

**RANCANG BANGUN PERMAINAN EDUKASI PENGENALAN
LAGU ANAK**

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun Oleh :

HENDRA IRAWAN ATUQO

NIM. 105060800111036

KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI

UNIVERSITAS BRAWIJAYA

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER

MALANG

2015

LEMBAR PERSETUJUAN

RANCANG BANGUN PERMAINAN EDUKASI PENGENALAN LAGU ANAK

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun Oleh:

HENDRA IRAWAN ATUQO

NIM. 105060800111036

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Erig M. Adams J, S.T., M.Kom.

Wibisono Sukmo Wardhono, S.T., M.T.

NIP. 19850410 201212 1 001

NIK. 8204040610091

LEMBAR PENGESAHAN

RANCANG BANGUN PERMAINAN EDUKASI PENGENALAN LAGU ANAK

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun Oleh:

HENDRA IRAWAN ATUQO

NIM. 105060800111036

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 23 juni 2015

Penguji I

Dr. Eng Herman Tolle, ST., MT.

NIP. 197408232000121001

Penguji II

Drs. Marji, MT.

NIP. 19670801 199203 1 001

Penguji III

Agi Putra Kharisma, ST., MT.

NIK. 2013048604301001

Mengetahui

Ketua Program Studi Teknik Informatika/Illmu Komputer

Drs. Marji, MT.

NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

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 dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan 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 yaitu UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70.

Undang-undang No. 20 Tahun 2003 Pasal 25 ayat 2 yang berisi “Lulusan perguruan tinggi yang karya ilmiahnya digunakan untuk memperoleh gelar akademik, profesi, atau vokasi terbukti merupakan jiplakan dicabut gelarnya.” dan Pasal 70 yang berisi “Lulusan yang karya ilmiah yang digunakannya untuk mendapatkan gelar akademik, profesi, atau vokasi sebagaimana dimaksud dalam Pasal 25 ayat (2) terbukti merupakan jiplakan dipidana dengan pidana penjara paling lama dua tahun dan/atau pidana denda paling banyak Rp200.000.000,00 (dua ratus juta rupiah).”

Malang, 23 Juni 2015

Mahasiswa,

Hendra Irawan Atuqo

105060800111036

KATA PENGANTAR

Puji dan syukur Penulis panjatkan kehadiran Allah SWT, karena hanya dengan rahmat dan bimbingannya Penulis dapat menyelesaikan tugas akhir dengan judul “Rancang Bangun Permainan Edukasi Pengenalan Lagu Anak” dengan baik. Tanpa rahmat dan bimbingan dari Tuhan Yang Maha Esa, maka niscaya Penulis tidak akan dapat menyelesaikan tugas akhir ini dengan baik dan tepat waktu.

Terima kasih pula Penulis sampaikan kepada pihak-pihak yang telah membantu Penulis dalam penyelesaian tugas akhir ini. Pihak-pihak tersebut antara lain:

1. Orang tua Penulis, Bambang Hariono dan Suwarti yang telah memberi motivasi, kasih sayang serta dukungan moril dan materil kepada Penulis. Kakak Winda Putri Lestari S.H, adik Ahmad Rizal M, dan Haris Rosyid R yang telah memberikan semangat dari awal sampai akhir pengerjaan tugas akhir ini.
2. Bapak Drs. Marji, M.Si. dan Issa Arwani, ST., MT. selaku Ketua dan Sekretaris Program Studi Informatika serta segenap Bapak/Ibu Dosen, Staff Administrasi dan Perpustakaan Program Studi Teknik Informatika Universitas Brawijaya.
3. **Bapak Eriq M. Adams J, S.T., M.Kom dan Wibisono Sukmo Wardhono, S.T., M.T.** Selaku dosen pembimbing I dan dosen pembimbing II yang telah banyak memberikan bimbingan, ilmu dan saran dalam penyusunan tugas akhir ini.
4. Bapak Issa Arwani, ST., MT. selaku dosen pembimbing akademik yang telah memberikan bimbingan, ilmu dan saran selama penulis belajar.
5. Semua teman-teman angkatan 2009, dan 2010 terimakasih atas segala bantuannya selama menjadi mahasiswa.

6. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya tugas akhir ini.

Semoga jasa dan amal baik mendapatkan balasan dari Allah SWT. Ibarat tak ada gading yang tak retak, dengan segala kerendahan hati, penulis menyadari sepenuhnya bahwa skripsi ini masih belum sempurna. Oleh karena itu kritik dan saran untuk kesempurnaan skripsi ini, senantiasa penulis harapkan dari berbagai pihak.

Malang, 23 Juni 2015

Penulis



ABSTRAK

Hendra Irawan Atuqo. 2015. Rancang Bangun Permainan Edukasi Pengenalan Lagu Anak. Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang. Dosen Pembimbing : Eriq M. Adams J, S.T., M.Kom dan Wibisono Sukmo Wardhono, S.T., M.T.

Permainan edukasi pengenalan lagu anak ini merupakan sebuah permainan edukasi yang memberikan pengajaran tentang lagu-lagu anak dan cara memainkannya. Selain itu permainan edukasi ini dapat melatih indera pendengaran dan daya ingat anak karena *gameplay* nya yang mengharuskan pemain mengikuti lagu yang diputar. Alat musik yang terdapat pada *game* ini, hampir sama dengan piano yang jika ditekan dapat mengeluarkan suara akan tetapi memiliki *art* dan animasi yang berbeda. Tujuan dari permainan edukasi ini adalah mengenalkan lagu-lagu anak kepada anak. Pembuatan permainan edukasi ini dimulai dari desain *game* yang dibagi kedalam pencarian ide *game*, penyusunan elemen-elemen *game*, membuat *prototype*, dan menguji dengan *play testing*. Implementasi dari permainan ini menggunakan *game engine* Unity3D dengan bahasa pemrograman C# dan Adobe Photoshop CS6 untuk implementasi dari *art*. Pengujian dari permainan ini menggunakan pengujian unit dan integrasi yang menggunakan teknik pengujian *white box testing* dengan metode *basis path*, *test suite* dengan teknik pengujian *black box testing* yang menguji tiap proses yang ada pada *game* dan *usability testing* yang merupakan pengujian terhadap pengguna. Dari hasil pengujian unit dan integrasi didapat kesimpulan bahwa semua kasus uji yang telah dilakukan didapatkan hasil sesuai dengan yang diharapkan dan untuk pengujian dengan *test suite* didapat kesimpulan bahwa keseluruhan proses pada *game* telah valid. Untuk *usability testing* didapat kesimpulan aplikasi ini telah memenuhi kebutuhan yaitu dapat mengenalkan dan menunjukkan cara bermain lagu-lagu anak.

Kata Kunci : Permainan Edukasi, Lagu anak

ABSTRACT

Hendra Irawan Atuqo. 2015. *Design of Educational Games the Children's Songs Introduction*. Thesis Study Program Informatics/Computer Science, Information Technology Program and Computer Science, University of Brawijaya, Malang. Advisor: Eriq M. Adams J, S.T., M.Kom and Wibisono Sukmo Wardhono, S.T., M.T.

The children's songs introduction of educational game is educational game that give an explanation about children's songs and how to play it. Besides this educational game can train the sense of hearing and memory children because of the gameplay from the game that requires the players to follow the played song. The instrument music of the game is like piano that produce a sound when we press it but have different art and animation. The main purpose of this educational game is to introducing the children's songs to a child. The first step in constructing this educational game is game design that divide into searching game idea, set game elements, make prototype and play testing. The implementation of this game is using a game engine Unity3D with C# language programing and Adobe Photoshop CS6 for implementing art. The testing of this game is using a unit testing and integration testing that use white box testing technique with basis path method, test suite that use black box testing technique that will test each proses in the game and usability testing. The conclusion of unit testing and integration testing is all test case that already done has met expectations and the conclusion of test suite is all proses in the game is validated. The conclusion of usability testing is this application already accomplish the purpose and that purpose is introducing children's songs and giving a lesson how to play it.

Keywords : Educational Game, Children's Song

DAFTAR ISI

KATA PENGANTAR	i
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI	v
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
DAFTAR LAMPIRAN	xiii
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	4
BAB II	5
KAJIAN PUSTAKA DAN DASAR TEORI	5
2.1 <i>Game</i>	5
2.1.1 Game Edukasi	5
2.2 Lagu Anak	7
2.2.1 Sejarah Lagu Anak	7
2.2.2 Ciri-ciri Lagu Anak	8
2.2.3 Kualitas Lagu Anak	8

2.3	Unity 3D <i>Game Engine</i>	9
2.4	Adobe Photoshop	11
2.5	Audacity	13
2.6	<i>Iterative Rapid Prototyping</i>	14
2.7	Pengujian Perangkat Lunak.....	14
2.7.1	Black Box Testing.....	14
2.7.2	White Box Testing	15
BAB III		16
METODOLOGI PENELITIAN DAN DESAIN		16
3.1	Metodologi Penelitian	16
3.1.1	Studi Literatur	17
3.1.2	Desain Game	18
3.1.3	Implementasi Game	18
3.1.4	Evaluasi Game	19
3.2	Desain <i>Game</i>	19
3.2.1	Ide Game	19
3.2.2	Formal Element Game	20
3.2.2.1	<i>Player</i>	20
3.2.2.2	<i>Objectives</i>	20
3.2.2.3	<i>Rule</i>	20
3.2.2.4	<i>Game State</i>	21
3.2.2.5	<i>Information</i>	22
3.2.2.6	<i>Squencing</i>	22
3.2.2.7	<i>Game System</i>	22
3.2.3	Paper Prototyping.....	22



3.2.3.1	Perlengkapan <i>Game</i>	22
3.2.3.2	Cara Bermain	24
3.2.4	Digital Prototyping.....	28
3.2.4.1	<i>Digital Prototyping Level Game</i>	28
3.2.4.2	<i>Digital Prototyping Gameplay Utama</i>	30
3.2.4.3	<i>Digital Prototyping Gameplay Latihan</i>	31
3.2.5	Play Testing.....	32
3.2.5.1	Desain <i>Play Testing</i>	32
3.2.5.2	Hasil dan Analisis <i>Play Testing</i>	32
BAB IV		37
IMPLEMENTASI.....		37
4.1	Pemilihan Teknologi dan Platform	37
4.1.1	Pemilihan Teknologi Perangkat Keras.....	37
4.1.2	Pemilihan Teknologi Perangkat Lunak.....	38
4.2	Implementasi <i>Level Game</i>	39
4.2.1	Implementasi Fungsi Terbukanya Level.....	39
4.2.2	Implementasi Level Game Art.....	40
4.2.3	Implementasi Level Game Screen	42
4.3	Implementasi <i>Gameplay Utama</i>	43
4.3.1	Implementasi Fungsi Alat Musik Box	43
4.3.2	Implementasi Fungsi Pemutaran Lagu.....	44
4.3.3	Implementasi Fungsi Chance	46
4.3.4	Implementasi Fungsi Score dan Combo	47
4.3.5	Implementasi <i>Gameplay Utama Art</i>	49
4.3.6	Implementasi <i>Gameplay Utama Screen</i>	49



4.4	Implementasi <i>Gameplay</i> Latihan	50
4.4.1	Implementasi Fungsi Free Mode.....	50
4.4.2	Implementasi Fungsi Learn Song	50
4.4.3	Implementasi Fungsi Play Song.....	52
4.4.4	Implementasi <i>Gameplay</i> Latihan Art.....	53
4.4.5	Implementasi <i>Gameplay</i> Latihan Screen	53
BAB V.....		55
EVALUASI.....		55
5.1	Pengujian Unit.....	55
5.1.1	Pengujian Unit Alat Musik Box.....	55
5.1.2	Pengujian Unit Chance.....	57
5.1.3	Pengujian Unit Score dan Combo.....	59
5.1.4	Analisis Pengujian Unit	63
5.2	Pengujian Integrasi.....	63
5.2.1	Pengujian Integrasi Game Over	64
5.2.2	Analisis Pengujian Integrasi.....	66
5.3	Pengujian Dengan <i>Test Suite</i>	66
5.3.1	Hasil Test Suite	68
5.3.2	Analisis Test suite	73
5.4	<i>Usability Testing</i>	73
5.4.1	Hasil dan Analisis Usability Testing.....	74
BAB VI.....		79
KESIMPULAN DAN SARAN.....		79
6.1	Kesimpulan	79
6.2	Saran.....	80

DAFTAR PUSTAKA DP-1

LAMPIRAN L-1



DAFTAR GAMBAR

Gambar 2.1 Komponen inti Adobe Photoshop.....	13
Gambar 3.1 Diagram alir penelitian.....	16
Gambar 3.2 <i>Board Game</i>	23
Gambar 3.3 Kartu Nada.....	24
Gambar 3.4 Cara Bermain 1.....	24
Gambar 3.5 Cara Bermain 2.....	25
Gambar 3.6 Cara Bermain 3.....	25
Gambar 3.7 Cara Bermain 4.....	26
Gambar 3.8 Cara Bermain 5.....	26
Gambar 3.9 Cara Bermain 6.....	27
Gambar 3.10 Cara Bermain 7.....	27
Gambar 3.11 Cara Bermain 8.....	28
Gambar 3.12 <i>Digital Prototyping Level Game</i>	29
Gambar 3.13 <i>Digital Prototyping Gameplay Utama</i>	30
Gambar 3.14 <i>Digital Prototyping Gameplay Latihan</i>	31
Gambar 3.15 Hasil <i>play testing</i> pertanyaan pertama.....	33
Gambar 3.16 Hasil <i>play testing</i> pertanyaan kedua.....	34
Gambar 3.17 Hasil <i>play testing</i> pertanyaan ketiga.....	35
Gambar 3.18 Hasil <i>play testing</i> pertanyaan keempat.....	35
Gambar 3.19 Hasil <i>play testing</i> pertanyaan keenam.....	36
Gambar 4.1 Implementasi <i>Art Open Level</i>	41
Gambar 4.2 Implementasi <i>art locked level</i>	41
Gambar 4.3 Implementasi <i>play level screen</i>	42

Gambar 4.4 Implementasi <i>train level screen</i>	43
Gambar 4.5 <i>Art Alat Musik Box</i>	49
Gambar 4.6 <i>Gameplay Utama Screen</i>	50
Gambar 4.7 <i>Art Menu Latihan</i>	53
Gambar 4.8 Implementasi <i>Gameplay Latihan Screen</i>	54
Gambar 5.1 <i>Flow Graph Alat Musik Box</i>	56
Gambar 5.2 <i>Flow Graph Chance</i>	58
Gambar 5.3 <i>Flow Graph Score dan Combo</i>	61
Gambar 5.4 <i>Flow Graph Game Over</i>	65
Gambar 5.5 Hasil <i>usability testing</i> pertanyaan pertama	74
Gambar 5.6 Hasil <i>usability testing</i> pertanyaan kedua	75
Gambar 5.7 Hasil <i>usability testing</i> pertanyaan ketiga	76
Gambar 5.8 Hasil <i>usability testing</i> pertanyaan keempat	77
Gambar 5.9 Hasil <i>usability testing</i> pertanyaan kelima	77

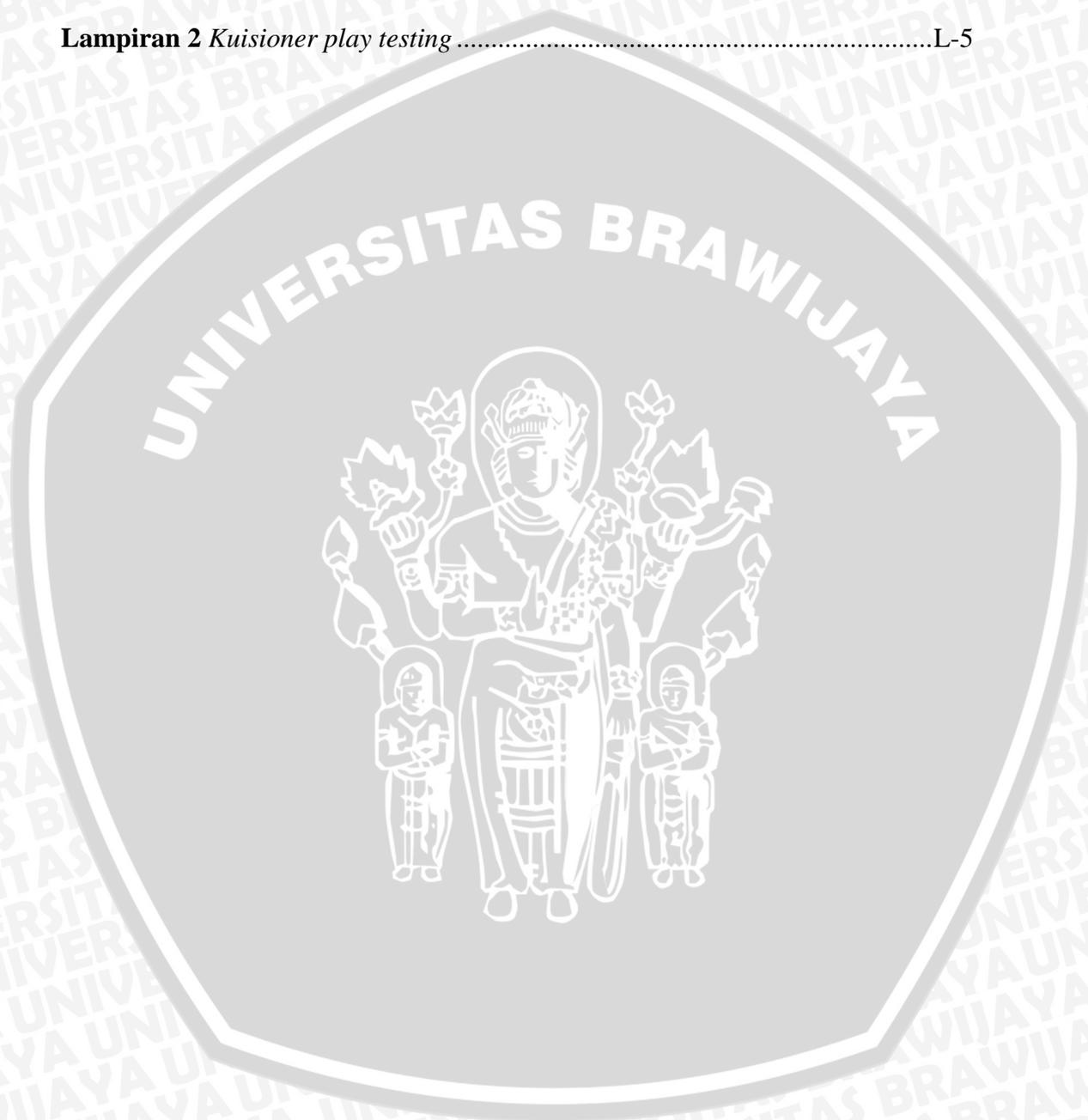
DAFTAR TABEL

Tabel 3.1 Program yang digunakan untuk implementasi	18
Tabel 3.2 <i>Game State</i> Permainan Edukasi Pengenalan Lagu Anak	21
Tabel 3.3 Rancangan level	28
Tabel 4.1 Spesifikasi PC (<i>Personal Computer</i>).	37
Tabel 4.2 Spesifikasi Android <i>device</i>	38
Tabel 4.3 Perangkat lunak yang digunakan.....	38
Tabel 4.4 Pseudocode Fungsi Terbukanya Level.....	40
Tabel 4.5 Pseudocode Fungsi Alat Musik <i>Box</i>	43
Tabel 4.6 Pseudocode Fungsi Pemutaran Lagu.....	44
Tabel 4.7 Pseudocode Fungsi <i>Chance</i>	46
Tabel 4.8 Pseudocode Fungsi <i>Score</i> dan <i>Combo</i>	47
Tabel 4.9 Pseudocode Fungsi <i>Learn Song</i>	51
Tabel 4.10 Pseudocode Fungsi <i>Play Song</i>	52
Tabel 5.1 Pemodelan <i>Flow Graph</i> Alogaritma Alat Musik <i>Box</i>	55
Tabel 5.2 Tabel Kasus Uji Untuk Pengujian Alat Musik <i>Box</i>	57
Tabel 5.3 Pemodelan <i>Flow Graph</i> Alogaritma <i>Chance</i>	57
Tabel 5.4 Tabel Kasus Uji Untuk Pengujian <i>Chance</i>	59
Tabel 5.5 Pemodelan <i>Flow Graph</i> Alogaritma <i>Score</i> dan <i>Combo</i>	59
Tabel 5.6 Tabel Kasus Uji Untuk Pengujian <i>Score</i> dan <i>Combo</i>	62
Tabel 5.7 Pemodelan <i>Flow Graph</i> Alogaritma <i>Game Over</i>	64
Tabel 5.8 Kasus Uji Untuk Pengujian <i>Game Over</i>	65
Tabel 5.9 Hasil <i>Test Suite</i>	68

DAFTAR LAMPIRAN

Lampiran 1 *Test suite*L-1

Lampiran 2 *Kuisisioner play testing*L-5



BAB I

PENDAHULUAN

1.1 Latar Belakang

Lagu ataupun musik merupakan bagian terpenting untuk menjaga kesehatan fisik dan mental. Disadari atau tidak dengan mendengarkan lagu, seseorang dapat menghadirkan suasana yang dapat mempengaruhi batinnya. Apakah suasana tersebut sedih ataupun bahagia, semua tergantung lagu yang didengar. Selain untuk menjaga kesehatan, musik ataupun lagu merupakan alat yang sempurna untuk meningkatkan kemampuan berbahasa, kelincahan dan keterampilan emosional pada anak baik di rumah, di sekolah, maupun saat bermain [DON-02].

Lagu anak merupakan pilihan lagu yang sempurna untuk membantu perkembangan anak, akan tetapi kepopulerannya saat ini telah kalah dengan lagu-lagu yang bersifat remaja ataupun dewasa. Fakta dan realita yang terjadi, semakin menjamurnya musik di blantika nusantara saat ini, khususnya dengan munculnya ratusan band di kalangan anak muda dan dewasa dengan aneka macam aliran dan judul lagu membuat lagu anak kian menghilang. Padahal kalau diteliti lebih lanjut lagu anak memiliki pesan moral yang lebih baik untuk anak dibandingkan lagu-lagu dewasa yang cenderung ke percintaan.

Salah satu cara mengenalkan lagu anak yang kian meredup kepopulerannya kepada anak adalah dengan *game*. Teknologi *game* dapat memotivasi pembelajaran dan melibatkan pemain, sehingga proses pembelajaran lebih menyenangkan [VIR-05]. Di sisi lain bahwa bermain *game* merupakan hal yang tidak asing lagi di kalangan anak-anak bahkan tidak jarang bermain *game* dijadikan sebuah hobi oleh anak. Hasil survey menunjukkan 92% anak usia 2-17 tahun memainkan *game* video dan *game* komputer [WAL-01]. Dalam pembuatannya, *game* yang dikhususkan untuk anak-anak biasanya memiliki visualisasi yang cerah dan meriah. Pemandangan gunung atau langit misalnya, dengan adanya visualisasi yang cerah

dan meriah dapat memberikan suasana gembira untuk anak. *Game* yang dikhususkan untuk anak-anak ini biasanya merupakan *game* edukasi.

Game edukasi merupakan *game* yang di maksudkan untuk memancing minat belajar anak terhadap materi pelajaran sambil bermain, sehingga dengan perasaan senang diharapkan anak bisa lebih mudah memahami materi pelajaran yang disajikan. Penggunaan *game* edukasi pun kian meningkat, hal ini dibuktikan dengan munculnya berbagai macam jenis *game* edukasi seperti *game* berhitung yang dimaksudkan untuk pembelajaran pelajaran matematika dengan cara yang menyenangkan, *game puzzel* gambar yang dimaksudkan untuk melatih visualisasi anak, *game* musik untuk melatih indra pendengaran anak dan masih banyak lagi. Semua *game* edukasi tersebut memiliki tema yang berbeda-beda, akan tetapi memiliki tujuan yang sama yakni sebagai media pembelajaran yang menyenangkan untuk anak-anak.

Pada *game* edukasi yang bertemakan musik, biasanya menggunakan lagu-lagu anak dengan tema yang gembira. Selain itu dengan menggunakan lagu anak *user* atau anak-anak diharapkan mendapatkan pesan moral yang terkandung dalam lagu anak tersebut. *Game* edukasi yang bertemakan musik biasanya memiliki *gameplay* menirukan lagu yang diputar. Kondisi menang jika *user* dapat menirukan dengan benar lagu yang diputar dan sebaliknya *user* kalah jika tidak bisa menirukan lagu tersebut. Tujuan utama *gameplay* yang seperti ini adalah membuat *user* dapat memainkan lagu anak-anak tersebut dengan baik dan benar.

Berdasarkan pada permasalahan diatas penulis tertarik untuk membuat *game* edukasi dengan tema musik dimana lagu-lagu anak yang akan menjadi topik utama pada *game* edukasi ini. Diharapkan dengan adanya *game* edukasi ini, *user* dapat belajar lagu anak dengan cara yang menyenangkan. Dari penelitian ini penulis mengangkat judul **“RANCANG BANGUN PERMAINAN EDUKASI PENGENALAN LAGU ANAK”**.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas maka didapat rumusan masalah yang meliputi:

1. Bagaimana mendesain *gameplay* permainan edukasi pengenalan lagu anak yang dapat dimainkan dengan mudah dan menyenangkan sesuai dengan penggunanya yaitu anak usia 4-7 tahun?
2. Bagaimana implementasi permainan edukasi pengenalan lagu anak?
3. Bagaimana mengetahui hasil dari usability permainan edukasi pengenalan lagu anak?

1.3 Batasan Masalah

Agar diperoleh hasil pembahasan yang sesuai dengan apa yang diharapkan, maka perlu diberikan pembatasan masalah pada pengembangan *game* ini, yaitu :

1. *Game* dirancang untuk *mobile game* dengan menggunakan *game engine* Unity3D serta memiliki tampilan grafis 2D.
2. Lagu yang akan digunakan pada *game* merupakan lagu anak-anak dengan tema yang menarik dan dapat dengan mudah dimainkan.

1.4 Tujuan Penelitian

Adapun tujuan dari tugas akhir ini adalah merancang, mengimplementasi, dan menguji permainan edukasi pengenalan lagu anak baik secara fungsionalitas maupun usability.

1.5 Manfaat Penelitian

Manfaat yang diperoleh dari perancangan *game* edukasi pengenalan lagu anak ini:

1. Meningkatkan pengetahuan tentang lagu-lagu anak yang ada.
2. Sarana alternatif belajar musik dengan cara yang menyenangkan.

3. Penerapan ilmu yang telah diperoleh penulis dari Program Studi Teknik Informatika Universitas Brawijaya.
4. Untuk mengembangkan bidang keilmuan penulis khususnya dibidang teknologi game.

1.6 Sistematika Penulisan

Sistematika pembahasan ditunjukkan untuk memberikan gambaran dan uraian dari penulisan skripsi ini secara garis besar yang meliputi beberapa bab sebagai berikut:

Bab I : Pendahuluan

Memuat latar belakang, rumusan masalah batasan masalah, tujuan, manfaat dan sistematika penelitian.

Bab II : Kajian Pustaka Dan Dasar Teori

Menguraikan tentang kajian pustaka dan dasar teori yang mendasari perancangan, pengembangan, implementasi, dan pengujian dari *game*.

Bab III : Metodologi Penelitian Dan Desain Game

Membahas metodologi yang akan digunakan dalam penelitian yang terdiri dari studi pustaka, analisis kebutuhan, metode desain, metode implementasi, metode pengujian serta pengambilan kesimpulan, dan saran dari penelitian “Rancang Bangun Permainan Edukasi Pengenalan Lagu Anak”.

Bab IV : Implementasi

Memuat implementasi dari Permainan Edukasi Pengenalan Lagu Anak yang berdasarkan metodologi dan desain yang telah dibuat.

Bab V : Evaluasi

Memuat proses hasil pengujian dan evaluasi terhadap sistem yang telah diimplementasi.

Bab VI : Kesimpulan Dan Saran

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan serta saran-saran untuk pengembangan lebih lanjut.

BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

2.1 *Game*

Game merupakan suatu sistem yang memiliki aturan-aturan tertentu dimana pemain akan terlibat di dalam suatu permasalahan sehingga dapat menghasilkan suatu hasil yang dapat diukur yaitu menang atau kalah [ZIM-03]. *Game* adalah sesuatu hal yang dimainkan dengan suatu aturan tertentu yang biasa digunakan untuk tujuan kesenangan dan dapat juga digunakan untuk tujuan pendidikan. *Game* umumnya melibatkan stimulasi mental, fisik, atau keduanya. Banyak *game* yang dapat membantu mengembangkan keterampilan praktis yang berfungsi sebagai latihan, atau melakukan peran pendidikan, simulational, atau psikologis. *Game* yang melakukan peran pendidikan biasanya disebut dengan *game* edukasi.

2.1.1 *Game* Edukasi

Game edukasi adalah permainan yang telah dirancang khusus untuk mengajar orang tentang suatu subjek tertentu, memperluas konsep, memperkuat pembangunan, memahami sebuah peristiwa historis atau budaya, atau membantu mereka dalam mempelajari keterampilan dalam bermain. Suatu *game* harus memiliki kriteria-kriteria tertentu agar bisa di sebut *game* edukasi. Perancang yang baik haruslah memenuhi kriteria dari *game* edukasi itu sendiri [HUR-09]. Berikut ini merupakan kriteria dari *game* edukasi, yaitu:

1. Nilai Keseluruhan (*Overall Value*)

Nilai keseluruhan dari suatu *game* terpusat pada desain dan panjang durasi *game*. Aplikasi ini dibangun dengan desain yang menarik dan interaktif. Untuk penentuan panjang durasi, aplikasi ini menggunakan fitur *timer*.

2. Dapat Digunakan (*Usability*)

Mudah digunakan dan diakses adalah poin penting bagi pembuat *game*. Aplikasi ini merancang sistem dengan interface yang *user friendly* sehingga *user* dengan mudah dapat mengakses aplikasi.

3. Keakuratan (*Accuracy*)

Keakuratan diartikan sebagai bagaimana kesuksesan model/gambaran sebuah *game* dapat dituangkan ke dalam percobaan atau perancangannya. Perancangan aplikasi ini harus sesuai dengan model *game* pada tahap perencanaan.

4. Kesesuaian (*Appropriateness*)

Kesesuaian dapat diartikan bagaimana isi dan desain *game* dapat diadaptasikan terhadap keperluan *user* dengan baik. Aplikasi ini menyediakan menu dan fitur yang diperlukan *user* untuk membantu pemahaman *user* dalam menggunakan aplikasi.

5. Relevan (*Relevance*)

Relevan artinya dapat mengaplikasikan isi *game* ke target *user*. Agar dapat relevan terhadap *user*, sistem harus membimbing mereka dalam pencapaian tujuan pembelajaran. Karena aplikasi ini ditujukan untuk anak-anak maka desain antarmuka harus sesuai dengan nuansa anak-anak, yaitu menampilkan warna-warna yang ceria.

6. Objektifitas (*Objectives*)

Objektifitas menentukan tujuan *user* dan kriteria dari kesuksesan atau kegagalan. Dalam aplikasi ini objektivitas adalah usaha untuk mempelajari hasil dari permainan.

7. Umpan Balik (*Feedback*)

Untuk membantu pemahaman *user* bahwa permainan (*performance*) mereka sesuai dengan objek *game* atau tidak, *feedback* harus disediakan.

Aplikasi ini menyajikan animasi dan efek suara yang mengindikasikan kesuksesan atau kegagalan permainan.

2.2 Lagu Anak

Lagu merupakan sebuah gubahan seni nada dan suara. Nada dan suara yang dihasilkan dari urutan, kombinasi, dan hubungan temporal, sehingga menciptakan gubahan musik yang memiliki kesatuan dan kesinambungan. Dengan kata lain, lagu dapat diartikan sebagai ragam nada atau suara yang berirama. Sementara lagu anak-anak adalah lagu-lagu yang diciptakan khusus untuk dinikmati dan dikonsumsi oleh anak-anak. Lagu anak-anak, selain dibuat untuk dinyanyikan, biasanya memiliki nilai moral yang sangat bermanfaat bagi anak kecil, baik bagi kesehariannya sendiri, maupun bagi pergaulannya. Lagu anak-anak biasanya membimbing dan memberikan pengetahuan baru akan sesuatu hal kepada si anak [ZEP-10].

2.2.1 Sejarah Lagu Anak

Pada era tahun 90an, banyak sekali beredar album lagu-lagu anak yang digandrungi oleh masyarakat. Penyanyi lagu-lagu tersebut seperti Bondan Prakoso, Joshua, Kak Seto, dan masih banyak lagi menjadi ikon lagu-lagu anak. Dalam segi promosi pun, banyak perusahaan lagu anak-anak mengeluarkan biaya yang tidak sedikit untuk perkembangan produknya. Bahkan kepopuleran lagu anak merambah sampai ke pendidikan. Lagu anak yang memiliki nilai moral yang baik biasanya dimasukkan ke dalam pelajaran siswa tingkat kanak-kanak.

Lagu-lagu anak yang pernah muncul dikalangan anak-anak tersebut, memang sempat naik daun. Namun entah mengapa, pada tahun 2000an lagu-lagu anak seakan-akan lenyap dari peredaran dan lama kelamaan anak-anak cenderung lebih memilih lagu-lagu Pop Dewasa sebagai lagu favorit mereka. Anak-anak yang dulu dengan mudah hafal lagu-lagu anak sekarang lebih memilih menghafal lagu-lagu Pop Dewasa yang lebih ke percintaan.

Para pencipta lagu anak, yang pernah terkenal seperti Ibu Kasur, Papa T-Bob, dan lain-lain, seakan-akan menghilang dan tidak pernah muncul kembali. Sebenarnya saat ini pun masih banyak pencipta lagu-lagu anak, namun nama

mereka tidak sebesar Ibu Kasur dan Papa T-Bob. Beberapa penyanyi cilik juga mulai bermunculan, namun eksistensi mereka kurang banyak mendapat tanggapan yang baik dari anak-anak [ZEP-10].

2.2.2 Ciri-ciri Lagu Anak

Lagu anak sebenarnya sama dengan lagu-lagu lain, hanya saja lagu anak sedikit berbeda dan memiliki ciri-ciri khusus. Berikut ciri-ciri lagu anak:

1. Memiliki bentuk yang sederhana.
2. Tema lagu disesuaikan dengan jiwa anak yang masih polos.
3. Bahasa yang digunakan sederhana sehingga mudah dipahami anak-anak.
4. Lompatan nada tidak terlalu jauh.
5. Isi lagu bersifat mendidik kearah positif, misalnya: cinta orang tua, mengagungkan nama Tuhan, cinta tanah air, lingkungan, dan sebagainya.
6. Contoh musik anak-anak: Balonku Ada Lima, Pok Ame-Ame, Kasih Ibu, Pelangi, dan lain-lain.

2.2.3 Kualitas Lagu Anak

Lagu anak adalah lagu yang pantas didengarkan dan dinyanyikan untuk anak-anak, dan selain mengandung unsur hiburan, akan lebih baik jika mengandung unsur pendidikan juga. Lagu anak harus memiliki kualitas yang baik agar penggunaannya mendapatkan hal-hal yang positif dari lagu anak tersebut [ZEP-10].

Kualitas dari sebuah lagu anak-anak bisa dilihat dari segi:

1. Nada / Bit

Menyenangkan dan lucu, tidak terlalu keras dan bit tidak terlalu cepat.

2. Lirik

Mudah dipahami dan Menggunakan kata-kata yang sederhana dan tidak terlalu panjang (Terutama lagu yang mengandung nilai pendidikan dan

moral), boleh panjang asalkan mengandung sebuah cerita yang menarik dan mudah dipahami anak-anak, dan memiliki kata-kata yang berbobot.

3. Tema lagu

Mengandung pesan moral dan nilai pendidikan. Pesan moral yaitu yang berguna bagi anak-anak (kebaikan, persahabatan, kerajinan dan lain-lain) dan tidak mengandung hal-hal yang hanya diperuntukkan untuk orang-orang dewasa (cinta, selingkuh, pertengkaran). Sedangkan pesan yang mengandung nilai pendidikan adalah sarana mempermudah anak-anak untuk belajar tentang sebuah materi mata pelajaran tertentu.

4. Atraktif

Bisa mengajak anak-anak untuk bergerak seperti menari, olah raga, bertepuk tangan, dan menggerakkan bagian tubuh mereka.

5. Tidak cengeng

Jika lagu tersebut cengeng maka bisa membuat anak-anak kehilangan semangat di masa kecilnya dan melemahkan mental anak-anak. Karena dunia anak-anak adalah dunia yang seharusnya penuh keceriaan.

6. Tidak menimbulkan kerancuan

Seperti yang pernah terjadi pada lagu anak-anak jaman dahulu. Kadang mengandung kerancuan makna.

7. Disukai anak-anak

2.3 Unity 3D Game Engine

Game engine adalah perangkat lunak yang dirancang untuk membuat dan mengembangkan *video game*. Fungsi utama yang diberikan oleh *game engine* meliputi *rendering* untuk 2D atau 3D *graphic*, *collision detection*, *sound*, *scripting*, animasi, *artificial intelligence*, *networking*, *memory management*, *threading*, dan *scene graph*. *Game engines* memberikan perangkat untuk *visual development* dengan tambahan komponen perangkat lunak yang dapat dipakai berulang kali.

Perangkat ini pada umumnya memberikan *integrated development environment* yang dapat mempermudah, serta mempercepat pengembangan *game*.

Game Engine menyediakan fitur yang paling penting dari dunia pemrograman *game* adalah perenderan 3D, jaringan, grafik, dan *scripting*. *Game Engine* juga menyediakan perenderan dari lingkungan *game*. Setiap *game* menggunakan sistem yang berbeda untuk mengatur bagaimana aspek visual dari *game* akan dibentuk. Dengan menciptakan lingkungan grafik yang konsisten dan mempopulasikan lingkungan itu dengan objek-objek yang mematuhi hukum fisika, *game engine* memungkinkan *game* untuk dapat berkembang sesuai dengan garis produksi dan jalan cerita yang masuk akal. *Game engine* mengenkapsulasi karakteristik dari dunia nyata, seperti waktu, gerakan, efek gravitasi, dan hukum fisika lainnya.

Salah satu *game engine* dengan fitur lengkap dan banyak digunakan oleh *developer game* saat ini adalah Unity 3D. Unity 3D merupakan alat *authoring* yang terintegrasi untuk menciptakan *video game* 3D atau konten interaktif lainnya seperti visualisasi arsitektur atau *real-time* animasi 3D. Lingkungan pengembangan Unity 3D yang berjalan pada Microsoft Windows dan Mac OS X, dan *game* yang dihasilkan dapat dijalankan pada Windows, Mac, Xbox 360, PlayStation 3, Wii, iPad, iPhone, serta *platform* Android.

Unity 3D memiliki dua versi yang berbeda, yang pertama adalah versi gratis dan kedua adalah versi berbayar. Perbedaan antara dua versi dari Unity 3D adalah pada fiturnya. Pada Unity 3D yang berbayar memiliki fitur yang lebih lengkap. Walaupun demikian, Unity 3D versi gratis sudah cukup untuk membuat *game* yang berkualitas [FIN-04].

Unity 3D memiliki banyak kelebihan dibanding *game engine* lain diantaranya adalah:

1. 2D dan 3D *game engine*

Selain untuk pembuatan *game* dengan grafis 3D, *software* ini juga dapat digunakan untuk membuat *game* dengan grafis 2D.

2. Navigasi yang *modern*

Sistem navigasi yang digunakan pada *software* ini adalah navigasi bebas. Hal ini sangat memudahkan pengguna terutama dalam pembuatan *game* dengan grafis 3D karena navigasinya tidak jauh berbeda dengan *software modeling* 3D seperti blender dan maya.

3. Penggunaan *scripting* yang mudah

Pada Unity 3D *scripting* dilakukan pada *software* yang berbeda, hal ini dimaksudkan agar *user* tidak bingung saat membuat *game*.

4. Multi *platform*

Unity 3D dapat *build* ke berbagai *platform* diantaranya Windows, Mac, Xbox 360, PlayStation 3, Wii, iPad, iPhone, serta *platform* Android.

5. Multi genre *game*

Unity 3D memberikan kebebasan kepada developer untuk membuat *game* dengan berbagai macam genre mulai dari FPS, TPS, RPG, dan lain-lain.

2.4 Adobe Photoshop

Photoshop adalah perangkat lunak editor citra buatan Adobe Systems yang dikhususkan untuk pengeditan foto/gambar dan pembuatan efek. Perangkat lunak ini banyak digunakan oleh fotografer digital dan perusahaan iklan sehingga dianggap sebagai pemimpin pasar (*market leader*) untuk perangkat lunak pengolah gambar/foto, dan, bersama Adobe Acrobat, dianggap sebagai produk terbaik yang pernah diproduksi oleh Adobe Systems. Versi kedelapan aplikasi ini disebut dengan nama Photoshop CS (*Creative Suite*), versi sembilan disebut Adobe Photoshop CS2, versi sepuluh disebut Adobe Photoshop CS3, versi kesebelas adalah Adobe Photoshop CS4 dan versi yang terakhir (ketigabelas) adalah Adobe Photoshop CS6. Photoshop tersedia untuk Microsoft Windows, Mac OS X, dan Mac OS; versi 9 ke atas juga dapat digunakan oleh sistem operasi lain seperti Linux dengan bantuan perangkat lunak tertentu seperti CrossOver [PER-04].

Adobe Photoshop memiliki komponen-komponen inti yang membangun *software* tersebut menjadi *software* pengolah citra yang berkualitas. Berikut merupakan komponent inti dari Adobe Photoshop:

1. *Title bar*

Menampilkan nama file yang sedang aktif dan bisa di gunakan untuk memindahkan posisi jendela photoshop pada dekstop.

2. *Menu bar*

Berisi deretan menu yang meliputi : *File, Edit, Image, Layer, Select, Filter, View, Windows, Help*.

3. *Ruler*

Atau yang disebut juga penggaris yang berguna untuk menentukan posisi serta ukuran suatu objek.

4. *Status bar*

Bagian ini menampilkan berbagai informasi tentang objek dan perkakas yang sedang di pilih.

5. *Drawing windows*

Daerah yang digunakan sebagai lembar kerja untuk berkreasi dengan foto atau gambar.

6. *Toolbox*

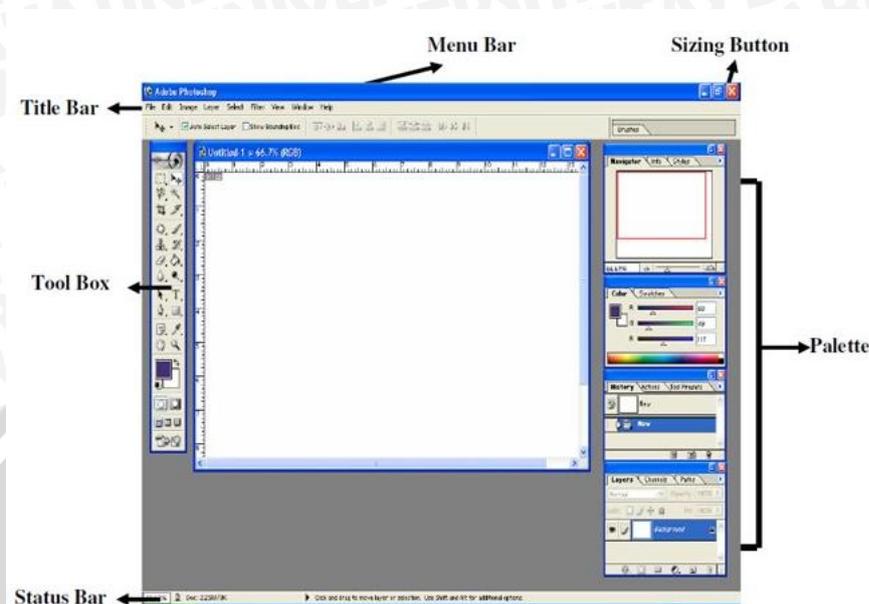
Komponen yang digunakan untuk memberikan akses perintah ke Photoshop.

7. *Pallette*

Komponen ini digunakan untuk memilih dan mengatur berbagai parameter.

8. *Sizing button*

Tombol ukuran yang berada pada kanan atas jendela Potoshop.



Gambar 2.1 Komponen inti Adobe Photoshop

2.5 Audacity

Audacity merupakan sebuah *software* yang berfungsi untuk editing audio (khususnya musik). Audacity dapat memotong, menambahkan, hingga menambahkan bermacam efek supaya hasil *output file* audio yang diedit sesuai dengan yang diinginkan pengguna. Intinya, kamu dapat menggunakan aplikasi ini layaknya seorang editor audio profesional, namun dengan metode yang lebih sederhana, mudah, dan juga praktis.

Audacity merupakan *freeware* alias *software* gratis, dimana aplikasi ini dapat dijalankan pada *platform* Linux, Windows, dan MacOS. *Interface* dari *software* ini diterjemahkan ke dalam berbagai bahasa dan beberapa kegunaan dari Audacity adalah [AUD-14]:

1. Merekam suara secara langsung
2. Merekam suara dari komputer.
3. Dapat mengedit berbagai macam file audio seperti WAV, AIFF, FLAC, MP2, MP3, OGG, dan lain-lain.

4. Memotong, mengcopy lagu atau file audio
5. Merubah kecepatan audio dan lagu.

2.6 *Iterative Rapid Prototyping*

Iterative rapid prototyping merupakan proses mendesain *game* yang dimulai dari mencari ide *game*, membuat *prototype* dimana *prototype* dibagi menjadi 2 yakni *paper prototype* dan *digital prototype*, mengimplementasi *prototype* dan mengevaluasinya. Proses-proses tersebut akan dilakukan berulang-ulang hingga mendapatkan hasil *game* yang diinginkan [SEC-09].

2.7 **Pengujian Perangkat Lunak**

Pengujian perangkat lunak sendiri bertujuan untuk menguji apakah sebuah perangkat lunak itu pantas untuk dipasarkan atau diumumkan ke khalayak umum. Pengujian perangkat lunak sendiri ada dua macam cara pengujiannya [NID-12].

Pengujian perangkat lunak memerlukan perancangan kasus uji (*test case*) agar dapat menemukan kesalahan dalam waktu singkat dan usaha minimum. Berbagai macam metode perancangan kasus uji telah berevovusi. Metode ini menyediakan pendekatan sistematis untuk pengujian oleh developer. Terlebih lagi metode ini menyediakan mekanisme yang dapat membantu memastikan kelengkapan dari pengujian dan menyediakan kemungkinan tinggi untuk menemukan kesalahan-kesalahan dalam perangkat lunak. Teknik atau metode perancangan kasus uji yang digunakan adalah *black box testing* dan *white box testing* [PRE-10:443].

2.7.1 **Black Box Testing**

Black box testing adalah teknik pengujian yang hanya berdasarkan kebutuhan dan spesifikasi. *Black box testing* juga disebut sebagai behavioral testing dan berfokus pada kebutuhan fungsi dari perangkat lunak [PRE-10:460]. Proses umu yang terjadi pada *black box testing* yaitu:

1. Kebutuhan atau spesifikasi dianalisa terlebih dahulu

2. Penentuan input valid terpilih berdasarkan spesifikasi untuk menentukan perangkat lunak berjalan dengan benar. Input yang tidak valid juga harus dipilih untuk memverifikasi bahwa perangkat lunak dapat mendeteksinya dan menanganinya dengan baik.
3. Penentuan output yang diharapkan sesuai input yang telah dipilih.
4. Pengujian dibuat dengan input yang telah dipilih
5. Pengujian dijalankan
6. Output yang sebenarnya dibanding dengan output yang diharapkan
7. Penentuan dibuat menyangkut perangkat lunak berfungsi sesuai dengan spesifikasi yang telah ditentukan.

2.7.2 *White Box Testing*

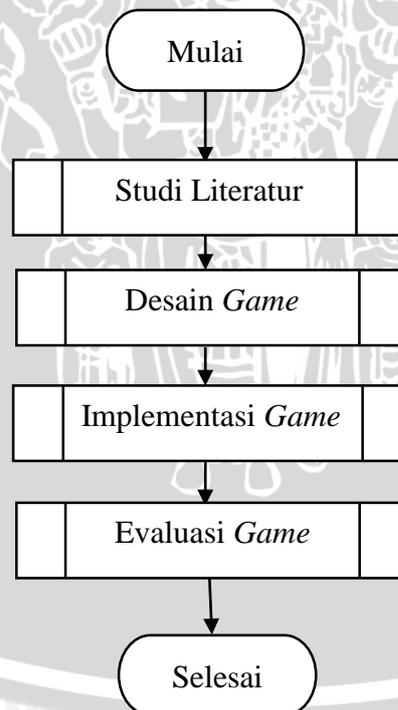
White box testing adalah teknik pengujian yang menguji berdasarkan jalur internal, struktur, dan implementasi dari perangkat lunak yang sedang diuji. *White box testing* adalah teknik pengujian yang menggunakan struktur kontrol dari prosedur yang terdapat dalam perancangan untuk membuat kasus uji. Ada dua jenis pengujian yang termasuk dalam *white box testing* yaitu *basis path testing* dan *control structure testing*. *Basis test case* yang dapat digunakan untuk menghitung kompleksitas desai prosedural untuk menghitung banyak jalur eksekusi. *Test case* tersebut digunakan untuk menjamin apakah tiap statement telah dijalankan minimal satu kali tiap pengujian. Dalam konteks *basis path*, nilai yang dihitung di dalam *cyclomatic complexity* menjelaskan banyak jumlah jalur independent yang harus disediakan dalam pengujian *basis path* [PRE - 10:445].

BAB III

METODOLOGI PENELITIAN DAN DESAIN

3.1 Metodologi Penelitian

Pada bab ini akan dijelaskan mengenai langkah – langkah yang akan dilakukan dalam pengerjaan penelitian. Pembuatan *game* ini dimulai dari Studi literatur, desain *game*, implementasi *game*, dan evaluasi *game*. Untuk metodologi yang digunakan dalam pembuatan *game* ini adalah metodologi *Iterative Rapid Prototyping*. Pada metode *Iterative Rapid Prototyping* ini desain dari *game* akan dilakukan berulang-ulang dan apabila telah mendapatkan desain yang diinginkan maka akan dilanjutkan ke proses implementasi. Berikut merupakan diagram alir dari pelaksanaan penelitian secara keseluruhan dapat dilihat pada gambar 3.1.



Gambar 3.1 Diagram alir penelitian.

Pada gambar 3.1, awal penelitian adalah studi literatur dimana referensi untuk membuat rancangan berupa materi pembelajaran dan *gameplay* yang sudah ada. Untuk alur yang kedua adalah desain dari game dimana pada desain akan diawali dengan mencari ide *game* yang akan dibuat dan dilanjutkan dengan membuat *prototype gameplay* dari ide tersebut. Alur selanjutnya adalah mengimplementasi *prototype* yang telah dibuat dan mengevaluasinya.

3.1.1 Studi Literatur

Studi literatur yaitu mengumpulkan data dengan membaca dan mempelajari teori-teori dan literatur-literatur yang berkaitan dengan tema dari skripsi. Teori dan literatur tersebut diperoleh dari buku, jurnal, e-book, dan website. dalam studi literatur akan dikumpulkan data-data yang valid dan terpercaya sehingga aplikasi yang dibuat dapat bermanfaat bagi para penggunanya. Studi literatur yang digunakan dalam pembuatan *game* ini adalah:

1. *Game*
 - a. *Game* edukasi
2. Lagu Anak
 - a. Sejarah Lagu Anak
 - b. Ciri-ciri Lagu Anak
 - c. Kualitas Lagu Anak
3. Unity3D *game engine*
4. Adobe Photoshop
5. Audacity
6. *Iterative Rapid Prototyping*
7. Pengujian Perangkat Lunak
 - a. *Black Box Testing*



3.1.2 Desain Game

Desain *game* merupakan proses merancang dan mendesain sebuah *game*. Didalam sebuah desain *game* terdapat kebutuhan serta prosedur teknis dari *game* yang akan dibuat. Tujuan utama dari dibuatnya perancangan dan desain dari sebuah *game* adalah memberikan gambaran umum dari *game* tersebut kepada pengguna. Dengan adanya gambaran umum ini pengguna dapat mengetahui bagaimana pembuatan dan cara kerja *game* ini nantinya.

Pada desain *game* ini hal pertama yang akan dilakukan adalah mencari ide dari *game*. Ide dari *game* ini meliputi inti dari *gameplay* dan aturan aturan yang ada pada *game* nantinya. Setelah mendapatkan ide hal selanjutnya yang akan dilakukan adalah membuat formal elemen dari *game*. Formal elemen ini *player*, rule, objective, dan beberapa elemen *game* lainnya. Hal ketiga yang dilakukan dalam mendesain *game* adalah membuat *prototype*. Terdapat 2 *prototype* yang harus dibuat yaitu *paper prototype* dan *digital prototype*. *Paper prototype* dibuat pada kertas sedangkan *digital prototype* pada komputer dengan menggunakan *asset-asset* seadanya. Setelah pembuatan *prototype*, hal terakhir yang dilakukan dalam desain *game* ini adalah *play test* untuk menguji seberapa menyenangkannya *game* tersebut.

3.1.3 Implementasi Game

Implementasi dalam pembuatan *game* ini merupakan penerapan dari desain *game* yang telah dibuat. Implementasi yang dilakukan disini meliputi pemilihan teknologi dan *platform* dan implementasi dari *prototype* yang telah dibuat. Untuk program yang digunakan dalam implementasi ditunjukkan pada Tabel 3.1.

Tabel 3.1 Program yang digunakan untuk implementasi

Implementasi	Program
Desain art dan desain aset-aset pada permainan	Adobe Photoshop CS 5

Bahasa pemrograman dan aplikasinya	Monodevelop C#
Implementasi <i>gameplay</i> , level dan <i>screen</i>	Unity 3D <i>game engine</i>

3.1.4 Evaluasi Game

Evaluasi pada *game* ini dimaksudkan untuk menunjukkan apakah *game* telah bekerja sesuai dengan yang diharapkan ataukah *game* masih terdapat *bug* yang perlu diperbaiki. Evaluasi dibagi kedalam 3 bagian yaitu evaluasi dengan pengujian *black box*, evaluasi dengan pengujian *white box* dan pengujian yang dilakukan kepada pengguna *usability testing*.

3.2 Desain Game

Pada desain *game* ini akan dibagi kedalam beberapa bagian diantaranya adalah ide *game*, *formal element game*, *paper prototyping*, *digital prototyping*, dan *play test*.

3.2.1 Ide Game

Ide dari *game* ini berasal dari alat musik piano yang dapat dimainkan dengan cara menekan tiap not yang ada pada piano tersebut. Pada *game* ini pemain akan diberikan *box* yang jika ditekan dapat mengeluarkan suara. Suara yang dikeluarkan pada tiap *box* berbeda-beda sesuai dengan tangga nada lagu dasar yaitu do, re, mi, fa, so, la, si, dan do kecil. Untuk membedakan suara yang akan muncul, pada tiap *box* akan diberikan tanda berupa tangga nada lagu dasar.

Untuk aturan permainan pada *game* pengenalan lagu anak ini adalah pemain akan diberikan 5 kesempatan untuk mengikuti lagu yang diputar pada alat musik *box* yang telah dibuat. Lagu-lagu yang diputar pada *game* ini merupakan lagu-lagu anak yang pernah populer dimasanya. Jika pemain gagal mengikuti lagu yang diputar maka kesempatan pemain akan berkurang 1. Jika kesempatan yang diberikan telah habis maka *game* akan berakhir atau *game over*.

Didalam *game* ini nantinya akan diberikan menu latihan yang dapat membantu pemain saat pemain kesulitan dalam menyelesaikan lagu yang diberikan. Pada menu latihan ini terdapat beberapa pilihan yang dapat dipilih oleh pemain, yang pertama adalah menu *free* dimana pemain dapat memainkan alat musik *box* secara bebas, kedua menu *learn song* dimana pemain dapat memainkan lagu dengan cara mengikuti tanda-tanda yang ada dan yang terakhir pemain dapat melihat lagu akan dimainkan mulai dari awal sampai akhir.

3.2.2 *Formal Element Game*

Formal element game merupakan elemen-elemen inti penyusun dari *game*. Pada *game* pengenalan lagu anak ini *formal element game* dibagi kedalam beberapa bagian diantaranya adalah *player*, *objective*, *rule*, *game state*, *information*, *sequencing*, dan *game system*.

3.2.2.1 *Player*

Player yang dimaksud adalah jumlah pemain yang terdapat pada *game* pengenalan lagu anak ini. Pada *game* ini hanya terdapat 1 pemain yang bisa bermain. Pemain ini akan diberikan instruksi oleh *game system* yang nantinya akan menjadi lawan pemain itu sendiri. Interaksi yang terjadi antara pemain dan *game system* ini disebut *solitaire* (1 *player vs game system*).

3.2.2.2 *Objectives*

Objectives yang dimaksud adalah apa yang akan pemain lakukan pada *game* pengenalan lagu anak ini. Pada *game* ini pemain akan ditunjukkan bagaimana memainkan lagu anak oleh *game system*. Setelah ditunjukkan cara bermainnya, pemain harus menirukan dan memainkan lagu anak tersebut.

3.2.2.3 *Rule*

Rule merupakan aturan-aturan yang ada pada sebuah permainan. *Rule* dibagi kedalam 3 bagian diantaranya adalah *setup* dimana dilakukan pada awal permainan dan hanya dilakukan sekali, *progression of play* (apa yang terjadi selama permainan berlangsung), dan *resolution* (kondisi menang dan kalah). Dari 3 bagian tersebut maka *game* pengenalan lagu anak ini memiliki *rule* :

1. *Setup* : memilih lagu yang ingin dimainkan.
2. *Progression of play* : *game system* akan menunjukkan cara memainkan lagu anak yang telah dipilih oleh pemain dan pemain harus mengikutinya dengan benar.
3. *Resolution* : jika pemain dapat menyelesaikan lagu anak yang dipilih maka pemain menang dan jika pemain gagal mengikuti lagu anak yang dipilih maka pemain kalah.

3.2.2.4 *Game State*

Game state merupakan kumpulan dari *state* yang ada pada *game*. Pada *game* pengenalan lagu anak ini terdapat beberapa *state* yang dimulai dari *state* pemilihan lagu dan diakhiri dengan *state* pengumpulan *score* akhir. Untuk lebih jelasnya akan ditunjukkan pada Tabel 3.2.

Tabel 3.2 *Game State* Permainan Edukasi Pengenalan Lagu Anak

state	keterangan
<i>State</i> pemilihan lagu	<i>State</i> ini merupakan proses pemain memilih lagu
<i>State</i> pemutaran lagu oleh <i>game system</i>	Disini merupakan proses lagu dimainkan oleh <i>game system</i>
<i>State</i> pemain memainkan lagu	Disini merupakan proses pemain memainkan lagu sesuai dengan yang telah <i>game system</i> mainkan
<i>State</i> pengecekan benar atau salah	Disini proses pengecekan lagu yang dimainkan oleh pemain benar atau salah
<i>State</i> pengumpulan <i>score</i> akhir	Disini merupakan proses pengumpulan <i>score</i> akhir

3.2.2.5 Information

Information merupakan informasi-informasi yang diberikan oleh *game system* kepada pemain. Pada *game* pengenalan lagu anak ini informasi yang diberikan oleh *game system* kepada pemain berupa instruksi-instruksi seperti tanda *listen* yang berarti pemain harus memperhatikan *game system* yang sedang memainkan lagu dan tanda *your turn* yang berarti giliran pemain memainkan lagu.

3.2.2.6 Sequencing

Maksud dari *sequencing* disini adalah apakah permainan ini termasuk *real time* ataukah menggunakan mode *turn based*. Pada *game* pengenalan lagu anak ini mode yang digunakan adalah *turn based* yang berarti bergiliran. Pada *game* ini, *game system* akan memainkan lagu lebih dulu. Setelah itu giliran pemain yang memainkan lagu sesuai dengan lagu yang dimainkan oleh *game system*.

3.2.2.7 Game System

Terdapat 3 *game system* pada *game* pengenalan lagu anak ini yang pertama adalah level *system* dimana pemain memilih lagu. Pada awal permainan, level *system* ini hanya akan membuka 1 lagu dan jika pemain dapat menyelesaikan lagu tersebut maka lagu-lagu lain akan terbuka. *Game system* selanjutnya adalah *gameplay* utama *system* dan *gameplay* latihan *system*. *Gameplay* utama *system* adalah *gameplay* dari *game* ini sedangkan *gameplay* latihan merupakan *gameplay* tambahan untuk mempermudah pemain saat pemain kesulitan memainkan *game* ini.

3.2.3 Paper Prototyping

Sesuai dengan ide *game* yang telah didapat maka hal selanjutnya yang akan dilakukan adalah membuat *prototype game* dengan media kertas atau yang biasa disebut *paper prototyping*. *Paper prototyping* akan dibagi kedalam 2 bagian diantaranya adalah perlengkapan *game* dan cara bermain.

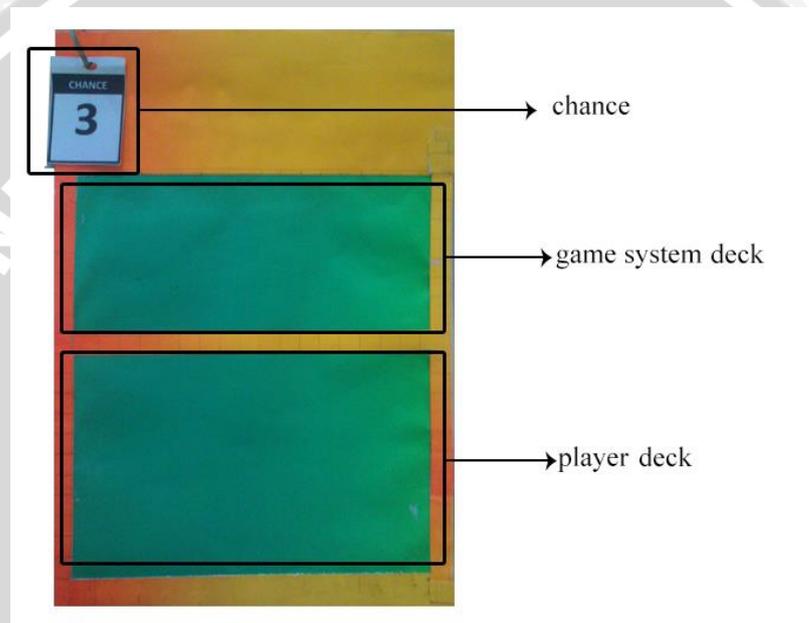
3.2.3.1 Perlengkapan Game

Perlengkapan *game* yang dibutuhkan untuk *paper prototyping game* pengenalan lagu anak adalah *board game* dan kartu nada. *Board game* merupakan

tempat pemain akan bermain sedangkan kartu nada merupakan alat yang digunakan untuk bermain. Berikut penjelasan dari *board game* dan kartu nada :

1. *Board game*

Terdapat 3 bagian penting penyusun dari *board game* diantaranya adalah *chance*, *game system deck*, dan *player deck*. Gambaran *board game* ditunjukkan pada Gambar 3.2.



Gambar 3.2 *Board Game*

Berikut penjelasan dari tiap bagian yang ada pada *board game* :

- Chance* merupakan kesempatan yang dimiliki oleh pemain untuk bermain.
- Game system deck* merupakan tempat *game system* meletakkan kartu nada
- Player deck* merupakan tempat pemain meletakkan kartu nada.

2. Kartu nada

Kartu nada merupakan kumpulan dari kartu-kartu yang mempresentasikan nada-nada lagu. Sesuai dengan tangga nada lagu dasar yang memiliki 7 nada, maka kartu nada juga memiliki 7 macam yaitu kartu DO, kartu RE, kartu MI, kartu FA, kartu SO, kartu LA, dan kartu SI. Gambaran kartu nada ditunjukkan pada Gambar 3.3.



Gambar 3.3 Kartu Nada

3.2.3.2 Cara Bermain

Cara bermain *game* pengenalan lagu anak pada *paper prototyping* ini adalah sebagai berikut :

1. *Game system* akan mengeset 1 kartu nada pada *game system deck*. Kartu nada tersebut diumpamakan sebagai nada awal dari sebuah lagu anak.



Gambar 3.4 Cara Bermain 1

2. Setelah diset kartu nada, pemain akan diperlihatkan *game system deck*. Setelah diperlihatkan pada pemain maka *game system deck* akan ditutup.



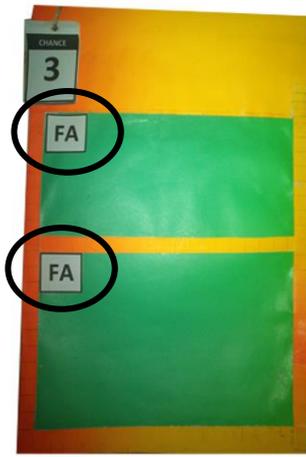
Gambar 3.5 Cara Bermain 2

3. Setelah *game system deck* tertutup, maka giliran pemain untuk mengeset kartu nada pada *player deck*. Kartu nada yang diset harus sesuai dengan yang ada pada *game system deck*.



Gambar 3.6 Cara Bermain 3

4. Setelah player mengeset kartu nada, proses yang dilakukan selanjutnya adalah pengecekan kartu nada. Pengecekan ini dilakukan dengan cara membuka penutup yang ada pada *game system deck*.



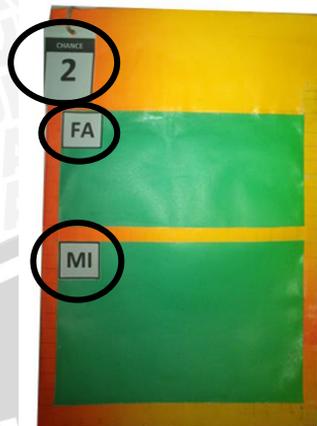
Gambar 3.7 Cara Bermain 4

5. Jika kartu nada yang ada pada *game system deck* dan *player deck* sama, maka permainan akan dilanjutkan dengan penambahan 1 kartu nada yang ada pada *game system deck* dan permainan akan kembali ke nomer 2.



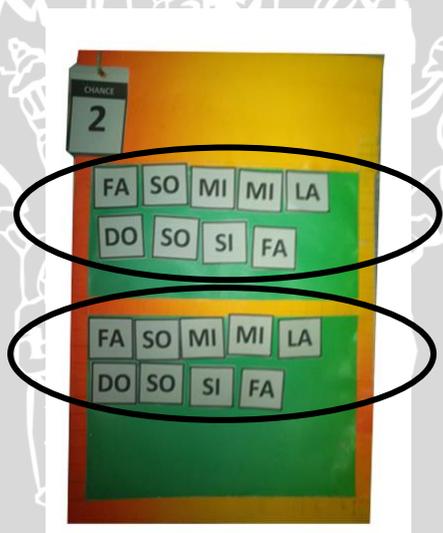
Gambar 3.8 Cara Bermain 5

6. Jika kartu nada yang ada pada *game system deck* dan *player deck* tidak sama, maka *chance* akan berkurang 1 dan permainan akan dilanjutkan tanpa penambahan kartu nada pada *game system deck* dan permainan akan kembali ke nomer 2.



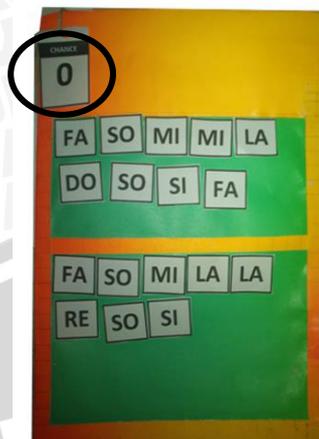
Gambar 3.9 Cara Bermain 6

7. Jika pemain berhasil mengikuti kartu nada yang dipasang oleh *game system* sampai akhir, maka pemain dianggap menang.



Gambar 3.10 Cara Bermain 7

8. Sedangkan jika *chance* yang dimiliki pemain habis, maka pemain dianggap kalah.



Gambar 3.11 Cara Bermain 8

3.2.4 Digital Prototyping

Setelah *paper prototyping* hal selanjutnya yang akan dilakukan adalah membuat *digital prototyping*. *Digital prototyping* merupakan implementasi awal dimana beberapa fungsi utama dari *game* mulai dibuat. Pada *digital prototyping* akan membahas 3 bagian dari *game* yakni *level game*, *gameplay* utama, dan *gameplay* latihan.

3.2.4.1 Digital Prototyping Level Game

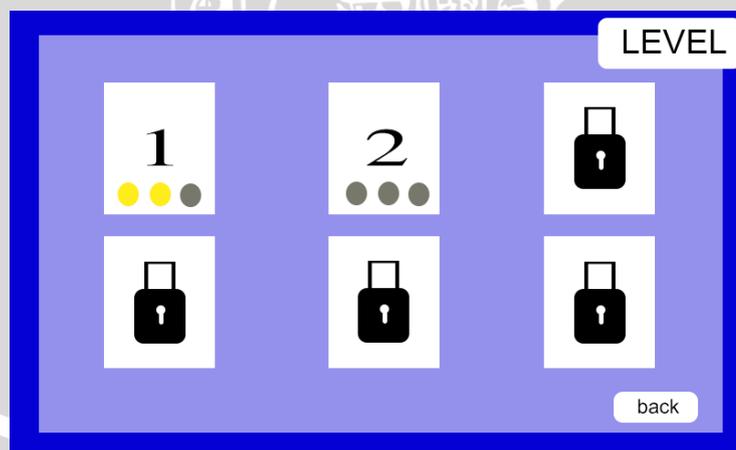
Digital prototyping level game ini memiliki 6 level yang berbeda, dimana pada setiap levelnya memiliki tema lagu anak yang berbeda-beda. Untuk lebih jelasnya akan ditunjukkan pada Tabel 3.3.

Tabel 3.3 Rancangan level

no	level	Keterangan
1.	Level 1	Lagu yang digunakan pada level 1 adalah “cicak-cicak didinding” yang memiliki jumlah nada sebanyak 31
2.	Level 2	Lagu yang digunakan pada level 2 adalah “twinkle-twinkle little star” yang memiliki jumlah nada sebanyak 47

3.	Level 3	Lagu yang digunakan pada level 3 adalah “kasih ibu” yang memiliki jumlah nada sebanyak 47
4.	Level 4	Lagu yang digunakan pada level 4 adalah “kebunku” yang memiliki jumlah nada sebanyak 51
5.	Level 5	Lagu yang digunakan pada level 5 adalah “balonku ada lima” yang memiliki jumlah nada sebanyak 64
6.	Level 6	Lagu yang digunakan pada level 4 adalah “ambilkan bulanbu” yang memiliki jumlah nada sebanyak 64

Dari Tabel 3.3 dapat disimpulkan bahwa untuk setiap level lagu yang dipakai adalah lagu-lagu anak dan untuk tingkat kesulitan level digunakan jumlah nada sebagai patokannya. Jumlah nada yang paling sedikit akan dijadikan sebagai level yang paling rendah yaitu 1 sedangkan nada yang paling banyak akan dijadikan level yang paling tinggi. Berikut gambaran *digital prototyping* level game yang ditunjukkan pada Gambar 3.12.



Gambar 3.12 *Digital Prototyping Level Game*

Pada *digital prototyping* level ini terdapat fungsi utama yaitu fungsi terbukanya level. Disini level yang terkunci dapat dibuka dengan cara mendapatkan bintang setidaknya 1 pada level sebelumnya. Misalkan, pada Gambar 3.12

ditunjukkan bahwa level 2 terbuka karena level 1 telah mendapatkan 2 bintang dan level 3 masih terkunci karena level 2 belum mendapatkan bintang.

3.2.4.2 *Digital Prototyping Gameplay Utama*

Digital prototyping gameplay utama ini merupakan implementasi awal dari *paper prototyping* yang telah menjelaskan *gameplay* dari *game* ini. Untuk lebih jelasnya akan ditunjukkan pada Gambar 3.13.



Gambar 3.13 *Digital Prototyping Gameplay Utama*

Pada *digital prototyping gameplay* utama ini mulai dibuat beberapa fungsi diantaranya adalah :

1. Fungsi alat musik *box*, pada fungsi ini alat musik *box* akan mengeluarkan suara sesuai dengan tanda nada yang ada pada *box* jika ditekan.
2. Fungsi pemutaran lagu, pada fungsi ini lagu akan diputar mulai dari 1 nada awal, lalu pemain harus mengikutinya. Jika pemain benar maka fungsi pemutaran lagu akan memutar 1 nada berikutnya dan jika salah maka fungsi pemutaran lagu akan mengulang nada awal tadi. Begitu pula seterusnya hingga nada terakhir lagu berhasil diikuti.
3. Fungsi *chance*, fungsi ini akan berhubungan dengan nyawa pemain. Jika pemain gagal mengikuti nada yang diputar maka *chance* akan berkurang dan jika *chance* 0 maka *game* berakhir.

4. Fungsi *score* dan *combo*, fungsi ini akan membahas tentang bagaimana pemain dapat mendapatkan *score* dan memperoleh *combo* untuk melipat gandakan *score*.

3.2.4.3 Digital Prototyping Gameplay Latihan

Digital prototyping gameplay latihan memiliki *gameplay* yang sama dengan *gameplay* utama akantetapi memiliki beberapa fungsi khusus untuk pemain berlatih. Untuk lebih jelasnya akan ditunjukkan pada Gambar 3.14.



Gambar 3.14 *Digital Prototyping Gameplay* Latihan

Pada *digital prototyping gameplay* latihan ini dibuat beberapa fungsi khusus untuk latihan pemain diantaranya adalah :

1. Fungsi *free mode*, fungsi ini sama dengan fungsi alat musik *box* yang ada pada *prototype gameplay* utama
2. Fungsi *learn song*, fungsi ini akan membuat pemain dapat berlatih memainkan lagu dengan cara mengikuti tanda-tanda yang diberikan. Tanda tersebut akan menunjukkan nada mana yang benar dari lagu yang dimainkan.
3. Fungsi *play song*, fungsi ini akan memainkan lagu mulai dari nada awal hingga nada yang terakhir.

3.2.5 Play Testing

Play testing merupakan pengujian yang dilakukan kepada pengguna untuk mengetahui apakah *game* dapat diterima oleh pengguna atau tidak. Pengujian ini dilakukan dengan cara memberikan kuisioner kepada responden yaitu responden. Tujuan dari pengujian sendiri adalah untuk mengetahui respon pengguna sehubungan dengan penggunaan aplikasi ini. Adapun langkah-langkah yang dilakukan dalam mengambil data untuk keperluan pengujian adalah dengan mempersilakan responden untuk bermain *game* terlebih dahulu. Setelah itu, memberikan kuisioner yang terdapat pada lampiran L-5 kepada responden untuk diisi.

3.2.5.1 Desain Play Testing

Untuk mendapatkan data yang dapat menguji tingkat kepuasan pengguna terhadap *game* ini dibutuhkan desain *play testing* yang baik. Berikut ini merupakan pertanyaan yang akan digunakan dalam kuisioner dari *play testing* :

1. Bagaimana dengan *gameplay* yang disajikan?
2. Bagaimana dengan efisiensi saat memainkan?
3. Bagaimana dengan tingkat kesulitan dari *game*?
4. Seberapa menyenangkankah *game* ini?
5. Bagaimana dengan manfaat yang didapat pemain dari *game* ini?

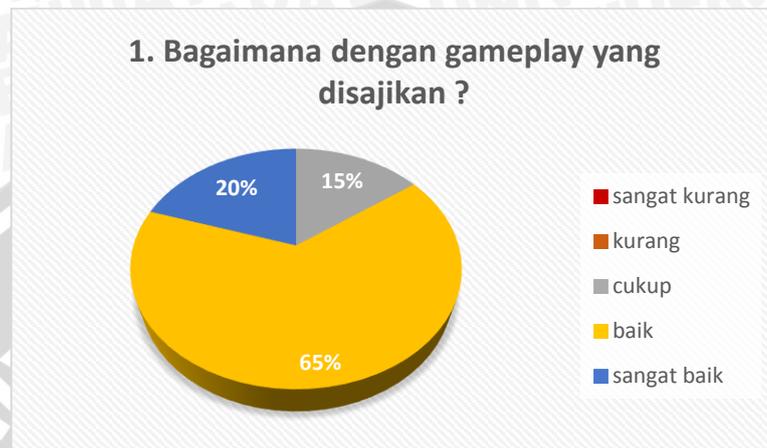
Pertanyaan yang disebutkan di atas akan dimasukkan ke dalam kuisioner dari *play testing* dan ditambahkan dengan beberapa kolom saran pada tiap pertanyaannya. Kolom saran dimaksudkan untuk memberikan masukan agar *game* bias lebih baik lagi.

3.2.5.2 Hasil dan Analisis Play Testing

Pada hasil dan analisis *play testing* ini akan membahas hasil dan analisa tiap jawaban dari pertanyaan yang telah dibuat pada *play testing*. Jawaban dari

pertanyaan ini didapat dari responden yaitu 20 responden. Berikut hasil dan analisa *play testing*:

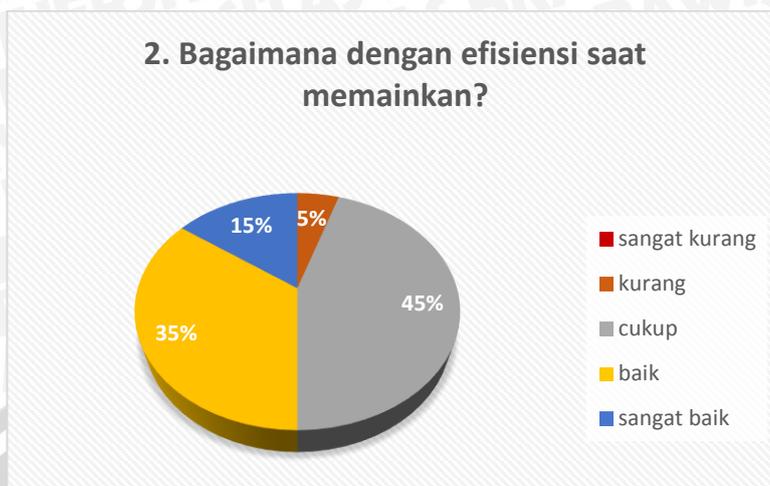
1. Pertanyaan pertama



Gambar 3.15 Hasil *play testing* pertanyaan pertama

Dari pertanyaan kuisioner yang pertama tentang *gameplay* dari *game* pengenalan lagu anak ini didapatkan hasil yang ditunjukkan pada Gambar 3.15. Dari gambar tersebut didapat kesimpulan bahwa sebagian besar responden menjawab baik. Dari keseluruhan jawaban didapat 65% responden menjawab baik, 20% menjawab sangat baik dan 15% menjawab cukup. Walaupun banyak responden menjawab baik tidak menutup kemungkinan bahwa *gameplay* akan dirubah agar lebih baik lagi karena banyak responden yang memberikan saran bahwa *gameplay* seharusnya lebih menarik lagi sehingga pemain tidak akan bosan untuk memainkan *game* pengenalan lagu anak ini.

2. Pertanyaan kedua

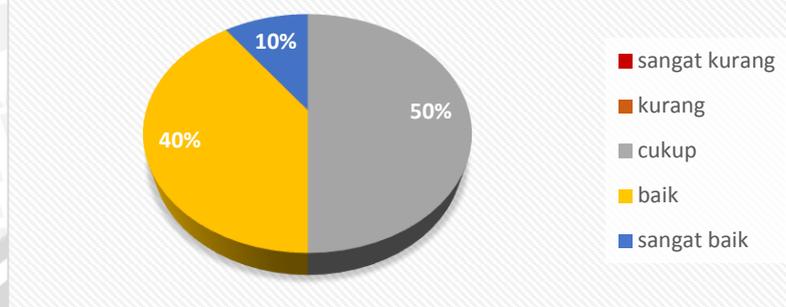


Gambar 3.16 Hasil *play testing* pertanyaan kedua

Dari pertanyaan kuisisioner yang kedua tentang efisiensi saat memainkan *game* ini didapatkan hasil yang ditunjukkan pada Gambar 3.16. Dari gambar tersebut didapat kesimpulan bahwa sebagian besar responden menjawab cukup dan baik. Dari keseluruhan jawaban didapat 5% menjawab kurang, 45% menjawab cukup, 35% menjawab baik dan sisanya yaitu 15% menjawab sangat baik. Berbeda dengan hasil pertanyaan yang pertama, hasil dari pertanyaan kedua ini terdapat responden yang menjawab kurang. Hal ini disebabkan pada *gameplay* pemain harus selalu mendengarkan nada mulai dari awal. Sebagai contohnya ketika pemain telah berada pada nada-nada akhir, pemain tetap harus mendengarkan nada mulai dari awal. Hal ini yang membuat pemain jenuh mengganggu sehingga *game* terasa kurang efisien. Keluar dari hal tersebut kebanyakan responden menjawab cukup dan baik dikarenakan *game* yang simple dan dapat dimainkan oleh siapa saja.

3. Pertanyaan ketiga

3. Bagaimana dengan tingkat kesulitan dari game?

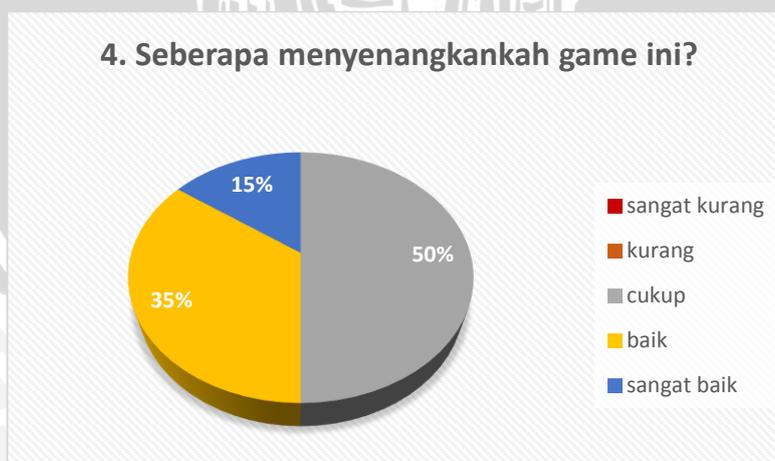


Gambar 3.17 Hasil *play testing* pertanyaan ketiga

Dari pertanyaan ketiga tentang tingkat kesulitan dari game ini didapatkan hasil yang ditunjukkan pada Gambar 3.17. Dari gambar tersebut didapat kesimpulan bahwa sebagian besar responden menjawab cukup dan baik. Dari keseluruhan jawaban didapatkan 50% responden menjawab cukup, 40% menjawab baik dan 10% menjawab sangat baik. Tidak seperti pertanyaan kedua, pada pertanyaan ketiga ini tidak ada responden yang menjawab kurang. Hal ini dapat diartikan bahwa tingkat kesulitan dari *game* telah sesuai dengan yang diharapkan.

4. Pertanyaan keempat

4. Seberapa menyenangkankah game ini?



Gambar 3.18 Hasil *play testing* pertanyaan keempat

Dari pertanyaan keempat tentang seberapa menyenangkan *game* ini didapatkan hasil yang ditunjukkan pada Gambar 3.18. Dari gambar tersebut didapatkan kesimpulan bahwa sebagian besar responden menjawab cukup. Dari keseluruhan jawaban didapatkan 50% responden menjawab cukup, 35% menjawab baik, dan 15% menjawab sangat baik. Pada pertanyaan keempat ini, tidak ada responden yang menjawab kurang atau sangat kurang. Hal ini dapat diartikan bahwa *game* telah memenuhi harapan karena dapat diterima oleh pengguna. Beberapa saran yang diberikan oleh responden agar *game* ini lebih menyenangkan adalah dengan memperbaiki *gameplay* dan memberikan lebih banyak pilihan lagu anak.

5. Pertanyaan kelima



Gambar 3.19 Hasil *play testing* pertanyaan keenam

Dari pertanyaan kelima tentang manfaat yang didapat dari memainkan *game* ini didapatkan hasil yang ditunjukkan pada Gambar 3.19. Dari gambar tersebut didapat kesimpulan bahwa sebagian besar responden menjawab baik. Dari keseluruhan jawaban 20% responden menjawab cukup, 60% menjawab baik dan 20% menjawab sangat baik. Pada pertanyaan keenam ini tidak ada responden yang menjawab kurang, hal ini disebabkan karena *game* ini memiliki manfaat yang pasti yaitu mengenalkan lagu-lagu anak kepada anak. Selain mengenalkan lagu anak responden juga memberikan pendapat bahwa *game* ini juga dapat melatih daya ingat anak.

BAB IV

IMPLEMENTASI

Bab ini akan membahas bagaimana implementasi dari *game* ini nantinya. Implementasi yang dimaksudkan pada *game* ini adalah implementasi dari fungsi-fungsi yang terdapat pada *game*. Bab implementasi ini akan dibagi kedalam beberapa bagian diantaranya adalah pemilihan teknologi dan platform, implementasi level *game*, implementasi *gameplay* utama, dan implementasi *gameplay* latihan.

4.1 Pemilihan Teknologi dan Platform

Pemilihan teknologi dan platform akan dibagi kedalam 2 bagian yakni, pemilihan teknologi perangkat keras dan pemilihan teknologi perangkat lunak.

4.1.1 Pemilihan Teknologi Perangkat Keras

Teknologi perangkat keras yang digunakan untuk membuat *game* ini adalah sebuah laptop atau pc sebagai pembuatan *game* dan android *device* sebagai alat *testing* dari *game* ini.

1. PC (*Personal Computer*)

Spesifikasi PC (*Personal Computer*) yang akan dipakai dalam pembuatan *game* ini akan dijelaskan pada Tabel 4.1.

Tabel 4.1 Spesifikasi PC (*Personal Computer*).

Personal Computer (PC)	
<i>Processor</i>	Intel Core i5 520M @ 2.53 GHz
<i>Memory (RAM)</i>	DDR3 1066 MHz SDRAM 2 x SODIMM socket for expansion up to 8GB SDRAM

<i>Harddisk</i>	2.5" 9.5mm SATA 500GB 5400rpm; 7200rpm
<i>Chipset</i>	Mobile Intel® HM55 Express Chipset
<i>Graphic Card</i>	NVIDIA® GeForce® GT 425M 1GB DDR3 VRAM
<i>Monitor</i>	LED backlit, Asus Splendid Video Intelligent Technology

2. *Android device*

Spesifikasi *Android device* yang digunakan untuk *game testing* akan dijelaskan pada Tabel 4.2.

Tabel 4.2 Spesifikasi *Android device*

Android device	
<i>Processor</i>	Dual-core 1 GHz Cortex-A9
<i>Memory(RAM)</i>	8 GB (5 GB user available), 512 MB RAM
<i>Screen Display</i>	LED-backlit LCD, capacitive touchscreen, 16M colors

4.1.2 Pemilihan Teknologi Perangkat Lunak

Dalam pengembangan *game* ini dibutuhkan berbagai macam perangkat lunak untuk mendukung dalam pembuatan *asset* dari *game*. Pada Tabel 4.3 akan dijelaskan berbagai macam perangkat lunak yang digunakan.

Tabel 4.3 Perangkat lunak yang digunakan



No	Perangkat Lunak	Keterangan
1	Windows 7 Home Premium	Sebagai OS (<i>Operating System</i>) dari PC, dimana digunakan untuk pengembangan <i>game</i> .
2	Android 4,2 ICS	Sebagai OS (<i>Operating System</i>) dari Android <i>device</i> , dimana digunakan sebagai pengujian dari <i>game</i> .
3	Unity3D 4.5	Sebagai <i>software</i> pembuat <i>game</i> (<i>game engine</i>).
4	Adobe Photoshop CS 5	Sebagai aplikasi pembuat dan pengolah gambar dari <i>game</i> .
5	Monodevelop C#	Sebagai aplikasi pemrograman yang menggunakan bahasa C#.
6	Audacity	Sebagai aplikasi pembuat dan pengolah suara dari <i>game</i> .

4.2 Implementasi Level *Game*

Pada implementasi level *game* ini akan dibagi kedalam beberapa bagian diantaranya adalah implementasi fungsi terbukanya level dan implementasi *levelgame art* dan *level game screen*.

4.2.1 Implementasi Fungsi Terbukanya Level

Pada level *game* terdapat fungsi utama yaitu fungsi terbukanya level. Pada *game* pengenalan lagu anak ini awalnya hanya memiliki 1 level yang terbuka. Setelah level yang terbuka tersebut berhasil dimainkan dan setidaknya mendapatkan 1 bintang maka level selanjutnya akan terbuka. Untuk lebih jelasnya akan ditunjukkan pada Tabel 4.4.

Tabel 4.4 Pseudocode Fungsi Terbukanya Level

Pseudocode Fungsi Terbukanya Level
<p>DEKLARASI AWAL</p> <pre>Public Int prevStarLevel; Public bool openLevel = false;</pre> <p>DESKRIPSI</p> <ol style="list-style-type: none"> 1. MULAI 2. Jika, <code>prevStarLevel > 0</code> 3. Maka, <code>openLevel = true;</code> 4. SELESAI

Penjelasan pseudocode dari fungsi terbukanya level pada Tabel 4.4 adalah :

1. Pada deklarasi awal terdapat 2 variabel yaitu `prevStarLevel` dan `openLevel`. `prevStarLevel` sebagai variabel yang menyimpan jumlah bintang pada level sebelumnya dan variabel `openLevel` sebagai variabel penunjuk apakah level terbuka atau masih terkunci.
2. Pada baris 2-3 ditunjukkan bahwa ketika `prevStarLevel` memiliki nilai yang lebih dari 0 maka `openLevel` bernilai `true` atau dengan kata lain level akan terbuka.

4.2.2 Implementasi Level Game Art

Pada implementasi level *game art* ini akan dibagi kedalam 2 bagian diantaranya adalah implementasi *art open* level dan implementasi *art locked* level.

1. Implementasi *art open* level

Sesuai dengan *prototype level game* dibutuhkan *art* yang dapat menunjukkan tema dari level. Diantaranya adalah judul lagu anak yang terdapat pada level tersebut, tema gambar dari lagu tersebut dan yang terakhir adalah bintang yang menunjukkan tingkat penyelesaian lagu tersebut. Pada Gambar 4.1 akan ditunjukkan implementasi *art open level*.



Gambar 4.1 Implementasi *Art Open Level*

2. Implementasi *art locked level*

Untuk implementasi *art locked level* hampir sama dengan implementasi *art open level* hanya saja tidak ada bintang pada *art* tersebut. Untuk judul dari *art locked level* ini adalah "locked !" dan gambarnya adalah gembok. Untuk lebih jelasnya akan ditunjukkan pada Gambar 4.2.



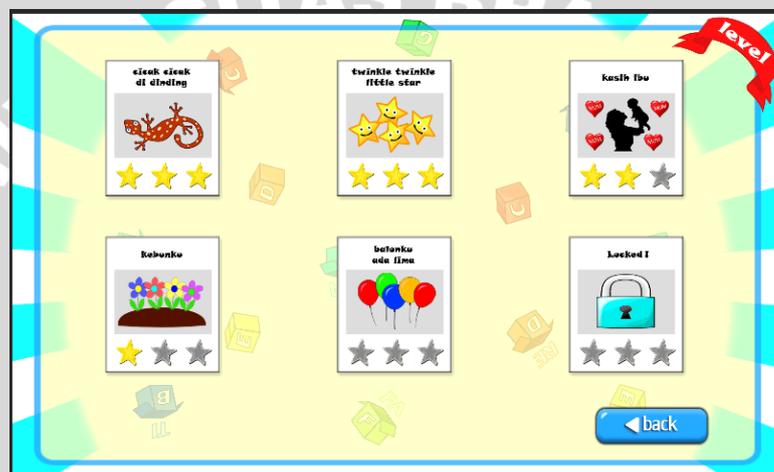
Gambar 4.2 Implementasi *art locked level*

4.2.3 Implementasi Level Game Screen

Implementasi level *game screen* ke dalam 2 bagian yaitu level *screen* *gameplay* utama dan level *screen* *gameplay* latihan.

1. Level *screen* *gameplay* utama

Pada implementasi ini terdapat menu level yang dapat dipilih oleh pemain dan menu level tersebut akan menuju ke *gameplay* utama. Selain itu ada menu *back* untuk kembali ke menu utama.



Gambar 4.3 Implementasi *play* level *screen*

2. Level *screen* *gameplay* latihan

Pada implementasi dari *train* level *screen* hampir sama dengan *play* level *screen*, yang membedakan adalah tidak adanya bintang sebagai penanda tingkat keberhasilan memainkan level tersebut. Implementasi *train* level *screen* akan ditunjukkan pada Gambar 4.4.



Gambar 4.4 Implementasi train level screen

4.3 Implementasi Gameplay Utama

Pada implementasi *gameplay* utama ini akan dibagi kedalam beberapa bagian diantaranya adalah implementasi fungsi alat musik *box*, implementasi fungsi pemutaran lagu, implementasi fungsi *chance*, implementasi fungsi *score*, dan *combo*, implementasi *gameplay* utama *art* dan implementasi *gameplay* utama *screen*.

4.3.1 Implementasi Fungsi Alat Musik *Box*

Fungsi alat musik *box* ini merupakan proses dimana pada setiap *box* yang ada pada *game* dapat mengeluarkan suara dan animasi jika mendapat *input* dari pemain yang berupa *touch*. Pada Tabel 4.5 akan ditunjukkan bagaimana proses tersebut dijalankan.

Tabel 4.5 Pseudocode Fungsi Alat Musik *Box*

Pseudocode Fungsi Alat Musik <i>Box</i>
<pre> DEKLARASI AWAL Public Int boxNumber; Animator anim; audioListener audio; </pre>

DESKRIPSI

```
5. MULAI
6. Jika, di beri input
7.     Maka, check boxNumber;
8.     anim = anim.boxNumber;
9.     audio = audio.boxNumber;
10.     SELESAI
```

Penjelasan pseudocode dari fungsi alat musik *box* pada Tabel 5.4 adalah:

1. Pada deklarasi terdapat variabel *boxNumber*, *anim*, dan *audio*. *boxNumber* sebagai identifikasi tiap *box music* dimulai dari *do* dengan *boxNumber* 1, *re* dengan *boxNumber* 2 dan seterusnya hingga *do* kesil dengan *boxNumber* 8. Variable *anim* merupakan variabel pembawa animasi dari *box* dan *audio* sebagai variabel yang membawa suara dari *box*
2. Baris 2-3 merupakan kondisi dimana jika terdapat *input touch* dari pemain maka akan mengecek *box* mana yang dipilih melalui *boxNumber*.
3. Baris 4-5 menunjukkan jika *boxNumber* telah diketahui maka akan mengeluarkan animasi dan suara dari *boxNumber* tersebut.

4.3.2 Implementasi Fungsi Pemutaran Lagu

Fungsi pemutaran lagu ini merupakan proses dimana lagu akan diputar mulai dari awal nada hingga akhir nada. Pada awal proses ini 1 nada awal akan diputar, lalu pemain harus mengikuti nada tersebut. Jika berhasil maka nada yang diputar akan bertambah 1 dan jika gagal maka nada yang diputar tidak bertambah. Proses tersebut akan diulang-ulang hingga nada terakhir. Untuk lebih jelasnya akan ditunjukkan pada Tabel 4.6.

Tabel 4.6 Pseudocode Fungsi Pemutaran Lagu

Pseudocode Fungsi Pemutaran Lagu

DEKLARASI AWAL

```
Int[] nadaLagu;  
Bool listen, play, boxPlayBenar;  
Int nadaAwal = 0;  
Int nadaAkhir = 1;
```

DESKRIPSI

1. MULAI
2. Jika, listen
3. Maka, jika, nadaAwal <= nadaAkhir
4. Maka, box yang diplay = nadaLagu[nadaAwal];
5. nadaAwal ++;
6. jika, nadaAwal == nadaAkhir
7. Maka, play = true;
8. listen = false;
9. jika, play
10. maka, jika, boxPlayBenar == true
11. maka, nadaAkhir++; listen = true;
12. jika boxPlayBenar == false
13. maka, listen = true;
14. SELESAI

Penjelasan pseudocode dari fungsi pemutaran lagu pada Tabel 4.6 adalah:

1. Dari deklarasi awal terdapat berbagai variabel mulai dari nadaLagu hingga nadaAkhir. Variabel nadaLagu merupakan array yang menyimpan nada dari

lagu dan variabel yang lain merupakan variabel yang akan digunakan pada *gameplay*.

2. Baris 2 menunjukkan sistem sedang memutar lagu. Pada baris 3-5 menunjukkan kondisi ketika nadaAwal masih kurang dari nadaAkhir maka nadaAwal pada nadaLagu akan diputar dan nadaAwal akan bertambah. Pada baris 6-8 menunjukkan kondisi ketika nadaAwal sama dengan nadaAkhir maka *play mode* akan dimlai.
3. Pada baris 9 menunjukkan sistem memulai proses *play mode*. pada baris 10-11 menunjukkan kondisi ketika pemain berhasil menirukan nada yang diputar maka nadaAkhir akan bertambah 1 dan kembali ke proses memutar lagu. Pada baris 12-13 terdapat kondisi ketika pemain tidak berhasil menirukan nada yang diputar maka akan langsung kembali ke proses memutar lagu tanpa menambahkan nadaAkhi.

4.3.3 Implementasi Fungsi *Chance*

Fungsi *chance* merupakan kesempatan berapa kali pemain dapat melakukan kesalahan. *Chance* ini akan berkurang setiap kali pemain melakukan kesalahan dan *game* akan berakhir jika *chance* tersebut habis atau sama dengan 0.

Tabel 4.7 Pseudocode Fungsi *Chance*

Pseudocode Fungsi <i>Chance</i>
DEKLARASI AWAL
Public int live;
Bool playerWrong, gameEnd;
DESKRIPSI
1. MULAI
2. Jika, live > 0
3. Maka, jika, playerWrong
4. Maka, live --;

```

5. Jika, live <= 0
6. Maka, gameEnd;
7. SELESAI

```

Penjelasan Pseudocode fungsi *chance* pada Tabel 4.7 adalah:

1. Pada deklarasi awal terdapat variabel *live* sebagai jumlah *chance*, *playerWrong* sebagai penunjuk kondisi jika pemain melakukan kesalahan, *gameEnd* sebagai penunjuk kondisi game akan berakhir.
2. Baris 2 menunjukkan ketika *chance* masih ada atau tidak sama dengan 0 maka akan melakukan baris 3-4. Pada baris 3-4 menunjukkan ketika pemain melakukan kesalahan maka *chance* akan berkurang 1.
3. Baris 5-6 menunjukkan kondisi ketika *chance* sudah habis maka *game* akan berakhir

4.3.4 Implementasi Fungsi *Score* dan *Combo*

Fungsi *score* dan *combo* digabung karena memiliki keterkaitan yang erat. *Score* didapat ketika pemain berhasil menirukan nada yang diputar sedangkan *combo* akan bertambah 1 ketika pemain berhasil menirukan nada 10 kali berturut-turut. *Combo* juga akan berkurang jika pemain melakukan kesalahan. *Score* yang didapat akan dikalikan dengan *combo*.

Tabel 4.8 Pseudocode Fungsi *Score* dan *Combo*

Pseudocode Fungsi <i>Score</i> dan <i>Combo</i>
DEKLARASI AWAL Public int score = 50; Public comboRange = 0; Public int combo = 1;

```
Bool playerRight, playerWrong;
DESKRIPSI
1. MULAI
2. Jika, comboRange <= 10
3. Maka, combo = 1;
4. Jika, comboRange > 10 && comboRang <= 20
5. Maka, combo = 2;
6. Jika, comboRange > 20
7. Maka, combo = 3;

8. Jika, playerRight
9. Maka, comboRange++;
10.     Score = score * combo
11. Jika, playerWrong
12. Maka, comboRange = comboRange - 5;
13. SELESAI
```

Penjelasan Pseudocode fungsi *score* dan *combo* pada Tabel 4.8 adalah:

1. Pada deklarasi awal terdapat variabel *score* sebagai *score* yang didapat pemain, *comboRange* sebagai jumlah berturut-turut pemain dapat mengikuti nada yang diputar, *combo* sebagai *combo game*, dan variabel kondisi *playerRight* dan *playerWrong*.
2. Pada baris 2-7 menunjukkan bagaimana mengeset *combo* melalui *comboRange*. Ketika *player* berhasil mengikuti nada berturut-turut 10 kali maka *combo* menjadi 2 dan jika berhasil sampai 20 kali lebih maka *combo* akan menjadi 3.

3. Baris 8-10 menunjukkan ketika pemain berhasil mengikuti nada maka *comboRange* akan bertambah 1 dan *score* akan bertambah sejumlah *score* dikalikan *combo*.
4. Pada baris 11-12 menunjukkan ketika pemain melakukan kesalahan maka *comboRange* akan berkurang sejumlah 5.

4.3.5 Implementasi *Gameplay* Utama *Art*

Art yang dibutuhkan dalam *gameplay* utama adalah *art* alat musik dari *game* yaitu berupa *box* (*box music*). Pada *box* tersebut diberikan tanda berupa tangga lagu yaitu do, re, mi, fa, so, la, ti, dan do kecil. Selain itu pada *box* juga diberikan tanda kunci dasar dan warna yang berbeda-beda. Untuk lebih jelasnya akan ditunjukkan pada Gambar 4.5.



Gambar 4.5 Art Alat Musik *Box*

4.3.6 Implementasi *Gameplay* Utama *Screen*

Pada desain *prototype* telah dijelaskan bahwa *gameplay* utama merupakan inti dari permainan ini. Sesuai dengan *digital prototype* maka dibuatlah berbagai macam menu dan HUD pada *gameplay* utama *screen* yang bertujuan mempermudah pemain dalam memainkan *game* ini. Untuk lebih jelasnya ditunjukkan pada Gambar 4.6.



Gambar 4.6 Gameplay Utama Screen

4.4 Implementasi *Gameplay* Latihan

Pada implementasi *gameplay* latihan ini akan dibagi kedalam beberapa bagian diantaranya adalah implementasi fungsi *free mode*, implementasi fungsi *lear song*, implementasi fungsi *play song*, implementasi *gameplay* latihan *art* dan implementasi *gameplay* latihan *screen*.

4.4.1 Implementasi Fungsi *Free Mode*

Pada implementasi *free mode* tidak jauh berbeda pada implementasi fungsi alat musik *box* yang terdapat pada implementasi *gameplay* utama. Pada fungsi alat musik *box* dijelaskan bahwa jika terdapat *input* maka *box* akan mengeluarkan animasi dan suara. Hal ini sama dengan implementasi *free mode*, dimana jika terdapat *input* dari pemain yang berupa *touch* maka akan mendeteksi *box* dan mengeluarkan animasi dan suara dari *box* tersebut.

4.4.2 Implementasi Fungsi *Learn Song*

Pada desain *digital prototype* telah dijelaskan bahwa fungsi *learn song* ini merupakan pilihan menu pada *gameplay* latihan dimana menu ini akan memberikan petunjuk kepada pemain tentang nada yang ada pada lagu anak yang telah dipilih oleh pemain. Nada yang ditunjukkan pada pemain akan dimulai dari nada awal dan pemain harus mengikuti nada tersebut agar dapat melihat nada selanjutnya.

Tabel 4.9 Pseudocode Fungsi *Learn Song*

Pseudocode Fungsi <i>Learn Song</i>
DEKLARASI AWAL Public int[] nadaLagu; Int nada = 0; Int boxNumber; Bool playerRight;
DESKRIPSI 1. MULAI 2. boxNumber = nadaLagu[nada] 3. Jika, ada input 4. Maka, jika, playerRight 5. Maka, nada++ 6. SELESAI

Penjelasan pseudocode fungsi *learn song* pada Tabel 4.9 adalah:

1. Pada deklarasi awal terdapat variabel *nadaLagu* sebagai penyimpan nada lagu, *boxNumber* sebagai identitas *box*, *nada* sebagai penanda dari mana nada dimulau, dan *playerRight* sebagai kondisi jika player dapat mengikuti nada lagu dengan benar.
2. Pada baris 2 menunjukkan bahwa *boxNumber* yang merupakan identitas *box* akan sama dengan *nadaLagu* ke *nada* atau *nadaLagu* awal karena *nada* sama dengan 0.
3. Pada baris 3-5 menunjukkan ketika ada *input* dari pemain maka akan mengecek apakah *box* yang dipilih oleh pemain sudah benar. Jika *box* sudah benar maka *nada* akan bertambah 1 atau menuju ke *nadaLagu* selanjutnya.

4.4.3 Implementasi Fungsi *Play Song*

Pada perancangan dijelaskan bahwa fungsi *play song* merupakan salah satu fungsi pada *gameplay* latihan yang akan memainkan lagu anak yang telah dipilih oleh pemain secara berulang-ulang. Hal ini bertujuan agar pemain terbiasa mendengarkan lagu tersebut dan dapat memainkannya dengan benar.

Tabel 4.10 Pseudocode Fungsi *Play Song*

Pseudocode Fungsi <i>Play Song</i>	
DEKLARASI AWAL	
Public Int[] nadaLagu;	
Int nadaAwal = 0;	
Int nadaAkhir = nadaLagu.length;	
Int boxNumber;	
Bool boxPlay, repeat;	
DESKRIPSI	
1.	MULAI
2.	jika, nadaAwal < nadaAkhir
3.	Maka, boxNumber = nadaLagu[nadaAwal];
4.	boxPlay = true;
5.	nadaAwal ++;
6.	jika, nadaAwal == nadaAkhir
7.	Maka, repeat == true;
8.	SELESAI

Penjelasan pseudocode fungsi *play song* pada Tabel 4.10 adalah:

1. Pada deklarasi awal terdapat variabel *nadaLagu* sebagai penyimpan nada dari lagu, *nadaAwal* sebagai nada awal lagu yang dimulai dari 0, *nadaAkhir*

sebagai nada akhir lagu yang memiliki nilai panjang dari nadaLagu, boxNumber sebagai identitas *box*, dan variabel kondisi boxPlay dan repeat.

2. Baris 2-5 menunjukkan ketika nada awal masih kurang dari nada akhir maka boxNumber memiliki nilai nadaLagu awal dan menjalankan boxPlay. boxPlay merupakan kondisi dimana *box* pada boxNumber akan dimainkan. Setelah boxPlay maka nadaAwal akan bertambah 1.
3. Baris 6-7 menunjukkan ketika nadaAwal telah sampai pada nadaAkhir maka repeat menjadi true atau akan mengulang dari awal.

4.4.4 Implementasi *Gameplay* Latihan *Art*

Pada *gameplay* latihan, *art* yang digunakan tidak jauh beda dengan *art* yang terdapat pada *gameplay* utama yaitu menggunakan *art* alat musik *box*. Selain *art* alat musik *box* tersebut, terdapat juga *art* menu latihan. *Art* menu latihan ditunjukkan pada Gambar 4.7.



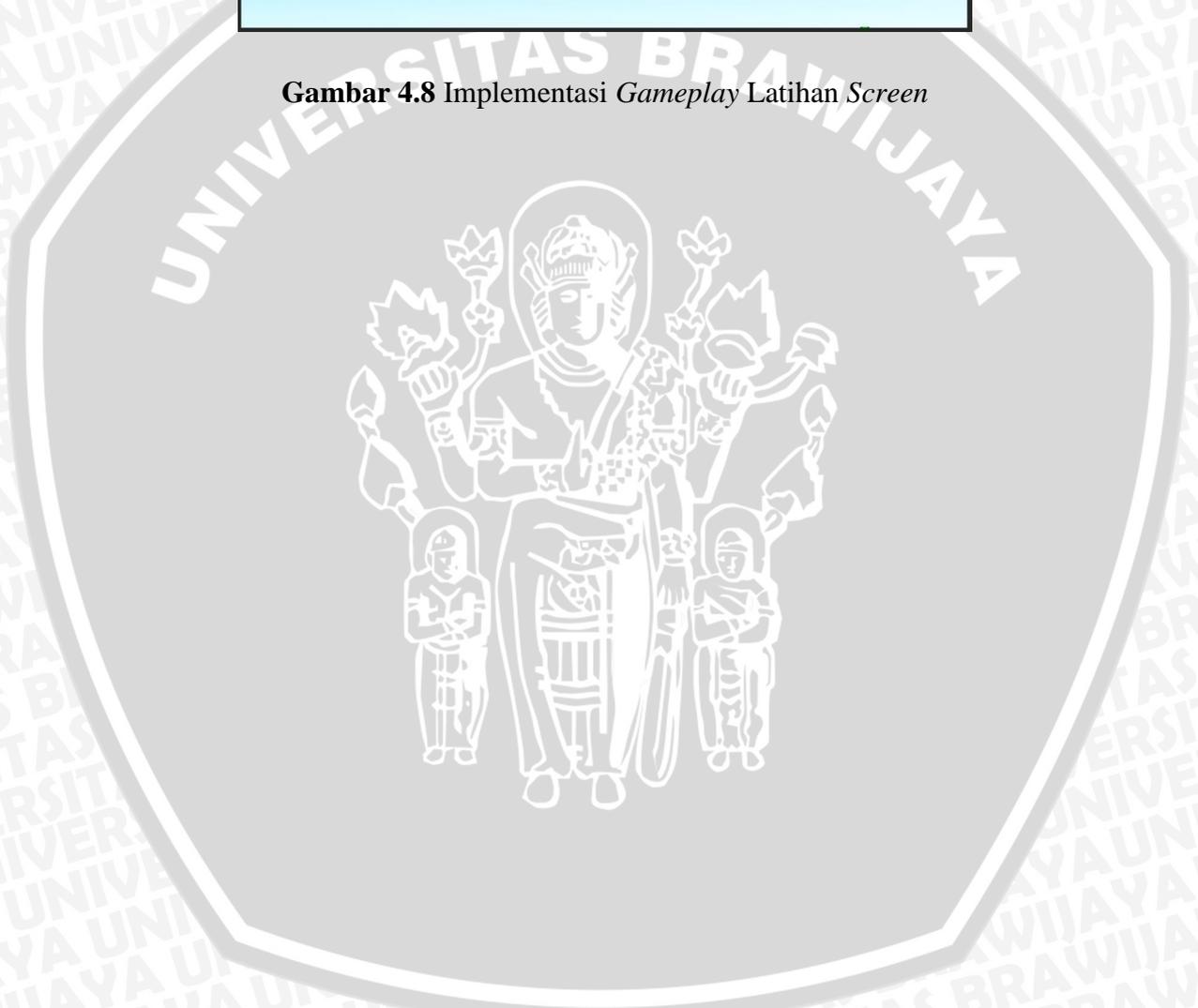
Gambar 4.7 *Art* Menu Latihan

4.4.5 Implementasi *Gameplay* Latihan *Screen*

Pada desain *prototype* telah dijelaskan bahwa *gameplay* latihan merupakan *gameplay* untuk yang dikhususkan untuk latihan pemain dalam menyelesaikan lagu-lagu anak yang ada pada *gameplay* utama. Dalam *gameplay* latihan *screen* terdapat berbagai macam menu untuk latihan diantaranya adalah *free mode*, *study song*, dan *play song*. Implementasi *gameplay* latihan *screen* akan ditunjukkan pada Gambar 4.8



Gambar 4.8 Implementasi *Gameplay* Latihan Screen



BAB V

EVALUASI

Bab ini akan membahas mengenai pengujian dan analisis permainan edukasi pengenalan lagu anak yang telah dibangun. Proses pengujian dilakukan melalui pengujian unit, pengujian integrasi, pengujian dengan *test suite*, dan *usability testing*. Pengujian unit dan pengujian integrasi menggunakan teknik pengujian *white box (white box testing)* sedangkan pengujian dengan *test suite* menggunakan teknik pengujian *black box (black box testing)*. Untuk *usability testing*, dilakukan kepada pengguna atau *user* yaitu anak-anak.

5.1 Pengujian Unit

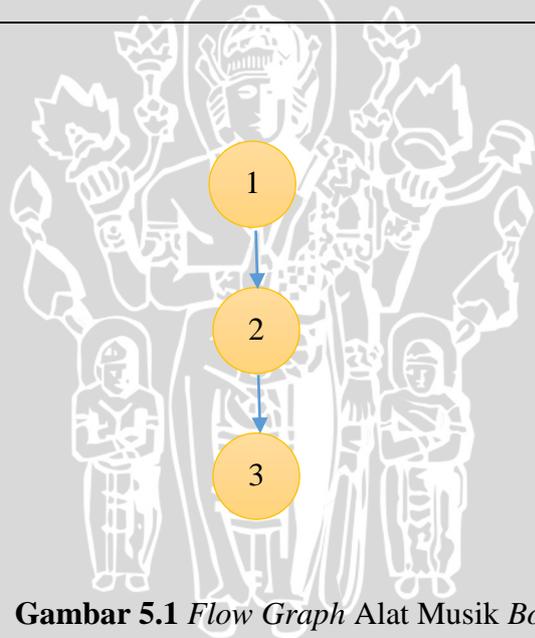
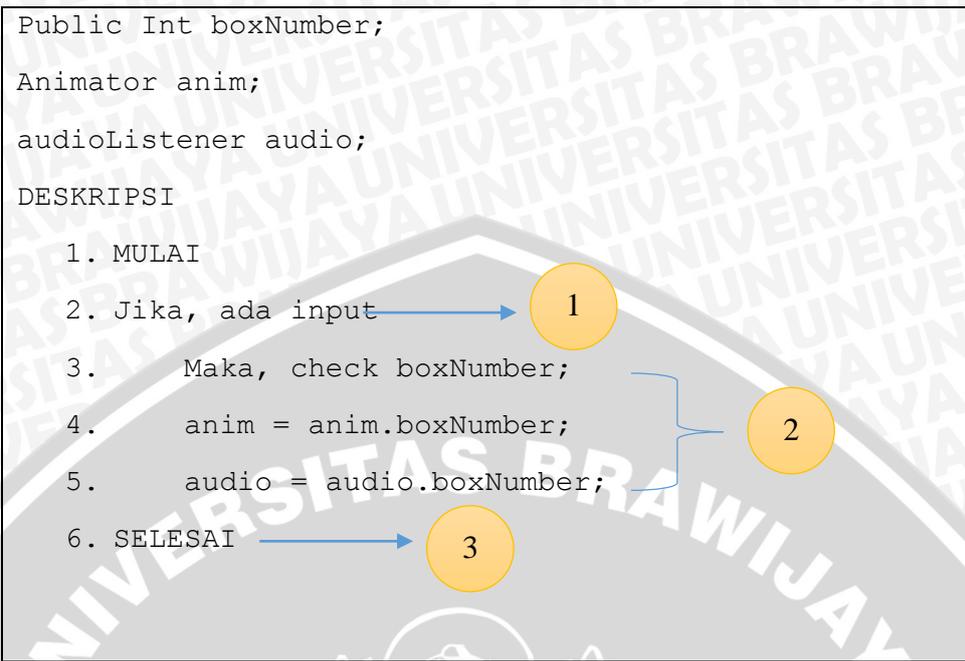
Pada pengujian unit digunakan metode *white box testing* dengan teknik *basis path testing*. Pada teknik *basis path testing* pengujian dilakukan dengan memodelkan algoritma pada sebuah *flow graph*, menentukan *cyclometric complexity*, dan melakukan uji kasus untuk setiap *path* yang ada.

5.1.1 Pengujian Unit Alat Musik Box

Alat musik *box* merupakan alat musik utama yang ada pada *game* ini. Alat musik ini memiliki bentuk kubus dan dapat mengeluarkan suara jika mendapat *input* berupa *touch* dari pemain. Selain suara, alat musik ini juga mengeluarkan animasi ketika mendapat *input*. Pemodelan alat musik *box* dalam bentuk *flow graph* ditunjukkan pada Table 5.1.

Tabel 5.1 Pemodelan *Flow Graph* Algoritma Alat Musik Box

Pemodelan <i>Flow Graph</i> Algoritma Alat Musik Box
DEKLARASI AWAL



Gambar 5.1 Flow Graph Alat Musik Box

Pemodelan ke dalam *flow graph* yang telah dilakukan pada alat musik *box* menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul atau (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 2 - 3 + 2
 \end{aligned}$$



= 1

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka ditentukan 1 buah basis set dari jalur *independent*, yaitu :

Jalur 1: 1-2-3

Tabel 5.2 Tabel Kasus Uji Untuk Pengujian Alat Musik *Box*

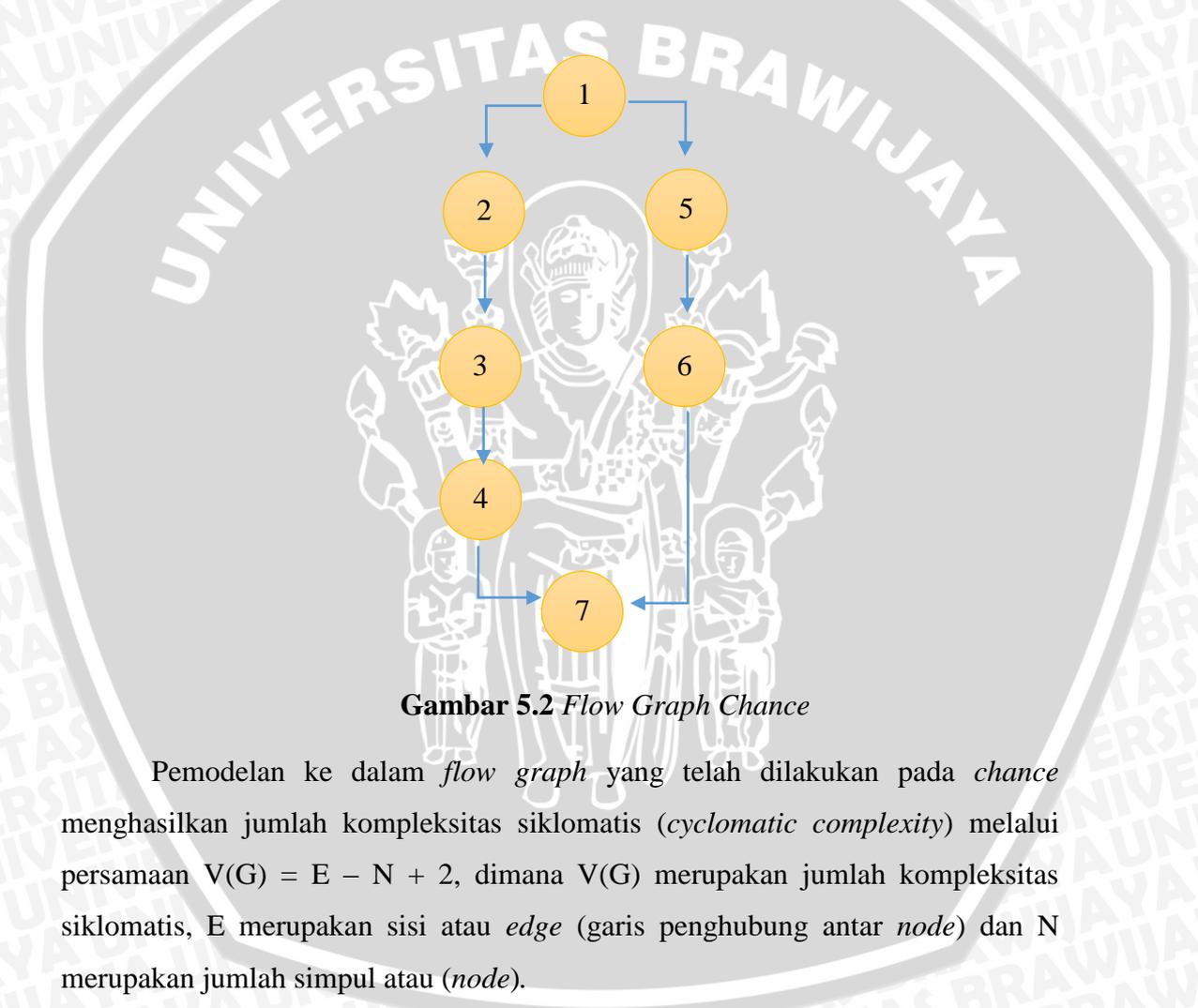
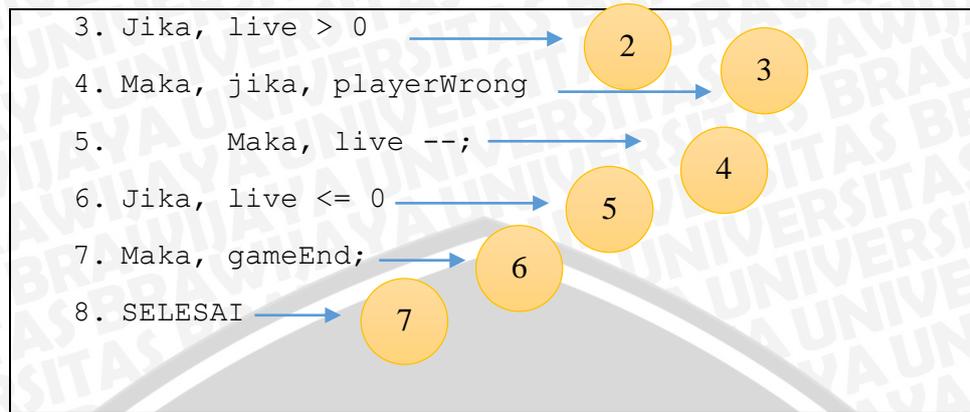
Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Ketika ada <i>input touch</i> dari pemain	Alat musik akan mengeluarkan nada dan menampilkan animasi sesuai <i>box</i> yang diberi <i>input</i>	Alat musik akan mengeluarkan nada dan menampilkan animasi sesuai <i>box</i> yang diberi <i>input</i>

5.1.2 Pengujian Unit *Chance*

Chance merupakan kesempatan yang dimiliki oleh pemain, apabila pemain melakukan kesalahan maka *chance* akan berkurang. Jika *chance* sudah 0 maka permainan akan berakhir atau *game over*. Pemodelan *chance* dalam bentuk *flow graph* ditunjukkan pada Table 5.3.

Tabel 5.3 Pemodelan *Flow Graph* Alogaritma *Chance*

Pemodelan <i>Flow Graph</i> Alogaritma <i>Chance</i>	
DEKLARASI AWAL	
Public int live;	
Bool playerWrong, gameEnd;	
DESKRIPSI	
1. MULAI	
2. Live = 5	→ 1



Gambar 5.2 Flow Graph Chance

Pemodelan ke dalam *flow graph* yang telah dilakukan pada *chance* menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul atau (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 7 - 7 + 2 \\
 &= 2
 \end{aligned}$$



Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka ditentukan 1 buah basis set dari jalur *independent*, yaitu :

Jalur 1 : 1-2-3-4-7

Jalur 2 : 1-5-6-7

Tabel 5.4 Tabel Kasus Uji Untuk Pengujian *Chance*

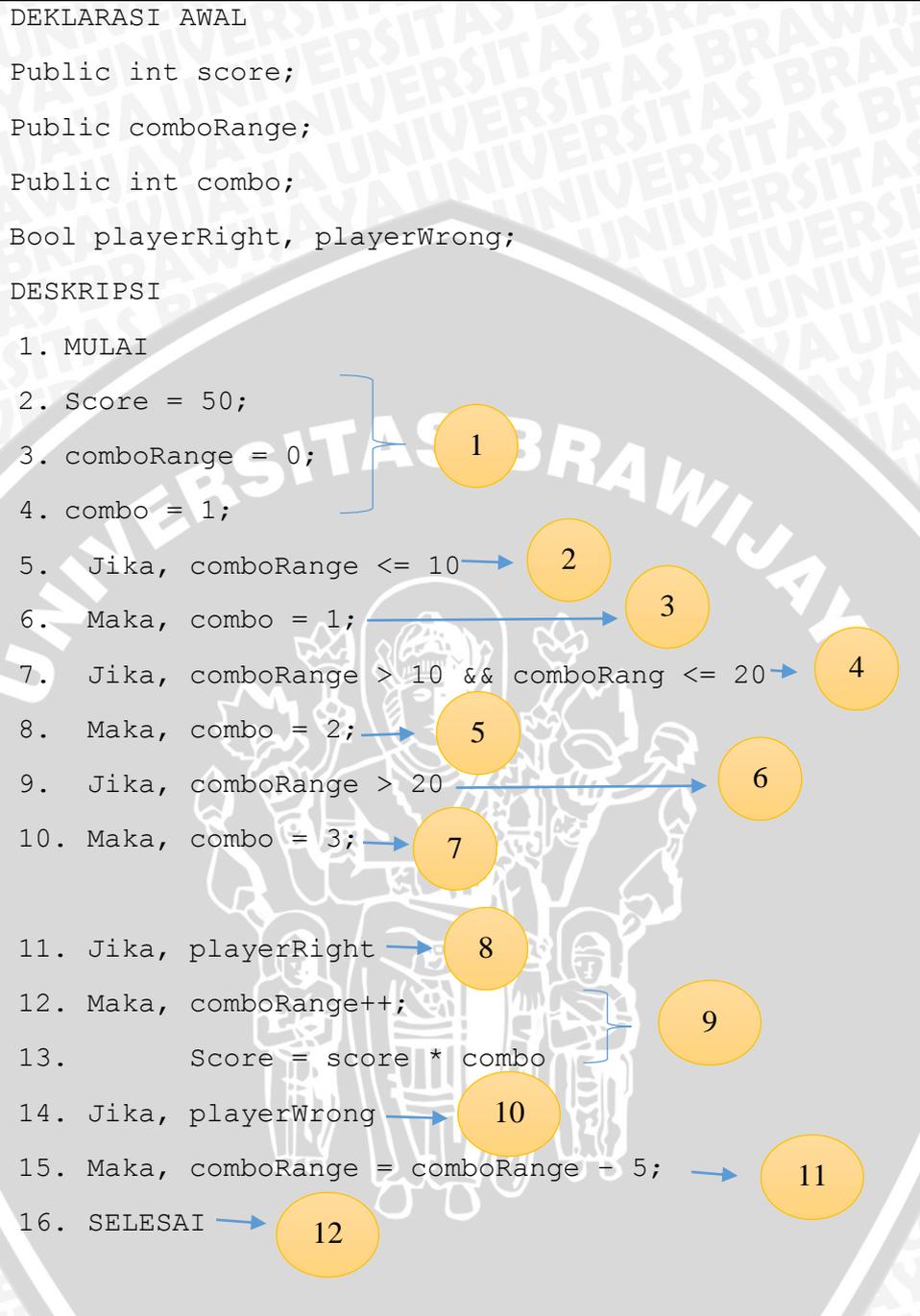
Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Ketika masih memiliki <i>chance</i> ($chance > 0$) dan player melakukan kesalahan	<i>Chance</i> akan berkurang 1 dan pemain dapat melanjutkan permainan	<i>Chance</i> akan berkurang 1 dan pemain dapat melanjutkan permainan
2	Ketika <i>chance</i> habis atau <i>chance</i> sama dengan 0	Permainan selesai (<i>game over</i>)	Permainan selesai (<i>game over</i>)

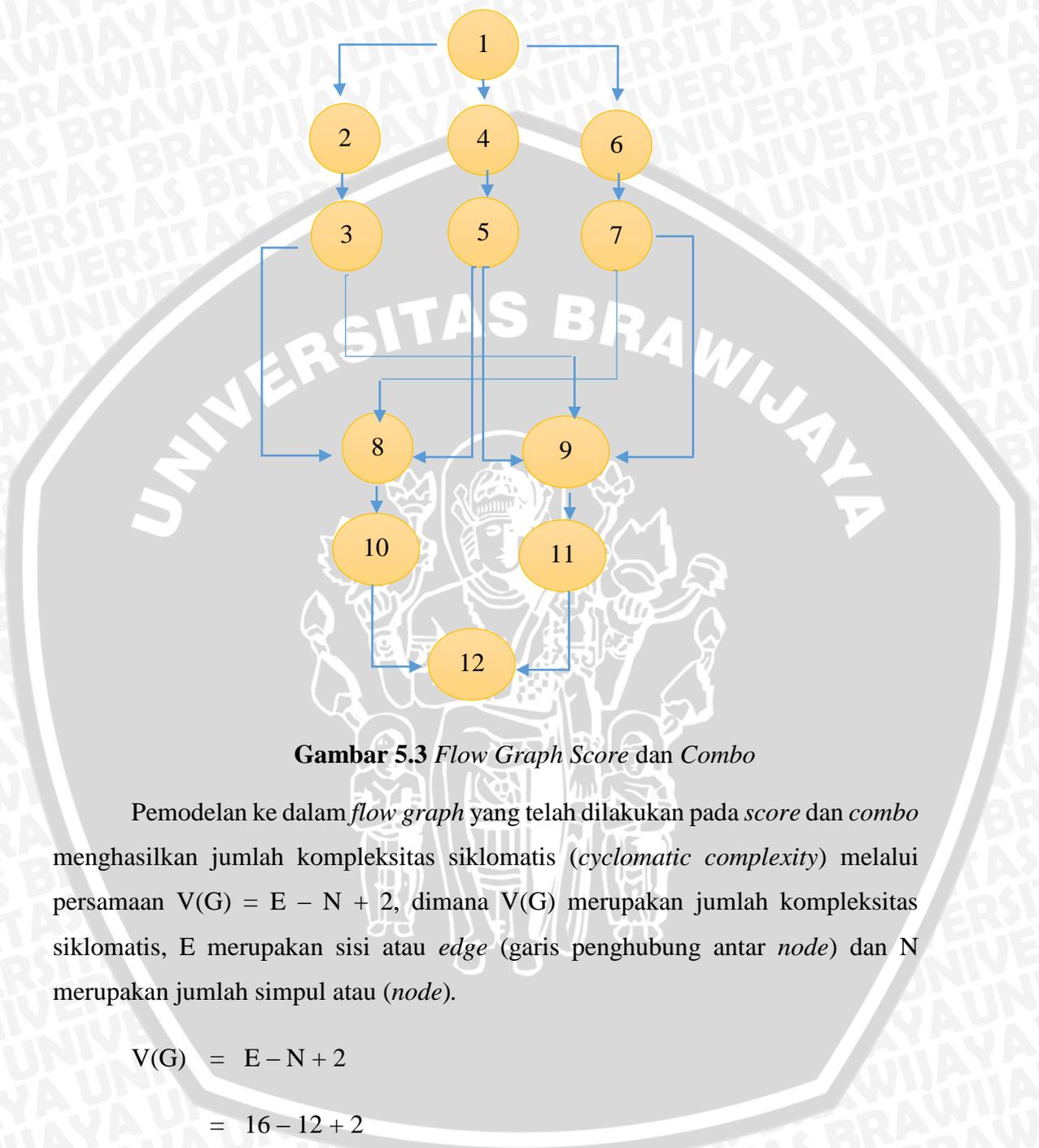
5.1.3 Pengujian Unit *Score* dan *Combo*

Pengujian unit *score* dan *combo* digabung karena memiliki keterkaitan yang erat. *Score* didapat ketika pemain berhasil menirukan nada yang diputar sedangkan *combo* akan bertambah 1 ketika pemain berhasil menirukan nada 10 kali berturut-turut. *Combo* juga akan berkurang jika pemain melakukan kesalahan. *Score* yang didapat akan dikalikan dengan *combo*. Pemodelan *score* dan *combo* dalam bentuk *flow graph* ditunjukkan pada Table 5.5.

Tabel 5.5 Pemodelan *Flow Graph* Alogaritma *Score* dan *Combo*

Pemodelan *Flow Graph* Alogaritma *Score* dan *Combo*





Gambar 5.3 *Flow Graph Score dan Combo*

Pemodelan ke dalam *flow graph* yang telah dilakukan pada *score* dan *combo* menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul atau (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 16 - 12 + 2 \\
 &= 6
 \end{aligned}$$

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka ditentukan 6 buah basis set dari jalur *independent*, yaitu :

Jalur 1 : 1-2-3-8-10-12

Jalur 2 : 1-4-5-8-10-12

Jalur 3 : 1-6-7-8-10-12

Jalur 4 : 1-2-3-9-11-12

Jalur 5 : 1-4-5-9-11-12

Jalur 6 : 1-6-7-9-11-12

Tabel 5.6 Tabel Kasus Uji Untuk Pengujian *Score* dan *Combo*

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	Pemain berhasil menirukan lagu dan <i>combo bar</i> berisi kurang dari 10	<i>Score</i> yang didapat dikalikan 1 dan <i>combo bar</i> bertambah 1	<i>Score</i> yang didapat dikalikan 1 dan <i>combo bar</i> bertambah 1
2	Pemain berhasil menirukan lagu dan <i>combo bar</i> berisi lebih dari 10 dan kurang dari 20	<i>Score</i> yang didapat dikalikan 2 dan <i>combo bar</i> bertambah 1	<i>Score</i> yang didapat dikalikan 2 dan <i>combo bar</i> bertambah 1
3	Pemain berhasil menirukan lagu dan <i>combo bar</i> berisi lebih dari 20	<i>Score</i> yang didapat dikalikan 3 dan <i>combo bar</i> bertambah 1	<i>Score</i> yang didapat dikalikan 3 dan <i>combo bar</i> bertambah 1
4	Pemain gagal menirukan lagu dan	Pemain tidak dapat <i>score</i> dan <i>combo bar</i> berkurang 5	Pemain tidak dapat <i>score</i> dan <i>combo bar</i> berkurang 5

	<i>combo bar</i> berisi kurang dari 10		
5	Pemain gagal menirukan lagu dan <i>combo bar</i> berisi lebih dari 10 dan kurang dari 20	Pemain tidak dapat <i>score</i> dan <i>combo bar</i> berkurang 5	Pemain tidak dapat <i>score</i> dan <i>combo bar</i> berkurang 5
6	Pemain gagal menirukan lagu dan <i>combo bar</i> berisi lebih dari 20	Pemain tidak dapat <i>score</i> dan <i>combo bar</i> berkurang 5	Pemain tidak dapat <i>score</i> dan <i>combo bar</i> berkurang 5

5.1.4 Analisis Pengujian Unit

Hasil yang didapatkan dari pengujian unit adalah :

1. Berdasarkan kesesuaian antara hasil pengujian tiap unit dan hasil yang diharapkan tiap unit pada *game* pengenalan lagu anak, maka dapat diambil kesimpulan bahwa unit modul dari program sudah sesuai dengan *output* yang diharapkan.
2. Berdasarkan hasil perhitungan *cyclomatic complexity* dari tiap *flow graph* kode unit, kode unit yang menghasilkan jalur uji terbanyak adalah kode untuk fungsi *score* dan *chance* sebanyak 6 kasus uji. Jumlah kasus uji kode ini sebagian besar dipengaruhi oleh seleksi kondisi.

5.2 Pengujian Integrasi

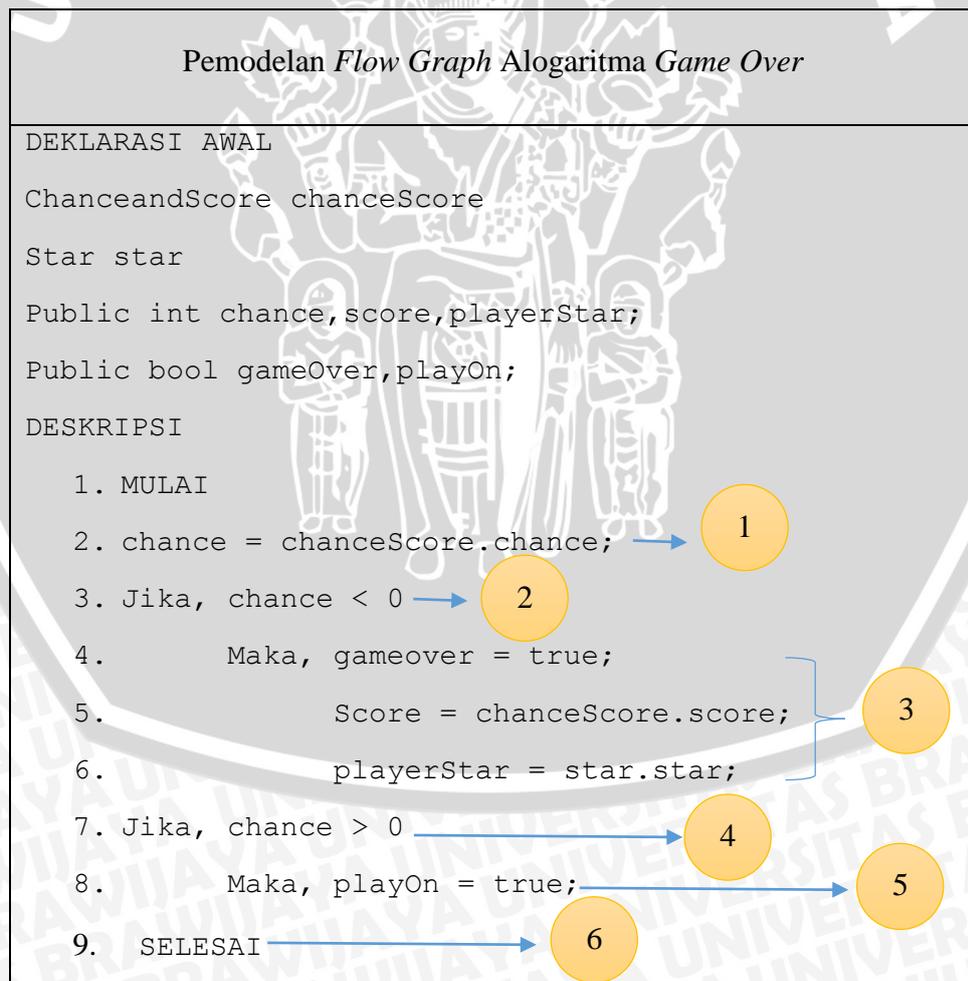
Pengujian integrasi diterapkan pada method yang mengintegrasikan fungsionalitas dari *class-class* lain untuk melakukan sebuah operasi tertentu. Pada pengujian integrasi *game* pengenalan lagu anak ini digunakan metode *white box testing* dengan teknik *basis path testing*. Pada teknik *basis path testing*, proses pengujian dilakukan dengan memodelkan algoritma pada suatu *flow graph*,

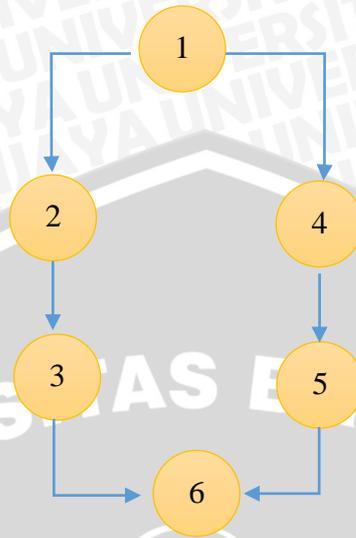
menentukan kompleksitas siklomatis (*cyclomatic complexity*), menentukan sebuah basis set dari jalur independen dan memberikan kasus uji (*test case*) pada setiap basis set yang telah ditentukan.

5.2.1 Pengujian Integrasi *Game Over*

Untuk menjalankan fungsi *game over*, dibutuhkan fungsi yang ada pada *class* lain diantaranya adalah *class chance and score* dan *class star*. *Class chance and score* ini dibutuhkan fungsi *game over* untuk mengetahui kondisi menang dan kalah, sedangkan *class star* ini digunakan untuk menunjukkan berapa bintang yang didapat pemain setelah permainan berakhir. Pemodelan *game over* ke dalam bentuk *flow graph* akan ditunjukkan pada Tabel 5.7.

Tabel 5.7 Pemodelan *Flow Graph* Alogaritma *Game Over*





Gambar 5.4 Flow Graph Game Over

Pemodelan ke dalam *flow graph* yang telah dilakukan pada fungsi *game over* menghasilkan jumlah kompleksitas siklomatis (*cyclomatic complexity*) melalui persamaan $V(G) = E - N + 2$, dimana $V(G)$ merupakan jumlah kompleksitas siklomatis, E merupakan sisi atau *edge* (garis penghubung antar *node*) dan N merupakan jumlah simpul atau (*node*).

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 6 - 6 + 2 \\
 &= 2
 \end{aligned}$$

Berdasarkan dari nilai *cyclomatic complexity* yang telah didapatkan dari perhitungan maka ditentukan 2 buah basis set dari jalur *independent*, yaitu :

Jalur 1 : 1-2-3-6

Jalur 2 : 1-4-5-6

Tabel 5.8 Kasus Uji Untuk Pengujian *Game Over*



Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang didapatkan
1	<i>Chance</i> kurang dari sama dengan 0	<i>Window game over</i> muncul dan menampilkan <i>score</i> akhir serta bintang yang didapat pemain	<i>Window game over</i> muncul dan menampilkan <i>score</i> akhir serta bintang yang didapat pemain
2	<i>Chance</i> lebih besar dari 0 (pemain masih memiliki <i>chance</i>)	Permainan berlanjut	Permainan terus berlanjut

5.2.2 Analisis Pengujian Integrasi

Hasil yang didapatkan dari pengujian integrasi adalah :

1. Berdasarkan kesesuaian antara hasil pengujian integrasi dan hasil yang diharapkan pada *game* pengenalan lagu anak, maka dapat diambil kesimpulan bahwa integrasi modul dari program sudah sesuai dengan *output* yang diharapkan.
2. Berdasarkan hasil perhitungan *cyclomatic complexity* dari *flow graph* kode integrasi fungsi *game over* didapatkan jumlah jalur uji sebanyak 2 kasus uji. Jumlah kasus uji kode ini sebagian besar dipengaruhi oleh seleksi kondisi.

5.3 Pengujian Dengan Test Suite

Pada *test suite* ini menggunakan teknik pengujian *black box*, *test suite* sendiri merupakan sekumpulan dari *test case* sebuah *game* yang saling berhubungan. Pada *test suite* ini akan diuji proses-proses yang ada pada *game* apakah telah sesuai dengan yang diharapkan atau masih terdapat *bug* yang perlu diperbaiki. Pada *game* pengenalan lagu anak ini sendiri *test suite* dibagi kedalam 3 *region* atau wilayah diantaranya adalah pada *main menu scene*, *level scene*, dan

gameplay scene. Pada tiap *region* tersebut terdapat berbagai macam proses yang akan diuji diantaranya:

1. *Main menu scene*

Pada *main menu scene* pengujian dilakukan mulai dari proses game masuk pada *main menu* dan dilanjutkan dengan proses-proses yang terdapat pada *main menu* seperti proses pengaturan *background music*, proses *menu tutorial*, dan proses *exit game*. Pengujian pada *main menu* ini diakhiri dengan proses masuknya *game* pada *level scene*.

2. *Level scene*

Pada *level scene* pengujian dilakukan mulai dari proses masuknya *game* pada *level scene* dilanjutkan dengan proses yang ada di dalamnya seperti proses kembali ke *main menu*, proses terbuka, dan terkuncinya level. Pengujian pada *level scene* ini diakhiri dengan masuknya *game* pada *gameplay scene*.

3. *Gameplay scene*

Pada *gameplay scene* pengujian dilakukan mulai dari *game* masuk ke *gameplay scene* dan dilanjutkan dengan proses yang ada pada *gameplay scene*. Proses-proses yang ada pada *gameplay scene* diantaranya adalah proses memainkan musik pada *box*, proses *scoring*, proses menang, dan kalah dan lain-lain. Pengujian pada *gameplay scene* diakhiri dengan keluarnya *game* dari *gameplay scene*.

Untuk mengetahui tingkat kebenaran dari pengujian *game* pengenalan lagu anak ini, *test suite* yang telah dirancang ditunjukkan pada Lampiran L-1. *Test suite* tersebut akan dibagikan ke penguji yang terdiri dari responden. Responden akan diminta memainkan *game* pengenalan lagu anak ini dan mengisi *test suite* tersebut.

5.3.1 Hasil Test Suite

Pada Tabel 5.9 akan ditunjukkan hasil dari pengujian menggunakan *test suite*.

Tabel 5.9 Hasil Test Suite

Main Menu Scene			
No.	Test case	Prosedur dan Input	Hasil
1.	Animasi background dapat bergerak dan animasi box pada background dapat berputar?	-	Valid
2.	Background Music (BGM) dapat dihidupkan dan dimatikan melalui tombol bergambar nada lagu?	Pemain menekan tombol BGM	Valid
3.	Tombol bergambar buku akan membuka menu tutorial?	Pemain menekan tombol tutorial	Valid
4.	Tombol next pada menu tutorial akan membuka tutorial selanjutnya?	Pemain menekan tombol <i>next</i> pada tutorial window	Valid
5.	Tombol back pada menu tutorial akan membuka tutorial sebelumnya?	Pemain menekan tombol <i>back</i> pada tutorial <i>window</i>	Valid
6.	Tombol play akan membuka play level scene?	Pemain menekan tombol <i>play</i>	Valid
7.	Tombol train akan membuka train level scene?	Pemain menekan tombol <i>train</i>	Valid

8.	Tombol berbentuk silang akan membuka menu exit?	Pemain menekan tombol <i>exit</i>	Valid
9.	Tombol yes pada menu exit berarti keluar dari game?	Pemain menekan tombol <i>yes</i> pada <i>exit window</i>	Valid
10.	Tombol no pada menu exit berarti kembali ke menu utama?	Pemain menekan tombol <i>no</i> pada <i>exit window</i>	Valid
11.	Tombol back pada mobile device akan membuka menu exit?	Pemain menekan tombol <i>back</i> pada android <i>device</i>	Valid
Level Scene			
Play Level Scene			
1.	Level yang terbuka akan menuju level <i>gameplay</i> ketika dipilih?	Pemain menekan gambar level yang terbuka	Valid
2.	Level yang terkunci tidak dapat dipilih?	Pemain menekan gambar level yang tertutup	Valid
3.	Bintang pada tiap level didapat setelah bermain dan mendapatkan bintang?	Pemain memainkan <i>game</i> dan setidaknya menyelesaikan setengah dari lagu	Valid
4.	Tombol back berarti akan menuju ke main menu scene?	Pemain menekan tombol back	Valid

5.	Tombol back pada mobile device berarti akan menuju ke main menu scene?	Pemain menekan tombol <i>back</i> pada android <i>device</i>	Valid
<i>Train Level Scene</i>			
1.	Level yang terbuka akan menuju train <i>gameplay</i> ketika dipilih?	Pemain menekan gambar level yang terbuka	Valid
2.	Level yang terkunci tidak dapat dipilih?	Pemain menekan gambar level yang tertutup	Valid
3.	Tombol back berarti akan menuju ke main menu scene?	Pemain menekan tombol back	Valid
4.	Tombol back pada mobile device berarti akan menuju ke main menu scene?	Pemain menekan tombol <i>back</i> pada android <i>device</i>	Valid
<i>Gameplay Scene</i>			
<i>Gameplay utama Scene</i>			
1.	Awal permainan diawali dengan hitung mundur?	-	Valid
2.	Background awan dan balon dapat bergerak?	-	Valid
3.	Tanda “listen” berarti nada dari lagu dimainkan?	-	Valid

4.	Tanda “ <i>listen</i> ” berarti pemain tidak dapat memainkan alat musik <i>box</i> ?	Pemain mencoba menekan alat musik <i>box</i>	Valid
5.	Tanda “ <i>your turn</i> ” berarti pemain dapat memainkan alat musik <i>box</i> ?	Pemain mencoba menekan alat musik <i>box</i>	Valid
6.	<i>Chance</i> berkurang 1 ketika pemain salah dalam mengikuti nada yang diputar?	Pemain menekan alat musik <i>box</i> secara random tanpa mengikuti <i>game system</i>	Valid
7.	<i>Score</i> bertambah 50 dikalikan <i>combo</i> ketika pemain benar dalam mengikuti nada yang diputar?	Pemain menekan alat musik <i>box</i> mengikuti <i>game system</i>	Valid
8.	<i>Combo bar</i> akan bertambah ketika pemain benar dan berkurang ketika pemain salah dalam mengikuti nada yang diputar?	Pemain menekan alat musik <i>box</i> mengikuti <i>game system</i> dan tidak mengikuti <i>game system</i>	Valid
9.	<i>Combo</i> bertambah ketika <i>combo bar</i> full?	Pemain menekan alat musik <i>box</i> mengikuti <i>game system</i>	Valid
10.	Background Music (BGM) dapat dihidupkan dan dimatikan melalui tombol bergambar nada?	Pemain menekan tombol BGM	Valid
11.	Tombol yang bergambar 2 garis vertical berarti membuka menu <i>pause</i> ?	Pemain menekan tombol <i>pause</i>	Valid
12.	Permainan berhenti ketika menu <i>pause</i> terbuka?	-	Valid

13.	Tombol <i>resume</i> berarti permainan dilanjutkan?	Pemain menekan tombol <i>resume</i>	Valid
14.	Tombol <i>try again</i> berarti permainan diulang dari awal?	Pemain menekan tombol <i>try again</i>	Valid
15.	Tombol menu berarti kembali ke menu utama?	Pemain menekan tombol menu	Valid
16.	Membuka menu <i>game over</i> ketika <i>chance</i> dari pemain 0 atau semua nada lagu dapat di mainkan dengan benar?	-	Valid
17.	Menu <i>game over</i> akan memproses <i>score</i> yang didapat, <i>best score</i> , dan jumlah bintang yang didapat?	-	Valid
18.	Tombol <i>try again</i> pada menu <i>game over</i> berarti permainan diulang?	Pemain menekan tombol <i>try again</i>	Valid
19.	Tombol menu pada menu <i>game over</i> berarti kembali ke <i>main menu scene</i> ?	Pemain menekan tombol menu	Valid
20.	Tombol <i>back</i> pada android device akan membuka menu <i>pause</i> ?	Pemain menekan tombol <i>back</i> pada android <i>device</i>	Valid
<i>Gameplay latihan Scene</i>			
1.	Background awan dan balon dapat bergerak?	-	Valid

2.	Background Music (BGM) dapat dihidupkan dan dimatikan melalui tombol bergambar nada?	Pemain menekan tombol BGM	Valid
3.	Tombol <i>back</i> berarti kembali ke main menu scene?	Pemain menekan tombol <i>back</i>	Valid
4.	Semua box dapat dimainkan pada menu “free mode”?	Pemain menekan semua <i>box</i> pada alat musik <i>box</i>	Valid
5.	Menu “ <i>learn the song</i> ” memberikan tanda nada mana yang harus diikuti?	Pemain menekan gambar <i>learn the song</i>	Valid
6.	Menu “ <i>play the song</i> ” memainkan lagu secara otomatis?	Pemain menekan gambar <i>play the song</i>	Valid
7.	Tombol <i>back</i> pada android device akan kembali ke <i>main menu scene</i> ?	Pemain menekan tombol <i>back</i> pada android <i>device</i>	Valid

5.3.2 Analisis Test suite

Berdasarkan hasil yang ditunjukkan oleh pengujian yang dilakukan dengan *test suite*, didapatkan hasil semua proses yang ada pada *game* telah valid. Maka dapat disimpulkan bahwa implementasi dan fungsionalitas dari *game* pengenalan lagu anak telah memenuhi kebutuhan yang telah dijabarkan pada daftar kebutuhan.

5.4 Usability Testing

Usability testing merupakan pengujian yang dilakukan pada pengguna yang sebenarnya. Pada *game* pengenalan lagu anak ini, *usability testing* dilakukan pada anak usia 6-7 tahun. Tujuan dilakukannya *usability testing* ini adalah untuk mengetahui efektifitas *game* dalam mengenalkan dan mengajari anak tentang lagu anak. Pada *game* pengenalan lagu anak ini *usability testing* dilakukan dengan cara

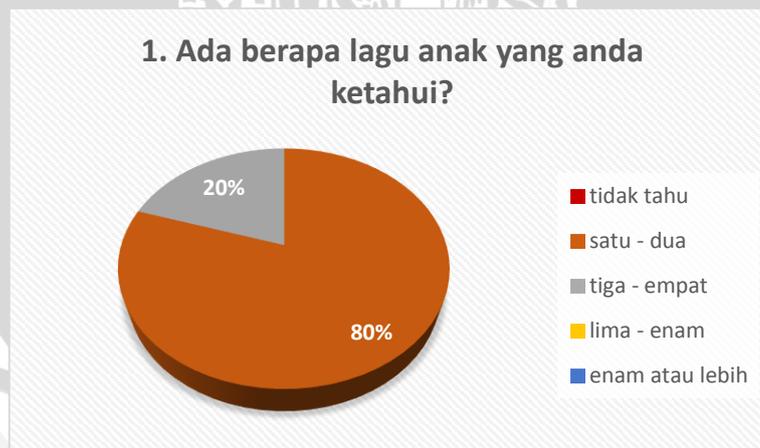
memberikan beberapa pertanyaan kepada pengguna. Berikut merupakan pertanyaan yang diberikan kepada pengguna :

1. Ada berapa lagu anak yang anda ketahui?
2. Apakah anda bisa memainkan lagu anak tersebut?
3. Ada berapa lagu anak yang anda ketahui setelah memainkan *game* ini?
4. Apakah anda bisa memainkan lagu anak tersebut setelah memainkan *game* ini?
5. Apakah anda suka konsep belajar sambil bermain seperti pada *game* ini?

5.4.1 Hasil dan Analisis *Usability Testing*

Hasil *usability testing* didapat pada 5 responden anak-anak. Responden tersebut diberikan pertanyaan sesuai dengan yang ada pada *usability testing*. Pertanyaan pertama dan kedua dilakukan sebelum responden memainkan *game* dan pertanyaan ketiga sampai kelima dilakukan setelah responden memainkan *game*. Berikut merupakan hasil dan analisis tiap pertanyaan:

1. Pertanyaan pertama



Gambar 5.5 Hasil *usabiliy testing* pertanyaan pertama

Pertanyaan pertama ini diberikan kepada responden sebelum memainkan *game*. Hal ini bertujuan untuk mengetahui pengetahuan responden tentang

lagu anak. Dari kelima responden, didapat jawaban 20% mengetahui 3-4 lagu dan sisanya hanya mengetahui 1-2 lagu saja. Dari jawaban tersebut menunjukkan bahwa, tingkat pengetahuan mereka tentang lagu anak masih sangat minim.

2. Pertanyaan kedua

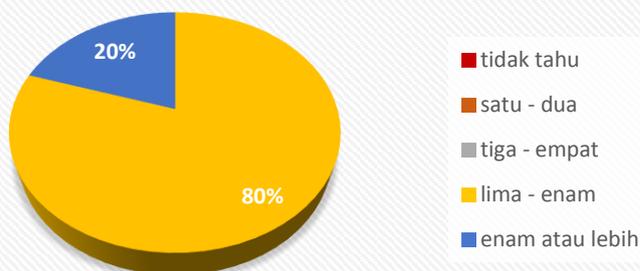


Gambar 5.6 Hasil *usability testing* pertanyaan kedua

Pertanyaan kedua ini juga diberikan sebelum responden memainkan *game*. Hal ini bertujuan untuk mengetahui apakah mereka bisa memainkan lagu anak tersebut. Dari kelima responden, didapat jawaban 20% dari mereka dapat memainkan sebagian lagu tersebut dan sisanya yaitu 80% tidak dapat memainkan lagu tersebut. Dari jawaban tersebut menunjukkan bahwa, walaupun responden mengetahui lagu anak belum tentu mereka dapat memainkan lagu anak tersebut dengan baik.

3. Pertanyaan ketiga

3. Ada berapa lagu anak yang anda ketahui setelah memainkan *game* ini?

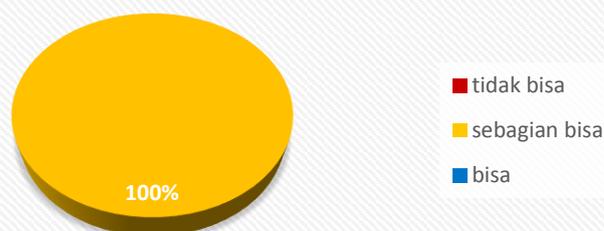


Gambar 5.7 Hasil *usability testing* pertanyaan ketiga

Pertanyaan ketiga ini diberikan pada responden setelah mereka memainkan *game*. Hal ini bertujuan untuk mengetahui seberapa jauh pengetahuan mereka tentang lagu anak setelah memainkan *game* pengenalan lagu anak ini. Dari kelima responden, didapat jawaban 20% dari mereka mengetahui enam atau lebih lagu dan sisanya yaitu 80% mengetahui lima sampai enam lagu. Dari jawaban tersebut menunjukkan bahwa, setelah memainkan *game* lagu anak ini, pengetahuan responden tentang lagu anak meningkat. Dimana sebelum memainkan *game* ini, sebagian besar dari mereka hanya mengetahui satu sampai dua lagu dan setelah memainkan *game* ini, sebagian besar dari mereka mengetahui lima sampai enam lagu.

4. Pertanyaan keempat

4. Apakah anda bisa memainkan lagu anak tersebut setelah memainkan *game* ini?



Gambar 5.8 Hasil *usability testing* pertanyaan keempat

Pertanyaan keempat ini diberikan pada responden setelah mereka memainkan *game*. Hal ini bertujuan untuk mengetahui apakah mereka dapat memainkan lagu anak tersebut setelah mereka memainkan *game* pengenalan lagu anak ini. Dari kelima responden, didapat jawaban, keseluruhan atau 100% dari mereka menjawab sebagian lagu bisa dimainkan. Dari jawaban tersebut menunjukkan bahwa terjadi peningkatan pengetahuan tentang memainkan lagu anak dari responden. Dimana sebelum memainkan *game*, sebagian besar dari responden tidak dapat memainkan lagu anak tersebut dan setelah memainkan *game*, keseluruhan dari mereka dapat memainkan sebagian dari lagu anak tersebut.

5. Pertanyaan kelima

**Gambar 5.9** Hasil *usability testing* pertanyaan kelima

Pertanyaan kelima ini diberikan kepada responden setelah mereka memainkan *game*. Hal ini bertujuan untuk mengetahui apakah mereka menyukai konsep belajar sambil bermain seperti yang ada pada *game* pengenalan lagu anak ini. Dari kelima responden, didapat jawaban keseluruhan atau 100% dari mereka menjawab suka. Hal ini menunjukkan bahwa konsep belajar sambil bermain ini lebih disukai karena dengan bermain tidak menimbulkan suatu tekanan mental terhadap anak.

Dari keseluruhan hasil dan analisa pertanyaan pada *usability testing* ini didapat kesimpulan bahwa aplikasi ini telah memenuhi kebutuhan dan sesuai dengan tujuan yaitu membangun sebuah permainan edukasi yang dapat mengenalkan dan menunjukkan cara bermain lagu-lagu anak.



BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

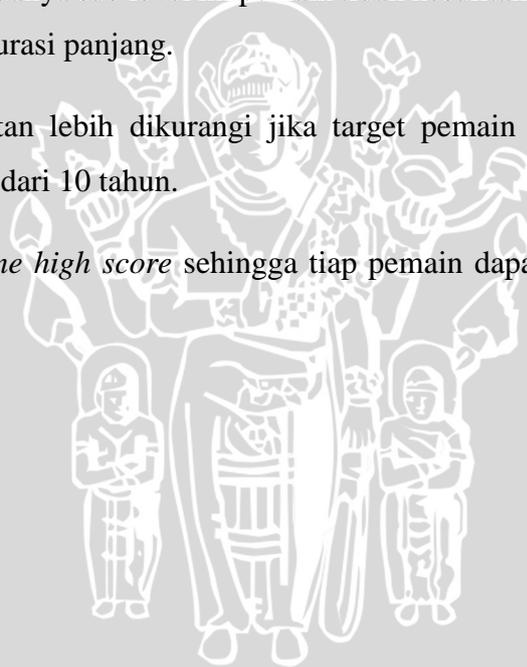
Game edukasi pengenalan lagu anak adalah *game* edukasi yang diharapkan bermanfaat untuk mengenalkan lagu-lagu anak kepada anak serta sebagai sarana alternatif belajar musik yang menyenangkan untuk anak. Dari hasil pengamatan selama desain, implementasi dan proses evaluasi *game* yang dilakukan, diambil kesimpulan sebagai berikut:

1. Berdasarkan hasil dari pengujian *play testing* yang dilakukan dengan cara memberikan kuisioner kepada responden, didapatkan kesimpulan bahwa desain *game* edukasi pengenalan lagu anak ini dapat diterima dengan baik oleh responden dan siap untuk diimplementasikan. Hal tersebut dibuktikan dengan jumlah persentase jawaban cukup, baik, dan sangat baik dari responden lebih dari 90%.
2. Berdasarkan hasil dari pengujian yang dilakukan dengan pengujian unit dan pengujian integrasi yang menggunakan teknik pengujian *white box* dengan metode *basis path*, didapatkan kesimpulan bahwa dari semua kasus uji yang dilakukan didapatkan hasil *output* sesuai dengan yang diharapkan.
3. Berdasarkan hasil dari pengujian yang dilakukan dengan *test suite* yang menggunakan teknik pengujian *black box*, didapatkan kesimpulan bahwa semua proses yang ada pada *game* telah valid.
4. Berdasarkan hasil dari *usability testing* yang dilakukan kepada pengguna didapat kesimpulan bahwa aplikasi ini telah memenuhi kebutuhan dan sesuai dengan tujuan yaitu membangun sebuah permainan edukasi yang dapat mengenalkan dan menunjukkan cara bermain lagu-lagu anak.

6.2 Saran

Saran untuk pengembangan *game* edukasi pengenalan lagu anak lebih lanjut adalah:

1. Penambahan lagu-lagu anak lain yang lebih global karena pada *game* ini lagu anak yang global hanya ada 1 yaitu “Twinkle Twinkle Little Star”.
2. Penambahan fitur *skip* nada pada setiap level. Dengan adanya fitur *skip* nada ini pemain tidak perlu menunggu pemutaran nada sampai selesai saat memainkan lagu yang dipilih.
3. Level diperluas dengan sub level yang didalamnya terdapat bagian-bagian lagu. Dengan adanya sub level ini pemain tidak kesulitan ketika memainkan lagu yang berdurasi panjang.
4. Tingkat kesulitan lebih dikurangi jika target pemain adalah anak yang berusia kurang dari 10 tahun.
5. Diberikan *online high score* sehingga tiap pemain dapat membandingkan *score* mereka.



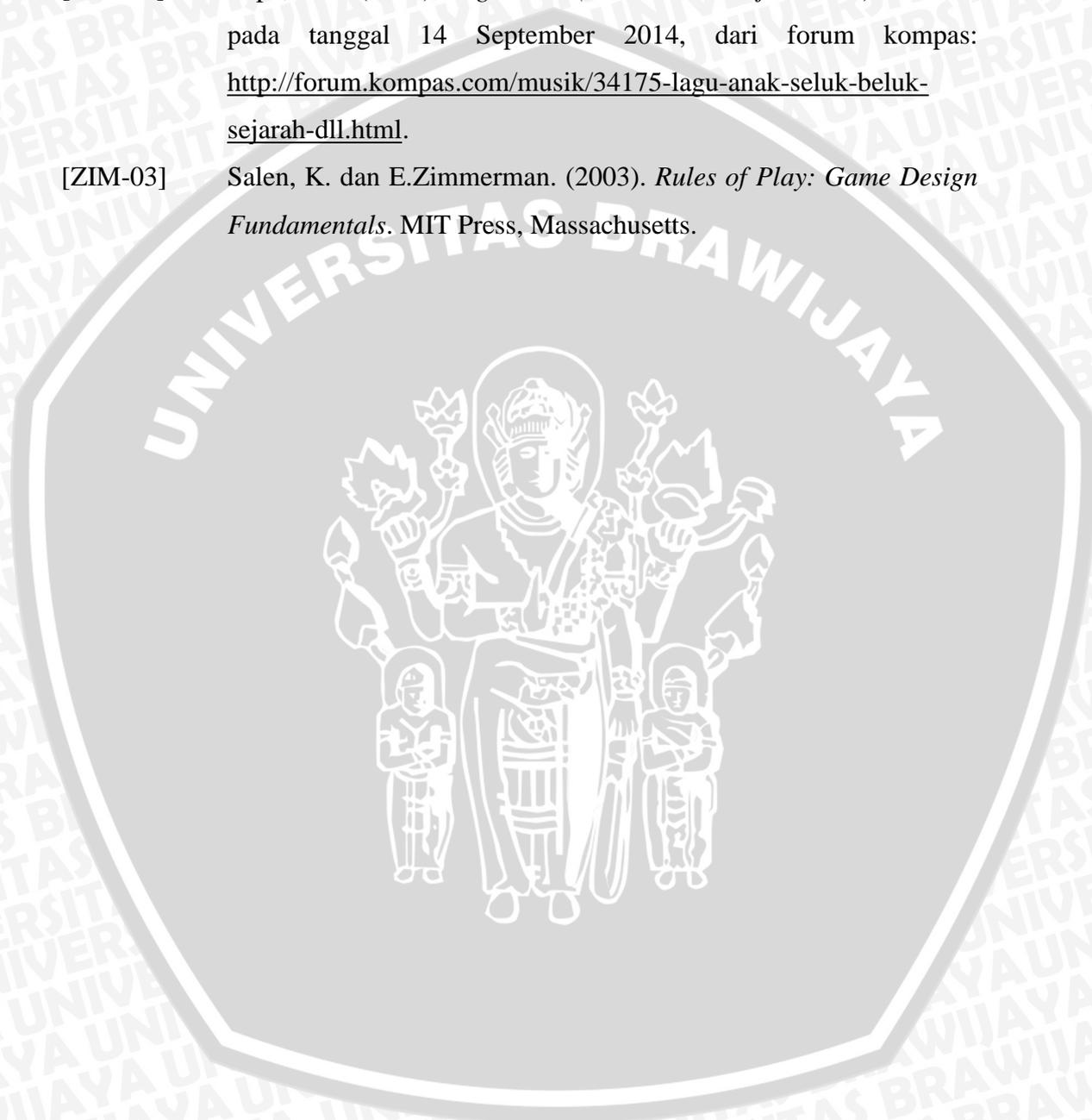
DAFTAR PUSTAKA

- [AUD-14] Audacity. (2014). *About Audacity*. Diambil pada tanggal 27 agustus 2014, dari web resmi Audacity:
<http://audacity.sourceforge.net/about/>.
- [DON-02] Campbell, Don. (2002). *Efek Mozart*. Terjemahan dari Bahasa Inggris oleh T. Hermaya. Cetakan kedua. Jakarta. Gramedia Pustaka Utama.
- [FIN-04] Finney, Kenneth C. (2004). *3D Game Programming All In One*. Thomson Course Technology PTR 2004.
- [HUR-09] Hurd, Daniel dan Erin Jennings. (2009). *Standardized Educational Games Ratings: Suggested Criteria*. Karya Tulis Ilmiah.
- [NID-12] Nidhra, Srinivas. (2012). *BLACK BOX AND WHITE BOX TESTING TECHNIQUES –A LITERATURE REVIEW*. International Journal 2 (2).
- [PER-04] Permana, Budi. (2004). *Seri Penuntun Praktis Adobe Photoshop 7.0*, cetakan ketiga, PT. ElexMedia Komputindo, Jakarta.
- [PRE-10] Pressman, Roger S. (2010). *Software Engineering: A Practiyioner's Approach, Seventh Edition*, McGraw Hill.
- [SEC-09] Schreiber, Ian dan Sebastian Sohn. (2009). *Game Design Concept*. Creative Commons Attribution, United States.
- [VIR-05] Virvou, M. (2005). *Combining Software Games with Education: Evaluation of Its Educational Effectivennes*. Dari jurnal *Educational Technology and Society*.
- [WAL-01] Walsh, D. (2001). *Sixth Annual Video and Computer Report Card*. Diambil pada tanggal 19 agustus 2014, dari Lembaga Penjaminan Mutu Pendidikan Jawa tengah:

<http://www.lmpjateng.go.id/web/index.php/arsip/karya-tulis-ilmiah/794-pengembangan-game-edukasi-untuk-media-bantu-pembelajaran-drill-and-practice-sebagai-persiapan-siswa>

[ZEP-10] Zepe, Kak. (2010). *Lagu Anak (Seluk Beluk, Sejarah, dll)*. Diambil pada tanggal 14 September 2014, dari forum kompas: <http://forum.kompas.com/musik/34175-lagu-anak-seluk-beluk-sejarah-dll.html>.

[ZIM-03] Salen, K. dan E.Zimmerman. (2003). *Rules of Play: Game Design Fundamentals*. MIT Press, Massachusetts.



LAMPIRAN

Lampiran 1. Test suite



Test suite

Nama :

NIM :

Berikut merupakan test suite untuk menguji fungsionalitas game edukasi pengenalan lagu anak dengan nama “box of music”. Pengujian meliputi proses yang ada pada tiap scene apakah sudah berjalan sesuai yang diinginkan. Pada test suite ini ada 2 jawaban yang dapat dipilih yaitu “Ya” berarti sudah sesuai dan “Tidak” masih perlu diperbaiki. Jawablah dengan tanda centang (√) pada kolom yang telah disediakan.

Scene Menu Utama		
Overall Proses Menu Utama	Tidak	Ya
1. Animasi background dapat bergerak dan animasi box pada background berputar?		
2. Background Music (BGM) dapat dihidupkan dan dimatikan melalui tombol bergambar nada lagu?		
3. tombol bergambar buku akan membuka menu tutorial?		

4.	Tombol back pada menu tutorial akan membuka tutorial sebelumnya?		
5.	Tombol next pada menu tutorial akan membuka tutorial selanjutnya?		
6.	Tombol play akan membuka scene level utama ?		
7.	Tombol train akan membuka scene level latihan?		
8.	Tombol berbentuk silang berarti membuka menu exit?		
9.	Tombol yes pada menu exit berarti keluar dari game?		
10.	Tombol no pada menu exit berarti kembali ke menu utama?		
11.	Tombol back pada mobile device akan membuka menu exit?		
Scene Level Utama dan Level Latihan			
Overall Proses Level Utama		Tidak	Ya
1.	Level yang terbuka akan menuju gameplay utama ketika dipilih?		
2.	Level yang terkunci tidak dapat dipilih?		
3.	Bintang pada tiap level didapat setelah bermain dan mendapatkan bintang?		
4.	Tombol back berarti akan menuju ke scene menu utama?		
5.	Tombol back pada mobile device berarti akan menuju ke menu utama?		
Overall Proses Level Latihan		Tidak	Ya
1.	Level yang terbuka akan menuju gameplay latihan ketika dipilih?		

2.	Level yang terkunci tidak dapat dipilih?		
3.	Tombol back berarti akan menuju ke scene menu utama?		
4.	Tombol back pada mobile device berarti akan menuju ke menu utama?		
Scene Gameplay Utama dan Gameplay Latihan			
Overall Proses Gameplay Utama		Tidak	Ya
1.	Awal permainan diawali dengan hitung mundur?		
2.	Background awan dan balon dapat bergerak?		
3.	Tanda "listen" berarti nada dari lagu dimainkan ?		
4.	Tanda "listen" berarti pemain tidak dapat memainkan alat musik box?		
5.	Tanda "your turn" berarti pemain dapat memainkan alat musik box?		
6.	Chance berkurang 1 ketika pemain salah dalam mengikuti nada yang diputar?		
7.	Score bertambah 50 dikalikan combo ketika pemain benar dalam mengikuti nada yang diputar?		
8.	Combo bar akan bertambah ketika pemain benar dan berkurang ketika pemain salah dalam mengikuti nada yang diputar?		
9.	Combo bertambah ketika combo bar full?		
10.	BGM dapat dihidupkan dan dimatikan melalui tombol bergambar nada?		
11.	Tombol yang bergambar 2 garis vertical berarti membuka menu pause?		

12.	Permainan berhenti ketika menu pause terbuka?		
13.	Tombol resume berarti permainan dilanjutkan?		
14.	Tombol try again berarti permainan diulang dari awal?		
15.	Tombol menu berarti kembali ke menu utama?		
16.	membuka menu game over ketika chance dari pemain 0 atau semua nada lagu dapat di mainkan dengan benar?		
17.	Menu game over akan memproses score yang didapat, best score dan jumlah bintang yang didapat?		
18.	Tombol try again pada menu game over berarti permainan diulang?		
19.	Tombol menu pada menu game over berarti kembali ke menu utama?		
20.	Tombol back pada android device akan membuka menu pause?		
Overall proses Gameplay Latihan		Tidak	Ya
1.	Background awan dan balon dapat bergerak?		
2.	BGM dapat dihidupkan dan dimatikan melalui tombol bergambar nada?		
3.	Tombol back berarti kembali ke menu utama?		
4.	Semua box dapat dimainkan pada menu "free mode"?		
5.	Menu "learn the song" memberikan tanda nada mana yang harus diikuti?		
6.	Menu "play the song" memainkan lagu secara otomatis?		
7.	Tombol back pada android device berarti kembali ke menu utama?		

Lampiran 2. Kuisioner *play testing*



Testing Questionnaire

Nama :

NIM :

***Dimohon untuk mengisi dengan sebenar-benarnya**

No.	Pertanyaan	Sangat kurang	Kurang	Cukup	Bagus	Sangat bagus
1.	Bagaimana dengan gameplay yang disajikan?					
*Saran untuk gameplay :						
2.	Bagaimana dengan efisiensi saat memainkan?					
*Saran untuk efisiensi :						
3.	Bagaimana dengan tingkat kesulitan dari game?					
*Saran untuk level dari game :						



4.	Seberapa menyenangkan game ini?					
*Saran agar game lebih menyenangkan :						
5.	Bagaimana dengan art yang disajikan pada game ini?					
*Saran untuk art dari game :						
3.	Bagaimana dengan manfaat yang didapat pemain dari game ini?					
*alasan dari jawaban anda :						

