

Berikut ini adalah saran yang dapat diberikan untuk penelitian selanjutnya.

1. *Gameplay* dibuat lebih interaktif dan melibatkan interaksi antar pemain yang lebih kuat.
2. Hewan ditambahkan sesuai keanekaragaman fauna yang ada di Indonesia.
3. Aspek pengenalan materi dibuat lebih mendalam.

DAFTAR PUSTAKA

Brathwaite, B. & Schreiber, I., 2009. *Challenges for Game Designers*. Boston: Course Technology.

Darmawan, D., 2012. *Inovasi Pendidikan: Pendekatan Praktik Teknologi Multimedia dan Pembelajaran Online*. Bandung: PT Remaja Rosdakarya.

Federoff, M. A., 2002. *Heuristics and Usability Guidelines for the Creation and Evaluation of Fun in Video Games*. Bloomington: Indiana University.

Ferdiana, R., 2006. *Membangun Aplikasi Smart Client dengan Visual C# dan Visual Web Developer Express*. Yogyakarta: ANDI.

Hidayat, M. L. & Sriyanto, 2015. Implementasi Metode Mind Mapping dan Metode Resitasi dalam Pelajaran IPS Materi Persebaran Fauna di Indonesia Kelas VII SMP IP Assalamah Ungaran Kabupaten Semarang. *Edu Geography*, 3(8), pp. 19-26.

Keesee, G. S., 2011. *Teaching and Learning Resources / Educational Games*. [Online]

Available at: <http://teachinglearningresources.pbworks.com/w/page/35130965/Educational%20Games>

[Diakses 20 Desember 2015].

Pressman, R. S. P., 2001. *Software Engineering, A Practitioner's Approach*. 5th penyunt. New York: Thomas Casson.

ProFauna, 2012. *Apa itu PROFAUNA? | ProFauna Indonesia*. [Online] Available at: <http://www.profauna.net/id/tentang-profauna/apa-itu-profauna#.Vqg4EFJKXdQ>

[Diakses 23 Desember 2015].

Roedavan, R., 2014. *UNITY Tutorial Game Engine*. Bandung: Penerbit Informatika.

Rogers, S., 2010. *Level Up! The Guide to Great Video Game Design*. West Sussex: John Wiley & Sons Ltd.

Schreiber, I., 2009. *Game Design Concepts*, New York: Creative Commons Attribution 3.0.

Schultz, C. P., Bryant, R. & Langdell, T., 2005. *Game Testing All in One*. Boston: Thomson Course Technology PTR.

Tim Abdi Guru, 2007. *IPS Geografi untuk SMP Kelas VIII Jilid 2*. Jakarta: Penerbit Erlangga.

WWF Indonesia, 2015. *Spesies | WWF Indonesia*. [Online] Available at: <http://www.wwf.or.id/program/spesies/>

[Diakses 20 Desember 2015].

LAMPIRAN A KUESIONER PRA PENELITIAN

Berilah tanda silang (X) pada pilihan jawaban yang kalian anggap sesuai. Angket ini tidak berpengaruh pada nilai sekolah, maka isilah sesuai dengan keadaan yang sebenarnya!

1. Apakah menurutmu materi IPS tentang “Persebaran Fauna di Indonesia” penting?
 - a. Penting
 - b. Tidak penting
2. Bagaimanakah tingkat kesulitan dari materi “Persebaran Fauna di Indonesia”?
 - a. Mudah
 - b. Sedang
 - c. Sulit
 - d. Sangat sulit
3. Apa yang membuat materi tersebut sulit?
 - a. Materi tidak menarik
 - b. Terlalu banyak yang harus dihafalkan
 - c. Cara penyampaian materi kurang menarik
 - d. Lain-lain :
4. Metode apa yang dilakukan oleh guru di kelas untuk menyampaikan materi tersebut?
 - a. Penjelasan secara lisan (ceramah)
 - b. Diskusi
 - c. Permainan
 - d. Praktikum
5. Metode pembelajaran seperti apa yang kamu inginkan untuk membantumu menguasai materi?

.....

LAMPIRAN B KUESIONER PENGUJIAN *FOCUS TESTING*

Berilah tanda silang (x) pada nilai yang anda anggap paling sesuai!

Ket :

1 = Sangat Tidak Setuju

2 = Tidak Setuju

3 = Sedang

4 = Setuju

5 = Sangat Setuju

NO	PERNYATAAN	NILAI				
1	Game ini menyenangkan	1	2	3	4	5
2	Tampilan grafis game ini menarik	1	2	3	4	5
3	Audio game sudah sesuai	1	2	3	4	5
4	Anda tidak mengalami kesulitan mengoperasikan game	1	2	3	4	5
5	Harga-harga item pada game ini sudah sesuai	1	2	3	4	5
6	Alur permainan game ini menarik	1	2	3	4	5
7	Anda ingin melanjutkan game ini hingga selesai	1	2	3	4	5
8	Game mengenalkan Anda tentang fauna di Indonesia	1	2	3	4	5
9	Informasi fauna pada game dapat dipahami	1	2	3	4	5