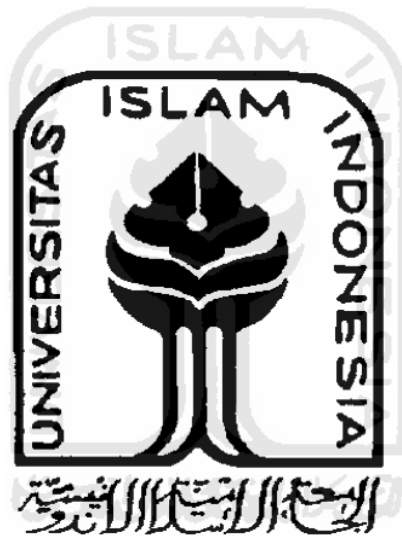


Aplikasi Game Shooter Berbasis Sistem Operasi Android Menggunakan Interaksi Touchscreen

TUGAS AKHIR

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



oleh :

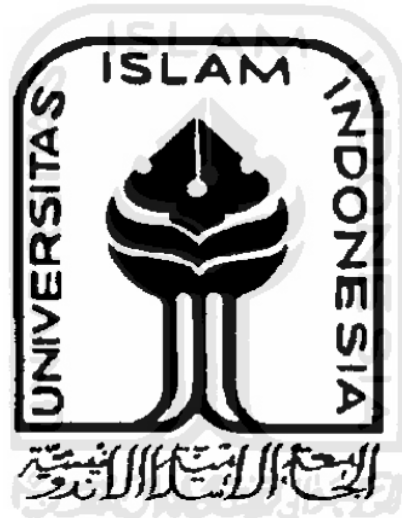
Nama : Aryo Agung Prakoso
No. Mahasiswa : 06 523 066

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2010**

Aplikasi Game Shooter Berbasis Sistem Operasi Android Menggunakan Interaksi Touchscreen

TUGAS AKHIR

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



oleh :

Nama : Aryo Agung Prakoso
No. Mahasiswa : 06 523 066

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2010**

LEMBAR PENGESAHAN PEMBIMBING

Aplikasi Game Shooter Berbasis Sistem Operasi Android Menggunakan Interaksi Touchscreen

TUGAS AKHIR



Pembimbing

Affan Mahtarami, S.Kom., M.Kom.

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

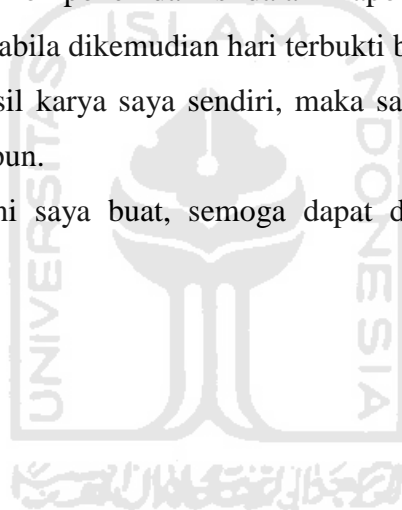
Saya yang bertanda tangan dibawah ini :

Nama : Aryo Agung Prakoso

No. Mahasiswa : 06 523 066

Menyatakan bahwa seluruh komponen dan isi dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti bahwa ada beberapa bagian dari karya ini bukanlah hasil karya saya sendiri, maka saya akan siap menanggung risiko dan konsekuensi apapun.

Demikianlah pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.



Yogyakarta, 9 Maret 2011

Aryo Agung Prakoso

LEMBAR PENGESAHAN PENGUJI
Aplikasi Game Shooter Berbasis Sistem Operasi
Android Menggunakan Interaksi Touchscreen

TUGAS AKHIR

oleh :

Nama : Aryo Agung Prakoso

No. Mahasiswa : 06 523 066

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat

Untuk Memperoleh Gelar Sarjana Teknik Informatika Fakultas

Teknologi Industri Universitas Islam Indonesia

Yogyakarta,

Tim Penguji

Affan Mahtarami, S.Kom., MT, _____

Ketua

Syarif Hidayat S.Kom., M.I.T. _____

Anggota I

Yudi Prayudi, S.SI, M.Kom. _____

Anggota II

Mengetahui,

Ketua Jurusan Teknik Informatika

Universitas Islam Indonesia

Yudi Prayudi, S.SI, M.Kom.

HALAMAN PERSEMBAHAN

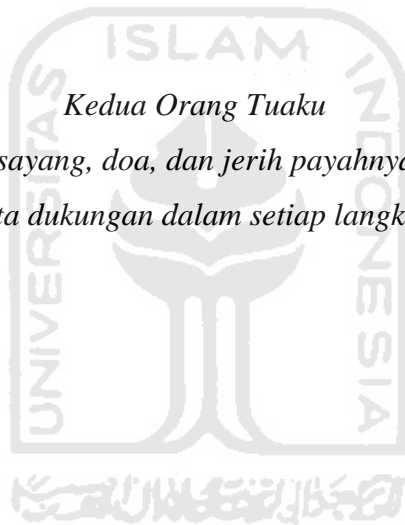
Tugas Akhir ini Aku Persembahkan Untuk :

Allah SWT

Yang selalu melimpahkan Rahmat dan Hidayahny, memberikanku akal pikiran untuk berkarya,serta anugrah yang tak terhitung banyaknya

Kedua Orang Tuaku

Terimakasih atas kasih sayang, doa, dan jerih payahnya mendidik hingga saat ini,serta dukungan dalam setiap langkahku.



MOTTO

“Orang-orang yang beriman dan hati mereka menjadi tentram dengan mengingat Allah, ingatlah hanya dengan mengingat Allah hati mereka menjadi tentram.”

“sesungguhnya setelah kesulitan tersimpan sebuah kemudahan”

“Ilmu Allah tidaklah terhingga, seumpama dibuat pena dari seluruh pohon di bumi dan seluruh lautan sebagai tintanya, niscaya kamu tidak akan mampu menuliskan semua ilmu-Nya ”

“Sesungguhnya orang yang masuk surga adalah orang yang bermanfaat untuk orang disekitarnya”

“Jika kamu tidak memiliki apa yang kamu sukai, maka sukailah yang kamu miliki saat ini”

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu'alaikum Wr.Wb.

Alhamdulillah rabbil'alamin, puji syukur kehadiran Allah SWT, atas limpahan hidayah, taufiq serta 'inayah-Nya, sehingga Tugas Akhir yang berjudul “Aplikasi *Game Shooter* Berbasis Sistem Operasi Android Menggunakan Interaksi *Touchscreen*” ini dapat terselesaikan dengan baik. Shalawat serta salam semoga senantiasa tercurah atas Nabi Muhammad SAW, para kerabat, serta pengikutnya hingga hari kiamat nanti.

Laporan Tugas Akhir ini disusun untuk melengkapi salah satu syarat guna memperoleh gelar Sarjana Strata-1 (S1) Jurusan Teknik Informatika di Universitas Islam Indonesia.

Pada kesempatan ini penyusun ingin mengucapkan banyak terima kasih kepada pihak-pihak yang mempunyai andil besar dalam pelaksanaan dan penyelesaian laporan tugas akhir ini, antara lain :

1. Ibu, dan Adik serta semua keluarga besarku untuk doa dan dukungannya.
2. Bapak Affan Mahtarami, ST., MT., selaku Dosen Pembimbing Tugas Akhir yang senantiasa membimbing penyusun agar dapat lancar menyelesaikan tugas akhir ini.
3. Bapak Yudi Prayudi, S.Si, M.Kom selaku Ketua Jurusan Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Seseorang yang insyaAllah selalu ada di hati, terimakasih atas dukungan semangat, dan doanya selama ini.
5. Bapak, Ibu dosen Teknik Informatika dan dosen-dosen Universitas Islam Indonesia. Terima kasih atas semua ilmu yang diberikan.
6. Mas Dayu Bagus Permana, makasi banget atas *sharing* ilmunya.

7. Teman-teman informatika *Fire'06* (Hari, Bamas, Rahmat, Titi, Lantip, dan semuanya) khususnya dan seluruh teman-teman angkatan lainnya yang memberikan semangat dan do'a sehingga penyusun dapat menyelesaikan tugas akhir ini dengan baik.
8. Teman-teman asisten dari seluruh Laboratorium informatika yang telah memberikan semangat, dukungan dan sharing ilmunya.
9. Semua pihak yang tidak dapat disebutkan satu persatu yang telah membantu sehingga laporan tugas akhir ini dapat terselesaikan.

Penyusun menyadari sepenuhnya bahwa laporan ini masih jauh dari sempurna, karena itu saran dan kritik sangat penyusun harapkan untuk perbaikan di masa mendatang.

Di tengah keterbatasan penyusun dalam mengerjakan laporan tugas akhir ini, penyusun berharap kiranya laporan ini bermanfaat bagi pembaca. Semoga Allah SWT membimbing dan menyertai setiap langkah kita. Amiin.

Wassalamu'alaikum Wr.Wb.

Yogyakarta, 9 Maret 2011

Penyusun

SARI

Perkembangan teknologi sangat berpengaruh terhadap perkembangan game saat ini, dimana perubahannya akhir-akhir ini makin pesat berkembang. Seiring dengan perkembangan teknologi maka game sekarang juga dapat dimainkan pada *handphone* terutama *smartphone*, dan *Operating System (OS) smartphone* terbaru saat ini adalah Android. Karena semua *smartphone* Android memiliki interaksi *touchscreen*, maka perlu diancang suatu aplikasi *game* yang berjalan pada OS Android dan menggunakan interaksi berbasis *touchscreen* sebagai *interface*-nya.

Perancangan dan pembuatan Aplikasi Game Shooter Berbasis Sistem Operasi Android Menggunakan Interaksi Touchscreen memerlukan adanya beberapa tahapan, yaitu : mendefinisikan masalah utama, pengumpulan data, perancangan perangkat lunak, pembuatan aplikasi, dan melakukan evaluasi dan pengujian sistem.

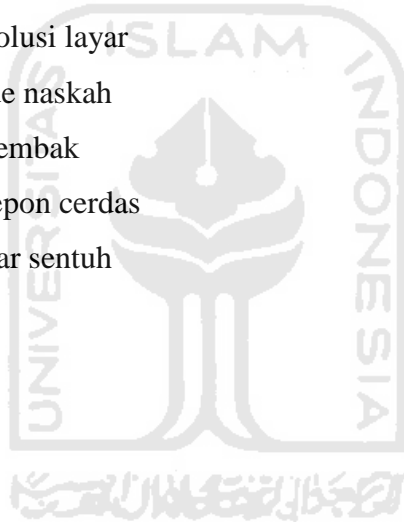
Aplikasi game yang telah dibuat dapat berjalan pada OS Android yang bisa dijalankan dengan interaksi *touchscreen*, namun memiliki beberapa kelemahan karena memerlukan *resource memory* yang besar, dan akan lebih baik lagi jika Interaksi bawaan pada *device* Android yaitu *accelerometer* juga dipai untuk interaksinya.

Kata Kunci : Flash, Android, Game, Shooter.

TAKARIR

<i>ActionScript 3.0</i>	Kode bahasa pemrograman yang baru pada Flash
<i>Activity Diagram</i>	Diagram aktivitas
<i>Background</i>	Latar belakang
<i>Coding</i>	memberi kode logika pada program
<i>Crosshair</i>	garis sasaran tembak
<i>Developer</i>	Pengembang
<i>Device</i>	Piranti
<i>Emulator</i>	Sistem operasi yang dapat menjalankan sistem operasi lainnya.
<i>Gadget</i>	Piranti elektronik canggih yang praktis dan ringkas dibawa
<i>Game</i>	Permainan
<i>Game Art</i>	Komponen pada permainan yang tampak oleh mata
<i>Game Engine</i>	sebuah perangkat lunak sistem yang dirancang untuk penciptaan dan pengembangan game
<i>Graphic User Interface</i>	Adalah citra grafis yang ditampilkan di layar komputer yang memungkinkan untuk mengakses aplikasi software
<i>Input</i>	Masukan
<i>Interface</i>	Antarmuka
<i>Landscape</i>	Pemandangan
<i>Layout</i>	Lembaran
<i>Main Menu</i>	Menu utama
<i>Memory Resource</i>	Memori yang terpakai
<i>Mixing Sound</i>	Mencampur suara
<i>Native Application</i>	Aplikasi dengan ekstensi eksekusi yang dikenali oleh suatu sistem operasi

<i>Open Source</i>	Perintah-perintah program yang tersedia secara gratis berikut untuk digunakan oleh kalangan luas. Boleh dimodifikasi dan digunakan oleh siapa saja.
<i>Operating system</i>	Sistem operasi
<i>Output</i>	Keluaran
<i>Random</i>	Acak
<i>Reload</i>	Mengisi kembali
<i>Render Mode</i>	Mode pemrosesan gambar
<i>Runtime</i>	Waktu saat eksekusi program
<i>Screen Resolution</i>	Resolusi layar
<i>Script</i>	Kode naskah
<i>Shooter</i>	Penembak
<i>Smartphone</i>	Telepon cerdas
<i>Touch Screen</i>	Layar sentuh



DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN PEMBIMBING.....	ii
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR.....	iii
LEMBAR PENGESAHAN PENGUJI.....	iv
HALAMAN PERSEMBAHAN.....	v
MOTTO.....	vi
KATA PENGANTAR.....	vii
SARI.....	ix
TAKARIR.....	x
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	xiv
BAB I PENDAHULUAN.....	15
1.1 Latar Belakang.....	15
1.2 Rumusan Masalah.....	16
1.3 Batasan Masalah.....	16
1.4 Tujuan Penelitian.....	16
1.5 Manfaat Penelitian.....	16
1.6 Metode Penelitian.....	17
1.7 Sistematika Penulisan.....	18
BAB II LANDASAN TEORI.....	20
2.1 Perkembangan Sistem Operasi <i>Smartphone</i>	20
2.2 Pengertian Sistem Operasi Android.....	21
2.3 Pengertian Game Secara Umum.....	21
2.4 <i>Game</i> Pada <i>Smartphone</i> Android.....	24
2.5 Pengembangan <i>Game</i> Dengan Flash dan Air Pada Android.....	24
2.6 Arsitektur Platform Android dan <i>Air Runtime</i>	25

2.7	Bahasa Pemrograman	25
2.8	Game Shooter	26
BAB III METODOLOGI.....		27
3.1	Metode Analisis.....	27
3.2	Analisis Kebutuhan	27
3.2.1	Analisis Kebutuhan Input.....	27
3.2.2	Analisis Kebutuhan Fungsi dan Kinerja	28
3.2.3	Analisis Kebutuhan Output.....	28
3.2.4	Analisis kebutuhan Perangkat Keras.....	28
3.2.5	Analisis Kebutuhan Perangkat Lunak.....	29
3.3	Perancangan Perangkat Lunak	31
3.3.1	Metode Perancangan	31
3.3.2	Hasil Perancangan.....	31
3.3.2.1	<i>Activity Diagram</i>	31
3.3.2.2	Perancangan Antarmuka	32
BAB IV HASIL DAN PEMBAHASAN		35
4.1	Batasan Implementasi	35
4.2	Tahap Pembuatan Proses.....	36
4.3	Implementasi Pembuatan Proses.....	37
4.4	Implementasi Interface.....	44
4.4.1	Implementasi Main Menu Aplikasi Game Shooter.....	44
4.4.2	Implementasi Interface Gameplay dan System.....	45
4.5	Analisis Kinerja Perangkat Lunak	46
4.5.1	Analisis Kesesuaian Dengan Landasan Teori.....	46
4.5.2	Analisis Kinerja Dengan menggunakan Responden	47

BAB V SIMPULAN DAN SARAN.....	54
5.1 Simpulan.....	54
5.2 Saran.....	54
DAFTAR PUSTAKA	55
LAMPIRAN.....	57

DAFTAR GAMBAR

Gambar 2.1 <i>Android Architecture</i>	25
Gambar 2.2 Beberapa Game yang ada di Android market	26
Gambar 3.1 <i>Activity Diagram</i>	32
Gambar 3.2 Perancangan antarmuka <i>Splash Screen</i>	33
Gambar 3.3 Perancangan antarmuka <i>Main Menu</i>	33
Gambar 3.4 Perancangan antarmuka <i>Level</i>	34
Gambar 4.1 Tahap Pembuatan Layout Dasar Pada Flash.....	37
Gambar 4.2 Tahap Pembuatan <i>Art</i> Pada <i>Game</i>	38
Gambar 4.3 Tahap <i>Layering</i> dan <i>Labeling</i>	38
Gambar 4.4 Tahap Pembuatan Menu Utama.....	39
Gambar 4.5 Tahap Pembuatan <i>Level Gameplay</i>	39
Gambar 4.6 Tahap Pembuatan <i>Crosshair</i>	40
Gambar 4.7 Tahap <i>Mixing</i> Musik	41
Gambar 4.8 Tahap Penggabungan <i>Content Game</i>	42
Gambar 4.9 Tahap <i>Publishing</i>	42
Gambar 4.10 Tahap <i>Install To Emulator</i>	43
Gambar 4.11 Tahap Pengujian.....	43
Gambar 4.13 <i>Interface Gameplay</i> dan <i>System</i>	45

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi sangat berpengaruh terhadap perkembangan game saat ini, dimana perubahannya akhir-akhir ini makin pesat berkembang. Pada sekitar awal dekade 80-an, sebenarnya sudah ada persaingan ketat antar perusahaan game dalam memasarkan produknya. Game yang populer dengan nama video game ini hanya bisa dimainkan oleh satu atau dua orang pemain pada sebuah console. Pesawat televisi dibutuhkan sebagai media tampilan. Jenis game-nya juga masih sangat sederhana dengan grafik yang sangat kasar. Tampilannya mirip seperti game dari bahasa program Java, yang bisa dimainkan di handphone sekarang ini. Perusahaan game yang terkenal pada saat itu adalah Atari, Sega dan Nintendo.

Seiring dengan perkembangan teknologi maka game sekarang juga dapat dimainkan pada *handphone* terutama *smartphone*, dan *Operating System (OS) smartphone* terbaru saat ini adalah Android. Android merupakan OS untuk *smartphone* yang berbasis *Open Source*, bisa diibaratkan seperti *Operating System linux ubuntu* pada sebuah *PC*. Android dikembangkan oleh Google, dan pertama kali dikenal pada bulan *november 2007*.

Sebagian pengguna *smartphone* berusia muda yaitu 42% penggunanya berumur 18-34 tahun dan hanya 6% penggunanya manula. Sedangkan Android sendiri penggunanya sebagian besar generasi muda, setengah penggunanya berusia 18-34 tahun. Sedangkan pengguna RIM BB dan Microsoft Mobile, sebagian besar berusia 35 tahun ke atas. Disini terlihat sekali bahwa generasi muda yang paling mudah menerima teknologi baru, menggunakan system operasi yang baru dan menawarkan banyak fitur.

Dikarenakan Android tergolong OS baru yang semua device pendukungnya menggunakan fasilitas *touchscreen* sebagai *interface*-nya, maka perlu dibuat aplikasi game yang mampu memaksimalkan fasilitas *touchscreen* tersebut.

1.2 Rumusan Masalah

Bagaimana merancang dan membuat sebuah aplikasi game pada *Operating System* Android yang menggunakan sistem interaksi *touchscreen*.

1.3 Batasan Masalah

Batasan masalah untuk aplikasi game ini adalah sebagai berikut:

1. Aplikasi akan di-debug dan diuji menggunakan emulator Smartphone berbasis OS Android versi 2.2 / Froyo.
2. Interaksi menggunakan *touchscreen*.
3. Aplikasi yang dibuat adalah game.

1.4 Tujuan Penelitian

Tujuan pembuatan tugas akhir ini adalah merancang suatu aplikasi game yang berjalan pada OS Android dan menggunakan interaksi berbasis *touchscreen* sebagai *interface*-nya.

1.5 Manfaat Penelitian

Berikut beberapa manfaat yang diperoleh dari pembuatan tugas akhir ini, manfaat tersebut antara lain:

1. Untuk Pengguna
 - a. Sebagai hiburan untuk mengisi waktu luang.
 - b. Melatih kesabaran dan ketenangan.
 - c. Menghilangkan stress

2. Untuk *Developer*
 - a. Sebagai game *sample* untuk memaksimalkan fasilitas touchscreen pada Android.
 - b. Sebagai referensi *Cross-Platform* game antara Flash, Air, dan Android.

1.6 Metode Penelitian

Perancangan dan pembuatan Aplikasi Game Shooter Berbasis Sistem Operasi Android Menggunakan Interaksi Touchscreen memerlukan adanya beberapa metodologi penelitian yang dilaksanakan demi kelancaran penelitian tindakan yang dilakukan, metodologi penelitian tersebut antara lain:

1. Mendefinisikan masalah utama dan menetapkan tujuan. Pokok permasalahan dalam penelitian ini telah dijelaskan di rumusan masalah, tujuan penelitian pun telah dijelaskan sebelumnya.
2. Mengumpulkan data primer dan sekunder yang diperoleh dari internet dan telaah studi pustaka yang terdapat di beberapa literatur mengenai permasalahan terkait. data penelitian diperoleh dari beberapa sumber tertulis yang akurat kebenarannya. Sumber-sumber diperoleh dari:
 - a. Buku-buku dan literatur berkaitan dengan OS Android.
 - b. Pencarian melalui internet.
3. Perancangan perangkat lunak. Tahapan perancangan perangkat lunak merupakan tahap awal dalam proses pembangunan suatu perangkat lunak. Dalam tahapan ini juga dijelaskan mengenai analisis kebutuhan perangkat lunak yang disertai gambaran perancangan perangkat lunak. Metode yang digunakan dalam perancangan kebutuhan perangkat lunak adalah rekayasa perangkat lunak.
4. Pembuatan Aplikasi *game*. Metode ini merupakan salah satu tahapan

terpenting dalam penelitian, karena inti dari penelitian adalah pembuatan sebuah perangkat lunak, dalam tahapan ini perancangan yang telah dibuat sebelumnya kemudian diimplementasikan ke dalam bahasa pemrograman dan antar muka sistem. *Software* yang dibutuhkan untuk membangun sistem antara lain:

- a. *Emulator*: Android SDK
 - b. *Script Editor* dan *Interface*: Adobe Flash CS5
 - c. *Runtime Environment*: Adobe Air Runtime
5. Melakukan evaluasi dan pengujian sistem. Metode yang terakhir adalah testing dan evaluasi sistem yang selanjutnya dapat dijadikan bahan revisi apabila terdapat kesalahan dalam sistem.

1.7 Sistematika Penulisan

Sistematika penulisan untuk laporan tugas akhir ini terdiri dari 5 bab, dengan penjelasan sebagai berikut:

BAB I Pendahuluan, merupakan pembahasan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan laporan.

BAB II Landasan Teori, membahas dasar-dasar teori yang digunakan dalam penelitian serta tinjauan pustaka penelitian yang sejenis. Dasar-dasar teori yang dibahas mencakup penjelasan OS Android, peran OS Android terhadap perkembangan *smartphone*, dan penjelasan kebutuhan sistem.

BAB III Metodologi, merupakan penjelasan tentang metode analisis kebutuhan perangkat lunak, metode analisis yang digunakan serta hasil analisis, *input*, *output*, kebutuhan perangkat lunak, kebutuhan perangkat keras, antar muka (*interface*) yang dibutuhkan dan fungsi yang diharapkan. Pada bagian ini juga memuat metode perancangan perangkat lunak yang akan digunakan, hasil perancangan perangkat lunak. Selain itu juga memuat perancangan antarmuka sistem.

BAB IV Hasil dan Pembahasan, menjelaskan implementasi perangkat lunak disertai dengan kelebihan dan kekurangan aplikasi. Meliputi tampilan antar muka, bahasa pemrograman yang digunakan, prosedur bahasa pemrograman *Actionscript 3.0*, pengujian dan penjelasan fungsi-fungsi yang merupakan tahap lanjutan dari proses menjadi suatu perangkat lunak dari sejumlah menu, proses, *input* dan *output* yang mewakili pengolahan data.

BAB V Kesimpulan dan Saran, merupakan kesimpulan berupa rangkuman-rangkuman hasil penelitian dan saran-saran mengenai keterbatasan yang ditemukan selama proses pembuatan laporan penelitian dan selama proses perancangan serta pembuatan aplikasi.



BAB II

LANDASAN TEORI

2.1 Perkembangan Sistem Operasi *Smartphone*

Sistem Operasi *gadget* di dunia saat ini yang terkenal tercatat ada 7, antara lain: Android, Apple Iphone, Palm WebOS, Microsoft Window Mobile, Palm, RIM BlackBerry, dan Symbian. Masing-masing mengklaim sebagai Sistem Operasi yang terbaik oleh pembuatnya. Berkaitan dengan pasar pengguna *smartphone* sendiri berdasarkan data Nielsen, di Amerika sekarang pertumbuhan pengguna Android cukup pesat, hanya dalam 1 tahun saja share pengguna Android menjadi hampir 20% dari sebelumnya tidak lebih dari 5%. Dengan iPhone perbedaan pengguna Android mulai tipis sekitar 9% saja, termasuk dengan raksasa RIM Blackberry (RIM BB) yang masih menguasai pasar *smartphone* di Amerika. Microsoft Mobile yang awalnya berambisi besar untuk mengalahkan RIM BB dan iPhone ternyata malah menurun akibat banyak penggunanya yang beralih ke sistem operasi yang baru yang bernama Android.

RIM BB memang saat ini masih menguasai pasar *smartphone* di Amerika, terutama untuk pengguna berumur 45 tahun keatas, sedangkan untuk generasi yang lebih muda (dibawah 44 tahun) masih menggunakan iPhone. Pengguna Android sendiri mempunyai kecenderungan dibawah usia 44 tahun, sedangkan Microsoft mobile masih disukai oleh generasi tua yaitu usia 55 tahun keatas.

Sebagian pengguna *smartphone* berusia muda yaitu 42% penggunanya berumur 18-34 tahun dan hanya 6% penggunanya manula. Sedangkan Android sendiri penggunanya sebagian besar generasi muda, setengah penggunanya berusia 18-34 tahun. Sedangkan pengguna RIM BB dan Microsoft Mobile, sebagian besar berusia 35 tahun ke atas. Disini terlihat sekali bahwa generasi muda yang paling

mudah menerima teknologi baru, menggunakan system operasi yang baru dan menawarkan banyak fitur.

2.2 Pengertian Sistem Operasi Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform *open source* (terbuka) bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisannya perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD).

2.3 Pengertian Game Secara Umum

Dalam bahasa Indonesia “Game” berarti “permainan”. Permainan yang dimaksud dalam game juga merujuk pada pengertian sebagai “kelincahan intelektual” (*intellectual playability*). Sementara kata “game” bisa diartikan sebagai arena keputusan dan aksi pemainnya. Ada target-target yang ingin dicapai pemainnya. Kelincahan intelektual, pada tingkat tertentu, merupakan ukuran sejauh mana game itu menarik untuk dimainkan secara maksimal.

Dahulu istilah game identik dengan anak-anak selaku pemain. Yang ada dibenak kita adalah suatu kegiatan yang dilakukan oleh anak-anak yang menurut mereka itu dapat

menyenangkan hati mereka. Segala bentuk kegiatan yang memerlukan pemikiran, kelincahan intelektual dan pencapaian terhadap target tertentu dapat dikatakan sebagai *game*. Tetapi yang akan dibahas pada kesempatan ini adalah *game* yang terdapat di komputer, baik off line maupun online. Saat ini perkembangan games di komputer sangat cepat. Para pengelola industri game berlomba-lomba untuk menciptakan game yang lebih nyata dan menarik untuk para pemainnya. Hal inilah yang membuat perkembangan games di komputer sangat cepat. Sehingga *games* bukan hanya sekedar permainan untuk mengisi waktu luang atau sekedar hobi. Melainkan sebuah cara untuk meningkatkan kreatifitas dan tingkat intelektual para penggunanya.

Jadi, bermain game adalah suatu proses “*fine tuning*” (atau penyamaan frekuensi) dari logika berpikir anak-anak kita dengan logika berpikir aplikasi komputer yang canggih tadi. Pada saat bersamaan, game juga secara nyata mempertajam daya analisis para penggunanya untuk mengolah informasi dan mengambil keputusan cepat yang jitu. Namun, tentu saja kenyataan juga harus kita masukkan kedalam perhitungan. Kenyataan itu diantaranya adalah kecanduan para pemain / penggunanya yang akut terhadap permainan komputer semacam ini. Mereka bisa lupa segala-galanya akan tugas mereka yang lain termasuk tugas menuntut ilmu.

Aplikasi games dapat dijalankan pada berbagai platform. Beberapa platform media yang bisa dijadikan pilihan untuk advergaming adalah :

1. *Personal Computer*

berupa aplikasi games pada PC. Games ini dijalankan pada personal computer. Pada media Personal Computer, durasi waktu penayangan adalah tidak terbatas, selama games tersebut bagus dan mampu menghibur audience/pemain, maka selama itulah tingkat kemungkinan untuk dilihat audience akan semakin tinggi. Pembuatan Games ini biasanya menggunakan Language C++ / C#, membutuhkan waktu 3-6 bulan tergantung dengan kompleksitas games tersebut.

2. *Web Based Games*

Yaitu aplikasi games yang diletakkan pada server di internet dimana audience/pemain hanya perlu menggunakan akses internet dan browser untuk mengakses games tersebut. Corporate yang memiliki keinginan mengumpulkan komunitas internet dan sekaligus memperkenalkan product/brandnya kepada komunitas Internet sangat cocok memilih media games online internet ini. Pilihan ini banyak digunakan karena dapat meningkatkan traffic pada suatu website. Bila games yang dibutuhkan simple maka diperlukan waktu pengembangan yang relative lebih singkat 1-2 bulan.

3. *SmartPhones/PocketPC*

Aplikasi games yang berjalan pada *mobile devices* ber-*Operating System*(OS) seperti Symbian, iOS, Android, dan Blackberry. Product yang memiliki target market status ekonomi A,B dan karakter pengusaha/ eksekutif muda sangat cocok apabila menggunakan media PocketPC/SmartPhones sebagai bagian aktivitas kegiatan advertising. *Games* bertemakan strategi ataupun simulasi bisnis pada umumnya lebih disukai oleh orang-orang golongan ini. Waktu pengembangan biasanya memerlukan waktu 3-6 bulan tergantung dengan kompleksitas games tersebut

4. *CellPhones / Mobile Phones*

Aplikasi *games* untuk cell-phones atau mobile phones sebenarnya bisa dimanfaatkan untuk mendapatkan potensial pelanggan dengan golongan Status Ekonomi Responden A,B dan C. Produk yang memiliki target golongan ABG, remaja, anak kuliah, eksekutif muda sangat tepat menggunakan media ini, mengingat pada golongan inilah mereka suka sekali mencoba dan mengeksplorasi features aplikasi yang tersedia pada mobile/cellphones mereka. Mereka biasanya aktif mencari aplikasi bersifat fun atau game yang dapat mengisi waktu senggang mereka. Waktu pengembangan untuk membuat aplikasi pada *devices* ini relative lebih pendek yaitu 1-3 bulan tergantung dengan kompleksitas games tersebut.

2.4 **Game Pada Smartphone Android**

Ada banyak platform game yang dapat berjalan pada OS Android, mulai dari Java, 3D, Unity, Flex, Air, hingga Flash yang sudah ter-*support* mulai dari OS Android seri 2.1 Eclair.

Game berbasis Flash sudah sangat lama populer dikalangan umum karena keindahan *interface*-nya dan kemudahan dalam pembuatan game tersebut akibat banyaknya komunitas dan forum di internet yang mem bahas nya. Akan tetapi game / aplikasi berbasis Flash pada Android masih terbatas berjalan di browser smartphone tersebut dengan menggunakan browser yang ter-*install Flash Player*.

Untuk bisa menjalankan game / aplikasi berbasis Flash secara *native* pada smartphone Android maka kita membutuhkan *Air Runtime* dan masih harus mengubahnya menjadi *Air application* yang kemudian diubah menjadi aplikasi *native* pada Android (.apk).

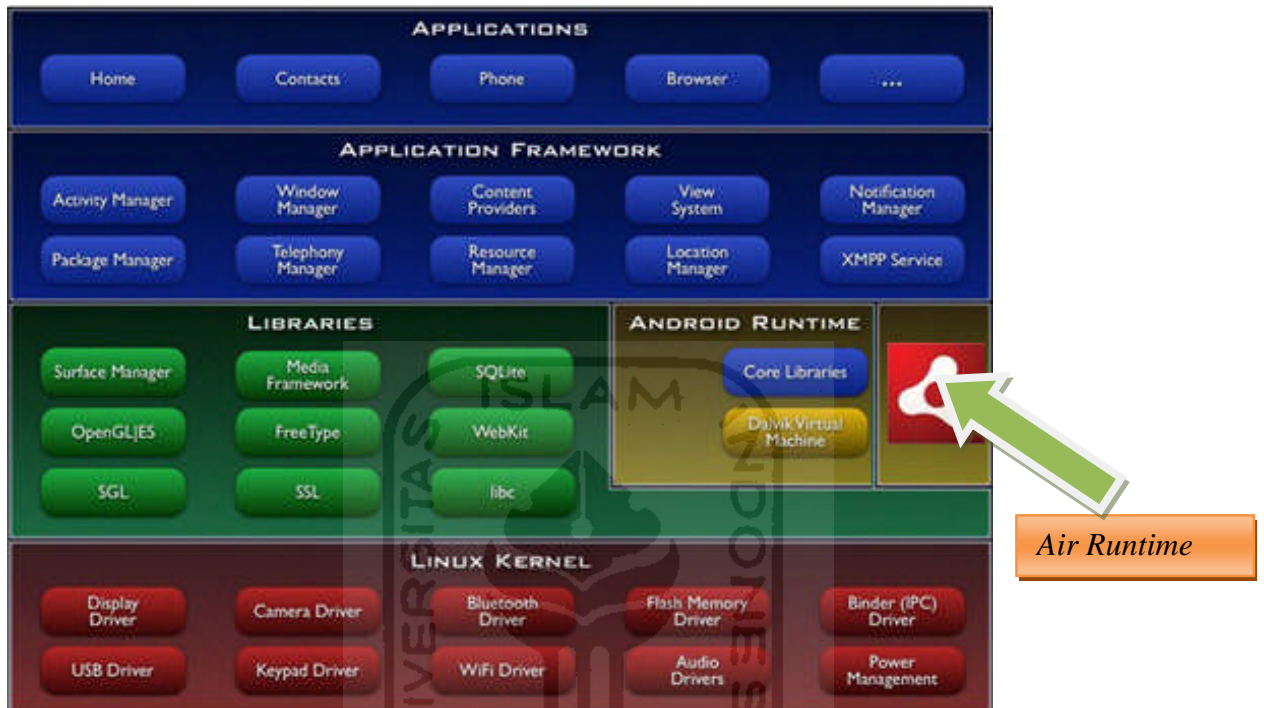
2.5 **Pengembangan Game Dengan Flash dan Air Pada Android**

Pengembangan game adalah proses di mana sebuah game dibuat. Pengembangan game dilakukan oleh seorang developer, bisa 1 orang atau 1 perusahaan besar. Biasanya, game komersial berskala besar dibuat oleh tim pengembang dalam sebuah perusahaan yang mengkhususkan pada game komputer atau konsol.

Dalam pengembangan *game* dalam Android OS sendiri dapat menggunakan berbagai cara, beberapa diantaranya adalah dgn menggunakan Java yaitu Eclipse, maupun menggunakan Flash ActionScript 3.0.

Penggunaan Flash ActionScript3.0 dalam hal ini sebenarnya sebagai jembatan antara Air Application dgn Android, karena file akan dikonversi menjadi aplikasi Air yang berekstensi .apk. Kemudian untuk *tester*-nya akan digunakan Android yang didapat dari Google.

2.6 Arsitektur Platform Android dan *Air Runtime*



Gambar 2.1 *Android Architecture*

(Sumber : <http://images.linuxidx.com/go.php>)

Air Application hanya dapat berjalan pada *Air Runtime Library* yang telah diinstall pada *device* Android, dan hanya dapat berjalan pada Android versi Froyo (2.2) keatas karena dukungan *library* yang telah memadai untuk mengeksekusi aplikasi.

2.7 Bahasa Pemrograman

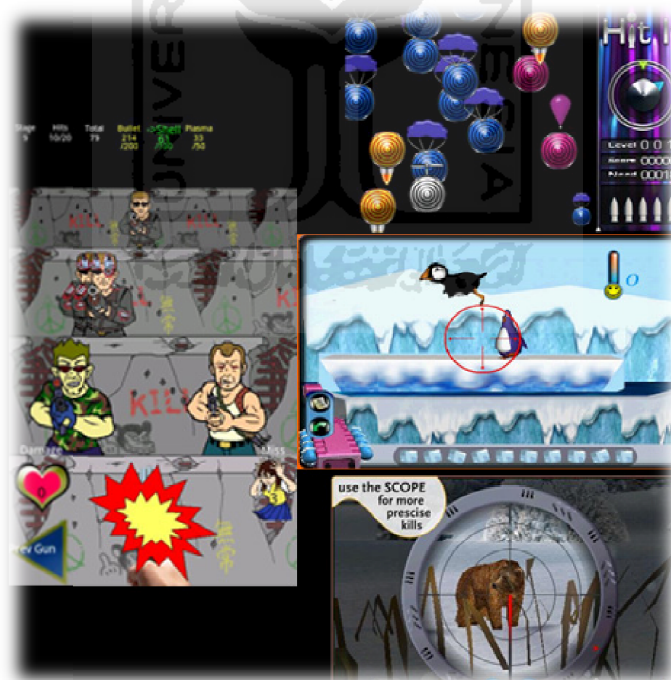
Bahasa Pemrograman yang digunakan pada pembuatan *game* ini adalah Actionscript 3.0 yang ditulis dalam software Flash. Kemudian bahasa ActionScript 3.0 tersebut akan dikonversi menjadi aplikasi Air oleh ekstensi pada flash yang disebut *Air Extension for Android*, hal ini dilakukan agar Aplikasi yg akan dcompile

memiliki hasil akhir menjadi *native application* yg memiliki ekstensi .apk yang dikenali langsung oleh *applicaton manager* Android.

2.8 Game Shooter

Game Shooter adalah jenis game yang menggunakan pistol/senapan ataupun senjata ledak sejenis sebagai daya tarik utama permainannya, yaitu dgn membidik sasaran dgn *crosshair* maupun dengan tembakan langsung tanpa bidikan dengan menghamburkan peluru.

Beberapa game shooter yang tersedia di Android market sampai saat ini adalah *Penguin shooter*, *Finger shooting*, *Mad Shooting*, dan *Deer Hunter*, *screenshoot* bisa dilihat pada gambar 2.2.



Gambar 2.2 Beberapa Game yang ada di Android market

BAB III

METODOLOGI

3.1 Metode Analisis

Metode Analisis digunakan untuk menguraikan sistem aplikasi game *shooter* pada sistem operasi Android menjadi komponen-komponen untuk diidentifikasi dan dievaluasi permasalahannya. Sistem yang dianalisis adalah sistem yang berisi informasi tentang segala sesuatu yang berkaitan dengan pengekplorasian desain *Graphic User Interface* (GUI), dalam bentuk multimedia yang berupa *coding*, gambar, animasi, vector, dan suara yang diaplikasikan pada sistem operasi Android .

Sistem yang dianalisis adalah cara kerja aplikasi game *shooter* yang berjalan pada sistem operasi Android berbasis *touchscreen*. Tahap analisis ini merupakan tahapan yang paling penting dalam program yang dirancang, karena jika terjadi kesalahan dalam tahap ini akan menyebabkan terjadinya kesalahan pada tahap selanjutnya. Karena itu dibutuhkan suatu metode sebagai pedoman dalam mengembangkan sistem yang dibangun.

3.2 Analisis Kebutuhan

Dari metode dan langkah yang dilakukan maka hasil analisis yang diperoleh adalah kebutuhan *input*, fungsi dan kinerja, kebutuhan *output*, kebutuhan perangkat keras, dan kebutuhan perangkat lunak

3.2.1 Analisis Kebutuhan Input

Input adalah suatu bentuk masukan dan berupa data yang telah ada yang dibutuhkan oleh perangkat lunak sehingga dapat mencapai tujuan yang diinginkan. Masukan dari aplikasi game *shooter* ini adalah masukan yang dihasilkan dari interaksi pengguna pada layar *touchscreen smartphone*.

3.2.2 Analisis Kebutuhan Fungsi dan Kinerja

Fungsi dan kinerja yang dibutuhkan pada aplikasi ini adalah sebagai berikut:

1. Membaca *input* dari *touchscreen*.
2. Pergerakan target sasaran pada layar secara *random*.
3. Interaksi dengan objek lain, seperti pada target sasaran.
4. Animasi pada benda, seperti animasi karakter dan menu.
5. *Mute* dan *unmute* pada suara ketika aplikasi dijalankan.
6. Score untuk menghitung nilai tiap objek sasaran yg terkena interaksi.
7. Tombol *reload* untuk mengisi peluru kembali.
8. Animasi untuk pembuka pada *story*.

3.2.3 Analisis Kebutuhan Output

Keluaran (*output*) dari aplikasi ini adalah berupa game yang berjalan pada device Android menggunakan layar *touchscreen*. Objek pada aplikasi ini berupa teks, *vector* dan grafik disertai animasi.

3.2.4 Analisis kebutuhan Perangkat Keras

Perangkat keras digunakan sebagai alat pengolah data yang bekerja secara otomatis mengolah data yang berbentuk teks, gambar, audio dan animasi. Diperlukan komputer yang memiliki kemampuan mengolah grafis yang baik. Spesifikasi komponen yang dibutuhkan akan dibagi menjadi dua bagian yaitu dari sisi pembangunan aplikasi dan dari sisi pengguna.

Adapun spesifikasi komponen perangkat keras yang diperlukan untuk pembangunan aplikasi adalah sebagai berikut:

1. Piranti input berupa *keyboard* dan *mouse*
2. Piranti output berupa monitor dengan resolusi minimal 1024x768, dan *speaker*
3. Kartu grafis yang mendukung *Direct 9*, dengan memori minimal 512MB

4. Prosesor minimal dual core dan memiliki kecepatan 2 Ghz
5. Memori RAM minimal 2GB
6. Hardisk yang memiliki ruang kosong minimal 8GB

Spesifikasi komponen perangkat keras yang diperlukan oleh pengguna yang mengakses aplikasi game *shooter* ini adalah sebagai berikut:

1. Piranti input berupa *touchscreen*
2. Piranti output berupa *screen* dengan resolusi minimal 240x400, dan *speaker*
3. Google Android 2.2 (Froyo) *operating system*
4. ARMv7-A *processor* 600mhz
5. Memori RAM minimal 256MB
6. SD Card yang memiliki ruang kosong minimal 20MB
7. H.264 and AAC hardware decoders

3.2.5 Analisis Kebutuhan Perangkat Lunak

Selain perangkat keras, perangkat lunak juga diperlukan dalam pembangunan aplikasi dan dalam menjalankan aplikasi. Perangkat lunak yang dibutuhkan akan dibagi menjadi dua sisi, pada sisi pembangunan aplikasi dan dari sisi pengguna aplikasi.

Perangkat lunak yang dibutuhkan pada pembangunan aplikasi adalah sebagai berikut:

1. *Sistem operasi*, sistem operasi yang dibutuhkan untuk pembangunan aplikasi adalah antara lain Windows XP, 7 (seven), Mac OSX tiger atau Leopard
2. *Android SDK dan AVD Manager*, digunakan untuk menagtur pembuatan emulator Android dan mengenali koneksi USB dari PC / laptop ke device Android.
3. JDK 5, digunakan untuk menjalankan Android SDK dan AVD manager berserta emulator-nya.

4. Adobe Flash CS5, merupakan aplikasi yang digunakan membangun keseluruhan aplikasi game mulai dari animasi, hingga *coding*.
5. Flashpro_extensionforair, merupakan program tambahan pada Adobe *Flash* untuk mengkonversi *Actionscript 3.0* pada Adobe Flash menjadi AIR agar bisa berjalan dan dapat melakukan instalasi aplikasi secara *native* (memiliki ekstensi program .apk) pada sistem operasi Android.
6. Adobe *illustrator*, merupakan aplikasi yang digunakan untuk mengedit vector dan langsung melakukan export pada Adobe *Flash* .
7. Corel *Draw*, merupakan aplikasi tambahan yang berfungsi untuk melakukan proses *tracing* pada *vector*.
8. Cool Edit Pro v 2.0, merupakan aplikasi tambahan yang berfungsi untuk memotong dan mengatur suara untuk kebutuhan *sound* pada Adobe *Flash*.
9. *Adobe Photoshop*, Adobe Photoshop digunakan untuk mengedit, membuat gambar serta untuk memodifikasi tekstur yang dipakai di Adobe *Flash*.
10. Adobe AIR Runtime 2.5 untuk win32 / Mac, merupakan *enviromntment* yang berjalan pada sistem operasi windows untuk menjalankan aplikasi AIR.
11. Adobe AIR Runtime 2.5 untuk Android, merupakan *enviromntment* yang berjalan pada sistem operasi Android (minimal Android 2.2 / Froyo) untuk menjalankan aplikasi pada *emulator*.

Sedangkan perangkat lunak yang diperlukan oleh pengguna aplikasi adalah:

1. Google Android 2.2 (Froyo) *operating system*.
2. Adobe AIR Runtime 2.5 untuk Android, merupakan *enviromntment* yang berjalan pada sistem operasi Android (minimal Android 2.2 / Froyo) untuk menjalankan aplikasi berbasis AIR.

3.3 Perancangan Perangkat Lunak

3.3.1 Metode Perancangan

Dalam perancangan sistem ini, sistem yang dibangun digambarkan dalam salah satu bentuk diagram UML yaitu menggunakan *activity diagram*.

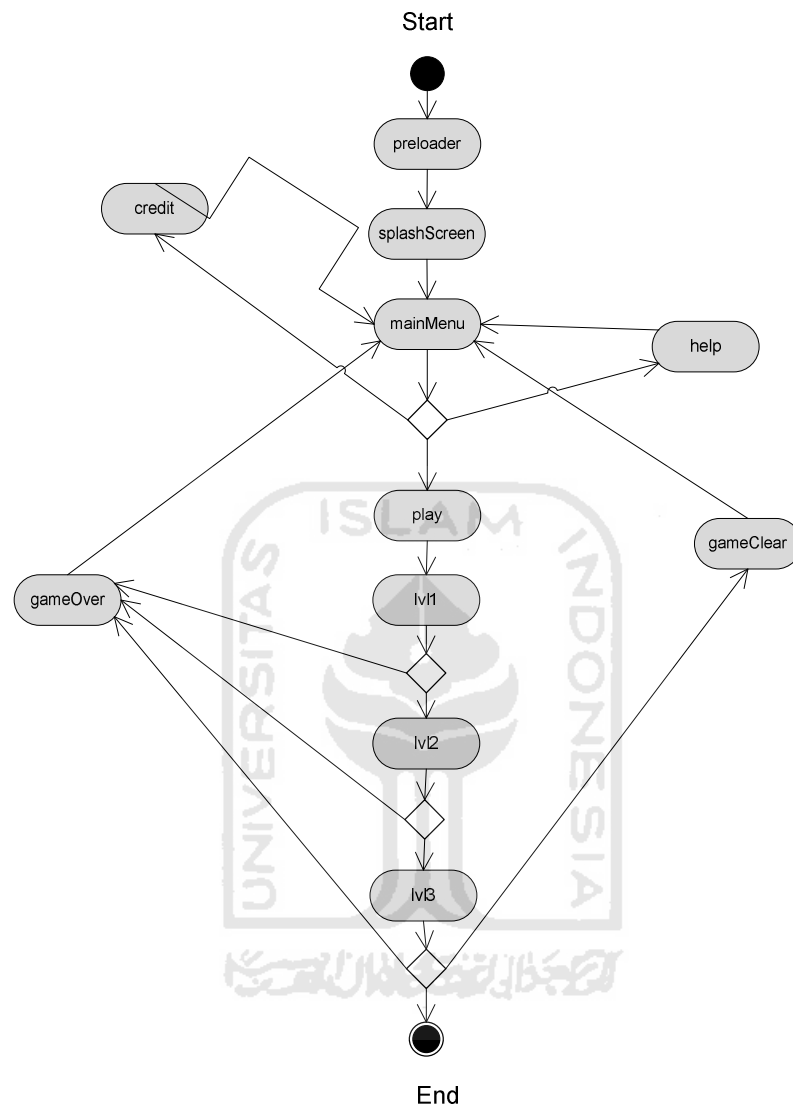
3.3.2 Hasil Perancangan

Hasil pada tahap perancangan berkaitan erat dengan hasil tahap analisis. Karena pada tahap analisis telah ditemukan metode, perangkat lunak yang digunakan, serta fungsi-fungsi yang terdapat pada aplikasi ini. Hasil perancangan aplikasi akan dituangkan dalam *activity diagram* yang bertujuan untuk menggambarkan berbagai aliran aktivitas dalam aplikasi yang sedang dirancang.

3.3.2.1 Activity Diagram

Pada umumnya *activity diagram* tidak menampilkan secara detail urutan proses, namun hanya memberikan gambaran global bagaimana urutan prosesnya. Sehingga seringkali diagram ini digunakan untuk memodelkan aktivitas bisnis dalam level konseptual. Diagram ini sangat mirip dengan *flowchart* karena kita dapat memodelkan suatu alur kerja dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke dalam *state*. Akan tetapi perbedaannya dengan dengan *flowchart* adalah *activity diagram* dapat mendukung perilaku paralel sedangkan *flowchart* tidak.

Tujuan pembuatan *activity diagram* ini adalah untuk menggambarkan semua aktivitas global yang terjadi dalam sebuah sistem. Dengan melihat *activity diagram*, pengguna dapat mengetahui apa saja yang dapat dilakukan oleh sebuah sistem.



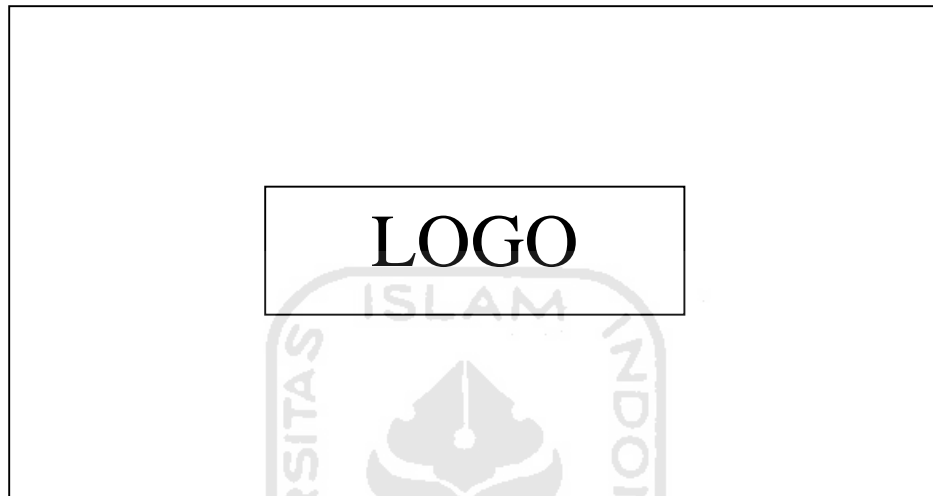
Gambar 3.1 Activity Diagram

3.3.2.2 Perancangan Antarmuka

Perancangan antarmuka bertujuan untuk memberikan kemudahan dalam mengimplementasikan perangkat lunak yang akan dibangun. Antarmuka ini juga berfungsi sebagai sarana interaksi antara manusia dan komputer. Antarmuka pada

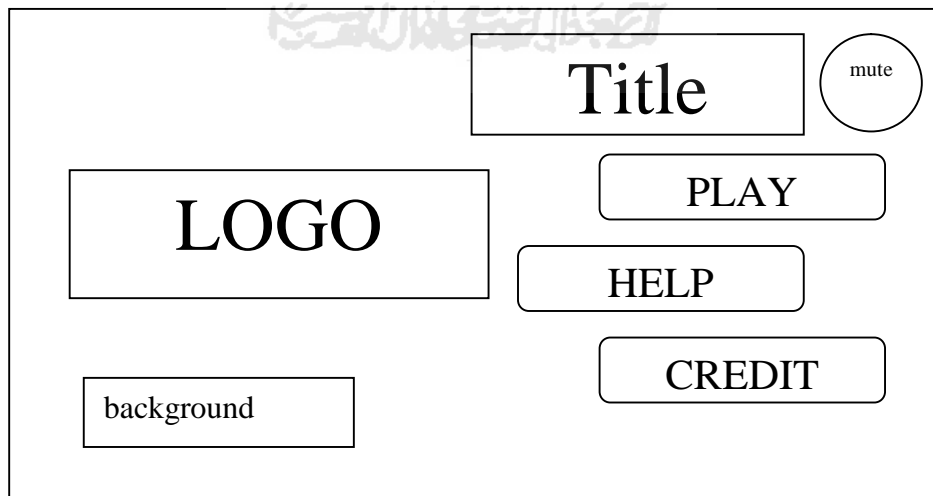
aplikasi game *shooter* ini merupakan gambaran antarmuka yang tampak pada halaman aplikasi.

1. Antarmuka *Splash Screen* :



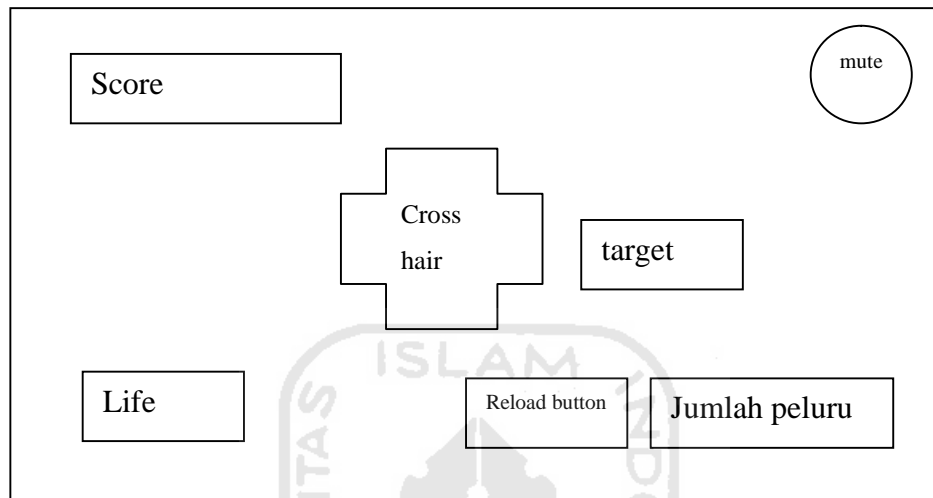
Gambar 3.2 Perancangan antarmuka *Splash Screen*

2. Antarmuka *Main Menu* :



Gambar 3.3 Perancangan antarmuka *Main Menu*

3. Antarmuka *Level* :



Gambar 3.4 Perancangan antarmuka *Level*

BAB IV

HASIL DAN PEMBAHASAN

Hasil dan pembahasan mengutarakan tentang implementasi perangkat lunak yang meliputi batasan implementasi dan implementasi perangkat lunak berupa main menu, rancangan antarmuka, pengujian program dan pengujian menu, analisis kinerja program dan kelebihan serta kekurangan aplikasi.

4.1 Batasan Implementasi

Aplikasi *game* ini dirancang menggunakan Flash CS5. Aplikasi *game* ini memiliki batasan-batasan implementasi yaitu :

1. *Device*

Aplikasi *game* ini hanya dapat berjalan pada device yang memiliki interaksi touchscreen dan minimal memiliki prosesor ARM7 didalamnya.

2. *Screen Resolution*

Aplikasi *game* ini hanya dapat berjalan pada resolusi layar minimal 240x400 (WQVGA400).

3. *Runtime*

Untuk dapat berjalan pada device android memerlukan *Adobe Air Runtime* minimal versi 2.5.

4. *Lite Application*

Untuk dapat memperlancar performa *game* dalam hal mengurangi *lag time* dan *delay* pada saat *game* berjalan maka ada beberapa interaksi dan animasi yang diberi limit.

5. *Operasting System (OS)*

Aplikasi *game* ini berjalan pada OS Android Froyo.

4.2 Tahap Pembuatan Proses

1. Analisis data

Mengumpulkan berbagai literatur tentang *flash* dan *air application* pada android dan apa saja yang diperlukan untuk mewujudkan *cross platform* tersebut.

2. Pembuatan konsep

Mengkonsep *gameplay* dan tema cerita pada aplikasi *game* yang akan dibuat, konsep tema sangat penting pada aplikasi *game* agar pembuatan desain *art* selaras dengan *gameplay*.

3. Desain

Merancang dan membuat *activity diagram* sebagai media perancangan aplikasi.

4. Pemodelan

Membuat *Art* untuk digunakan sebagai visualisasi *game* dan tampilan GUI (*Graphical User Interface*) dengan software Adobe Illustrator, yang dibuat menjadi file *vector*, seperti : logo, *crosshair*, target, tombol, *background*, dll.

5. Aransemen Musik

Merupakan *backsound* dan *sound effect* dari aplikasi *game* ini. Dilakukan *mixing* dengan menggunakan software Cool Edit Pro v 2.0.

6. Pengkodean dan pembuatan *Game Engine*

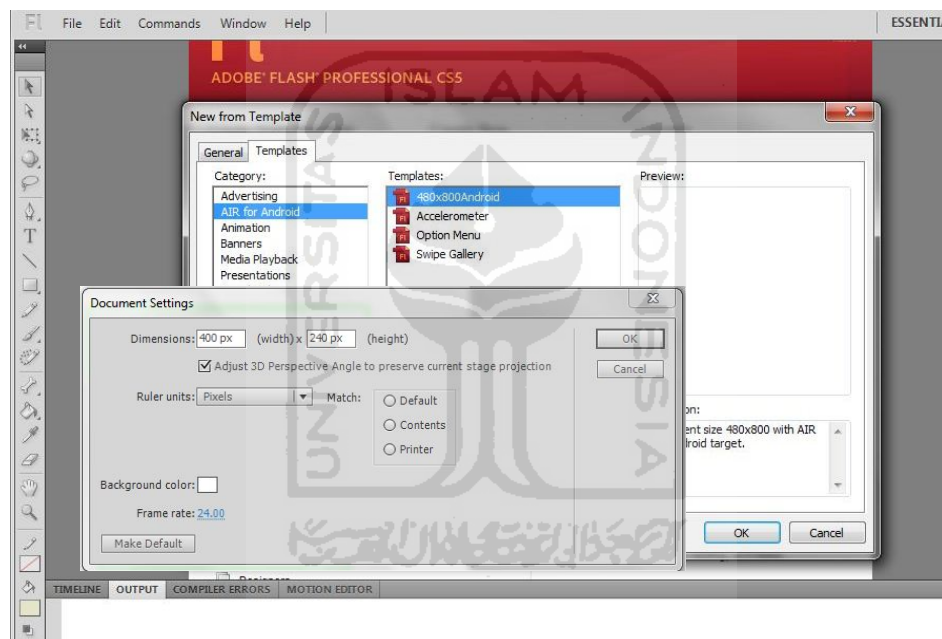
Dengan Flash CS5, aplikasi *game* akan dibangun menggunakan *ActionScript 3.0* dan diatur pada optimasi *render mode* GPU untuk kemudian mengkonversi menjadi *Air Application*, lalu membangun aplikasi native android yang berekstensi *.apk*.

7. Pengujian

Mengadakan pengujian dan menganalisis aplikasi secara sederhana seperti mengujikannya ke beberapa user.

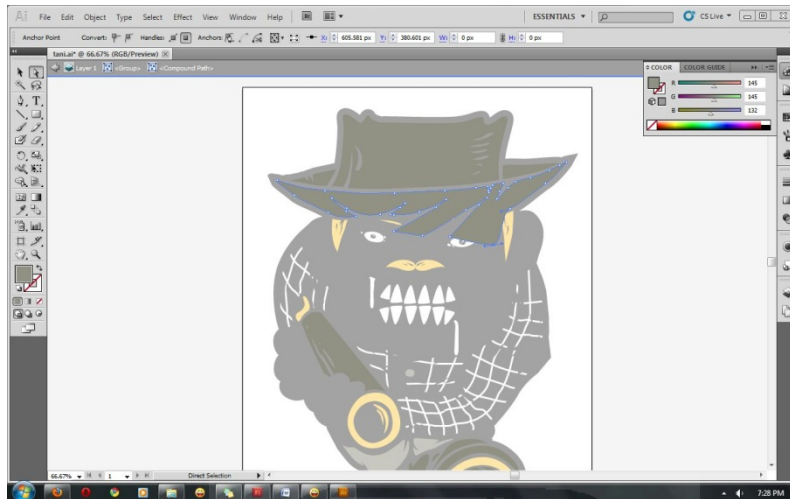
4.3 Implementasi Pembuatan Proses

Pertama-tama yang dilakukan adalah menyiapkan *layout* pada Flash menggunakan *Air for Android extension*. Layout tersebut akan dijadikan dasar aplikasi *game*. Kemudian melakukan *setting layout* untuk disesuaikan dgn resolusi layar device android, resolusi yang tepat yaitu WQGA400 (240 x 400px) untuk orientasi *potrait*, tapi karena akan dibuat untuk orientasi *landscape* maka diatur menjadi (400 x 240px) dengan 24fps. dapat dilihat pada Gambar 4.1



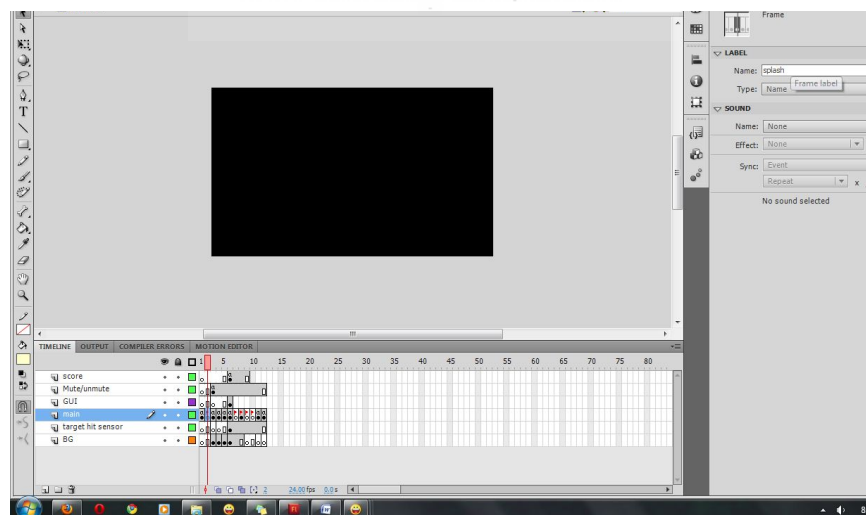
Gambar 4.1 Tahap Pembuatan Layout Dasar Pada Flash

Berikutnya adalah pembuatan *Art* untuk digunakan sebagai visualisasi *game*, seperti : logo, *crosshair*, tombol, target, dan *background*. Pembuatan dilakukan pada Adobe Illustrator sebagai vector agar dapat diekspor sebagai swf pada Flash, dapat dilihat pada Gambar 4.2.



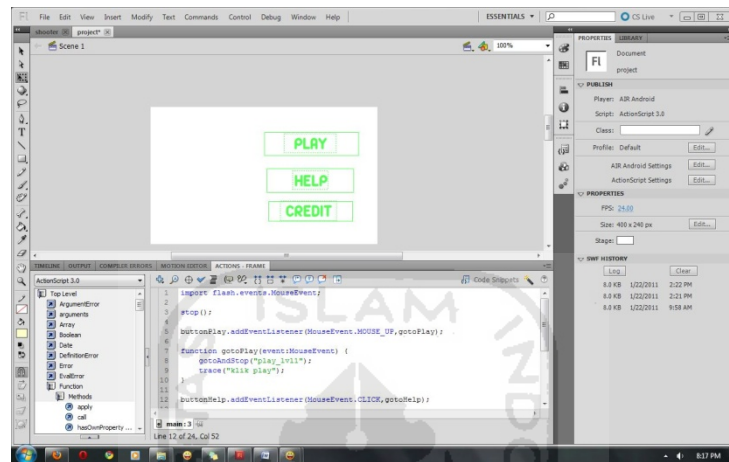
Gambar 4.2 Tahap Pembuatan Art Pada Game

Selanjutnya, penataan dan penamaan *layer* dan label setiap *frame* sebagai agar lebih mudah mengecek dan memilih GUI yang akan digunakan, dan wadah terjadinya aktivitas pada game, seperti : *splash screen*, menu, level1, level2, *game over*, *victory*, *help*. Penataan *frame* ini bertujuan agar saat setiap *event* dalam suatu *frame* habis maka dapat dengan mudah berpindah ke *frame* dengan *event* yang lain, dilihat pada Gambar 4.3



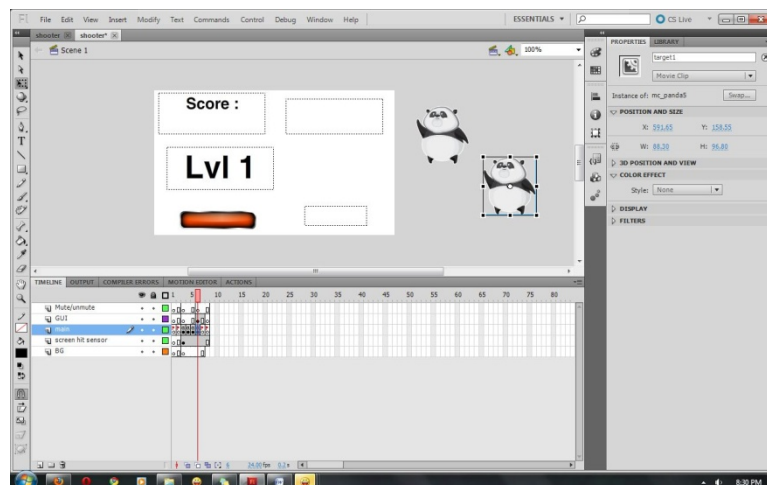
Gambar 4.3 Tahap Layering dan Labeling

Kemudian tahap berikutnya adalah pembuatan *Game Engine* pada Flash. Pembuatan *Game Engine* akan menggunakan art yang sederhana agar lebih mudah dan cepat dalam pengujian *trial and error*-nya. Tahap pertama adalah membuat menu utama terlebih dahulu, dapat dilihat pada gambar berikut :



Gambar 4.4 Tahap Pembuatan Menu Utama

Tahap selanjutnya adalah membuat *level gameplay*, ini adalah inti dari aplikasi *game* yang akan dibuat, disini akan dibuat variasi kecepatan gerak objek target, *array* skor, jumlah *life*, hingga reload *event*. Disini juga akan dibuat percabangan *event* antara *game over* dan lanjut ke *next level / victory event*.



Gambar 4.5 Tahap Pembuatan *Level Gameplay*

Setelah *layout* pada *frame* ditata dan dirasa cukup baik, maka langkah selanjutnya adalah pembuatan pergerakan target. Target *sample* kemudian dimasukkan ke dalam Flash, untuk kemudian digerakkan dan mendeteksi benturan dengan batas pada akhir layar dengan seperti yang tertulis pada halaman lampiran (a).

Tahap berikutnya adalah, pembuatan *drag cursor* sebagai *crosshair* untuk mengarahkan tembakan ke target, disini dibuat 2 objek yaitu cursor untuk mendeteksi *event drag* dan *point* untuk mendeteksi tembakan ke target apakah mengenai target atau tidak, yang berguna untuk *scoring* dan memberi efek suara, kemudian diberikan *script* seperti yang tertulis pada halaman lampiran (b).

pada tahap ini, dapat dilihat pada gambar berikut :

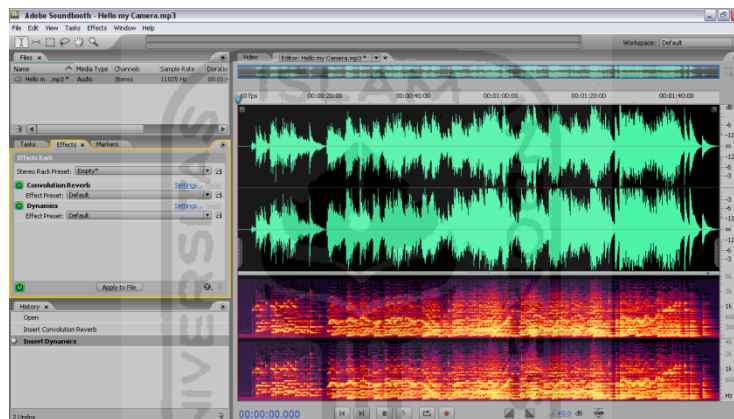


Gambar 4.6 Tahap Pembuatan *Crosshair*

Berikutnya yang tidak kalah penting adalah pengkodean script untuk GUI (*Graphical User Interface*) yang merupakan antarmuka yang tampak saat aplikasi *game* dijalankan. Ditahap ini dibuat 3 macam GUI yaitu *main menu*, *gameplay*, dan *system*. Pada GUI *main menu* memuat 3 tombol menu, yaitu tombol *play*, *help*, dan *credit*. Sedangkan pada GUI *gameplay* memuat tombol *reload*. Terakhir pada GUI

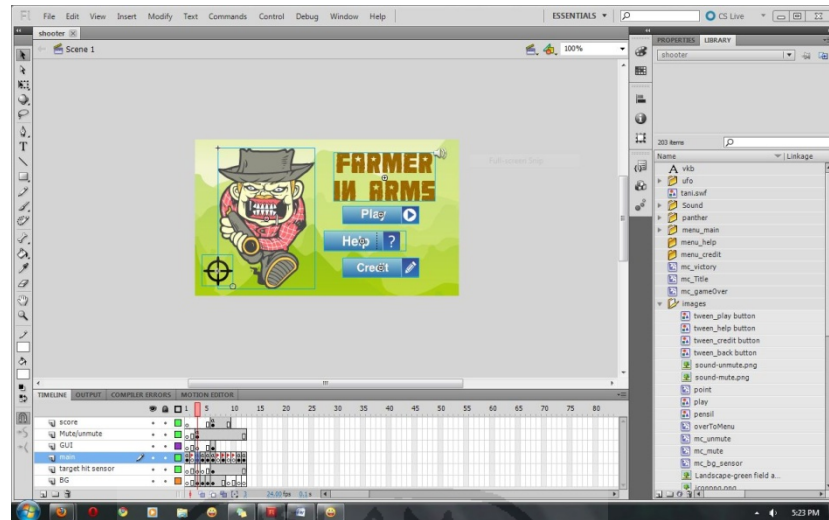
system memuat tombol *mute/unmute*. Adapun *script* yang digunakan pada GUI, seperti yang tertulis pada halaman lampiran (c).

Kemudian, tahap berikutnya adalah *mixing* musik sebagai *background* dan *sound effect* dari aplikasi *game* yang sedang dibangun. Untuk tahap ini, saya menggunakan *software* Cool Edit Pro. Untuk tahapan ini, dapat dilihat pada gambar berikut



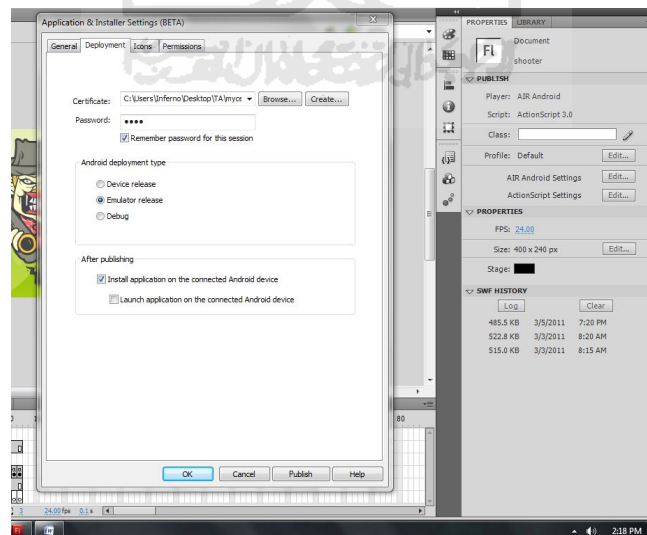
Gambar 4.7 Tahap *Mixing* Musik

Setelah *game engine*, *art*, dan *sound* selesai, kini tinggal menggabungkan semuanya, dimulai dari mengimpornya ke library kemudian memasukkan *art* ke main menu, dan setiap frame *game engine*, kemudian terakhir memasukkan *sound* agar game menjadi lebih hidup. Untuk tahapan ini, dapat dilihat pada Gambar 4.8



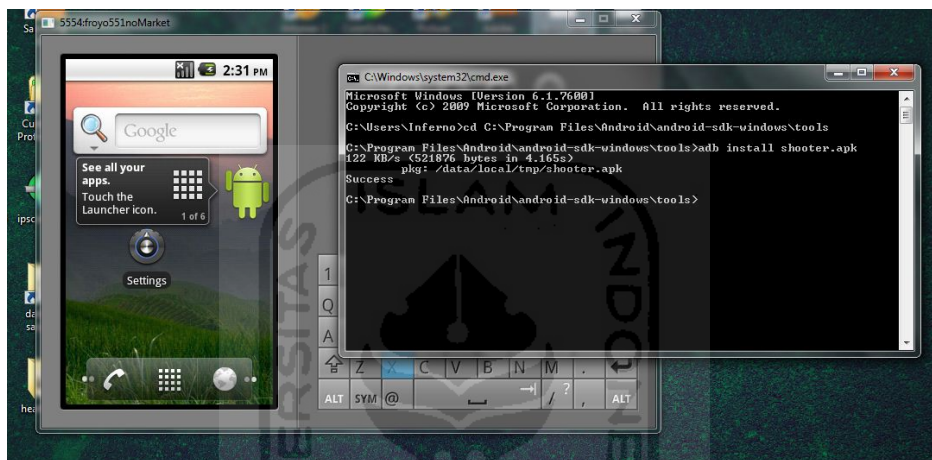
Gambar 4.8 Tahap Penggabungan *Content Game*

Setelah penggabungan selesai, aplikasi akan di-*publish* menjadi file .apk agar bs diinstall pada *device* Android. Pada saat mem-*publish* aplikasi *game*, atur pada “*Android Deployment Type*” menjadi *emulator release*. Untuk tahapan ini, dapat dilihat pada Gambar 4.9



Gambar 4.9 Tahap *Publishing*

Terakhir, setelah aplikasi *game* ter-*publish* maka akan dilakukan tahap pengujian pada emulator menggunakan SDK Android, dengan membuat *Android Virtual Device* (AVD) dan meng-*installnya* menggunakan command promp (CMD) pada Windows menggunakan perintah : `adb install namaAplikasi.apk`, lalu mengubah orientasi layar AVD menjadi *landscape* menggunakan `Ctrl+F11`. Untuk tahapan ini, dapat dilihat pada Gambar 4.10 dan Gambar 4.11



Gambar 4.10 Tahap *Install To Emulator*



Gambar 4.11 Tahap Pengujian

4.4 Implementasi Interface

Implementasi *Interface* aplikasi *Game Shooter* ini adalah *Main Menu*, beserta *Interface gameplay* dan *system*, adalah sebagai berikut.

4.4.1 Implementasi Main Menu Aplikasi Game Shooter

Implementasi *main menu* aplikasi *Game Shooter* merupakan tampilan utama saat membuka aplikasi *game* ini. Pada tampilan ini, *user* disuguhkan beberapa tombol menu, yaitu :

1. Menu *Play* : Berfungsi untuk mulai memainkan *game*.
2. Menu *Help* : Merupakan menu yang berisi bantuan dan petunjuk penggunaan aplikasi *game*.
3. Menu *Credit* : Berisi sekilas (*credit*) tentang *Game Shooter*.

Implementasi *main menu*, dapat dilihat pada Gambar 4.7 berikut.



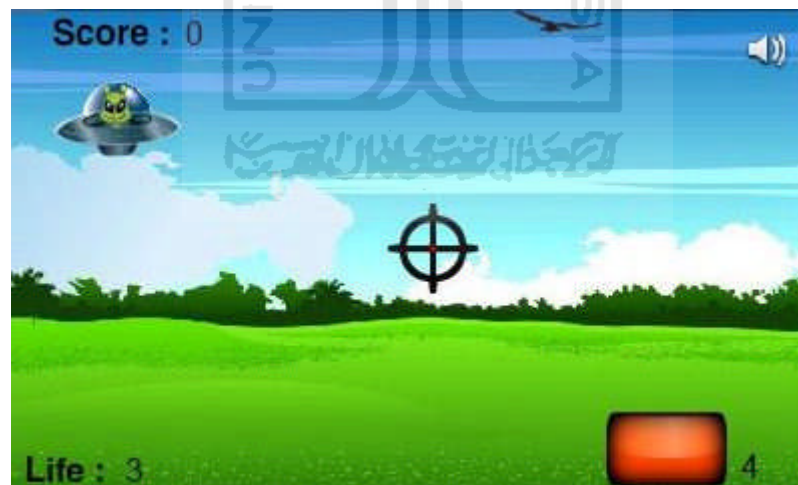
Gambar 4.12 Antarmuka *Game Shooter*

4.4.2 Implementasi Interface Gameplay dan System

Implementasi *interface* pada aplikasi *Game Shooter* merupakan desain antarmuka yang mempunyai beberapa interface utama, yaitu :

1. *Cursor Crosshair* : Berfungsi untuk mengarahkan tembakan kearah target yang sedang bergerak.
2. Tombol *Reload* : Untuk mengisi ulang jumlah peluru kejumlah maksimal.
3. Tombol *Mute / Unmute* : Berfungsi untuk mematikan dan menghidupkan kembali semua *sound* dalam aplikasi *game*.

Implementasi *gameplay* dan *system* dalam aplikasi *Game Shooter*, dapat dilihat pada Gambar 4.8 berikut.



Gambar 4.13 *Interface Gameplay dan System*

Setelah semua *content* digabungkan, maka aplikasi *game* akan dites di emulator SDK Android

4.5 Analisis Kinerja Perangkat Lunak

Untuk menguji sejauh aplikasi *game* dengan pengguna setelah diberi tindakan oleh pengguna ke dalam game.

4.5.1 Analisis Kesesuaian Dengan Landasan Teori

Sesuai dengan landasan teori dalam urutan pembuatan aplikasi *Game Shooter*, analisis yang dilakukan terhadap teori pembuatan *Game Shooter* pada aplikasi *game* yang dibangun adalah

1. Jenis aplikasi

Aplikasi yang dibuat berupa *game* yang berjalan pada *device* dengan sistem operasi Android.

2. Interaksi

Pengguna berinteraksi dengan objek-objek , mulai dari *crosshair* pada aplikasi *game* yang digerakkan menuju sasaran target, hingga tombol *menu* pada *main menu* dan tombol *mute/unmute*.

3. Tipe Ekstensi Aplikasi

Aplikasi yang dibangun dengan ekstensi *file* yang bertipe *.apk* (untuk Sistem Operasi Android). Sehingga memudahkan pengguna, karena dapat dijadikan *native application* pada sistem operasi tersebut.

Setelah dilakukan pengujian sederhana terhadap sistem, dapat diketahui kelebihan dan kekurangan sistem sebagai berikut :

1. Kelebihan Aplikasi

- a) Menggunakan *full* interaksi *touchscreen* untuk pengoperasiannya.
- b) Adanya animasi dalam perpindahan frame sehingga terlihat lebih menarik.
- c) Dilengkapi dengan sound effect yang menunjang *game*.
- d) Dapat di-*install* sebagai aplikasi *native* android bukan file yang dieksekusi lewat memory saja, jadi bisa terlihat di *list application manager*.

2. Kekurangan Aplikasi

- a) Memerlukan Air Runtime untuk dapat berjalan pd device Android, seperti halnya java pada PC.
- b) Banyak animasi dan tampilan yang diminimalisasi agar dapat memperlancar performa.
- c) Memerlukan *resource memory* yang tidak sedikit untuk menjalankannya.

4.5.2 Analisis Kinerja Dengan menggunakan Responden

Analisis kinerja sistem dapat dibuat setelah melalui proses pengujian aplikasi. Pada analisis kinerja aplikasi ini terdapat 10 orang responden yang terdiri dari mahasiswa pengguna *smartphone* Android. Dengan telah diujinya aplikasi ini oleh responden, diharapkan akan didapatkan hasil analisis yang lebih berbobot setelah responden mencoba program aplikasi tersebut. Pertanyaan kuisisioner yang diberikan responden terkait dengan:

1. Tampilan dan desain

Pertanyaan yang diajukan : Apakah menurut anda tampilan dan desain program aplikasi ini bagus dan menarik ?

Tujuan dari pemberian pertanyaan tersebut adalah untuk menilai apakah tampilan dan desain program tersebut sudah menjadi suatu media informasi yang menarik untuk masyarakat umum atau apakah belum .

2. Kemudahan penggunaan aplikasi

Pertanyaan yang diajukan : Apakah anda merasa mudah dalam menggunakan aplikasi ini ?

Tujuan dari pemberian pertanyaan tersebut adalah untuk mengetahui apakah aplikasi ini sudah menjadi suatu aplikasi yang *user friendly*.

3. Kejelasan cerita *game*

Pertanyaan yang diajukan : Bagaimanakah menurut anda cerita tema pada aplikasi game ini menarik?

Tujuan dari pemberian pertanyaan tersebut adalah untuk mengetahui apakah aplikasi sudah cukup dapat menjelaskan tema dari keseluruhan *story* pada *game*.

4. Manfaat utama dari sebuah *game*

Pertanyaan yang diajukan : Apakah menurut anda program game ini dapat menghibur anda?

Tujuan dari pemberian pertanyaan tersebut adalah untuk mengetahui manfaat dari aplikasi ini apakah sudah memberikan informasi yang cukup kepada pengguna.

5. Performa dalam *device* Android

Pertanyaan yang diajukan : Bagaimanakah menurut anda performa (tidak lambat / lag time) game ini ?

Tujuan dari pemberian pertanyaan tersebut adalah untuk mengetahui apakah data-data yang tersaji sudah cukup untuk menambah pengetahuan pengguna

tentang Aplikasi Multimedia Edukasi Perakitan Komputer Berbasis Simulasi 3D ataukah belum.

Tabel 4.1 Tabel Responden

No	Jenis Kelamin	Usia	Status / pekerjaan
1.	Pria	21	Mahasiswa Sumatera
2.	Pria	21	Mahasiswa asal Jawa
3.	Pria	22	Mahasiswa asal kalimantan
4.	Pria	22	Mahasiswa asal kalimantan
5.	Pria	23	Mahasiswa asal Jawa
6.	Wanita	24	Mahasiswa asal Sunda
7.	Pria	22	Mahasiswa asal Jawa
8.	Pria	26	Mahasiswa asal Paembang
9.	Pria	23	Mahasiswa asal Lombok
10.	Wanita	22	Mahasiswa asal Palembang

Tabel 4.2 Tabel Pertanyaan dan Tabel *Poin*

No	Pertanyaan	Jumlah responden yang memilih			Jumlah responden dikalikan dengan poin			Rata-rata
		Kurang	Cukup	Baik	Kurang (1)	Cukup (2)	Baik (3)	
1	Apakah menurut anda tampilan dan desain program aplikasi ini bagus dan menarik?	0	7	3	0 x 1	7 x 2	3 x 3	2,3
2	Apakah anda merasa mudah dalam menggunakan aplikasi ini ?	0	9	1	0 x 1	9 x 2	1 x 3	2,1
3	Bagaimanakah menurut anda cerita tema pada aplikasi game ini?	2	5	3	2 x 1	5 x 2	3 x 3	2.1
4	Apakah menurut anda program game ini dapat menghibur anda?informasi seputar komputer	0	4	6	1 x 1	4 x 2	6 x 3	2,6

5	Bagaimanakah menurut anda performa (tidak lambat / lag time) game ini ?	0	6	4	0 x 1	6 x 2	4 x 3	2,4
---	---	---	---	---	-------	-------	-------	-----

Untuk memudahkan proses penghitungan hasil kuisisioner, maka untuk setiap jawaban yang diberikan oleh kesepuluh orang responden diberikan *range* nilai sebagai berikut :

Nilai 1 untuk jawaban kurang

Nilai 2 untuk jawaban cukup

Nilai 3 untuk jawaban baik

Nilai tersebut kemudian digunakan untuk menghitung nilai rata-rata dari jawaban responden, rumus untuk menghitung nilai rata-rata tersebut adalah:

$$\bar{X} = \sum_{i=1}^n \frac{(S_i)x(r_i)}{n}$$

Keterangan :

\bar{X} = rata-rata nilai

S = skala nilai

r = responden

n = jumlah responden

4.3.3 Hasil Pengujian

Dari Tabel Pertanyaan dan Tabel *Poin* dapat disimpulkan :

1. Tampilan dan desain

Data yang diperoleh terhadap sepuluh responden mengenai tampilan dan desain pada aplikasi ini terdapat tujuh responden menjawab cukup, tiga responden menjawab baik dan tidak ada responden yang menjawab kurang baik dalam tampilan dan desain aplikasi ini. Dari hasil ini dapat disimpulkan bahwa tampilan dan desain aplikasi ini cukup baik karena hasil kuisisioner menunjukkan nilai rata-rata 2.3.

2. Kemudahan penggunaan aplikasi

Dari hasil kuisisioner terhadap sepuluh responden mengenai kemudahan dalam penggunaan aplikasi ini terdapat sembilan responden menjawab cukup, satu responden menjawab baik dan tidak ada responden yang menjawab kurang baik dalam kemudahan penggunaan aplikasi Ditunjukkan dengan nilai rata-rata untuk pertanyaan ini adalah 2.1.

3. Kejelasan cerita *game*

Dari hasil kuisisioner terhadap sepuluh responden mengenai kejelasan simulasi pada aplikasi ini terdapat lima responden menjawab cukup, tiga responden menjawab baik dan dua responden yang menjawab kurang baik dalam kejelasan cerita *game*. Ditunjukkan dengan nilai rata-rata untuk pertanyaan ini adalah 2.1.

4. Manfaat utama dari sebuah *game*

Dari hasil kuisisioner terhadap sepuluh responden mengenai manfaat program dari segi aplikasi ini sebagai hiburan, terdapat empat responden menjawab cukup, enam responden menjawab baik dan tidak ada responden yang menjawab kurang baik dalam kejelasan pada simulasi. Ditunjukkan dengan nilai rata-rata untuk pertanyaan ini adalah 2.6.

6. Performa dalam *device* Android
7. Dari hasil kuisioner terhadap sepuluh responden mengenai Performa dalam *device* Android ini, terdapat enam responden menjawab cukup, empat responden menjawab baik dan tidak ada responden yang menjawab kurang baik dalam kejelasan pada simulasi. Ditunjukkan dengan nilai rata-rata untuk pertanyaan ini adalah 2.4.



BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil penelitian, analisis, perancangan sistem dan pembuatan program sampai dengan tahap penyelesaian, maka dapat ditarik beberapa kesimpulan antara lain sebagai berikut :

1. Aplikasi *Game Shooter* Berbasis Sistem Operasi Android Menggunakan Interaksi *Touchscreen* telah berhasil dibangun dan sudah bisa dijalankan pada device *Android*.
2. Aplikasi yang berjalan harus menggunakan *Air Runtime* agar dapat berjalan ketika dieksekusi.
3. Diperlukan resource *memory* yang besar untuk menjalankan *Air Application* dan *Runtime* di device *Android*.

5.2 Saran

Berdasarkan kekurangan dan keterbatasan yang ada di dalam Aplikasi *Game Shooter* ini, maka disarankan :

1. Dalam pengembangan berikutnya dapat digunakan *accelerometer* sebagai interaksi aplikasi selain *touchscreen*, karena semua device *Android* telah dilengkapi dengan *accelerometer*.
2. Karena keterbatasan pada device *Android* untuk pengembangannya sekarang, maka akan lebih baik lagi jika dijalankan pada device *Android* yang lebih baik spesifikasinya terutama *memory* dan *processor*.
3. Animasi yang interaktif dibatasi pada aplikasi ini karena keterbatasan *resource memory*, untuk kedepannya bisa ditambahkan dengan interaksi-interaksi dan animasi yang lebih menarik.

DAFTAR PUSTAKA

- [ANO10a] Anonim. *Installing the SDK (Software Development Kit)*. Diakses 3 Januari 2011 dari <http://developer.android.com/sdk/installing.html>
- [ANO09b] Dazzer. *muting sounds*. Diakses pada 8 Januari 2011 dari <http://www.kirupa.com/forum/showthread.php?t=280533>
- [ANO10c] Anonim. *Using Tween in AS3 to Move Objects*. Diakses pada 8 Januari 2011 dari <http://asgamer.com/2009/using-tweener-in-as3-to-move-objects>
- [ANO10d] Tiar Oktaviandi. *Teknologi Game*. Diakses pada 3 Januari 2011 dari <http://tayerpups.wordpress.com/2010/02/22/teknologi-game/>
- [ANO10e] Anonim. *Making a Complete Flash Game: Menus, UI, Screens, or Windows*. Diakses pada 11 Januari 2011 dari <http://asgamer.com/2009/making-a-complete-flash-game-menus-ui-screens-or-windows>
- [ANO10f] Sierakowski. *Starting with AIR on Android - how to set everything up?* Diakses pada 5 Januari 2011 dari <http://sierakowski.eu/list-of-tips/73-starting-with-air-on-android-how-to-set-everything-up.html>

[ANO11g] Sierakowski. *Collision detection methods, hitTest and hitTestObject alternatives*. Diakses pada 26 Januari 2011 dari <http://sierakowski.eu/list-of-tips/39-collision-detection-methods-hittest-and-hittestobject-alternatives.html>

[ANO11g] Peter De Haan. *Using the Timer Class In ActionScript 3.0*. Diakses pada 18 Januari 2011 dari <http://blogs.adobe.com/pdehaan/2006/07/using-the-timer-class-in-actionscript-3.html>



LAMPIRAN

(a.) var targetSpeed1:int = 4;

```
target1.addEventListener(Event.ENTER_FRAME, targetLoop1);
```

```
function targetLoop1(e:Event)
{
    if(animTarget1){
        target1.x -= targetSpeed1;
    } else {
        target1.x = target1.x;
    }

    if (target1.hitTestObject(batas1))
    {
        target1.removeEventListener(Event.ENTER_FRAME, targetLoop1);
        removeChild(target1);
        jmlTargetMax--;
        jmlLife--;
    }
    //dan seterusnya hingga target terakhir
```

(b.) cursor.addEventListener(MouseEvent.MOUSE_DOWN, cursorDown);

```
cursor.addEventListener(MouseEvent.MOUSE_UP, cursorUp);
```

```
cursor.addEventListener(MouseEvent.CLICK, cursorClick);
```

```
function cursorClick(e:MouseEvent){
}

```

```
function cursorDown(e:MouseEvent){
```

```
e.target.startDrag();
```

```
}
```

```
stage.addEventListener(Event.ENTER_FRAME, followPoint);
```

```
function followPoint(e:Event) {
```

```
point.x=cursor.x;
```

```

    point.y=cursor.y;
}

function cursorUp(e:MouseEvent){
e.target.stopDrag();
    if( jmlpeluru ==0){
        trace("empty");
        empty.play();
    }
    else{

        if(!point.hitTestObject(target1)&&!point.hitTestObject(target2)&&!point.hitTestObject(target3)&&!point.hitTestObject(target4)&&!point.hitTestObject(target5)&&!point.hitTestObject(target6)&&!point.hitTestObject(target7)&&!point.hitTestObject(target8)&&!point.hitTestObject(target9)&&!point.hitTestObject(target10)&&!point.hitTestObject(target11)&&!point.hitTestObject(target12))
        {
            jmlpeluru--;
            gun.play();
            pb= new ledakan(this.stage,100,mouseX,mouseY);
        }
        else{
            if(point.hitTestObject(target1)){
                jmlpeluru--;
                jmlTargetMax--;
                gun.play();
                target1.removeEventListener(Event.ENTER_FRAME,targetLoop1);
                removeChild(target1);
                arrayscore[0]=arrayscore[0]+200;
                pb= new ledakan(this.stage,100,mouseX,mouseY);
            }
            else{
            }
        }

        //dan seterusnya hingga target terakhir
    }
}

```

(c.)

Main menu :**1. Tombol Play**

```
//menu play
buttonPlay.addEventListener(MouseEvent.CLICK, gotoPlay);

function gotoPlay(event:MouseEvent) {
    trace("play lvl 1");
    buttonPlay.play();
    timerPlay1.start();
    timerPlay2.start();
    sndPlay.play();
}

timerPlay2.addEventListener(TimerEvent.TIMER, timerMenuPlay2);

function timerMenuPlay2(event:TimerEvent):void{
    timerPlay2.stop();
    gotoAndStop("lvl1");
    Tween.addTween(this, { y:0, time: 1 } );
}

timerPlay1.addEventListener(TimerEvent.TIMER, timerMenuPlay1);

function timerMenuPlay1(event:TimerEvent):void{
    timerPlay1.stop();
    Tween.addTween(this, { y:-250, time: 1 } );
}

////////////////////////////////////
```

2. Tombol Help

```
//menu Help
buttonHelp.addEventListener(MouseEvent.CLICK, gotoHelp);

function gotoHelp(event:MouseEvent) {
    trace("to Help");
    buttonHelp.play();
    timerHelp1.start();
    timerHelp2.start();
    sndBase.play();
}

timerHelp2.addEventListener(TimerEvent.TIMER, timerMenuHelp2);

function timerMenuHelp2(event:TimerEvent):void{
    timerHelp2.stop();
    gotoAndStop("help");
    Tween.addTween(this, { y:0, time: 1 } );
}
```

```

}

timerHelp1.addEventListener(TimerEvent.TIMER, timerMenuHelp1);

function timerMenuHelp1(event:TimerEvent):void{
    timerHelp1.stop();
    Tweener.addTween(this, { y:-250, time: 1 } );
}

////////////////////////////////////

```

3. Tombol *Credit*

```

//menu Credit
buttonCredit.addEventListener(MouseEvent.CLICK, gotoCredit);

function gotoCredit(event:MouseEvent) {
    trace("to Credit");
    buttonCredit.play();
    timerCredit1.start();
    timerCredit2.start();
    sndBase.play();
}

timerCredit2.addEventListener(TimerEvent.TIMER, timerMenuCredit2);

function timerMenuCredit2(event:TimerEvent):void{
    timerCredit2.stop();
    gotoAndStop("credit");
    Tweener.addTween(this, { y:0, time: 1 } );
}

timerCredit1.addEventListener(TimerEvent.TIMER, timerMenuCredit1);

function timerMenuCredit1(event:TimerEvent):void{
    timerCredit1.stop();
    Tweener.addTween(this, { y:-250, time: 1 } );
}

```

Gameplay :

1. Tombol *Reload*

```

//reload button
var jmlpeluru:int = 4;

peluru.text = jmlpeluru.toString(10);
reload.addEventListener(MouseEvent.CLICK, isiPeluru);

function isiPeluru(e:MouseEvent)
{
    if (jmlpeluru ==0){
        jmlpeluru=jmlpeluru+4;
    }
}

```

```

        reloading.play();
    }
    else {
    }
}

```

2. Cursor drag crosshair

```

//cursor drag
cursor.addEventListener(MouseEvent.CLICK, cursorClick);
cursor.addEventListener(MouseEvent.MOUSE_DOWN, cursorDown);
cursor.addEventListener(MouseEvent.MOUSE_UP, cursorUp);
function cursorClick(e:MouseEvent){
}

function cursorDown(e:MouseEvent){
e.target.startDrag();
}

stage.addEventListener(Event.ENTER_FRAME, followPoint);
function followPoint(e:Event) {
point.x=cursor.x;
point.y=cursor.y;
}

function cursorUp(e:MouseEvent){
e.target.stopDrag();
if( jmlpeluru ==0){
trace("empty");
empty.play();
}
else{

if(!point.hitTestObject(target1)&&!point.hitTestObject(target2)&&!point.hitTestObject(target3)&&!point.hitTestObject(target4)&&!point.hitTestObject(target5)&&!point.hitTestObject(target6)&&!point.hitTestObject(target7)&&!point.hitTestObject(target8)&&!point.hitTestObject(target9)&&!point.hitTestObject(target10)&&!point.hitTestObject(target11)&&!point.hitTestObject(target12))
{
jmlpeluru--;
gun.play();
pb= new ledakan(this.stage,100,mouseX,mouseY);
}
else{
if(point.hitTestObject(target1)){
jmlpeluru--;
jmlTargetMax--;
gun.play();
target1.removeEventListener(Event.ENTER_FRAME, targetLoop1);
removeChild(target1);
arrayscore[0]=arrayscore[0]+200;
pb= new ledakan(this.stage,100,mouseX,mouseY);
}
else{
}
}
}
}

```

System :

3. Tombol Mute / Unmute

```
//mute button
unmute.addEventListener(MouseEvent.CLICK,diMute);

function diMute(event:MouseEvent) {
    SoundMixer.soundTransform = new SoundTransform(0);
    unmute.visible = false;
}

mute.addEventListener(MouseEvent.CLICK,diUnMute);

function diUnMute(event:MouseEvent) {
    SoundMixer.soundTransform = new SoundTransform(1);
    unmute.visible = true;
}
//
```

