# BAB III

# METODOLOGI DAN PERANCANGAN SISTEM

## 3.1 Metodologi Penelitian

Metodologi penelitian yang akan digunakan akan dijabarkan sebagai berikut.

1. Studi Literatur

Studi literatur dilakukan dengan mencari-cari sumber informasi yang beredar saat ini. Informasi-informasi yang dibutuhkan dalam penelitian ini antara lain adalah pencak silat, *game*, *virtual reality*, HMSAM (Hedonic-Motivation System Adoption Model), *fisher-yates shuffle,* dan skala likert. Informasi-informasi yang telah didapatkan akan dijadikan acuan untuk membangun aplikasi.

2. Perancangan Sistem

Perancangan sistem dimulai dengan membuat alur permainan dengan pembuatan perancangan spesifikasi permainan, *flowchart,* perancangan antar muka, perancangan pemakaian *asset* pada aplikasi, dan perancangan kuesioner dengan pemodelan HMSAM.

3. Pembuatan Aplikasi

Aplikasi dikembangkan dengan menggunakan Unity. Aplikasi akan memuat algoritma *fisher-yates shuffle* sebagai algoritma untuk mengacak posisi awal objek pukulan dari *game* serta implementasi gerakan pukulan ilmu seni bela diri pencak silat berupa gerakan lengan antara lain gerakan pukulan, pukulan sangkal, tebangan, dan dobrakan.

4. Uji Coba Aplikasi

Tahan uji coba aplikasi dilakukan dengan dua tahap. Tahap pertama adalah mengujicobakan pergerakan pukulan yang tepat untuk objek pukulan yang bersangkutan sesuai dengan pergerakan lengan seni ilmu bela diri pencak silat. Tahap berikutnya adalah uji coba aplikasi untuk memastikan algoritma *fisher-yates shuffle* berfungsi. Pengujian dilakukan dengan cara memainkan permainan pada sebuah level. Output diperoleh dari hasil proses debug dari aplikasi Unity terhadap isi dari *array* kumpulan posisi awal yang telah ditetapkan. Algoritma *fisher-yates shuffle* akan dilakukan sebanyak yang dibutuhkan oleh program untuk mengacak posisi awal setiap objek. Tahap uji coba berikutnya adalah dengan mengujicobakan aplikasi kepada pemain serta mengisi kuesioner dari pemodelan HMSAM. Tahap uji coba ini dilakukan untuk memperoleh nilai *immersive* dan *behavioral intention to use* sesuai dengan pemodelan HMSAM sebagai nilai kepuasan pemain.

5. Analisis Ujicoba Aplikasi

Analisis hasil uji coba aplikasi dilakukan dengan cara membandingkan hasil probabilitas dari sebuah objek pukulan yang sama pada permainan yang berbeda. Hal ini dapat menjadikan acuan keberhasilan algoritma *fisher-yates shuffle* dalam proses pengacakan posisi awal dari sebuah objek pukulan. Pengecekan tidak adanya *bug* dari segi permainannya. Hal ini digunakan untuk memastikan pergerakan pukulan yang sesuai dengan objek pukulan yang tepat. Analisis ujicoba dilanjutkan dengan menghitung nilai *immersive* dan *behavioral intention to use* dari hasil pengisian kuisioner pemain dengan menggunakan pemodelan HMSAM.

6. Penulisan Laporan

Penulisan laporan karya ilmiah mencakup seluruh rangkaian penelitian dimulai dengan studi literatur hingga kesimpulan hasil penelitian. Penulisan ini dilakukan dengan adanya bantuan dari dosen pembimbing bersangkutan.

## 3.2 Analisa dan Perancangan Sistem

Aplikasi yang akan dibuat adalah sebuah *game* ritme *single player. Game* ini menyediakan lagu beserta *level* dari lagu tersebut. Permainan akan berjalan selama masih ada objek pukulan yang ada. Pemain diharapkan untuk dapat menghancurkan objek pukulan yang bergerak dari posisi awal yang akan diacak melalui algoritma *fisher-yates shuffle.* Objek pukulan ini akan bergerak menuju posisi pemain sesuai dengan cara pukulan dari ilmu seni bela diri pencak silat yang ada pada permainan. Pemain dapat menggunakan tombol 'A' yang ada pada *controller* kanan dari oculus touch untuk bernavigasi.

### 3.2.1 Spesifikasi Permainan

Permainan yang akan dibangun adalah permainan ritme *Virtual Reality* yang dimana pemain harus memukul objek pukulan yang ada sesuai dengan gerakan pukulan yang ada pada seni ilmu bela diri pencak silat. Berikut perancangan *formal element* yang akan ada sebagai spesifikasi permainan .

1. *Player*

   *a) Single Player Game*

   Permainan ini hanya dapat dimainkan oleh satu orang pemain saja.

   *b) Player vs Game*

   Permain akan berusaha untuk memenangkan permainan dengan sekumpulan aturan dan rintangan yang telah disiapkan dalam aplikasi.

2. *Objective*

   Pemain dapat menyelesaikan *level* dengan mendapatkan skor setinggi-tingginya.

3. *Procedures*

   a) Pemain masuk ke halaman main menu yang kemudian akan mengarahkan pemain ke dalam permainan inti dengan menekan tombol *play*. Jika pemain ingin memahami cara memainkan permainan, permain dapat menekan tombol *tutorial*.

   b) Pemain diberikan list lagu untuk dipilih, kemudian pemain juga perlu memilih *level* permainan yang diinginkan sebelum menekan tombol *play*

   c) Permainan akan dimulai dengan seiringnya lagu berjalan. Pemain harus memukul objek pukulan tersebut dengan gerakan pencak silat yang sesuai untuk mendapatkan skor. Objek pukulan akan mendekati pemain, pemain hanya dapat memukul objek setelah sudah cukup dekat dan dapat dipukul.

d) Permainan akan selesai ketika lagu berhenti dan halaman *result* akan menampilkan performa pemain dari permainan yang baru saja dimainkan. Pemain dapat memilih lagu berikutnya dan level berikutnya kembali menuju prosedur b).

e) Pemain dapat kembali ke halaman main menu dengan menekan tombol *back* dan menghentikan permainan dengan menekan tombol exit pada main menu.

4. *Rules*

Pemain tidak dapat memulai permainan tanpa adanya *lagu* dan *level* yang sudah dipilih. Objek pukulan yang berwarna gelap dipukul dengan menggunakan tangan kanan sedangkan objek pukul dengan warna terang dipukul dengan menggunakan tangan kiri. Objek dobrakan memiliki warna khusus yang menandakan harus dipukul secara bersamaan dengan posisi tangan yang sesuai dengan gerakan dobrakan pencak silat.

5. *Resource*

Tangan yang dapat dikepalkan dengan menggunakan *hand trigger* pada *touch controller* dan dapat digerakan secara bebas untuk memukul objek pukulan tersebut. Gerakan yang diharapkan disediakan dalam bentuk *video* pengenalan pencak silat pada halaman tutorial.

6. *Conflict*

Pemain harus dapat menyelesaikan *level* dengan sekuat tenaga yang ada untuk dapat menggerakan tangan sesuai dengan gerakan pencak silat yang sesuai.

7. *Boundaries*

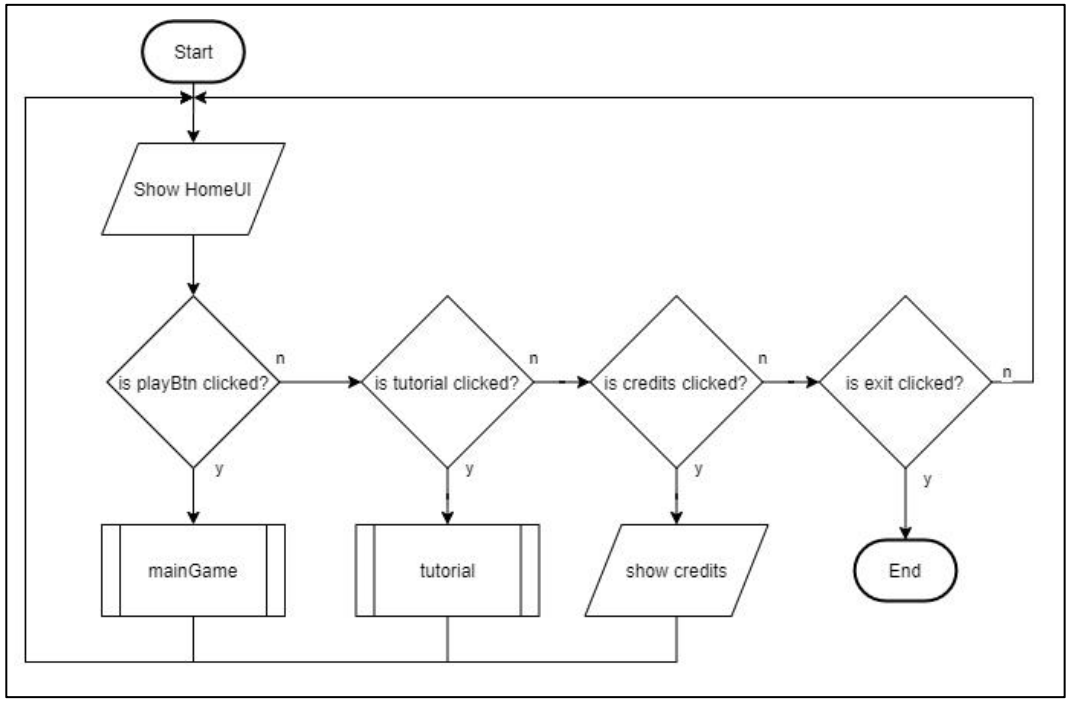
   Pemain hanya dapat bergerak pada batasan yang ada pada *virtual reality boundaries* secara nyata.

8. *Outcome*

   Halaman hasil akan menampilkan jumlah pukulan yang tepat, pukulan beruntun terbanyak serta *rating* dari performa pemain saat bermain.

### 3.2.2 Flowchart Main Menu

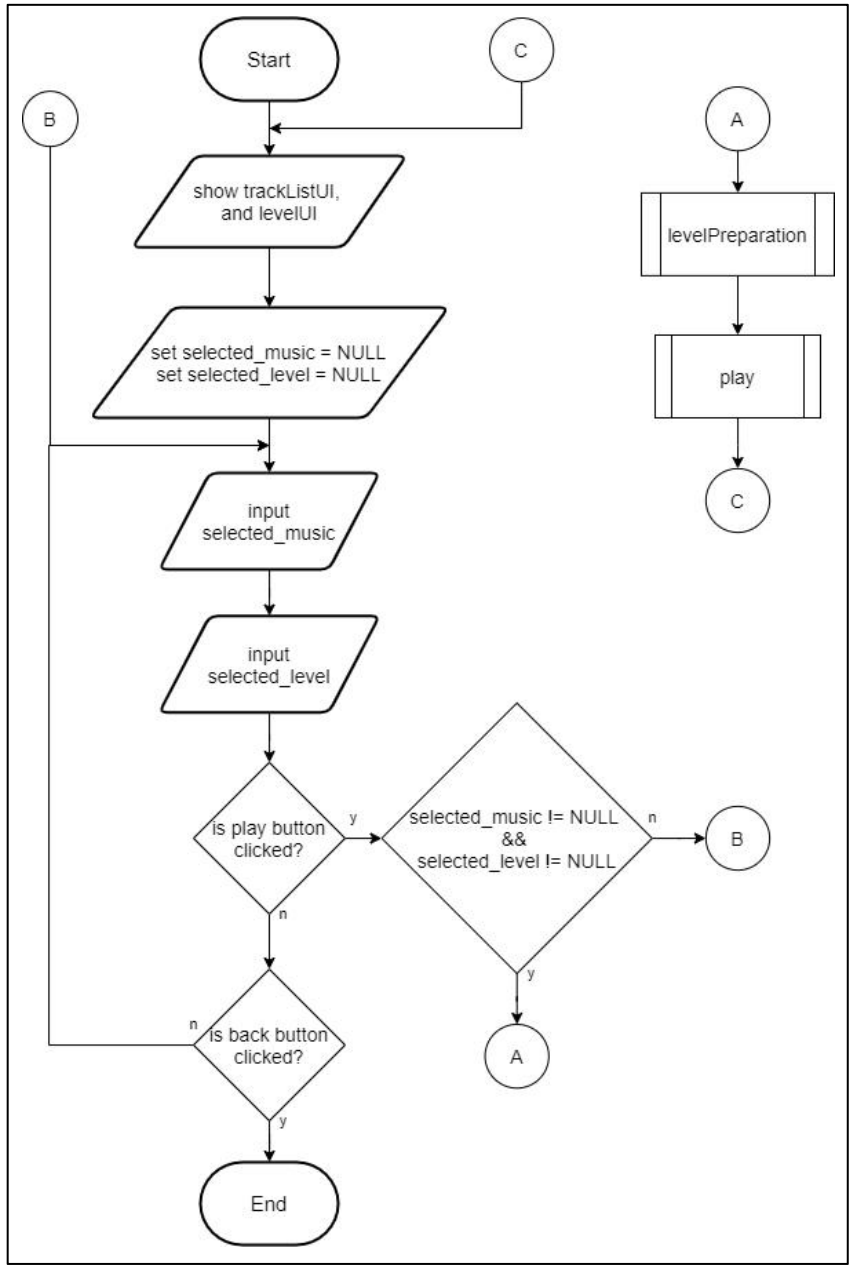
Aplikasi dimulai dengan menampilkan Main Menu sebagai antar muka yang pertama. Aplikasi menyediakan empat tombol yaitu, *play*, *tutorial*, *credits* dan *exit*. Jika pemain menekan tombol *play*, maka aplikasi akan melanjutkannya dengan menjalankan modul mainGame. Jika pemain menekan tombol *tutorial*, maka aplikasi akan menampilkan halaman *tutorial*. Jika pemain menekan tombol credits, maka aplikasi akan menampilkan halaman *credits*. Jika pemain menekan tombol *exit*, maka permaianan akan diselesaikan. Gambar 3.1 merupakan gambar dari Flowchart Main Menu.




Gambar 3.1 Flowchart Main Menu.

### 3.2.3 Flowchart Modul MainGame

Modul MainGame diawali dengan menampilkan halaman trackListUI dan levelUI. Pemain baru dapat memulai permainan dengan menekan tombol *play* setelah pemain memilih lagu dan level yang diinginkan. Setelah tombol play ditekan, aplikasi akan menyiapkan permainan melalui modul levelPreparation. Aplikasi akan memulai permainan ketika modul levelPreparation telah selesai dan siap untuk dijalankan. Gambar 3.2 merupakan Flowchart modul mainGame.




Gambar 3.2 Flowchart Modul mainGame

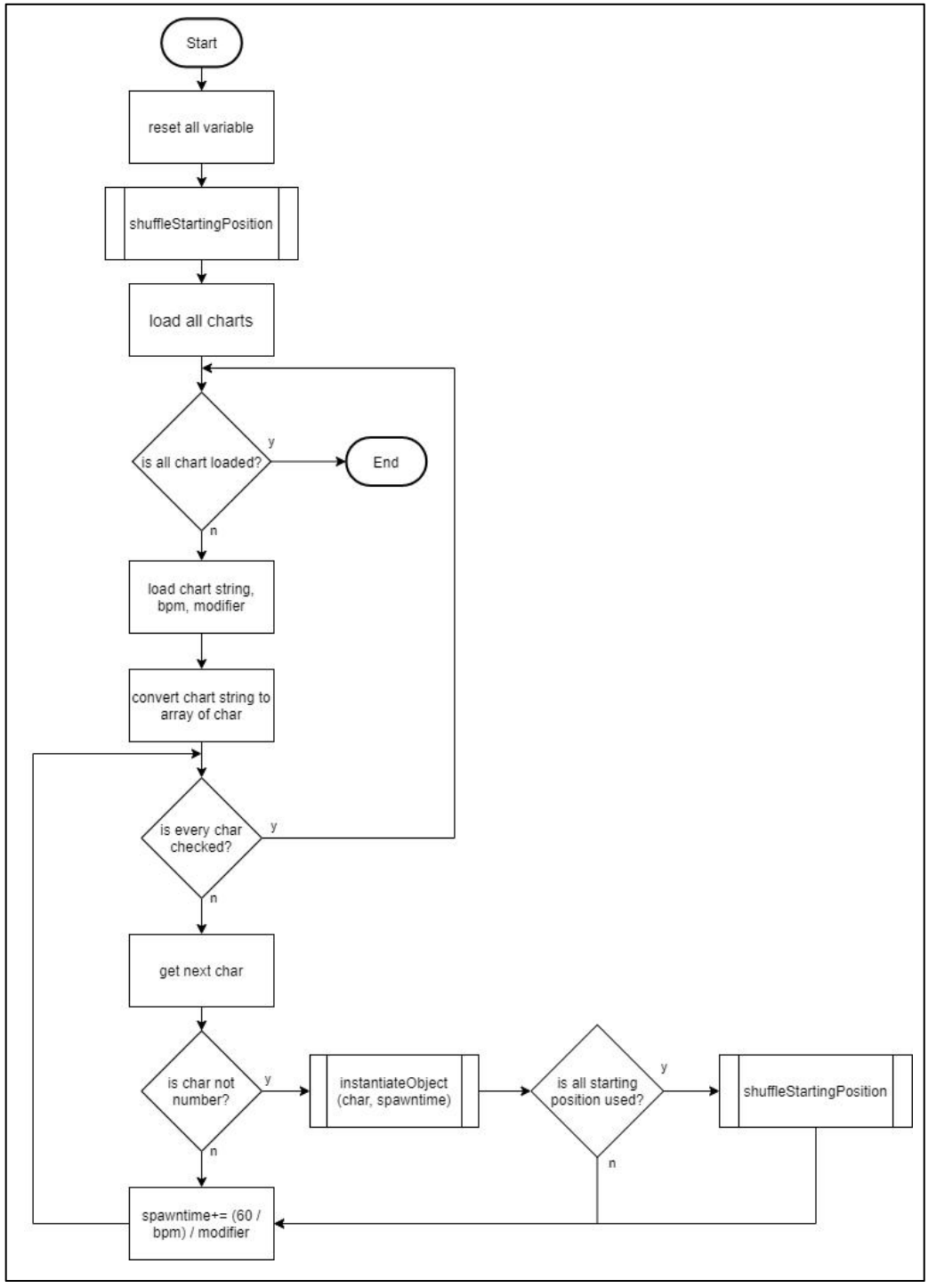
### 3.2.4 Flowchart Modul levelPreparation

Modul levelPreparation bertugas untuk mempersiapkan aplikasi untuk dapat siap untuk dimainkan. Modul ini diawali dengan proses memuat ulang seluruh variabel kembali seperti semula dan kemudian melakukan proses shuffling

menggunakan algoritma *fisher-yates shuffle* pada posisi awal dari objek-objek yang akan dikeluarkan. Aplikasi akan melakukan proses pemuatan charts dari lagu tersebut. Charts adalah notasi-notasi yang sudah disiapkan didalam sebuah *file* JSON yang menggambarkan objek apa yang akan dimunculkan pada ketukan yang bersangkutan. Sebuah chart memuat sebuah string notasi dan jumlah ketukan per menit. *String* pada chart dikonversikan terlebih dahulu menjadi *array of char.* Karakter-karakter yang ada pada *array* yang kemudian akan dicek oleh aplikasi objek pukulan apakah yang akan dimunculkan pada permainan. Jika karakter (huruf) tersebut adalah sebuah angka, maka aplikasi akan menjalankan modul instantiateObject untuk megnhasilkan objek pukulan sesuai dengan karakter (huruf) yang ada. Jika nilai pada list posisi tersebut sudah digunakan semuanya, maka aplikasi akan menjalankan modul shuffleStartingPoint kembali untuk mengacak kembali urutan dari posisi awal yang akan ditempati oleh objek pukulan selanjutnya. Spawntime adalah waktu yang dibutuhkan sebuah objek untuk menunggu gilirannya dikeluarkan pada permainan berlangsung. Spawntime didapatkan menggunakan rumus sebagai berikut.

$$spawntime = spawntime + (\frac{60}{BPM} * \frac{1}{multiplier}) \quad \ldots(1)$$

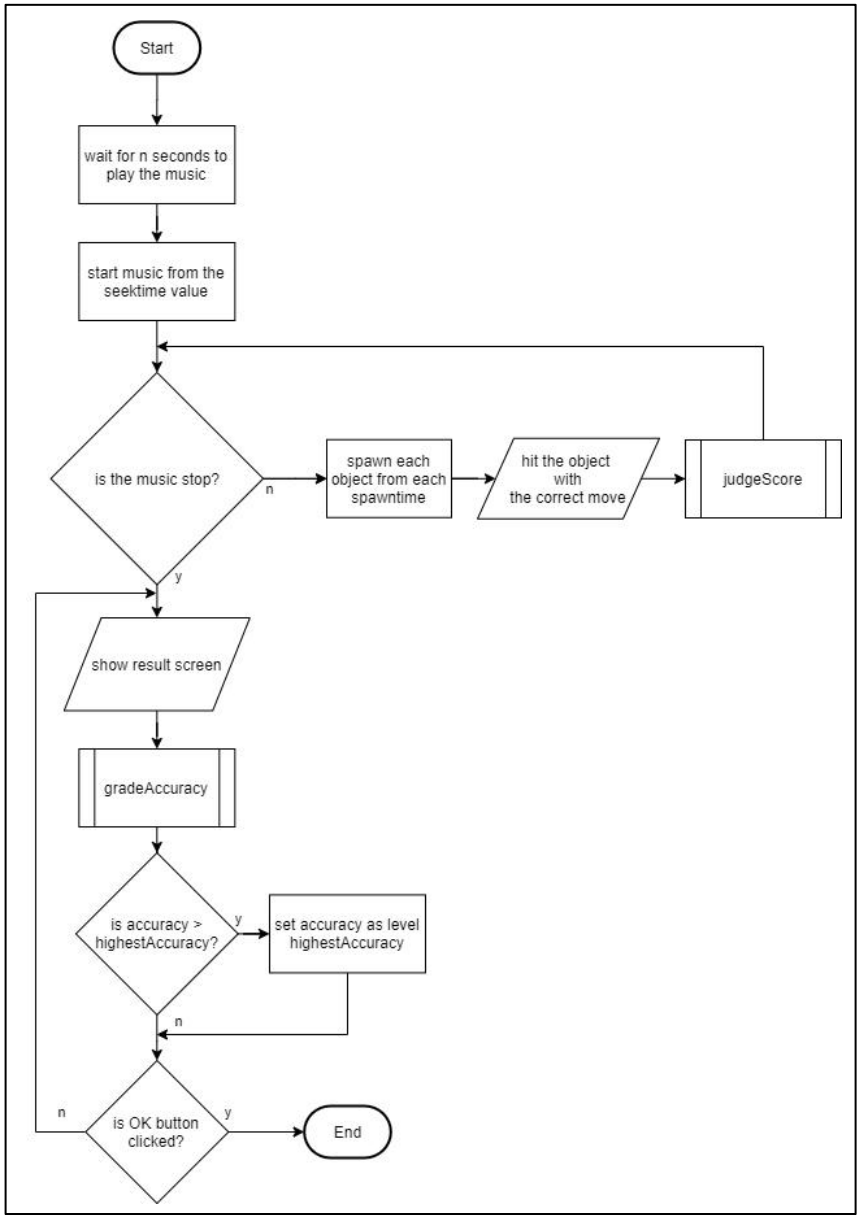
BPM merupakan satuan ketukan per menit dari lagu tersebut sedangkan multiplier adalah satuan yang digunakan untuk mempercepat atau memperlambat ketukan lagu agar objek dapat dikeluarkan tepat dengan lagu yang sedang dimainkan. Gambar 3.3 merupakan Flowchart dari modul levelPreparation.




Gambar 3.3 Flowchart Modul levelPreparation

### 3.2.5 Flowchart Modul play

Modul play bertugas untuk mengatur alur kerja aplikasi pada saat permainan dimulai. Modul ini dimulai dengan menetapkan waktu tunggu untuk lagu dan menetapkan *offset* lagu tersebut akan diputar. *Offset* digunakan untuk melompatkan n detik dari lagu tersebut sedangkan waittime berguna untuk menunggu sekian detik untuk lagu tersebut baru diputar. Variabel ini berfungsi untuk menyesuaikan ketukan lagu dengan ketukan objek pukulan yang ada pada permainan. Setelah lagu diputar, objek-objek pukulan yang telah disiapkan akan kemudian secara bergilir akan dikeluarkan sesuai dengan waktu spawntime yang dimiliki oleh masing-masing objek pukulan. Objek pukulan yang telah melawati waktu spawntime akan bergerak menuju pemain dengan kecepatan yang telah ditentukan. Setiap objek yang datang harus dipukul oleh pemain dengan tepat sesuai dengan gerakan yang sesuai. Pengeluaran objek akan terus dilakukan selama lagu tidak berhenti. Halaman hasil dari permainan akan ditampilkan setelah permainan berakhir. Jika *accuracy* yang didapatkan lebih besar dibandingkan dengan *levelAccuracy* maka *levelAccuracy* diset menjadi *accuracy* tersebut. Pemain dapat kembali ketika menekan tombol “ok” pada *result screen*. Gambar 3.4 merupakan Flowchart modul play.

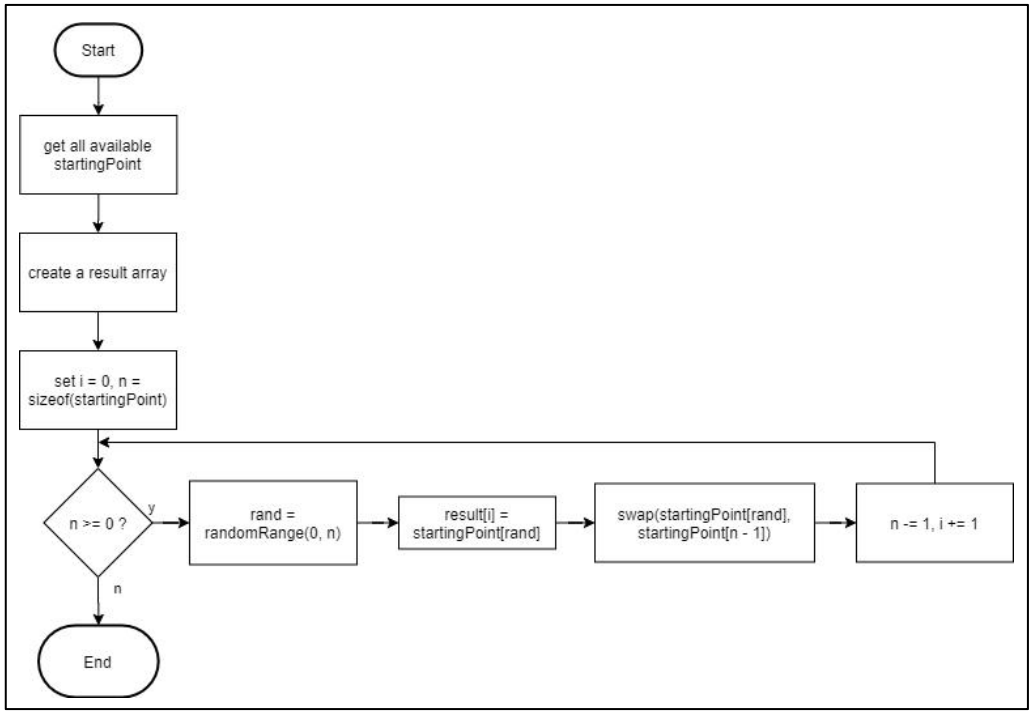



Gambar 3.4 Flowchart Modul play

### 3.2.6 Flowchart Modul shuffleStartingPoint

Modul shuffleStartingPoint berfungsi untuk mengacak sekumpulan *starting point* yang sudah disiapkan pada permainan sebagai posisi-posisi awal dari objek pukulan. Posisi-posisi tersebut sudah ditempatkan sebelumnya pada Unity Scene Editor. Modul shuffleStartingPoint dilanjutkan dengan mempersiapkan variabel

result untuk menampung hasil dari algoritma *fisher-yates shuffle.* Pengacakan dilakukan sebanyak *size of* startingPoint. Iterasi diawali dengan pengangkatan nilai random dari nilai 0 hingga n-1 dari starting point yang belum terpilih. Berdasarkan nilai random tersebut, tampung nilai startingPoint ke-nilai rand tersebut pada array hasil. Proses selanjutnya adalah dengan menukar startingPoint index ke-rand dengan startingPoint index ke-(n-1). Hal ini dilakukan untuk memastikan bahwa index-ke rand sudah terpilih dan tidak boleh dipilih kembali pada saat menaikan angka acak selanjutnya. Array result akan dikembalikan sebagai array startingPoint yang baru. Gambar 3.5 merupakan Flowchart dari modul shuffleStartingPoint.



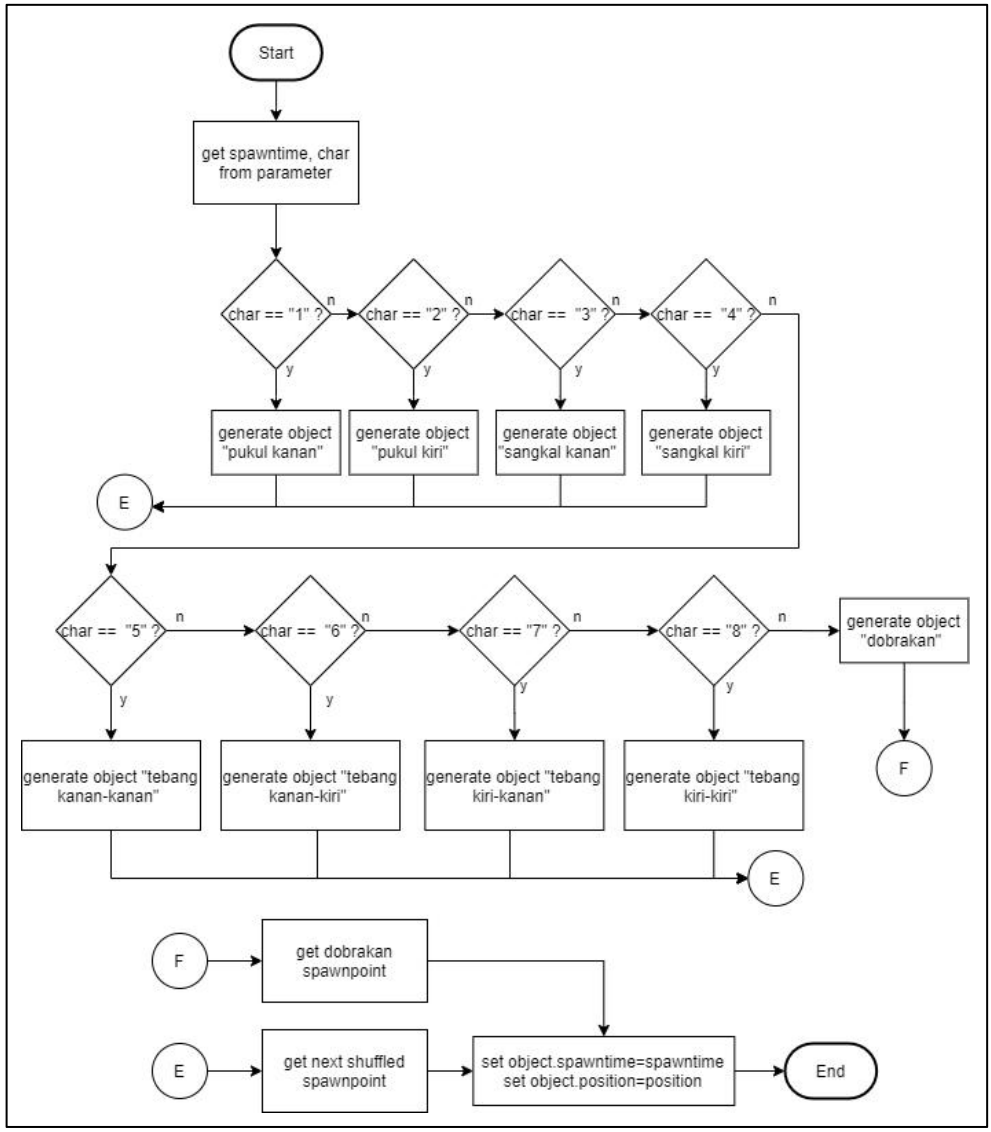

Gambar 3.5 Flowchart Modul shuffleStartingPoint

### 3.2.7 Flowchart Modul instantiateObject

Modul instantiateObject bertugas untuk memilih objek pukulan yang sesuai dengan karakter yang diterima pada modul instantiateObject. Pemilihan objek pukul dimulai dengan karakter '1' hingga '9'. Pemilihan menggunakan tipe data *char* karena chart menjadi asal karakter adalah dari sebuah string yang kemudian dikonversikan menjadi *array of char*. Sesuai dengan karakter yang diterima berikut adalah objek yang di-*generate* oleh aplikasi.

1. Pukul kanan
2. Pukul kiri
3. Sangkal kanan
4. Sangkal kiri
5. Tebang Kanan dari kanan
6. Tebang kanan dari kiri
7. Tebang kiri dari kanan
8. Tebang kiri dari kiri
9. Dobrakan

Objek yang telah dibuat kemudian akan diberikan waktu spawntime dan posisi selanjutnya yang sesuai dengan hasil random sebelumnya pada modul levelPreparation. Posisi hasil acakan tidak digunakan untuk objek dobrakan. Objek dobrakan menggunakan posisi awal dobrakan yang berbeda dengan yang lain. Gambar 3.6 merupakan gambar dari Flowchart Modul instantiateObject.

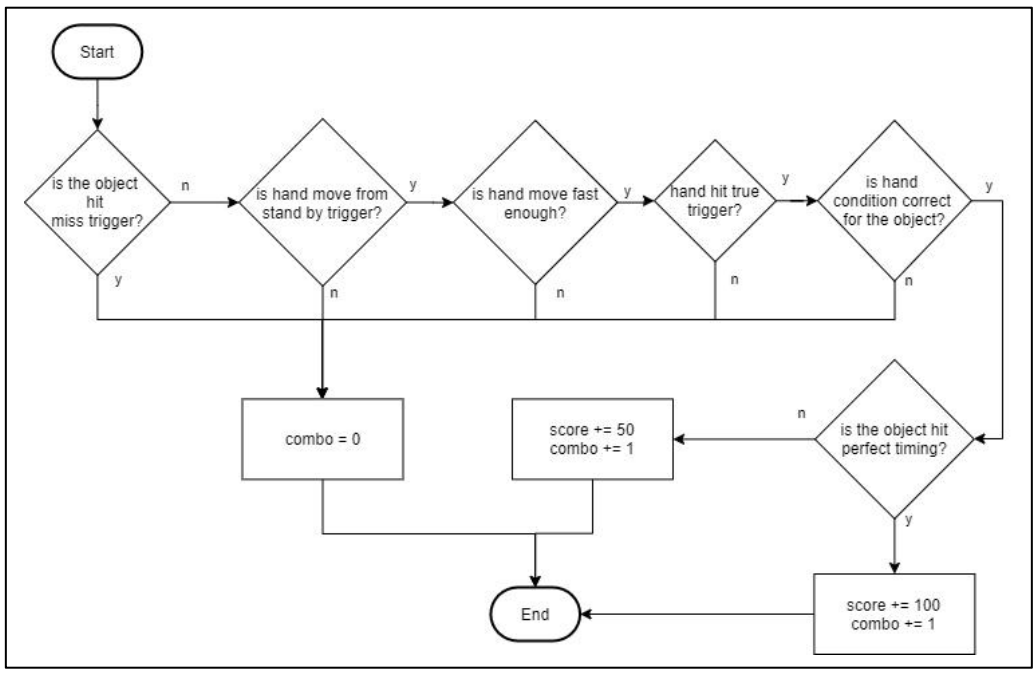



Gambar 3.6 Flowchart Modul instantiateObject

### 3.2.8 Flowchart Modul judgeScore

Modul judgeScore bertugas untuk mengecek nilai aksi yang telah pemain lakukan terhadap objek pukulan yang telah dikeluarkan. Pemain akan diberikan nilai secara penuh jika tangan pemain keluar dari zona *stand by* secara cepat. Jika tangan pemain sudah cukup cepat, tangan yang mengenai *true trigger* dengan kondisi tangan yang sesuai dengan objek pukulan yang bersangkutan adalah

pukulan yang benar. Jika objek yang dipukul dengan benar berada pada zona "*perfect timing"* maka akan diberikan nilai penuh dengan penambahan kombo pukulan sebanyak satu buah. Jika diluar *"perfect timing"*, diantaranya terlambat atau terlalu cepat, akan diberikan nilai sebanyak 50 dengan tetap pertambahan kombo. Jika salah satu dari kondisi pengecekan ada yang tidak benar, maka akan mendapatkan nilai miss atau gagal memukul objek pukulan. Jika terjadi demikian, kombo pukulan akan diset menjadi 0. Gambar 3.6 adalah gambar dari Flowchart modul judgeScore.




Gambar 3.7 Flowchart Modul judgeScore

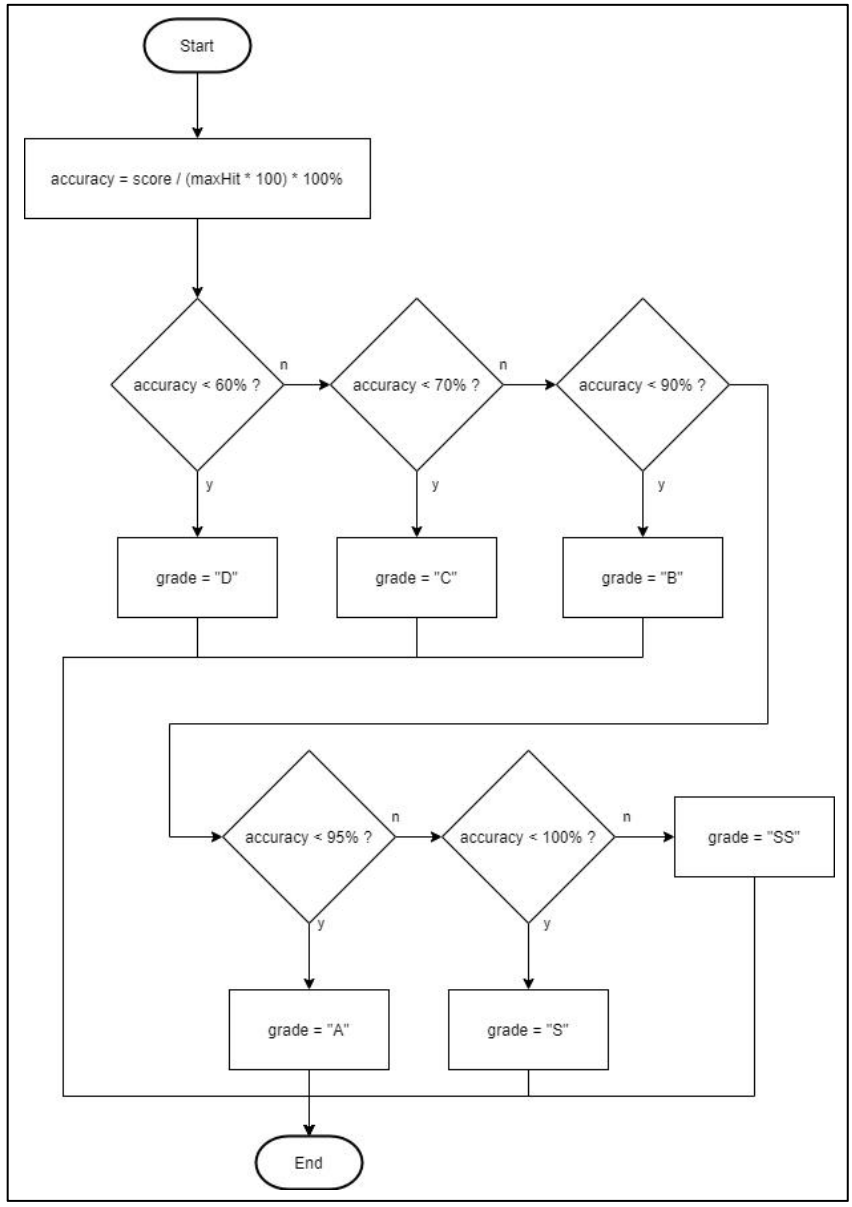
### 3.2.9 Flowchart Modul gradeAccuracy

Modul gradeAccuracy bertugas untuk mendapatkan grade berdasarkan score yang didapatkan dengan performa permainan pemain. Grade tertinggi yang dapat

dicapai adalah “SS” sedangkan yang terendah adalah “D”. Grade yang dapat dicapai pemain dapat dikelompokan dengan interval sebagai berikut.

1. 0% - 59.99% mendapatkan grade “D”
2. 60% - 69.99% mendapatkan grade “C”
3. 70% - 89.99% mendapatkan grade “B”
4. 90% - 94.99% mendapatkan grade “A”
5. 95% - 99.99% mendapatkan grade “S”
6. 100% mendapatkan grade “SS”

*Accuracy* didapatkan dengan membagi score pemain dengan hasil kali jumlah objek pukul dikalikan 100 dan kemudian dikalikan 100%. Accuracy yang didapatkan kemudian diklasifikasikan melalui interval yang sudah tersedia. Gambar 3.8 menggambarkan flowchart modul gradeAccuracy.

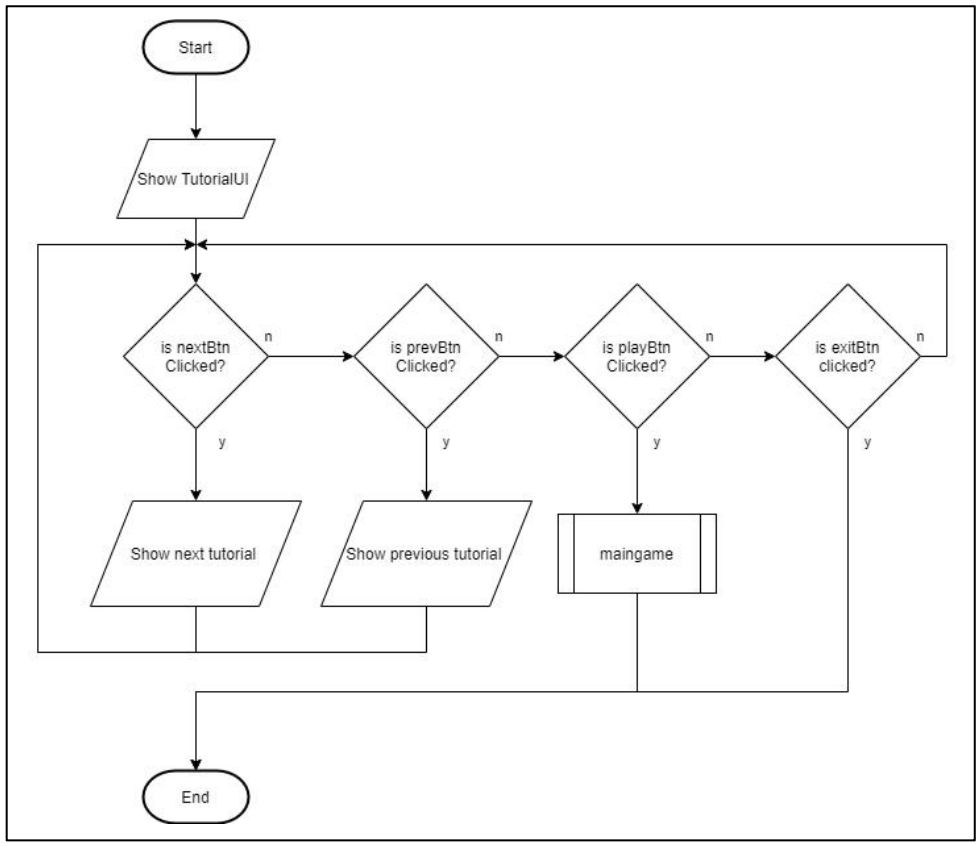



Gambar 3.8 Flowchart Modul gradeAccuracy

### 3.2.10 Flowchart Modul tutorial

Modul *tutorial* bertugas untuk mengendalikan halaman tutorial yang sedang ditampilkan. Jika pemain menekan tombol *next*, maka halaman tutorial akan diarahkan ke tutorial berikutnya sedangkan tombol *prev* yang ditekan, maka halaman tutorial akan dipindahkan ke halaman sebelumnya. Jika tombol *play*

ditekan maka pemain akan langsung menampilkan halaman trackList dan levelUI untuk memulai permainan. Pemain juga dapat menutup halaman tutorial dengan menekan tombol ‘x’ dan kembali ke mainmenu. Gambar 3.9 merupakan flowchart dari modul tutorial.




Gambar 3.9 Flowchart Modul Tutorial

### 3.3 Perancangan Antarmuka Aplikasi

*Game* yang dibuat membutuhkan sebuah proses perancangan untuk membangun aplikasi game tersebut. Perancangan aplikasi dimulai dengan perancangan dan peletakan antar muka, dilanjutkan dengan perancangan dan peletakan posisi lingkungan permainan yang dapat digunakan oleh pemain.

### 3.3.1 Rancangan Antarmuka HomeUI

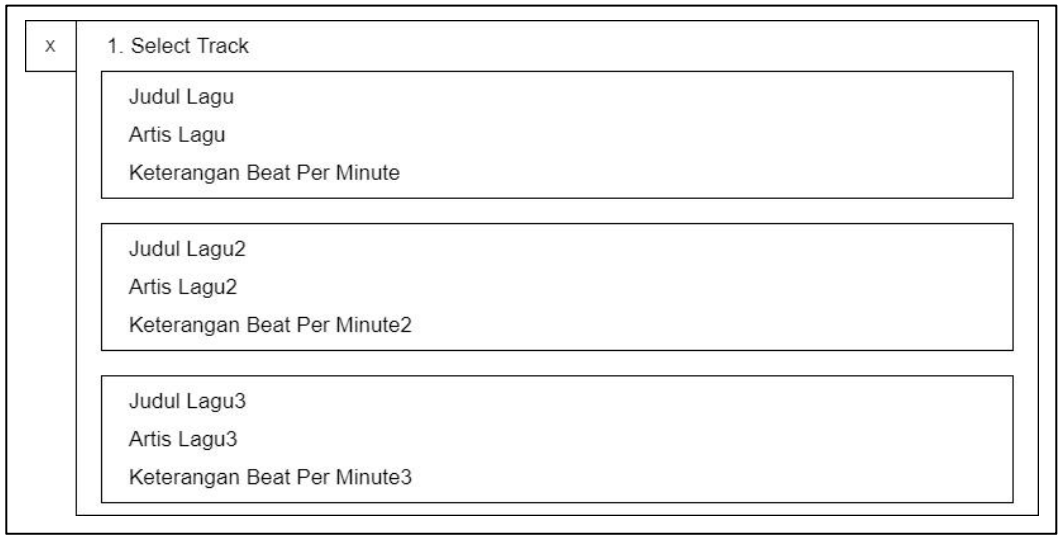
Antar muka Home berfungsi untuk mengawali permainan. Antar muka Home akan memiliki tombol *play* untuk masuk ke menu permainan, tombol *tutorial* untuk menampilkan cara bermain serta video singkat dari sumber gerakan ilmu seni bela diri pencak silat yang diimplementasikan pada permainan, tombol *credits* untuk halaman *credits*, dan tombol *exit* untuk mengakhiri permainan. Gambar 3.10 merupakan gambar dari rancangan antarmuka dari HomeUI.


Gambar 3.10 Rancangan Antar Muka HomeUI.

### 3.3.2 Rancangan Antarmuka TrackListUI

Antar muka TrackList berfungsi untuk memperbolehkan pemain untuk memilih lagu yang diinginkan. TrackList akan menampilkan judul, artis, dan BPM untuk lagu tersebut. Lagu-lagu akan digenerate secara otomatis sesuai

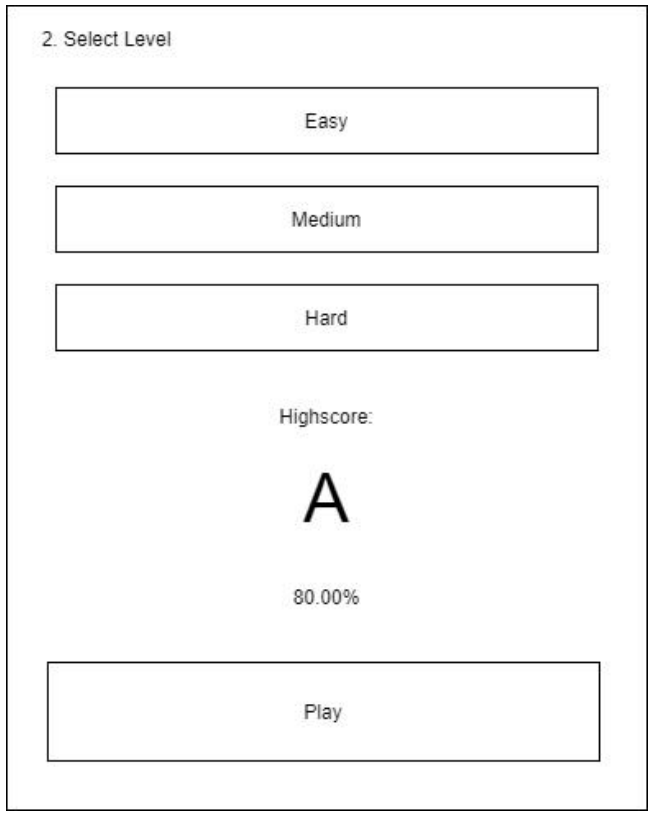
dengan list lagu yang telah dibuat. Gambar 3.11 merupakan rancangan antarmuka TrackListUI.




Gambar 3.11 Rancangan Antarmuka TrackListUI

### 3.3.3 Rancangan Antarmuka LevelUI

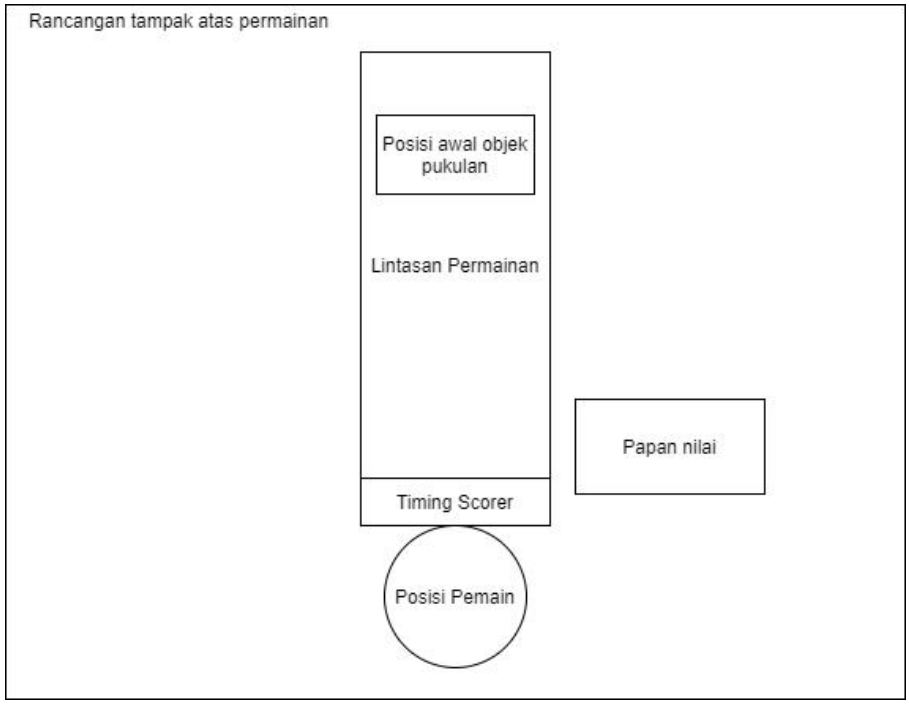
LevelUI berisi tiga level yang dapat dipilih oleh pemain. Level tersebut adalah *Easy*, *Medium*, dan *Hard*. Tingkat kesulitan dari permainan ditentukan dari pilihan level yang diinginkan oleh pemain. Pemain harus memilih level dan lagu terlebih dahulu untuk memulai permainan. Level UI juga akan menampilkan akurasi tertinggi yang telah dicapai pada lagu dan level tersebut. Gambar 3.12 adalah gambar dari rancangan antar muka dari LevelUI.




Gambar 3.12 Rancangan Antar Muka LevelUI

### 3.3.4 Permainan Tampak Atas

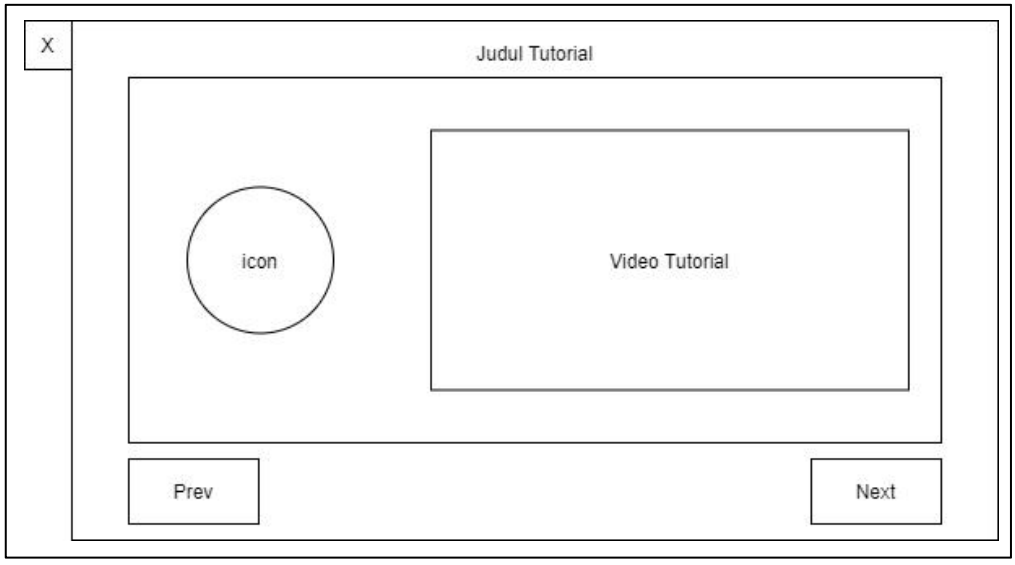
Pemain ditempatkan didepan lintasan permainan. Lintasan permainan adalah tempat objek-objek pukulan bertransisi. Objek pukulan akan memiliki titik-titik *spawn point* yang telah ditetapkan di atas lintasan tersebut. Spawn point akan diacak menggunakan metode *fisher-yates shuffle modern* sebagai titik awal dari objek pukulan. Pemain juga akan dihadapkan dengan sebuah "tembok" yang transparan yang berfungsi untuk menilai ketepatan waktu pukulan. Papan nilai juga berada disamping kanan lintasan yang berfungsi untuk memberikan informasi skor dan *combo* yang telah didapatkan oleh pemain. Gambar 3.13 merupakan gambar dari rancangan permainan tampak atas.




Gambar 3.13 Rancangan Permainan Tampak Atas

### 3.3.5 Rancangan Antarmuka Halaman Tutorial

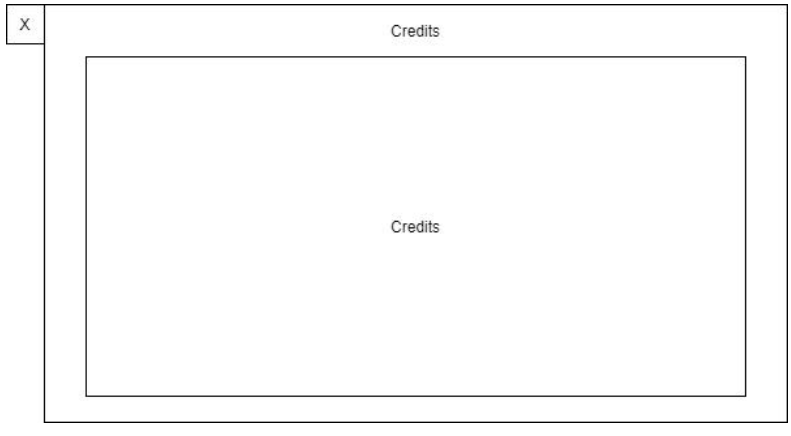
Halaman *tutorial* akan memiliki sebuah judul arahan bertujuan untuk memberikan pesan cara memainkan *game* langkah demi langkah. Halaman tutorial akan menampilkan satu buah *icon* dan sebuah *video* singkat bagaimana cara memukul yang tepat sesuai dengan *icon* yang bersangkutan. Sumber video akan disertakan pada halaman *credits* kepada pemilik *video* yang telah menjadi bagian dari *game* ini. Gambar 3.14 merupakan rancangan dari halaman *tutorial.*




Gambar 3.14 Rancangan Antarmuka Halaman Tutorial

### 3.3.6 Rancangan Antarmuka Halaman Credits

Halaman *credits* akan berfungsi untuk menampilkan pihak-pihak yang telah berkontribusi untuk pengembangan *game* ini. Halaman *credits* dapat diakses melalui tombol *credits* yang terletak pada mainmenuUI. Gambar 3.15 merupakan rancangan antarmuka halaman credits.




Gambar 3.15 Rancangan Antarmuka Halaman Credits

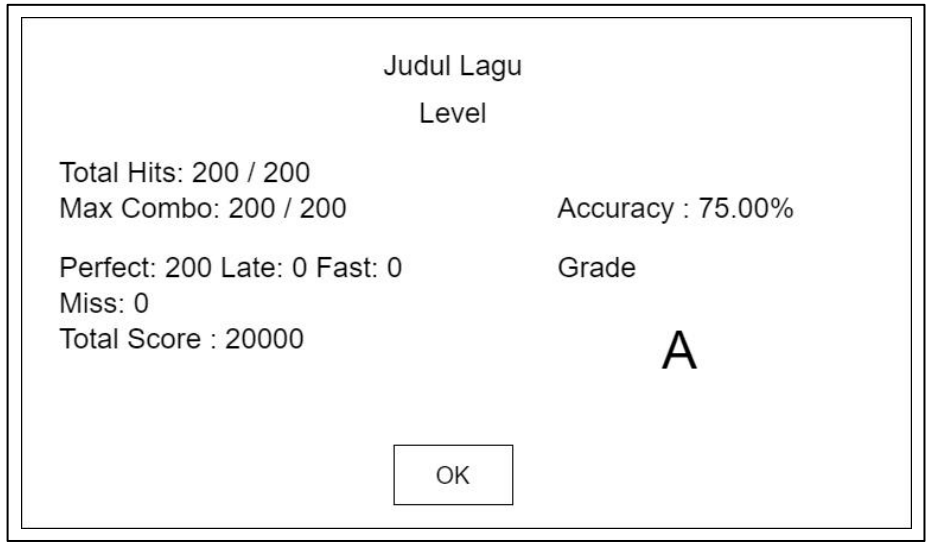
### 3.3.7 Rancangan Antarmuka Scoreboard

Scoreboard berfungsi untuk menampilkan jumlah skor dan kombo yang telah dihasilkan pemain. Scoreboard akan juga dapat menampilkan judul lagu yang sedang dimainkan. Gambar 3.16 menggambarkan rancangan antarmuka *scoreboard.*


Gambar 3.16 Rancangan Antarmuka Scoreboard

### 3.3.8 Rancangan Antarmuka Halaman Result

Halaman Result berguna untuk menampilkan ringkasan hasil permainan pada *level* tersebut. Halaman Result akan menampilkan jumlah pukulan yang tepat dan yang kurang, jumlah kombo maksimal yang didapatkan serta *grade* dari performa pemain saat itu. *Grade* permainan diperoleh dari perhitungan akurasi permainan. Gambar 3.17 merupakan gambar rancangan antarmuka Halaman Result.




Gambar 3.17 Rancangan Antarmuka Halaman Result

### 3.4 Perancangan Penggunaan Aset Permainan

Aset yang digunakan pada permainan adalah aset *free royalty* berasal dari berbagai sumber. Berikut adalah daftar aset yang digunakan dalam permainan dijelaskan pada tabel 3.1.

Tabel 3.1 Daftar Aset pada Permainan

| No | Nama | Fungsi | Sumber aset |
|---|---|---|---|
| 1 | Standard Asset Unity3d | *Terrain* dan e*nvironment* | Unity Asset Store |
| 2 | Oculus Integration | Integrasi alat oculus rift dan model tangan | Unity Asset Store |
| 3 | Block Brick | Objek Pukulan dan *particle effect* | Unity Asset Store |
| 4 | Wooden UI | Tampilan antar muka | Unity Asset Store |
| 5 | Training Table | Lintasan objek pukulan | Unity Asset Store |
| 6 | Vertical black board | *Scoreboard* permainan | Unity Asset Store |
| 7 | Wave - Jeff II | Lagu permainan | Youtube *free royalty* |

Tabel 3.1 Daftar Aset pada Permainan (lanjutan)

| No | Nama | Fungsi | Sumber aset |
|---|---|---|---|
| 8 | Hand icon | Petunjuk objek pukulan | Freepik, icon8, flaticon |
| 9 | Video teknik dasar pukulan pencak silat | Video tutorial untuk memperkenalkan seni ilmu bela diri pencak silat yang digunakan pada permainan | Youtube: https://www.youtube.com/watch?v=2g8egrJOcAw |
| 10 | Night Without Moon | Lagu permainan | Unity Asset Store |

### 3.5 Perancangan Kuesioner

Kuesioner akan berisikan pernyataan-pernyataan berdasarkan pemodelan metode HMSAM (Hedonic-Motivation System Adoption Model). Kuesioner ini akan membantu mengukur nilai *behavior intention to use* dan *immersion*. Pernyataan-pernyataan akan memiliki enam tingkat persetujuan. Enam tingkat persetujuan yang dapat dipilih oleh responden beserta skor pilihan tingkat persetujuan dijelaskan sebagai berikut.

a) 1 = Sangat tidak setuju, memiliki skor 1

b) 2 = Tidak setuju, memiliki skor 2

c) 3 = Sedikit tidak setuju, memiliki skor 3

d) 4 = Sedikit setuju, memiliki skor 4

e) 5 = Setuju, memiliki skor 5

f) 6 = Sangat setuju, memiliki skor 6

Beberapa butir pernyataan memiliki penilaian yang terbalik. Sehingga enam tingkat persetujuan yang dipilih oleh responden beserta skor pilihan tingkat persetujuan dijelaskan sebagai berikut

a) 1 = Sangat tidak setuju, memiliki skor 6

b) 2 = Tidak setuju, memiliki skor 5

c) 3 = Sedikit tidak setuju, memiliki skor 4

d) 4 = Sedikit setuju, memiliki skor 3

e) 5 = Setuju, memiliki skor 2

f) 6 = Sangat setuju, memiliki skor 1

Penggunaan jumlah tingkat persetujuan dengan angka genap digunakan untuk menghilangkan sikap netral responden terhadap sebuah pernyataan yang memaksakan responden untuk memilih kubu pro ataupun kontra dengan sebuah pernyataan. Hal ini berguna sebagai pengukuran apakah *game virtual reality* pengenalan pencak silat dengan metode *fisher-yates shuffle* berhasil menarik perhatian pemain atau tidak.

Kuesioner dibagi menjadi delapan bagian yang dimana merupakan bagian-bagian yang ada pada pemodelan HMSAM. Kedelapan bagian tersebut memiliki beberapa pernyataan-pernyataan yang mewakili tema dari bagian tersebut. Responden diharapkan dapat menjawab tingkat persetujuan akan pernyataan tersebut melalui enam tingkat persetujuan yang telah disiapkan.

Interval interpretasi persetujuan responden dapat dibagi menjadi enam bagian. Interval yang didapatkan berada pada nilai 16.67%. Berikut adalah penjelasan mengenai interval interpretasi persetujuan responden.

a) 0.00% - 16.66% = Sangat tidak setuju

b) 16.67% - 33.32% = Tidak setuju

c) 33.33% - 50.00% = Sedikit tidak setuju

d) 50.01% - 66.67% = Sedikit setuju

e) 66.68% - 83.34% = Setuju

f) 83.35% - 100.00% = Sangat setuju