BAB III
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan - perbaikannya.

3.1.1 Analisis Game Sejenis yang Sudah Ada

Sebelum merancang game yang akan dibuat, terlebih dahulu dilakukan analisis pada game sejenis yang sudah ada. Ini bertujuan agar game yang sudah dibuat nanti memiliki solusi dari permasalahan game yang sudah ada.

Game edukasi teori musik khususnya interval sudah ada beberapa, diantaranya *Intervals Music Quiz by Ms. Garrett* [**3**], *8notes Interval Ear Trainer* [**4**] dan *Woody's Music Theory Adventure* [**5**]. Setelah melakukan observasi terhadap tiga game tersebut, didapat kesimpulan seperti yang bisa dilihat pada tabel III.1

Tabel III.1 Analisis Game yang Sudah Ada

Nama Game	Jenis Game	Kelebihan	Kekurangan
Intervals Music Quiz by Ms. Garrett	Quiz	Quiz	*Gameplay* hanya quiz saja, tidak ada cerita
8notes Interval Ear Trainer	Ear Training	Ear Training	*Gameplay* hanya *ear training* saja, tidak ada cerita
Woody's Music Theory Adventure	Adventure + Quiz	Penggabungan antara quiz dengan *adventure game*	Tidak ada ear training

3.1.2 Analisis game yang akan dibangun

Game yang akan dibangun berupa game *adventure* yang dipadukan dengan quiz dan *ear training*. Game ini dibangun sebagai media pembelajaran alternatif yang lebih interaktif dan menyenangkan untuk mempelajari teori musik. Materi teori musik yang akan ditampilkan dalam game ini adalah tentang interval. Dengan adanya jalan cerita, *gameplay adventure*, serta memadukan quiz dan *ear training* sebagai unsur edukasi musiknya, diharapkan dapat menjadi alternatif solusi dari permasalahan yang ada di game sejenis yang sudah ada.

Fitur – fitur yang akan dibuat pada game ini adalah:

1. Pemberian materi
2. Game petualangan
3. Quiz
4. Ear training

3.1.2.1 Storyline

Putri diculik oleh monster. Untuk menyelamatkannya, Pangeran harus mendapat ilmu tentang interval dari 5 guru sebelum melawan monster.

1. Level 1

Pemain akan mendapatkan teori dasar interval dari guru 1. Kemudian pemain melakukan perjalanan untuk menemukan guru 2. Sebelum bisa belajar kepada guru 2, pemain akan diuji terlebih dahulu tentang teori yang diajarkan guru 1.

2. Level 2

Pemain akan mendapatkan ilmu tentang identitas interval bagian ke-1 dari guru 2. Kemudian pemain melakukan perjalanan untuk menemukan guru 3. Sebelum bisa belajar kepada guru 3, pemain akan diuji terlebih dahulu tentang teori yang diajarkan guru 2.

3. Level 3

Pemain akan mendapatkan ilmu tentang identitas interval bagian ke-2 dari guru 3. Kemudian pemain melakukan perjalanan untuk menemukan guru 4. Sebelum bisa belajar kepada guru 4, pemain akan diuji terlebih dahulu tentang teori yang diajarkan guru 3.

4. Level 4

Pemain akan mendapatkan ilmu tentang identitas interval bagian ke-3 dari guru 4. Kemudian pemain melakukan perjalanan untuk menemukan guru 5. Sebelum bisa belajar kepada guru 5, pemain akan diuji terlebih dahulu tentang teori yang diajarkan guru 4.

5. Level 5

Pemain akan mendapatkan ilmu tentang identitas interval bagian ke-4 dari guru 5. Kemudian pemain melakukan perjalanan untuk melawan monster dan menyelamatkan putri.

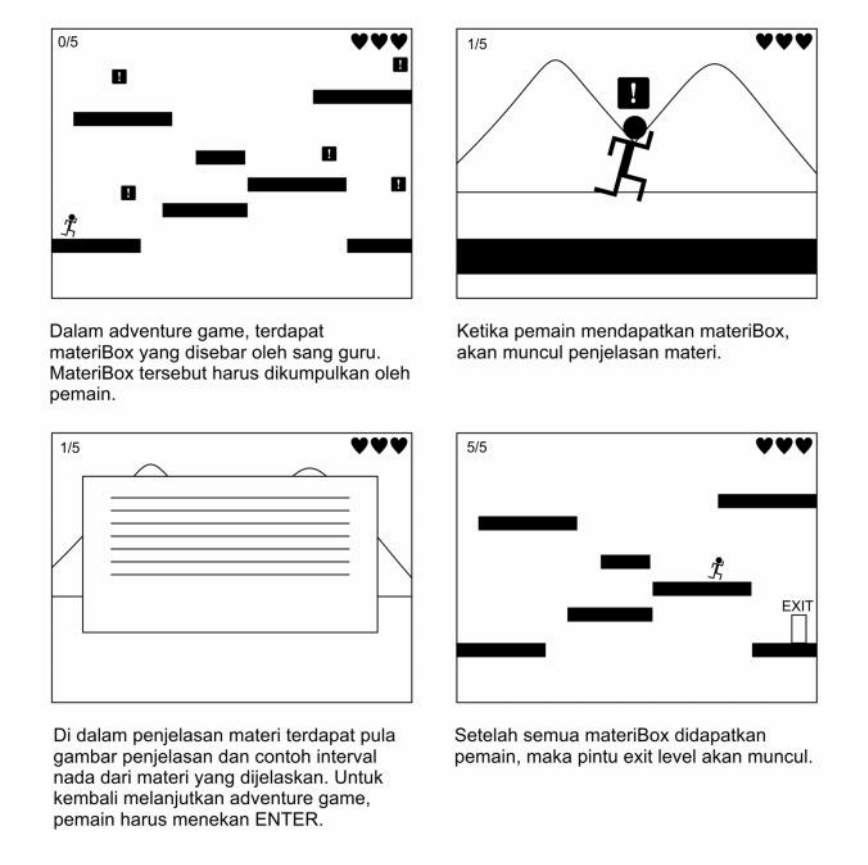
3.1.2.2 Storyboard

Storyboard adalah sketsa gambar yang disusun berurutan sesuai dengan naskah, dengan *storyboard* ide cerita dapat disampaikan kepada orang lain dengan lebih mudah, karena dapat menggiring khayalan seseorang mengikuti gambar-gambar yang tersaji sehingga menghasilkan persepsi yang sama.

Pada bagian ini akan dijelaskan *storyboard adventure* game dan pemberian materi, *storyboard* quiz, serta *storyboard ear training*.

1. Storyboard adventure game dan pemberian materi

Berikut ini adalah *storyboard* dari *gameplay adventure game* dan pemberian materi yang dapat dilihat pada gambar III.1 :



Gambar III.1 *Storyboard adventure game* dan pemberian materi

2. Storyboard quiz

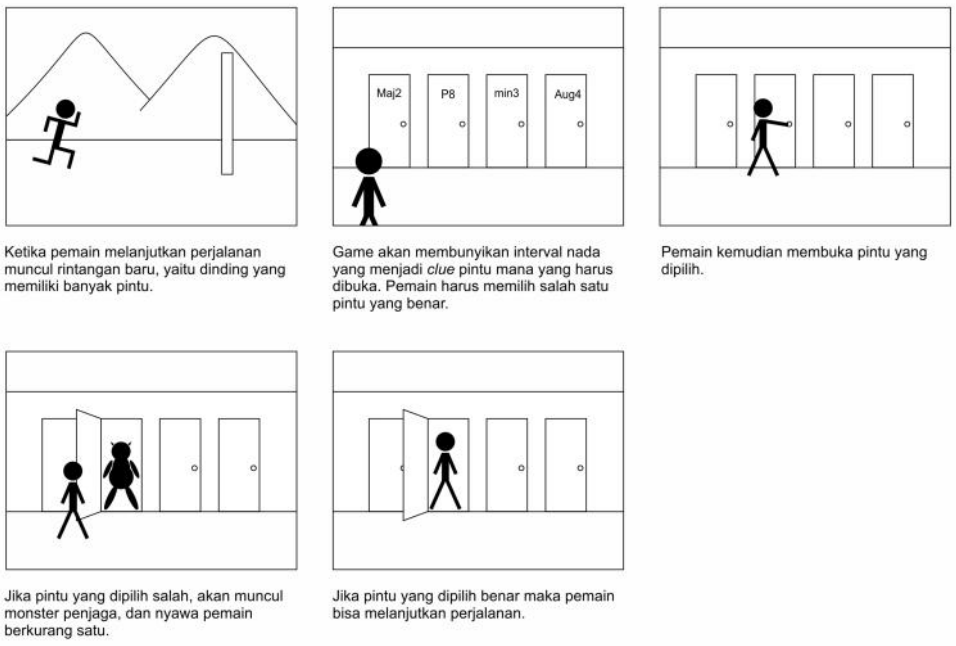
Berikut ini adalah *storyboard* dari *gameplay* quiz yang dapat dilihat pada gambar III.2:



Gambar III.2 *Storyboard* quiz

3. Storyboard ear training

Berikut ini adalah *storyboard* dari *gameplay ear training* yang dapat dilihat pada gambar III.3:



Gambar III.3 *Storyboard ear training*

3.1.2.3 Pelevelan

Dalam game ini terdapat 5 level, yang pada setiap levelnya mempunyai *gameplay* yang sama. Yang membedakan pada setiap levelnya yaitu :

1. Materi

Materi yang diberikan akan berbeda – beda sesuai level yang dimainkan. Materi yang diberikan diambil dari buku *Music Theory For Dummies* [**10**]. Materi – materi yang akan diberikan pada tiap level adalah:

a. Level 1

Materi yang diberikan di level ini adalah Pengenalan Interval, Unison, dan Octave.

b. Level 2

Tambahan materi yang diberikan di level ini adalah Major 2^{nd}, Minor 2^{nd}, dan Perfect 5^{th}.

c. Level 3

Tambahan materi yang diberikan di level ini adalah Major 3^{rd}, dan Minor 3^{rd}.

d. Level 4

Tambahan materi yang diberikan di level ini adalah Perfect 4^{th}, Major 6^{th}, dan Minor 6^{th}.

e. Level 5

Tambahan materi yang diberikan di level ini adalah Augmented 4^{th} / Diminished 5^{th}, Major 7^{th}, dan Minor 7^{th}.

2. Musuh

Semakin tinggi level, maka jenis musuh akan bertambah sehingga permainan menjadi lebih sulit. Jenis – jenis musuh yang terdapat dalam game ini adalah:

a. Monster Jelly
b. Monster Fly

Kemunculan monster – monster tersebut pada tiap level bisa dilihat pada tabel III.2 berikut :

Tabel III.2 Jumlah Musuh Tiap Level

Level	Monster Jelly	Monster Fly
1	3	0
2	5	0
3	4	2
4	3	3
5	3	4

Monster – monster tersebut memiliki bobot kesulitan yang berbeda yang bisa dihitung untuk keperluan *balancing* dari level yang dibuat. Monster Jelly memiliki bobot 1, sedangkan Monster Fly memiliki bobot 2. Perhitungan *balancing* dari level yang dibuat bisa dilihat pada tabel III.3 berikut :

Tabel III.3 *Level Balancing*

Level	Monster Jelly	Monster Fly	Skor
1	3 x 1 = 3	0 x 2 = 0	3
2	5 x 1 = 5	0 x 2 = 0	5
3	4 x 1 = 4	2 x 2 = 4	8
4	3 x 1 = 3	3 x 2 = 6	9
5	3 x 1 = 3	4 x 2 = 8	11

Dari tabel tersebut bisa dilihat bahwa skor masing – masing level meningkat dari level sebelumnya. Ini berarti bahwa terjadi peningkatan kesulitan pada tiap level.

3. Materi yang dijadikan soal quiz dan *ear training*

   Materi yang dijadikan soal quiz dan *ear training* bertambah sesuai materi yang diberikan pada *adventure game*. Selain itu, materi yang dijadikan soal pada level sebelumnya akan tetap muncul di level tersebut sehingga *range* materi yang ditanyakan akan menjadi lebih banyak.

4. Kesempatan replay pada ear training

   Pada *gameplay ear training* disediakan tombol *replay* untuk memainkan kembali interval soal, tetapi kesempatan untuk *replay* tersebut terbatas dan semakin tinggi level, maka semakin sedikit kesempatan *replay* yang diberikan. Pada level 1, kesempatan *replay* yang diberikan adalah 10 kali. Pada level 2, kesempatan *replay* yang diberikan adalah 8 kali. Pada level 3, kesempatan *replay* yang diberikan adalah 6 kali. Pada level 4, kesempatan *replay* yang diberikan adalah 4 kali. Sedangkan pada level 5, kesempatan *replay* yang diberikan adalah 2 kali.

3.1.2.4 Gameplay

Secara umum terdapat 4 bagian dalam tiap level pada game ini, yaitu :

1. Pemberian Materi

Pada bagian ini akan dijelaskan materi yang akan menjadi modal pemain untuk menjawab pertanyaan di bagian quiz dan *ear training*.

2. Platform / Adventure Game

Pada bagian ini pemain harus mencari *materiBox*. Pemberian materi dilakukan jika pemain mendapatkan *materiBox* tersebut. Jika semua *materiBox* di level tersebut telah berhasil didapatkan, maka pintu keluar level akan terbuka. Karakter dalam permainan digerakkan oleh pemain dengan *keyboard* untuk bergerak ke kiri, kanan, dan meloncat.

3. Quiz

Pada bagian ini pemain harus menjawab pertanyaan berupa teori tertulis yang teorinya telah dijelaskan sebelumnya pada bagian pemberian materi. Bentuk pertanyaan diibaratkan dengan mengisi password, jika salah satu jawaban salah atau urutannya tidak sesuai, maka semua jawaban dianggap salah. *Gameplay* quiz ini adalah dengan mencocokkan pertanyaan dengan pilihan jawaban yang disediakan.

4. Ear Training

Pada bagian ini, pemain akan dihadapkan rintangan yaitu dinding yang memiliki banyak pintu. Di pintu tersebut terdapat label berupa nama – nama interval. Game akan membunyikan dua buah nada yang intervalnya akan menjadi *clue* pintu mana yang harus dipilih pemain. Pertanyaan yang diberikan sebanyak 5 pertanyaan, dan pemain bisa memainkan kembali interval tersebut dengan menekan tombol *replay*. Jumlah kesalahan yang dibolehkan adalah 1 dari 5 pertanyaan dan akan dilihat dari skornya.

3.1.2.5 Scoring

Pada bagian ini akan dijelaskan bagaimana pemberian skor yang ada pada game ini. Pada game ini, bagian yang terdapat skor di dalamnya yaitu *Ear Training.* Pemain diberi 5 pertanyaan yang masing – masing skornya 20. Jika jawaban pemain salah maka tidak ada pengurangan nilai. Total kemungkinan skor tertinggi adalah 100. Jika skor pemain lebih besar atau sama dengan 80, maka pemain bisa melanjutkan ke level berikutnya.

3.1.3 Analisis Algoritma

Misalkan dalam *ear training* ada 10 soal yang akan diacak, maka array-nya adalah urutanSoal = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]. Array tersebut dimasukkan kedalam prosedur *shuffle* dimana proses pengacakan terjadi. Dari array tersebut didapat panjang array yang kemudian dimasukkan ke dalam variabel m (m = 10).

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

1. Ambil satu elemen secara acak dari elemen yang tersisa.

   Pengambilan elemen acak adalah berdasarkan elemen yang tersisa. Misalkan jika m = 9, maka elemen acak yang boleh diambil adalah 9 (array[0…8]).

2. Tukar dengan elemen saat ini.

   Penukaran dilakukan dengan memasukkan elemen saat ini (array[m]) ke dalam suatu variabel baru bernama *t*. Kemudian elemen saat ini (array[m]) diisi nilai dari elemen acak tadi (array[i]). Dan elemen acak (array[i]) diisi nilai dari variabel *t*.

3. Ulangi selama masih ada elemen yang tersisa

Selanjutnya dilakukan pengulangan sebanyak 10 kali untuk mengacak urutan soal tersebut. Iterasinya bisa dilihat pada tabel III.4.

Tabel III.4 Tabel iterasi Fisher-Yates Shuffle

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

Kolom *m* menunjukkan index elemen saat ini. Kolom *i* menunjukkan index dari elemen yang akan ditukar, nilai *i* tersebut diambil secara acak dari range yang diperbolehkan (*m*). Kolom array[i] menunjukkan nilai yang terdapat pada array ke-*i*. Kolom t menunjukkan nilai yang terdapat pada array ke-*m* (elemen saat ini). Array yang sudah *fix* menunjukkan array yang sudah tidak boleh ditukar kembali nilainya, jumlahnya bertambah seiring bertambahnya iterasi. Kemudian kolom array yang sudah di *swap* menunjukkan isi array setelah elemennya ditukar / *swap* pada setiap iterasi.

Tabel III.5 berikut adalah tahapan dan langkah algoritma Fisher-Yates Shuffle dan dimasukkan ke dalam *pseudocode*.

Tabel III.5 Tahapan dan langkah algoritma Fisher-Yates Shuffle

No	Algoritma	Pseudocode
1	Ketika masih ada elemen tersisa untuk diacak	while (m){
2	Ambil elemen secara acak dari elemen yang tersisa	i = Math.floor(Math.random() * m--);

3	Kemudian tukar dengan elemen saat ini	t = array[m];
		array[m] = array[i];
		array[i] = t;
		}

3.1.4 Analisis Ear Training

Pada bagian ini akan dibahas bagaimana analisis *ear training* dalam aspek psikologis, teknis, dan pembagian interval ke dalam tiap level ditinjau dari sisi *ear training*.

3.1.4.1 Psikologis

Pada bagian ini akan dibahas apakah *ear training* dapat dilakukan kepada rentang usia target pengguna yaitu 9 – 12 tahun. Setelah dilakukan studi literatur tentang perkembangan kemampuan musikal pada anak [**25**], didapat data sebagai berikut :

1. Hargreaves dan Zimmerman (1992) menyatakan bahwa “Pada usia 8 tahun, persepsi melodi anak-anak beroperasi dalam sistem tonal yang

semakin stabil. Pada usia ini informasi melodi disimpan dan diproses sesuai dengan referensi tonal bukan oleh skema kontur" [**25**].

2. Davidson, McKernon, dan Gardner (1981) menemukan bahwa anak berusia 5 tahun belajar bernyanyi baik dalam kunci yang sama atau mengubah kunci secara tiba-tiba dan kemudian tetap didalamnya [**25**].
3. Imberty (1969) menyimpulkan dari penelitiannya tentang *cadences* bahwa sampai usia 5 tahun, anak mempersepsikan bahwa lagu selesai ketika musik berhenti. Pada usia 8 tahun anak telah merasakan efek dari *perfect cadence* (V ke I), dan pada usia 10 tahun anak sudah mengerti *perfect* dan *imperfect cadences* [**25**].

Dari data tersebut bisa dilihat bahwa pada usia 5 tahun, kemampuan musikal anak sudah terlihat tetapi masih belum pada tahap yang stabil. Baru pada usia 8 tahun kemampuan musikal anak sudah mencapai tahap yang stabil dan mulai matang. Oleh karena itu dapat disimpulkan bahwa *ear training* bisa dilakukan pada rentang usia target pengguna yaitu 9 – 12 tahun.

3.1.4.2 Teknis

Ear training dilakukan dengan membunyikan dua buah nada yang kemudian harus dijawab intervalnya. Nada pertama (*root*) yang dibunyikan adalah nada acak, tetapi nada kedua diambil sesuai interval yang ingin dimainkan. Satuan jarak interval dari nada *root* ke nada kedua bisa dilihat pada tabel III.6.

Tabel III.6 Satuan Jarak Interval

Nama Interval	Interval	Interval (dalam satuan semitone)	Keterangan
Unison	0	0	*root* + 0
Minor 2nd	½	1	*root* + 1
Major 2nd	1	2	*root* + 2
Minor 3rd	1 ½	3	*root* + 3
Major 3rd	2	4	*root* + 4
Perfect 4th	2 ½	5	*root* + 5
Augmented 4th / Dim 5th	3	6	*root* + 6
Perfect 5th	3 ½	7	*root* + 7
Minor 6th	4	8	*root* + 8
Major 6th	4 ½	9	*root* + 9
Minor 7th	5	10	*root* + 10

Nama Interval	Interval	Interval (dalam satuan semitone)	Keterangan
Major 7^{th}	5 ½	11	*root* + 11
Perfect 8^{th} / Octave	6	12	*root* + 12

Range nada yang disediakan adalah 2 oktaf, diurutkan dari C1 (dengan kode 1) sampai B2 (dengan kode 24) seperti yang bisa dilihat pada tabel III.7.

Tabel III.7 Pengkodean Nada

Nada	Kode
C1	1
C#1	2
D1	3
D#1	4
E1	5
F1	6
F#1	7
G1	8
G#1	9
A1	10
A#1	11
B1	12
C2	13
C#2	14
D2	15
D#2	16
E2	17
F2	18
F#2	19
G2	20
G#2	21
A2	22
A#2	23
B2	24

Jadi jika pertanyaan adalah Augmented 4^{th} dengan *root* hasil *random* adalah C1, maka nada dua-nya adalah 1 + 6 = 7 atau F#1.

3.1.4.3 Pembagian Level Interval

Interval – interval tadi kemudian disebar ke lima level yang tersedia dalam game yang akan dibangun. Pembagian interval adalah berdasarkan kemiripan

suara dari tiap interval. Sebelumnya, interval – interval tersebut dibagi menjadi dua kelompok berdasarkan *interval quality*-nya. Kelompok pertama adalah kelompok *Major – Minor*. Yang kedua adalah *special case interval* yang terdiri atas *Unison, Perfect, Augmented* dan *Diminished*. Kelompok *Major – Minor* disusun berdasarkan *interval quantity*-nya, dari 2^{nd}, 3^{rd}, sampai 7^{th}. Sedangkan *special case interval* disusun berdasarkan kemiripan suaranya dari *Unison, Octave, Perfect* 5^{th}, *Perfect* 4^{th}, sampai *Augmented* 4^{th}. Penyusunan tersebut didasari oleh perbandingan frekuensi dari masing – masing interval, dimana *Unison* memiliki perbandingan 1:1, *Octave* 2:1, *Perfect* 5^{th} 3:2, *Perfect* 4^{th} 4:3, dan *Augmented* 4^{th} 45:32. Setelah itu kelompok tersebut dikombinasikan dalam satu level, dan hasil pembagiannya bisa dilihat pada tabel III.8.

Tabel III.8 Pembagian Level Interval

Level	Interval
1	Unison, Octave
2	Perfect 5^{th}, Major 2^{nd}, Minor 2^{nd}
3	Major 3^{rd}, Minor 3^{rd}
4	Perfect 4^{th}, Major 6^{th}, Minor 6^{th}
5	Augmented 4^{th} / Dim 5^{th}, Major 7^{th}, Minor 7^{th}

3.1.5 Spesifikasi Kebutuhan Perangkat Lunak (SKPL)

Spesifikasi kebutuhan perangkat lunak berisi deskripsi dari kebutuhan perangkat lunak yang akan dibangun. SKPL dari game edukasi *adventure* teori musik bisa dilihat pada tabel III.9.

Tabel III.9 Spesifikasi Kebutuhan Perangkat Lunak (SKPL)

Kode Kebutuhan	Kebutuhan
SKPL-F-1	Sistem dapat menampilkan materi
SKPL-F-2	Sistem dapat menampilkan *adventure game*
SKPL-F-3	Sistem dapat menampilkan quiz
SKPL-F-4	Sistem dapat menampilkan *ear training*
SKPL-F-5	Sistem dapat menampilkan menu *about*

3.1.6 Analisis Kebutuhan Non-Fungsional

Analisis non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

Pada analisis kebutuhan sistem non-fungsional ini dijelaskan analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras, dan analisis pengguna.

3.1.6.1 Analisis Kebutuhan Perangkat Keras

Agar aplikasi dapat berjalan dengan baik, maka dibutuhkan perangkat keras yang sesuai dengan kebutuhan aplikasi. Di bagian ini akan dijelaskan analisis kebutuhan perangkat keras dari sisi pengembang dan dari sisi pemain.

1. Dari sisi pengembang

Perangkat keras yang digunakan untuk membangun game ini adalah sebagai berikut:

1. Prosesor 2 GHz
2. Memori 1 GB
3. *Space* kosong di *harddisk* 5 GB
4. VGA card 256 MB
5. Monitor 1280 x 900
6. Mouse dan keyboard
7. Speaker

2. Dari sisi pemain

Spesifikasi minimum perangkat keras yang dibutuhkan agar dapat menjalankan game edukasi teori musik ini adalah sebagai berikut:

1. Prosesor dengan kecepatan 1.8 Ghz
2. Memori 256 Mb

3. *Space* kosong di *harddisk* 15 MB
4. VGA card
5. Monitor
6. Mouse dan keyboard
7. Speaker

3.1.6.2 Analisis Kebutuhan Perangkat Lunak

Perangkat keras komputer tidak berarti tanpa perangkat lunak, begitu juga sebaliknya. Jadi perangkat lunak dan perangkat keras saling mendukung satu sama lain. Perangkat keras hanya berfungsi jika diberikan instruksi - intruksi kepadanya. Instruksi - instruksi inilah yang disebut dengan perangkat lunak.

1. Dari sisi pengembang

Perangkat lunak yang digunakan untuk mengembangkan game ini adalah sebagai berikut:

1. Sistem operasi Windows XP SP3, Vista SP1, Windows 7, Mac OSX 10.5.7, Mac OSX 10.6.3
2. FlashBuilder 4.5 sebagai *Integrated Development Environment* (IDE)-nya
3. Flex SDK 4.6
4. Flash Debugger 11
5. Aplikasi pengolahan grafis menggunakan Adobe Illustrator CS5 dan Adobe Fireworks CS5
6. Aplikasi pengolahan suara menggunakan Apple Garage Band
7. Aplikasi pembuatan *map adventure game* menggunakan DAME

2. Dari sisi pemain

Untuk menjalankan game edukasi teori musik ini dibutuhkan perangkat lunak, antara lain:

1. Sistem operasi Windows, Mac OS, atau Linux
2. Flash Player 10

3.1.6.3 Analisis Pengguna

Analisis pengguna merupakan analisis terhadap pengguna yang akan menggunakan aplikasi yang telah dibangun. Perangkat keras dan perangkat lunak yang ada tidak akan berguna apabila tidak ada pengguna yang mengoperasikannya. Oleh karena itu dibuatlah analisis pengguna yang berdasarkan pada user *profile*nya, yakni berdasarkan *user knowledge & experience, user job, task & needs, user physicological characteristic*, dan *user physical characteristic* seperti yang bisa dilihat pada tabel III.10.

Tabel III.10 Analisis Pengguna

KNOWLEDGE AND EXPERIENCE	
Computer Literacy	Rendah, minimal bisa menggunakan komputer
Pengalaman Sistem	Sedang, minimal biasa menggunakan sistem operasi Windows
Pengalaman Aplikasi	Sedang, minimal pernah memainkan game *adventure* sejenis seperti Super Mario Bros
Penggunaan Sistem Lain	Jarang
Edukasi	Sekolah Dasar
Kemampuan Membaca	Bisa membaca dengan lancar. Setara dengan kemampuan membaca anak kelas 3 SD.
Kemampuan Mengetik	"*Hunt and Peck*" (10 WPM).
Bahasa / Kultur	Bahasa Indonesia
JOB / TASK / NEED	
Frekuensi Penggunaan	*Occasional*
Tingkat Kepentingan *Task*	Lemah
Interaksi Sosial	Tidak diperlukan
PSYCHOLOGICAL CHARACTERISTICS	
Attitude	Positif
Motivasi	Tinggi
Stress Level	*No Stress*

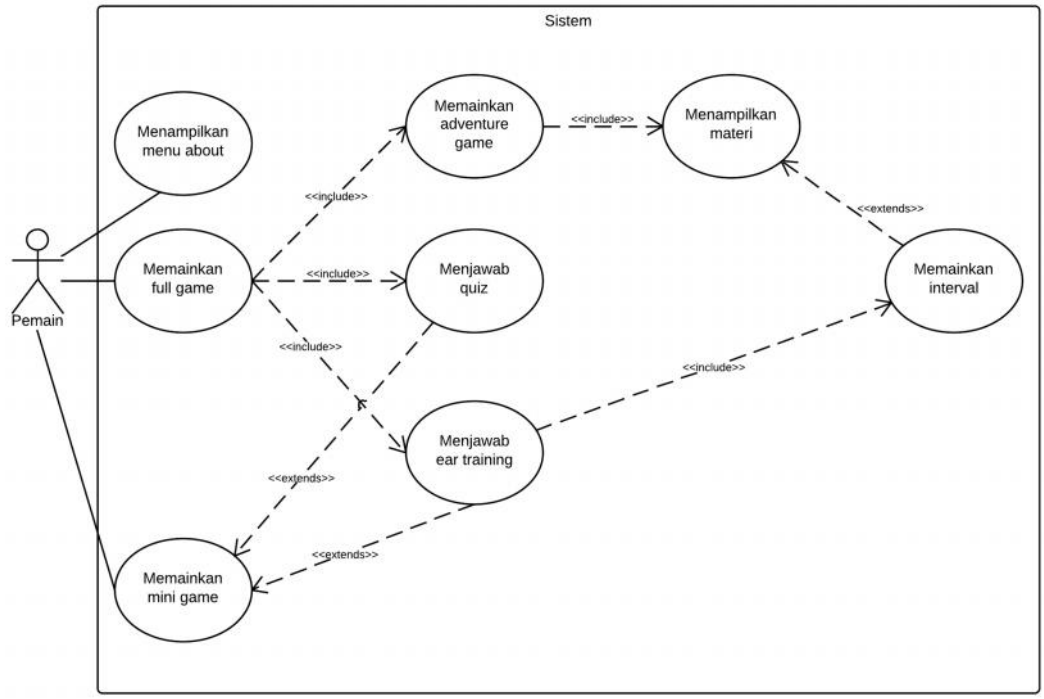
PHYSICAL CHARACTERISTICS	
Umur	9 – 12 tahun
Jenis Kelamin	Laki – laki dan Perempuan
Handedness	Kanan, Kidal, dan Keduanya
Disabilities	Tidak ada masalah pendengaran.

3.1.7 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik sesuai kebutuhan. Pada bagian ini akan dibahas tentang *use case diagram, activity diagram, class diagram*, dan *sequence diagram.*

3.1.7.1 Use Case Diagram

Pada sub bab berikut ini dijelaskan bagaimana *Use Case* yang terjadi dalam pembangunan game ini yang dimodelkan dalam sekumpulan *Use Case* dan *Actor* dan bagaimana hubungan–hubungannya yang akan dipetakan dalam diagram *Use Case,* dilengkapi dengan skenario untuk menjelaskan dari gambaran *Use Case* yang ada dengan menggambarkan skenario per *Use Case. Use Case Diagram* dapat dilihat pada gambar III.4:



Gambar III.4 Use Case Diagram

1. Actor Definition

Actor Definition berfungsi untuk menjelaskan *Actor* yang terdapat pada *Use Case Diagram*. *Actor Definition* diterangkan pada tabel III.11.

Tabel III.11 *Actor Definition*

No	Actor	Deskripsi
1	Pemain	Orang yang memainkan game

2. Use Case Definition

Use Case Definition berfungsi untuk menjelaskan fungsi *Use Case* yang terdapat pada *Use Case Diagram*. *Use Case Definition* diterangkan pada tabel III.12 di bawah ini.

Tabel III.12 *Use Case Definition*

No	Use Case	Deskripsi
1	Memainkan *full game*	Proses untuk memainkan permainan utama
2	Menampilkan menu *about*	Proses untuk menampilkan menu *about*
3	Memainkan *mini game*	Proses untuk memainkan *mini game*
4	Memainkan adventure	Proses untuk melakukan permainan berupa

	game	*adventure game*
5	Menjawab quiz	Proses untuk melakukan quiz berupa soal tertulis
6	Menjawab ear training	Proses untuk melakukan *ear training* berupa interval nada yang dimainkan
7	Menampilkan materi	Proses untuk menampilkan materi edukasi
8	Memainkan interval	Proses untuk memainkan interval

3. Use Case Descriptions

Sebuah diagram yang menunjukkan *use case* dan aktor mungkin menjadi titik awal yang baik, tetapi tidak memberikan detail yang cukup untuk desainer sistem untuk benar-benar memahami persis bagaimana kebutuhan sistem akan dipenuhi. Cara terbaik untuk mengungkapkan informasi penting ini adalah dalam bentuk penjelasan berbasis teks [**1**].

a. Use Case Description Memainkan Full Game

Use Case Description dari *Use Case* Memainkan *Full Game* dijelaskan pada tabel III.13 berikut :

Tabel III.13 *Use Case Description* Memainkan *Full Game*

Nama Use Case	Memainkan Full Game	
Tujuan	Untuk memainkan permainan utama	
Kondisi Awal	Pemain berada di menu utama	
Kondisi Akhir	Permainan selesai dilakukan	
Pemicu	Pemain memilih menu mainkan	
Aliran Utama	Step	Aksi
	1.	Pemilih memilih menu mainkan
	2	Sistem menampilkan menu pilih level
	3. Include::Memainkan Adventure Game	Sistem menampilkan *adventure game*
	4. Include::Menjawab Quiz	Sistem menampilkan quiz
	5. Include::Menjawab Ear Training	Sistem menampilkan *ear training*

b. Use Case Description Menampilkan Menu About

Use Case Description dari *Use Case* Menampilkan Menu *About* dijelaskan pada tabel III.14 berikut :

Tabel III.14 *Use Case Description* Menampilkan Menu *About*

Nama Use Case	Menampilkan Menu About	
Tujuan	Untuk menampilkan menu *about*	
Kondisi Awal	Pemain berada di menu utama	
Kondisi Akhir	Sistem menampilkan menu *about*	
Pemicu	Pemain memilih menu *about*	
Aliran Utama	Step	Aksi
	1.	Pemain memilih menu *about*
	2.	Sistem menampilkan menu *about*

c. Use Case Description Memainkan Mini Game

Use Case Description dari *Use Case* Memainkan *Mini Game* dijelaskan pada tabel III.15 berikut :

Tabel III.15 *Use Case Description* Memainkan *Mini Game*

Nama Use Case	Memainkan Mini Game	
Tujuan	Untuk memainkan *mini game*	
Kondisi Awal	Pemain berada di menu utama	
Kondisi Akhir	Sistem menampilkan *mini game* quiz atau *ear training*	
Pemicu	Pemain memilih menu *mini game*	
Aliran Utama	Step	Aksi
	1.	Pemain memilih menu *mini game*
	2.	Sistem menampilkan menu *mini game*
	3.	Pemain memilih *mini game* quiz
	4.	Sistem menampilkan menu pilih level
	5.	Pemain memilih level
	6. Extends::Menjawab Quiz	Sistem menampilkan quiz
Ekstensi	Step	Aksi Cabang
	3.1	Pemain memilih mini game ear training
	3.2 Include::Memilih Level	Sistem menampilkan menu pilih level
	3.3	Pemain memilih level
	3.4 Extends::Menjawab Ear Training	Sistem menampilkan *ear training*

d. Use Case Description Memainkan Adventure Game

Use Case Description dari *Use Case* Memainkan *Adventure Game* dijelaskan pada tabel III.16 berikut :

Tabel III.16 *Use Case Description* Memainkan *Adventure Game*

Nama Use Case	Memainkan Adventure Game	
Tujuan	Untuk melakukan permainan berupa *adventure game*	
Kondisi Awal	Pemain sudah memilih level	
Kondisi Akhir	Menampilkan quiz	
Pemicu	Pemain sudah memilih level	
Aliran Utama	Step	Aksi
	1.	Sistem mengambil peta sesuai level yang dipilih
	2.	Sistem menampilkan peta permainan
	3.	Pemain menggerakkan karakter permainan
	4.	Sistem menggerakkan pemain sesuai input pemain
	5.	Cek *collision* dengan musuh
	6.	Cek *collision* dengan ranjau / rintangan
	7. Include::Menampilkan materi	Jika pemain mendapatkan *materiBox*, sistem menampilkan materi
	8.	Jika semua *materiBox* sudah didapatkan, sistem memunculkan *exit level*
Ekstensi	Step	Aksi Cabang
	5.1	Jika bertabrakan dengan musuh, cek posisi pemain dan musuh
	5.1.1	Jika posisi pemain lebih tinggi dari musuh, maka musuh mati
	5.1.2	Jika posisi pemain lebih rendah atau sama dengan musuh, maka pemain mati dan *reset* posisi pemain
	6.1	Jika bertabrakan dengan ranjau maka pemain mati dan *reset* posisi pemain

e. Use Case Description Menjawab Quiz

Use Case Description dari *Use Case* Menjawab Quiz dijelaskan pada tabel III.17 berikut :

Tabel III.17 *Use Case Description* Menjawab Quiz

Nama Use Case	Menjawab Quiz	
Tujuan	Untuk melakukan quiz berupa soal tertulis	
Kondisi Awal	Pemain sudah menyelesaikan *adventure game*	
Kondisi Akhir	Menampilkan *Ear Training*	
Pemicu	Pemain sudah menyelesaikan *adventure game*	
Aliran Utama	Step	Aksi
	1.	Sistem mengambil soal dari *storage*

	2.	Sistem mengacak soal
	3.	Sistem menampilkan soal
	4.	Pemain menjawab soal
	5.	Sistem mengecek jawaban
	6.	Sistem menampilkan hasil quiz
Ekstensi	Step	Aksi Cabang
	5.1	Jawaban salah
	5.2	Sistem menampilkan jawaban yang benar

f. Use Case Description Menjawab Ear Training

Use Case Description dari *Use Case* Menjawab *Ear Training* dijelaskan pada tabel III.18 berikut :

Tabel III.18 *Use Case Description* Menjawab *Ear Training*

Nama Use Case	Menjawab Ear Training	
Tujuan	Untuk melakukan *ear training* berupa interval nada yang dimainkan	
Kondisi Awal	Pemain sudah menyelesaikan quiz	
Kondisi Akhir	Menampilkan hasil *ear training*	
Pemicu	Pemain sudah menyelesaikan quiz	
Aliran Utama	Step	Aksi
	1.	Sistem mengambil soal dari *storage*
	2.	Sistem mengacak soal
	3.	Sistem menampilkan soal
	4. Include::Memainkan Interval	Sistem memainkan interval soal
	5.	Pemain menjawab soal
	6.	Sistem mengecek jawaban
	7.	Sistem melakukan proses rekam jumlah jawaban benar
	8.	Mengecek jumlah soal yang sudah dijawab
	9.	Sistem menampilkan hasil ear training
Ekstensi	Step	Aksi Cabang
	4.1	Pemain menekan tombol *replay*
	4.2	Kurangi satu kesempatan *replay*
	4.2.3	Jika kesempatan replay habis, set tombol ‘replay’ *inactive*
	4.3	Sistem memainkan kembali interval soal
	6.1	Jawaban salah
	6.2	Sistem menampilkan jawaban benar

g. Use Case Description Menampilkan Materi

Use Case Description dari *Use Case* Menampilkan Materi dijelaskan pada tabel III.19 berikut :

Tabel III.19 *Use Case Description* Menampilkan Materi

Nama Use Case	Menampilkan Materi	
Tujuan	Untuk menampilkan materi edukasi	
Kondisi Awal	Jumlah *materiBox* yang sudah didapat belum bertambah	
Kondisi Akhir	Jumlah *materiBox* yang sudah didapat bertambah satu	
Pemicu	Pemain mendapatkan *materiBox*	
Aliran Utama	Step	Aksi
	1.	Sistem mengambil materi dari *storage*
	2.	Sistem menampilkan materi
	3.	Sistem menambah jumlah *materiBox* yang sudah didapat
	4.	Sistem menutup dialog materi
Ekstensi	Step	Aksi Cabang
	2.1	Pemain memainkan contoh
	2.2 Extends::Memainkan Interval	Sistem memainkan contoh interval

h. Use Case Description Memainkan Interval

Use Case Description dari *Use Case* Memainkan Interval dijelaskan pada tabel III.20 berikut :

Tabel III.20 *Use Case Description* Memainkan Interval

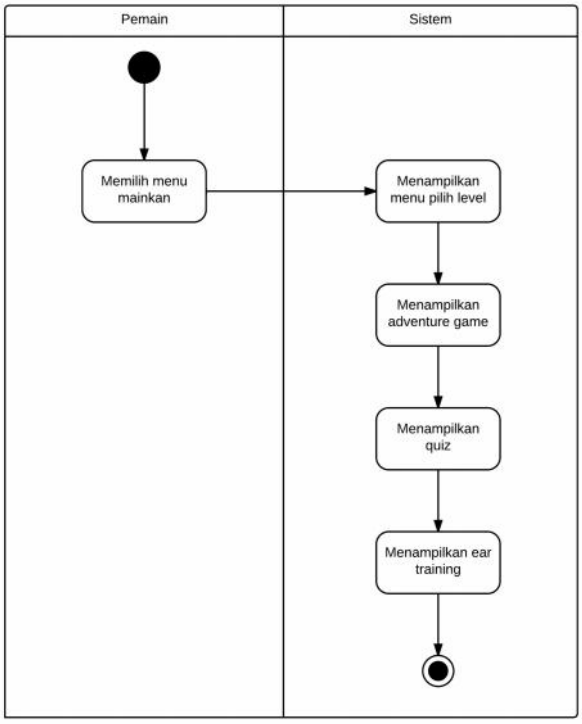
Nama Use Case	Memainkan Interval	
Tujuan	Untuk memainkan interval	
Kondisi Awal	Jenis interval sudah terdefinisi	
Kondisi Akhir	Sistem memainkan interval nada	
Pemicu	Sistem membuat instansi objek dari kelas interval	
Aliran Utama	Step	Aksi
	1.	Sistem mendapat input nama interval
	2.	Sistem mengacak nada *root*
	3.	Sistem mengisi interval sesuai dengan nama interval
	4.	Sistem membuat nada kedua dari nama pertama ditambah interval
	5.	Sistem memainkan interval nada

3.1.7.2 Activity Diagram

Activity Diagram menunjukkan aktivitas oleh manusia atau objek dalam cara yang mirip dengan *Flow Chart*.

1. Activity Diagram Memainkan Full Game

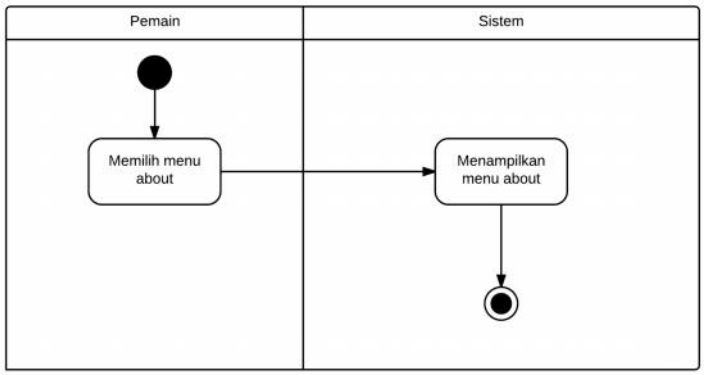
Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas Memainkan *Full Game* yang dapat dilihat pada gambar III.5.



Gambar III.5 *Activity Diagram* Memainkan *Full Game*

2. Activity Diagram Menampilkan Menu About

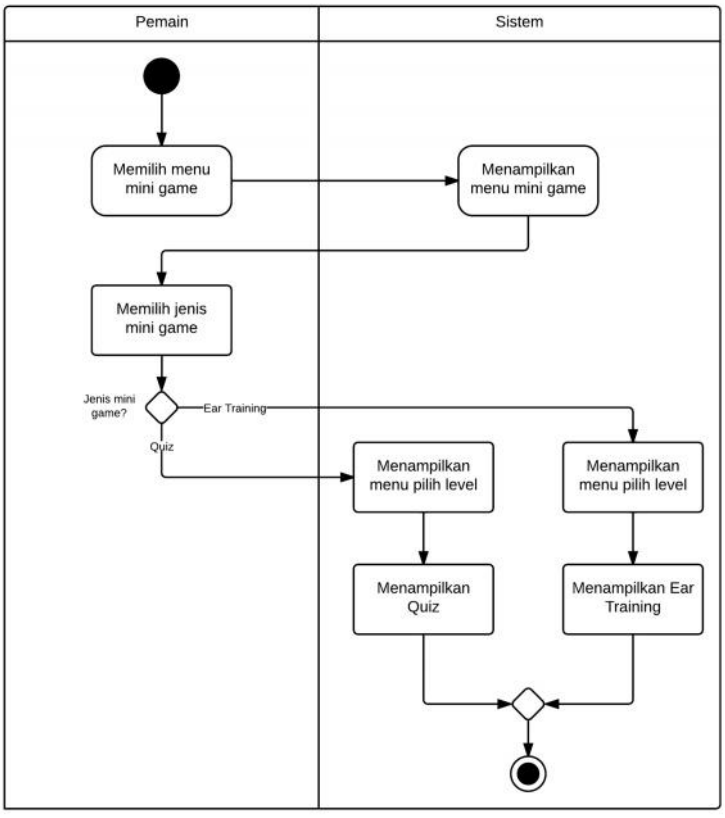
Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas Menampilkan Menu *About* yang dapat dilihat pada gambar III.6.



Gambar III.6 *Activity Diagram* Menampilkan Menu *About*

3. Activity Diagram Memainkan Mini Game

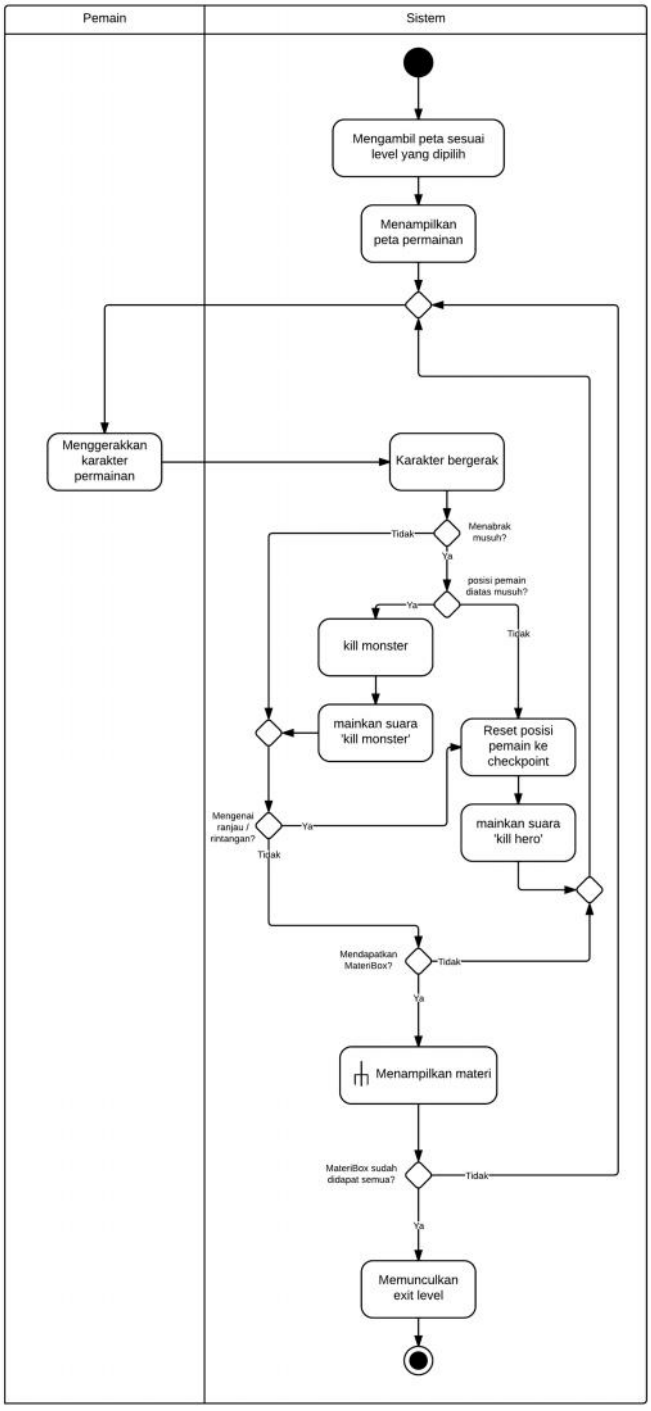
Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas Memainkan *Mini Game* yang dapat dilihat pada gambar III.7.



Gambar III.7 *Activity Diagram* Memainkan *Mini Game*

4. Activity Diagram Memainkan Adventure Game

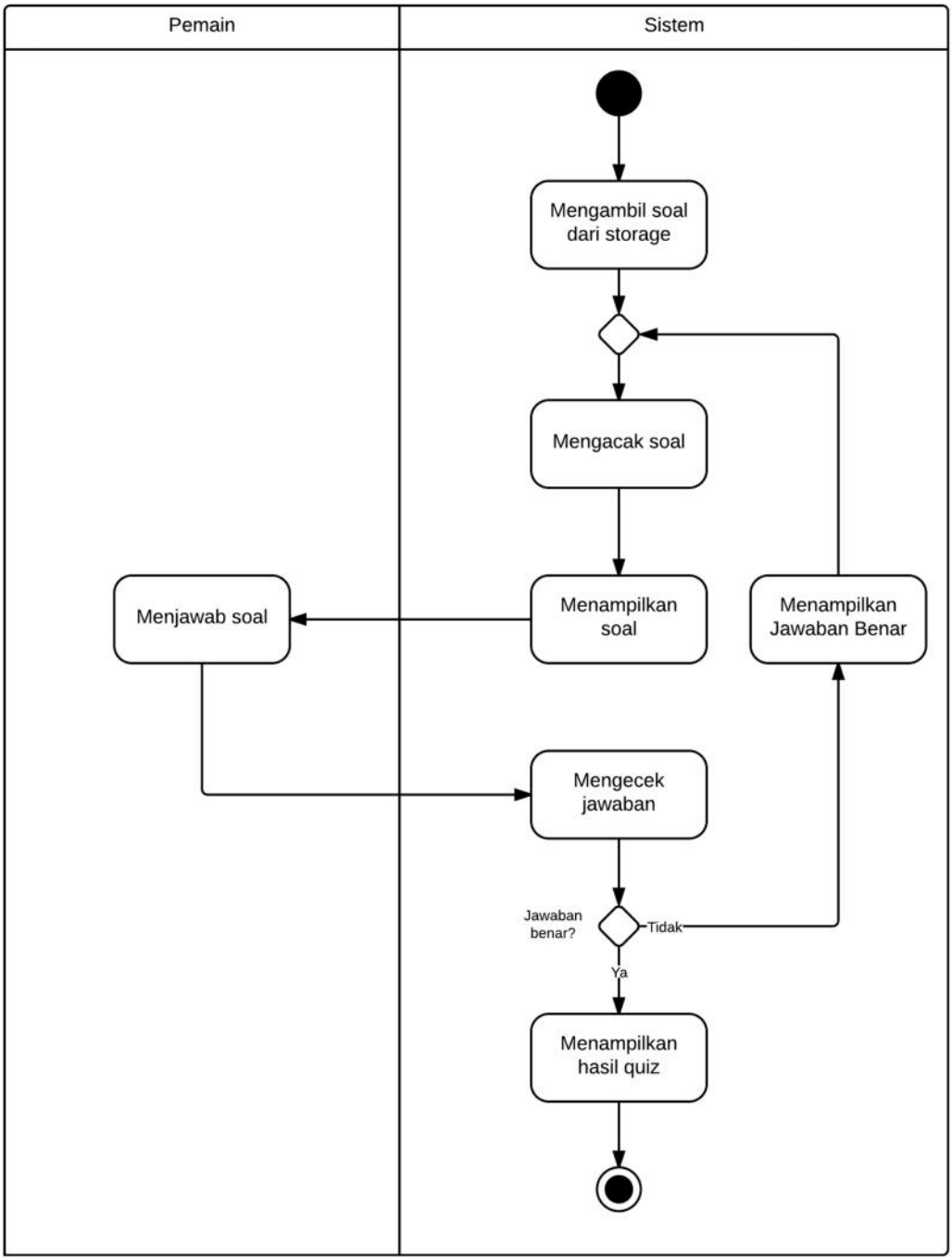
Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas Memainkan *Adventure Game* yang dapat dilihat pada gambar III.8.



Gambar III.8 *Activity Diagram* Memainkan *Adventure Game*

5. Activity Diagram Menjawab Quiz

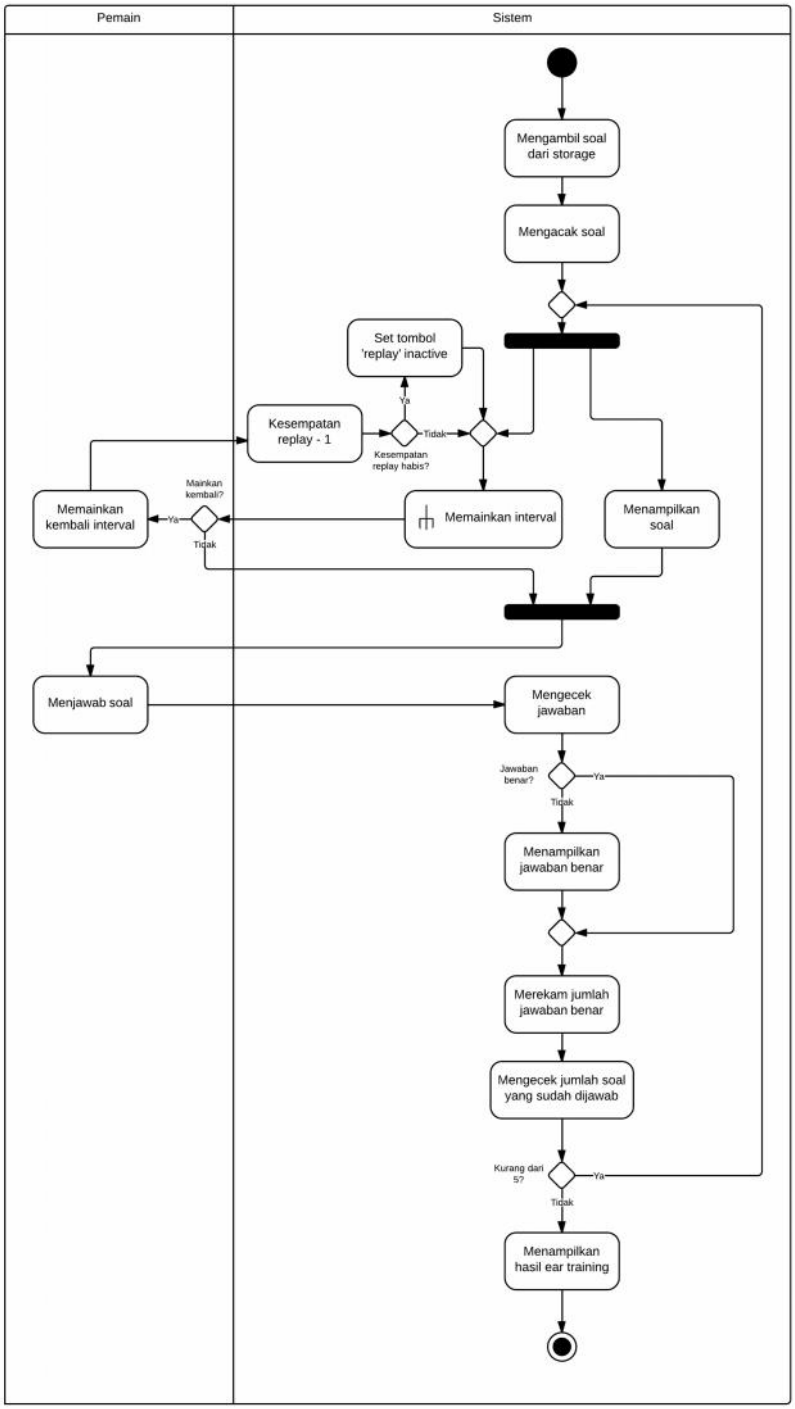
Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas Menjawab Quiz yang dapat dilihat pada gambar III.9.



Gambar III.9 *Activity Diagram* Menjawab Quiz

6. Activity Diagram Menjawab Ear Training

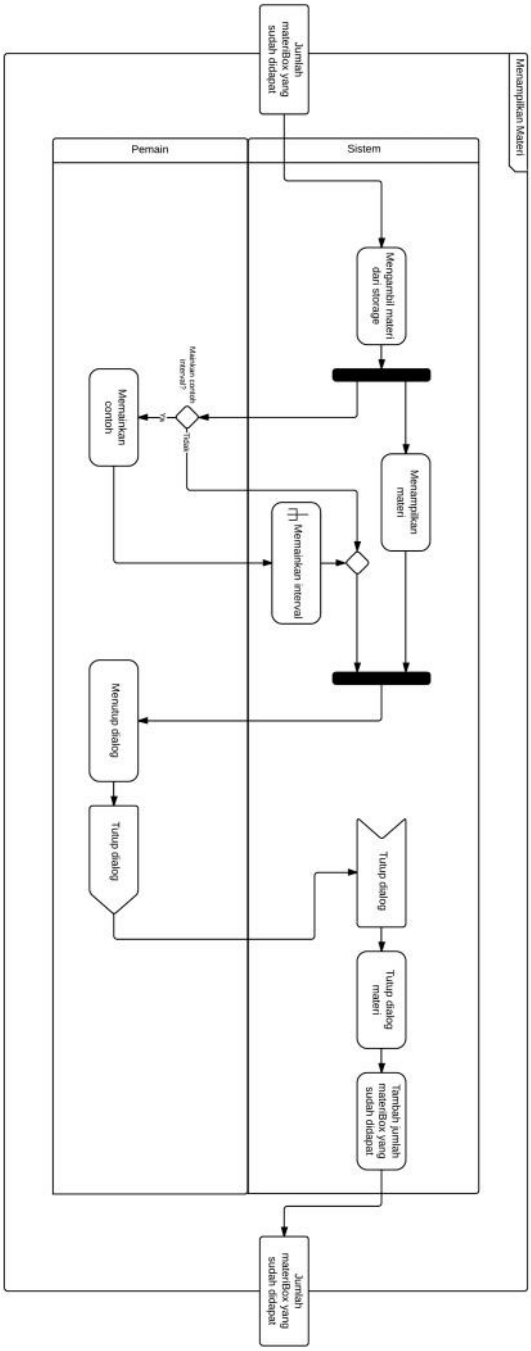
Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas Menjawab *Ear Training* yang dapat dilihat pada gambar III.10.



Gambar III.10 *Activity Diagram* Menjawab *Ear Training*

7. *Activity Diagram* Menampilkan Materi

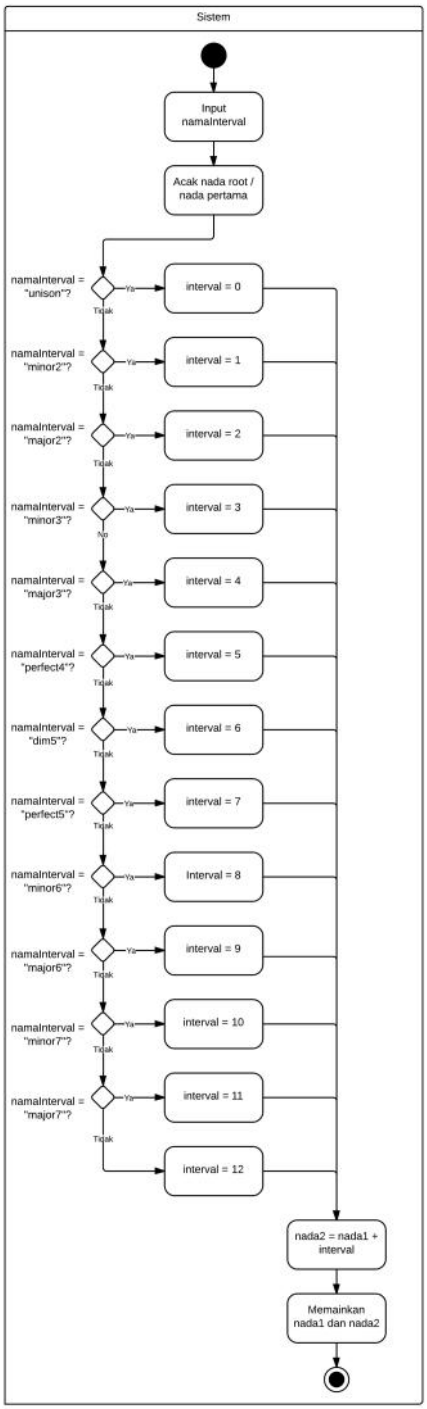
Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas Menampilkan Materi yang dapat dilihat pada gambar III.11.



Gambar III.11 *Activity Diagram* Menampilkan Materi

8. *Activity Diagram* Memainkan Interval

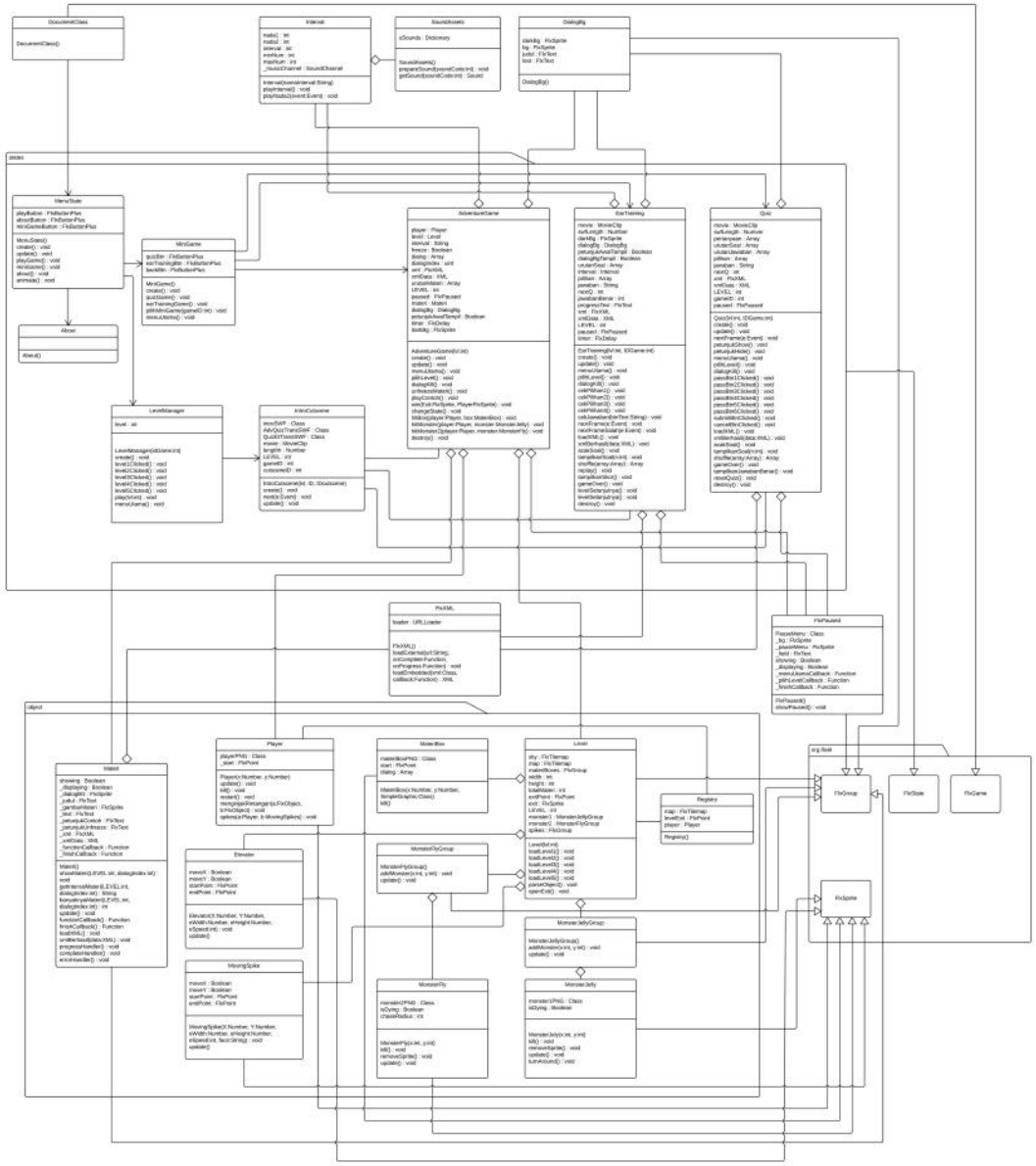
Berikut ini adalah diagram yang menunjukan alur aksi pada aktivitas Memainkan Interval yang dapat dilihat pada gambar III.12.



Gambar III.12 *Activity Diagram* Memainkan Inverval

3.1.7.3 Class Diagram

Class Diagram menunjukkan kelas dan bagaimana mereka dapat dipasang bersama-sama. Perancangan *class diagram* pada game ini dapat dilihat pada gambar III.13 berikut.



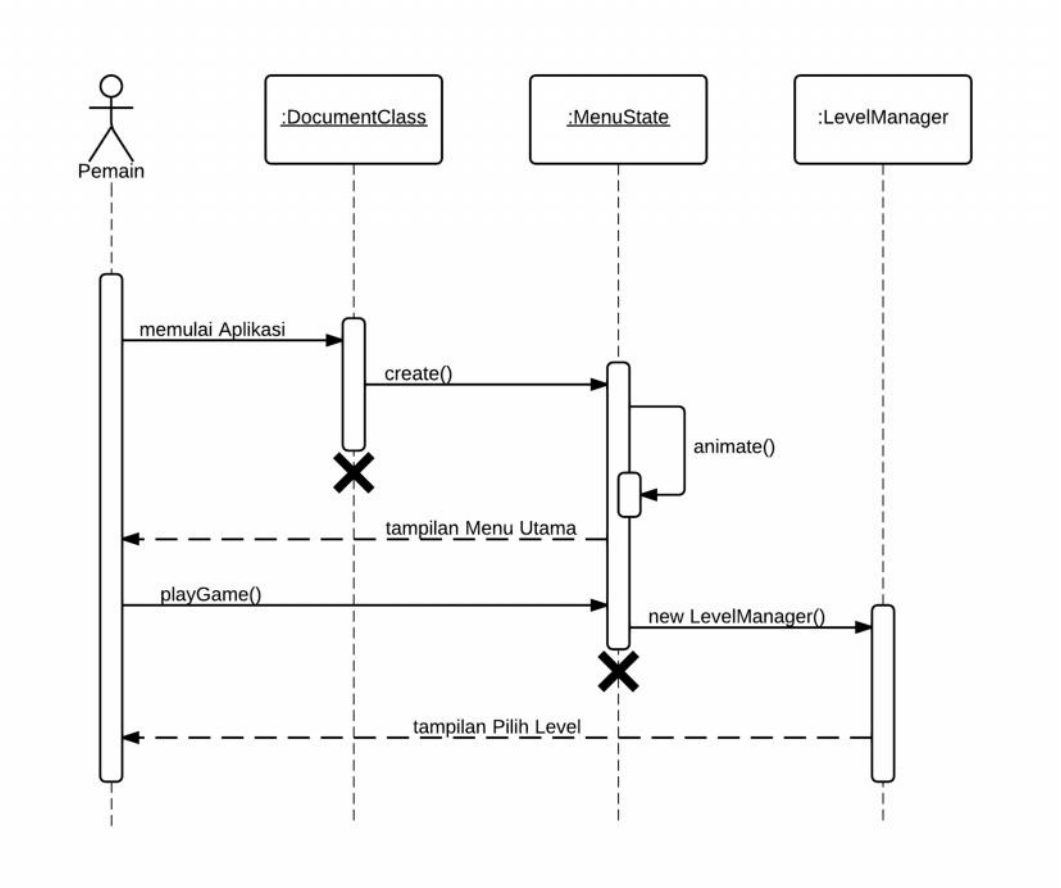
Gambar III.13 *Class Diagram*

3.1.7.4 Sequence Diagram

Sequence diagram menggambarkan tindakan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

1. Sequence Diagram Memainkan Full Game

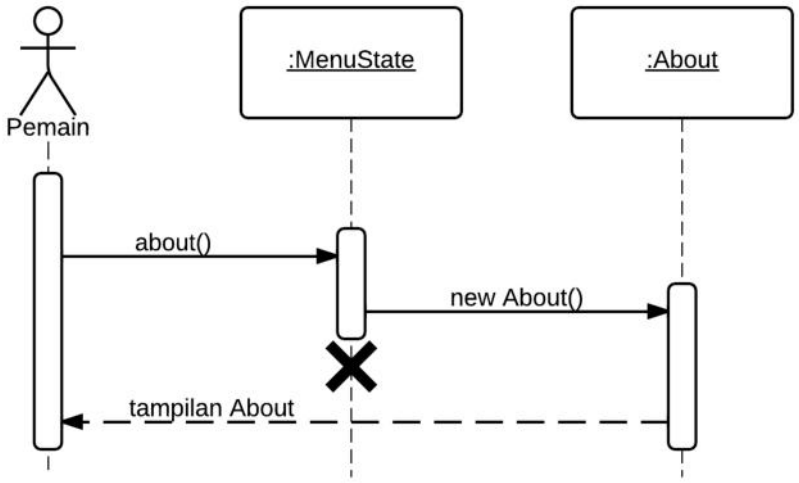
Diagram ini menjelaskan interaksi antar partisipan di dalam aktivitas Memainkan *Full Game* dalam urutan waktu. Untuk lebih jelasnya lihat gambar III.14.



Gambar III.14 *Sequence Diagram* Memainkan *Full Game*

2. Sequence Diagram Menampilkan Menu About

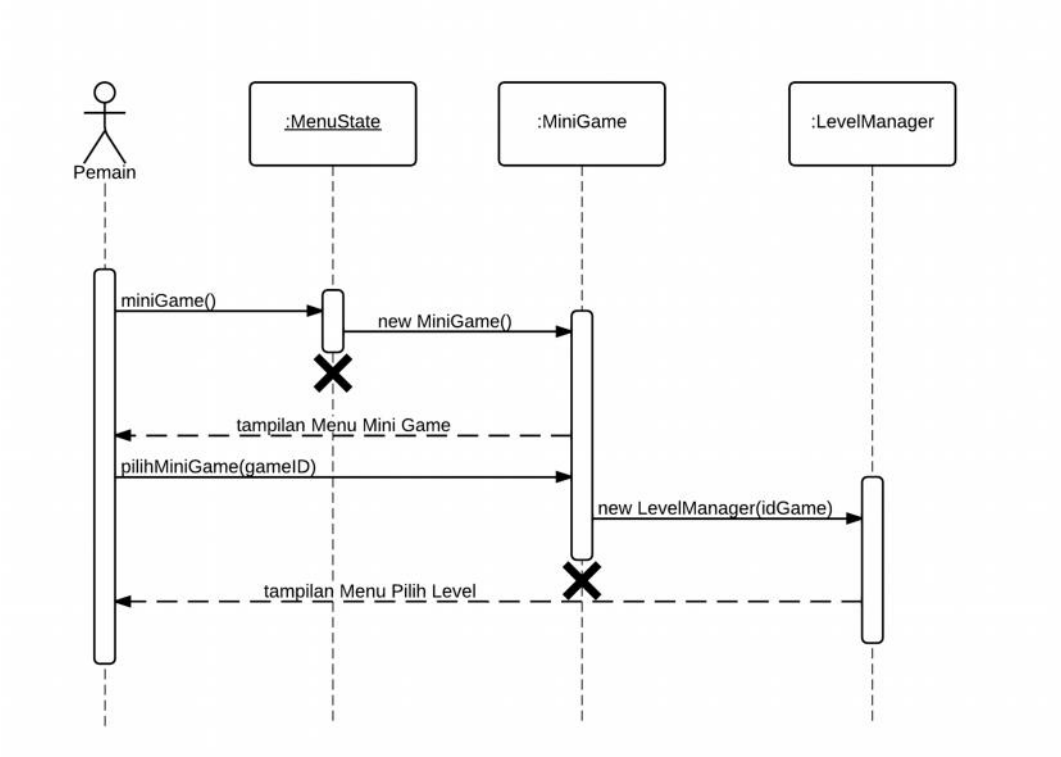
Diagram ini menjelaskan interaksi antar partisipan di dalam aktivitas Menampilkan Menu *About* dalam urutan waktu. Untuk lebih jelasnya lihat gambar III.15.



Gambar III.15 *Sequence Diagram* Menampilkan Menu *About*

3. Sequence Diagram Memainkan Mini Game

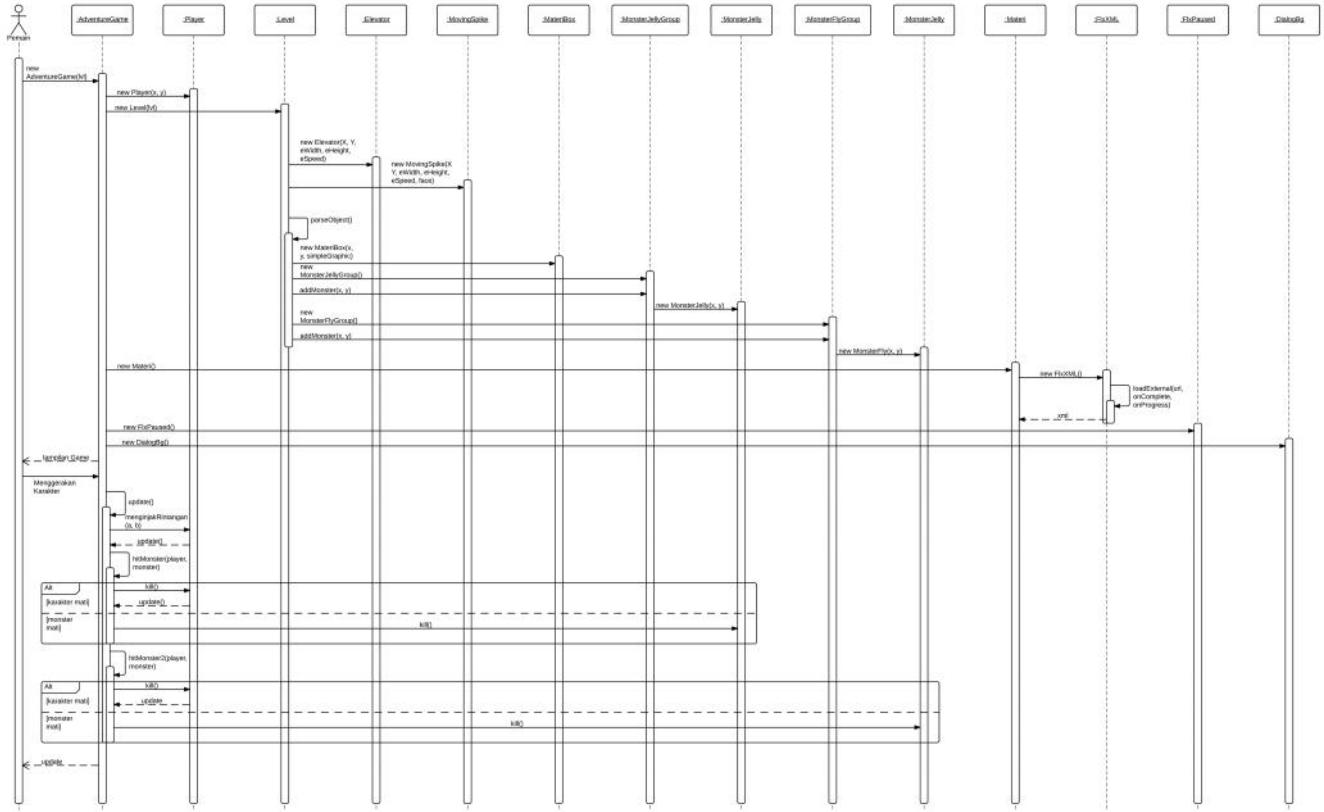
Diagram ini menjelaskan interaksi antar partisipan di dalam aktivitas Memainkan *Mini Game* dalam urutan waktu. Untuk lebih jelasnya lihat gambar III.16.



Gambar III.16 *Sequence Diagram* Memainkan *Mini Game*

4. Sequence Diagram Memainkan Adventure Game

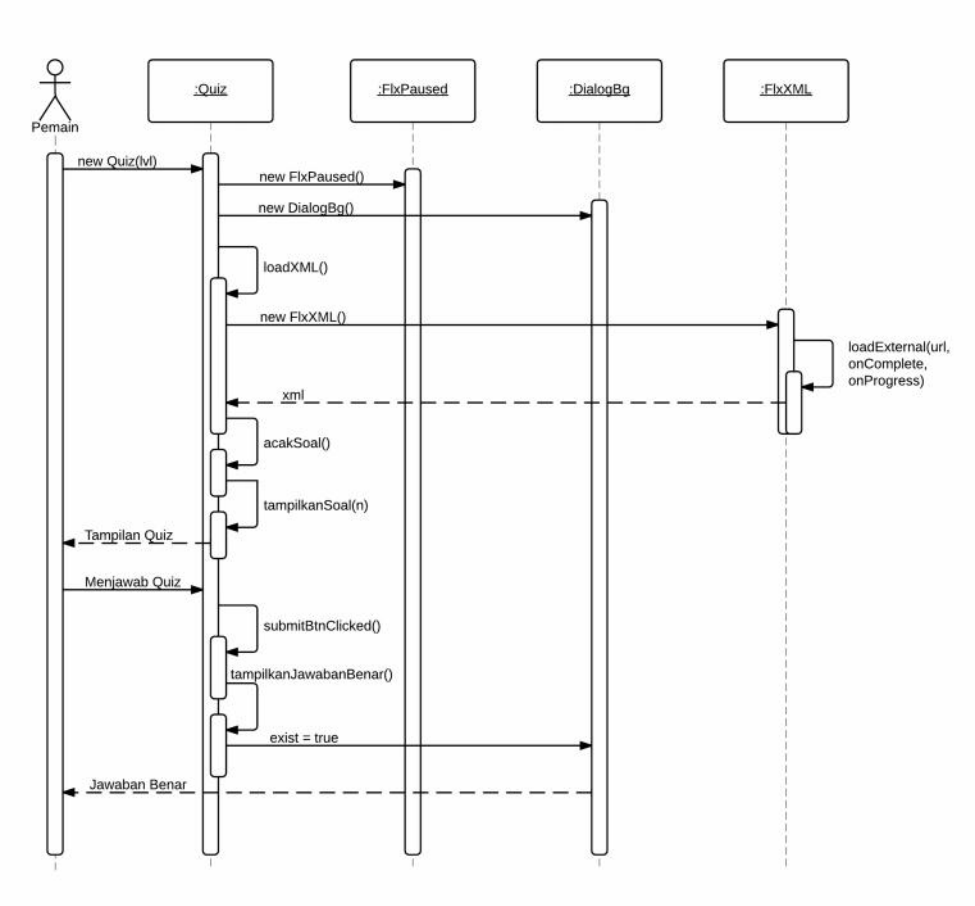
Diagram ini menjelaskan interaksi antar partisipan di dalam aktivitas Memainkan *Adventure Game* dalam urutan waktu. Untuk lebih jelasnya lihat gambar III.17.



Gambar III.17 *Sequence Diagram* Memainkan *Adventure Game*

5. Sequence Diagram Menjawab Quiz

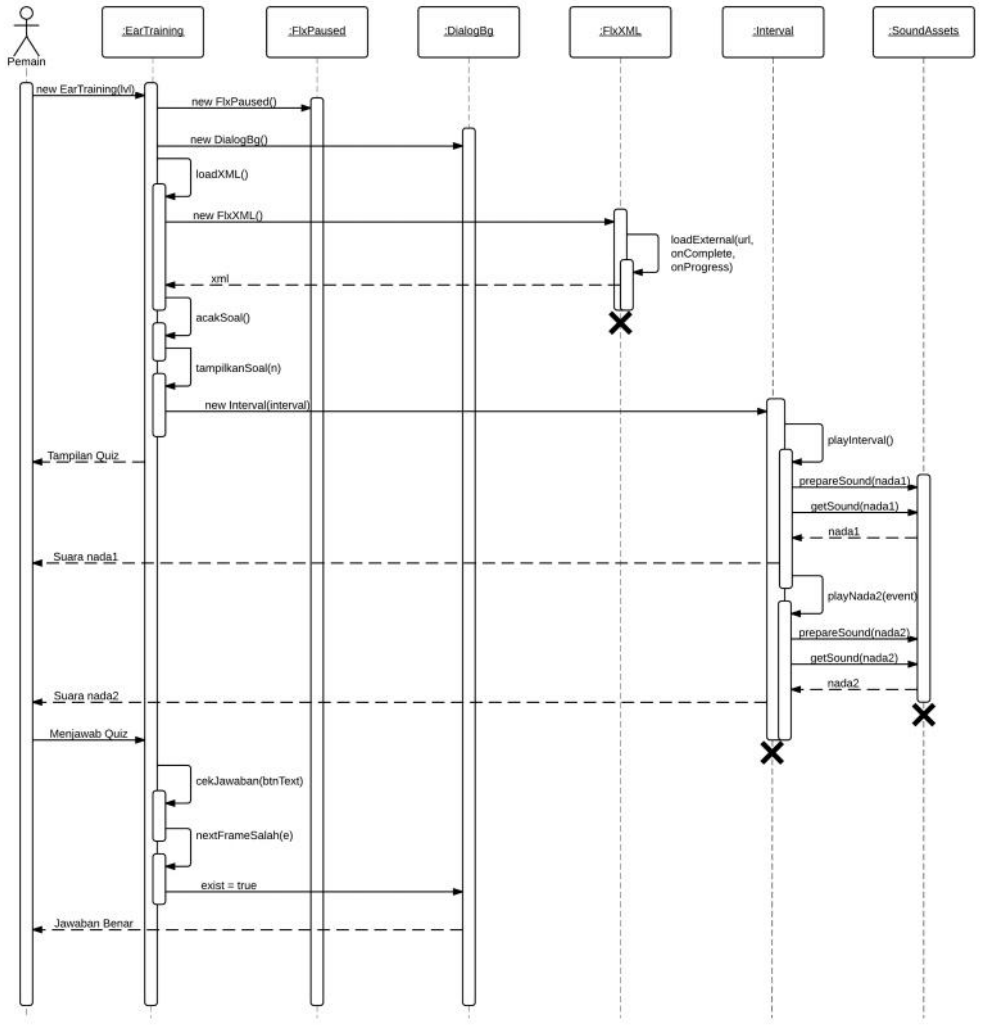
Diagram ini menjelaskan interaksi antar partisipan di dalam aktivitas Menjawab Quiz dalam urutan waktu. Untuk lebih jelasnya lihat gambar III.18.



Gambar III.18 *Sequence Diagram* Menjawab Quiz

6. Sequence Diagram Menjawab Ear Training

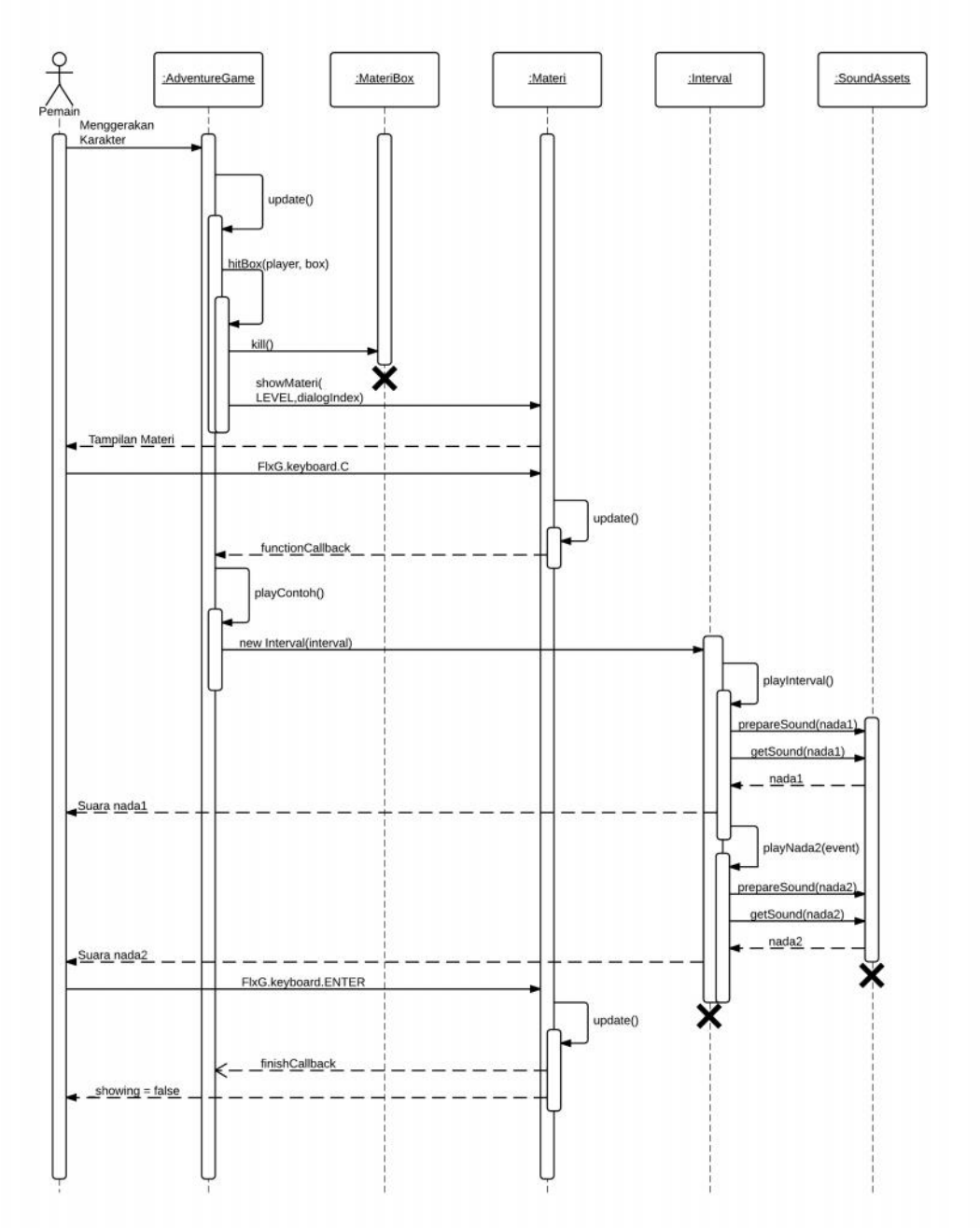
Diagram ini menjelaskan interaksi antar partisipan di dalam aktivitas Memainkan *Full Game* dalam urutan waktu. Untuk lebih jelasnya lihat gambar III.19.



Gambar III.19 *Sequence Diagram* Menjawab *Ear Training*

7. *Sequence Diagram* Menampilkan Materi

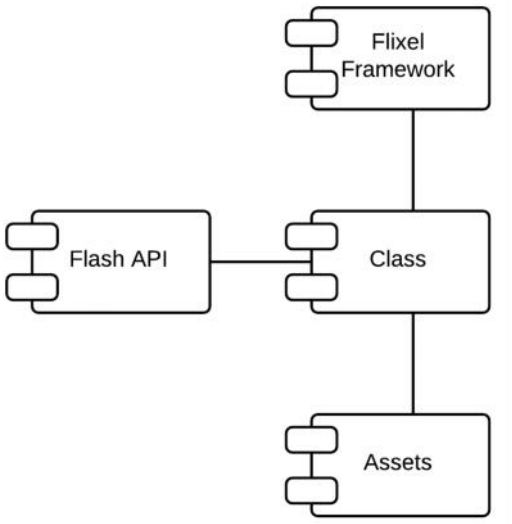
Diagram ini menjelaskan interaksi antar partisipan di dalam aktivitas Menampilkan Materi dalam urutan waktu. Untuk lebih jelasnya lihat gambar III.20.



Gambar III.20 *Sequence Diagram* Menampilkan Materi

3.1.7.5 Component Diagram

Component Diagram menggambarkan struktur dan hubungan antar komponen piranti lunak termasuk ketergantungan (*dependency*). Komponen piranti lunak yang yang dimaksud adalah modul yang berisi *source code* atau *binary code*, yang ada *library* atau *excutetable* yang muncul pada *compile time*, *link time* ataupun pada *runtime*. *Component Diagram* pada game ini bisa dilihat pada gambar III.21 berikut :



Gambar III.21 *Component Diagram*

3.2 Perancangan Sistem

Di dalam bagian ini akan dijelaskan perancangan komponen permainan, perancangan antarmuka, perancangan pesan dan jaringan semantik dari game edukasi *adventure* teori musik.

3.2.1 Perancangan Komponen Permainan

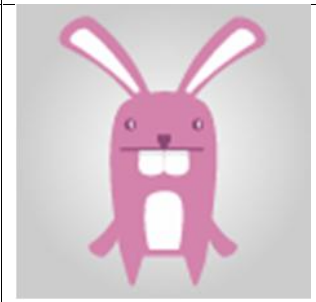
Di dalam bagian ini akan dijelaskan karakter yang terdapat pada game dan *storyboard* game edukasi *adventure* teori musik.

3.2.1.1 Karakter

Di dalam bagian ini akan dijelaskan karakter – karakter yang terdapat pada game ini seperti yang bisa dilihat pada tabel III.21.

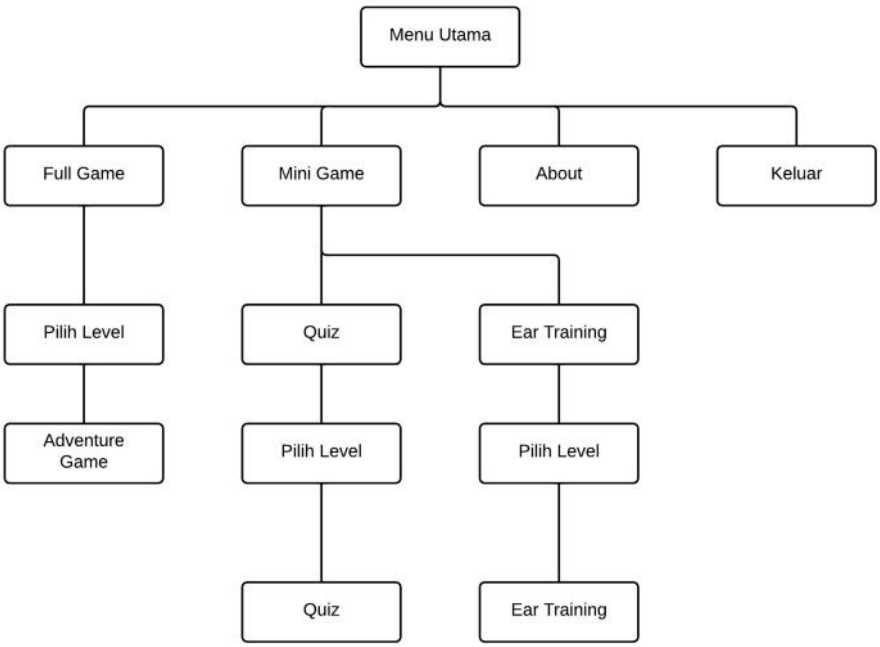
Tabel III.21 Karakter

No	Nama	Gambar	Keterangan
1	Pangeran Yong		1. Merupakan karakter utama yang akan dimainkan oleh pemain 2. Mati jika menabrak monster atau jebakan di adventure game 3. Bisa membunuh monster dengan menginjaknya.
2	Putri Hyun		1. Merupakan tunangan dari Pangeran Yong 2. Diculik oleh Gorgom 3. Tujuan utama pangeran melakukan perjalanan adalah untuk menyelamatkan putri.
3	Monster Gorgom		1. Merupakan musuh utama dari Pangeran Yong 2. Merupakan karakter yang menculik Putri Hyun
4	Monster Jelly		1. Anak buah Gorgom 2. Berada di Adventure Game 3. Pemain akan mati jika menabrak monster ini 4. Berpatroli di darat dan akan berbalik arah jika menemui tembok atau ujung platform 5. Cara membunuhnya adalah dengan diinjak

5	Monster Fly		1. Anak buah Gorgom 2. Berada di Adventure Game 3. Pemain akan mati jika menabrak monster ini 4. Terbang mengejar pemain jika pemain berada di jangkauannya 5. Cara membunuhnya adalah dengan diinjak
6	Monster Bunny		1. Anak buah Gorgom 2. Berada di Ear Training 3. Jika pemain memilih pintu yang salah, monster ini akan muncul

3.2.2 Perancangan Struktur Menu

Perancangan struktur menu pada game edukasi *adventure* teori musik pada gambar III.22 berikut :



Gambar III.22 Perancangan struktur menu

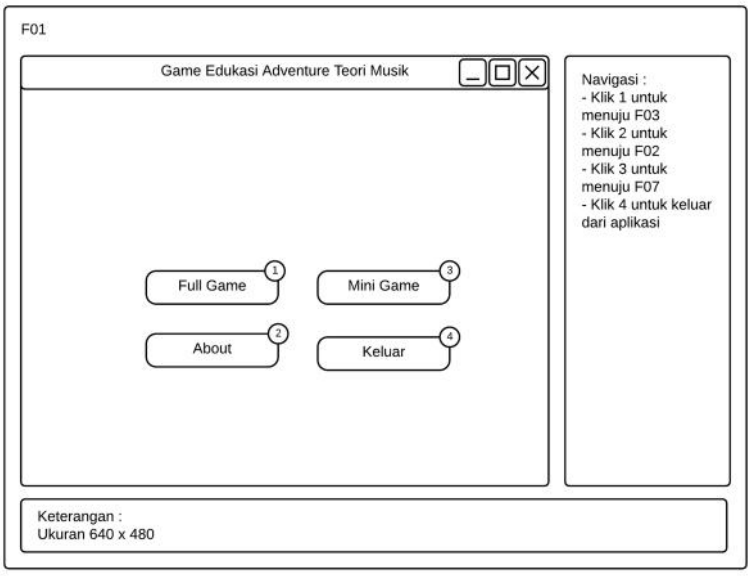
3.2.3 Perancangan Antarmuka

Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan mempermudah dalam mengimplementasikan aplikasi serta akan memudahkan pembuatan aplikasi.

Pada bagian ini akan dijelaskan perancangan antarmuka menu utama, perancangan antarmuka menu *about*, perancangan antarmuka pilih level, perancangan antarmuka menu *mini game*, perancangan antarmuka *adventure* game, perancangan antarmuka quiz, dan perancangan antarmuka *ear training.*

3.2.3.1 Perancangan Antarmuka Menu Utama

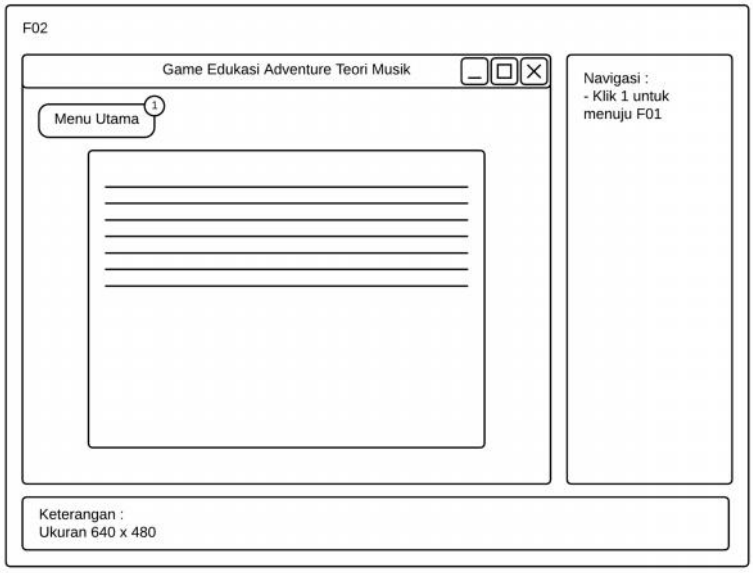
Menu utama merupakan menu yang akan muncul pertama kali pada saat pengguna membuka game. Gambar III.23 menunjukan perancangan antarmuka Menu Utama.



Gambar III.23 Perancangan Antarmuka Menu Utama

3.2.3.2 Perancangan Antarmuka *About*

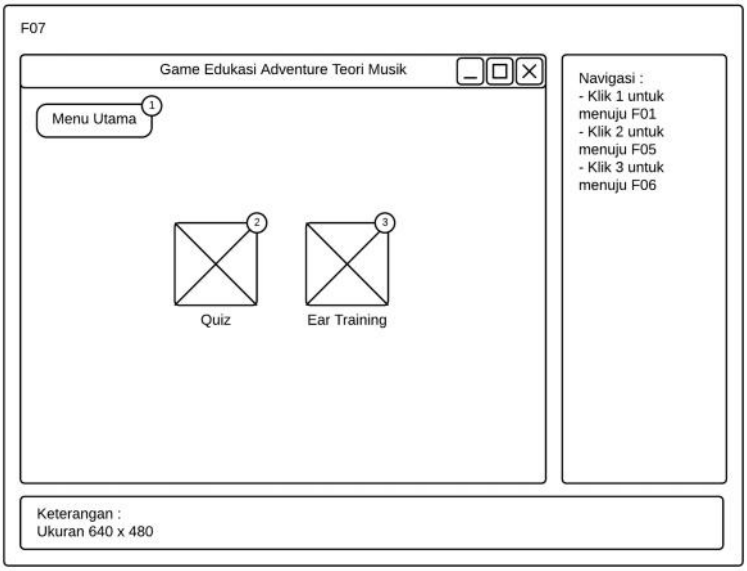
Menu *about* merupakan menu yang menampilkan profil penulis. Gambar III.24 menunjukan perancangan antarmuka Menu *About*.



Gambar III.24 Perancangan Antarmuka *About*

3.2.3.3 Perancangan Antarmuka Menu *Mini Game*

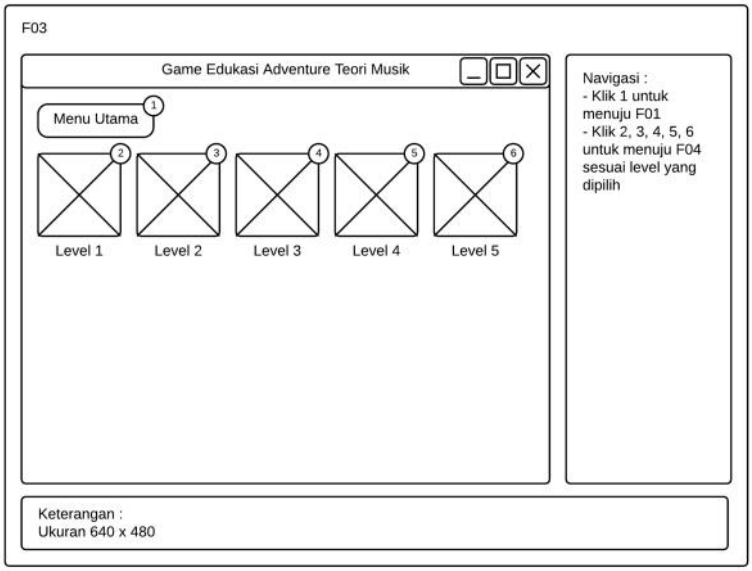
Menu *Mini Game* merupakan menu yang menampilkan pilihan *mini game* yang akan dimainkan pengguna. Seperti yang ditunjukkan pada gambar III.25.



Gambar III.25 Perancangan Antarmuka Menu Mini Game

3.2.3.4 Perancangan Antarmuka Pilih Level

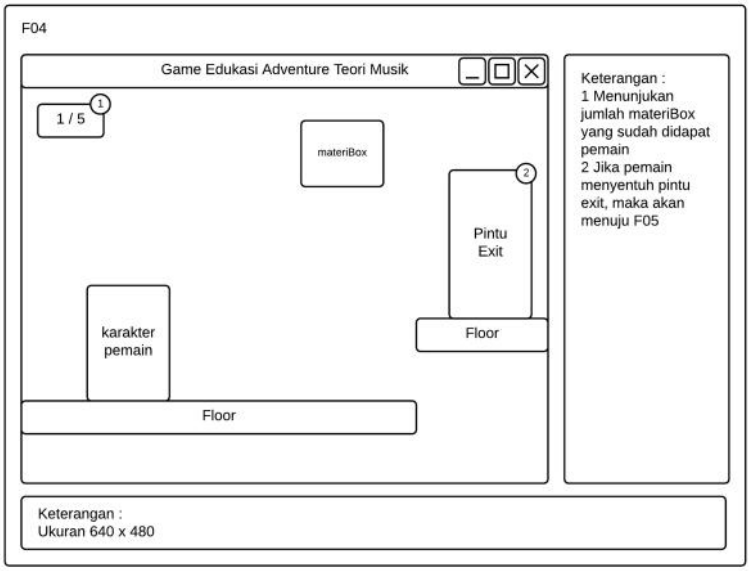
Pilih Level merupakan tampilan yang menampilkan pilihan level yang akan dimainkan pengguna. Seperti yang ditunjukkan pada gambar III.26.



Gambar III.26 Perancangan Antarmuka Pilih Level

3.2.3.5 Perancangan Antarmuka *Adventure Game*

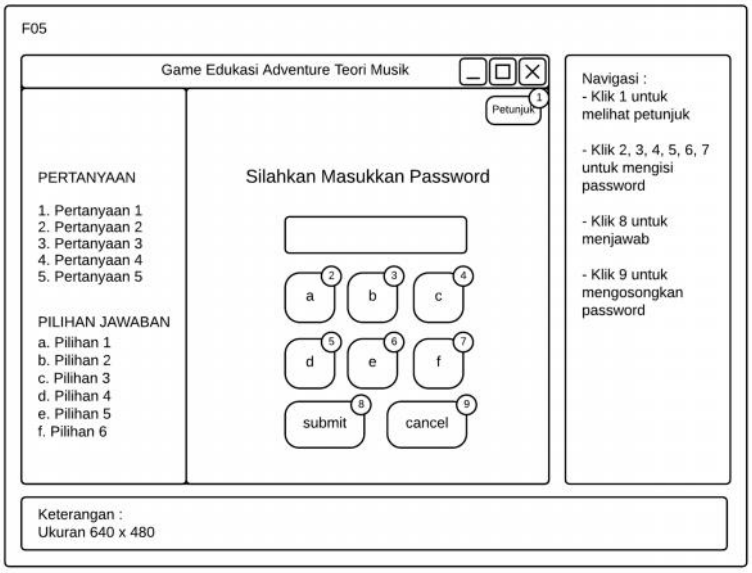
Antarmuka *Adventure Game* merupakan tampilan yang menampilkan *Adventure Game*. Seperti yang ditunjukkan pada gambar III.27.



Gambar III.27 Perancangan Antarmuka *Adventure Game*

3.2.3.6 Perancangan Antarmuka Quiz

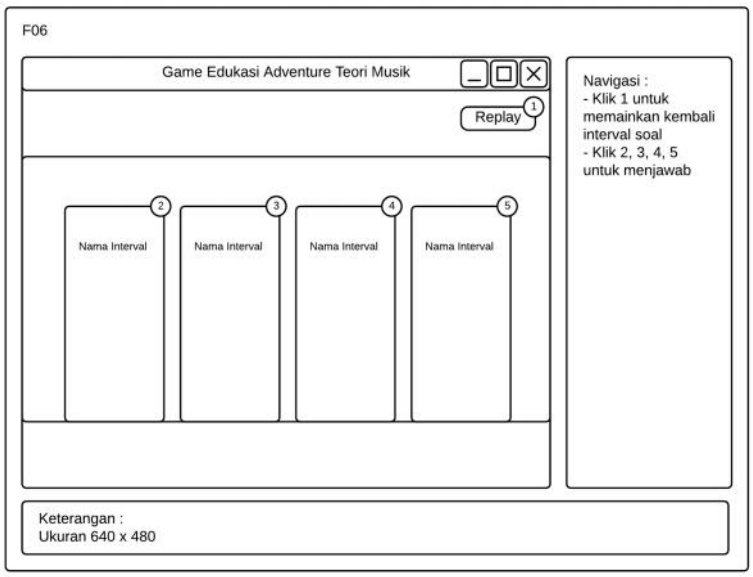
Antarmuka Quiz merupakan tampilan yang menampilkan *gameplay* quiz. Gambar III.28 menunjukan perancangan antarmuka Quiz.



Gambar III.28 Perancangan Antarmuka Quiz

3.2.3.7 Perancangan Antarmuka *Ear Training*

Antarmuka *Ear Training* merupakan tampilan yang menampilkan *gameplay ear training*. Seperti yang ditunjukkan pada gambar III.29.



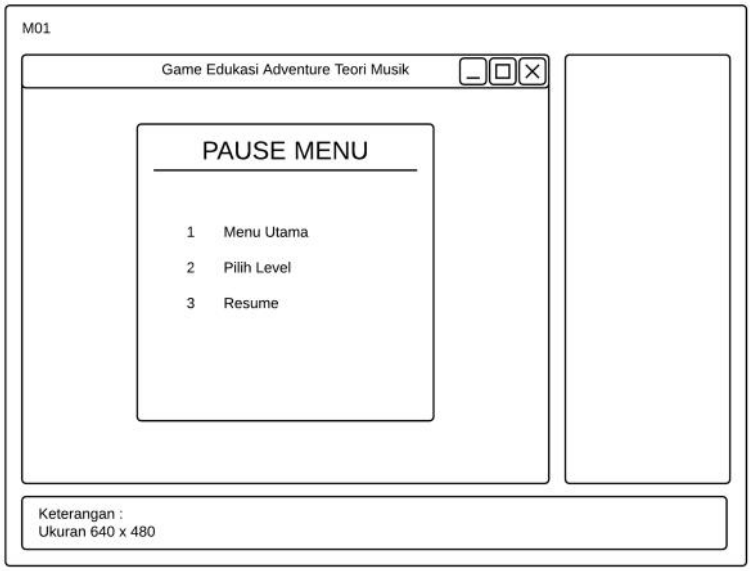
Gambar III.29 Perancangan Antarmuka *Ear Training*

3.2.4 Perancangan Pesan

Pada bagian ini akan dijelaskan perancangan pesan yang terdapat pada game edukasi *adventure* teori musik yaitu perancangan pesan *pause menu*, perancangan pesan materi, perancangan pesan jawaban quiz benar, perancangan pesan jawaban quiz salah, perancangan pesan *ear training score*, dan perancangan pesan level selesai.

3.2.4.1 Perancangan Pesan *Pause Menu*

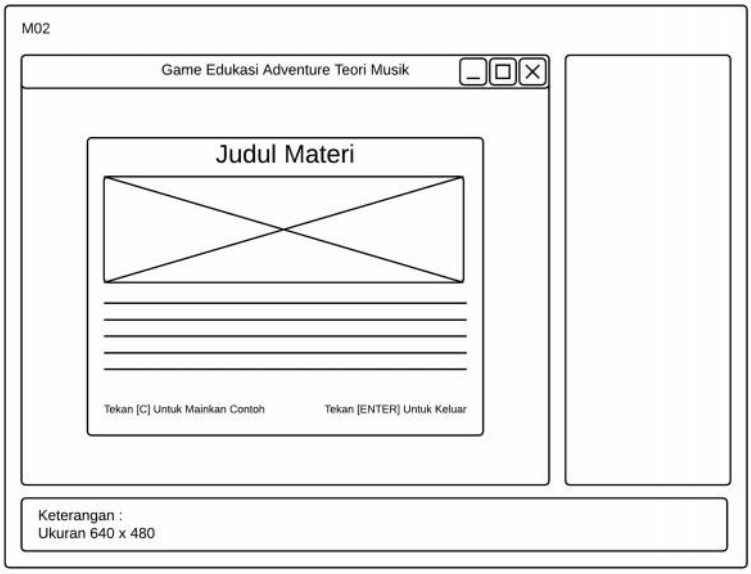
Pesan *Pause Menu* merupakan pesan yang muncul ketika pengguna menampilkan *pause menu.* Untuk lebih jelasnya dapat dilihat pada gambar III.30.



Gambar III.30 Perancangan Pesan *Pause Menu*

3.2.4.2 Perancangan Pesan Materi

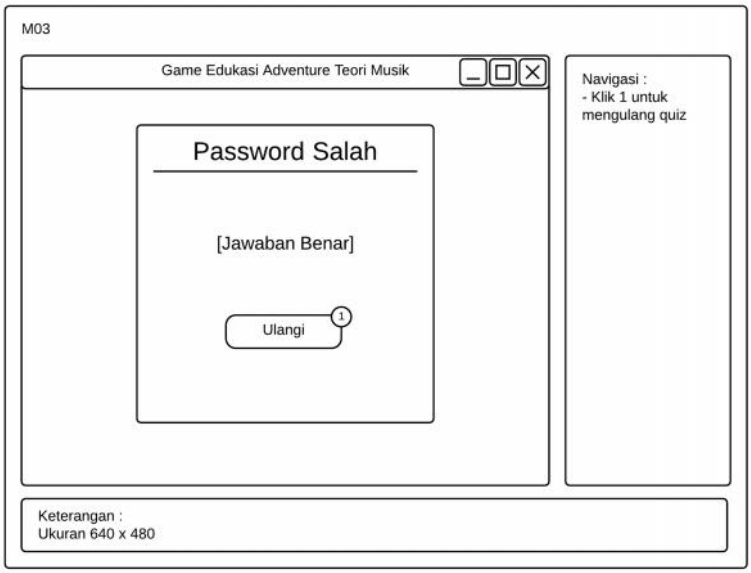
Pesan Materi merupakan pesan yang berisi materi pembelajaran yang muncul ketika pengguna mendapatkan materiBox. Untuk lebih jelasnya dapat dilihat pada gambar III.31.



Gambar III.31 Perancangan Pesan Materi

3.2.4.3 Perancangan Pesan Jawaban Quiz Salah

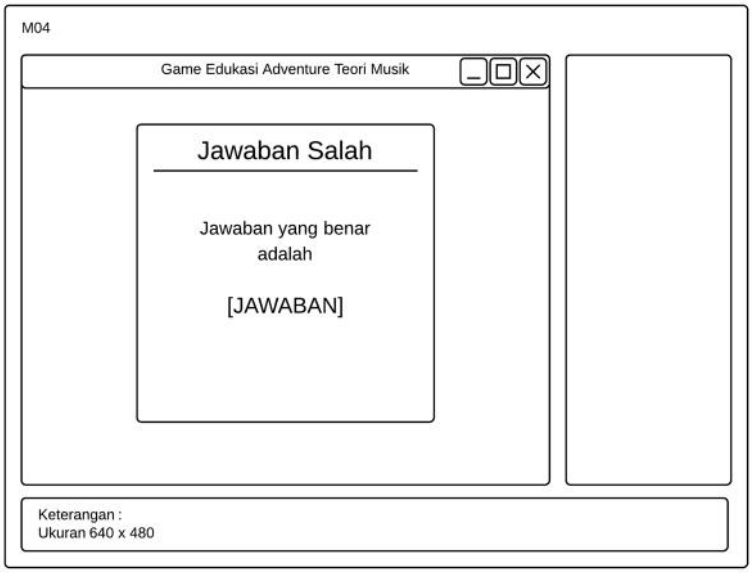
Pesan Jawaban Quiz Salah merupakan pesan yang muncul ketika pengguna menjawab pertanyaan quiz dengan jawaban yang salah. Untuk lebih jelasnya dapat dilihat pada gambar III.32.



Gambar III.32 Perancangan Pesan Jawaban Quiz Salah

3.2.4.4 Perancangan Pesan *Ear Training* Salah

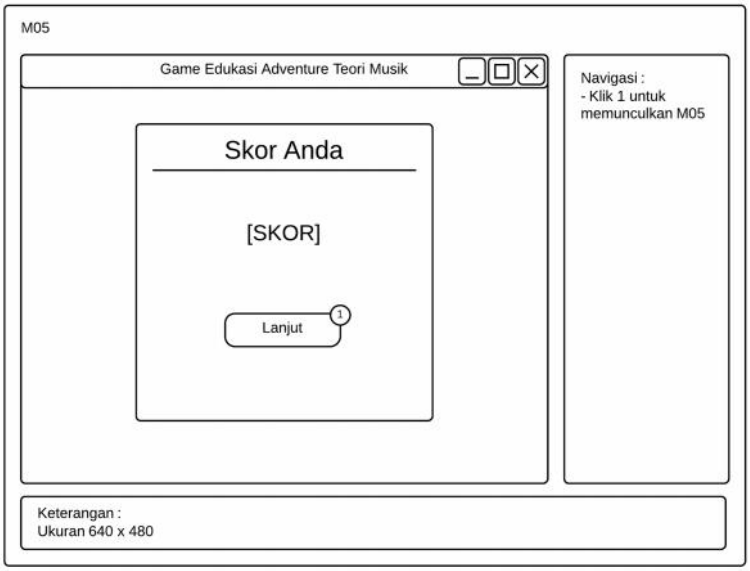
Pesan *Ear Training* Salah merupakan pesan yang muncul ketika pengguna menjawab pertanyaan *ear training* dengan jawaban yang salah. Untuk lebih jelasnya dapat dilihat pada gambar III.33.



Gambar III.33 Perancangan Pesan *Ear Training* Salah

3.2.4.5 Perancangan Pesan Ear Training Score

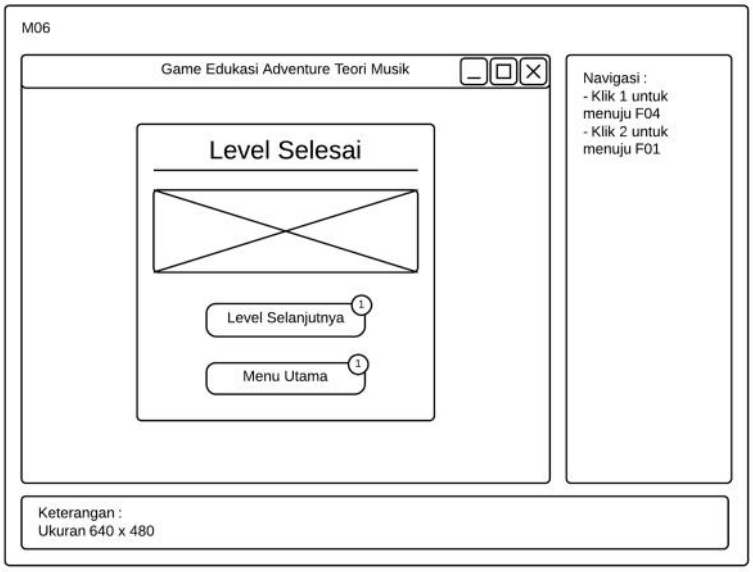
Pesan *Ear Training Score* merupakan pesan yang muncul ketika pengguna selesai menjawab semua soal *ear training*. Pesan ini berisi skor hasil dari *ear training test* pengguna. Untuk lebih jelasnya dapat dilihat pada gambar III.34.



Gambar III.34 Perancangan Pesan *Ear Training Score*

3.2.4.6 Perancangan Pesan Level Selesai

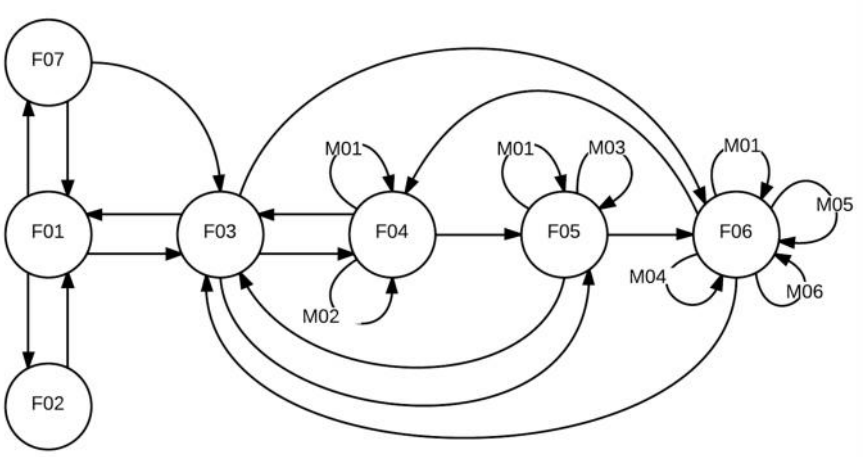
Pesan Level Selesai merupakan pesan yang muncul ketika pengguna selesai memainkan suatu level dalam game. Pesan ini berisi tombol navigasi untuk pengguna ke level selanjutnya atau kembali ke menu utama. Untuk lebih jelasnya dapat dilihat pada gambar III.35.



Gambar III.35 Perancangan Pesan Level Selesai

3.2.5 Jaringan Semantik

Untuk mengetahui keterhubungan dari setiap tampilan yang dibuat dapat menggunakan jaringan semantik. Gambar III.36 berikut adalah jaringan semantik pada game edukasi *adventure* teori musik.



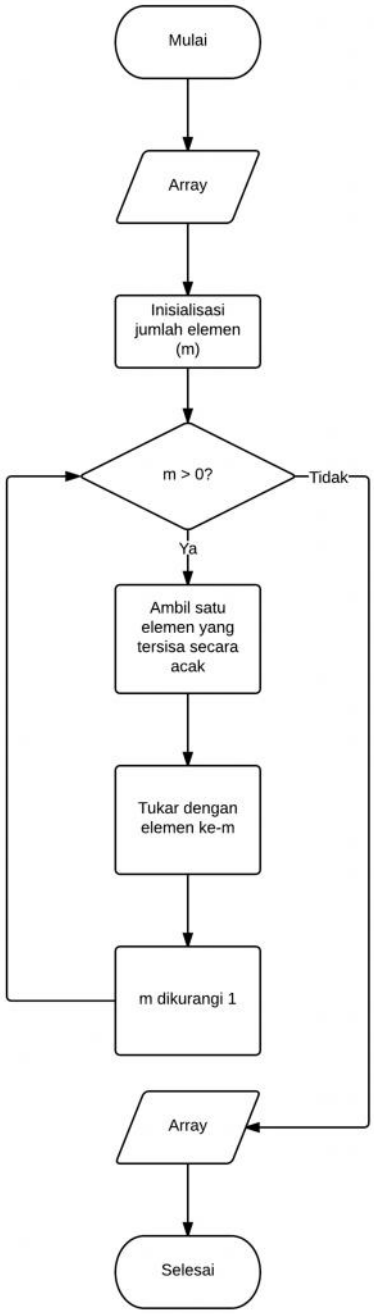
Gambar III.36 Jaringan Semantik

3.2.6 Perancangan Metode

Perancangan metode merupakan perancangan yang dilakukan untuk menetapkan detail algoritma yang akan dinyatakan ke dalam suatu program. Perancangan metode menjelaskan tentang bagaimana program aplikasi berjalan. Perancangan metode pada game edukasi *adventure* teori musik terdiri dari *flowchart* Fisher-Yates Shuffle, *flowchart* pembuatan password yang benar dari quiz, *flowchart behavior* monster jelly, *flowchart behavior* monster fly, *flowchart* hitMonster, serta *flowchart behavior* elevator dan movingSpike.

3.2.6.1 Flowchart Fisher-Yates Shuffle

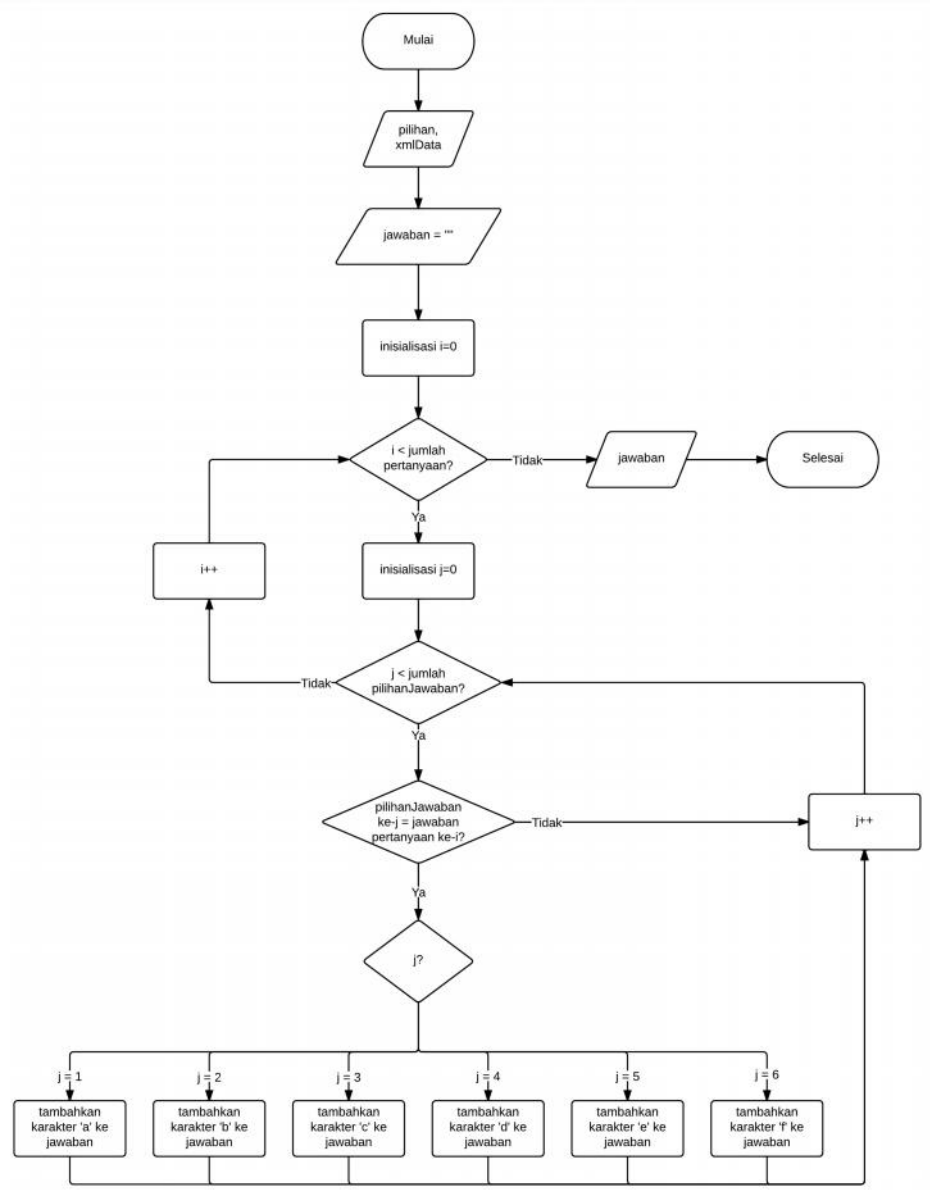
Flowchart Fisher-Yates Shuffle menggambarkan proses pengacakan array yang digunakan untuk mengacak soal, serta posisi pilihan jawaban pada quiz dan *ear training.* Untuk lebih jelasnya dapat dilihat pada gambar III.37.



Gambar III.37 *Flowchart* Fisher-Yates Shuffle

3.2.6.2 *Flowchart* pembuatan *password* yang benar dari quiz

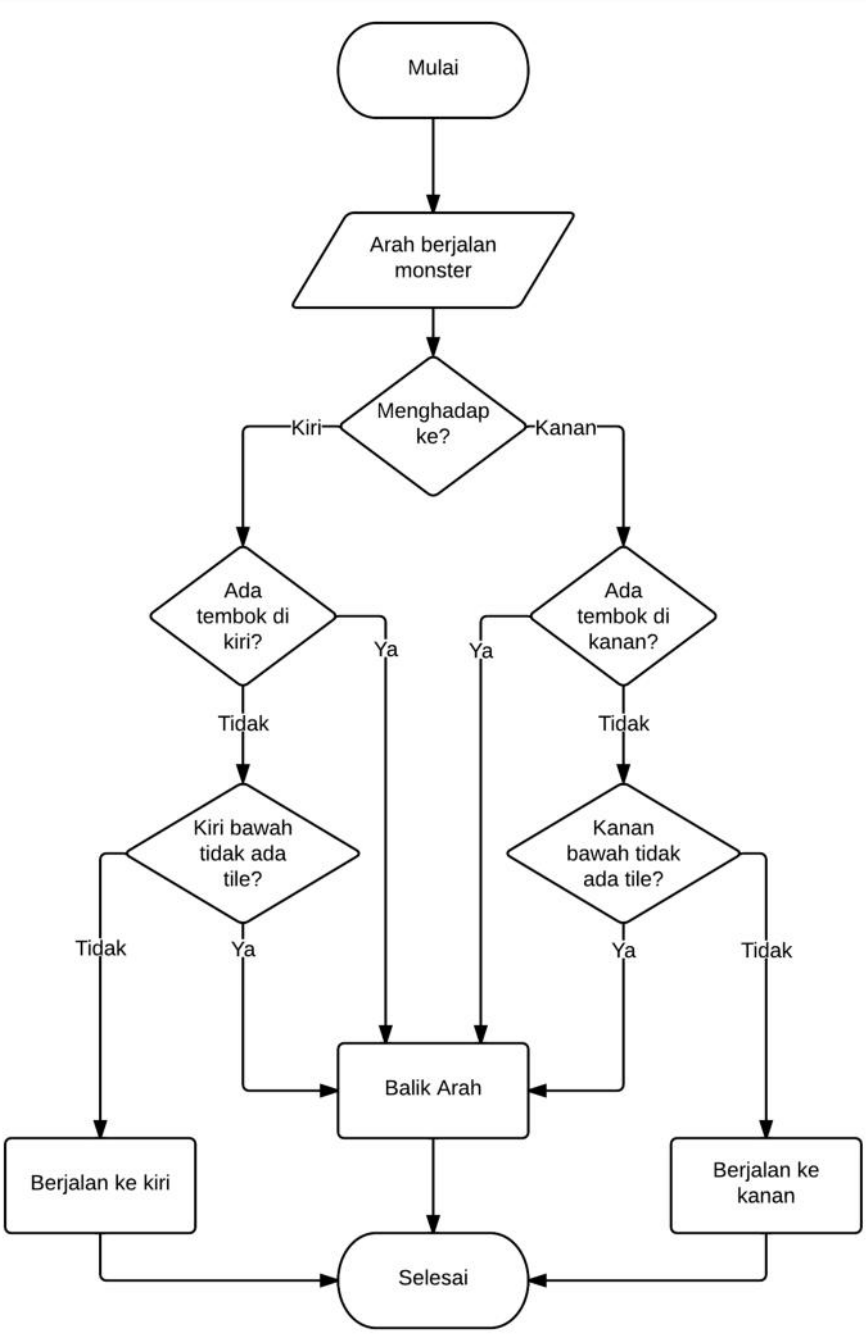
Flowchart ini menggambarkan proses pembuatan *password* yang benar pada *gameplay* quiz. Karena pertanyaan diacak dan posisi pilihan jawaban juga diacak, maka dibutuhkan algoritma pencarian untuk membuat kunci jawaban. Algoritma pencarian yang digunakan adalah *sequential search*. Untuk lebih jelasnya dapat dilihat pada gambar III.38.



Gambar III.38 *Flowchart* Proses Pembuatan *Password* Benar Dari Quiz

3.2.6.3 Flowchart behavior monster jelly

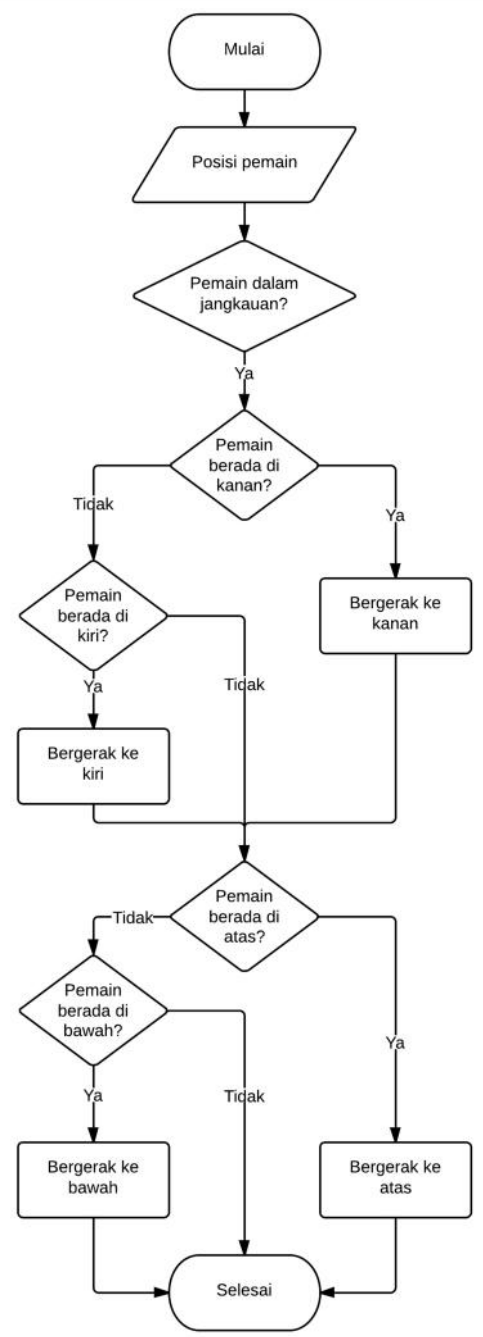
Flowchart ini menggambarkan *behavior* dari monster jelly. Monster ini berpatroli pada *platform* dan akan berbalik arah jika menemui tembok atau menemui ujung *platform*. Untuk lebih jelasnya dapat dilihat pada gambar III.39.



Gambar III.39 *Flowchart Behavior* Monster Jelly

3.2.6.4 Flowchart behavior monster fly

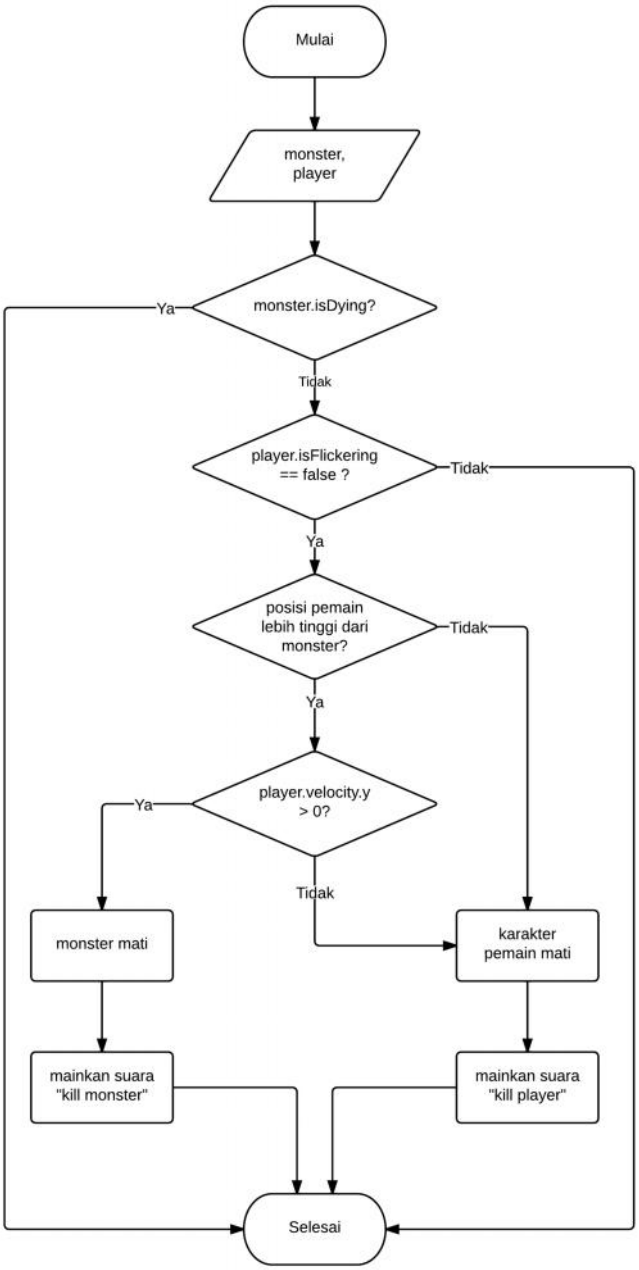
Flowchart ini menggambarkan *behavior* dari monster fly. Monster ini akan mengejar pemain jika pemain berada dalam jangkauan monster. Untuk lebih jelasnya dapat dilihat pada gambar III.40.



Gambar III.40 *Flowchart Behavior* Monster Fly

3.2.6.5 *Flowchart* hitMonster

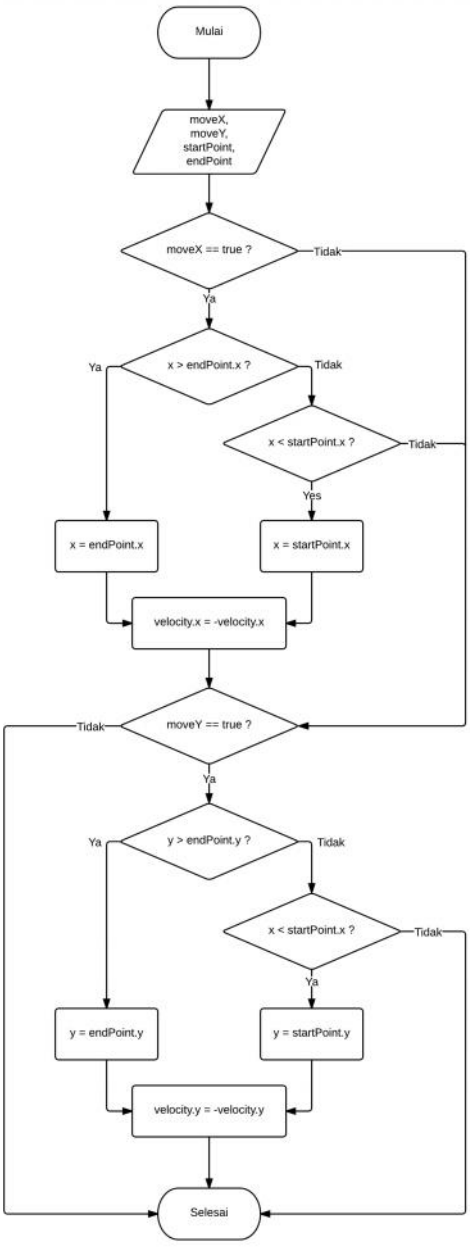
Flowchart ini menggambarkan *method* hitMonster, yaitu *method* yang terdapat dalam Class AdventureGame yang dipanggil jika terjadi *collision* antara objek Player dan Monster. Untuk lebih jelasnya dapat dilihat pada gambar III.41.



Gambar III.41 *Flowchart* hitMonster

3.2.6.6 *Flowchart behavior* elevator dan movingSpike

Flowchart ini menggambarkan *behavior* dari class elevator dan movingSpike. *Flowchart* ini menggambarkan isi *method update* yang terdapat pada kedua class tersebut. Untuk lebih jelasnya dapat dilihat pada gambar III.42.



Gambar III.42 *Flowchart behavior* elevator dan movingSpike