

BAB III

METODOLOGI PENELITIAN

3.2 Metodologi Penelitian

Metodologi penelitian yang akan digunakan dalam perancangan *game* ini adalah sebagai berikut.

1. Studi Literatur

Pada tahap ini, peneliti melakukan studi literatur dengan cara mencari, mempelajari, dan membaca penelitian-penelitian terdahulu yang bersumber dari jurnal ilmiah, karya tulis, dan artikel. Studi literatur dilakukan pada bidang yang terkait dengan *fighting game*, *game design*, *game architecture*, *game design document*, GUESS, unity, *online machine learning*, *artificial noedify*, dan *Adaptive AI*.

2. Rancangan Sistem

Pada tahap ini, akan dilakukan perancangan terhadap sistem yang akan dibangun, misalnya *game environment*, algoritma *learning* pada *agent*, dan sebagainya.

3. Pemrograman Sistem

Pada tahap ini akan dilakukan implementasi program berdasarkan rancangan yang telah dibuat pada tahap sebelumnya. Implementasi akan dilakukan menggunakan Unity untuk membangun *game enviromenment* dan *training* pada *agent* dengan menggunakan noedify pada saat *runtime*.

4. Pengujian

Pada tahap ini akan dilakukan pengujian terhadap game yang telah dibuat. Proses pengujian akan dilakukan untuk mengukur tingkat kepuasan pemain terhadap *game* yang dibuat dan mengukur performa *adaptive AI*. Pengukuran tingkat kepuasan pemain akan dilakukan dengan melakukan *playtesting* pada *game* yang telah dibuat terhadap responden yang dipilih secara acak. Responden yang telah memainkan game kemudian akan diberikan kuesioner GUESS (*Game User Experience Satisfaction Scale*), menggunakan skala *likert* 7 poin (1 = sangat tidak setuju, 5 = cukup setuju, 7 = sangat setuju). Pengukuran performa *adaptive AI* akan dilakukan dengan mempertandingkan *adaptive AI* yang telah dibangun dengan *static AI* yang dibuat dengan menggunakan *finite state machine* (FSM).

5. Evaluasi

Pada tahap ini, akan dilakukan evaluasi terhadap data-data yang diperoleh dari pengisian kuesioner GUESS dan hasil pertandingan antara *adaptive AI* dengan *static AI* yang telah dilakukan pada tahap pengujian. Hasil dari evaluasi kuesioner GUESS akan digunakan untuk mengukur tingkat kepuasan responden terhadap game yang telah dibuat. Hasil pertandingan antara *adaptive AI* dengan *static AI* akan digunakan untuk mengukur performa *adaptive AI*.

3.3 Perancangan Sistem

Perancangan sistem dari *game* yang akan dibangun pada penelitian ini adalah sebagai berikut.

3.2.1 Game Design Document

Judul: Extreme FighterZ

Berikut adalah elemen-elemen dalam *game* Extreme FighterZ.

1. Characters:

a. Billy Lee

Billy Lee adalah ketua geng Blue Falcon yang menguasai area Barat dari kota New York. Billy Lee menggunakan gaya bertarung Xin Fu Xiao yang dipelajarinya dari seorang pertapa saat dirinya berkelana di China. Setelah menguasai teknik Xin Fu Xiao, Billy Lee ingin mencoba kekuatan yang dimilikinya. Ia memutuskan untuk pulang ke kampung halamannya di New York dan menjadi yang terkuat disana dengan mengalahkan semua geng yang berada di kota New York.

b. Bad Dude

Bad Dude adalah ketua geng Black Wolf yang menguasai area Timur dari kota New York. Bad Dude menggunakan gaya bertarung *street fighting* yang dipelajarinya sendiri selama masa kecilnya yang berat hidup di jalanan. Sepanjang masa hidupnya ia hanya peduli untuk menjadi yang terkuat. Hal ini dikarenakan dengan kepercayaannya bahwa yang lemah tidak pantas untuk hidup.

2. Story

Kota New York adalah kota yang dipenuhi dengan gerombolan geng ganas. Diantara semua geng yang menguasai kota New York, terdapat dua geng terkuat yang saling bersaing untuk menjadi penguasa terbesar di kota New York. Kedua geng tersebut ialah Blue Falcon yang menguasai area Barat dan Black Wolf yang menguasai area Timur.

Karena pertarungan sengit yang dilakukan oleh Blue Falcon dan Black Wolf untuk menguasai kota New York tidak pernah usai, maka diputuskan untuk diadakan pertarungan bernama Extreme FighterZ untuk menentukan siapa yang akan menjadi pemimpin kota New York. Pertarungan diadakan setiap tahun dan dilakukan oleh seorang perwakilan dari Blue Falcon dan Black Wolf. Yang mewakili Blue Falcon adalah Billy Lee dan yang menjadi perwakilan Black Wolf adalah Bad Dude. Pertarungan yang akan menentukan nasib kota New York akan dimulai sekarang!!!!!!

3. Gameplay

a. Goals

Memenangkan pertarungan dengan menghabiskan health point musuh sebanyak tiga kali.

b. Skills

Pemain menggunakan mouse untuk memilih menu dan menggunakan keyboard untuk mengontrol pergerakan karakter dalam *game*. Aksi yang dapat dilakukan karakter dalam *game* yang dibuat dapat dilihat pada tabel 3.1

Tabel 3.1 Aksi Pemain

Tombol	Aksi
Panah kiri	Bergerak ke kiri
Panah kanan	Bergerak ke kanan
Panah atas	Lompat
D	Serangan normal
S	Bertahan
A	Serangan berat
F	Serangan jarak jauh
D saat lompat	<i>Jump attack</i>
D saat terjatuh	<i>Grounded attack</i>

c. Game Mechanics

Game ini merupakan *fighting game* dua dimensi. Untuk memenangkan *game* ini, pemain harus memenangkan pertandingan sebanyak tiga kali sebelum musuh dapat memenangkan pertandingan sebanyak tiga kali. Untuk memenangkan pertandingan, pemain harus dapat mengurangi *health point* yang dimiliki musuh menjadi nol.

Dalam *game* ini, pemain dapat melakukan aksi berupa bergerak, serangan normal, serangan berat, serangan jarak jauh, bertahan, *jump attack*, dan *ground attack*. Berikut adalah keterangan dari beberapa aksi yang dapat dilakukan.

i. Serangan normal

Serangan normal dapat dilakukan sebanyak tiga kali beruntun untuk melakukan kombo. Serangan normal memiliki *damage* yang

lebih kecil dari pada serangan berat. Namun, serangan normal lebih cepat daripada serangan berat.

ii. Serangan berat

Serangan berat memiliki *damage* yang lebih besar dari pada serangan normal. Namun, serangan normal lebih lambat daripada serangan normal. Serangan berat dapat menembus gerakan bertahan.

iii. Bertahan

Dapat menahan semua jenis serangan kecuali serangan berat.

iv. Serangan jarak jauh

Serangan jarak jauh dapat digunakan dengan menggunakan energi. Setiap memulai ronde, energi dan *health point* akan diatur menjadi 100. Jumlah energi yang diperlukan untuk melakukan serangan jarak jauh bergantung pada karakter yang sedang digunakan.

d. Losing

Pemain akan kalah saat lawan dapat mengurangi *health point* dari pemain terlebih dahulu

e. Adaptive AI Implementation

Setiap akhir dari ronde, *adaptive AI* akan di-*training* menggunakan *plugin* noedify berdasarkan dari data *training set* yang didapatkan pada ronde sebelumnya.

4. Art Style

Berikut adalah asset yang digunakan dalam pengembangan *game* Extreme FighterZ.

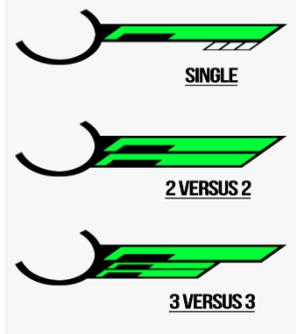
Tabel 3.2 Game Assets

Jenis Asset	Gambar	Nama	Author	Sumber
Sprite Sheets		billylee	G.M.Spectre	http://maximoff.alreadyread.net/SpriteSheets/

Tabel 3.3 Game Assets (Lanjutan)

Jenis Asset	Gambar	Nama	Author	Sumber
Sprite Sheets		baddude	G.M.Spectre	http://maximoff.alreadyread.net/SpriteSheets/
Image		bgArena2	J	https://eric9.artstation.com/projects/aA110

Tabel 3.4 Game Assets (Lanjutan)

Jenis Asset	Gambar	Nama	Author	Sumber
Sprite		myHealth	Friendfisher	https://www.kindpng.com/imgv/ToTmTbR_street-fighter-5-health-bar-png-fighting-game/
Sprite Sheets		Projectile	Bongwater-bandit	https://www.deviantart.com/bongwater-bandit/art/sprite-fx-projectiles-587001178
Font	FIGHTING SPIRIT	Fighting Spirit 2	PressGang Studios	https://www.dafont.com/fighting-spirit-tbs.font
Font	SINGLE FIGHTER	Single Fighter	Subectype	https://www.dafont.com/single-fighter.font
Font	KEMCO PIXEL	Kemco Pixel	Jayvee D. Enaguas	https://www.dafont.com/kemco-pixel.font
Image		UI_Long_Strip_Decay	SpeedTutor	https://assetstore.unity.com/publishers/45609

5. Music and Sound

Berikut adalah asset *background music* dan *sound effect* yang digunakan dalam pengembangan *game* Extreme FighterZ.

Tabel 3.5 Audio Asset

Nama	Type	Author	Sumber
buttonHover	sfx	volkbutphan	https://freesound.org/people/volkbutphan/sounds/485486/
buttonClick	sfx	-	https://www.zapsplat.com/music/button-tick/
Jump	sfx	Felixyadomi	https://freesound.org/people/felixyadomi/sounds/456375/
Landing	sfx	Zepurple	https://freesound.org/people/zepurple/sounds/540272/
Swing	sfx	-	https://www.sfxbuzz.com/download/7-free-fight-sounds-sound-effects/111-swing-sound-effect-for-fight-sfx-464499_elynych0901_punch-kick-being-blocked-2
Block	sfx	Elynch0901	
Walking	sfx	SpliceSound	https://freesound.org/people/SpliceSound/sounds/170505/
Picking	sfx	-	https://www.zapsplat.com/music/game-tone-answer-selected-locked-in-bright-chime-1/
changeSelection	sfx	-	https://www.zapsplat.com/music/game-menu-item-select-2/
explode1	sfx	-	https://www.zapsplat.com/music/short-small-explosion-with-no-tail-3/
explode2	sfx	-	https://www.zapsplat.com/music/short-small-explosion-with-no-tail-4/

Tabel 3.6 Audio Asset (Lanjutan)

Nama	Type	Author	Sumber
Theme1	bgm	-	http://soundimage.org/wp-content/uploads/2016/03/Street-Mayhem_Looping.mp3
Theme2	bgm	-	http://soundimage.org/wp-content/uploads/2016/07/Retro-Frantic_V001_Looping.mp3
Theme3	bgm	Steve Oxen	https://www.fesliyanstudios.com/royalty-free-music/download/too-much-funk/648

Asset audio yang tidak terdapat dalam tabel 3.3 merupakan asset yang dibuat sendiri oleh penulis menggunakan aplikasi perekam suara.

6. Technical Description

a) Platform

Game Extreme FighterZ dibuat untuk platform komputer menggunakan sistem operasi Windows.

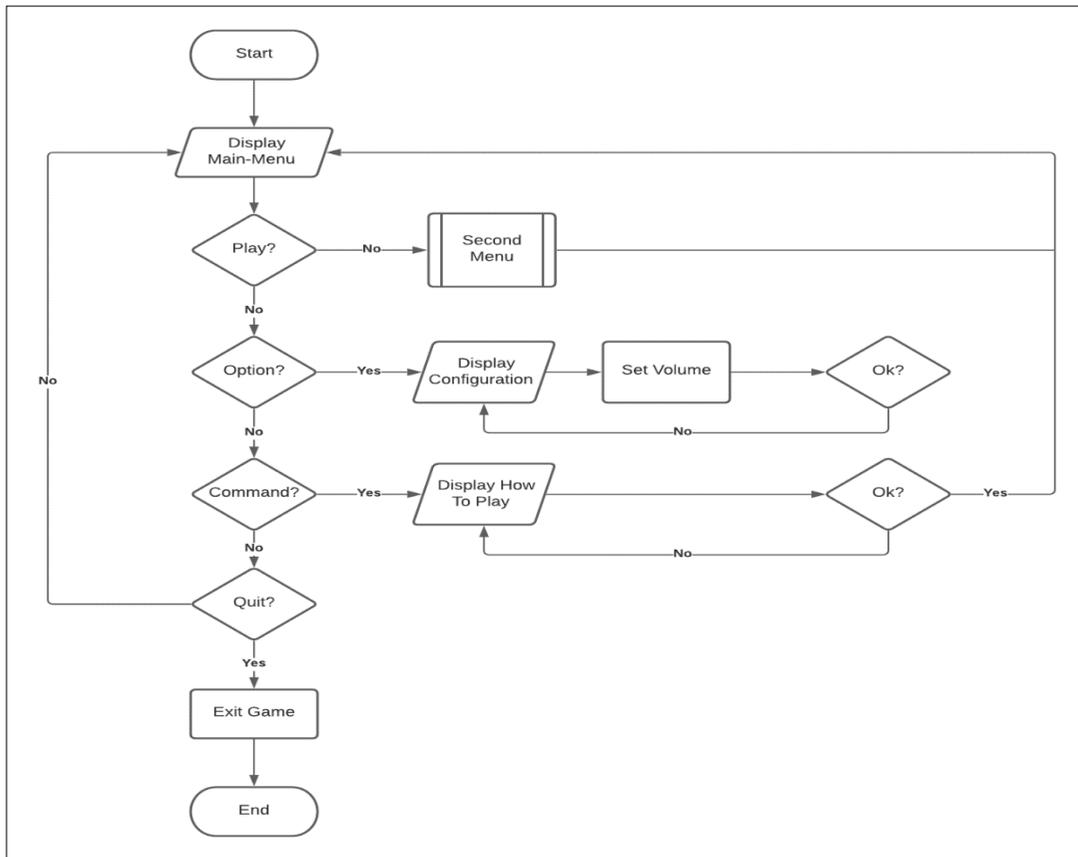
b) Tools

Game Extreme FighterZ dibuat menggunakan *game engine* Unity.

Dalam pembuata *adaptive AI*, digunakan *plugin* bernama Noedify.

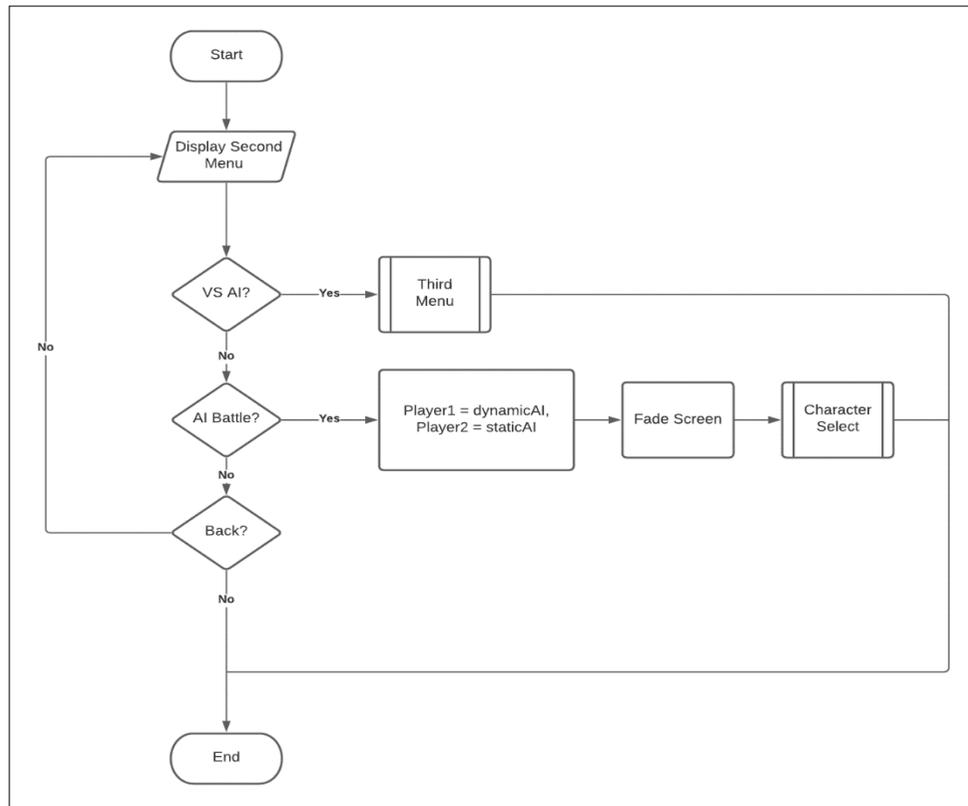
3.2.2 Flowchart

Berikut adalah tampilan desain *flowchart* yang digunakan dalam pengembangan *game* Extreme FighterZ. Dimulai dari *flowchart main-menu* yang dapat dilihat pada gambar 3.1



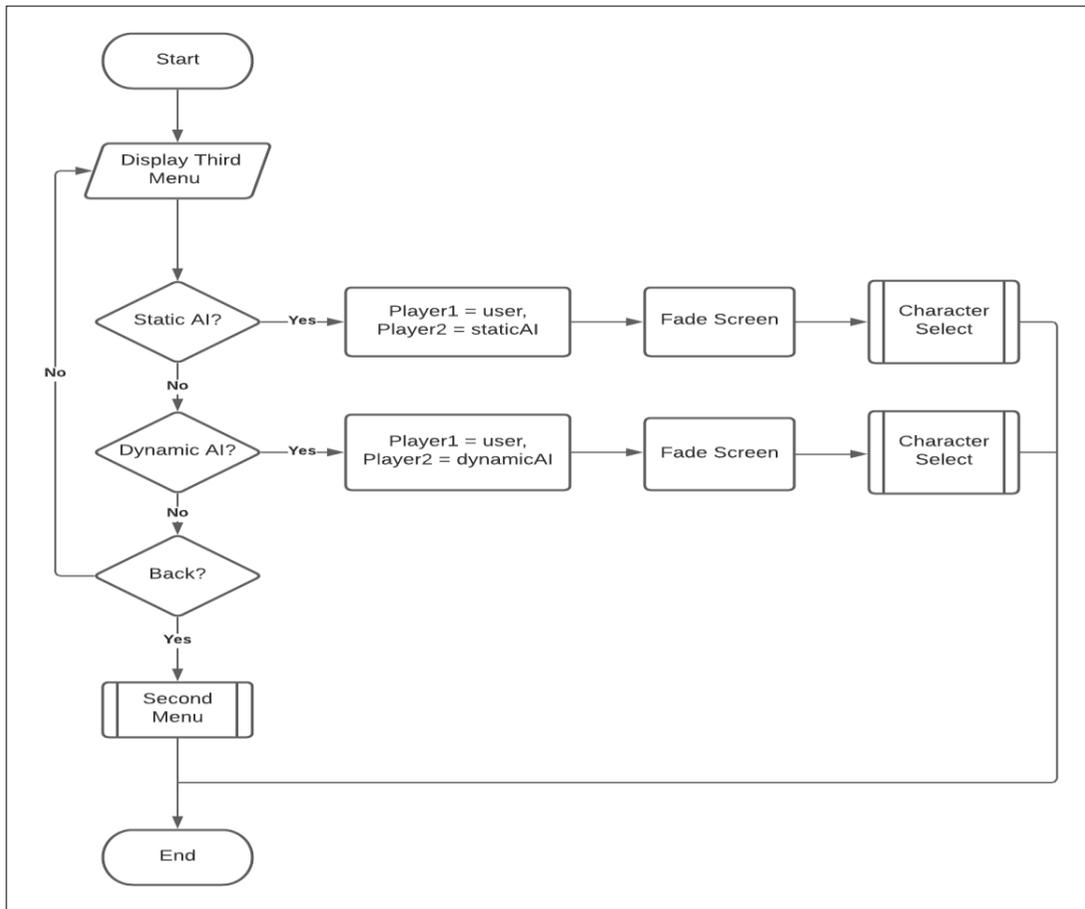
Gambar 3.1 Flowchart Main Menu

Gambar 3.1 menunjukkan alur *main-menu* dari game Extreme FighterZ. Pada bagian *main-menu*, terdapat empat menu yang dapat dipilih oleh pemain, yaitu *Play*, *Option*, *Command*, dan *Quit*. Jika pemain memilih menu *Play*, akan muncul tampilan *second menu* dengan alur *flowchart* yang dapat dilihat pada gambar 3.2. Jika pemain memilih menu *Option*, akan muncul tampilan *configuration* di mana pemain dapat mengatur volume musik dan *sound effect* pada *game*. Jika pemain memilih menu *Command*, akan muncul tampilan *command* yang akan menunjukkan semua aksi yang dapat dilakukan oleh pemain dalam *game*. Terakhir, jika pemain memilih menu *Quit*, maka pemain akan keluar dari *game* Extreme FighterZ.



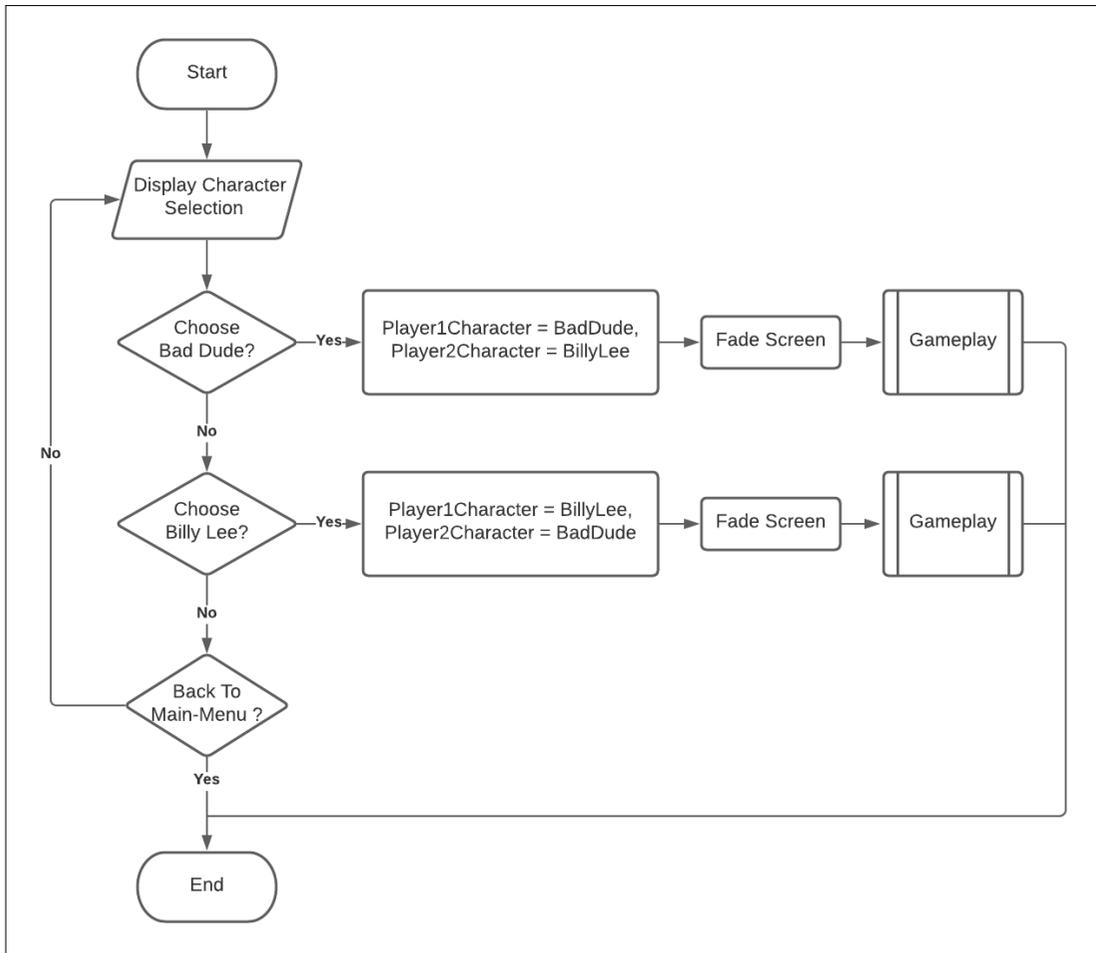
Gambar 3.2 Flowchart Second Menu

Gambar 3.2 menunjukkan alur *second menu* dari game Extreme FighterZ. *Second menu* adalah tampilan *menu* yang muncul saat pemain memilih *menu play* pada bagian *main-menu*. Pada bagian *second menu*, terdapat tiga pilihan menu yang dapat dipilih oleh pemain, yaitu *VS AI*, *AI Battle*, dan *back*. Jika pemain memilih menu *VS AI*, akan muncul tampilan *third menu* dengan alur yang dapat dilihat pada gambar 3.3. Jika pemain memilih menu *AI Battle*, maka pemain pertama akan diatur menjadi *dynamic AI/adaptive AI* dan pemain kedua akan diatur menjadi *static AI*, kemudian pemain akan berpindah menuju halaman *character*. Fungsi dari menu *AI Battle* adalah agar pemain dapat melihat pertarungan antara *adaptive AI* dengan *static AI*. Terakhir jika pemain memilih *menu Back*, maka pemain akan kembali menuju *main-menu*.



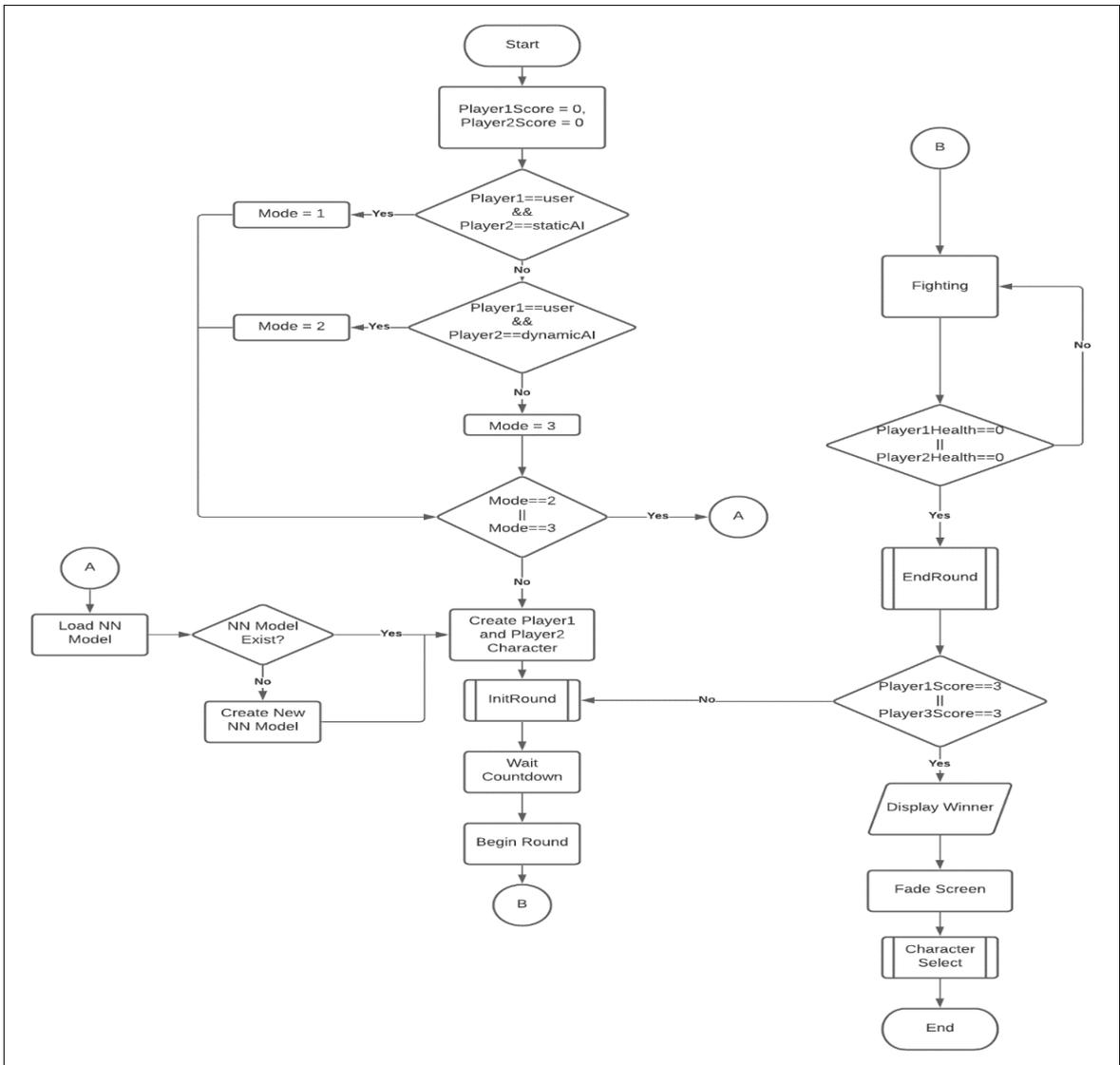
Gambar 3.3 Flowchart Third Menu

Gambar 3.3 menunjukkan alur *third menu* dari game *Extreme FighterZ*. *Third menu* merupakan tampilan *menu* yang muncul saat pemain memilih *menu VS AI* pada bagian *second menu*. Pada bagian *third menu* terdapat tiga pilihan yang dapat dipilih oleh pemain, yaitu *Static AI*, *Dynamic AI*, dan *Back*. Jika pemain memilih *menu Static AI*, tipe *AI* lawan akan diatur menjadi *static AI* dan pemain akan berpindah menuju halaman *character selection*. Jika pemain memilih *menu Dynamic AI*, tipe *AI* lawan akan diatur menjadi *adaptive AI* dan pemain akan berpindah menuju halaman *character selection*. Terakhir, jika pemain memilih *menu Back*, maka pemain akan kembali menuju *second menu*.



Gambar 3.4 Flowchart Character Selection

Gambar 3.4 menunjukkan alur halaman *character selection* dari game Extreme FighterZ. Pada halaman *character selection* pemain dapat memilih salah satu dari dua karakter yang ada pada game. Saat pemain telah memilih salah satu karakter, maka karakter yang tidak dipilih oleh pemain akan secara otomatis digunakan oleh lawan. Setelah pemilihan karakter selesai, pemain akan berpindah ke halaman *gameplay* untuk memainkan permainan. Jika pemain ingin kembali menuju *main-menu*, pada halaman *character selection* terdapat tombol Back To Main-Menu yang dapat digunakan untuk kembali menuju *main-menu*,



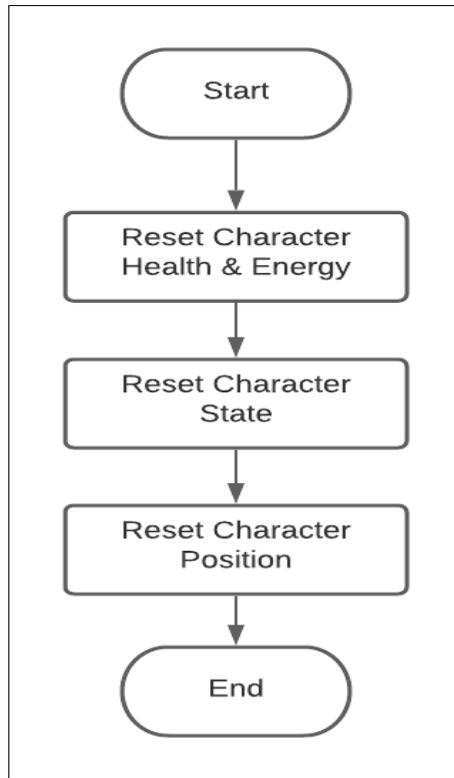
Gambar 3.5 Flowchart Gameplay

Gambar 3.5 menunjukkan alur *gameplay* dari *game* Extreme FighterZ. Saat pemain masuk ke halaman *gameplay*, sistem akan menentukan mode permainan dengan mengecek tipe pemain satu dan pemain dua. Jika mode permainan adalah mode dua atau mode tiga, berarti salah satu pemain merupakan *adaptive AI*. Oleh karena, itu sistem akan melakukan *load* terhadap model *neural network* yang telah dibuat untuk digunakan oleh *adaptive AI*. Jika proses *load model* gagal, maka sistem akan membuat model *neural network* yang baru.

Model *neural network* yang digunakan pada *game* Extreme FighterZ dibuat dengan menggunakan *plugin* noedify. Struktur dari model *neural network* yang dibuat terdiri dari sebuah *input layer* yang menerima 17 jenis *input* yang didapat dari *game environment*, sebuah *output layer* yang mengeluarkan 10 jenis *output* untuk menentukan aksi dari *adaptive AI*, dan sebuah *hidden layer*.

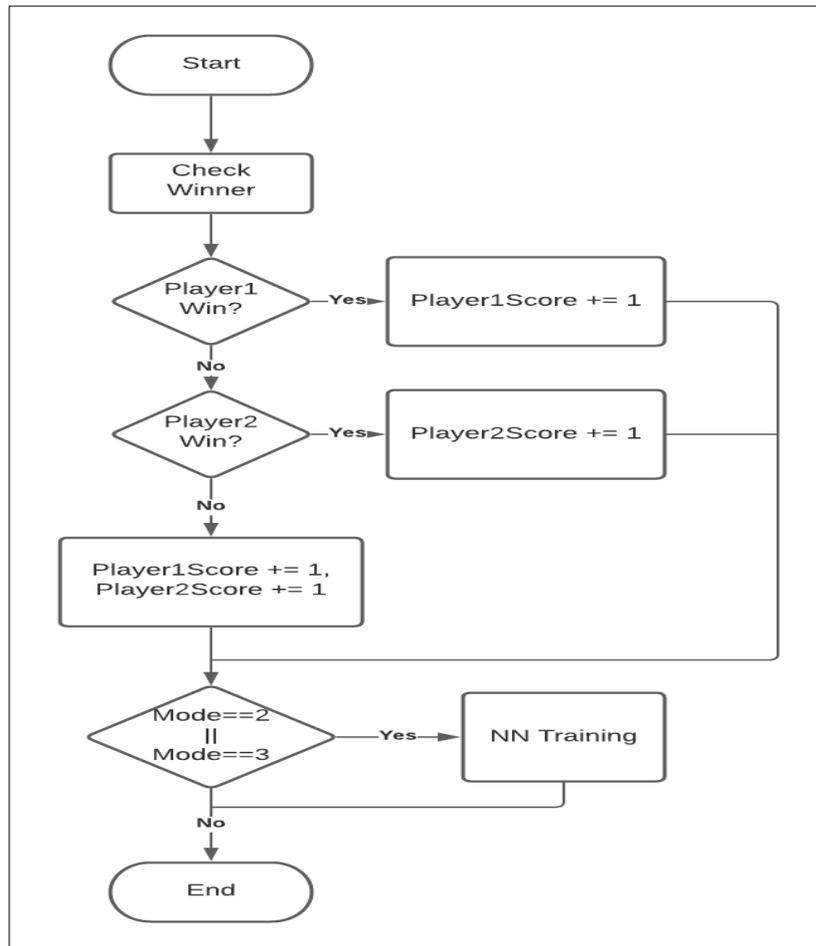
Setelah sistem menentukan mode permainan, sistem akan *men-spawn* karakter yang telah dipilih oleh pemain pada *spawn position* yang telah ditentukan. Sebelum memulai setiap ronde, akan dilakukan proses *initRound* yang akan dijelaskan pada gambar 3.6. Setelah proses *initRound* selesai dilakukan hitungan mundur yang dimulai dari tiga sebelum setiap ronde dimulai. Saat proses hitungan mundur selesai, ronde akan dimulai dan pemain dapat memulai proses pertarungan.

Selama proses pertarungan berjalan, sistem akan mengecek status *health point* dari pemain satu dan pemain dua. Jika status *health point* pemain satu atau pemain dua mencapai nol, ronde tersebut akan berakhir dan akan dilakukan proses *endRound* yang akan dijelaskan pada gambar 3.7. Setelah proses *endRound* selesai dijalankan, sistem akan mengecek *score* dari pemain satu dan pemain dua. Jika *score* dari pemain satu atau pemain dua ada yang mencapai tiga, maka pertandingan akan dianggap selesai, kemudian sistem akan menampilkan nama pemenang dan kembali ke halaman *character selection*. Jika *score* pemain satu dan pemain dua belum ada yang mencapai tiga, maka proses *initRound* akan dilakukan kembali untuk memulai ronde selanjutnya.



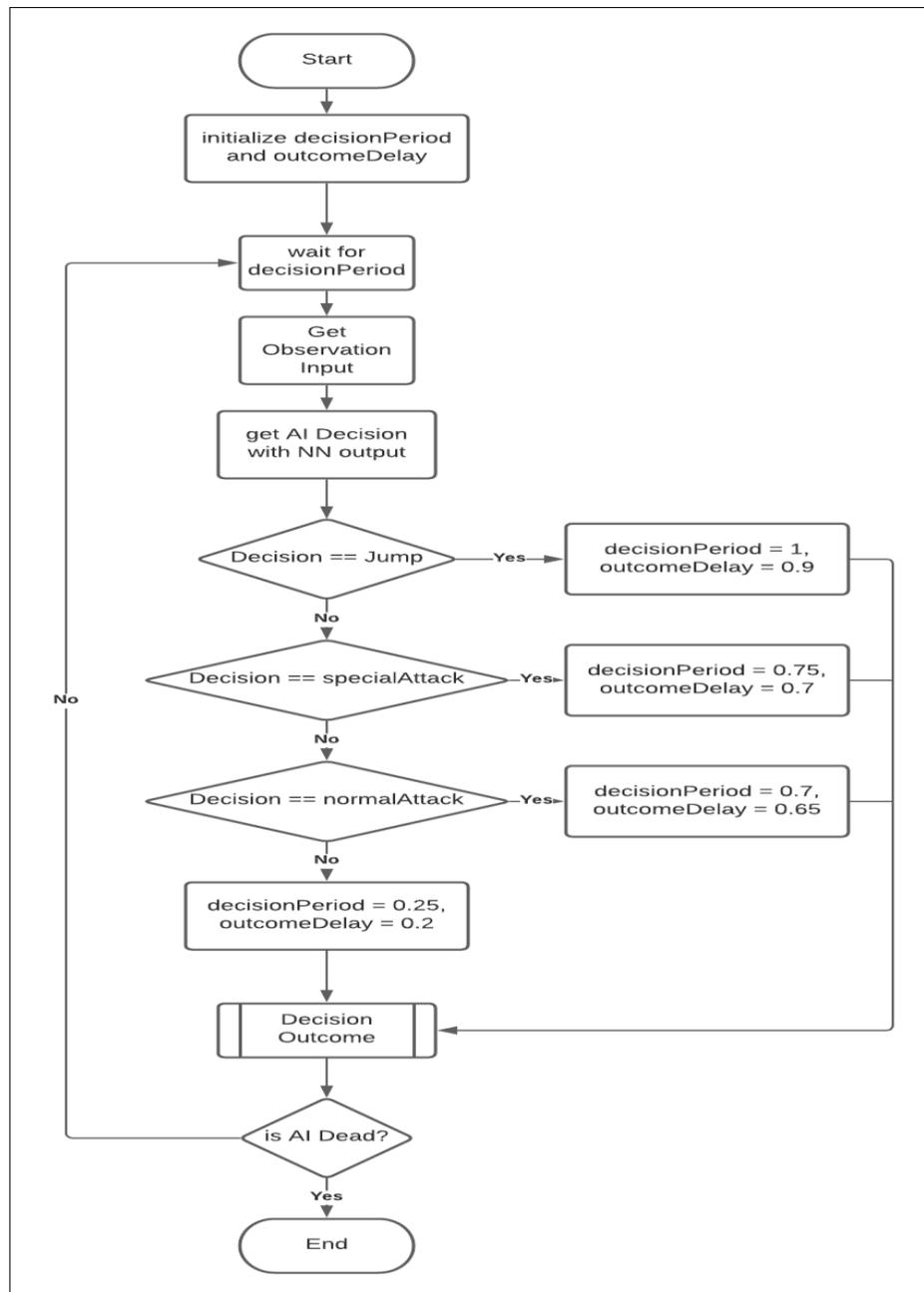
Gambar 3.6 Flowchart initRound

Gambar 3.6 merupakan alur dari proses `initRound`. Pada proses `initRound`, status *health point* dan *energy* yang dimiliki pemain satu dan pemain dua akan di set kembali menjadi 100. Kemudian posisi dari pemain satu dan pemain dua akan di-*reset* menuju *spawn position* yang telah ditentukan.



Gambar 3.7 Flowchart endRound

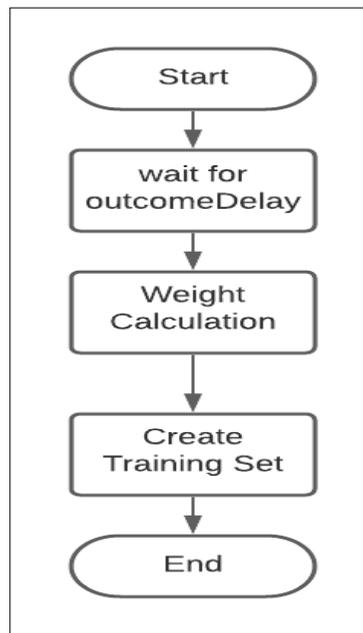
Gambar 3.7 merupakan alur dari proses endRound. Pada proses endRound, sistem akan mengecek pemenang dari ronde yang baru saja berakhir. Setelah sistem menentukan pemenang dari ronde tersebut, sistem akan menambah satu poin pada *score* pemegang ronde tersebut. Jika ronde tersebut dianggap seri, sistem akan memberikan satu poin kepada kedua pemain. Setelah proses pemberian poin selesai, sistem akan mengecek mode dari permainan. Jika mode permainan adalah mode dua atau mode tiga, maka akan dilaksanakan proses *training* pada *adaptive AI* menggunakan *training set* yang diperoleh pada ronde tersebut.



Gambar 3.8 Flowchart Adaptive AI Behaviour

Gambar 3.8 merupakan alur dari *adaptive AI behaviour* yang menunjukkan proses pengambilan keputusan yang dilakukan oleh *adaptive AI*. Setiap kali *adaptive AI* akan mengambil *decision* baru, *adaptive AI* akan menunggu selama *decisionPeriod* yang telah ditentukan. Setelah mencapai *decisionPeriod*, *adaptive*

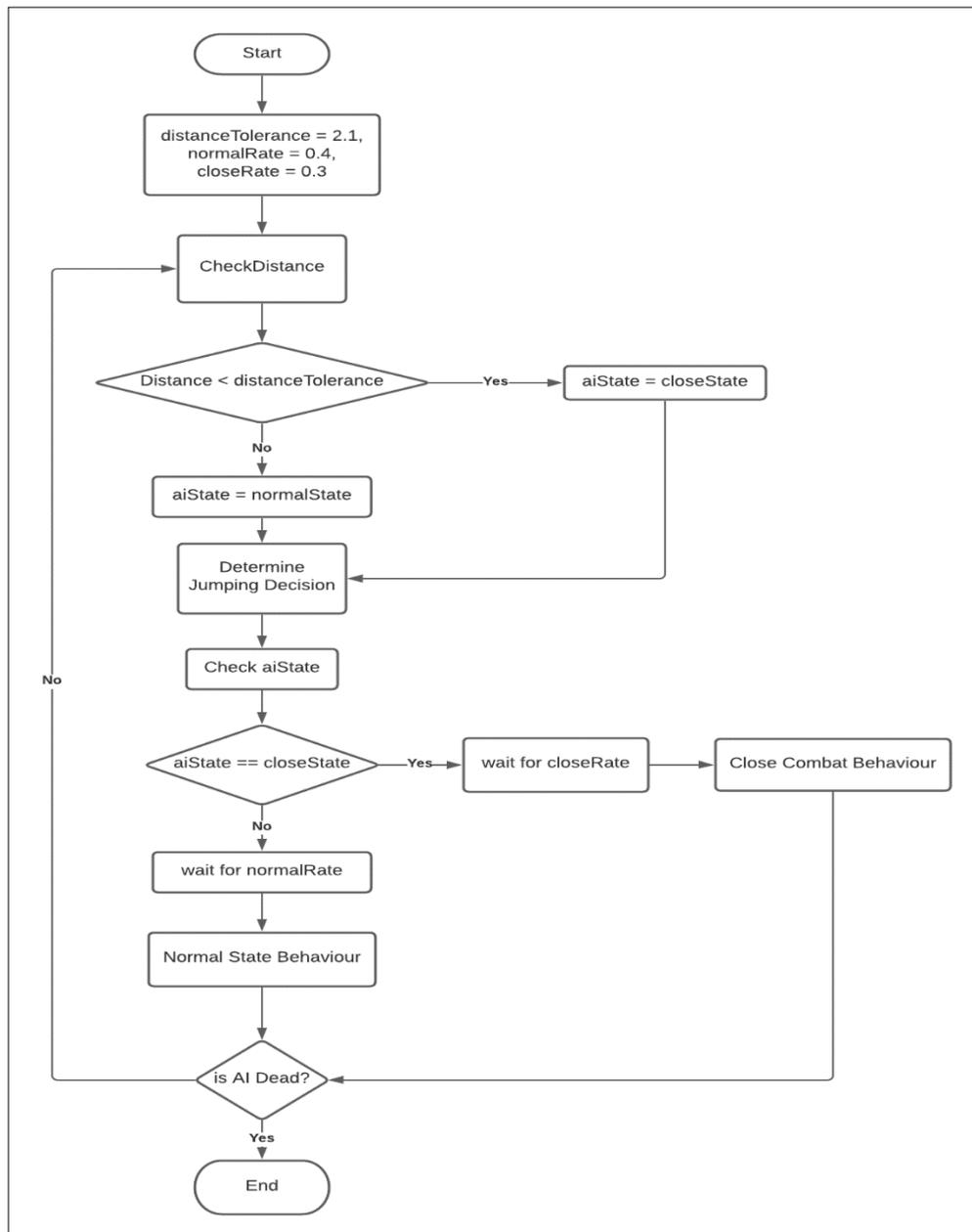
AI akan mengambil *decision* baru berdasarkan nilai perhitungan dari model *neural network*. Berdasarkan keputusan yang diambil, nilai dari *decisionPeriod* dan *outcomeDelay* akan dirubah. Setelah proses pengambilan *decision* selesai, *decision* yang diambil akan diberi *weight* dan dijadikan *training set* baru melalui proses *Decision Outcome* yang akan dijelaskan pada gambar 3.9. Proses ini akan berulang terus hingga pertandingan berakhir atau AI mati.



Gambar 3.9 Flowchart Decision Outcome

Gambar 3.9 merupakan alur dari proses *Decision Outcome*. Proses *Decision Outcome* merupakan proses yang dilakukan untuk memberikan *weight* pada *decision* yang dilakukan *adaptive AI* dan membentuk *training set* berdasarkan data *input*, *decision*, dan *weight* tersebut. Setiap sebelum memulai proses, sistem akan menunggu selama *outcomeDelay* yang telah ditentukan. Setelah mencapai *outcomeDelay*, sistem akan melakukan kalkulasi untuk menghitung *weight* yang

akan diberikan pada *decision* yang telah diambil berdasarkan dengan nilai yang telah disediakan. Setelah *weight* selesai dikalkulasi, *weight* akan di *clamp* dengan batas nol dan satu. Setelah semua proses selesai, akan dibuat *training set* berdasarkan *input*, *output*, dan *weight* yang telah ditentukan.



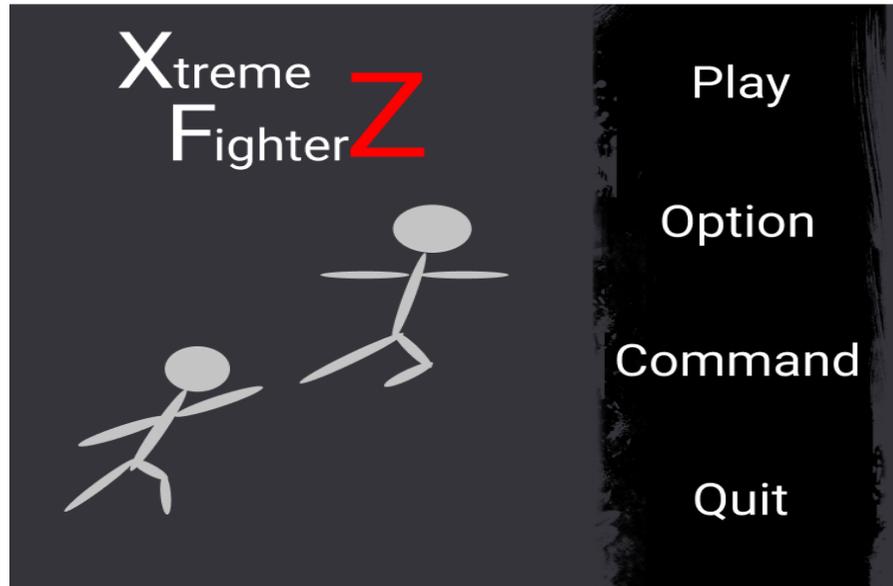
Gambar 3.10 Flowchart Static AI Behaviour

Gambar 3.10 merupakan alur dari *static AI behaviour* yang menunjukkan proses pengambilan keputusan yang dilakukan oleh *static AI*. Pertama, *AI* akan mengecek jarak antara *static AI* dengan lawan. Jika jarak yang didapat lebih kecil dari *distanceTolerance* yang telah ditentukan, *aiState* akan dirubah menjadi *closeState* dan sebaliknya jika jarak lebih besar dari *distanceTolerance*, maka *aiState* akan dirubah menjadi *normalState*. Saat *aiState* adalah *closeState*, *AI* harus menunggu selama *closeRate* yang telah ditentukan sebelum melakukan *close comba behaviour*. Saat *aiState* adalah *normalState*, *AI* harus menunggu selama *normalRate* yang telah ditentukan sebelum melakukan *normal state behaviour*.

Close state behaviour merupakan pengambilan keputusan yang dilakukan *static AI* saat berdekatan dengan lawan. Saat berada dalam *close state behaviour*, *AI* akan memiliki kemungkinan lebih besar untuk melakukan serangan dan aksi bertahan. *Normal state behaviour* merupakan pengambilan keputusan yang dilakukan *static AI* saat berada jauh dari lawan. Saat berada dalam *normal state behaviour*, *AI* tidak akan melakukan serangan jarak dekat dan *AI* akan memiliki kemungkinan yang besar untuk bergerak mendekati lawan.

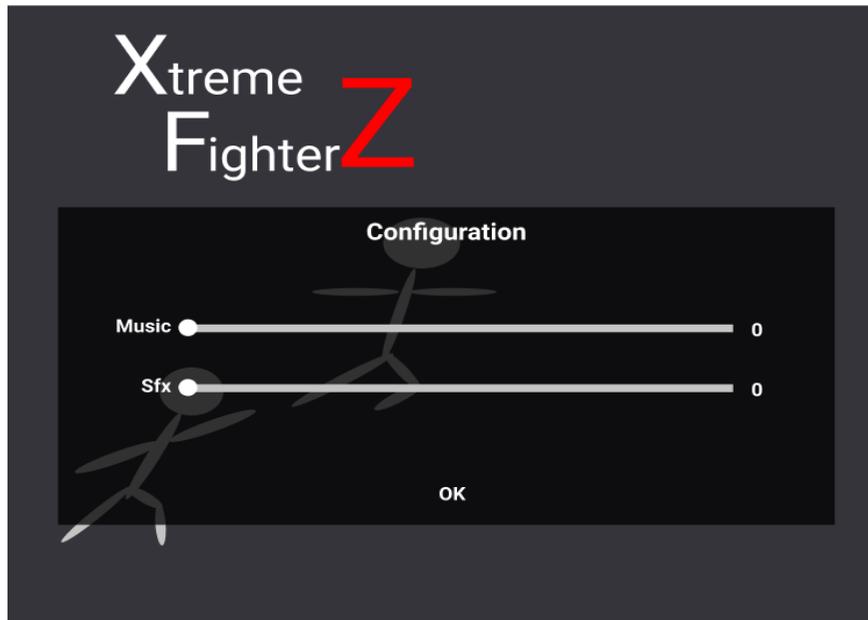
3.2.3 Mockup

Berikut merupakan tampilan desain *mockup* yang digunakan pada *game* Extreme FighterX. Dimulai dari tampilan *main-menu* yang dapat dilihat pada gambar 3.11



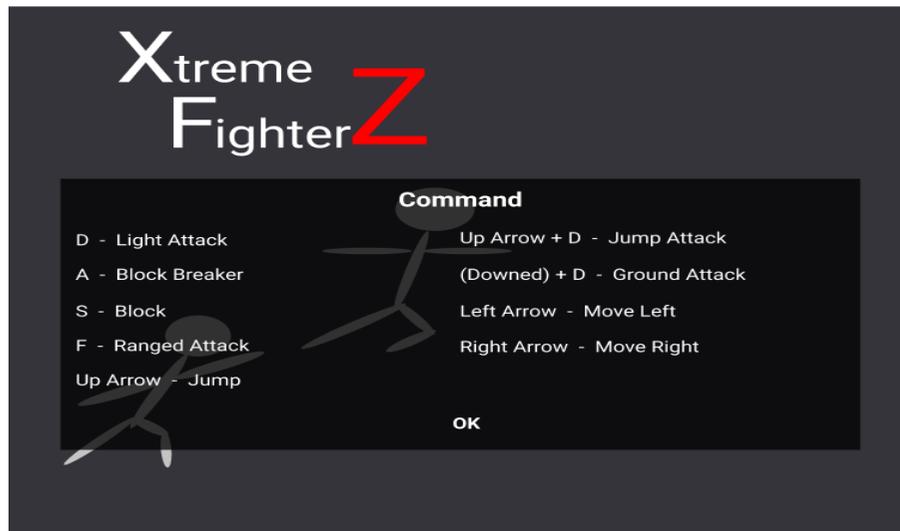
Gambar 3.11 Desain Mockup Main-Menu

Gambar 3.11 merupakan tampilan *mockup main-menu* pada *game* Extreme FighterZ. Pada bagian *main-menu* pemain dapat memilih mode permainan dengan menekan tombol *Play*, membuka tampilan konfigurasi dengan menekan tombol *Option*, melihat control karakter dengan menekan tombol *Command*, dan keluar dari *game* dengan menekan tombol *Quit*.



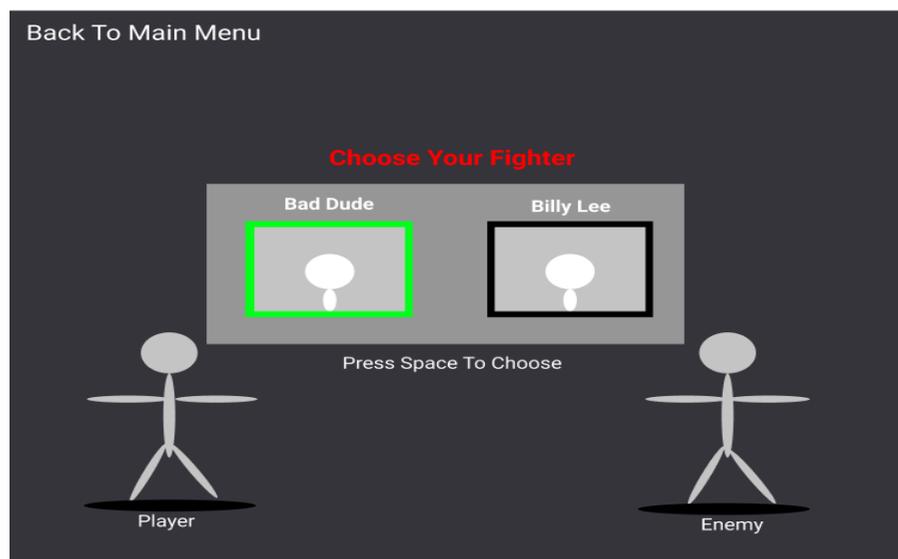
Gambar 3.12 Desain Mockup Configuration

Gambar 3.12 merupakan desain dari tampilan *mockup configuration* yang muncul saat pemain memilih menu *Option* pada *main-menu*. Pada tampilan *configuration* terdapat dua buah slider yang dapat digunakan untuk mengatur volume musik dan *sound effect*. Untuk menutup tampilan *configuration* dan membuka kembali *main-menu*, pemain dapat menekan tombol ok pada bagian bawa tampilan *configuration*.



Gambar 3.13 Desain Mockup Command

Gambar 3.13 merupakan desain dari tampilan *mockup command* yang muncul saat pemain memilih *menu Command* pada *main-menu*. Pada tampilan *command*, terdapat daftar aksi yang dapat digunakan pemain untuk mengontrol karakter dalam *game*. Untuk menutup tampilan *command* dan kembali ke *main-menu*, pemain dapat menekan tombol ok pada bagian bawa tampilan *command*.



Gambar 3.14 Desain Mockup Character Selection

Gambar 3.14 merupakan desain *mockup* halaman *character selection* pada *game* Extreme FighterZ. Pada halaman *character selection*, pemain dapat memilih salah satu karakter dari dua karakter yang ada pada *game*, yaitu Bad Dude dan Billy Lee. Jika pemain ingin kembali ke *main-menu*, pemain dapat menekan tombol *Back To Main Menu*.



Gambar 3.15 Desain Mockup Gameplay

Gambar 3.15 merupakan desain *mockup gameplay* dari *game* Extreme FighterZ. Pada bagian atas, terdapat *health bar* dan *energy bar* yang dimiliki masing-masing pemain. Selain itu, pada bagian bawah *energy bar* masing-masing pemain, terdapat *win indicator* yang akan menunjukkan jumlah ronde yang telah dimenangkan oleh pemain tersebut.