BAB III
PERANCANGAN DAN PEMBAHASAN

3.1. Analisa Kebutuhan

3.1.1 Analisa Kebutuhan Fungsional

Kebutuhan fungsional adalah jenis kebutuhan yang berisi proses-proses apa saja yang dikerjakan oleh sistem. Maka dapat disimpulkan bahwa sistem yang dibutuhkan dalam pembuatan *game* ini adalah yang memiliki kriteria sebagai berikut:

1. Tampilan

   Tampilan pada *game* ini sangat sederhana dengan gambar latarnya berwarna hitam dan terdapat gambar pesawat.

2. Suara pendukung

   Untuk suara *game* ini menggunakan instrumen musik agar suara yang membuat *game* lebih menarik.

3.1.2 Analisa Kebutuhan Non Fungsional

Dalam analisa kebutuhan *non fungsional* untuk menjalankan game ini dibutuhkan seperti pada table III.1 di bawah ini:

Table III.1 Spesifikasi Perangkat Lunak

No.	Nama Perangkat Lunak	Spesifikasi
1	System Operasi	
2	IDE	Adobe Flash Player CS6
3	Tools	Adobe Photoshop CS6 Adobe Air for Android

Sumber : Penelitian Perancangan (2018)

Tabel di atas menerangkan bahwa dalam pembuatan aplikasi ini (*game)*, sistem operasi yang digunakan adalah *Windows* 7 *Ultimate* 32bit, *software* yang digunakan dalam pembuatan aplikasi ini dengan menggunakan Adobe Flash Player dan juga mengunakan *software* pendukung seperti Adobe Photoshop dan Adobe Air.

3.2. Perancangan Perangkat Lunak

Bagian perancangan perangkat lunak akan membahas mengenai karakteristik *software*, perancangan *storyboard*, perancangan antar muka, dan state *transition diagram*.

3.2.1. Karakteristik *Software*

1. *Format*

   Aplikasi yang dibuat merupakan *mobile game* dengan nama pesawat bertahan yang dapat dimainkan dengan mengunakan perangkat *mobile phone* ataupun *smartphone android* dengan spesifikasi dari *mobile phone* minimal *Jelly Bean* 4.1. dengan halaman-halaman yang memiliki fungsi masing-masing. Diantaranya halaman menu, halaman permainan dan halaman *game over.*

2. *Rules*

   Pada *game* ini *user* harus menembak objek pesawat yang akan menabrakan diri ke karakter utama atau pesawat utama. Setiap pesawat yang hancur akan mendapatkan nilai 5.

3. *Policy*

   Pada permainan *user* hanya bisa memilih mulai *game* dan keluar *game*. Di dalam halaman mulai permainan, user hanya bisa melihat perolehan nilai dan melihat HP *bar*.

4. *Role*

   User berperan sebagai pesawat utama penembak yang menembak objek pesawat yang akan mendekat.

5. *Decisions*

   Keputusan yang dapat diambil *user* pada halaman menu utama hanya mulai *game*, keluar *game* dan pengaturan suara. Dalam halaman mulai *game*, *user* dapat menembak pesawat dengan mode tekan dan tahan untuk menembak. Dalam halaman *game over* pemain dapat melihat nilai yang didapat dan terdapat pilihan kembali ke menu utama.

6. *Score Model*

   Score model di akumulasi dari tiap pesawat yang meledak baik yang di tembak maupun yang menabrak karakter utama. Nilai yang di dapat dari tiap pesawat yang meledak adalah 5 point.

7. *Indicator*

   User memainkan dengan menekan dan tahan untuk menembak, dan mengahalangi musuh yang datang

8. *Symbol*

Symbol yang digunakan hanya pada pengaturan suara di dalam halaman menu utama dengan tombol suara.

3.2.2. Perancangan *Storyboard*

Pada pembuatan *game* ini, di gunakan perancangan *storyboard* untuk halaman awal dari aplikasi. Kemudian pengunaan *storyboard* di setiap *scene-scene* berikutnya. Rancangan *storyboard* yang di terapkan pada masing-masing *scene* antara lain:

1. *Stage* 1 : Menu Utama.

   Ketika program di jalankan makan akan langsung memunculkan menu utama yang brada pada *scene* 1. Didalam menu utama terdapat menu mulai yang akan menuju ke bagian memulai permainan, Terdapat juga menu untuk mengatur suara dari aplikasi, jika di tekan akan mematikan suara dari aplikasi. Jika tombol keluar yang di pilih, maka aplikasi akan langsung keluar.

2. *Stage* 2 : Halaman Permainan.

   Halaman ini merupakan pemain mulai memainkan *mobile game system defender* dengan *rules* dan *role* yang telah dijelaskan sebelumnya.

3. *Stage* 3: Halaman *Game Over*

   Halaman *game over* merupakan halaman diaman karakter utama mati atau HP bar menjadi 0, maka akan berpindah ke halaman *game over* dan menampilkan *score* yang dicapai.

Dari penjelasan mengenai rancangan *storyboard*, didapatkan *storyboard* yang lebih terperinci pada table III.3 berikut ini:

Table III. 2 Rancangan *Storyboard*

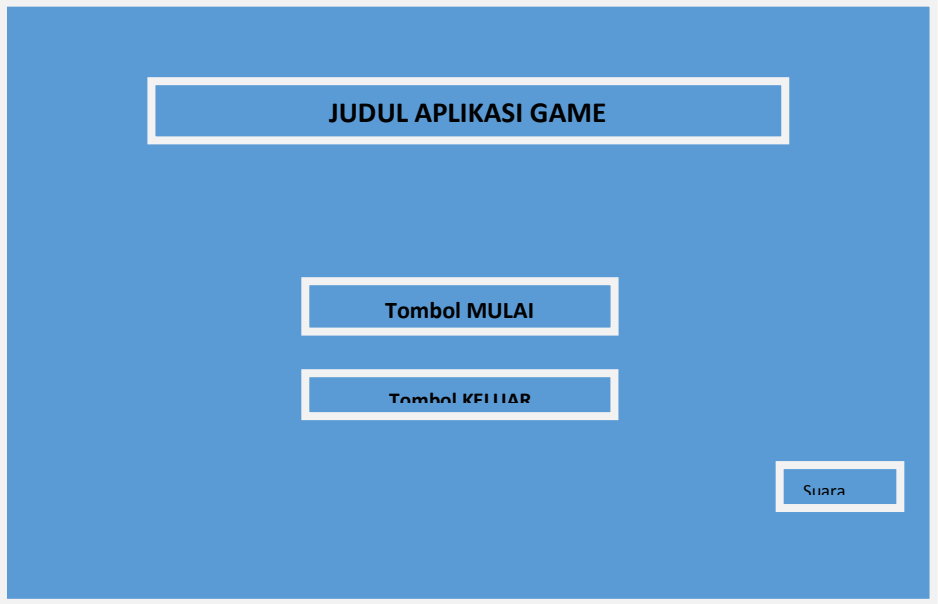
Menu	Isi	Keterangan
Menu Utama	Halaman yang berisikan menu-menu yang akan di pilih.	Background musik di jalankan Background menu dan tombol mewakili isi dari aplikasi
Halaman Permainan	Halaman permainan merupakan halaman utama. Permainan dimulai pada halaman tersebut	*User* mulai memainkan permainan, terdapat *health bar* dari karakter yang merupakan indikator berakhirnya permainan
Halaman *Game Over*	Halaman *game over* yang menunjukan bahwa pemainan telah selesai dan menampilkan skore yang didapat	Hanya terdapat tombol kembali kemenu utama

Sumber : Penelitian Perancangan (2018)

3.2.3. Peracangan Antar Muka

User interface merupakan bagian dari aplikasi yang langsung berhadapan dengan *user* atau pengguna. Berikut ini merupakan gambaran dari *user interface* yang terdapat dalam *game* ini:

1. Halaman Menu Utama



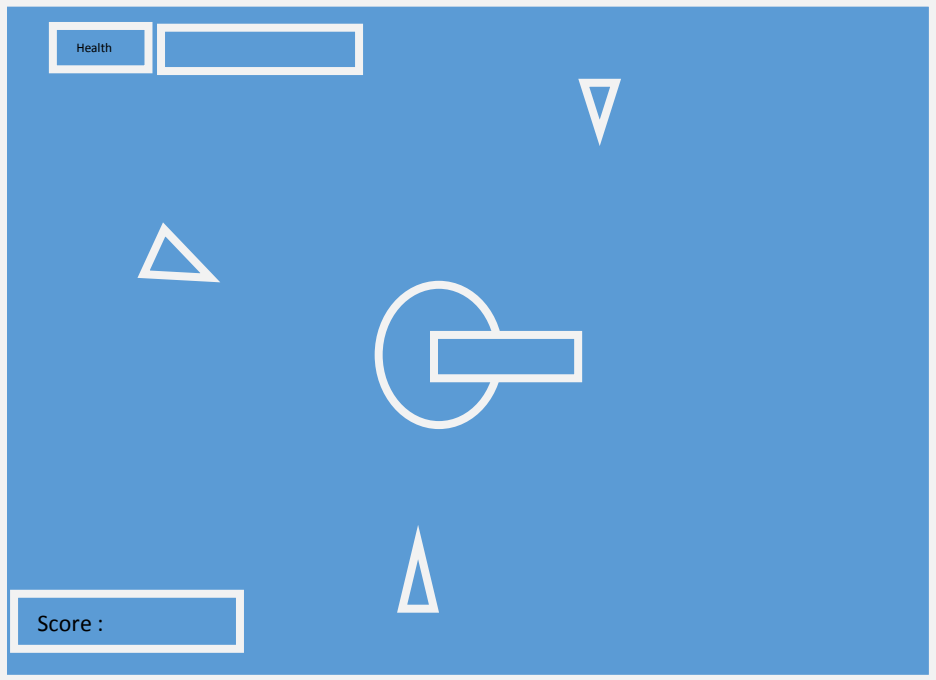
Sumber : Penelitian Perancangan (2018)

Gambar III.1 Halaman Menu Utama

Keterangan:

a. Pada saat game dijalankan *background* musik akan dijalankan secara otomastis.
b. Tombol mulai di tampilkan untuk masuk ke menu permainan dan memulai permainan.
c. Tombol suara berguna untuk mematikan suara musik pada aplikasi.
d. Tombol keluar berfungsi untuk keluar dari *game*.

2. Halaman Permainan



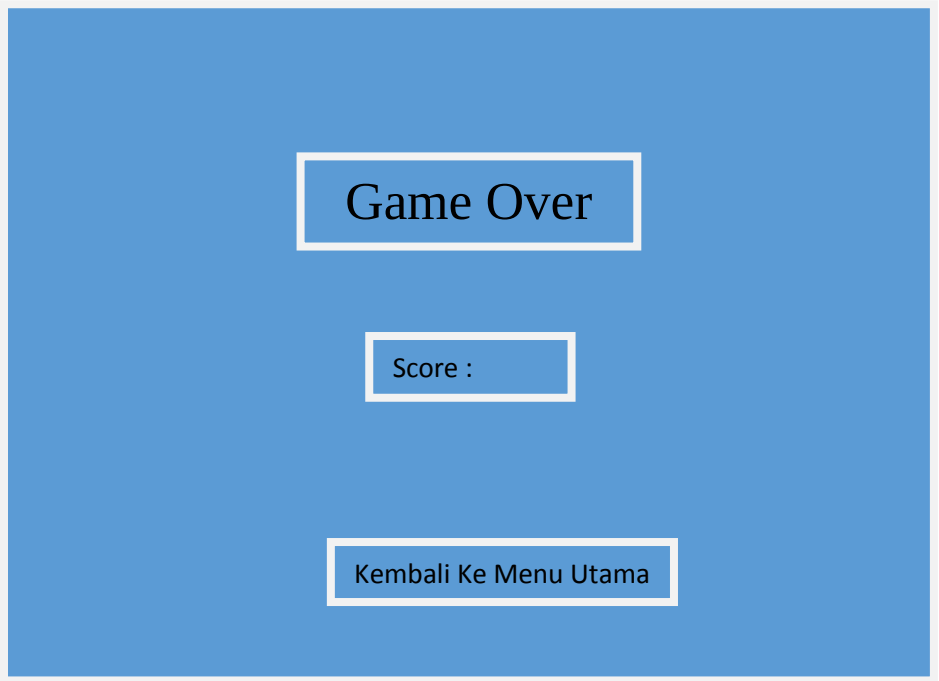
Sumber : Penelitian Perancangan (2018)

Gambar III.2 Halaman Permainan

Keterangan:

a. Terdapat karakter utama yang berwarna biru putih yang akan di gunakan
b. Gambar segitiga berwarna ungu merupakan objek pesawat yang akan menabrak karakter utama
c. Terdapat tampilan nilai sementara
d. Terdapat *health bar* karakter utama

3. Halaman Game Over



Sumber : Penelitian Perancangan (2018)

Gambar III.3 Halaman Game Over

Keterangan:

a. Terdapat *text game over* yang menunjukan permainan telah selesai.

b. Terdapat *text* nilai yang dicapai.

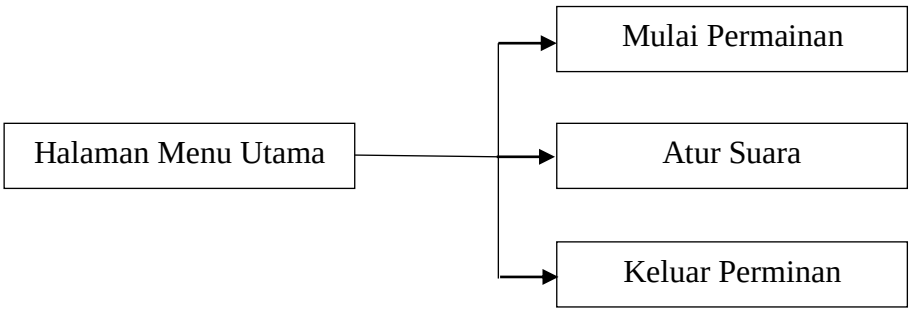
c. Terdapat tombol kembali ke menu utama.

3.2.4. State Transition Diagram

State Transition Diagram adalah suatu diagram yang menggambarkan bagaimana suatu proses dihubungkan satu sama lain dalam waktu yang bersamaan. *State Transition Diagram* digambarkan dengan sebuah *state* yang

berupa komponen *sistem* yang menunjukkan bagaimana kejadian-kejadian tersebut dari satu *state* ke *state* yang lain. Pemodelan ini juga penulis gunakan dalam menjelaskan alur-alur dari aplikasi yang penulis rancang:

1. *Stage* Menu Utama



Sumber : Penelitian Perancangan (2018)

Gambar III.4 *State Transition Diagram* Halaman Menu Utama

Pada gambar *state Transition diagram* diatas menunjukan bahwa halaman menu utama merupakan halaman yang menjadi tampilan utama yang merupakan keterangan dari 3 *stage*, yaitu stage mulai permainan, aturan suara dan keluar dari permainan.

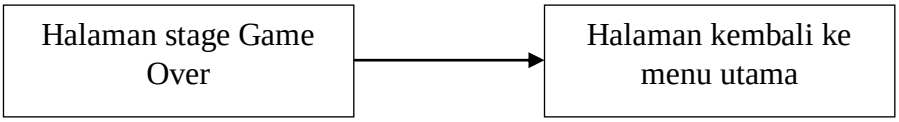
2. *Stage* Permainan

Sumber : Penelitian Perancangan (2018)

Gambar III.5*State Transition Diagram* Halaman *Stage* Permainan

Gamabar III.5 menjelaskan halaman dari *stage* permainan apabila *user* kalah, akan berpindah ke halaman *stage game over*.

3. *Stage Game Over*



Sumber : Penelitian Perancangan (2018)

Gambar III.6*State Transition Diagram* Halaman *Game Over*

Gambar III.6 bahwa dari halaman *game over* menyediakan tombol kembali ke menu utama dan dapat berpindah ke halaman menu utama.

3.3. Implementasi dan Pengujian Unit

3.3.1. Implementasi

1. Implementasi Tampilan Game

Implementasi untuk tampilan game pesawat bertahan berdasarkan hasil dari *storyboard* adalah:

a. Tampilan Menu Utama

Tampilan menu utama ini berisi tentang mulai permainan, aturan suara dan keluar dari permainan.

Sumber : Hasil Penelitian (2018)

Gambar III.7 Halaman Menu Utama

b. Tampilan Permainan

Tampilan permainan berisi karakter utama yang berwarna biru putih yang berada di tengah, gambar segitiga berwarna ungu merupakan

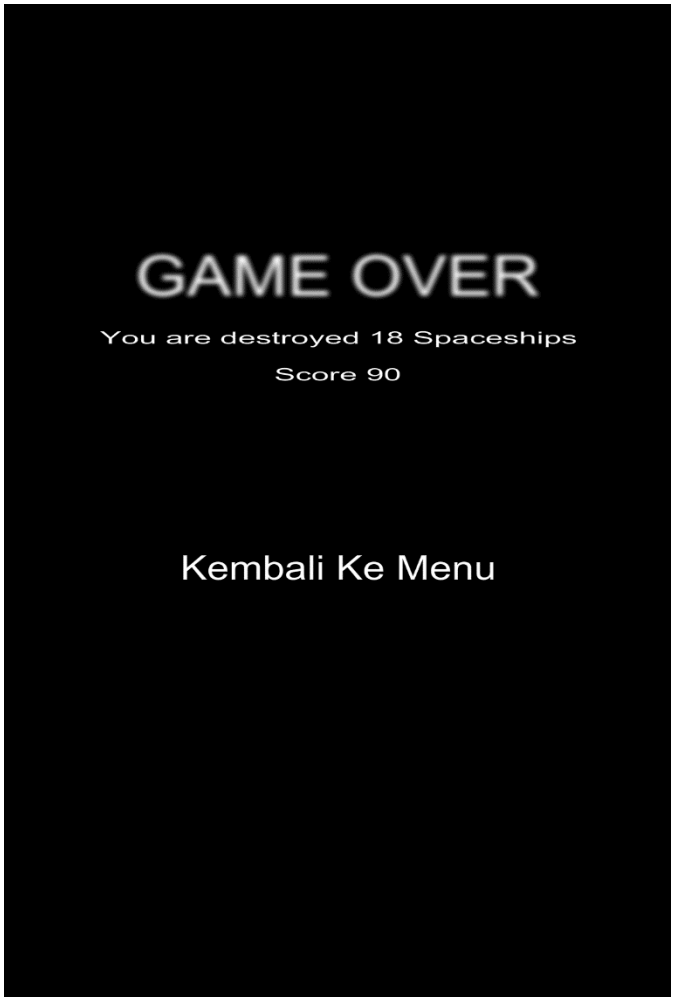
objek pesawat yang akan menabrak karakter utama, tampilan nilai atau *score* sementara, dan tampilan *health bar* untuk karakter utama.

Sumber : Hasil Penelitian (2018)

Gambar III.8 Halaman Permainan

c. Tampilan *Game Over*

Tampilan *game over* ini berisi *text game over* yang menunjukan permainan telah selesai, terdapat *text* nilai yang dicapai, dan terdapat tombol kembali ke menu utama.



Sumber : Hasil Penelitian (2018)

Gambar III.9 Halaman Game Over

3.3.2. Pengujian Unit

1. Pengujian *Black Box / Black Box Testing*

   Pengujian terhadap program yang dibuat menggunakan pengujian *black box* atas terhadap proses masukan dan keluaran.

Tabel III.3 Pengujian *Black Box*

No	INPUT	PROSES	OUTPUT	HASIL PENGUJIAN

1	Halaman Menu Utama	stop(); import flash.events.MouseEvent; import flash.media.Sound; import flash.media.SoundMixer; import flash.system.fscommand; import flash.desktop.NativeApplication; import flash.events.TouchEvent; *//tween semua isi layar* tweenIn(); *//tambah click listener ke tombol start* start_btn.addEventListener(MouseEvent .CLICK, startGame); *//atur perulangan proTipTimer* proTipTimer=new Timer(5000);//langsung ulang proTipTimer.addEventListener(TimerE vent.TIMER, newProTip);//event listener proTipTimer.start();//mulai proTipTimer	Menampilkan halaman Utama yang terdapat tombol pengatur suara On / Off, tombol keluar dan tombol mulai permainan.	Sesuai

		//EXIT GAME for windows btn_exit.addEventListener(MouseEvent. CLICK , bexit); function bexit(event:MouseEvent):void {fscommand("quit");}		
2	Tombol Pengatur Suara	function setMute(vol){ var sTransform:SoundTransform = new SoundTransform(0,0); sTransform.volume = vol; SoundMixer.soundTransform = sTransform; } var Mute:Boolean = false; btn_mute.addEventListener (MouseEvent.CLICK,togglebtn_mute); function togglebtn_mute(event:Event) {if(Mute){ Mute = false; setMute(1); Mute.valueOf(); }	Background musik On ketika tekan tombol suara akan Off	Sesuai

		else{ Mute = true; setMute(0); Mute.valueOf(); } }		
3	Tombol Keluar	btn_exit.addEventListener(MouseEvent. CLICK , bexit); function bexit(event:MouseEvent):void {fscommand("quit");}	Keluar dari permainan	Sesuai
4	Tombol Mulai Permainan	start_btn.addEventListener(MouseEvent .CLICK, startGame); private function startGame(m:MouseEvent){ targetFrame="Game"; tweenOut(); //tween isi layar keluar SoundMixer.stopAll();	Mulai menuju ke halaman permainan	Sesuai
5	Halaman Permainan	const SPAWN_RATE_COOLDOWN=80; const SPAWN_COOLDOWN=50;	Tampilan permainan berisi karakter utama, objek pesawat yang akan menabrak karakter	Sesuai

		public function Level() { //add event listener untuk memunculkan pesawat baru setiap frame addEventListener(Event.ENTER_FRAME, everyFrame); ships=new Array();//atur array pesawat bullets=new Array();//atur array peluru gameOver=false;//game tidak berakhir saat dimulai ledakan = new meledak();//suara ledakan score=0;//skor dimulai dengan nilai 0 spawning=false;//jangan mulai spawning saat dimainkan spawnCounter=0;//mulai hitung numToSpawn=1;//mulai munculkan 1 pesawat numToSpawnCounter=0; //atur waktu untuk menghitung	utama, tampilan nilai atau score sementara, dan tampilan health bar untuk karakter utama.	

		timer = new Timer(3000, 1); //3000 ms = 3 detik timer.addEventListener(TimerEvent.TI MER_COMPLETE, beginSpawning); //jalankan timer timer.start(); } protected function beginSpawning(t:TimerEvent){ spawning=true;//atur variable spawning ke true } //transitionOut protected function transitionOut(t:TimerEvent){ //buat event var ev:LevelEvent = new LevelEvent(LevelEvent.GAME_OVER, score); //kirim event ini		

		dispatchEvent(ev); } protected function everyFrame(e:Event) { //tambah hitungan counter spawnCounter++; numToSpawnCounter++; //fungsi peluru doBullets(); //fungsi pesawat doShips(); //munculkan pesawat doSpawning(); if(!gameOver){ //fungsi player thePlayer.update(); //cek untuk melihat player mati if(thePlayer.health<=0){		

		//atur gameOver ke true gameOver=true; //gunakan kembali timer untuk perlambat transisi ke layar gameOver timer=new Timer(900,1); timer.addEventListener(TimerEvent.TIMER_COMPLETE, transitionOut); timer.start(); //munculkan ledakan beruntun (ExplosionChain) var ex=new ExplosionChain(60, 0.3, 60, 20, 45); //posisi dari ledakan pada pesawat ex.x=thePlayer.x; ex.y=thePlayer.y; addChild(ex);//tambahkan ke layar ledakan.play(); } }		

		//fungsi buffs player //menambahkan HP bar healthBar.gotoAndStop((thePlayer.healt h/10)*100); //fungsi skor output.text="Score: "+score*5; //fungsi game over } //fungsi untuk perbarui semua pesawat protected function doShips(){ //buat perulangan to iterate melalui semua kapal; for (var count=ships.length-1; count>=0; count--) { var sh = ships[count]; //update pesawat sh.update(); //tambah perulangan baru untuk		

		melewati semua bullets for (var bcount=bullets.length-1; bcount>=0; bcount--) { //jika bullets menyentuh pesawat if (sh.hitTestObject(bullets[bcount])) { //jika menyentuh berarti meledak sh.health--;//kehilangan HP 1 poin removeChild(bullets[bcount]);//hapus dari layar bullets.splice(bcount, 1);//hapus dari daftar } } //jika pesawat menabrak player if(sh.hitTestObject(thePlayer)){ //lukai player thePlayer.health--;		

		//lenyapkan pesawat sh.health=0; } //jika pesawat meledak if (sh.health <= 0) { //proleh skor score++; //munculkan ledakan beruntun (ExplosionChain) var ex=new ExplosionChain(5, 1, 10, 5, 15); //posisi dari ledakan pada pesawat ex.x=sh.x; ex.y=sh.y; addChild(ex);//tambahkan ke layar removeChild(sh);//hapus pesawat dari layar		

		ships.splice(count, 1);//hapus pesawat dari daftar } } }		
6	Halaman *Game Over*	function doGameOver(e:LevelEvent){ //atur lastScore menjadi skor dari kustom level event kita lastScore=e.score; targetFrame="GameOver"; tweenOut(); //tween isi layar keluar }	berisi text game over terdapat text nilai yang dicapai, dan terdapat tombol kembali ke menu utama.	Sesuai
7	Tombol Kembali Menu Utama	//tambah click listener ke tombol return return_btn.addEventListener(MouseEve nt.CLICK, returnGame);	kembali ke menu utama.	Sesuai

Sumber : Penelitian Perancangan (2018)

3.3.3. *Suport*

Berikut ini merupakan *spesifikasi minimum game* pesawat bertahan berbasis *android*.

Tabel III.4 Spesifikasi Minimum *Smartphone* Untuk Aplikasi

Kebutuhan	Keterangan
Perangkat	Android Smartphone
Sistem Operasi	Android, versi Jelly Bean 4.1
Processor	Qualcomm MSM7227A Snapdragon, 1 GHz Cortex-A5 GPU Andreno 200
RAM	512 MB

Sumber : Penelitian Perancangan (2018)