

**RANCANG BANGUN GAME COMPUTER ARCADE,
STUDI KASUS : GAME SPACE FIGHT, THE LEGEND OF BATU CORPORATION**

TUGAS AKHIR

**Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Tehnik Informatika**



Nama : Puthut Galih Martiyandu

No.Mahasiswa : 99523159

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2011

LEMBAR PENGESAHAN PEMBIMBING

**RANCANG BANGUN GAME COMPUTER ARCADE,
STUDI KASUS : GAME SPACE FIGHT, THE LEGEND OF
BATU CORPORATION**

TUGAS AKHIR

Oleh:

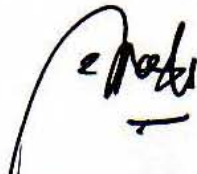
Nama : Puthut Galih Martiyandu

No. Mahasiswa : 99523159

Yogyakarta, Agustus 2011

Menyetujui,

Pembimbing I



Yudi Prayudi, S.Si, M.Kom

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini.

Nama : Puthut Galih Martiyandu

No. Mahasiswa : 99523159

Menyatakan bahwa Tugas Akhir ini adalah hasil pekerjaan saya sendiri, dan sepanjang sepengetahuan saya, tidak berisi materi yang ditulis oleh orang lain sebagai persyaratan penyelesaian studi di Universitas Islam Indonesia atau perguruan tinggi lain, kecuali bagian-bagian tertentu yang saya ambil sebagai acuan dengan mengikuti tata cara dan etika penulisan karya ilmiah yang lazim. Jika ternyata terbukti pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab saya.

Yogyakarta, Agustus 2011



Puthut Galih Martiyandu

LEMBAR PENGESAHAN PENGUJI
RANCANG BANGUN GAME COMPUTER ARCADE,
STUDI KASUS : GAME SPACE FIGHT, THE LEGEND OF BATU CORPORATION

TUGAS AKHIR

Oleh :

Nama : Puthut Galih Martiyandu

No. Mahasiswa : 99523159

Telah Diperthankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta,

Agustus 2011

Tim Penguji

Yudi Prayudi, S.Si, M.Kom

Ketua

Ami Fauzijah, S.T, M.T

Anggota I

Affan Mahtarami, S.Kom, M.T

Anggota II

Mengetahui,

Ketua Jurusan Teknik Informatika

Universitas Islam Indonesia



Yudi Prayudi, S.Si, M.Kom

Kata Pengantar

Bismillahirrahmanirrahim

Assalamualaikum Warahmatullah Wabarakatuh

Syukur Alhamdulillah penulis panjatkan kehadiran Allah SWT, karena dengan segala rahmat, hidayah, dan inayahNya penulis dapat menyelesaikan penulisan Tugas Akhir dengan judul “ RANCANG BANGUN GAME COMPUTER ARCADE, STUDI KASUS : GAME SPACE FIGHT, THE LEGEND OF BATU CORPORATION “. Shalawat serta salam tidak lupa penulis haturkan pada Nabi Muhammad SAW, keluarga, sahabat, serta umat – umat beliau yang istiqamah mengikutinya sampai akhir zaman.

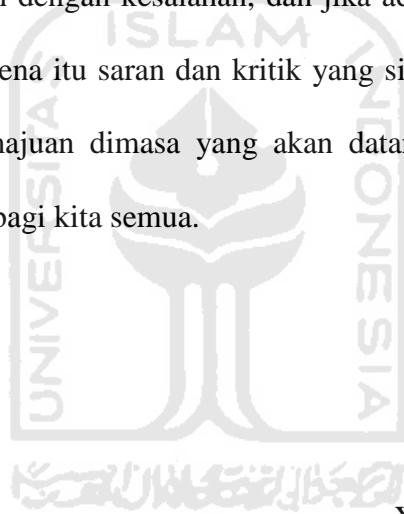
Kelancaran dalam pembuatan Tugas Akhir ini tidak terlepas dari bantuan berbagai pihak, baik secara langsung maupun tidak langsung. Pada kesempatan ini penulis mengucapkan terima kasih kepada :

1. Bapak, Ibu, keluarga tercinta yang telah memberikan pengorbanan, dorongan, serta motivasi penuh kepada penulis demi tercapainya cita cita yang penulis inginkan.
2. Bapak Yudi Prayudi S.Si, M.Kom selaku Ketua Jurusan Teknik Informatika sekaligus dosen pembimbing tunggal yang dengan sabar memberikan bimbingan dalam pembuatan dan penyusunan Tugas Akhir ini.

3. Pak Endro, pak Marzuki, teman-teman sekaligus saudara kost “Al Mukmin” yang telah banyak membantu, menghibur, serta memotivasi selama pengerjaan Tugas Akhir.
4. Semua pihak yang telah membantu memberikan sumbangan baik tenaga, pikiran, maupun materi.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna baik dari segi kualitas maupun kuantitas. Hal ini merupakan kewajaran, sebab penulis hanyalah seorang manusia yang penuh dengan kesalahan, dan jika ada lebihnya semata – mata karena Allah SWT, oleh karena itu saran dan kritik yang sifatnya membangun sangat penulis butuhkan demi kemajuan dimasa yang akan datang dan semoga penulisan Tugas Akhir ini bermanfaat bagi kita semua.

Amin ya Rabbal Alamin



Yogyakarta, 2011

(Puthut Galih.M)

ABSTRAKSI

Game Space Fight : The Legend Of Batu Corporation adalah sebuah permainan yang menggunakan software Adobe Flash CS 9. Game ini termasuk dalam jenis Arcade Scrolling Game dimana tugas user adalah menembakkan peluru ke sebuah sasaran dengan tepat.

Metode yang digunakan dalam pembuatan game ini yang pertama – tama adalah merencanakan jenis game yang akan dibuat, kemudian tahap kedua adalah membuat rancangan dan desain gambar, setelah itu membuat program dan menuliskan source code dengan menggunakan software Adobe Flash CS 9, langkah terakhir yaitu menguji program game yang selesai dibuat.

Penggunaan Adobe Flash CS 3 dengan metode object oriented programming dan AS 3 untuk game ini sangat tepat, karena software ini sudah mendukung pemrograman multimedia dengan tipe data yang lengkap. Game ini juga memberikan tambahan alternatif dari game berjenis *arcade scrolling game*.

DAFTAR ISI

Halaman Pengesahan Dosen Pembimbing	ii
Halaman Pernyataan Keaslian	iii
Halaman Pengesahan Dosen Penguji	iv
Halaman Persembahan	v
Kata Pengantar	vi
Abstraksi	vii
Daftar Isi	viii

BAB I PENDAHULUAN

1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3

1.6 Metode Penelitian	4
1.7 Sistematika Penulisan	4

BAB II LANDASAN TEORI

2.1 Teori Dalam Pembuatan Program Permainan	7
2.2 Tipe dan Jenis Game	9
2.3 Pemrograman Action Script	16
2.4 Konsep Pemrograman Berorientasi Objek Pada AS3	17

BAB III METODOLOGI

3.1 Analisa Kebutuhan Proses	19
3.2 Skenario Game	20
3.3 Perancangan	22
3.3.1 Analisis Input	22
3.3.2 Analisis Output	24
3.4 Kebutuhan Perangkat Lunak	25

3.5 Kebutuhan Perangkat Keras	26
-------------------------------------	----

BAB IV HASIL DAN PEMBAHASAN

4.1 Implementasi	27
4.2 Batasan Implementasi	27
4.3 Implementasi Rancangan Antar Muka	28
4.3.1 Antarmuka Menu Utama	28
4.3.2 Antarmuka Permainan	28
4.3.3 Implementasi Prosedural	29
4.4 Pembahasan	
4.4.1 Analisis Kesesuaian	51

BAB V KESIMPULAN

5.1 Kesimpulan	55
5.2 Saran	55
5.3 Daftar Pustaka	57

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini, game komputer merupakan alternatif hiburan disela – sela mengisi waktu luang, dan tentunya akan jauh lebih mengasyikkan dan menyenangkan apabila bisa merancang sendiri game tersebut dan kemudian melihat orang lain sebagai user memainkan game yang kita buat sendiri. Untuk membuat game memang tidaklah mudah, apalagi kalau melihat perkembangan game computer belakangan ini yang kaya akan kreatifitas animasi, efek-efek visual memukau dan dengan alur cerita yang menarik, dimana untuk merancang game seperti itu dibutuhkan para programmer professional yang handal di bidangnya.

Game dibuat untuk digunakan sebagai sarana untuk menghibur diri saat penat maupun saat *stress*. *Game* sendiri merupakan sebuah aplikasi sederhana yang dapat di buat dari beberapa macam *software*. Adapun *software* yang sering dipakai dalam pembuatan *game* antara lain pascal, phyton, borlan delphi maupun *software- software* yang digunakan dalam desain grafis seperti flash, *panel* studio, 3d max, atupun sejenis Adobe illustrator. Sebenarnya *game* dapat dibuat sendiri, meskipun untuk membuat *game* tersebut dikatakan sangat rumit, walaupun hanya sebuah *game* sederhana. Dengan menggunakan fasilitas pada Flash, dapat dihasilkan *game* yang kualitas gambarnya baik, karena didalam flash terdapat *tool- tool* yang sangat menarik dan

bagus. Selain itu untuk menghasilkan game yang baik dalam Flash juga dilengkapi dengan fasilitas *action Script* yang dapat digunakan untuk membuat animasi gambar dengan baik. Dengan menggunakan *fasilitas* dari Flash bukan tidak mungkin pecinta *game* dapat membuat *game* sendiri sesuai keinginannya.

Mulai tahun 2000, yaitu pada Flash versi 5, flash mulai menambahkan fasilitas *action script* yang dikenal dengan nama ActionScript 1, dan pada versi 7 diperbaharui dengan ActionScript 2, sampai pada versi terbarunya yaitu Flash 8. Pada tahun 2005, Macromedia diakuisi oleh Adobe yang memang terkenal sebagai software pengolahan visual, yaitu Adobe Flash CS 9 dan kini keluaran terbarunya, Adobe Flash CS 10. Kedua software ini adalah pengembangan dari software sebelumnya yaitu macromedia flash 8. Flash CS 9 dan 10 mempunyai fasilitas untuk mengolah action script terbaru dibanding versi – versi pendahulunya , yaitu Action Script 3.0 .

Action Script 3.0 (As3) pada flash 9 dan 10 sudah mendukung konsep OOP (Object Oriented Programming), yaitu pemrograman berorientasi obyek. Variabel dalam AS3 mempunyai tipe data yang lebih spesifik. Hal ini dimaksudkan untuk meminimalisasi kesalahan pengalokasian memori pada data. AS3 juga bersifat *case sensitive* sehingga pengaturan penulisan sintaks menjadi lebih ketat.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam pembuatan game ini adalah, bagaimana membuat rancangan dan implementasi game menggunakan metode Object Oriented Programming pada ActionScript 3 (as3).

1.3 Batasan Masalah

Batasan yang ada pada game ini adalah :

- a. Implementasi akan diterapkan pada pembuatan Rancang Bangun Game Komputer Arcade Dengan AS 3 Pada Adobe Flash CS3, Studi Kasus : Game Space Fight, The Legend Of Batu Corporation.
- b. Jumlah pemain untuk game ini adalah single player yang dimainkan secara lokal.
- c. Storyboard atau skenario jalannya permainan dibatasi hanya satu level permainan.
- d. Game tidak dilengkapi fasilitas untuk fungsi save, load game, dan pause.

1.4 Tujuan Penelitian

- a. Merancang aplikasi game sederhana menggunakan Adobe Flash
- b. Merancang game bertipe Arcade side scrolling game

1.5 Manfaat Penelitian

- a. Untuk memberikan tambahan pilihan atau alternatif dari game berjenis arcade side scrolling game.
- b. Menghasilkan Game sederhana yang tidak membutuhkan proses instalasi kedalam sistem operasi seperti game – game berat lain pada umumnya.

1.6 Metode Penelitian

Metode yang digunakan dalam pembuatan game ini adalah sebagai berikut :

- a. Analisis Requirement, yaitu merencanakan jenis (tipe) game yang akan dibuat, dalam tulisan ini yaitu tipe arcade scrolling game.
- b. Design, yaitu membuat rancangan dan desain obyek yang akan dibuat, seperti : karakter dalam game, background, suara.
- c. Coding, yaitu membuat program dan menuliskan source code dengan menggunakan software aplikasi Adobe Flash cs 9 dengan metode object oriented programming.
- d. Uji Program, adalah langkah terakhir setelah program selesai dirancang sepenuhnya.

1.7 Sistematika Penulisan

Untuk mempermudah dalam memahami isi laporan penelitian, maka dikemukakan sistematika penulisan laporan agar menjadi satu kesatuan yang utuh. Pada laporan penelitian ini terdiri dari tiga bagian pokok, pertama yaitu bagian pendahuluan, kemudian yang kedua bagian tubuh atau isi laporan, dan bagian akhir yaitu kesimpulan dan saran. Bagian pertama memuat halama judul, lembar pengesahan, halaman motto, kata pengantar, abstraksi, daftar isi, tabel – tabel, dan daftar gambar. Pada bagian tubuh laporan berisi inti dari laporan penelitian yang dipisah – pisah menjadi beberapa bagian / bab, yakni :

1. Bab I : Pendahuluan

Bab ini berisi tentang latar belakang masalah, materi, tujuan, manfaat penulisan, dan juga sistematika penulisan.

2. Bab II : Landasan Teori

Memuat teori – teori yang berhubungan dengan penelitian.

3. Bab III : Metodologi

Bagian ini memuat uraian langkah – langkah penyelesaian, uraian tentang metode analisis kebutuhan perangkat lunak yang dipakai, serta hasil analisis kebutuhan perangkat lunak yang berupa analisis kebutuhan proses, analisis kebutuhan masukan, kebutuhan perangkat lunak, perangkat keras, kebutuhan antar muka, dan yang terakhir yakni tahap perancangan system.

4. Bab IV : Hasil dan Pembahasan

Bab ini membahas tentang implementasi hasil kerja system serta analisis kinerja dari perangkat lunak. Bagian hasil memuat tentang hasil dan bagaimana hasil tersebut dicapai. Sedangkan bagian pembahasan memuat pembahasan hasil perancangan program yang diperoleh. Pembahasan dapat berupa uraian tentang mengapa hasil diperoleh, kelebihan dan kelemahan apa dari system yang dibuat.

5. Bab V : Kesimpulan dan Saran

Merupakan bagian akhir dari inti laporan, yakni berisi kesimpulan – kesimpulan yang merupakan rangkuman dari hasil analisis kerja pada bagian –

bagian sebelumnya serta saran yang perlu diperhatikan berdasarkan keterbatasan yang ditemukan dalam sistem.



BAB II

LANDASAN TEORI

2.1 Teori Dalam Pembuatan Program Permainan

Urutan pembuatan dan pengembangan program permainan adalah sebagai berikut [HEN89]:

- a. Menentukan jenis / tipe permainan apa yang ingin dibuat.

Penentuan ini sebagai dasar sebelum mulai bekerja sampai didapatkan ide / gagasan yang tepat sebelum merancang program permainan yang akan dibuat.

- b. Pendefinisian model permainan dan tujuannya

Setelah mendapatkan gagasan atau ide yang tepat untuk merancang game , langkah selanjutnya yaitu menerapkan ide tersebut dalam bentuk model permainan . Pada tahap ini *blueprint* dari perancangan game yang akan dibuat dituliskan secara jelas dengan menentukan *story board* atau jalan cerita dari game yang akan dibuat, misalnya menentukan misi – misi yang akan dijalankan pada tiap level permainan, kemudian diimplementasikan dalam diagram *Flow Chart* .

- c. Informasi untuk pemain

Informasi bagi pemain (user) ketika game sedang berjalan sebagai statistic jalannya permainan , misalnya tampilan score, indicator nyawa, batas waktu, dan yang lainnya.

d. Penghargaan

Setelah user selesai bermain, tentunya untuk memotivasi agar mau bermain lagi maka perlu ditampilkan ucapan penghargaan berupa ucapan selamat apabila user memenangkan game tersebut, kemudian menampilkan namanya setelah selesai bermain, serta menampilkan 5 nama dengan nilai tertinggi dalam game. Apabila kalah pun maka akan ditampilkan tulisan supaya bisa mengulang permainan itu dari awal.

e. Variasi

Untuk membuat supaya user tidak bosan untuk memainkan game, maka diperlukan variasi misalnya tampilan desain antar muka dibuat semenarik mungkin dan kemampuan game dalam berinteraksi dengan pemain lebih interaktif. Tetapi variasi ini pun tidak boleh terlalu berlebihan, karena akan membuat alur permainan menjadi tidak konsisten.

f. Tingkat kesulitan

Dalam game misalnay terdapat tingkat kesulitan yang dapat dipilih sesuai apa yang user inginkan, sehingga membuat user akan merasa tertantang dan akan merasa senang jika bisa melewati tingkat kesulitan yang akan dihadapi.

g. Pastikan bahwa game tersebut dapat dimainkan

Satu hal yang sangat penting adalah, game yang dirancang tentunya harus dapat dimainkan, tampilan visual, grafik, dan suara pun harus didesain sebaik dan semenarik mungkin.

h. Rancang program semaksimal mungkin

Gunakan tehnik pemrograman yang sesuai dalam pembuatannya dan buatlah supaya program mudah dimodifikasi.

i. Uji Program

Program permainan yang telah selesai dibuat harus diuji. Pertama tentunya oleh perancangnya sendiri untuk menentukan kesalahan logis yang mungkin terjadi. Kedua oleh orang lain, untuk mengetahui kesalahan logis yang tidak terlihat oleh perancangnya, serta memberikan saran atau ide untuk dapat mengembangkan program yang akan dibuat selanjutnya.

2.2 Tipe dan Jenis Game

Tipe dan jenis – jenis game terbagi dalam beberapa kelompok, antara lain yaitu:

- a. Berdasar jenis platform atau alat yang digunakan, seperti:
- *PC Games*, yaitu game yang dimainkan dalam *PC (Personal Computer)*.
 - *Console Games*, yaitu game yang dimainkan menggunakan console tertentu, seperti *PlayStation*, *Nintendo Wii*, atau *Xbox*.
 - *Handheld Games*, yaitu game yang dimainkan dengan console khusus video game yang dapat dibawa kemana saja, seperti *PSP*, dan *Nintendo DS*.
 - *Mobile Games*, yaitu game yang terdapat di dalam *mobile Phone* atau *PDA*.

b. Berdasarkan “Genre” permainannya, yaitu:

- Action – Shooting

Game bertema aksi yang memerlukan kecepatan refleks yang tinggi, yang termasuk jenis game ini dibagi lagi menjadi beberapa kelompok:

- *First Person Shooter (FPS)*, seperti *Conter Strike*, *Call Of Duty*, *Doom*

- *Drive n’Shoot*

Game yang menggunakan unsur simulasi kendaraan tetapi tetap dengan misi untuk menembak dan menghancurkan lawan, seperti :

Spy Hunter, *Road Rash*.

- *Shoot em’up*, contohnya : *Raiden*, *1942*, *jets n guns*

- *Beat em’up*

Game pertarungan yang dipadu dengan petualangan, contohnya : *Double Dragon*, *Shinobi*.

- *Light Gun Shooting*

Game simulasi yang menggunakan alat bantu senjata, sehingga seolah-olah pemain berada di lokasi kejadian, contohnya: *Virtua Cop*, *Time Crisis*.

- *Fighting*

Jenis game pertarungan yang memerlukan hafalan jurus dalam memainkannya, dan tiap level akan bertemu karakteristik lawan yang

berbeda-beda, contohnya yaitu : *Street Fighter, Tekken, Mortal Kombat, King Of Fighter.*

- *Action – Adventure*

Jenis game yang memadukan unsur aksi dan petualangan, seperti misalnya memasuki gua bawah tanah, bergantung dari satu pohon ke pohon yang lain, mengambil permata didalam gua, atau mencari kunci yang digunakan untuk berpindah ke level berikutnya. Yang termasuk game ini contohnya : *Tomb Rider, Prince of Persia, Grand Theft Auto.*

- *Adventure*

Berbeda dengan game action – adventure, game ini murni lebih menekankan pada jalan cerita dan kemampuan berfikir pemain dalam menganalisa tempat secara visual, memecahkan sebuah teka – teki, maupun menyimpulkan rangkaian peristiwa. Contoh : *King Quest, Space Quest, Monkey Island.*

c. *Simulation*

Game jenis ini seringkali menggambarkan dunia di dalamnya sedekat mungkin dengan dunia nyata dan memperhatikan dengan detil berbagai faktor. Dari mencari jodoh dan pekerjaan, membangun rumah, gedung hingga kota, mengatur pajak dan dana kota hingga keputusan memecat atau menambah karyawan. Dunia kehidupan rumah tangga sampai bisnis membangun konglomerasi, dari jualan limun pinggir jalan hingga membangun laboratorium cloning. Game jenis ini membuat pemain harus berpikir untuk mendirikan,

membangun dan mengatasi masalah dengan menggunakan dana yang terbatas.

Contoh: *Sim City, The Sims, Tamagotchi.*

d. *RPG (Role Playing Game)*

Game jenis ini sesuai dengan terjemahannya, bermain peran, memiliki penekanan pada tokoh/peran perwakilan pemain di dalam permainan, yang biasanya adalah tokoh utamanya, dimana seiring user memainkannya, karakter tersebut dapat berubah dan berkembang ke arah yang diinginkan pemain (biasanya menjadi semakin hebat, semakin kuat, semakin berpengaruh, dll) dalam berbagai parameter yang biasanya ditentukan dengan naiknya **level**, baik dari status kepintaran, kecepatan dan kekuatan karakter, senjata yang semakin sakti, ataupun jumlah teman maupun makhluk peliharaan. Secara kebudayaan, pengembang game Jepang biasanya membuat Role Playing Game (**RPG**) ke arah cerita linear yang diarahkan seolah karakter kita adalah tokoh dalam cerita itu, seperti *Final Fantasy, Dragon Quest dan Xenogears*. Sedangkan pengembang game **RPG** Eropa, cenderung membuat karakter kita bebas memilih jalan cerita sendiri secara non-linear, seperti *Ultima, Never Winter Nights, Baldur's Gate, Elder Scroll, dan Fallout*.

e. *Strategy*

Game jenis strategi, layaknya bermain catur, justru lebih memerlukan keahlian berpikir dan memutuskan setiap gerakan secara hati-hati dan terencana. Video game strategi biasanya memberikan pemain atas kendali tidak hanya satu orang tapi minimal sekelompok orang dengan berbagai jenis tipe kemampuan, sampai

kendaraan, bahkan hingga pembangunan berbagai bangunan, pabrik dan pusat pelatihan tempur, tergantung dari tema ceritanya. Pemain game strategi melihat dari sudut pandang lebih meluas dan lebih kedepan dengan waktu permainan yang biasanya lebih lama dan santai dibandingkan game action. Unsur-unsur permainannya biasanya berkisar sekitar, prioritas pembangunan, peletakan pasukan, mencari dan memanfaatkan sumberdaya (uang, besi, kayu,minyak,dll), hingga ke pembelian dan peng-upgrade-an pasukan atau teknologi. Game ini terbagi lagi atas :

- *Real time Strategy*, game berjalan dalam waktu sebenarnya dan serentak antara semua pihak dan pemain harus memutuskan setiap langkah yang diambil saat itu juga berbarengan mungkin saat itu pihak lawan juga sedang mengeksekusi strateginya. Contoh: *Starcraft, Warcraft, Command and Conquer, Age of Empires, Stronghold Crusader*.
- *Turn based Strategy*, game yang berjalan secara bergiliran, saat pemain mengambil keputusan dan menggerakkan pasukan, saat itu pihak lawan menunggu, begitu pula sebaliknya, layaknya catur. Contoh: *Front Mission, Super robot wars, Final Fantasy tactics, Heroes of might and magic, Master of Orion*.

f. *Puzzle*

Game jenis ini sesuai namanya berintikan mengenai pemecahan teka-teki, baik itu menyusun balok, menyamakan warna bola, memecahkan perhitungan matematika, melewati labirin, sampai mendorong-dorong kota masuk ke

tempat yang seharusnya, itu semua termasuk dalam jenis ini. Sering pula permainan jenis ini adalah juga unsur permainan dalam video game petualangan maupun game edukasi. *Tetris, Minesweeper, Bejeweled, Sokoban dan Bomberman.*

g. *Vehicle Simulation*

Game jenis ini memberikan pengalaman atau interaktifitas sedekat mungkin dengan kendaraan yang aslinya, meskipun terkadang kendaraan tersebut masih eksperimen atau bahkan fiktif, tapi ada penekanan khusus pada detil dan pengalaman realistik menggunakan kendaraan tersebut. Contoh : *Need For Speed, Flight Simulator, Daytona, Grand Turismo.*

h. *Sport*

Game berjenis olahraga, sehingga pemain merasa seakan – akan bermain langsung mengikuti pertandingan tersebut. Contohnya : *FIFA 2010, Championship Manager, Winning Eleven, Pro Evolution Soccer.*

Selain jenis diatas, masih ada lagi jenis game dengan kategor – kategori lain, misalnya :

a. *Multiplayer Game Online*

Game yang sedang trend di Indonesia bahkan dunia, menjadi salah satu titik balik mengapa dunia game dan internet di Indonesia dapat berkembang. Dan karena dimainkan online dan dengan sistem pembayaran menggunakan voucher, pembajakan sudah tidak menjadi masalah lagi. Game yang dapat dimainkan secara bersamaan oleh lebih dari 2 orang (bahkan dapat mencapai

puluhan ribu orang dalam satu waktu) membuat pemain dapat bermain bersama dalam satu dunia virtual dari sekedar chatting hingga membunuh naga bersama teman yang entah bermain di mana. Umumnya permainan tipe ini dimainkan di PC dan bertema RPG, walau ada juga yang bertema music atau action. Contoh: *Ragnarok online, O2jam, World of Warcraft, Ayo Dance, Lineage, Rose online.*

b. *Casual Games*

Sesuai namanya, game yang casual itu tidak kompleks, mainnya rileks dan sangat mudah untuk dipelajari (bahkan cenderung langsung bisa dimainkan). Jenis ini biasanya memerlukan spesifikasi komputer yang standar pada jamannya dan ukurannya tidak lebih dari 100 MB karena biasanya dapat di download versi demo-nya di website resminya. Genre permainannya biasanya puzzle atau action sederhana dan umumnya dapat dimainkan hanya menggunakan mouse (biasanya game lain menggunakan banyak tombol tergantung game-nya). Contoh: *Diner Dash, Sally Salon, Bejeweled, Zuma, Feeding Frenzy, Insaniquarium.*

c. *Education Games*

Game jenis ini dibuat dengan tujuan spesifik sebagai alat pendidikan, entah untuk belajr mengenal warna untuk balita, mengenal huruf dan angka, matematika, sampai belajar bahasa asing. Developer yang membuatnya, harus memperhitungkan berbagai hal agar game ini benar-benar dapat mendidik, menambah pengetahuan dan meningkatkan ketrampilan yang memainkannya. Target segmentasi pemain harus pula disesuaikan dengan tingkat kesulitan dan

design visual ataupun animasinya. Contoh edugames : *Bobi Bola, Dora the explorer, Petualangan Billy dan Tracy.*

d. Advertising Games

Game sekarang telah menjadi sarana beriklan atau membangun *brand-awareness* yang efektif. Baik melalui internet maupun di mainkan di event-event suatu acara, advertising games terasa semakin dibutuhkan untuk menjaring calon konsumen bagi produk yang menggunakan advertising games ini. Contoh produk di indonesia yang membuat advertising games: *A-Mild, Rexona teens, Axe.*

2.3 Pemrograman Action Script

Di dalam perancangan game komputer, actionscript mutlak sangat diperlukan untuk memanipulasi data atau grafis supaya terlihat suatu kreatifitas dan interaktifitas yang menarik . Adapun actionscript merupakan salah satu pemrograman berorientasi objek yang tata bahasa , dan struktur pemrogramannya mirip dengan bahasa pemrograman C ++.

Walaupun kelihatannya actionscript terlihat sederhana, namun tetap saja dibutuhkan latihan rutin untuk bisa menguasainya, karena dalam merancang suatu game dibutuhkan kemampuan berlogika yang cukup tinggi. Pembuatan game memancing kita untuk berfikir bagaimana aturan game itu dijalankan oleh program.

Action Script merupakan bahasa pemrograman yang digunakan dalam Flash. Penggunaannya mempunyai tujuan :

- a. Memberikan kebebasan berkreasi bagi si perancang program
- b. Membuat animasi yang interaktif dan menarik
- c. Dapat menampilkan animasi tertentu

2.4 Konsep Pemrograman Berorientasi Objek pada AS3

Dalam konsep pemrograman modern dikenal satu konsep yang semakin mirip dengan kehidupan dunia nyata yaitu *Pemrograman Berorientasi Objek* atau *Object Oriented Programming (OOP)*. Disebut demikian, karena semua komponen pemrograman yang berupa objek – objek saling berinteraksi untuk mencapai suatu tujuan.

OOP adalah cara pemrograman yang mendasarkan segala sesuatunya pada objek . Sebuah aplikasi disusun oleh rangkaian berbagai macam objek, bahkan aplikasi itu sendiri bisa juga merupakan sebuah objek.

2.5 Pengenalan Object dan Class Pada AS3

Objek dalam OOP memiliki attribute dan method. Class adalah kumpulan objek yang mempunyai attribute dan method yang sama, adapun attribute diperlukan untuk mendukung fungsi pada method. Konsep class dan object dapat dianalogikan dengan seseorang yang bernama Budi sebagai contoh. budi ini adalah termasuk dalam class manusia , mempunyai attribute berupa mata lentik, hidung mancung, dan mulut lebar . Dengan atribut yang dimilikinay itu , Budi biasa melakukan aktifitasnya (method) seperti melihat dengan mata, bernafas dengan hidung , atau makan dengan

menggunakan mulutnya. Contoh lainnya adalah kendaraan sebagai sebuah class . Class kendaraan ini tentunya memiliki sifat – sifat tertentu, seperti memiliki roda , pintu, dan mesin. Selain memiliki sifat, class mobil juga dapat melakukan sesuatu seperti berjalan maju, mundur, belok kanan, kiri, dan sebagainya. Semua yang disebutkan merupakan struktur dari sebuah class kendaraan. Dengan adanya class kendaraan tersebut , maka kita dapat membuat sebuah instance dari class kendaraan, misalny sedan. Semua struktur yang dimiliki class kendaraan akan dimiliki pula oleh objek sedan sehingga sedan akan memiliki roda, pintu mesin, dan juga dapat bergerak kemana saja. Pada sebuah objek tidak semua attribute dapat diakses dari luar karena sebuah attribute class memiliki access specifier, yaitu *private* dan *public*. Attribute *private* adalah attribute yang dapat diakses oleh class tersebut, sedangkan class lain tidak dapat mengakses attribute ini. Attribute *public* dapat diakses dari luar class , misalny Budi memiliki attribute nomor telepon, PIN ATM atau attribute lain yang hanya diketahui oleh Budi sendiri, yang tidak boleh diketahui oleh object lain untuk alasan keamanan.

BAB III

METODOLOGI

Metodologi adalah penerapan langkah – langkah dari metode penelitian untuk menyelesaikan pembuatan game ini.

3.1 Analisa Kebutuhan Proses

Beberapa tahapan proses dalam pembuatan game ini mencakup:

a. *Analisis Requirement*

Tahap pertama dari analisis kebutuhan proses adalah menentukan jenis game apa yang akan dibuat, dalam penulisan ini jenis gamenya adalah *Arcade Scrolling Game*.

b. *Design*

Tahap berikutnya yaitu membuat rancangan komponen – komponen game yang dibutuhkan, dimana komponen – komponen itu meliputi : Rancangan gambar pesawat, rancangan gambar pesawat musuh, rancangan animasi ledakan, rancangan tempat score, dan juga latar belakang game.

c. *Coding*

Tahap ketiga setelah proses desain, yaitu menentukan algoritma, mendefinisikan prosedur dan fungsi – fungsi yang dibutuhkan dalam pembuatan game kemudian menuliskan source code di dalam prosedur dan fungsi tersebut, untuk mengendalikan jalannya permainan dengan menggunakan fasilitas ActionScript 3.0 pada software Adobe Flash CS3.

d. *Uji Program*

Tahap terakhir dari seluruh proses setelah game selesai dibuat, adalah uji coba terhadap game tersebut, dan juga untuk mengetahui dimana letak kesalahan dan kekurangannya.

3.2 Skenario Game

1. Tampilan Menu Utama

Tampilan antar muka dalam menu utama terdiri dari beberapa tombol yang terdiri dari Tombol **START** .

2. Tampilan Gambar Latar pada game (background)

Didalam arena permainan , stage akan menampilkan background dari game tersebut , dimana di dalamnya terdapat gambar berlatar awan berwarna biru.

3. Pergerakan pesawat utama

Pesawat utama akan bergerak ke atas , ke bawah , kanan , dan kiri dengan input keyboard Space untuk menembak dan panah untuk melakukan pergerakan.

4. Pergerakan pesawat musuh

Pesawat musuh akan muncul dan bergerak secara variatif dengan perintah actionscript.

5. Pesawat utama menembaki musuh

Pesawat utama akan bergerak sambil mengeluarkan tembakan dengan menggunakan input keyboard space menghancurkan pesawat musuh yang datang silih berganti , kemudian musuh akan hancur apabila nilai health nya sudah bernilai 0.

6. Musuh menembaki pesawat utama

Pesawat musuh juga bisa menembaki pesawat utama yang juga akan mengurangi nilai health dari pesawat utama , lalu akan hancur apabila nilai healthnya 0. Ada dua karakteristik pesawat musuh , musuh yang pertama akan menembakkan peluru bulat merah dari kanan dan kiri sayapnya , apabila mendekati pesawat jagoan . Sedangkan musuh yang kedua akan menembakkan peluru dari depan badan pesawat tersebut

7. Pengumpulan nilai / score

Setiap menghancurkan satu musuh , maka score akan bertambah sebanyak 100 point apabila mengenai musuh pertama dan 500 point untuk musuh kedua.

8. Berkurangnya nilai armor / nyawa pada pesawat utama

Tembakan musuh yang pertama akan menyebabkan berkurangnya nilai armor sebanyak 1 point , dan tembakan musuh kedua berkurang 5 point . Nilai armor pada pertama kali permainan dijalankan adalah 100 . apabila mencapai angka nol , maka pesawat jagoan akan hancur dan permainan pun berakhir.

9. Bertambahnya nilai kills

Pesawat utama utama ketika berhasil menghancurkan tiap satu pesawat musuh , maka nilai kills akan terus bertambah satu , mengindikasikan jumlah pesawat musuh yang terbunuh apabila terkena tembakan dari pesawat jagoan.

10. Munculnya ledakan kecil pada pesawat

Ketika terkena tembakan , baik pada pesawat jagoan ataupun pada pesawat musuh, maka akan muncul animasi ledakan kecil berwarna merah yang mengindikasikan bahwa pesawat terkena tembakan.

3.3 Perancangan

Perancangan pada game ini terdiri dari beberapa Analisis, yaitu :

3.3.1 Analisis Input

Komponen masukan utama mencakup :

1. Tampilan Menu Utama

Di dalam tampilan menu ini , terdapat satu tombol Start sebagai penghubung untuk melanjutkan ke frame selanjutnya apabila ditekan.

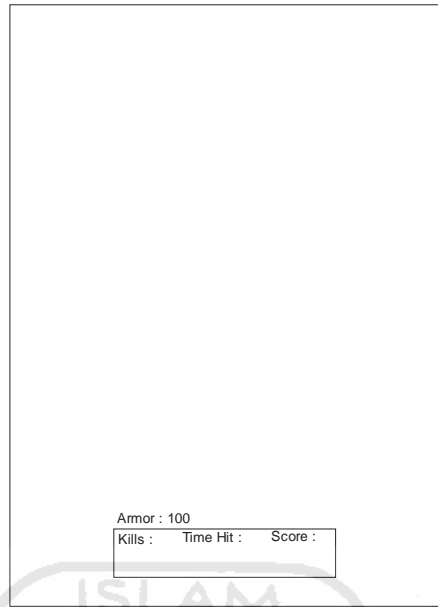
2. Tampilan Game

Setelah tombol start ditekan , maka user akan memasuki frame selanjutnya , yaitu tampilan pada saat game berjalan. Adapun rancangan komponen – komponen yang ada di dalam frame ini meliputi :

- a. Rancangan gambar pesawat utama
- b. Rancangan gambar pesawat musuh 1 dan 2
- c. Rancangan gambar animasi ledakan pesawat
- d. Rancangan gambar score board
- e. Rancangan gambar latar belakang



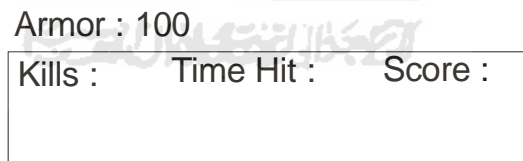
Gambar 3.1 Menu Utama



Gambar 3.2 Game

3.3.2 Analisis output

Analisis output mencakup pencatatan nilai armor , kills , time hit , serta score pada saat game dimainkan.



Gambar 3.3 Tampilan Score

a. Tampilan Score

Adalah tampilan indicator pesawat jagoan yang berupa teks input . Ada beberapa proses yang ditampilkan , yaitu :

b. Proses penambahan nilai atau score ketika pesawat menembak

Disini ditampilkan indicator apabila pesawat jagoan menembak dan tembakan tersebut mengenai pesawat musuh , maka nilai atau score akan bertambah 100 point jika mengenai musuh pertama , dan 500 point jika mengenai musuh kedua.

- c. Proses pengurangan nilai armor / nyawa ketika pesawat tertembak pesawat musuh

Apabila peluru dari musuh pertama mengenai pesawat pertama , maka nilai armor akan berkurang sebanyak 1 nilai , dan apabila terkena peluru dari musuh kedua maka nyawa akan berkurang sebanyak 5 point ,kemudian jika menabrak musuh maka nyawa juga akan berkurang sebanyak 5 point.

- d. Proses penambahan nilai kills

Ketika pesawat jagoan berhasil membunuh satu musuh , maka nilai kills pada tampilan text kills akan bertambah satu .

3.4 Kebutuhan Perangkat Lunak

Perangkat lunak (software) yang dipakai pada pembuatan game ini adalah :

- a. Adobe Flash CS 3 keatas

Software ini adalah release terbaru dari Flash, dimana versi terbaru ini sudah mendukung bahasa pemrograman actionscript 3.0 (as3), yang merupakan pengembangan dari actionscript sebelumnya. Fasilitas yang terdapat di

program ini juga sudah lebih lengkap dari versi pendahulunya dan lebih mudah untuk digunakan.

b. Adobe Photoshop CS 3

Software ini digunakan sebagai pengolah grafis untuk membuat komponen-komponen yang dibutuhkan dalam game, seperti pembuatan rancangan pesawat, peluru, dan latar belakang game.

c. Flash Player versi 9.0 atau 10.0

Merupakan file eksekusi yang digunakan setelah game ini selesai dibuat, sehingga user yang akan memainkan game ini tidak perlu untuk menginstal software adobe flash untuk bisa menjalankan program game ini.

3.5 Kebutuhan Perangkat Keras

Kebutuhan minimal Hardware (Perangkat Keras) yang dibutuhkan :

- a. Pentium IV 1,7 Ghz
- b. RAM 256 Mbytes
- c. Soundcard
- d. Kartu Grafis (Graphic card) 64 Mb
- e. Mouse dan Keyboard

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi

Implementasi merupakan proses dimana rancang bangun software telah dilihat dan diterjemahkan ke dalam bahasa pemrograman yang sesungguhnya. Pada bagian ini juga memuat dokumentasi dan penjelasan antarmuka serta prosedur – prosedur yang ada di dalam program permainan ini.

4.2 Batasan Implementasi

- a. Game ini hanya mencakup 1 level , tidak ada tingkat kesulitan yang bisa dipilih.
- b. Hanya satu pemain (single user) yang bisa memainkan game ini.
- c. Input yang digunakan adalah tombol panah untuk pergerakan pesawat dan tombol space.
- d. Tidak ada fasilitas save game dan load game untuk menyimpan dan memanggil kembali game yang telah di save.

4.3 Implementasi Rancangan Antarmuka

4.3.1 Antarmuka Menu Utama

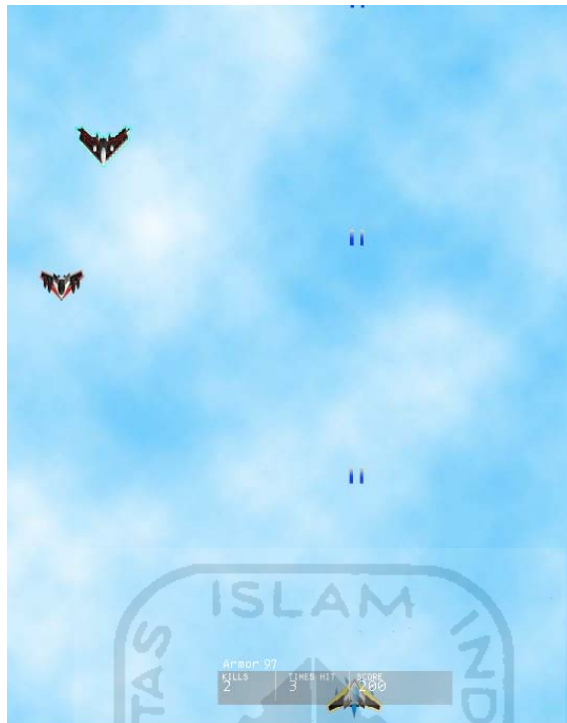
Program permainan ini pada dasarnya terdiri dari dari 2 buah frame utama , yaitu frame pertama yang selanjutnya disebut menu utama dan frame kedua yaitu dimana pada halaman ini proses permainan dan aturan permainan dimulai.



Gambar 4.1 Menu Utama

4.3.2 Antarmuka Permainan

Ketika tombol start ditekan , user akan masuk ke frame selanjutnya , yaitu frame dimana user bisa langsung bermain game.



Gambar 4.2 Game

4.3.3 Implementasi Prosedural

Adalah implementasi pada skenario game, meliputi fungsi – fungsi prosedural yang diterapkan di dalam masing- masing class sebagai berikut :

1. Prosedur Pada Menu Utama

`stop();`*//menghentikan animasi pada saat menu utama muncul agar game tidak langsung dijalankan*

`tombolStart.addEventListener(MouseEvent.CLICK,klikStart);`*//memasukkan input mouse pada stage di tombol startgame*

`function klikStart(Event:MouseEvent){`

`gotoAndStop("play");`*//jika tombol start ditekan,maka akan langsung ke frame berikutnya dan mulai bermain*

2. Prosedur pada class utama (SkyJustice.as)

Prosedur dalam class ini meliputi fungsi:

- a. Fungsi startSkyJustice, adalah fungsi untuk menempatkan komponen-komponen dalam game, yaitu pesawat, kedua karakter musuh, dan scoreboard.

- package

```
{
//mengimpor komponen-komponen movieclip, stage, event, system,
input mouse,suara ke dalam arena permainan
import flash.display.MovieClip;
import flash.display.Stage;
import flash.events.Event;
import flash.system.System;
import flash.events.MouseEvent;
import flash.media.Sound;
import flash.media.SoundChannel;
public class SkyJustice extends MovieClip
{
public static var musuh:Array=new Array();
public static var armorPesawat:int;
public var pesawat:Pesawat;
private var scoreHUD:ScoreHUD ;
public function startSkyJustice():void//fungsi inisialisasi secara
umum
{
//menempatkan movieclip pesawat di dalam stage
pesawat=new Pesawat(stage);//
addChild(pesawat);//menempatkan pesawat
pesawat.x=225;//posisi x pesawat dalam stage
pesawat.y=600;//posisi y pesawat dalam stage
```



```

scoreHUD = new ScoreHUD(stage); //menempatkan layar nilai
stage.addChild(scoreHUD); //kemudian menampilkannya
scoreHUD.x=200;//posisi x layar nilai dalam stage
scoreHUD.y=650;//posisi y layar nilai dalam stage
pesawat.addEventListener("hit",pesawatHit);
addEventListener(Event.ENTER_FRAME,loop);//menambahkan
fungsi baru, yaitu fungsi loop
}

```

b. Fungsi loop

Adalah fungsi untuk menempatkan ke dua musuh ke dalam stage, yang *source code* nya adalah sebagai berikut :

- ```

private function loop(e:Event):void
{
if (Math.floor(Math.random() * 100) == 2)
//menentukan bilangan random keluarnya pesawat musuh 1
{
Var musuh1:Musuh1=new Musuh1(stage,pesawat);
musuh1.addEventListener(Event.REMOVED_FROM_STAGE,
hilangkan);//menambahkan fungsi hilangkan pada class untuk
menghilangkan musuh apabila terbunuh
musuh1.addEventListener("killed",musuhTerbunuh);//menambahkan
frame label killed pada pesawat musuh1
musuh.push(musuh1);//memanggil movieclip musuh1
stage.addChild(musuh1);// kemudian menampilkannya di stage
}
if (Math.floor(Math.random() * 450) == 5)
//menentukan bilangan random keluarnya pesawat musuh 2
{
Varmusuh2:Musuh2=new Musuh2(stage,pesawat);// menambahkan
fungsi hilangkan pada class untuk menghilangkan musuh apabila
terbunuh

```

```

musuh2.addListener(Event.REMOVED_FROM_STAGE,
hilangkan);// menambahkan fungsi hilangkan pada class untuk
menghilangkan musuh apabila terbunuh
musuh2.addListener("killed",musuhTerbunuh);// menambahkan
frame label killed pada pesawat musuh1
musuh.push(musuh2);// memanggil movieclip musuh1
stage.addChild(musuh2);// kemudian menampilkannya di stage
}
}

```

c. Fungsi musuhTerbunuh, adalah fungsi yang digunakan sebagai tampilan dari terbunuhnya musuh, yang *source code* nya adalah sebagai berikut :

- ```

private function musuhTerbunuh(e:Event) : void
{
scoreHUD.updateKills(1); // bertambah 1 point apabila berhasil
membunuh tiap satu pesawat musuh
scoreHUD.updateScore(e.currentTarget.points); //menambahkab
score tiap menghancurkan satu pesawat musuh
}
private function hilangkan(e:Event):void//fungsi menghilangkan
musuh apabila tertembak atau keluar dari stage
{
musuh.splice(musuh.indexOf(e.currentTarget));
}
private function pesawatHit(e:Event):void//fungsi apabila pesawat
terkena tembakan atau menabrak musuh
{
scoreHUD.updateHits(1);//nilai hits bertambah 1 point bila terkena
tembakan musuh
scoreHUD.updateArmor(armorPesawat);//hits bertambah, maka
armor pada pesawat jagoan otomatis berkurang
if(armorPesawat==0)//jika armor sudah mencapai angka nol, dan
pesawat jagoan hancur,maka semua komponen pada game akan
dihilangkan semua,permainan berakhir
{
removeEventListener(Event.ENTER_FRAME,loop);
}
}

```

3. Prosedur pada proses pergerakan pesawat utama

Prosedur dalam class ini meliputi fungsi:

a. Fungsi *Gerak*

Prosedur Fungsi *Gerak* adalah fungsi pergerakan *pesawat*, yaitu fungsi untuk memasukkan input keyboard sebagai pergerakan pesawat dengan tombol panah pada keyboard. Berikut *source code* pada fungsi gerak :

- ```
public function Gerak(e:Event):void{
 if (currentLabel != "hancur")//menambahkan frame label hancur
pada movieclip pesawat
 {
 if (tombolBawahDitekan)
 {
 vy +=speed;
//jika tombol panah bawah ditekan, maka pesawat akan bergerak
kebawah mengikuti sumbu y
 }

 else if (tombolAtasDitekan)
//jika tombol panah atas ditekan, maka pesawat akan bergerak
keatas mengikuti sumbu y
 {
 vy -=speed;
 }
 else
 {
 vy *=friction;//efek pergerakan pesawat
 }
 if (tombolKiriDitekan)
 {
 vx -=speed;
//jika tombol panah kiri ditekan, maka pesawat akan bergerak
kekiri mengikuti sumbu x negatif
 }
 if (tombolKananDitekan)
 {
 vx +=speed;
//jika tombol panah kanan ditekan, maka pesawat akan bergerak
kekanan mengikuti sumbu x positif
 }
 }
}
```

```

else
{
vx *=friction;//efek pergerakan pesawat
}

```

b. Fungsi `tembakPeluru`

Prosedur Fungsi `tembakPeluru` adalah fungsi untuk menembakkan laserBiru pada pesawat kearah musuh dengan input tombol space pada keyboard dengan *source code* sebagai berikut :

- `if (tombolSpaceDitekan)`

```

{
 tembakPeluru();//menambahkan fungsi tembak peluru
}
private function tembakPeluru():void//fungsi untuk menembak
{
 if(bisaTembak);
 {
 batasStage.addChild(new LaserBiru(batasStage,x+vx,y-35));//
 menampilkan movieClip laserBiru pada saat pesawat menembak
 dengan input tombol space
 bisaTembak=false;
 jedaTembak.start();//menambahkan fungsi jeda waktu pada saat
 terakhir pesawat menembak dengan tembakan berikutnya
 }
 private function jedaTembakHandler(e:TimerEvent):void
 {
 bisaTembak=true;
 }
}

```

c. Fungsi `TakeHit`

Prosedur Fungsi `TakeHit` Adalah fungsi dengan *source code* yang mengeksekusi apabila pesawat terkena tembakan atau menabrak musuh.

- `public function takeHit():void`

```

{
 if(currentLabel != "hancur")
 {

```

```

SkyJustice.armorPesawat=armor;
if (armor==0)//jika armor pesawat sudah mencapai nol
{
dispatchEvent(new Event("hit"));
gotoAndPlay("hancur");//memainkan frame label "hancur"
//armor-=value;
return;
}
dispatchEvent(new Event("hit"));
}

```

#### d. Fungsi Hilangkan

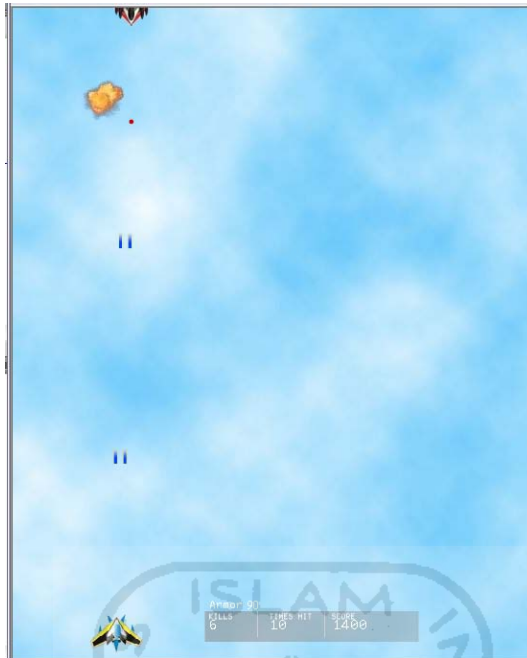
Prosedur Fungsi Hilangkan adalah fungsi dengan source code untuk mengeksekusi menghapus movieclip pesawat bila hancur.

- if (currentLabel == "hancurTotal")*//bila armor pesawat sudah 0, maka label pada frame adalah "hancur total" dan memainkan movieclip ledakan*

```

{
hilangkan();//setelah meledak, maka ditambahkan fungsi selanjutnya, yaitu fungsi hilangkan untuk menghapus movieclip pesawat karena permainan sudah berakhir
}
private function hilangkan():void
{
removeEventListener(Event.ENTER_FRAME, Gerak);
if(parent.contains(this))
batasStage.removeChild(this);
trace("gameOver");
}

```



**Gambar 4.3 Pesawat Menembak**

4. Prosedur pada pergerakan pesawat musuh 1

Prosedur dalam class ini meliputi fungsi :

a. Fungsi *Musuh1*

Fungsi *Musuh1*, adalah fungsi untuk pergerakan pesawat secara acak.

- ```

public function Musuh1(stageRef:Stage,target:Pesawat):void
{
    stop();//menghentikan movieclip supaya pesawat musuh tetap pada
    posisi pada saat belum meledak
    this.stageRef=stageRef;
    this.target=target;//menentukan sasaran yang akan dituju, yaitu
    pesawat jagoan
    x = Math.random() * stageRef.stageWidth;//posisi x pada saat
    pesawat musuh1 muncul
    y = -20;//posisi y pada saat pesawat musuh1 muncul
    addEventListener(Event.ENTER_FRAME,loop);//menambahkan
    fungsi baru, fungsi loop
}

```

b. Fungsi *loop*

Didalam fungsi ini, ditambahkan fungsi-fungsi baru, yaitu fungsi *removeself* untuk menghilangkan movieclip pesawat musuh, fungsi *fireWeapon*, yang merupakan fungsi pada saat pesawat musuh menembakkan pelurunya ke arah target, yaitu pesawat jagoan, dan fungsi *takeHit*, yaitu fungsi apabila musuh terkena peluru pesawat jagoan. Yang *source code* nya adalah sebagai berikut :

- ```
public class Musuh1 extends MovieClip//Musuh1 adalah fungsi turunan (inheritance) dari movieclip Musuh1
private var stageRef:Stage;
private var vy:Number=1;// variabel pergerakan pesawat
private var ay:Number=.1;
private var target:Pesawat;//menentukan target sasaran
private var hitPoint:int=1;//variabel nilai hit saat kena tembakan
private var armor:int=2;//variabel armor pada pesawat Musuh1
public var points:int=100;//variabel penambahan score
private function loop (e:Event):void
{
if(currentLabel !="hancur")//label frame masih dalam keadaan normal
{
vy += ay;
y += vy;
if (y > stageRef.stageHeight)
removeSelf();//menambahkan fungsi removeSelf
if (y - 5 < target.y && y + 5 > target.y)
fireWeapon();//menambahkan fungsi fireWeapon
}
if(currentLabel=="hancurTotal")
{
removeSelf();//jika terkena tembakan dan nilai armor=0, maka frame label akan bergerak ke frame berikutnya, kemudian fungsi removeSelf akan segera dieksekusi
}
if(hitTestObject(target.hit)&¤tFrame==1)//jika musuh1 bertabrakan dengan pesawat jagoan, maka:
{
stageRef.addChild(newLedakanKecil(stageRef,x,y));//
menampilkan animasi movieclip ledakan kecil pada pesawat musuh1 dan pesawat jagoan

```

```

target.armor+=hitPoint;//menambahkan hitpoint pada pesawat musuh1 ketika menabrak pesawat jagoan, yang otomatis mengurangi nilai armor pesawat musuh
target.takeHit();//pesawat jagoan juga berkurang armornya ketika menabrak pesawat musuh1
this.takeHit(target.hitPoint);//mengurangi nilai armor pada pesawat jagoan
}

```



**Gambar 4.4 Pesawat Musuh Muncul**

c. Fungsi `removeSelf`

Adalah fungsi untuk menghilangkan movieclip musuh1 dari stage apabila terkena tembakan dan hancur, dengan *source code* sebagai berikut :

- `private function removeSelf() : void`

```

{
 removeEventListener(Event.ENTER_FRAME, loop);
}

```



```

 if (stageRef.contains(this))
 stageRef.removeChild(this);
}

```

d. Fungsi fireWeapon

Adalah fungsi untuk menembak pada pesawat musuh1 ke arah sasarannya, yaitu pesawat jagoan. Dengan *source code* sebagai berikut :

- private function fireWeapon():void
 

```

 {
 stageRef.addChild(new PeluruMusuh(stageRef, target, x, y, -10));//
 menampilkan movieclip peluruMusuh pada saat objek movieclip
 mendekati pesawat jagoan, dan peluru akan keluar dari samping
 kiri
 stageRef.addChild(new PeluruMusuh(stageRef, target, x, y, 10));//
 menampilkan movieclip peluruMusuh pada saat objek movieclip
 mendekati pesawat jagoan, dan peluru akan keluar dari samping
 kiri
 }

```

e. Fungsi takeHit

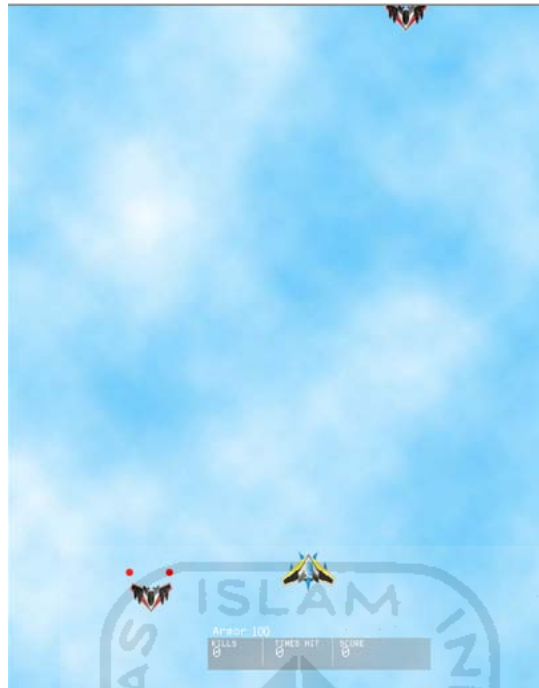
Prosedur Fungsi takeHit adalah fungsi untuk memainkan animasi ledakan pada saat hancur.

- public function takeHit(value:int):void
 

```

 {
 if(currentLabel != "hancur")
 {
 if (armor==0)//jika armor musuh sudah nol, maka:
 {
 dispatchEvent(new Event("killed"));
 rotation=Math.random()*360;
 gotoAndPlay("hancur");//ke frame dimana animasi ledakan
 dimainkan
 }
 }
 armor-=value;
 }

```



#### 4.5 Pesawat Musuh1 Menembak

#### 5. Prosedur pada pergerakan pesawat musuh2

Prosedur dalam class ini meliputi fungsi:

##### a. Fungsi musuh2

Fungsi Musuh2, yaitu fungsi untuk pergerakan pesawat secara acak.

- `public function Musuh2(stageRef:Stage,target:Pesawat):void`  
`{`  
`stop();`*menghentikan movieclip supaya pesawat musuh tetap pada posisi pada saat belum meledak*  
`this.stageRef=stageRef;`  
`this.target=target;`*menentukan target sasaran, yaitu si pesawat jagoan*  
`x = Math.random() * stageRef.stageWidth;`*posisi x pada saat musuh2 muncul*  
`y = -5;`*posisi y pada saat musuh2 muncul*  
`}`

```

jedaTembak=new Timer(500,1);//menentukan variabel timer pada
saat terakhir kali musuh2 menembak dengan tembakan berikutnya
jedaTembak.addListener(TimerEvent.TIMER,jedaTembakHandler);//m
enambahkan fungsi jedaTembakHandler untuk mengeksekusi fungsi
jedaTembak
addListener(Event.ENTER_FRAME, loop);//menambahkan
fungsi loop
}

```

b. Fungsi loop

Didalam fungsi ini, ditambahkan fungsi-fungsi baru, yaitu fungsi *removeSelf* untuk menghilangkan movieclip pesawat musuh, fungsi *fireWeapon*, yang merupakan fungsi pada saat pesawat musuh menembakkan pelurunya ke arah target, yaitu pesawat jagoan, dan fungsi *takeHit*, yaitu fungsi apabila musuh terkena peluru pesawat jagoan, dengan *source code* sebagai berikut :

- ```

private function loop (e:Event):void
{
if (currentLabel !="hancur")
{
if (y > stageRef.stageHeight)
removeSelf();//menambahkan fungsi removeSelf
if (x - 15 < target.x && x + 15 > target.x)
fireWeapon();//menambahkan fungsi fireWeapon
}
if (currentLabel == "hancurTotal")
{
removeSelf();//jika terkena tembakan dan nilai armor=0, maka frame
label akan bergerak ke frame berikutnya, kemudian fungsi
removeSelf akan segera dieksekusi
}
if(hitTestObject(target.hit)&&currentFrame==1)   jika   musuh2
bertabrakan dengan pesawat jagoan, maka:
{
stageRef.addChild(new      LedakanKecil(stageRef,      x,y));//
menampilkan animasi movieclip ledakan kecil pada pesawat
musuh2 dan pesawat jagoan
}

```

```

target.armor-=hitPoint;// menambahkan hitpoint pada pesawat musuh2 ketika menabrak pesawat jagoan, yang otomatis mengurangi nilai armor pesawat musuh2
target.takeHit();//mengurangi nilai armor pada pesawat jagoan
this.takeHit(target.hitPoint);// pesawat jagoan juga berkurang armornya ketika menabrak pesawat musuh1
}

```

c. Fungsi removeSelf

Prosedur Fungsi removeSelf Adalah fungsi dengan *source code* didalamnya untuk mengeksekusi menghilangkan amovieclip musuh2 dari stage apabila terkena tembakan dan hancur.

- ```

private function removeSelf() : void
{
removeEventListener(Event.ENTER_FRAME, loop);
if (stageRef.contains(this))
stageRef.removeChild(this);
}

```

d. Fungsi fireWeapon

Prosedur Fungsi fireWeapon adalah fungsi untuk menembak pada pesawat musuh1 ke arah sasarannya, yaitu pesawat jagoan, dengan *source code* sebagai berikut :

- ```

private function fireWeapon():void
{
if (tembakkan)
{
stageRef.addChild(newPeluruMusuh2(stageRef,target,x+10,y,30));//
memunculkan movieclip peluru musuh pada saat ditembakkan dari depan pesawat musuh2
tembakkan=false;
jedaTembak.start();//mengeksekusi fungsi jeda tembak
}
}
private function jedaTembakHandler(e:TimerEvent):void

```

```

{
    tembakkan=true;
}

```

e. Fungsi *takeHit*

Prosedur Fungsi *takeHit* Adalah fungsi untuk memainkan animasi ledakan pada saat hancur, dengan *source code* sebagai berikut :

- `public function takeHit(value:int):void`

```

{
    if(currentLabel != "hancur" )
    {
        if (armor==0) //jika armor musuh sudah nol, maka:
        {
            dispatchEvent(new Event("killed"));
            rotation=Math.random()*360;
            gotoAndPlay("hancur"); // ke frame dimana animasi ledakan dimainkan
        }
        armor-=value;
    }
}

```

6. Prosedur pada peluru pesawat utama

Prosedur dalam class ini mencakup fungsi:

a. Fungsi *loop*

Adalah prosedur fungsi pada movieclip peluru pesawat jagoan apabila mengenai pesawat musuh, juga berfungsi untuk menambahkan fungsi – fungsi berikutnya, yaitu fungsi hilangkan, dengan *source code* sebagai berikut :

- `private function loop(e:Event):void`

```

{
y-=kecepatan;//kecepatan peluru pada saat ditembakkan, yang
variabelnya telah ditentukan dalam class
if (y<0)//jika peluru sudah keluar dari stage,maka:
{
    hilangkan();//fungsi hilangkan akan dieksekusi
}
for (var i:int = 0; i < SkyJustice.musuh.length; i++)
{
if (hitTestObject(SkyJustice.musuh[i].hit))//
jika peluru mengenai musuh, maka:
{
    batasStage.addChild(new LedakanKecil(batasStage, x,y));//
movieclip ledakan kecil akan ditampilkan pada posisi dimana
pesawat musuh terkena peluru pesawat jagoan
    SkyJustice.musuh[i].takeHit(hitPoint);//
Setelah pesawat musuh tertembak, maka nilai hit akan bertambah 1
point
    hilangkan();//dan setelah peluru mengenai sasaran, dan movieclip
ledakan kecil ditampilkan, maka peluru akan hilang dengan
mengeksekusi fungsi hilangkan
}
}

```

b. Fungsi hilangkan

Mengeksekusi fungsi hilangkan pada saat peluru sudah keluar dari arena permainan atau sudah mengenai sasaran.

- private function hilangkan():void


```

{
removeEventListener(Event.ENTER_FRAME, loop);
if (batasStage.contains(this))
{
    batasStage.removeChild(this);//menghilangkan movieclip dari arena
permainan
}
}

```

7. Prosedur pada peluru Musuh1

Prosedur dalam class ini meliputi fungsi :

a. Fungsi loop

Prosedur fungsi loop adalah fungsi pada movieclip peluru musuh1 yang mengeksekusi apabila mengenai pesawat jagoan, juga berfungsi untuk menambahkan fungsi – fungsi berikutnya, yaitu fungsi removeself, dengan *source code* sebagai berikut :

- private function loop(e:Event):void
{
x+=vx;
if (x > stageRef.stageWidth || x < 0)//
karakteristik tembakan pada peluru musuh1 adalah muncul dari samping kiri dan kanan badan pesawat, maka jika peluru keluar dari arena permainan dari samping kiri dan kanan movieclip ini, maka:
{
removeSelf();//akan ditambahkan fungsi removeself untuk menghilangkan movieclip ini
}
if(stageRef.contains(target.hit))
{
if (hitTestObject(target.hit))//jika peluru mengenai pesawat jagoan, maka armor pada pesawat jagoan akan berkurang satu point, dan animasi ledakan kecil akan muncul pada posisi dimana peluru mengenai pesawat jagoan
{
target.armor-=hitPoint;
target.takeHit();
stageRef.addChild(newLedakanKecil(stageRef, x,y));
removeSelf();
}
}

b. Fungsi removeSelf

Mengeksekusi fungsi hilangkan pada saat peluru sudah keluar dari arena permainan atau sudah mengenai sasaran, dengan *source code* sebagai berikut :

- private function removeSelf() : void


```
{
removeEventListener(Event.ENTER_FRAME,loop);
if (stageRef.contains(this))
stageRef.removeChild(this); menghilangkan movieclip dari arena permainan
}
```

8. Prosedur pada peluru musuh2

Prosedur dalam class ini meliputi :

a. Fungsi tembak

Prosedur Fungsi tembak adalah fungsi pada movieclip peluru pesawat musuh2 apabila mengenai pesawat jagoan, juga berfungsi untuk menambahkan fungsi – fungsi berikutnya, yaitu fungsi hilangkan, dengan *source code* sebagai berikut :

- public function tembak(e:Event):void


```
{
y+=vy;
if(y>batasStage.stageHeight) // jika movieclip peluru sudah keluar dari arena permainan, maka movieclip peluru musuh akan dihilangkan dengan menambahkan fungsi hilangkan.
{
hilangkan();
}
if(batasStage.contains(target.hit))
{
if(hitTestObject(target.hit))
// jika peluru mengenai pesawat jagoan, maka armor pada pesawat jagoan akan berkurang satu point, dan animasi ledakan kecil akan muncul pada posisi dimana peluru mengenai pesawat jagoan
{
target.armor-=hitPoint;
target.takeHit();
batasStage.addChild(new LedakanKecil(batasStage,x,y));
hilangkan();
}
}
```


b. Fungsi hilangkan

Mengeksekusi fungsi hilangkan pada saat peluru sudah keluar dari arena permainan atau sudah mengenai sasaran.

- ```
public function hilangkan():void
{
 removeEventListener(Event.ENTER_FRAME,tembak);
 if (batasStage.contains(this))
 {
 batasStage.removeChild(this);// menghilangkan movieclip dari arena permainan
 }
}
```



**Gambar 4.6 Pesawat Musuh2 Menembak**

## 9. Prosedur pada LedakanKecil

Prosedur dalam class ini meliputi fungsi :

### a. Fungsi loop

- private function loop (e:Event)  
{  
  if (currentFrame==totalFrames)  
  {  
    hilangkan();  
  }  
}

### b. Fungsi hilangkan

- private function hilangkan():void  
{  
  removeEventListener(Event.ENTER\_FRAME,loop);  
  if (stageRef.contains(this))  
  {  
    stageRef.removeChild(this);  
  }  
}



**Gambar 4.7 Ledakan Kecil pada saat tertembak**

## 10. Prosedur pada ScoreBoard

Prosedur dalam class ini meliputi fungsi :

### a. Fungsi ScoreHud

- **public function ScoreHUD(stageRef:Stage)//*setelah permainan dimulai, maka angka – angka point pada text kill, hits, dan score dimulai dari angka nol, sedangkan pada armor dimulai dari angka 100***

```
{
this.stageRef = stageRef;
kills.text = "0";
hits.text = "0";
score.text = "0";
armor.text="100";
x = 10;
y = stageRef.stageHeight - height - 10;
}
```

### b. Fungsi updateKills

Prosedur Fungsi updateKills adalah fungsi untuk mencatat penambahan angka kills apabila pesawat berhasil menembak dan menghancurkan musuh, dengan source code sebagai berikut :

- **public function updateKills(value:Number) : void**

```
{
s_kills += value;
kills.text = String(s_kills);
}
```

### c. Fungsi updateHits

Prosedur Fungsi updateHits yaitu merupakan fungsi untuk mencatat penambahan nilai hits tiap kali pesawat terkena tembakan musuh, dengan souce code sebagai berikut :

- ```
public function updateHits(value:Number) : void
{
s_hits += value;
hits.text = String(s_hits);
}
```

d. Fungsi updateScore

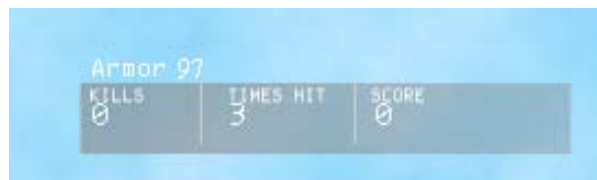
Prosedur Fungsi updateScore adalah fungsi untuk mencatat penambahan nilai score setiap kali pesawat jagoan berhasil menghancurkan lawan.

- ```
public function updateScore(value:Number) : void
{
s_score += value;
score.text = String(s_score);
}
```

e. Fungsi updateArmor

Prosedur Fungsi updateArmor adalah fungsi untuk mencatat pengurangan nilai armor setiap kali pesawat terkena tembakan dari musuh, dengan *source code* sebagai berikut :

- ```
public function updateArmor(value:int):void
{
s_armor =value;
armor.text=String(s_armor); }
```

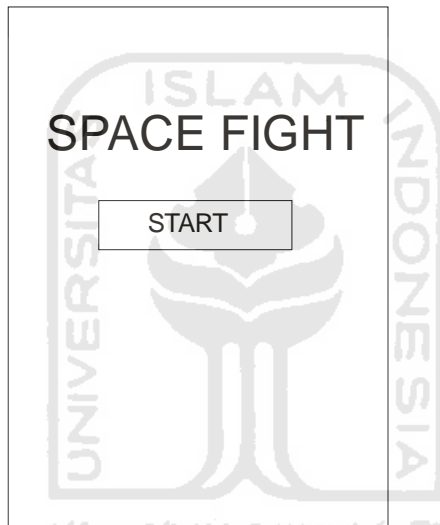


Gambar 4.8 ScoreBoard

4.4 Pembahasan

4.4.1 Analisis Kesesuaian

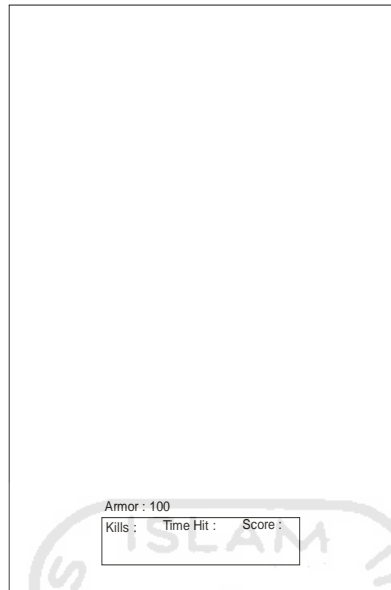
Analisis kesesuaian adalah perbandingan antara rancangan yang dibuat dengan implementasi sesungguhnya pada saat eksekusi program, yang disajikan dalam bentuk gambar, apakah sudah sesuai dengan perancangan.



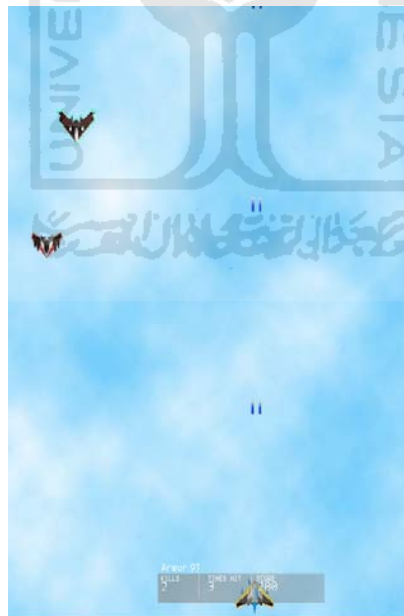
Gambar 4.9 Perancangan antar muka



Gambar 4.10 Implementasi Perancangan Antar Muka



Gambar 4.11 Rancangan Menu Game



Gambar 4.12 Implementasi Menu Game

Armor : 100

Kills :	Time Hit :	Score :

Gambar 4.13 Rancangan Score Board



Gambar 4.14 Implementasi Score Board

Sesuai dengan teori dalam urutan pembuatan program permainan pada bab 2 sebelumnya, analisis yang dilakukan pada program game SpaceFight adalah sebagai berikut :

- a. Tipe atau jenis game yang dibuat adalah berjenis arcade scrolling game.
- b. Terdapat informasi bagi pemain (user) ketika game sedang berjalan, yaitu adanya informasi tentang indikator nyawa bagi pesawat utama, berapa kali pesawat membunuh musuh, dan informasi nilai ketika user berhasil membunuh musuh.

4.4.2 Hasil Analisis

Game ini menggunakan tampilan 2D, rancangan pesawat utama yaitu berwarna kuning dan hitam, peluru yang ditembakkan berwarna biru dengan gradasi putih, warna latar permainan yaitu biru muda dengan gradasi putih, menampilkan bentuk

awan di angkasa. Ketika user menekan tombol start pada menu utama, maka program akan langsung menampilkan frame berikutnya, dan user bisa langsung bermain. Adapun tampilan pesawat musuh yang pertama yaitu berwarna merah dan hitam, dengan peluru berwarna merah yang akan menembak ke samping kiri dan kanan ketika sudah mendekati targetnya, yaitu pesawat jagoan. Musuh ke dua berwarna hitam dengan peluru berwarna merah dan akan menembak lurus ke arah pesawat jagoan.



BAB V

KESIMPULAN DAN SARAN

Setelah dilakukan pengujian dan analisis terhadap program permainan Game SpaceFight : The Legend Of Batu Corporation, maka dihasilkan beberapa kesimpulan sebagai berikut :

5.1 Kesimpulan

1. Game ini memberikan tambahan alternatif dari game berjenis arcade scrolling game.
2. Penggunaan Adobe Flash CS 3 dengan metode Object Oriented Programming dan Action Script 3.0 (AS3) untuk pembuatan game ini sangat tepat, karena mendukung pemrograman multimedia dengan tipe data yang lengkap.

5.2 Saran

Berdasarkan kesimpulan yang ditemukan, maka saran – saran yang dibutuhkan untuk tahap pengembangan game ini adalah sebagai berikut :

1. Game arcade ini membutuhkan tampilan antar muka yang menarik seperti misalnya animasi tombol pada saat diklik, demo game, dan warna – warna yang futuristik pada latar belakang supaya pemain tidak cepat bosan dalam bermain.

2. Menambahkan fungsi untuk menyimpan game yaitu menambahkan fasilitas save, load dan pause.
3. Menambahkan tingkat kesulitan yang bisa dipilih pemain sesuai tingkat kesulitan yang diinginkan, tentunya akan membuat pemain merasa tertantang terus.
4. Penghargaan, atau hadiah yang diberikan kepada user misalnya berupa nilai tertinggi juga diperlukan supaya user lebih tertantang lagi untuk menggunakan program ini.



DAFTAR PUSTAKA

- [ADO02] Adobe System Incorporated. *Programming ActionScript 3.0*. California : Adobe System Incorporated, 2002.
- [BER04] Bernard Hartanto dan Ferdianto Wijaya. *The Magic Of Flash MX 2004*. Jakarta : Elex Media Komputindo, 2004.
- [COL07] Collin Mook . *Essential ActionScript 3.0* . California : O'Reily Media , 2007.
- [KRI09] Kristo Radion. *Ultimate Game Design : Building RPG Games Using Adobe Flash Action Script*. Yogyakarta : Andi Offset, 2009.
- [PET07] Peter Elst, Sas Jacobs, Tood Yard. *Object-Oriented ActionScript 3.0*. USA : FriendsOfEducation, 2007
- [GRW08] Gary Rosenzweig. *Action Script 3.0 , Game Programming University*. Indiana Polish : Que Publishing, 2008.
- [WIL07] William B.Sanders and Chandima Cumaranatunge. *Action Script 3.0 Design Patterns*. California : O'Reily Media , 2007.

