

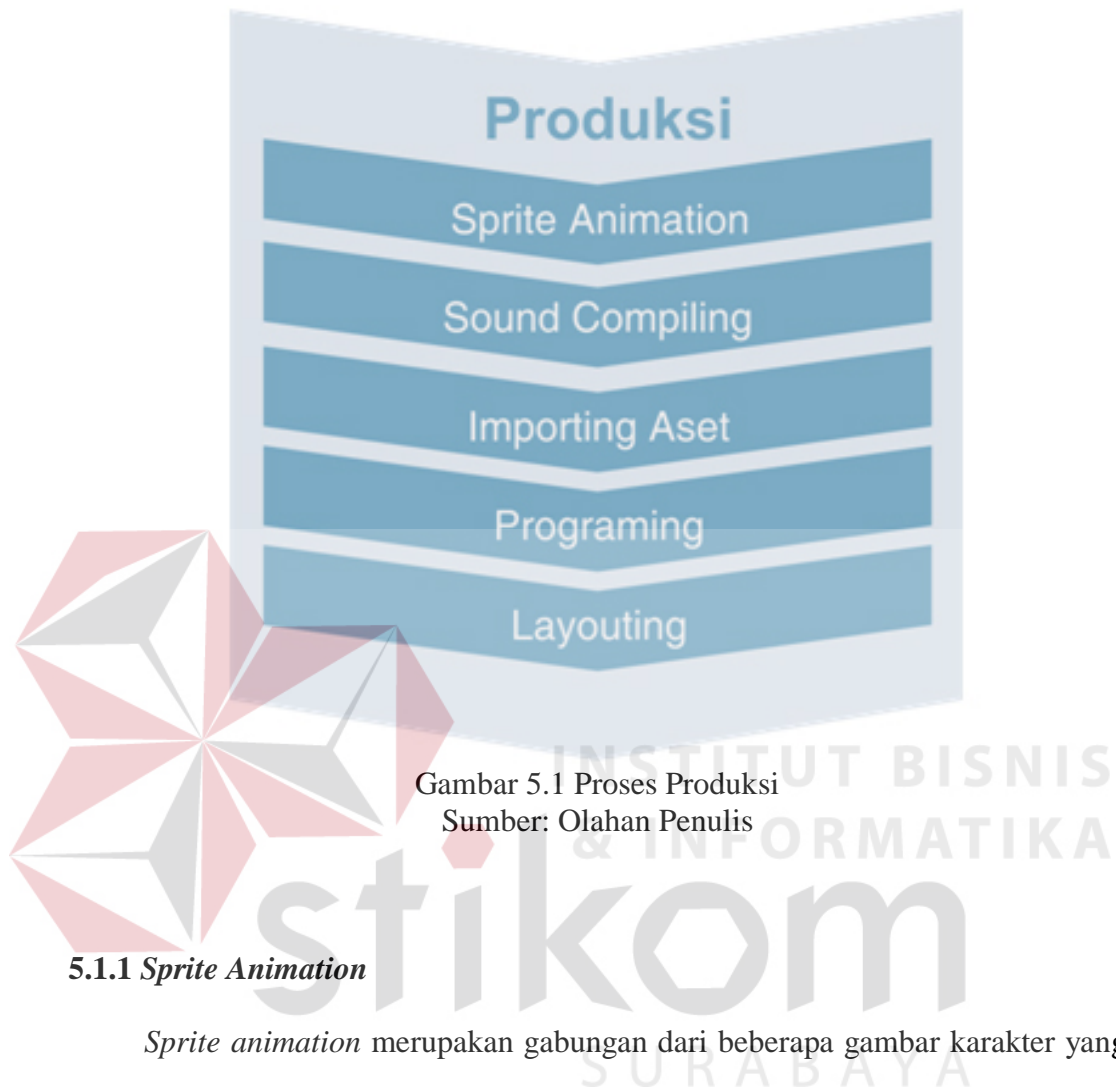
BAB V

IMPLEMENTASI KARYA

Laporan tugas akhir pada BAB V dijelaskan mengenai proses atau jalan cerita dalam desain dan pembuatan *game* bergenre *vehicular combat* untuk anak-anak. Sebelum memulai mendesain, terlebih dahulu dilakukan proses analisa *keyword* dan analisa warna seperti yang telah dijabarkan pada BAB III sebelumnya. Proses analisa *keyword* membantu dalam menentukan desain *gameplay* dan *environment*. Proses analisa ini membantu memudahkan dalam keseluruhan pembuatan *game* ini, baik itu pemilihan *font*, pembuatan karakter dan alur cerita dapat terintegrasi menjadi satu kesatuan. Setelah analisa *keyword* terlaksana maka dilanjutkan pada proses perancangan karya yang telah dijelaskan pada BAB IV, untuk dapat diimplementasikan pada produksi *game* ini.

5.1 Produksi

Tampak pada gambar 4.1 alur perancangan bahwa proses produksi dalam pembuatan *game* ini meliputi *sprite animation*, *sound compiling*, importing aset, programing, serta layouting. Pada tahap ini dijelaskan secara rinci proses apa saja yang perlu dilakukan dalam memproduksi sebuah *game*. Lihat gambar 5.1 proses produksi



Gambar 5.1 Proses Produksi
Sumber: Olahan Penulis

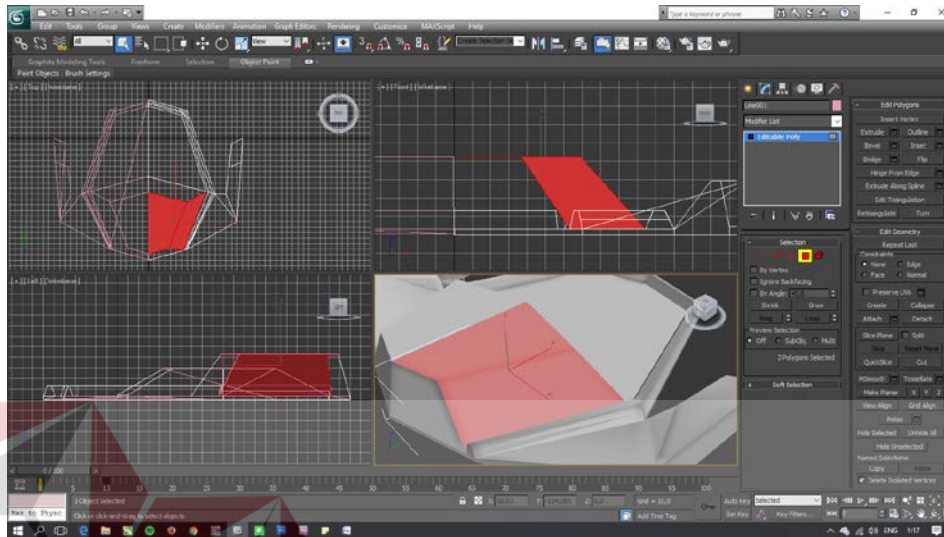
5.1.1 *Sprite Animation*

Sprite animation merupakan gabungan dari beberapa gambar karakter yang telah dibuat dan apabila di tampilkan secara berurutan akan menimbulkan kesan bergerak seperti berjalan atau meluncur. Sebelum sebuah gambar dapat menjadi sprite banyak proses yang harus dilalui terlebih dahulu. Berikutnya akan dijelaskan satu persatu mengenai proses tersebut.

1. *Modeling Dasar Asset*

Dalam karya ini semua gambar dibuat dengan bantuan *software 3ds Max* dan *Photoshop*, baik karakter (*player*), *enemy* maupun *obstacle* dan aset-aset lain-

nya. Karakter (*player*) dan *enemy* pada FGD adalah *modeling* dasar yang dibuat di software *3ds Max* dengan *default material* dan *default lighting*.

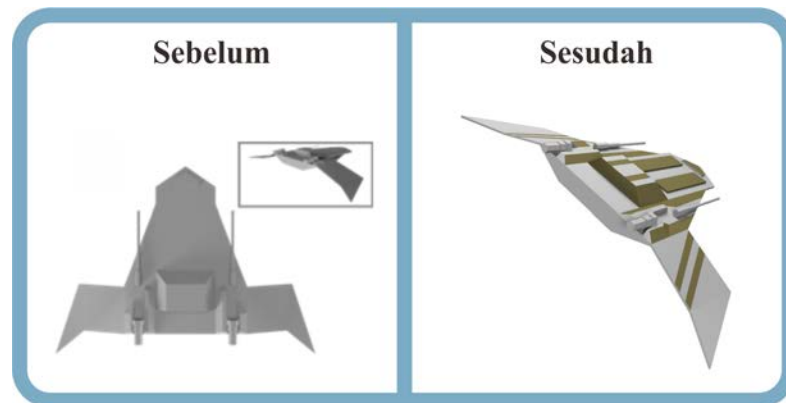


Gambar 5.2 *Modeling* dasar software *3ds max*

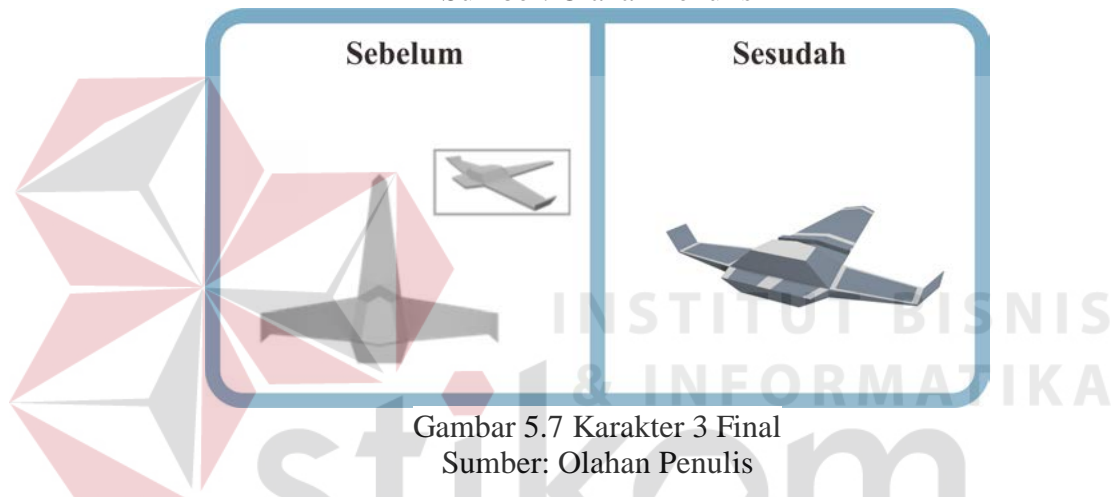
Sumber: Olahan Penulis

2. Penambahan warna dan *lighting*

Setelah acuan karakter (*player*) dan *enemy* yang telah dibuat dan terpilih melalui diskusi, maka tahap selanjutnya adalah pemberian warna dan diberi *lighting* agar hasil gambar tidak terlalu *flat*. Tahap pewarnaan dan penambahan cahaya dilakukan di software *3ds max*. Dalam tahap ini akan disesuaikan warna dan cahaya yang nanti akan tampak dalam *game*. Proses pewarnaan dan penambahan *lighting* bisa dilihat pada gambar 5.2 sampai 5.4 Perbedaan hasil dari sebelum dan sesudah diberi warna dan *lighting* dapat dilihat pada gambar 5.4 sampai 5.7.



Gambar 5.6 Karakter 2 Final
Sumber: Olahan Penulis

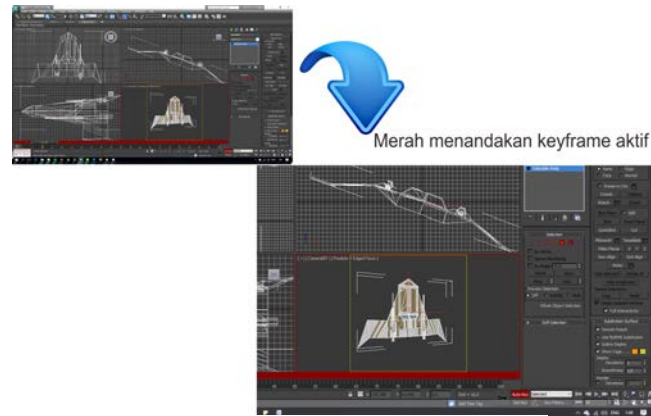


Gambar 5.7 Karakter 3 Final
Sumber: Olahan Penulis

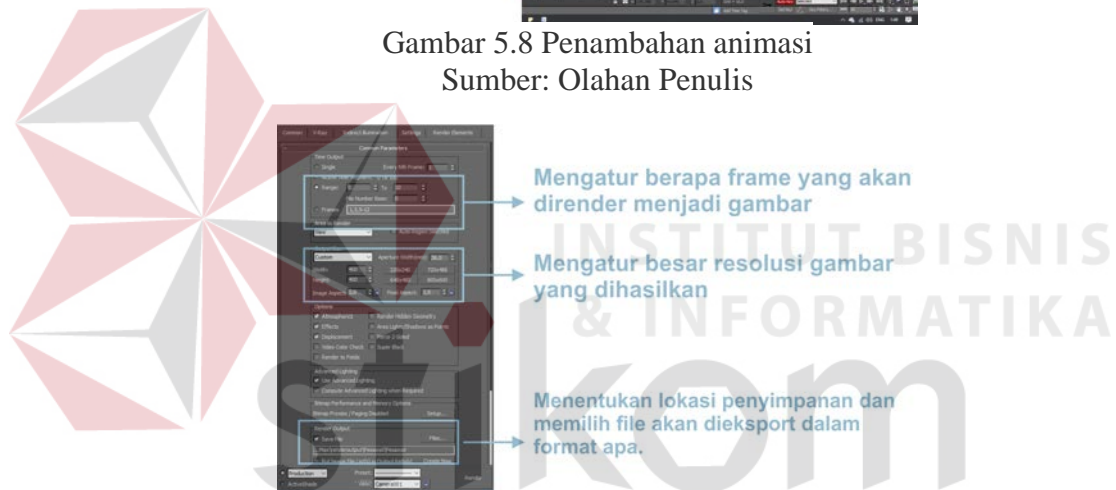
3. Animation dan Rendering

Setelah karakter (*player*), *enemy* dan aset-aset lain dibuat dan sudah final, tahap selanjutnya akan dibuat *sprite* sesuai dengan animasi. Animasi akan dilakukan di *software 3ds Max*. Disini animasi hanya melakukan gerakan sederhana seperti *merotate* dan *scaling*. Penambahan animasi dilakukan dengan cara menentukan *keyframe* dan menggerakkan objek saat *frame* aktif. Setelah objek sudah ditambahkan animasi, langkah selanjutnya adalah *rendering*, dimana *rendering* ini adalah mengubah bentuk 3d menjadi gambar yang nantinya akan dibutuhkan dalam *game*. File gambar dari *render* akan berbentuk

seperti gambar yang hampir sama tapi sedikit berubah hingga menjadi sebuah gerakan.



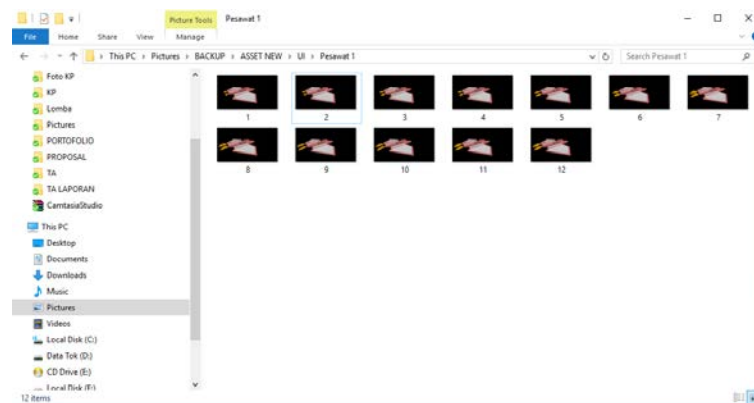
Gambar 5.8 Penambahan animasi
Sumber: Olahan Penulis



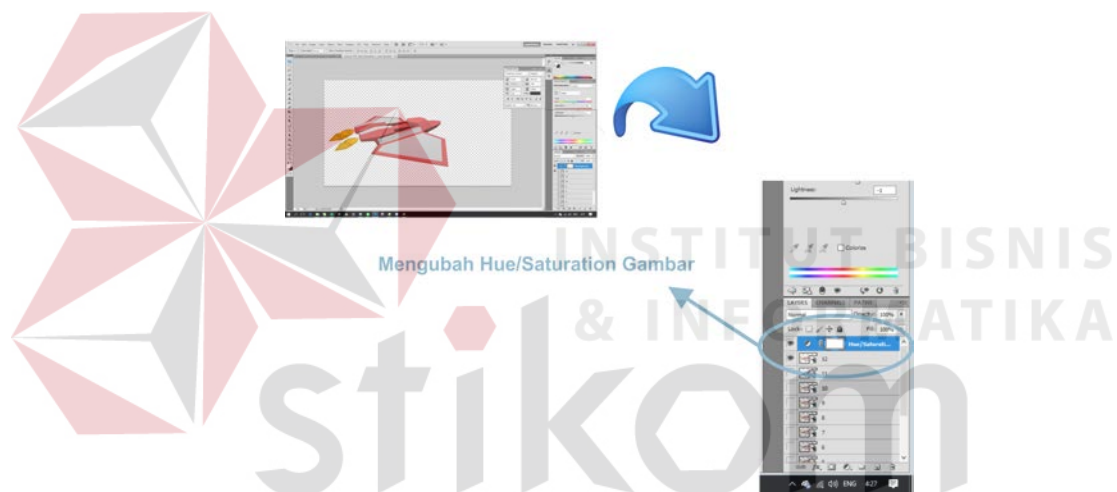
Gambar 5.9 Mengatur *Render Setting*
Sumber: Olahan Penulis

4. *Grading* Warna

Proses *rendering* ini diterapkan pada *background*, *enemy* dan *player*. Hasil *sprite* yang sudah jadi dapat dilihat pada gambar 5.11. setelah dirender semua gambar akan diedit satu persatu dengan *photoshop* agar pewarnaan tidak pudar dan *grading* warna pada gambar bisa di selaraskan, tahap bisa dilihat pada gambar 5.10 dan 5.11.



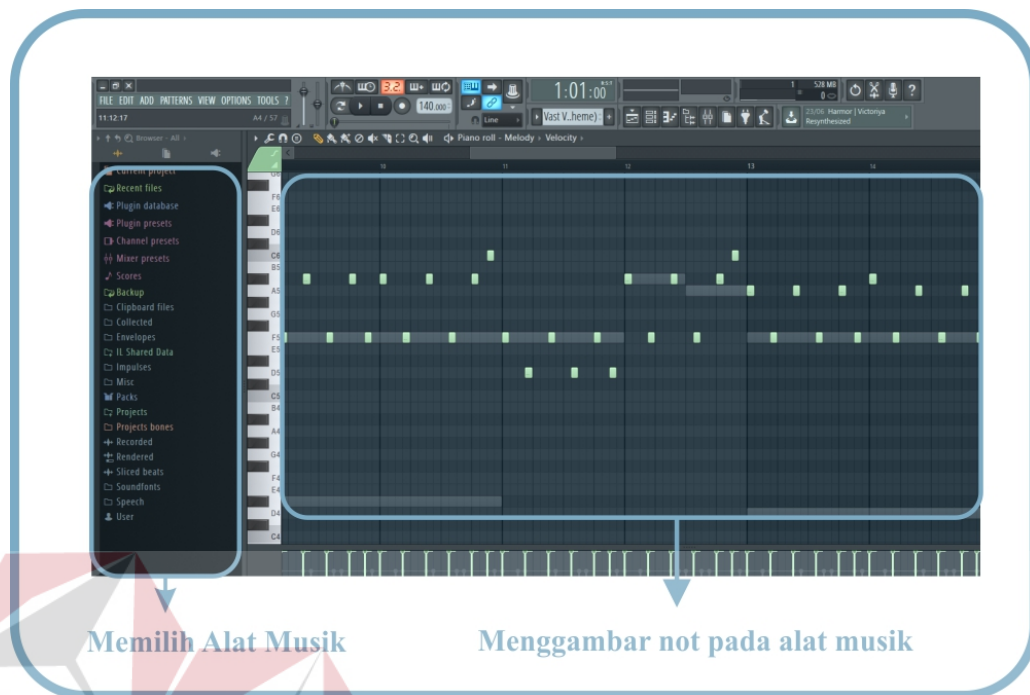
Gambar 5.10 Hasil *Render* yang akan di Edit
Sumber: Olahan Penulis



Gambar 5.11 Mengatur *Hue dan Saturation* pada *photoshop*
Sumber: Olahan Penulis

5.1.2 Sound Compiling

Pembuatan sound dalam game ini menggunakan *software frutyloop*. Dengan *software* ini memungkinkan untuk membuat audio midi maupun mp3. Di *frutyloop* ini sudah disediakan alat musik *virtual* jadi, audio dibuat secara menggambar *not* satu persatu dari tiap alat musik. Lihat gambar 5.13



Gambar 5.12 Pembuatan Sound
Sumber: Olahan Penulis

Audio dijadikan sebagai komponen pendukung dalam proses produksi untuk mendukung *game* agar lebih hidup dan tidak membosankan. Agar pemain tidak bosan audio dibedakan menjadi 4 yaitu pada *user interface*, *level 1* dan 2, *level 3* dan 4, dan *level 5* dan 6.

5.1.3 Importing Asset

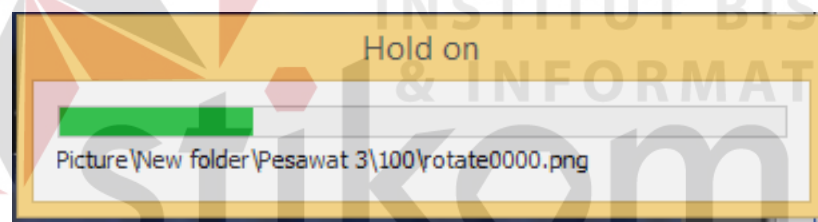
Import asset ke *Unity* adalah hal paling sederhana yang bisa kita lakukan. Ketika membuat *project* baru, pada direktori utama *project* kita, *Unity* secara otomatis akan menyediakan *folder* bernama “*Assets*”. Pada *folder* itulah kita harus menyalin (*copy paste*) semua *asset* yang akan kita gunakan pada *game*. Mulai dari gambar, suara, musik, *code*, animasi, apapun. Khusus *file* yang tidak terbaca oleh *Unity*, kita bisa memasukkannya ke *folder* “*Resources*”.

| Name | Date modified | Type | Size |
|-------------------------|--------------------|------------------------|-----------|
| Assets | 7/21/2017 8:21 PM | File folder | |
| Library | 7/22/2017 11:19 PM | File folder | |
| ProjectSettings | 7/22/2017 9:14 PM | File folder | |
| Temp | 7/23/2017 8:06 AM | File folder | |
| Assembly-CSharp | 7/22/2017 4:38 PM | Visual C# Project f... | 6 KB |
| Assembly-CSharp-vs | 7/22/2017 4:38 PM | Visual C# Project f... | 6 KB |
| constellation | 6/12/2017 1:39 PM | APK File | 20,088 KB |
| Constellation | 7/23/2017 8:06 AM | Microsoft Visual S... | 2 KB |
| Constellation.userprefs | 7/22/2017 11:14 PM | USERPREFS File | 5 KB |

Gambar 5.13 Struktur Direktori *Project*

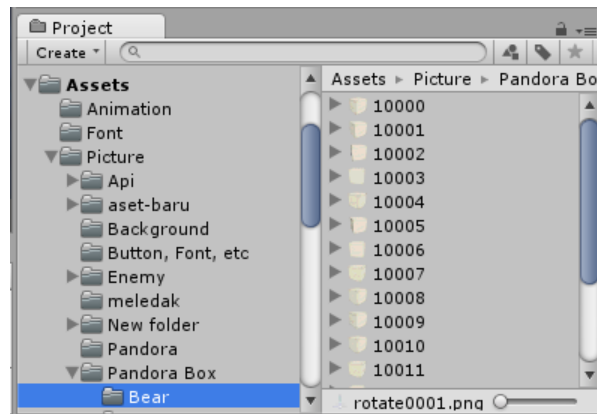
Sumber: Olahan Penulis

Setelah memasukkan *asset-asset* pada *folder* itu, *Unity* akan membaca *file* tersebut untuk disiapkan pada jendela *editor*. Kita harus menunggu proses tersebut selesai seperti yang terlihat di gambar 5.15.

Gambar 5.14Tampilan saat *unity* membaca *asset*

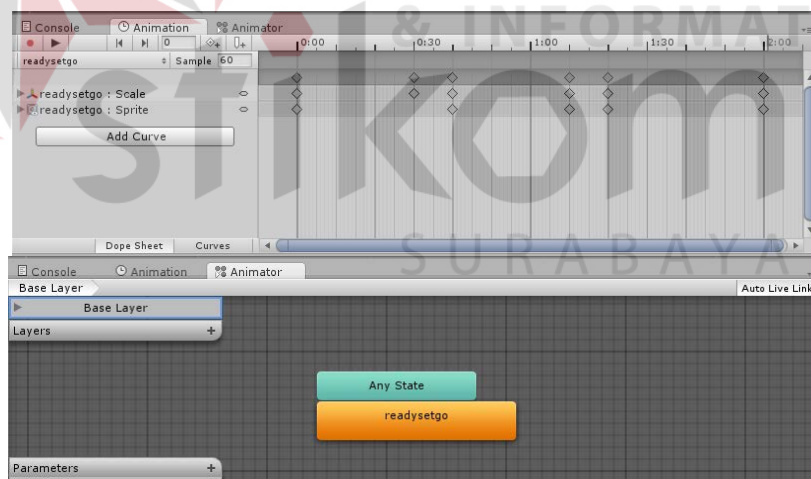
Sumber: Olahan Penulis

Bila *asset* sudah terbaca oleh editor *Unity*, maka tampilan *project* akan menampilkan aset-aset tersebut. Siap untuk digunakan ke dalam *game*. Penggunaan aset tersebut juga sangat mudah tinggal *copy paste*. Hal ini berlaku untuk semua *asset*, baik gambar, audio, dan sebagainya.



Gambar 5.15 Tampilan Jendela *Project* menampilkan aset
Sumber: Olahan Penulis

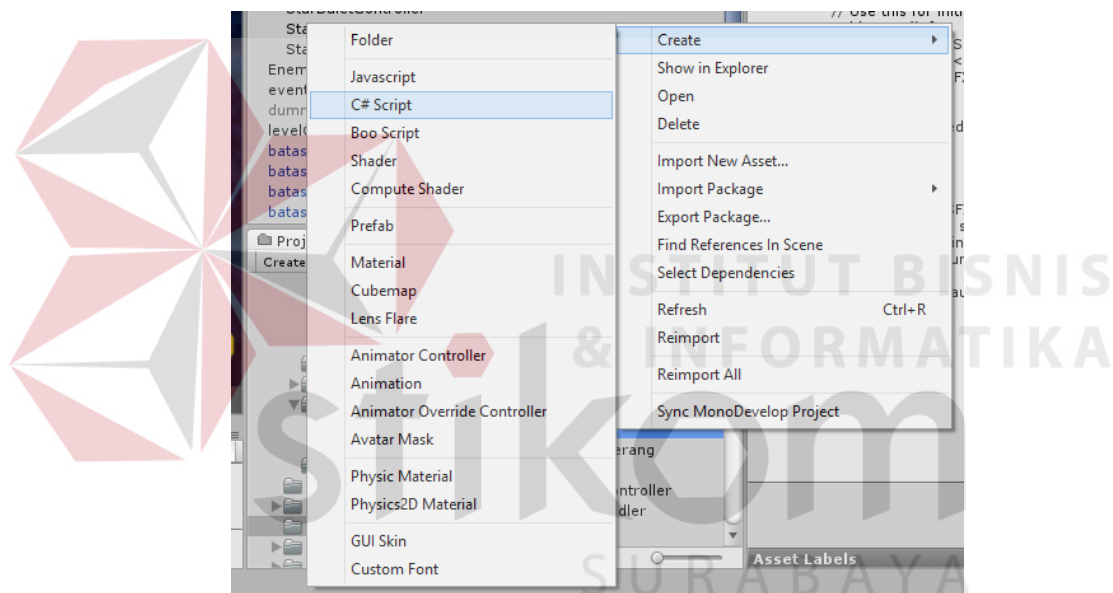
Setelah semua aset dimasukkan aset-aset yang memiliki gerakan bisa diatur kecepatan pergerakannya di *timeline* dan untuk menyesuaikan *FPS* (*Frame per Second*) nya, untuk lebih jelasnya lihat gambar 5.19.



Gambar 5.16 Mengedit Animasi *Timeline*
Sumber: Olahan Penulis

5.1.4 Programing

Bagian tersulit dari membuat *game* dengan *Unity* adalah saat menulis kode program. Ada 2 bahasa pemrograman yang (sering) digunakan *programmer* untuk mengembangkan *game* di *Unity*, yaitu *C#* dan *Javascript* yang lebih mudah adalah *C#* karena sintaks (aturan) yang digunakan mirip dengan kebanyakan bahasa pemrograman tingkat tinggi (Java, C++). Pertama-pertama, buatlah *script* kosong pada *Unity* dengan klik kanan jendela *Project*, pilih *Create*, lalu *C# Script*.



Gambar 5.17 Langkah-langkah membuat *script*
Sumber: Olahan Penulis

Unity akan membuatkan sebuah *script* yang isinya seperti berikut:

```

Imported Object
using UnityEngine;
using System.Collections;

public class Contoh : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

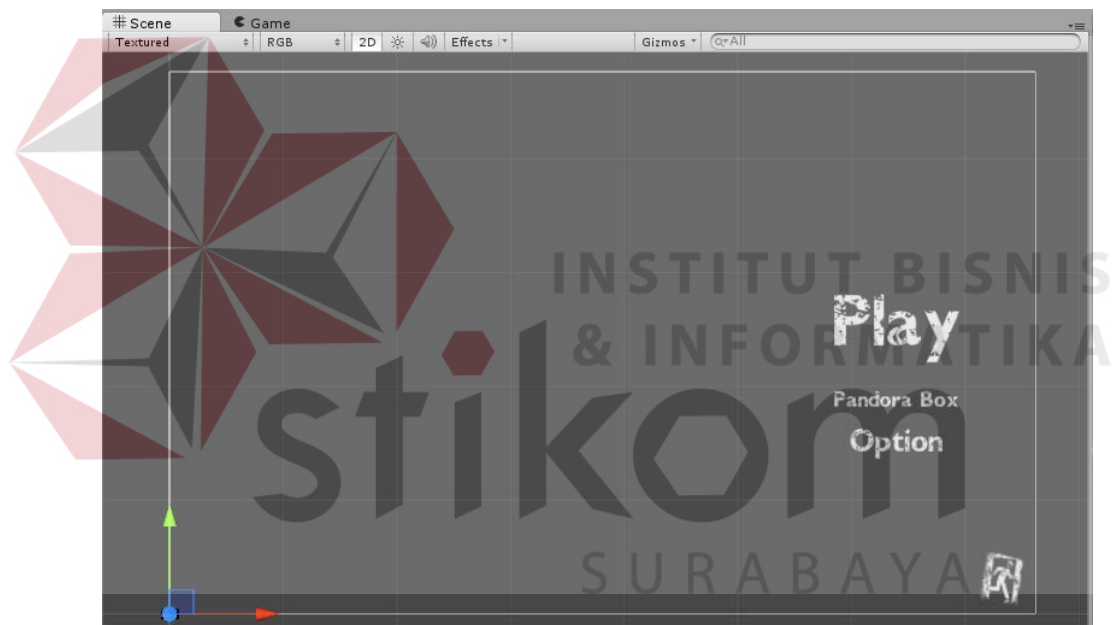
}

```

Gambar 5.18 Contoh *Script* kosong
Sumber: Olahan Penulis

5.1.5 Layouting

Pada proses layouting ini dijalankan proses penataan secara keseluruhan dalam satu frame atau disebut *scene* pada *game*. *Scene* inilah yang nantinya dapat dilihat pada layar oleh pemain. *Scene* dengan ukuran yang telah disesuaikan diisi dengan *background*, karakter, serta aset user interface lain yang akan nampak pada satu *room* tersebut, hingga jumlah sebuah *room* yang bisa dimainkan. (lihat gambar 5.20)



Gambar 5.20 *Layouting*
Sumber: Olahan Penulis

5.2 Sreenshoot



Gambar 5.21 Animasi sebelum memulai *game*

Sumber: Olahan Penulis



Gambar 5.22 *Userinterface Game*

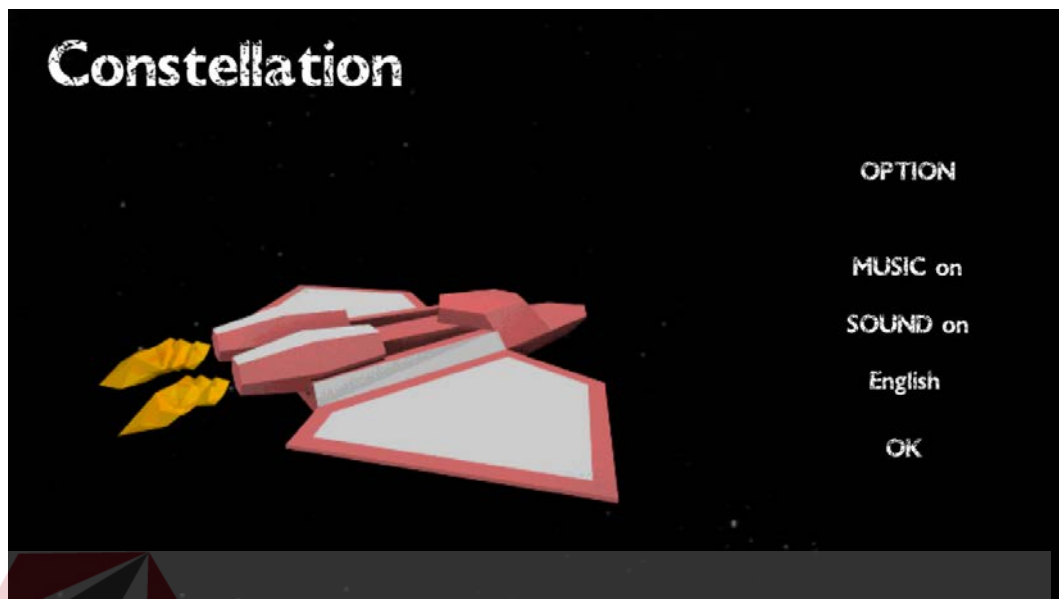
Sumber: Olahan Penulis



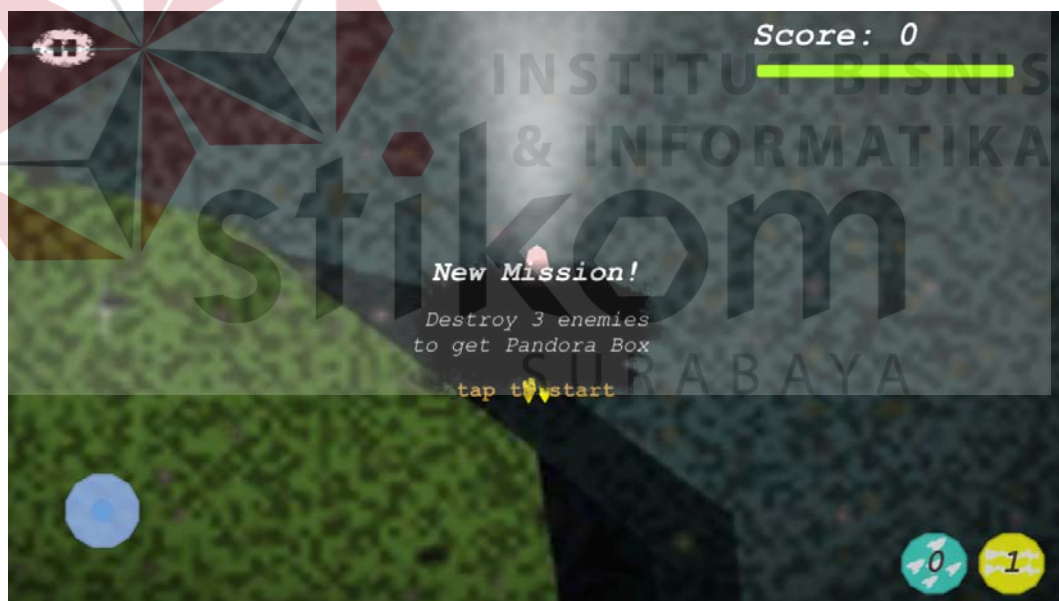
Gambar 5.23 Pemilihan *level*
Sumber: Olahan Penulis



Gambar 5.24 Isi menu Pandora Box
Sumber: Olahan Penulis



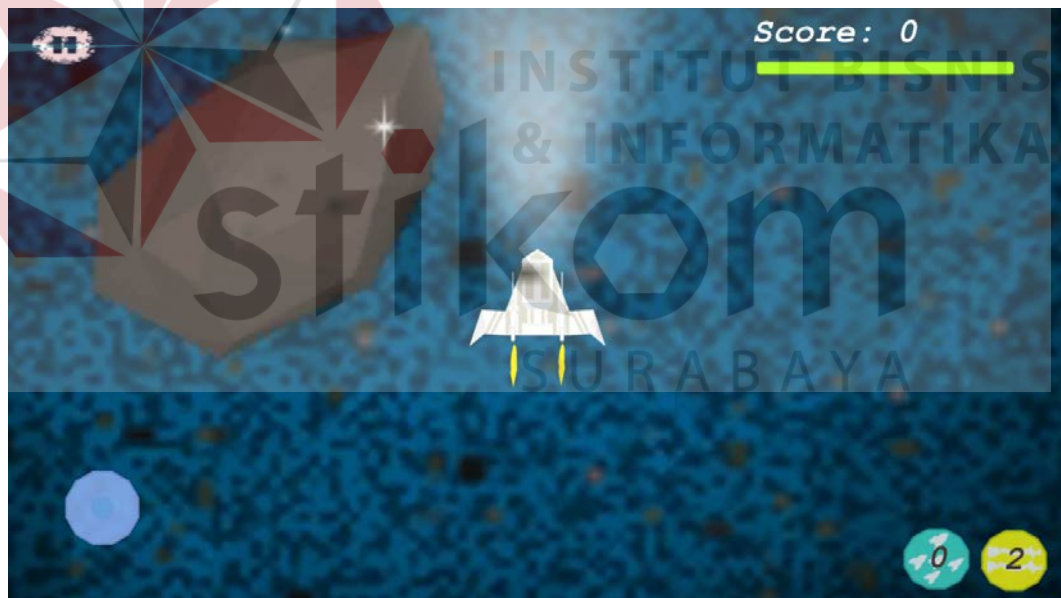
Gambar 5.25 Isi menu option
Sumber: Olahan Penulis



Gambar 5.26 Game play level 1
Sumber: Olahan Penulis



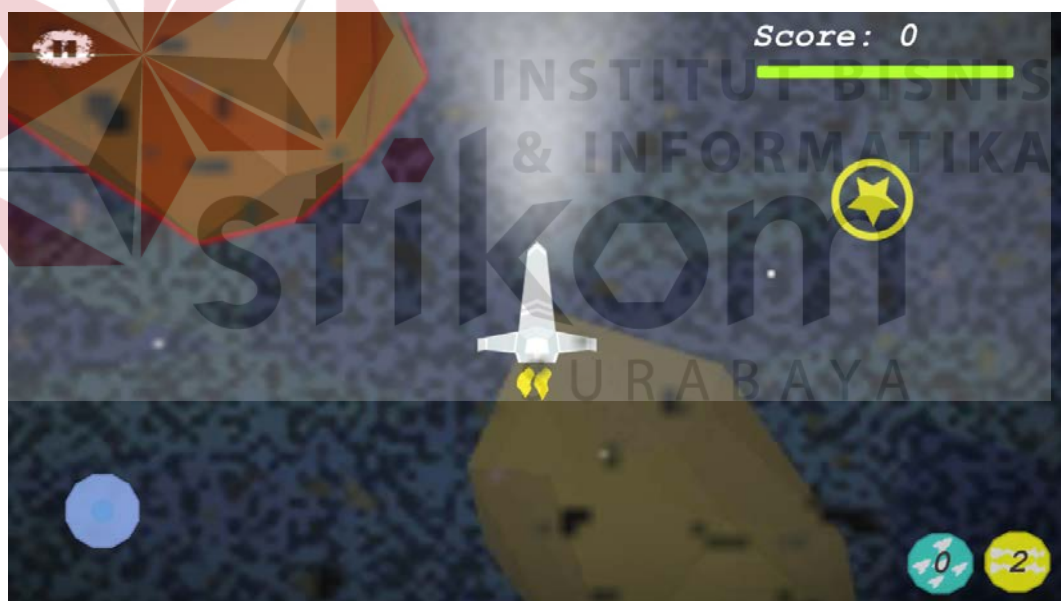
Gambar 5.27 *Game play level 2*
Sumber: Olahan Penulis



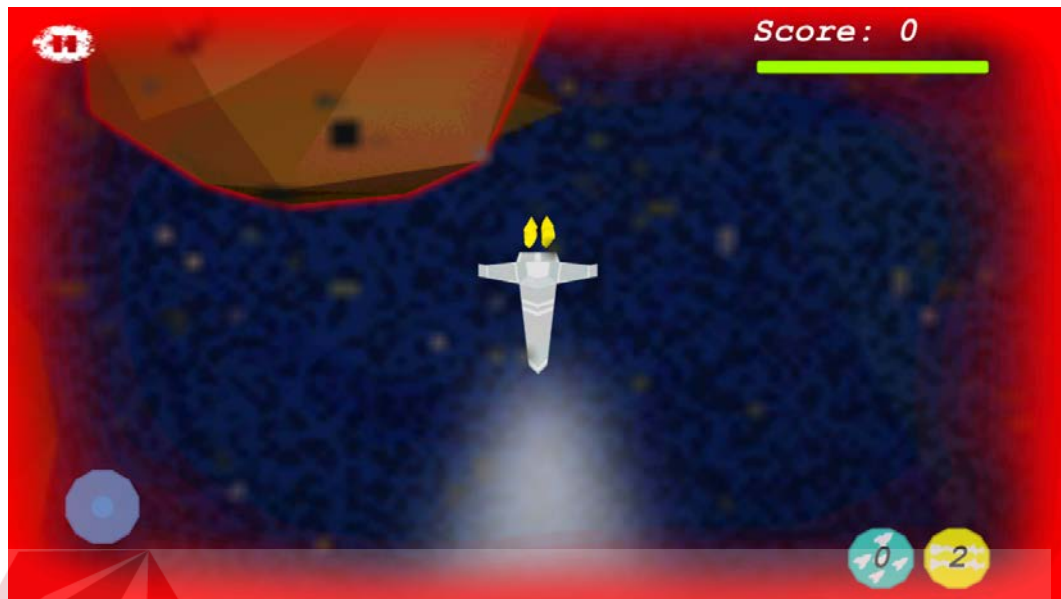
Gambar 5.28 *Game play level 3*
Sumber: Olahan Penulis



Gambar 5.29 *Game play level 4*
Sumber: Olahan Penulis



Gambar 5.30 *Game play level 5*
Sumber: Olahan Penulis



Gambar 5.31 *game play level 6 dan player terkena serangan*
Sumber: Olahan Penulis



Gambar 5.32 Exit menu
Sumber: Olahan Penulis