

BAB IV PERANCANGAN

Bab ini membahas mengenai perancangan permainan simulator angklung. Perancangan perangkat lunak dilakukan melalui dua tahap yaitu *game design* dan *technical design*.

4.1. *Game Design*

Bagian ini akan membahas perancangan *game* Simulator Angklung yang merupakan penjabaran lebih detail dari *ten page design document* yang telah dibuat pada bab III subbab 3.2.1.2. Berikut merupakan *game design document* dari game simulator Angklung.

4.1.1. *Cover*

Cover permainan Simulator angklung ditunjukkan pada Gambar 4.12.

4.1.2. *Game Title*

Permainan simulator angklung pada sistem operasi *android* ini bernama “Sarinande Angklung”.



Gambar 4.1. Cover game Simulator Angklung: Sarinande Angklung

Sumber: Perancangan

4.1.3. Game Overview

Game Sarinande Angklung adalah sebuah *game* simulator yang menyimulasikan alat musik angklung. *Game* ini memiliki tiga *gameplay* dimana dua *gameplay* memakai *accelerometer* dan satu *gameplay* memakai sentuhan layar pada *smartphone* sebagai inputan.

Gameplay pertama pada Sarinande Angklung disebut dengan *touch game*. Pemain akan memainkan sebuah *beat* tentang lagu daerah dengan menggunakan angklung sarinande yang berisikan delapan angklung. Pemain harus menyentuh angklung sesuai dengan *beat* yang turun dari atas layar. Ketika *beat* ditekan benar sesuai dengan nada angklungnya maka *score* akan bertambah dan jika salah maka akan keluar tulisan “*miss*”. Pemain dengan nilai tertinggi akan ditampilkan dalam *menu highscore*.

Gameplay kedua pada Sarinande Angklung disebut dengan *shake single player*. Pada mode ini, pemain akan memainkan angklung sesuai dengan nada yang dipilih, dengan cara menggerakkan *smartphone* searah sumbu x.

Gameplay ketiga disebut dengan *shake Multiplayer*, pemain akan memilih peran dalam *multiplayer* yaitu sebagai *server* atau *client*. Pemain yang berperan sebagai *server* akan menentukan *beat*, jenis nada yang dipilih, dan mengomando semua pemain yang berperan sebagai *client*. Sedangkan pemain yang berperan sebagai *client* hanya dapat menerima pesan atau perintah dari *server*. Dengan demikian semua pemain akan memainkan *beat* secara bersamaan. Pemain harus menggerakkan *smartphone* searah sumbu X ketika *beat* mengenai *Area Hit*. Ketika *beat* selesai, akan muncul tampilan yang berisikan jumlah *beat* yang telah dimainkan dengan benar.

4.1.4. Game Controls

Kontrol permainan Sarinande Angklung ini menggunakan *touch screen*, *accelerometer*, *keyboard*, *keypad*, dan *menu item* pada *smartphone*.

1. *Touch screen* merupakan kontrol utama pada Sarinande Angklung. *Touch screen* digunakan dalam memilih *menu* pada *game* dan menekan angklung Sarinande satu oktaf.

2. Kontrol *accelerometer* digunakan pada *gameplay* pada mode *shake*. Ketika *smartphone android* digerakkan searah sumbu x maka akan keluar suara nada angklung.
3. *Keyboard* digunakan untuk menginputkan teks pada form yang disediakan seperti memasukkan nama pada tampilan *highscore*, memasukkan *server IP* pada *multiPlayer*, memasukkan *title* dan *author* pada *menu composer*.
4. *Keypad* merupakan inputan yang berada pada hardware *smartphone* seperti *keypad volume*, *keypad back*, *keypad menu*, dan lain-lain. *Keypad* yang digunakan pada *Sarinande Angklung* adalah *keypad back* dan *keypad menu* yang digunakan untuk memunculkan kotak dialog atau menuju ke *scene* tertentu.
5. *Menu item* merupakan inputan pada *android* berupa *list* pilihan yang hanya boleh dipilih satu oleh pengguna. *Menu item* ini digunakan untuk memilih nada angklung yang berjumlah delapan nada.

4.1.5. *Gameplay Descriptions*

Gameplay pada *Sarinande Angklung* dibagi menjadi tiga. Berikut penjelasan secara rinci mengenai *gameplay* pada *Sarinande Angklung*.

1. *Touch Game*

Touch game merupakan sebuah *gameplay* yang dimainkan oleh satu pemain. Pada awal permainan, pemain akan disuguhkan tampilan untuk memilih *beat* yang akan dimainkan. Setelah pemain telah memilih *beat* dan *loading screen* selesai, pemain akan diperlihatkan delapan angklung dengan nada yang berbeda, *HUD score* pada bagian atas, area hit yang berupa garis lurus horizontal, mengeluarkan musik, serta akan muncul *beat* atau kotak-kotak yang turun dari atas layar. Ketika *beat* tersebut menyentuh area hit, maka pemain harus menekan angklung sesuai dengan nadanya. Jika pemain menekan angklung ketika *beat* berada pada area hit maka *score* bertambah 100 dan akan muncul tulisan “good”. Sedangkan jika *beat* telah tidak sama

dengan inputan angklung maka *score* dikurangi 50 keluar tulisan “*miss*”.

Pada akhir permainan akan keluar tampilan hasil *score* yang didapatkan.

2. *Shake Single Player*

Shake Single Player merupakan sebuah *gameplay* yang hanya dapat dimainkan oleh satu pemain. *Gameplay* ini menggunakan *accelerometer* Pada awal permainan, pemain akan diperlihatkan sebuah angklung dimana ketika *smartphone* digerakkan searah sumbu x maka gambar angklung akan bergetar, *smartphone* akan bergetar jika diaktifkan, dan mengeluarkan suara nada angklung. Pemain dapat memilih nada angklung dengan menekan *keypad menu* pada *smartphone*. Nada yang dapat dipilih terdapat delapan nada, yaitu do, re, mi, fa, sol, la, si, dan do'. Setelah nada dipilih, maka suara yang dihasilkan akan berubah sesuai dengan nada yang dipilih ketika *smartphone* digerakkan searah sumbu x.

3. *Shake Multiplayer*

Shake MultiPlayer merupakan sebuah *gameplay* yang dapat dimainkan maksimal sebanyak delapan orang. Dimana satu pemain berperan sebagai *server* dan tujuh pemain lainnya berperan sebagai *client*. *Gameplay* ini mengharuskan pemain agar terkoneksi pada jaringan lokal yang sama. Jika tidak terkoneksi pada jaringan local, maka pemain tidak dapat memainkan *multiPlayer*. Pada awal permainan pemain akan ditampilkan dua buah tombol yang digunakan untuk memilih peran. Terdapat dua peran yang ditawarkan yaitu sebagai *server* atau sebagai *client*.

Pemain berperan sebagai *server* kemudian akan muncul tampilan untuk memilih *beat* yang akan dimainkan. Setelah *beat* dipilih, akan keluar sebuah kotak dialog yang menampilkan ip *server*. Pada fase ini *server* akan menunggu semua *client* yang terkoneksi.

Pemain yang berperan sebagai *client* akan muncul sebuah kotak dialog yang berisikan sebuah *form* untuk mengisikan hostname dan ip *server*. Setelah terkoneksi ke *server*, maka pada bagian *server* akan keluar sebuah tulisan tentang *client* yang terkoneksi dan pada bagian *client* akan keluar

sebuah kotak dialog untuk *menunggu* pesan dari *server*. Fase ini akan berakhir ketika jumlah pemain yang terkoneksi di *server* sebanyak tujuh orang atau diberhentikan oleh *server*.

Setelah semua *client* terkoneksi oleh *server*, maka di bagian *server* akan muncul daftar *client* yang terkoneksi. Pada bagian ini, *server* akan memilih nada untuk setiap *client* yang terkoneksi. Kemudian nada dan *beat* yang dipilih akan dikirimkan ke *client*. Pada bagian *client* nada akan disimpan dalam sistem dan *beat* yang dikirimkan akan dicek. Apabila *beat* tersedia maka *client* melanjutkan permainan dan begitu sebaliknya, jika *beat* tidak tersedia maka *client* tidak dapat melanjutkan permainan. Setelah proses pemilihan *beat* dan nada selesai, pemain yang berperan sebagai *server* atau *client* dapat melanjutkan permainan dengan menekan tombol start play.

Setelah tombol *start* ditekan dan loading selesai, pada bagian *server* akan muncul sebuah kotak dialog yang berisikan teks “*start to play*” dan sebuah tombol. Sedangkan pada bagian *client* akan muncul kotak dialog yang bertuliskan “*wait*”. Pada saat pemain menekan tombol pada *server* maka kotak dialog wait pada semua *client* dan *server* menghilang, mengeluarkan musik, dan kotak *beat* akan turun dari atas layar. Ketika kotak *beat* menyentuh *area hit* maka pemain harus menggerakkan *smartphone* searah sumbu x. Jika pemain menggerakkan *smartphone* tepat ketika *beat* menyentuh area hit maka *score* bertambah dan muncul tulisan “*good*”. Jika tidak maka kotak *beat* bergerak ke bawah sampai kotak *beat* menghilang, ketika *smartphone* digerakkan ketika kotak *beat* berada di luar area hit maka *score* akan berkurang 25 poin. Di akhir permainan akan muncul kotak dialog untuk mengevaluasi berapa *beat* yang dimainkan dari jumlah total kotak *beat* yang harus dimainkan.

4.1.6. Support System

Support system merupakan fitur lain yang ditawarkan dalam Sarinande Angklung. Dalam permainan Sarinande Angklung terdapat lima support system, yaitu

1. *Composer*

Composer merupakan sebuah alat bantu yang digunakan untuk membuat *beat* yang kemudian akan dimainkan dalam fitur *Touch Game* dan *Shake Multiplayer*. Pengguna dapat mengomposisi sebuah *beat* dimana hasil dari komposisi ini disimpan dalam sebuah file dengan tipe *file .ANG* pada folder *Sarinande* di *SD Card*.

2. *Setting*

Setting merupakan sebuah alat bantu untuk melakukan pengaturan pada *game Sarinande Angklung*. Pengaturan pada *setting* berupa pengaturan volume musik dan *sfx*, serta pengaturan pengaktifan vibrasi.

3. *Highscore*

Highscore merupakan sebuah fitur untuk menampilkan nilai tertinggi dari setiap *beat* yang dimainkan pemain pada *touch game*.

4. *About*

About merupakan fitur pada *Sarinande Angklung* untuk menampilkan informasi mengenai angklung dan kredit dari pembuat *game Sarinande Angklung*.

5. *Record*

Record merupakan sebuah fitur untuk mengonversikan *beat* menjadi file audio dengan tipe *file AMR*. Hasil dari perekaman disimpan dalam folder *Sarinande* pada *SD Card*.

4.1.7. *Scoring*

Sistem *scoring* pada *game Sarinande Angklung* ini adalah dengan menghitung semua *beat* yang dimainkan dengan benar. *Beat* yang dimainkan dianggap benar jika pemain memasukkan inputan (menekan angklung pada *touch game* atau menggerakkan *smartphone* pada *shake multiplayer*) ketika kotak *beat* berada pada *area hit*. Jika inputan benar maka *score* akan ditambah 100 dan jika salah *score* akan dikurangi (50 poin pada *touch game* dan 25 poin pada *multiplayer*) Sistem *scoring* ini digunakan pada *gameplay touch game* dan *shake multiplayer*.

Pada *gameplay touch game* terdapat pengklasifikasian *score* berupa grade yang disimbolkan dalam huruf A, B, C, D, dan E. Sebelum dilakukan pengklasifikasian, jumlah *beat* yang benar dibagi dengan jumlah total *beat* yang dimainkan. Hasil dari perhitungan tersebut kemudian akan diklasifikasian sesuai dengan tabel berikut.

Tabel 4.1. Tabel pengklasifikasian *score* pada *touch game*

GRADE	RANGE
A	$X = 1$
B	$X \geq 0,75$ atau $X < 1$
C	$X \geq 0,5$ atau $X < 0,75$
D	$X \geq 0,25$ atau $X < 0,5$
E	$X < 0,25$

*X adalah hasil jumlah *beat* yang benar dibagi dengan jumlah total *beat* yang dimainkan

Sumber: Perancangan

4.1.8. Game Screens

Rancangan tampilan antar muka *game* Sarinande Angklung terdiri dari:

- **Menu Game**
 - **Main menu**
Main menu merupakan menu utama pada *game* Sarinande Angklung. Main menu merupakan gerbang penghubung fitur-fitur pada Sarinande Angklung. Prototipe dari main menu ditunjukkan pada Gambar 4.2.
 - **Pause menu**
Pause menu merupakan menu yang muncul ketika permainan dalam kondisi *pause*. Prototipe *pause menu* ditunjukkan pada Gambar 4.3.
 - **List menu**
List menu merupakan menu yang digunakan untuk menampilkan *file-file* pada *SD Card*. Pada *game* Sarinande Angklung list menu

digunakan untuk menampilkan *file* musik pada *composer* dan *file beat* pada *touch*. Prototipe *list menu* ditunjukkan pada Gambar 4.4.

- **Multiplayer menu**

MultiPlayer menu digunakan untuk memilih peran sebagai *server* atau sebagai *client*. Prototipe *multiplayer menu* ditunjukkan pada Gambar 4.5.

- **High Score menu**

Highscore menu digunakan untuk menampilkan semua nilai tertinggi pada *touch game*. Prototipe *high score menu* ditunjukkan pada Gambar 4.6.

- **Setting menu**

Setting menu digunakan untuk melakukan konfigurasi volume musik, *sfx*, dan konfigurasi vibrasi. Prototipe *setting menu* ditunjukkan pada Gambar 4.7.

- **About menu**

About menu digunakan untuk menampilkan informasi mengenai *angklung* dan kredit tentang *game*. Prototipe *about menu* ditunjukkan pada Gambar 4.8.

- **Record menu**

Record menu digunakan untuk memilih fitur pada *record*. Prototipe *about menu* ditunjukkan pada Gambar 4.9.

- **List client menu**

List client menu digunakan untuk menampilkan semua *client* yang terkoneksi ke *server*. Pengaturan pemilihan nada oleh *server* dilakukan pada *menu* ini. . Prototipe *list client menu* ditunjukkan pada Gambar 4.10.

- **How to play screen**

How to play screen menampilkan informasi bagaimana cara bermain *Sarinande Angklung*. . Prototipe *how to play screen* ditunjukkan pada Gambar 4.11.

- ***In game state***
 - ***“Touch game” in game state***

Prototipe *touch game* in *game state* ditunjukkan pada Gambar 4.12.
 - ***Shake Single Player in game state***

Prototipe *shake Single Player* ditunjukkan pada Gambar 4.13.
 - ***Shake multiplayer in game state***

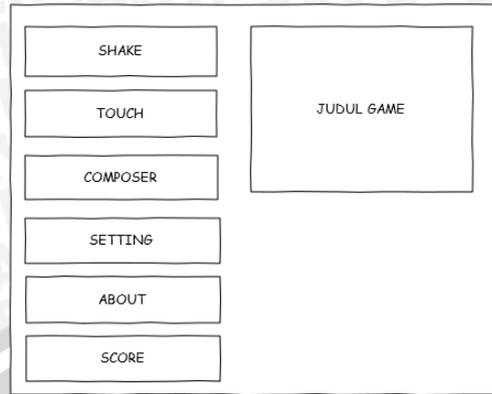
Prototipe *shake multiplayer* pada saat permainan ditunjukkan pada Gambar 4.14.
 - ***Composer in game state***

Prototipe *composer* ditunjukkan pada Gambar 4.15.
- ***Form Screen***
 - ***Composer form***

Composer form digunakan untuk memasukkan inputan judul dan pembuat *beat*. Prototipe *composer form* ditunjukkan pada Gambar 4.16.
 - ***High score form***

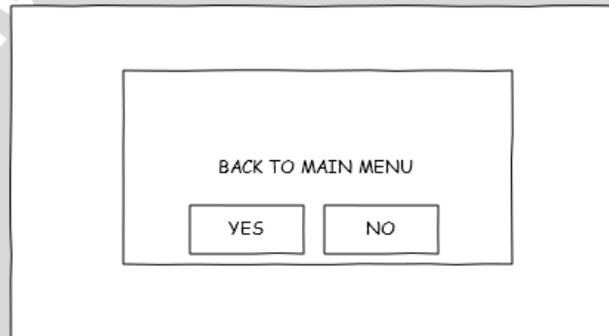
High score form digunakan untuk memasukkan inputan nama yang mendapatkan *score* tertinggi pada *touch game*. Prototipe *high score form* ditunjukkan pada Gambar 4.17.
 - ***Client form***

Client form digunakan untuk memasukkan inputan nama *hostname* dan *ip server* yang digunakan untuk mengoneksikan ke *server*. Prototipe *client form* ditunjukkan pada Gambar 4.18.



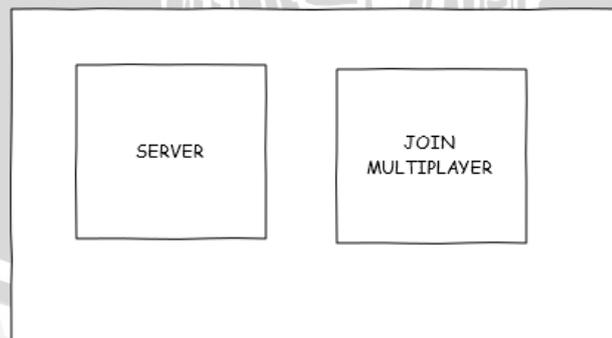
Gambar 4.2. Prototipe tampilan *main menu*

Sumber: Perancangan



Gambar 4.3. Prototipe tampilan *pause menu*

Sumber: Perancangan



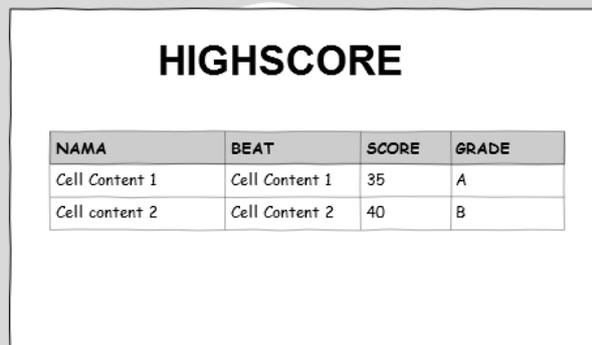
Gambar 4.4. Prototipe tampilan *multiplayer menu*

Sumber: Perancangan



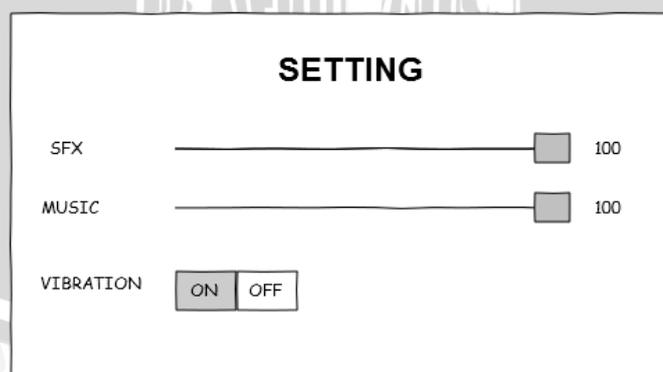
Gambar 4.5. Prototipe tampilan *list menu*

Sumber: Perancangan



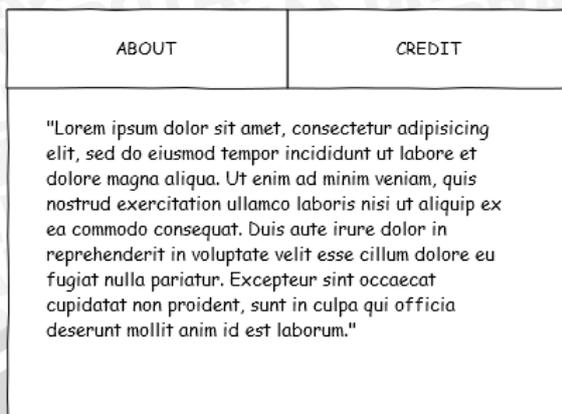
Gambar 4.6. Prototipe tampilan *high score menu*

Sumber: Perancangan



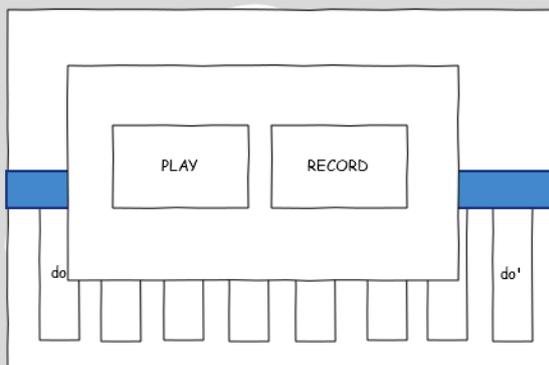
Gambar 4.7. Prototipe tampilan *setting menu*

Sumber: Perancangan



Gambar 4.8. Prototipe tampilan *about* menu

Sumber: Perancangan



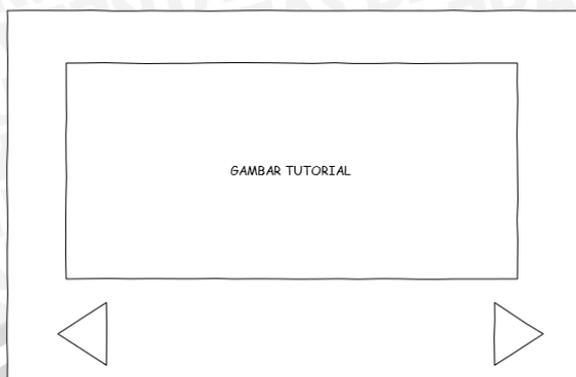
Gambar 4.9. Prototipe tampilan *record* menu

Sumber: Perancangan

MULTIPLAYER		
SERVER		
SERVER	127.0.0.1	do
CLIENT		
HOSTNAME	IP	NADA
CLIENT 1	127.0.0.1	RE
CLIENT 2	127.0.0.1	MI
CLIENT 3	127.0.0.1	FA
CLIENT 4	127.0.0.1	SOL
CLIENT 5	127.0.0.1	LA
CLIENT 6	127.0.0.1	SI
CLIENT 7	127.0.0.1	do'

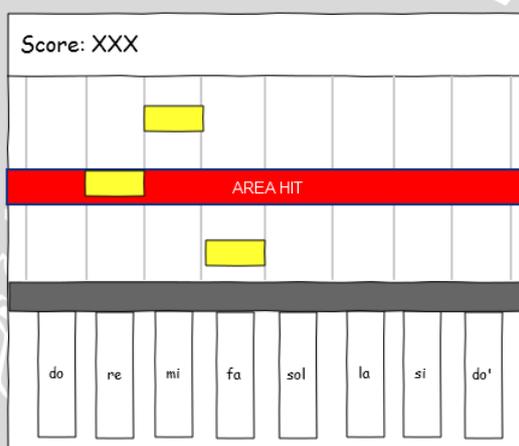
Gambar 4.10. Prototipe tampilan *list client* menu

Sumber: Perancangan



Gambar 4.11. Prototipe tampilan *how to play* screen

Sumber: Perancangan



Gambar 4.12. Prototipe tampilan *touch game* in game state

Sumber: Perancangan



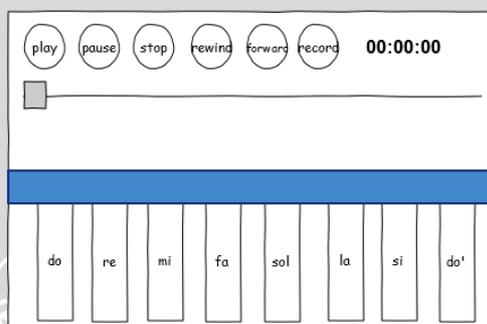
Gambar 4.13. Prototipe tampilan *shake Single Player* in game state

Sumber: Perancangan



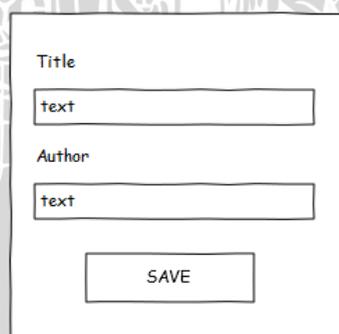
Gambar 4.14. Prototipe tampilan *shake multiplayer game state*

Sumber: Perancangan



Gambar 4.15. Prototipe tampilan *composer in game state*

Sumber: Perancangan



Gambar 4.16. Prototipe tampilan *composer form*

Sumber: Perancangan



RIGHT BEAT: XX
TOTAL BEAT: XX

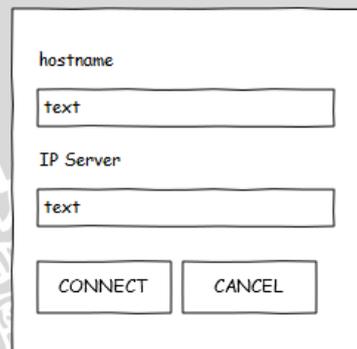
A

fill this field with your name

REPLAY SELECT SONG QUIT

Gambar 4.17. Prototipe tampilan *high score form*

Sumber: Perancangan



hostname

text

IP Server

text

CONNECT CANCEL

Gambar 4.18. Prototipe tampilan *client form*

Sumber: Perancangan

4.1.9. HUD systems

Penggunaan HUD pada *game* Sarinande Angklung terdapat *composer* dan *touch game*. Berikut penjelasan mengenai sistem HUD pada *game* Sarinande Angklung.

1. Touch game

Prototipe tampilan HUD *touch game* ditunjukkan pada Gambar 4.19.

Berikut merupakan keterangan pada Gambar 4.19.

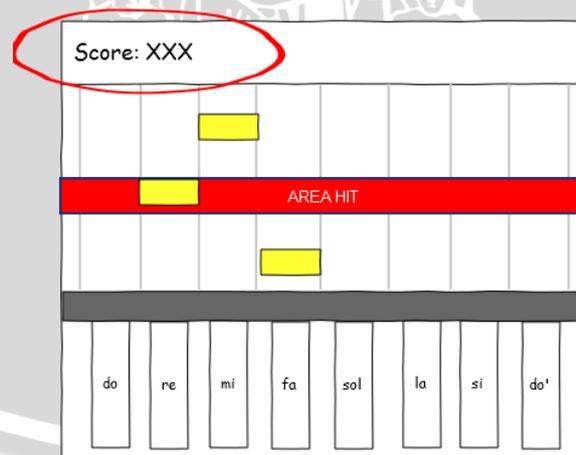
1) **Score**, panel *score* yang berisikan nilai yang didapat oleh pemain.

2. Composer

Prototipe tampilan HUD *composer* ditunjukkan pada Gambar 4.20.

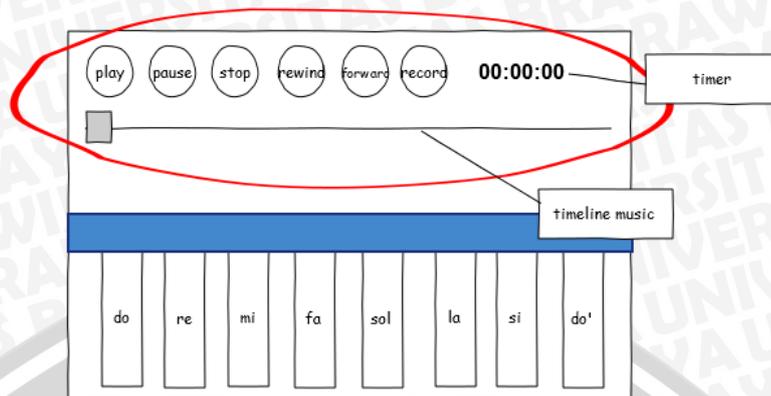
Berikut merupakan keterangan pada Gambar 4.20.

- 1) **Play**, merupakan tombol yang berfungsi untuk memainkan musik *background* pada saat mengomposisi *beat*.
- 2) **Pause**, merupakan tombol yang berfungsi untuk memberhentikan sementara musik *background*.
- 3) **Stop**, merupakan tombol yang berfungsi untuk memberhentikan musik *background*. Ketika music dimainkan, music akan bermain dari awal.
- 4) **Rewind**, merupakan tombol yang berfungsi untuk *menuju* ke waktu sebelumnya pada saat musik dimainkan.
- 5) **Forward**, merupakan tombol yang berfungsi untuk *menuju* ke waktu selanjutnya pada saat musik dimainkan.
- 6) **Record**, merupakan tanda apakah perekaman *beat* sedang dilakukan atau tidak.
- 7) **Timeline music**, merupakan tanda music sedang berjalan dan *menunjukkan* waktu musik dimainkan.
- 8) **Timer**, *menunjukkan* seberapa lama perekaman dilakukan. *Timer* hanya akan berjalan ketika perekaman dilakukan.



Gambar 4.19. Prototipe tampilan HUD *touch game*

Sumber: Perancangan

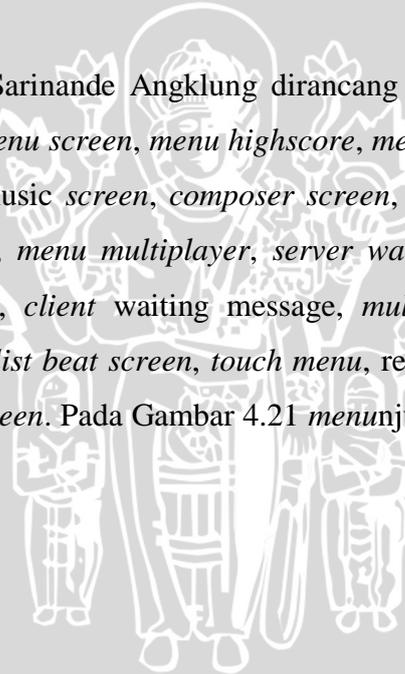


Gambar 4.20. Prototipe tampilan HUD pada *composer*

Sumber: Perancangan

4.1.10. Sitemap

Pada permainan Sarinande Angklung dirancang 23 *screen* yang terdiri dari *splash screen*, *main menu screen*, *menu highscore*, *menu about*, *menu setting*, *how to play screen*, *list music screen*, *composer screen*, *menu pause*, *composer form*, *shake single player*, *menu multiplayer*, *server waiting client screen*, *list client screen*, *client form*, *client waiting message*, *multiPlayer game screen*, *multiPlayer score screen*, *list beat screen*, *touch menu*, *record screen*, *toch game screen*, dan *game score screen*. Pada Gambar 4.21 menunjukkan *sitemap* dari game Sarinande Angklung.



nada akan digunakan pada *composer*, *touch game*, dan *record*. Prototipe elemen angklung satu nada ditunjukkan pada Tabel 4.3. dan prototipe elemen angklung delapan nada ditunjukkan pada table 4.4.

2. Kotak *beat*

Prototipe elemen kotak *beat* dirancang dua prototipe yaitu kotak *beat* untuk *shake multiPlayer* dan kotak *beat* untuk *touch game*. Prototipe elemen kotak *beat* untuk *touch game* ditunjukkan pada Tabel 4.5 dan kotak *beat* untuk *shake multiPlayer* ditunjukkan pada Tabel 4.6.

3. Area hit

Prototipe elemen *area hit* ditunjukkan pada table 4.7. Elemen *area hit* digunakan pada *shake multiPlayer* dan *touch game*. Ketika kotak *beat* menyentuh *area hit*, maka pemain harus menyentuh/menggoyangkan angklung sehingga pemain mendapatkan *score*.

4. Efek Teks

Efek teks merupakan sebuah elemen pada *game Sarinande Angklung* yang keluar ketika pemain memberikan inputan dimana kotak *beat* menyentuh *area hit*. Prototipe elemen efek teks dirancang dua yaitu “GOOD” dan “MISS”. Prototipe elemen “GOOD” ditunjukkan pada Tabel 4.8 dan prototipe elemen “MISS” ditunjukkan pada Tabel 4.9.

5. *Hand Cursor*

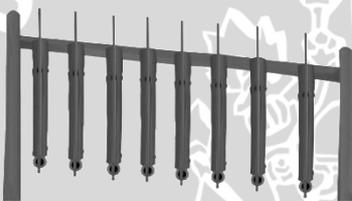
Prototipe *Hand Cursor* pemain ditunjukan oleh Tabel 4.10. Elemen ini yang akan menunjukkan angklung mana yang sedang dimainkan. Ketika angklung yang ditekan adalah angklung dengan nada do, maka *hand cursor* berpindah ke angklung do. *Hand cursor* digunakan bersamaan pada angklung delapan nada.

Tabel 4.2. Prototipe elemen angklung satu nada

Game Object	Profil
	<p>Nama : Angklung satu nada</p> <p>Keterangan : Angklung satu nada digunakan pada <i>shake Single Player</i>. Ketika <i>smartphone</i> digerakkan searah sumbu x, maka elemen ini akan memunculkan efek getar</p>

Sumber: Perancangan

Tabel 4.3. Prototipe elemen angklung delapan nada

Game Object	Profil
	<p>Nama : Angklung delapan nada</p> <p>Keterangan : Angklung yang terdiri dari delapan nada yaitu do, re, mi, fa, sol, la, si, dan do'. Elemen ini digunakan pada bagian <i>composer, touch game, dan record</i>.</p>

Sumber: Perancangan

Tabel 4.4. Prototipe elemen kotak *beat* pada *touch game*

Game Object	Profil
	<p>Nama : kotak <i>touch game beat</i></p> <p>Keterangan : Elemen ini akan mewakili setiap nada yang harus ditekan oleh pemain. Kotak <i>beat</i> akan keluar dari atas layar dan bergerak ke bawah layar dimana posisi koordinat x kotak <i>beat</i> berada persis di atas masing-masing angklung. Elemen ini digunakan pada <i>touch game</i></p>

Sumber: Perancangan

Tabel 4.5. Prototipe elemen kotak *beat* pada *shake multiplayer*

<i>Game Object</i>	Profil
	Nama : <i>multiplayer beat</i>
	Keterangan : Elemen ini akan mewakili setiap nada yang harus ditekan oleh pemain. Kotak <i>beat</i> akan keluar dari atas layar dan bergerak ke bawah. Elemen ini digunakan pada <i>shake multiplayer</i>

Sumber: Perancangan

Tabel 4.6. Prototipe elemen *area hit*

<i>Game Object</i>	Profil
	Nama : <i>area hit</i>
	Keterangan : Area hit digunakan pada <i>touch game</i> dan <i>shake multiPlayer</i> dimana memiliki panjang sama dengan panjang layar <i>smartphone</i> .

Sumber: Perancangan

Tabel 4.7. Prototipe elemen efek teks “GOOD”

<i>Game Object</i>	Profil
	Nama : <i>Good</i>
	Keterangan : Efek teks good keluar jika pemain memberikan inputan dengan benar ketika <i>beat</i> berada tepat di area hit.

Sumber: Perancangan

Tabel 4.8. Prototipe elemen efek teks “MISS”

<i>Game Object</i>	Profil
MISS	Nama : Miss
	Keterangan : Efek teks miss keluar jika pemain memberikan inputan salah ketika <i>beat</i> berada tepat di area hit.

Sumber: Perancangan

Tabel 4.9. Prototipe elemen *hand cursor*

<i>Game Object</i>	Profil
	Nama : <i>Hand cursor</i>
	Keterangan : <i>Hand cursor</i> digunakan untuk menunjukkan angklung delapan nada yang sedang ditekan

Sumber: Perancangan

4.1.12. *Music dan SFX*

Terdapat dua jenis musik latar pada permainan Sarinande angklung yaitu :

- 1) Musik utama yang dimainkan pada *menu* utama.
- 2) Musik *beat*, musik ini dimainkan pada in *game touch game* dan *shake multiplayer*. Musik *beat* merupakan file terpisah dari aplikasi dan diakses melalui SD Card. Musik yang dimainkan tergantung dari *beat* yang dimainkan.

Sound effect permainan Sarinande Angklung terdiri dari :

- 1) Delapan nada angklung, yaitu nada do, re, mi, fa, sol, la, si, dan do’.
- 2) Sound effect ketika menekan tombol.

4.2. *Technical design*

Pengimplementasian *game design* Simulator Angklung menggunakan ANDEngine dengan bahasa pemrograman berbasis objek yang dimodelkan dengan

menggunakan UML. Pemodelan yang akan dirancang pada permainan Sarinade Angklung ini terdiri dari perancangan diagram *use case*, diagram kelas, dan diagram *activity*.

4.2.1. Perancangan Diagram *Use case*

Diagram *use case* dalam perancangan *game* Sarinande Angklung digunakan untuk memodelkan *fungsi* dari *game*. Diagram *use case* ini melibatkan *user* sebagai aktor dan beberapa *use case*. Sebelum merancang diagram *use case*, yang harus dilakukan adalah mengidentifikasi aktor dan mengidentifikasi kebutuhan.

4.2.1.1. Identifikasi aktor

Tahap ini mempunyai tujuan untuk melakukan identifikasi terhadap aktor – aktor yang berinteraksi atau terlibat dalam *game*. Aktor dalam *game* Sarinande Angklung adalah pemain atau pengguna *game* Sarinande Angklung.

4.2.1.2. Identifikasi kebutuhan

Sebelum membuat permainan ini, terlebih dahulu dilakukan pengidentifikasian mengenai kebutuhan yang diperlukan. Kebutuhan tersebut menjelaskan mengenai spesifikasi sistem yang akan diimplementasikan serta spesifikasi perangkat keras dan perangkat lunak yang akan digunakan pada sistem. Berikut merupakan kebutuhan-kebutuhan yang akan diidentifikasi

a. Kebutuhan fungsional

Kebutuhan fungsional mendeskripsikan fungsionalitas atau layanan yang akan diberikan oleh sistem. Fungsi utama pada permainan Sarinande Angklung ditunjukkan pada Tabel 4.10.

Tabel 4.10. Daftar kebutuhan fungsional Sarinande Angklung

ID	KEBUTUHAN	AKTOR	USE CASE
SRS_01	Perangkat lunak harus menyediakan proses memainkan angklung dengan menggunakan <i>accelerometer</i> pada <i>shake Single Player</i>	Pemain	Memainkan <i>shake Single Player</i>

SRS_02	Perangkat lunak harus menyediakan <i>menu</i> untuk mengganti nada angklung pada <i>shake Single Player</i>	Pemain	Memilih nada angklung
SRS_03	Perangkat lunak harus menyediakan <i>menu</i> untuk memilih <i>file beat</i>	Pemain	Memilih <i>beat</i>
SRS_04	Perangkat lunak harus menyediakan proses memainkan <i>file beat</i> secara otomatis pada record	Pemain	Menjalankan demo <i>file beat</i>
SRS_05	Perangkat lunak harus menyediakan proses mengonversikan <i>file beat</i> menjadi <i>file audio</i>	Pemain	Mengonversikan <i>file beat</i> menjadi <i>file audio</i>
SRS_06	Perangkat lunak harus menyediakan proses untuk memainkan <i>file beat</i> yang dipilih, pada <i>touch game</i>	Pemain	Memainkan <i>touch game</i>
SRS_07	Perangkat lunak harus menyediakan tampilan untuk menampilkan <i>score</i> yang diperoleh setelah <i>touch game</i>	Pemain	Melihat <i>score touch game</i>
SRS_08	Perangkat lunak harus menyediakan tampilan untuk memilih peran pada <i>shake multiplayer</i>	Pemain	Memilih peran <i>multiplayer</i>
SRS_09	Perangkat lunak harus menyediakan tampilan untuk mengatur antara <i>server</i> dan <i>client</i> pada <i>multiplayer</i>	Pemain	Melakukan konfigurasi <i>multiplayer</i>
SRS_10	Perangkat lunak harus menyediakan proses untuk memainkan <i>file beat</i> pada <i>shake multiplayer</i>	Pemain	Memainkan <i>shake multiplayer</i>
SRS_11	Perangkat lunak harus menyediakan tampilan untuk menampilkan <i>score</i> yang didapat setelah memainkan <i>multiplayer</i>	Pemain	Melihat hasil <i>score multiplayer</i>

SRS_12	Perangkat lunak harus menyediakan proses untuk memiliha <i>background music</i> pada <i>composer</i>	Pemain	Memilih musik
SRS_13	Perangkat lunak harus menyediakan proses untuk menyimpan nada angklung yang kemudian akan disimpan dalam <i>file beat</i>	Pemain	Membuat <i>file beat</i>
SRS_14	Perangkat lunak harus menyediakan informasi mengenai angklung dan <i>credit</i> pembuat aplikasi	Pemain	Melihat <i>menu about</i>
SRS_15	Perangkat lunak harus menyediakan tampilan untuk pengaturan volume suara dan pengaturan vibrasi	Pemain	Melihat <i>menu setting</i>
SRS_16	Perangkat lunak harus menyediakan tampilan untuk menampilkan semua <i>score</i> tertinggi dari semua <i>beat</i> yang telah dimainkan oleh pengguna	Pemain	Melihat <i>menu highscore</i>
SRS_17	Perangkat lunak harus menyediakan petunjuk menggunakan aplikasi bagi pemain	Pemain	Melihat <i>menu how to play</i>

Sumber: Perancangan

b. Kebutuhan non-fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang tidak secara langsung berhubungan dengan fungsi spesifik yang disediakan oleh sistem tapi dapat berhubungan dengan properti sistem seperti kemampuan piranti yang digunakan. Berikut merupakan spesifikasi kebutuhan non-fungsional dari permainan mobile simulator angklung

1. Kebutuhan perangkat keras

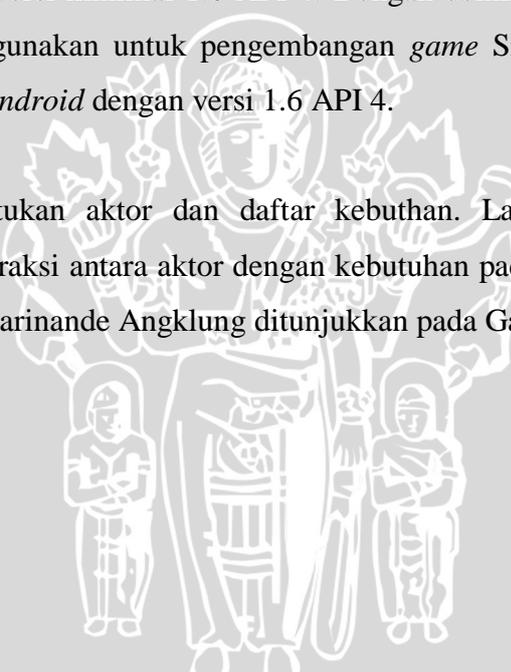
Untuk dapat memenuhi spesifikasi yang dibutuhkan, maka aplikasi ini memerlukan perangkat *smartphone* dengan spesifikasi sebagai berikut :

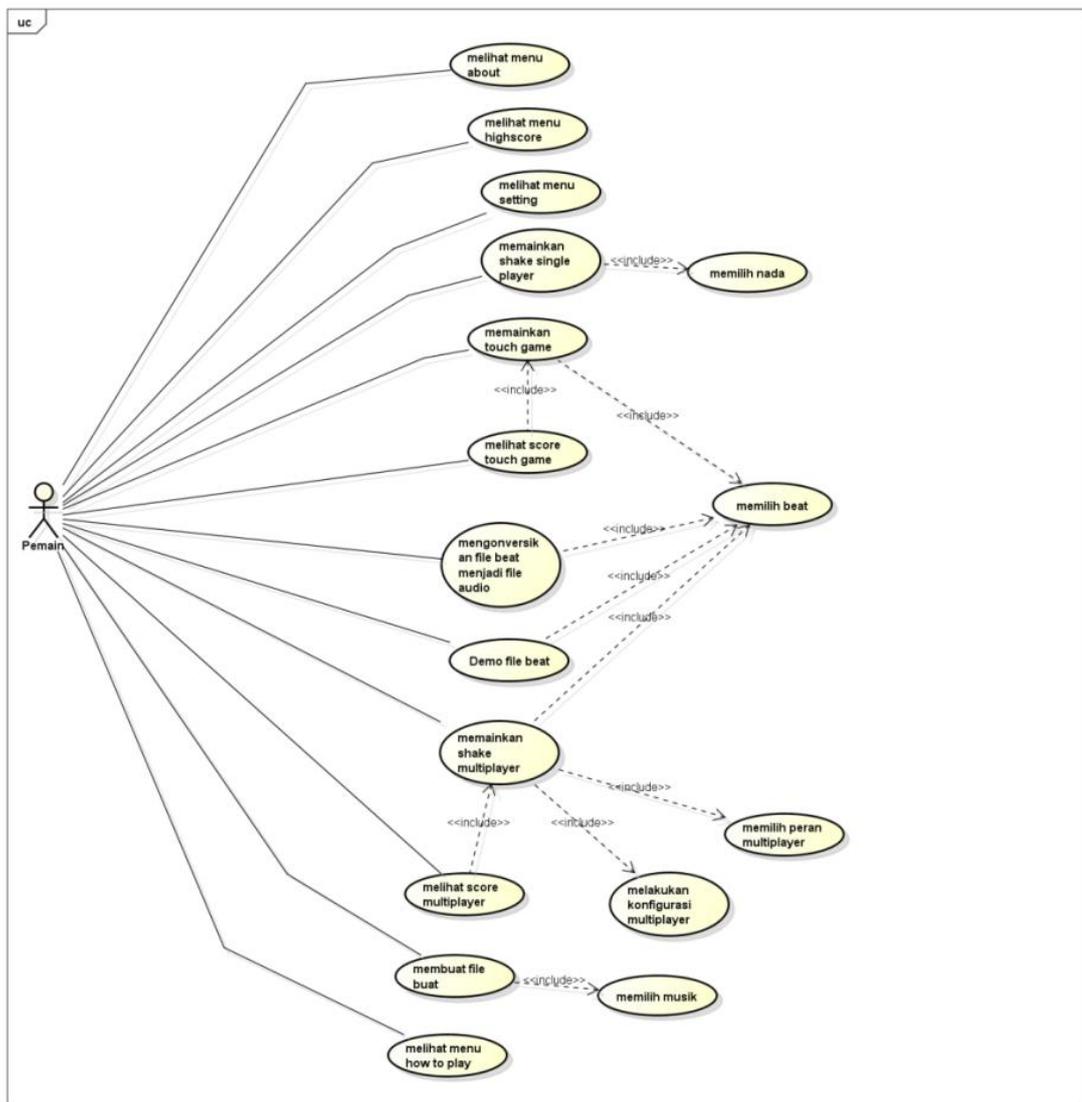
- Memiliki sensor *accelerometer*
- Memiliki koneksi internet (WIFI).
- Memiliki layar sentuh.

2. Kebutuhan perangkat lunak

Pengembangan game Sarinande Angklung menggunakan *game engine AndEngine* yang berjalan pada sistem operasi *android* dengan versi minimal 1.6 API 4. Dengan demikian sistem operasi yang digunakan untuk pengembangan game Simulator Angklung adalah *android* dengan versi 1.6 API 4.

Setelah menentukan aktor dan daftar kebutuhan. Langkah selanjutnya adalah memetakan interaksi antara aktor dengan kebutuhan pada sistem. Diagram *use case* untuk game Sarinande Angklung ditunjukkan pada Gambar 4.22.



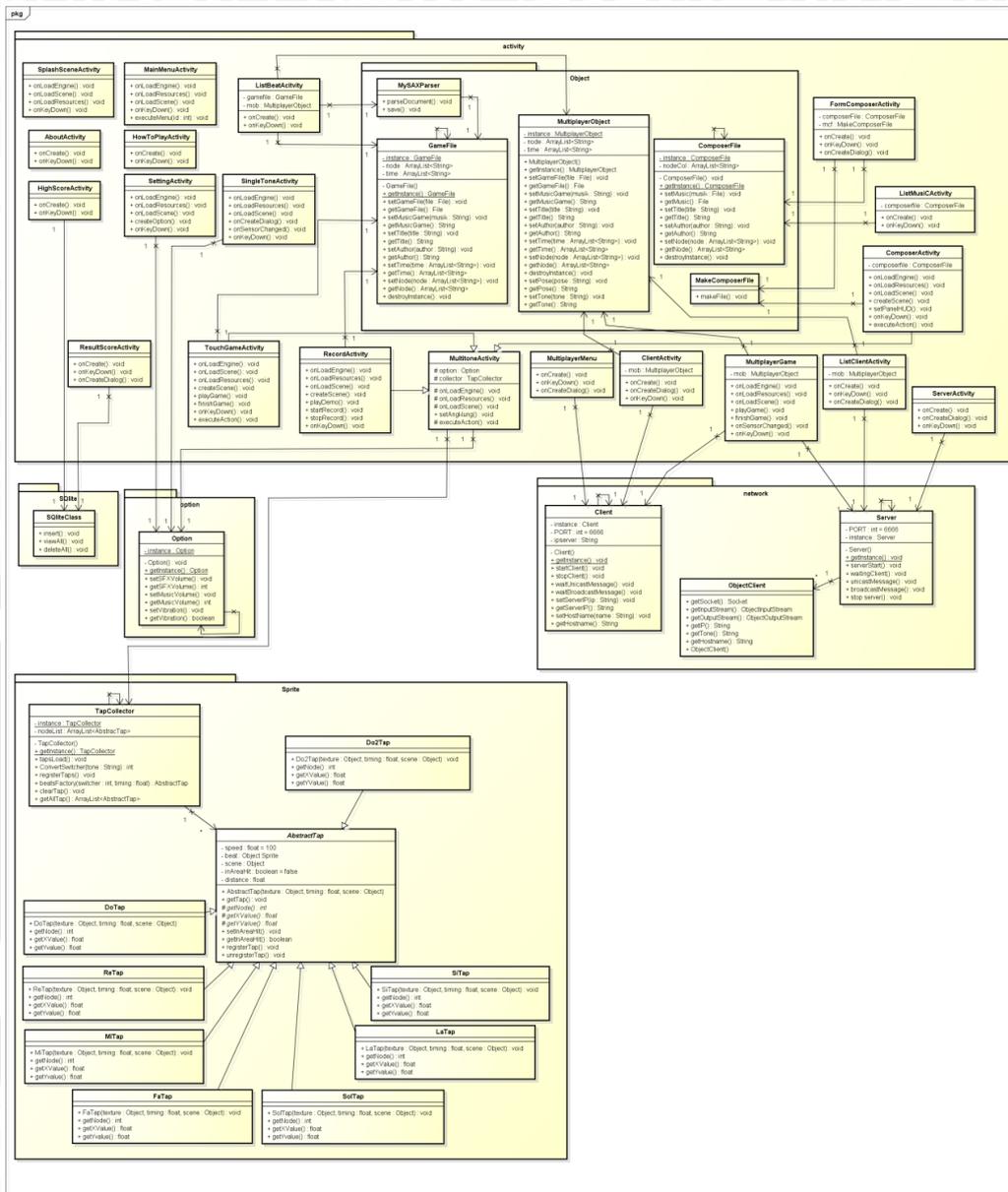


Gambar 4.22. Diagram use case permainan Sarinande Angklung

Sumber: Perancangan

4.2.2. Perancangan Diagram Kelas

Perancangan diagram kelas untuk memodelkan kelas-kelas dan *interface-interface* yang dibutuhkan dalam pembuatan perangkat lunak. Perancangan diagram kelas pada permainan sarinande angklung ditunjukkan pada Gambar 4.23.



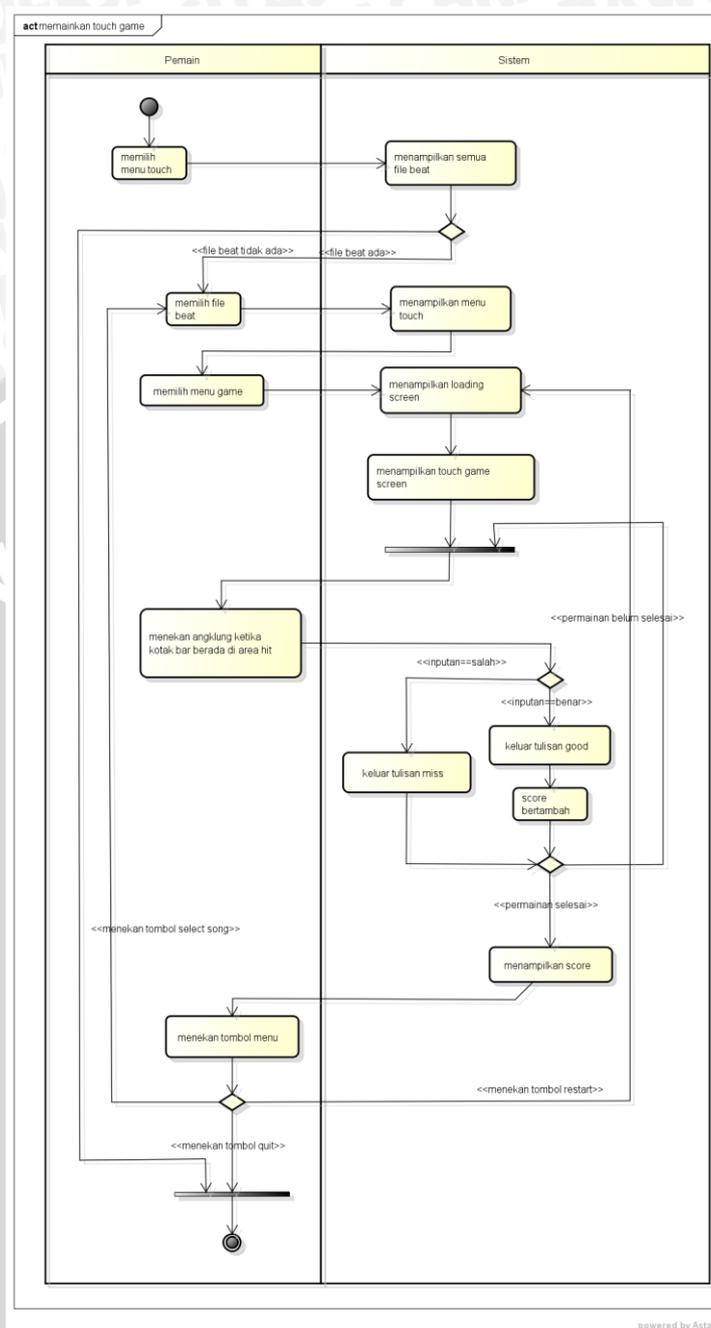
Gambar 4.23. Rancangan diagram kelas permainan sarinande anglung

Sumber: Perancangan

4.2.3. Perancangan Activity Diagram

Pembuatan *activity diagram* ini bertujuan untuk menggambarkan urutan aktivitas dari proses pada setiap *use case* yang ada. Berikut merupakan gambar dari masing-masing *activity diagram*.



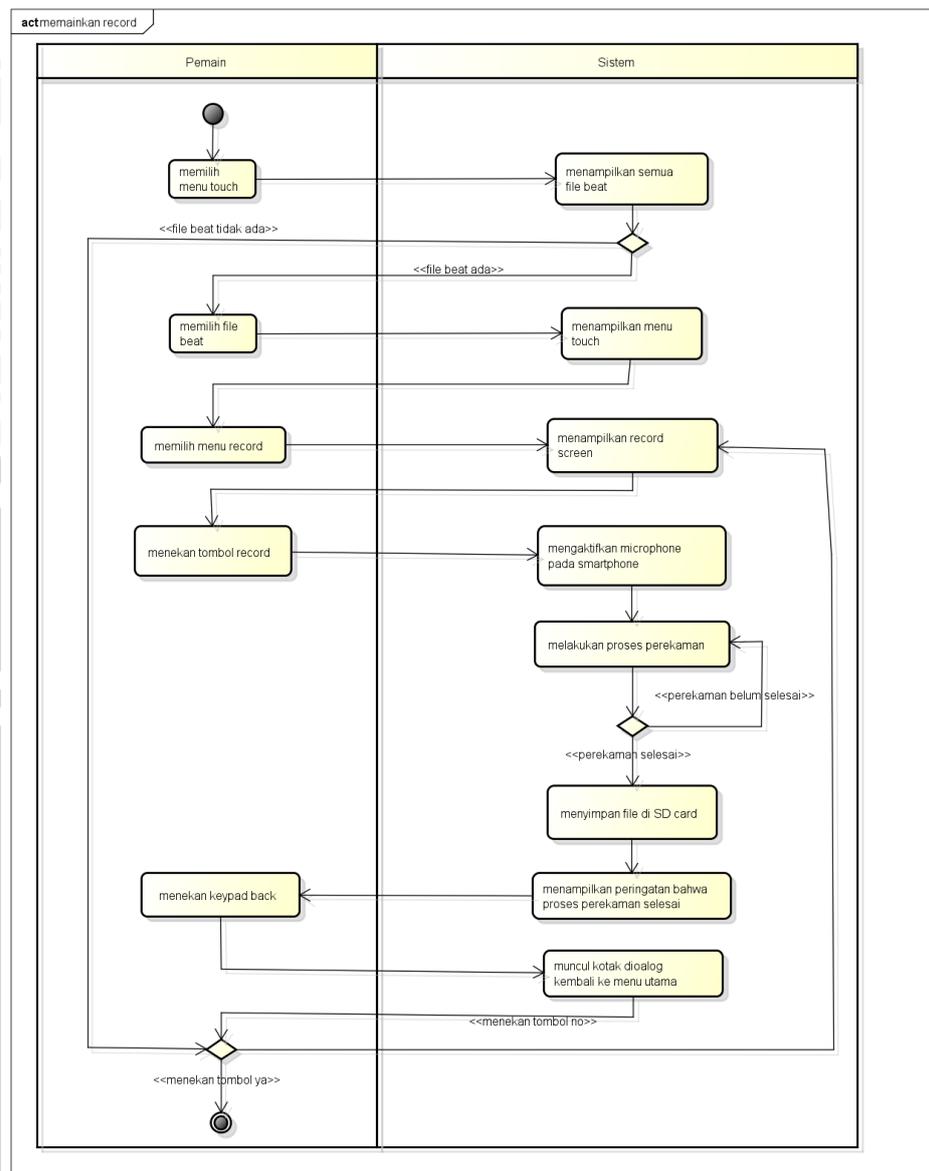


Gambar 4.25. Activity diagram memainkan touch game

Sumber: Perancangan

4.2.3.3. Activity diagram memainkan record

Pada Gambar 4.26. terdapat activity diagram memainkan record, yang didalamnya terdapat interaksi antara pengguna dengan sistem atau aplikasi yang ada.

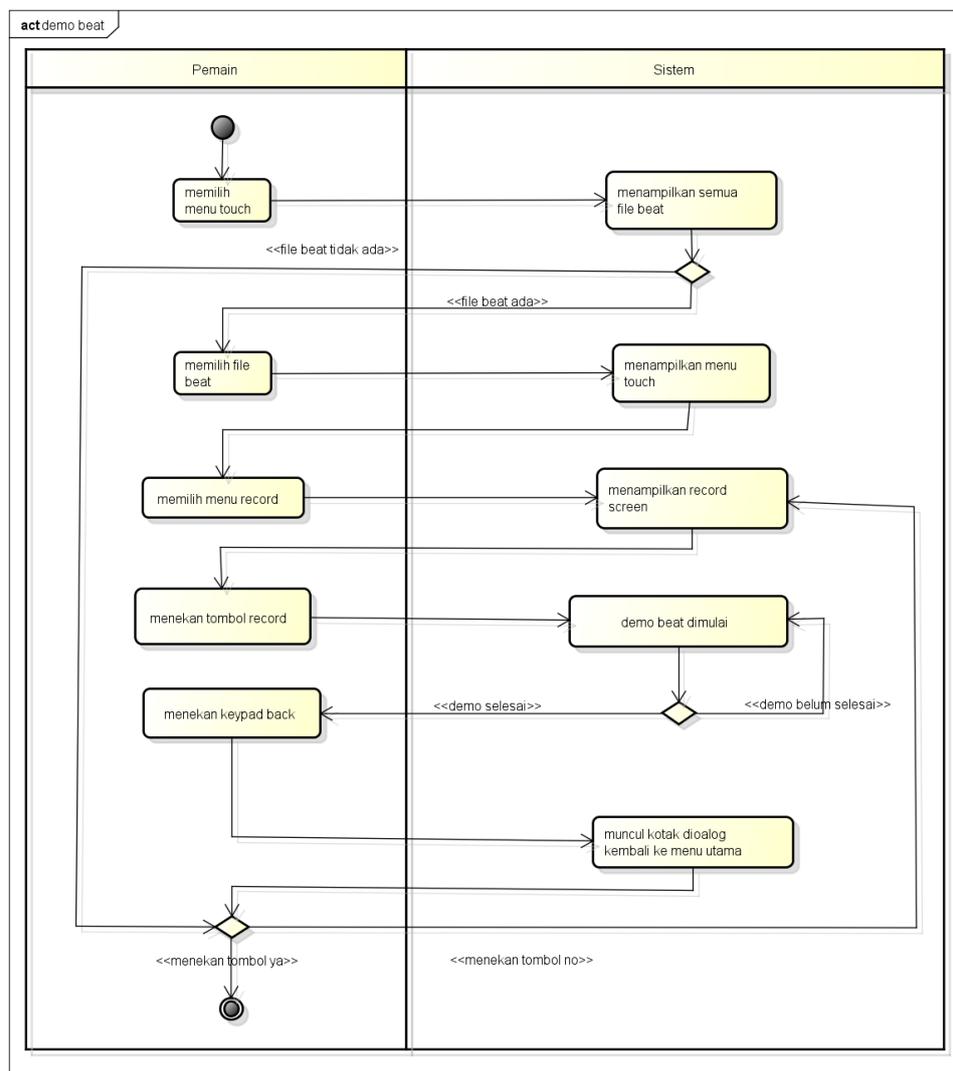


Gambar 4.26. Activity diagram memainkan record

Sumber: Perancangan

4.2.3.4. Activity diagram menjalankan demo beat

Pada Gambar 4.27. terdapat activity diagram menjalankan demo beat, yang didalamnya terdapat interaksi antara pengguna dengan sistem atau aplikasi yang ada

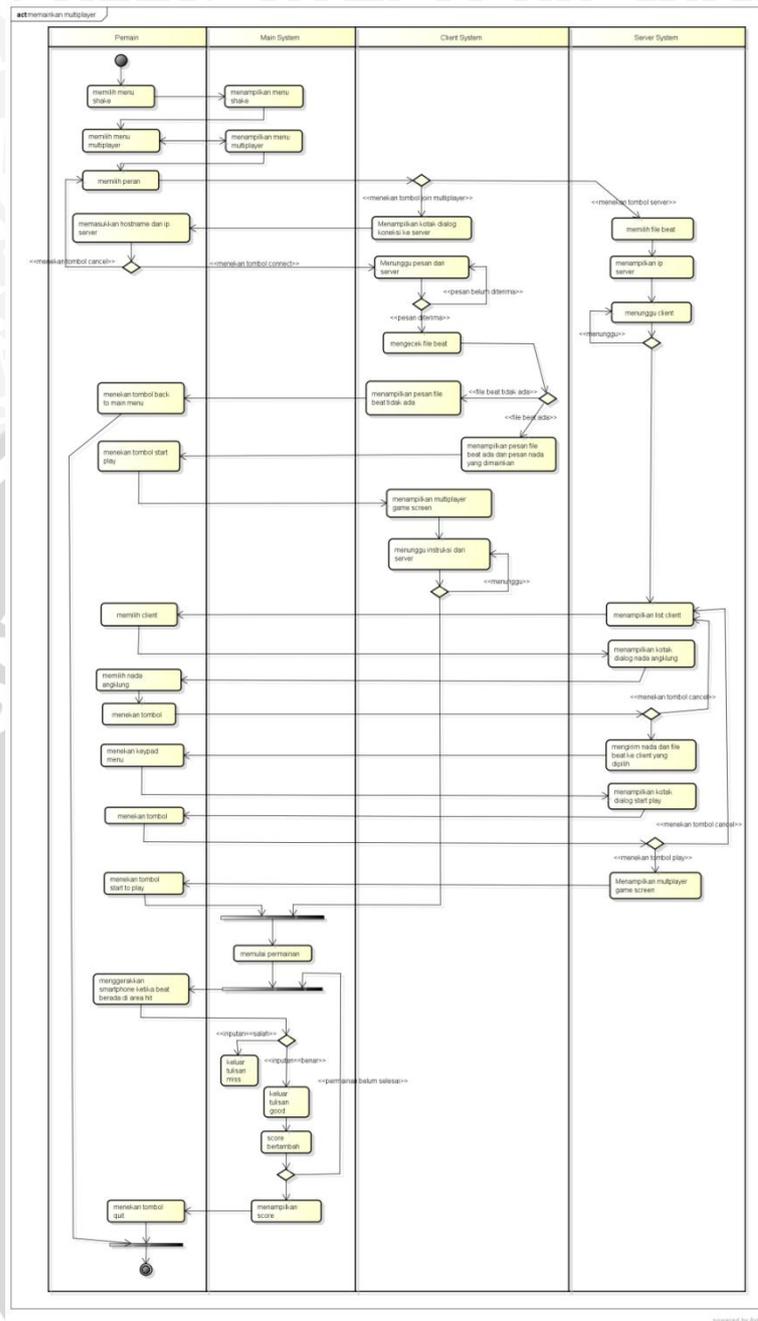


Gambar 4.27. Activity diagram menjalankan demo beat

Sumber: Perancangan

4.2.3.5. Activity diagram memainkan multiPlayer

Pada Gambar 4.28. terdapat activity diagram memainkan multiPlayer, yang didalamnya terdapat interaksi antara pengguna dengan sistem atau aplikasi yang ada.

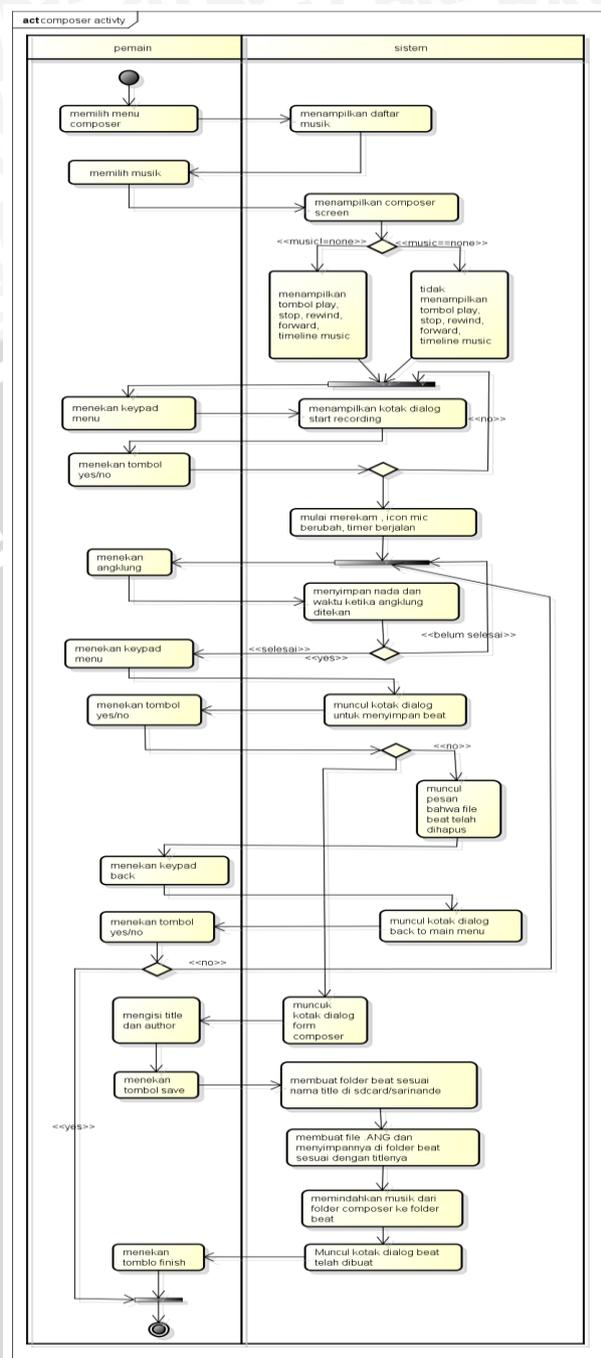


Gambar 4.28. Activity diagram memainkan multiplayer

Sumber: Perancangan

4.2.3.6. Activity diagram membuat file beat

Pada Gambar 4.29. terdapat activity diagram membuat file beat, yang didalamnya terdapat interaksi antara pengguna dengan sistem atau aplikasi yang ada.



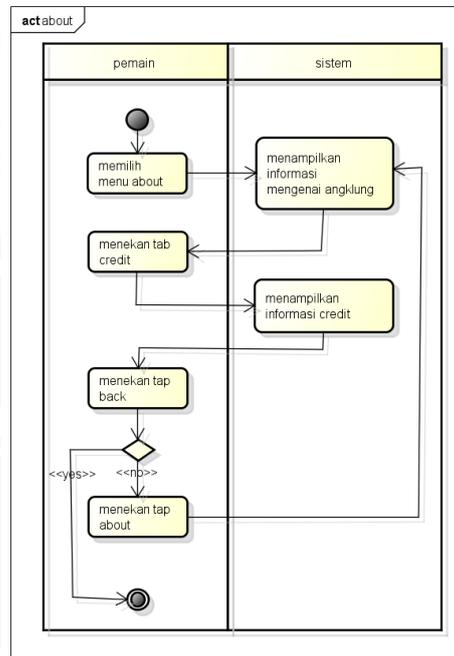
Gambar 4.29. Activity diagram membuat file beat

Sumber: Perancangan

4.2.3.7. Activity diagram melihat menu about

Pada Gambar 4.30. terdapat activity diagram melihat menu about, yang didalamnya terdapat interaksi antara pengguna dengan sistem atau aplikasi yang ada.



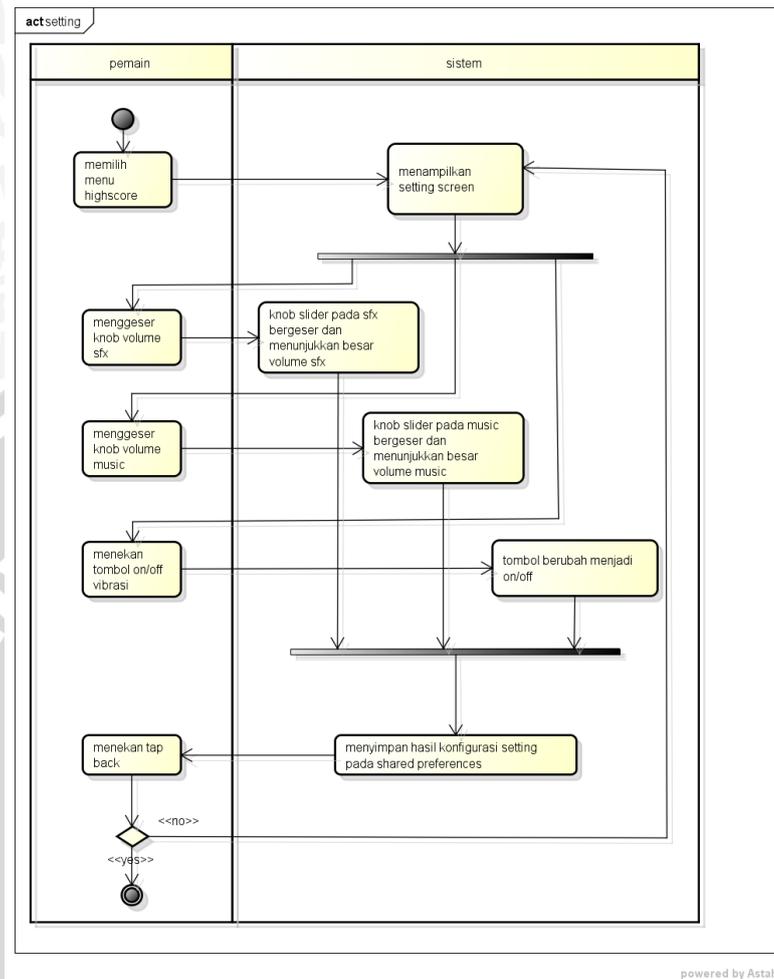


Gambar 4.30. Activity diagram melihat menu about

Sumber: Perancangan

4.2.3.8. Activity diagram melihat menu setting

Pada Gambar 4.31. terdapat activity diagram melihat menu setting, yang didalamnya terdapat interaksi antara pengguna dengan sistem atau aplikasi yang ada.

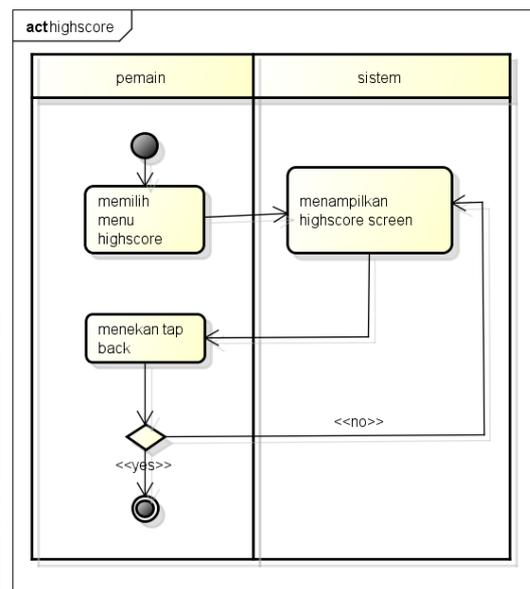


Gambar 4.31. Activity diagram melihat menu setting

Sumber: Perancangan

4.2.3.9. Activity diagram melihat menu highscore

Pada Gambar 4.32. terdapat *activity diagram* melihat *menu highscore*, yang didalamnya terdapat interaksi antara *pengguna* dengan sistem atau aplikasi yang ada.

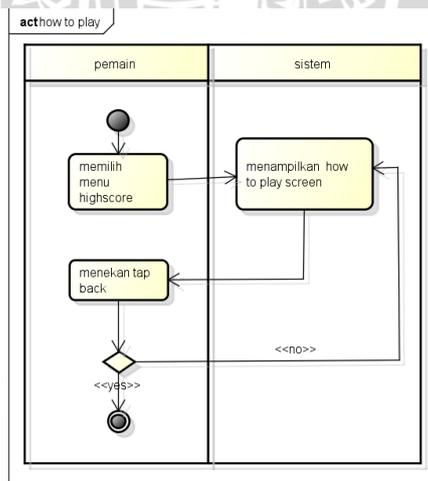


Gambar 4.32. Activity diagram melihat menu highscore

Sumber: Perancangan

4.2.3.10. Activity diagram melihat menu how to play

Pada Gambar 4.33. terdapat activity diagram melihat menu how to play, yang didalamnya terdapat interaksi antara pengguna dengan sistem atau aplikasi yang ada.



Gambar 4.33. Activity diagram melihat menu how to play

Sumber: Perancangan