

**PENGEMBANGAN APLIKASI GAME FLASH “SPACESHIP
IMPACT” MENGGUNAKAN ACCELEROMETER SENSOR PADA
PONSEL ANDROID**

SKRIPSI



Disusun Oleh :

ANGGA PUTRA KUSUMA

NIM : 06.12.643



**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

2012

RESEARCH REPORT ON THE EFFECTS OF
THE 1960-1961 WINTER STORMS ON
THE ECONOMY OF THE
UNITED STATES

CONFIDENTIAL

CONFIDENTIAL

RESEARCH REPORT ON THE EFFECTS OF

THE 1960-1961 WINTER STORMS ON

THE ECONOMY OF THE UNITED STATES

CONFIDENTIAL

RESEARCH REPORT ON THE EFFECTS OF

THE 1960-1961 WINTER STORMS ON

CONFIDENTIAL

LEMBAR PERSETUJUAN

PENGEMBANGAN APLIKASI GAME FLASH “SPACESHIP IMPACT” MENGGUNAKAN ACCELEROMETER SENSOR PADA PONSEL ANDROID

SKRIPSI

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik*

Disusun Oleh :

ANGGA PUTRA KUSUMA

NIM : 06.12.643

Mengetahui

Ketua Jurusan Teknik Elektro S-1



Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.101880089

Diperiksa dan Disetujui

Dosen Pembimbing I

Dosen Pembimbing II

(Dr. Eng. Aryunto Soetedjo, ST, MT)
NIP.Y.1030800417

(Michael Ardita, ST, MT)
NIP.1031000434

**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2012**

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : Angga Putra Kusuma

NIM : 06.12.643

Program Studi : Teknik Elektro S-1

Konsentrasi : Teknik Komputer dan Informatika S-1

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, 20 – Februari - 2012

Yang membuat Pernyataan,



Angga Putra Kusuma

NIM : 06.12.643

PENGEMBANGAN APLIKASI GAME FLASH “SPACESHIP IMPACT” MENGUNAKAN ACCELEROMETER SENSOR PADA PONSEL ANDROID

Angga Putra Kusuma

**Jurusan Teknik Elektro S-1, Konsentrasi T.Komputer dan Informatika
Fakultas Teknologi Industri, Institut Teknologi Nasional Malang
Jln. Raya Karanglo Km 2 Malang
unyuqueangga@live.com**

Dosen Pembimbing :

- 1. Dr. Eng. Aryuanto Soetedjo, ST, MT**
- 2. Michael Ardita, ST, MT**

Abstrak

Desain karakter sangat penting dalam menciptakan rasa ketertarikan dalam sebuah *game*. Banyak sekali tipe dan jenis *game* yang sudah beredar saat ini khususnya untuk jenis *game arcade*, *game* bisa dijalankan dengan berbagai media seperti pada PC, dan Smartphone dengan berbagai macam tipe *sistem operasi* yang berbeda-beda dan tidak semua jenis *game* yang sama bisa dimainkan pada *sistem operasi* yang berbeda termasuk pada *sistem operasi Android* yang tergolong baru untuk media Smartphone, berdasarkan kenyataan itu, maka dibuatlah pengembangan *game flash* SPACESHIP IMPACT ini dengan menggunakan sensor *accelerometer* sebagai kendali pada permainannya. *Game* yang telah dibuat dengan menggunakan software Adobe Flash atau Macromedia Flash pada umumnya hanya menampilkan desain karakter 2D, dan *game* ini juga dapat bekerja dengan baik pada ponsel Android yang dilengkapi dengan fitur *Accelerometer*.

Kata Kunci : ANDROID, Adobe Flash, Adobe Device Central, Accelerometer.

KATA PENGANTAR

Alhamdulillah, puji syukur kehadiranMu Ya Allah yang telah memberikan Rahmat dan HidayahNya, sehingga penulis dapat menyelesaikan skripsi dengan judul **“PENGEMBANGAN APLIKASI GAME FLASH “SPACESHIP IMPACT” MENGGUNAKAN ACCELEROMETER SENSOR PADA PONSEL ANDROID”** dengan lancar. Skripsi merupakan persyaratan kelulusan Studi di Jurusan Teknik Elektro S-1 Konsentrasi Teknik Komputer dan Informatika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

Keberhasilan penyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terima kasih kepada :

1. Bapak Ir. Soeparno Djiwo, MT selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Sidik Noertjahjono, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang
3. Bapak Ir. Yusuf Ismail Nakhoda, MT selaku Ketua Jurusan Teknik Elektro S-1
4. Bapak Dr. Eng. Aryuanto Soetedjo, ST, MT selaku Dosen pembimbing I
5. Bapak Michael Ardita, ST, MT selaku Dosen Pembimbing II.
6. Ayah dan Ibu serta saudara-saudara yang selalu memberikan do'a restu, dorongan dan semangat.

Teman-teman dan semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penulis semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan dan pembaca.

Malang, Februari 2012

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAKSI	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
DAFTAR LAMPIRAN	x
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	1
1.3. Batasan Masalah	2
1.4. Metodologi Penelitian	2
1.5. Tujuan Penelitian	3
1.5.1. Tujuan Umum.....	3
1.6. Sistematika Penulisan.....	3
BAB II LANDASAN TEORI	5
2.1. Pengertian Flash	5
2.2. Pengertian Accelerometer Sensor	6
2.3. Pengertian Android	8
2.3.1. Android versi 1.1	9
2.3.2. Android versi 1.5 (Cupcake)	10
2.3.3. Android versi 1.6 (Donut)	10

2.3.4. Android versi 2.0/2.1 (Eclair)	10
2.3.5. Android versi 2.2 (Froyo:Frozen Yoghurt)	11
2.3.6. Android versi 2.3 (Gingerbread)	11
2.3.7. Android versi 3.0/3.1 (Honeycomb)	12
2.3.8. Android versi 4.0 (ICS:Ice Cream Sandwich)	12
2.4. Desain Karakter	12
2.5. Grafik 2D	12
2.5.1. Pengertian Grafik 2D	12
2.6. Video Game	13
2.6.1. Pengertian Game dan Video Game	13
2.6.2. Macam – Macam Game	13
2.6.2.1. Game 2D	13
2.6.3. Platform	14
2.6.4. Genre	15
2.6.5. Arcade Game	15
2.7. Animasi dan Multimedia	15
2.7.1. Pengertian Animasi	15
2.7.1.1. Macam – Macam Animasi	15
2.7.2. Pengertian Multimedia	16
2.8. Software Pendukung	16
2.8.1. Adobe Flash.....	16
2.8.1.1. Tampilan dan Area Kerja Adobe Flash.....	16
2.8.1.2. Metode Animasi dalam Flash.....	18
2.8.1.3. Action Script	18
2.8.1.3.1. Area Kerja Action Script.....	19

2.8.1.3.2. Struktur Bahasa Action Script.....	21
2.8.1.3.3. Tanda Baca dalam Action Script.....	21
2.8.1.3.4. Variabel dan Tipe Data	22
2.8.1.3.5. Operator.....	23
2.8.1.3.6. Function.....	23
BAB III PERANCANGAN GAME	25
3.1. Konsep Game	25
3.2. Visualisasi Konsep Game	26
3.3. Alur Game	27
3.4. Pembuatan Karakter	31
3.4.1. Karakter Dalam Game	31
3.4.2. Hasil Pembuatan Karakter	33
3.5. Pembuatan Background.....	34
3.6. Pembuatan Movie Clip	35
3.7. Logika Dasar	37
3.8. Penerapan Action Script	38
3.8.1. Action Script Persiapan Permainan	39
3.8.2. Action Script Persiapan Karakter Utama (Spaceship)	39
3.8.3. Action Script Menggerakkan Karakter Utama	40
3.8.4. Action Script Persiapan Tembakan (Laser)	40
3.8.5. Action Script Gerakan Laser	41
3.8.6. Action Script Persiapan Musuh (Space UFO).....	41
3.8.7. Action Script Gerakan Musuh	42
3.8.8. Action Script Tumbukan Laser dengan Enemy	43

3.8.9. Action Script Membuat Ledakan	44
3.8.10. Action Script Menu Skor	44
3.8.11. Action Script Menampilkan Hasil Akhir Permainan	45
3.8.12. Action Script Menampilkan Option Menu	46
3.8.13. Action Script Keluar Aplikasi Pada Ponsel	47
3.9. Finishing Game	47
BAB IV IMPLEMENTASI DAN PENGUJIAN GAME	48
4.1. Implementasi Game “Spaceship Impact”	48
4.2. Pengujian Game “Spaceship Impact”	49
4.3. Tampilan Aplikasi Game “Spaceship Impact”	50
4.3.1. Icon Game Pada Ponsel	50
4.3.2. Menu Utama	51
4.3.3. Halaman Petunjuk Permainan	52
4.3.4. Eksekusi Menu Exit Game	53
4.3.5. Eksekusi Permainan	54
4.3.6. Halaman End Screen	59
4.3.7. Halaman Skor	61
BAB V PENUTUP	62
5.1. Kesimpulan	62
5.2. Saran	62
DAFTAR PUSTAKA	63
LAMPIRAN	-

DAFTAR GAMBAR

2.1. Grafik Komputer 2D dalam Desain Karakter 2D	13
2.2. Tampilan Lembar Kerja Adobe Flash Pro CS5	17
2.3. Action Panel	19
2.4. Tombol – Tombol diatas Area Script Panel	20
3.1. Alur Perancangan Game	25
3.2. Visualisasi Konsep Game	27
3.3. Alur Game	29
3.4. Alur <i>Score</i> dan <i>Level</i>	30
3.5. Membuat Karakter Spaceship	31
3.6. Penggunaan Tools Retangle, Oval, PolyStar	32
3.7. Penggunaan Pen Tools	32
3.8. Contoh Penggunaan Pen Tools	33
3.9. Hasil Desain Karakter 2D	34
3.10. Background Game	34
3.11. Mengimport File ke Library Adobe Flash	36
3.12. Membuat Movie Clip Karakter Utama	36
4.1. Tampilan Game pada Adobe Device Central	48
4.2. Tampilan Icon Game	50
4.3. Tampilan Menu Utama	51
4.4. Tampilan Petunjuk Permainan	52
4.5. Tampilan Eksekusi Menu Exit	53
4.6. Tampilan Eksekusi Permainan	54
4.7. Tampilan Menu Pause	55

4.8. Tampilan Level awal.....	56
4.9. Tampilan sebelum naik <i>Level</i>	57
4.10. Tampilan setelah naik <i>Level</i>	57
4.11. Tampilan <i>Level 2</i> dan setelah naik ke <i>Level 3</i>	58
4.12. Tampilan Skor Akhir	58
4.13. Tampilan Halaman End Screen	60
4.14. Tampilan Halaman Skor Tertinggi	61

DAFTAR TABEL

2.1. Arithmetic Operator	23
2.2. Comparisin Operator	23
3.1. Storyboard Game	26

DAFTAR LAMPIRAN

1. Berita Acara Ujian Skripsi	-
2. Form Perbaikan Skripsi	-
3. Form s-1 sampai dengan Form s-4	-
4. Source Code	-

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pembuatan *game* baik berskala besar atau kecil tidak terlepas dari berbagai elemen yang saling terkait satu sama lainnya. Salah satu elemen terpenting dalam pembuatan sebuah *game* adalah desain karakter. Selain diperlukan ide, perumusan jenis *game*, jalan cerita *game*, serta kemampuan dalam *programming*, desain karakter yang baik dapat menjadi faktor yang menentukan apakah sebuah *game* dapat diterima oleh para pengguna *game* atau tidak.

Desain karakter sangat penting dalam menciptakan rasa ketertarikan dalam sebuah *game*. Banyak sekali tipe dan jenis *game* yang sudah beredar saat ini khususnya untuk jenis *game arcade*, *game* bisa dijalankan dengan berbagai media seperti pada PC, dan Smartphone dengan berbagai macam tipe *sistem operasi* yang berbeda-beda dan tidak semua jenis *game* yang sama bisa dimainkan pada *sistem operasi* yang berbeda termasuk pada *sistem operasi Android* yang tergolong baru untuk media Smartphone.

Flash game yaitu *game* yang dibuat dengan menggunakan *software* Adobe Flash atau Macromedia Flash yang pada umumnya menampilkan desain karakter 2D.

Berdasarkan kenyataan itu, maka penulis mengangkat tema pembuatan *flash game* dengan judul “PENGEMBANGAN APLIKASI GAME FLASH “SPACESHIP IMPACT” MENGGUNAKAN ACCELEROMETER SENSOR PADA PONSEL ANDROID ”. *Game* yang berjenis *Arcade* yang hanya bisa dimainkan oleh satu orang dalam satu ponsel. Misi dalam *game* ini adalah untuk mengalahkan musuh di setiap *level* dengan menghancurkan musuh-musuh dalam jumlah tertentu.

1.2. Rumusan Masalah

Berdasarkan uraian latar belakang di atas dapat dirumuskan yaitu, bagaimana memanfaatkan desain karakter 2D dalam pembuatan *flash game arcade* “ SPACESHIP IMPACT ” pada ponsel ber-sistem operasi *Android* menggunakan Adobe Flash Pro CS5

1.3 Batasan Masalah

Agar diperoleh persamaan persepsi, maka dalam penyusunan laporan ini hanya dibatasi pada beberapa hal :

1. Membahas konsep pembuatan *game* yang meliputi desain karakter, desain tampilan *game* dan alur *game*.
2. Hanya membahas beberapa logika dasar untuk menggerakkan karakter di dalam *game* menggunakan *Accelerometer sensor* pada ponsel *touch screen* khususnya ponsel *Android*.
3. Menggunakan software Adobe Flash Professional CS5 untuk desainnya akan lebih memudahkan dalam pembuatannya sekaligus belajar bagaimana mendesain sebuah *game* yang khususnya di gunakan kedalam sebuah ponsel.
4. Di dalam permainan setiap level-nya dibedakan dari banyaknya jumlah musuh, karena semakin tinggi level-nya semakin banyak musuh yang dimunculkan.

1.4 Metodologi Penelitian

Dalam penyusunan laporan ini, diperlukan data atau informasi yang diperoleh dari berbagai sumber dan cara. Adapun metodologi penelitian ini dilakukan melalui beberapa cara yaitu :

1. Studi Literatur dan Observasi

Studi literatur dilakukan dengan cara mengumpulkan data-data dari buku-buku, majalah, artikel, diktat yang ada kaitannya dengan pembuatan *game flash*. Data-data yang diperoleh berdasarkan studi literatur ini sifatnya teoritis. Sedangkan observasi yaitu melakukan pengamatan langsung terhadap *game-game flash* yang telah ada sebelumnya.

2. Perancangan Game

Perancangan *game* meliputi penyusunan konsep *game*, mendesain tampilan, membuat alur *game* dan proses pembuatan *game* lainnya dari awal hingga akhir.

3. Implementasi Game

Implementasi *game* dilakukan dengan menguji aplikasi perancangan *game* menggunakan Adobe Flash Pro CS5 dan *software* pendukung lainnya.

4. Uji Coba

Uji Coba *game* dilakukan pada ponsel berbasis *Android* untuk mengetahui apakah *game* berjalan dengan baik atau tidak.

1.5 Tujuan Penelitian

1.5.1 Tujuan Umum

Tujuan umum dari penulisan laporan Skripsi ini adalah untuk membuat para pengguna ponsel *touch screen* khususnya ponsel bertipe *android* bisa memainkan game " SPACESHIP IMPACT " yang sebelumnya sudah ada pada ponsel *symbian* atau *java* dan biasanya menggunakan *keypad* atau tombol pada *touch screen*, disini diganti menggunakan *accelerometer sensor* sebagai kendalinya.

1.6 Sistematika Penulisan

Dalam penyusunan laporan Skripsi ini akan diuraikan menjadi beberapa bab, dan masing-masing bab akan dipaparkan dalam beberapa sub bab, diantaranya :

BAB I. Pendahuluan

Dalam bab ini akan menjelaskan latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan laporan Skripsi.

BAB II. Landasan Teori

Dalam bab ini akan membahas dan menjelaskan mengenai dasar teoritis yang menjadi landasan dan mendukung pelaksanaan penulisan laporan Skripsi.

BAB III. Perancangan Game

Dalam bab ini menyampaikan informasi tentang pembuatan *game* dari awal hingga akhir.

BAB IV. Implementasi Game

Dalam bab ini akan membahas pengujian aplikasi perancangan *game* menggunakan Adobe Flash Pro CS5 dan *software* pendukung lainnya.

BAB V. Penutup

Dalam bab ini akan disampaikan kesimpulan dan saran dari keseluruhan bahasan.

BAB II

LANDASAN TEORI

2.1 Pengertian Flash

Adobe Flash (dahulu bernama **Macromedia Flash**) adalah salah satu perangkat lunak komputer yang merupakan produk unggulan Adobe Systems. Adobe Flash digunakan untuk membuat gambar vektor maupun animasi gambar tersebut. Berkas yang dihasilkan dari perangkat lunak ini mempunyai *file extension* .swf dan dapat diputar di penjelajah web yang telah dipasang Adobe Flash Player. Flash menggunakan bahasa pemrograman bernama ActionScript yang muncul pertama kalinya pada Flash 5.

Sebelum tahun 2005, Flash dirilis oleh Macromedia. Flash 1.0 diluncurkan pada tahun 1996 setelah Macromedia membeli program animasi vektor bernama *FutureSplash*. Versi terakhir yang diluncurkan di pasaran dengan menggunakan nama 'Macromedia' adalah Macromedia Flash 8. Pada tanggal 3 Desember 2005 Adobe Systems mengakuisisi Macromedia dan seluruh produknya, sehingga nama Macromedia Flash berubah menjadi Adobe Flash.

Adobe Flash merupakan sebuah program yang didesain khusus oleh Adobe dan program aplikasi standar authoring tool professional yang digunakan untuk membuat animasi dan bitmap yang sangat menarik untuk keperluan pembangunan situs web yang interaktif dan dinamis. Flash didesain dengan kemampuan untuk membuat animasi 2 dimensi yang handal dan ringan sehingga flash banyak digunakan untuk membangun dan memberikan efek animasi pada website, CD Interaktif dan yang lainnya. Selain itu aplikasi ini juga dapat digunakan untuk membuat animasi logo, movie, game, pembuatan navigasi pada situs web, tombol animasi, banner, menu interaktif, interaktif form isian, e-card, screen saver dan pembuatan aplikasi-aplikasi web lainnya. Dalam Flash, terdapat teknik-teknik membuat animasi, fasilitas action script, filter, custom easing dan dapat memasukkan video lengkap dengan fasilitas playback FLV. Keunggulan yang dimiliki oleh Flash ini adalah ia mampu diberikan sedikit code pemrograman baik yang berjalan sendiri untuk mengatur animasi yang ada didalamnya atau digunakan

untuk berkomunikasi dengan program lain seperti HTML, PHP, dan Database dengan pendekatan XML, dapat dikolaborasikan dengan web, karena mempunyai keunggulan antara lain kecil dalam ukuran file outputnya.

Movie-movie Flash memiliki ukuran file yang kecil dan dapat ditampilkan dengan ukuran layar yang dapat disesuaikan dengan keinginan. Aplikasi Flash merupakan sebuah standar aplikasi industri perancangan animasi web dengan peningkatan pengaturan dan perluasan kemampuan integrasi yang lebih baik. Banyak fitur-fitur baru dalam Flash yang dapat meningkatkan kreativitas dalam pembuatan isi media yang kaya dengan memanfaatkan kemampuan aplikasi tersebut secara maksimal. Fitur-fitur baru ini membantu kita lebih memusatkan perhatian pada desain yang dibuat secara cepat, bukannya memusatkan pada cara kerja dan penggunaan aplikasi tersebut. Flash juga dapat digunakan untuk mengembangkan secara cepat aplikasi-aplikasi web yang kaya dengan pembuatan script tingkat lanjut. Di dalam aplikasinya juga tersedia sebuah alat untuk men-debug script. Dengan menggunakan Code hint untuk mempermudah dan mempercepat pembuatan dan pengembangan isi ActionScript secara otomatis. Untuk memahami keamanan Adobe Flash dapat dilihat dari beberapa sudut pandang, berdasarkan beberapa sumber referensi bahwa tidak ada perbedaan menyolok antara HTML dan JavaScript dimana didalamnya terdapat banyak tools yang dapat diambil dari SWF termasuk ActionScript. Sehingga kode data dapat terjamin keamanannya. Oleh sebab itu, semua kebutuhan data yang terdapat dalam SWF dapat diambil kembali melalui server. Keuntungan menggunakan metode yang sama dengan menggunakan aplikasi web yang standar adalah akan menjamin dan mengamankan penyimpanan dan perpindahan data.

2.2 Pengertian Accelerometer Sensor

Accelerometer adalah sebuah perangkat yang mengukur percepatan yang tepat, juga disebut percepatan empat. Sebagai contoh, sebuah accelerometer pada roket mempercepat melalui ruang akan mengukur laju perubahan kecepatan roket relatif terhadap frame inersial acuan. Namun, percepatan tepat diukur dengan accelerometer tidak selalu percepatan koordinat (laju perubahan kecepatan).

Sebaliknya, adalah percepatan yang terkait dengan fenomena berat badan yang dialami oleh massa uji beristirahat dalam bingkai referensi dari perangkat accelerometer. Untuk contoh di mana jenis percepatan berbeda, accelerometer akan mengukur nilai gram dalam arah ke atas ketika tersisa diam di tanah, karena massa di bumi memiliki berat $m * g$. Sebaliknya, accelerometer jatuh bebas gravitasi menuju pusat bumi akan mengukur nilai nol karena, meskipun kecepatan yang semakin banyak, tetapi baringkan dengan kerangka acuan di mana objek memiliki berat.

Accelerometer sehingga mengukur berat per unit (test) massa, sejumlah dengan dimensi percepatan yang kadang-kadang dikenal sebagai kekuatan tertentu, atau g-force (meskipun tidak berlaku a). Cara lain untuk menyatakan ini adalah bahwa dengan mengukur berat badan, accelerometer mengukur percepatan kerangka acuan jatuh bebas (kerangka acuan inersia) relatif terhadap itu sendiri (accelerometer). Ini percepatan terukur bukan akselerasi biasa Newton (dalam tiga dimensi), melainkan empat percepatan, percepatan yang jauh dari jalur geodesik dalam empat dimensi ruang-waktu.

Accelerometers Kebanyakan tidak menampilkan nilai mereka diukur, tetapi penawaran itu ke perangkat lain. Accelerometers nyata juga memiliki keterbatasan praktis dalam seberapa cepat mereka merespon perubahan percepatan, dan tidak bisa menanggapi perubahan di atas frekuensi tertentu perubahan.

Model tunggal dan multi-axis accelerometer yang tersedia untuk mendeteksi besar dan arah percepatan yang tepat (atau g-force), sebagai besaran vektor, dan dapat digunakan untuk orientasi akal (karena arah perubahan berat badan), koordinat percepatan (asalkan menghasilkan g-force atau perubahan g-force), getaran, shock, dan jatuh (kasus di mana perubahan percepatan yang tepat, karena cenderung menuju nol). Accelerometers micromachined semakin hadir di perangkat elektronik portabel dan video game controller, untuk mendeteksi posisi perangkat atau memberikan masukan untuk permainan.

Pasangan accelerometers diperpanjang atas wilayah ruang dapat digunakan untuk mendeteksi perbedaan (gradien) dalam percepatan yang tepat dari frame referensi yang terkait dengan poin. Alat ini disebut gradiometers gravitasi, dan mereka mengukur gradien dalam medan gravitasi. Pasang seperti accelerometers dalam teori mungkin juga mampu mendeteksi gelombang gravitasi.

2.3 Pengertian Android

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk 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 perilis 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).

Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. bekerja pada Google, di antaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android Inc. hanyalah sebagai perangkat

lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler.

Sekitar September 2007 sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010).

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat bergerak (*mobile*) yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android.

2.3.1 Android versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2.3.2 Android versi 1.5 (Cupcake)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

2.3.3 Android versi 1.6 (Donut)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang diintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak; teknologi *text to change speech* (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA).

2.3.4 Android versi 2.0/2.1 (Eclair)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi mobile terbaik (killer apps - aplikasi unggulan). Kompetisi ini berhadiah \$25,000

bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik.

Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah Shazam, Backgrounds, dan WeatherBug. Sistem operasi Android dalam situs Internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh MySpace dan Facebook.

2.3.5 Android versi 2.2 (Froyo: Frozen Yoghurt)

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

2.3.6 Android versi 2.3 (Gingerbread)

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.

2.3.7 Android versi 3.0/3.1 (Honeycomb)

Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (hardware) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom. Perangkat tablet dengan platform Android 3.0 akan segera hadir di Indonesia. Perangkat tersebut bernama Eee Pad Transformer produksi dari Asus. Rencana masuk pasar Indonesia pada Mei 2011.

2.3.8 Android versi 4.0 (ICS :Ice Cream Sandwich)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC.

2.4 Desain Karakter

Desain karakter mengacu pada penampilan, kepribadian seorang tokoh yang digambarkan dalam karakter, dan kemampuannya dalam mempresentasikan dalam bentuk digital virtual. Desain karakter telah menjadi komponen utama dalam sejarah perjalanan karya fiksi.

Komponen inti dalam desain karakter biasanya terdiri atas gender, ras, penampilan yang unik, tetap menjaga kesederhanaan dalam bentuk yang agak rumit, dan mudah diingat bagi yang telah melihatnya.

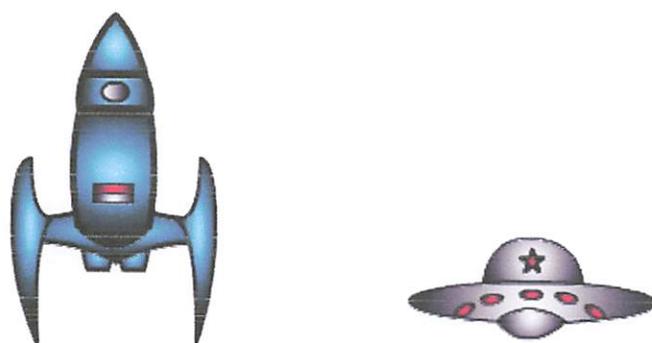
2.5 Grafik 2D

2.5.1 Pengertian Grafik 2D

Grafik komputer 2 dimensi biasa disebut dengan 2D atau bidang adalah bentuk dari benda yang memiliki panjang dan lebar. Grafik 2

Dimensi merupakan teknik penggambaran yang berpatokan pada titik koordinat sumbu x (datar) dan sumbu y (tegak).

Grafik komputer 2D adalah sebuah generasi gambar digital berbasis komputer, yang kebanyakan mengambil objek-objek dua dimensi (2D). Model-model yang digunakan pada desain grafis 2D biasanya tidak mendukung bentuk-bentuk tiga-dimensi, atau fenomena yang bersifat tiga dimensi, seperti pencahayaan, bayangan, pantulan, refraksi, dan sebagainya.



Gambar 2.1 Grafik komputer 2D dalam desain karakter 2D

2.6 Video Game

2.6.1 Pengertian Game dan Video Game

Menurut Agustinus Nilwan dalam bukunya “Pemrograman Animasi dan Game Profesional”, *game* diartikan sebagai suatu aktivitas terstruktur atau juga digunakan sebagai alat pembelajaran yang dapat dikarakteristikan dengan adanya peralatan apa yang digunakan pemain (seperti komputer), adanya peraturan dan adanya pengaruh *skill*, strategi dan keberuntungan pemain.

Sedangkan *video game* adalah sebuah “permainan” dengan dengan tampilan sebuah gambar atau visual yang dapat memberikan respon balik jika diberikan perintah-perintah tertentu menggunakan alat kontrol pada seperangkat sistem elektronik.

2.6.2 Macam-Macam Game

2.6.2.1 Game 2D

Game dua dimensi dapat diketahui berdasarkan ruangnya yang hanya memiliki dua sisi (X dan Y). Sedangkan untuk

gambarnya sendiri dapat menggunakan *vector* maupun *bitmap*. Untuk membuat animasi bergerak (berjalan, melompat, berlari, dan lain-lain) harus dibuat gambar satu persatu yang disebut dengan (*frame*). Kerealisasian gerakan ditentukan dari gambar yang dibuat, jumlah gambar (*frame*) yang digunakan, serta hitungan gambar per detik (*frame per second*). Semakin tinggi hitungan gambar per detik maka semakin halus gerakan yang akan dihasilkan.

Game dua dimensi memiliki dua konsep, yaitu:

1. *Static View*, dimana semua objek berada pada satu bidang dan gerakan karakter utama hanya terbatas pada bidang itu saja. Contoh : *3 Foot Ninja*.
2. *Side Scrolling View*, dimana objek-objek dan gambar latar akan terus bergerak ke kanan dan ke kiri sesuai dengan kecepatan gerakan karakter yang dimainkan. Contoh : *Sonic*.

2.6.3 Platform

Platform adalah sebuah perangkat sistem spesifikasi tertentu, dengan mengkombinasikan berbagai perangkat elektronik atau komputer sehingga bisa menjalankan sebuah *software* dan *video game*. Sistem *platform* antara lain, PC (*Personal Computer*) yang dikoneksikan dengan monitor, kedua *console* yaitu berupa sebuah perangkat khusus untuk memainkan *game* yang dihubungkan dengan televisi, ketiga *arcade* yaitu sebuah kotak kayu dengan dilengkapi monitor dan alat kontrol yang khusus untuk memainkan satu jenis permainan dan dioperasikan dengan memasukkan sebuah koin untuk menyewa, keempat *handheld* yaitu sebuah perangkat khusus *game* yang didesain *portable* dengan layar kecil sehingga mudah dibawa kemana-mana dan dimainkan kapan pun juga, kelima berupa beberapa *gadget* yang berupa alat dengan keperluan khusus tetapi dimasukkan salah satu fitur *game* sebagai pendukung (seperti PDA, *handphone*, kamera digital, MP3 *player* dan sebagainya).

2.6.4 Genre

Genre adalah kategori pada *game* yang didasari pada berbagai macam faktor seperti metode *gameplay* / cara memainkan, tipe pencapaian tujuan, cerita, tampilan, dan banyak lagi yang bisa dikombinasikan sehingga menciptakan *genre* yang berbeda. *Genre game* yang ada saat ini antara lain, *arcade, action, shooting, fighting, racing, simulation, real time strategy, role playing game*.

2.6.5 Arcade Game

Game bertipe *action* menjadikan pemain mengendalikan karakter utama dalam *game* tersebut untuk melakukan beberapa kegiatan (*action*) seperti melompat, menembak dan sebagainya. Contoh dari *game* bertipe *action* antara lain : Super Mario.

2.7 Animasi dan Multimedia

2.7.1 Pengertian Animasi

Menurut Ibiz Fernandes dalam bukunya "*Flash Animation & Cartooning: A creative Guide*", animasi didefinisikan sebagai berikut :
 "Animasi adalah sebuah proses merekam dan memainkan kembali serangkaian gambar statis untuk mendapatkan sebuah ilusi pergerakan."

2.7.1.1 Macam-Macam Animasi

Animasi dibagi menjadi dua, yaitu :

1. Animasi 2D, lebih dikenal dengan istilah kartun yang berasal dari kata '*Cartoon*' artinya gambar yang lucu. Dengan teknik ini, beberapa gambar 2D atau gambar yang mempunyai sudut pandang terbatas ditampilkan berurutan dengan aplikasi pembuat film animasi.
2. Animasi 3D, merupakan pengembangan dari animasi 2D. Dengan animasi 3D, karakter yang diperlihatkan semakin hidup dan nyata, mendekati wujud manusia aslinya. Animasi karakter 3D dapat dilihat dari banyak sudut pandang (atas, bawah, depan, belakang, kiri, kanan, perspektif).

2.7.2 Pengertian Multimedia

Menurut Rosch definisi animasi adalah kombinasi dari komputer dan video, menurut McCornick multimedia secara umum merupakan kombinasi tiga elemen, yaitu suara, gambar dan teks. Definisi lain dari multimedia yaitu dengan menempatkan dalam konteks seperti yang dilakukan Hofstetter, multimedia adalah pemanfaatan komputer untuk membuat dan menggabungkan teks, grafik, audio, gambar bergerak (video dan animasi) dengan menggabungkan *link* dan *tools* yang memungkinkan navigasi, berinteraksi, berkreasi, dan berkomunikasi.

2.8 Software Pendukung

2.8.1 Adobe Flash

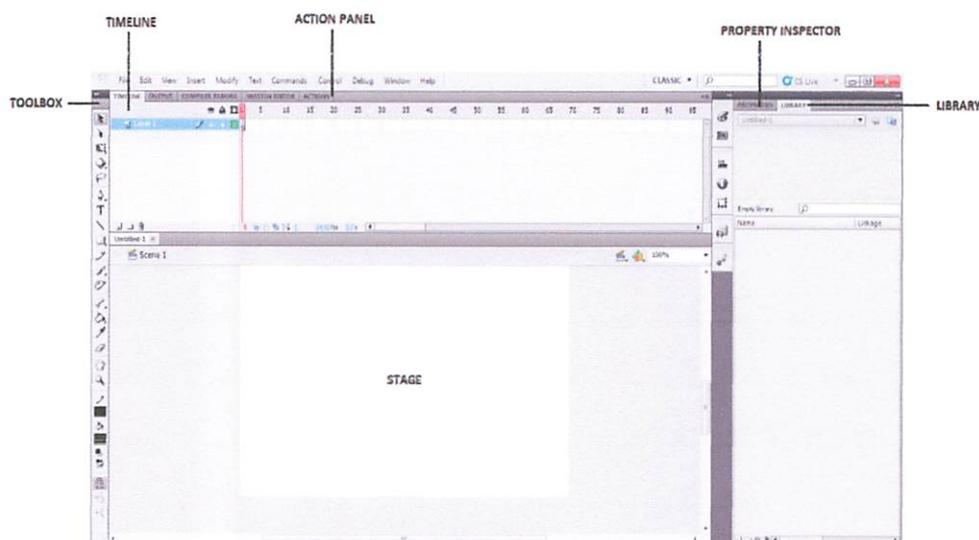
Adobe Flash merupakan *program* animasi 2D berbasis vektor, yang telah banyak digunakan oleh para animator untuk membuat berbagai animasi. Misalnya animasi kartun, web, movie, presentasi, Company profile, e-card, dan bahkan Game-pun bisa dibuat dalam program flash ini. (“Belajar sendiri Adobe Flash Pro CS5”, Penerbit Andi)

2.8.1.1 Tampilan dan Area Kerja Adobe Flash

Di dalam Adobe Flash terdapat beberapa tampilan yang sangat membantu dalam membuat berbagai bentuk objek, yaitu :

1. Toolbox

Toolbox berfungsi untuk menyimpan berbagai peralatan standar yang digunakan dalam membentuk atau memanipulasi bentuk-bentuk yang telah dibuat sebelumnya.



Gambar 2.2 Tampilan Lembar Kerja Adobe Flash Pro CS5

2. Stage

Stage dalam Flash berguna sebagai lembar kerja untuk pembuatan berbagai bentuk objek yang diinginkan.

3. Timeline

Timeline dalam Flash berguna untuk mengatur letak dan waktu dari animasi sebuah objek. Dalam *Timeline* ini terdapat dua bagian utama yaitu *Layer* untuk meletakkan berbagai objek dan *Frame* untuk mengatur waktu dari objek atau animasi yang akan ditampilkan.

4. Actions Panel

Tempat untuk menuliskan berbagai macam *Action Script Code* untuk sebuah objek atau *frame*.

5. Property Inspector

Berisi *property* dari objek yang telah dipilih sebelumnya. Setiap objek memiliki *property* yang berbeda dari objek lainnya.

6. Library

Library dalam Flash berguna untuk menyimpan berbagai objek maupun *symbol* yang digunakan dalam aplikasi yang akan dibuat. Sebuah objek dibuat ataupun diimport ke dalam Flash maka secara otomatis akan ditempatkan ke dalam *library*.

2.8.1.2 Metode Animasi dalam Flash

Pada dasarnya Adobe Flash membagi animasi dalam 2 metode, yaitu :

1. Frame by Frame Animation

Pembuatan animasi dengan cara melakukan perubahan objek pada setiap *frame* secara manual, sehingga dihasilkan perubahan gambar yang teratur. Metode ini biasanya digunakan pada animasi dengan perubahan bentuk objek secara terus-menerus. Misalnya, film kartun.

2. Tweened Animation

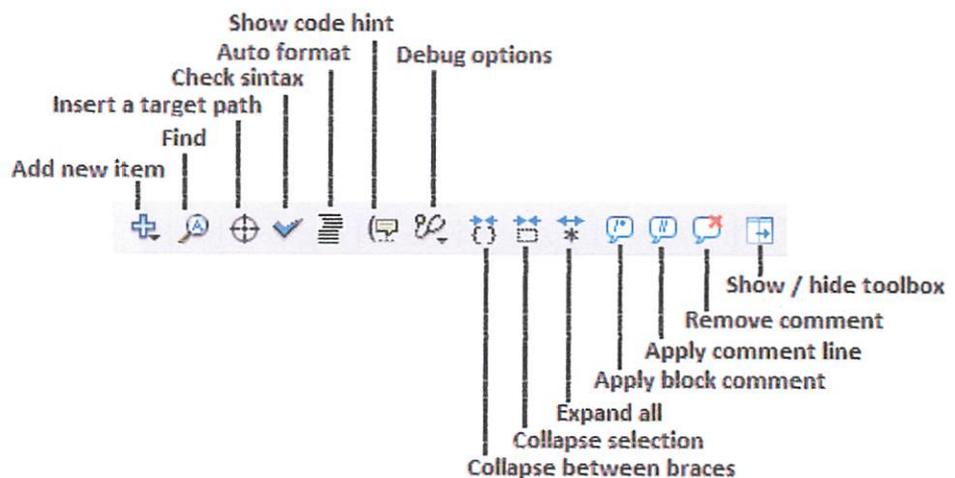
Pembuatan animasi dengan cara menentukan dua poin keadaan pada objek awal dan akhir, sedangkan Adobe Flash membuat rangkaian gerakan diantaranya. Animasi yang dihasilkan menggunakan metode ini adalah gerakan yang halus, perubahan letak, ukuran, rotasi, bentuk maupun warna.

2.8.1.3 Action Script

Action Script merupakan bahasa pemrograman di dalam Adobe Flash yang mengikuti pemrograman *java*. Fungsi *action script* bermacam-macam tergantung dengan tingkat kebutuhan *user* dalam mengembangkan sebuah produk atau aplikasi berbasis *flash*. Dengan *action script*, memungkinkan untuk membuat *movie* yang interaktif sehingga *user* bisa berinteraksi dengan *movie* tersebut, seperti mengontrol navigasi, memasukkan informasi, menggerakkan objek dengan *keyboard*, mengubah bentuk *mouse*, membuat menu, *games*, kuis dan lain-lain.

1. *Actions category*, menunjukkan *action* diberikan pada *frame* atau pada objek (*button* atau *Movie Clip*).
2. *Actions toolbox*, berisi kumpulan *Action Script* yang disediakan oleh Flash dan dapat dipakai langsung dengan melakukan klik ganda pada *action* yang diinginkan.
3. *Script navigator*, menampilkan lokasi di file FLA yang dihubungkan ke *Action Script*.
4. *Script panel*, merupakan tempat dituliskannya / ditematkannya kode-kode dari *Action Script*.
5. *Popup-menu*, berisi kumpulan perintah untuk mengatur *Actions Panel*.

Pada bagian atas *script panel* terdapat beberapa tombol yang berguna untuk membantu penulisan *script* / kode di area *script panel*.



Gambar 2.4 Tombol-tombol di atas area Script panel

1. *Add item*, untuk memasukkan kode *Action Script* ke *script panel* (selain melalui *Actions toolbox*).
2. *Find*, untuk mencari atau menemukan suatu kode dalam *Script panel*.

3. ***Insert a Target path***, untuk mencari target / sasaran objek, biasanya berupa *Movie Clip*.
4. ***Check Syntax***, untuk melakukan cek / pemeriksaan kesalahan pada sintaks kode yang dituliskan di *Script panel*.
5. ***Auto Format***, untuk merapikan tampilan kode di *Script panel*.
6. ***Show Code Hint***, untuk menampilkan penunjuk kode untuk *action* yang akan digunakan.
7. ***Debug Options***, untuk menandai suatu kode yang salah.
8. ***Collapse Between Braces***, untuk tutup script sebagai pembatas fungsi antar script.
9. ***Collapse Selection***, untuk menentukan script yang akan diberi tanda tutup.
10. ***Expand All***, memperluas semua script.
11. ***Apply Block Comment***, untuk menerapkan komentar blok.
12. ***Apply Line Comment***, untuk menerapkan komentar baris.
13. ***Remove Comment***, untuk menghaspus komentar.
14. ***Show / Hide Toolbox***, untuk menampilkan dan menyembunyikan toolbox.

2.8.1.3.2 Struktur Bahasa Action Script

Struktur bahasa yang digunakan oleh Flash adalah *dot syntax*. Contoh :

```
_root.player._x = 50 ;
```

Skrip tersebut berarti menetapkan nilai sumbu x *Movie Clip* bernama *player* sebesar 50.

2.8.1.3.3 Tanda Baca dalam Action Script

Tanda baca dalam action script meliputi :

1. **Curly Braces**, bisa disebut kurung kurawal ({ }) digunakan untuk mengelompokkan bahasa.
2. **Semicolon**, disebut juga titik koma (;), berguna untuk mengakhiri kalimat.

Contoh : on (release) {
 _root.player.gotoAndStop (1);
 }

3. **Comments**, digunakan untuk menuliskan komentar-komentar program, sebagai pengingat maupun penanda.

Untuk menuliskan komentar, ditandai dengan //.Contoh :

```
On (release) {
    // decrement nilai x
    _root.player._x -= 50 ;
}
```

2.8.1.3.4 Variabel dan Tipe Data

Ada empat macam tipe data, yaitu :

1. **Number**, untuk angka tertentu.
 Contoh : nilai = 100.
2. **String**, kumpulan berbagai angka, huruf maupun *symbol*.
 Contoh : Nama="rendi"
3. **Boolean**, nilai yang berisi *true* atau *false*.
 Contoh : melompat = false.
4. **Object**, kumpulan dari *properties*, atribut yang mendeskripsikan objek.
 Contoh :
 var dataKU:Object = new Object ();
 dataKU.nama = " Putra " ;
 dataKU.nomor = 85;

2.8.1.3.5 Operator

Operator terbagi menjadi 3, yaitu :

1. *Assignment operator* (=), digunakan untuk menetapkan nilai suatu variable.
2. *Arithmetic operator*, berisi simbol-simbol matematika.

Simbol	Keterangan
+	Penambah bilangan
-	Pengurang bilangan
*	Perkalian
/	Pembagi
++	Menambah dengan nilai 1
--	Mengurangi dengan nilai 1
+=	Menambah suatu nilai pada variabel
-=	Mengurangi suatu nilai pada variabel
*=	Mengalikan suatu nilai pada variabel
/=	Membagi suatu nilai pada variabel

Tabel 2.1 Tabel Aritmetic Operator

3. *Comparison operator*, digunakan untuk membandingkan dua ekspresi. Memiliki nilai *true* or *false*.

Simbol	Keterangan
==	Sama dengan
<	Kurang dari
>	Lebih dari
<=	Kurang atau sama dengan dari
>=	Lebih atau sama dengan dari
!=	Tidak sama dengan

Tabel 2.2 Tabel Comparison Operator

2.8.1.3.6 Function

Function adalah kumpulan dari baris kode yang dapat digunakan berkali-kali dengan teknik pemanggilan sederhana.

Contoh :

```
function cetak () {  
    trace ( " pesan dari penulis " );  
}  
cetak ();
```

BAB III PERANCANGAN GAME

Di dalam merancang *game* perlu disusun terlebih dahulu sebuah bagan alur yang dapat menjadi acuan dalam mengerjakan setiap tahapan perancangan *game*. Bagan yang disusun menggambarkan tahapan-tahapan apa saja yang harus dikerjakan terlebih dahulu dari awal hingga akhir perancangan yang dapat dijelaskan menjadi beberapa sub bab. Adapun bagan alur perancangan *game* dalam bab ini dapat dirangkum sebagai berikut :



Gambar 3.1 Alur Perancangan Game

3.1 Konsep Game

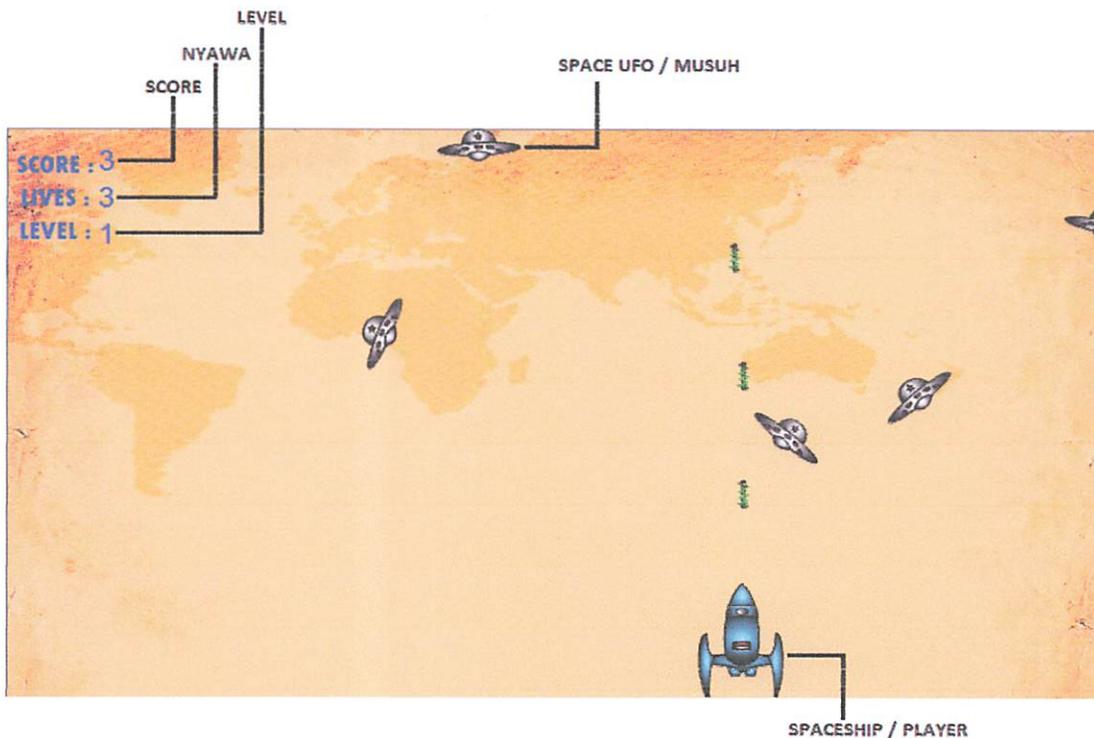
Langkah awal yang dilakukan dalam membuat *game* adalah membuat konsep atau *storyboard* yang dirumuskan dalam bentuk tabel. *Storyboard* yang disusun merupakan gambaran umum aplikasi *game* yang akan dihasilkan melalui tahapan perancangan yang meliputi judul dan jenis *game*, jenis *rating*, sistem kendali yang digunakan, penentuan karakter utama dan musuh, penentuan *background* dan gambaran sistem permainan.

Judul Game	Spaceship Impact
Jenis Game	<i>Arcade Casual</i>
Jenis Rating	Ditujukan untuk usia lebih dari 10 tahun
Sistem Kendali	Hanya dengan menggerakkan ponsel dengan fasilitas Accelerometer sensor yang sudah tersedia khususnya pada ponsel dengan OS Android.
Karakter Utama	Spaceship
Musuh	<i>Space ufo .</i>
Background	<i>Static view</i> dengan ukuran stage 854 x 480.
Sistem Permainan	<ul style="list-style-type: none"> - Pemain memiliki 3 nyawa, jika nyawa habis, permainan berakhir, pemain harus mengulang dari <i>level 1</i> atau <i>level</i> awal. - <i>Game</i> ini memiliki level tidak terbatas, dengan ketentuan minimal 10 skor untuk naik ke level berikutnya. - Hanya satu <i>nama user</i> saja dengan skor tertinggi yang akan ditampilkan pada layar skor.

Tabel 3.1 Tabel Storyboard Game

3.2 Visualisasi Konsep Game

Visualisasi konsep *game* dimaksudkan sebagai gambaran awal dalam membuat tampilan *game* seperti komposisi background, letak pemain dan musuh, tampilan meteran nyawa pemain dan sebagainya sebelum memasuki tahapan *scripting* atau memasukkan *action script*.



Gambar 3.2 Visualisasi Konsep Game

Setelah memasuki tahap *scripting*, visualisasi konsep bisa berubah atau dikembangkan dengan fitur-fitur tambahan jumlah musuh dan lain sebagainya sesuai dengan kebutuhan *game* nantinya.

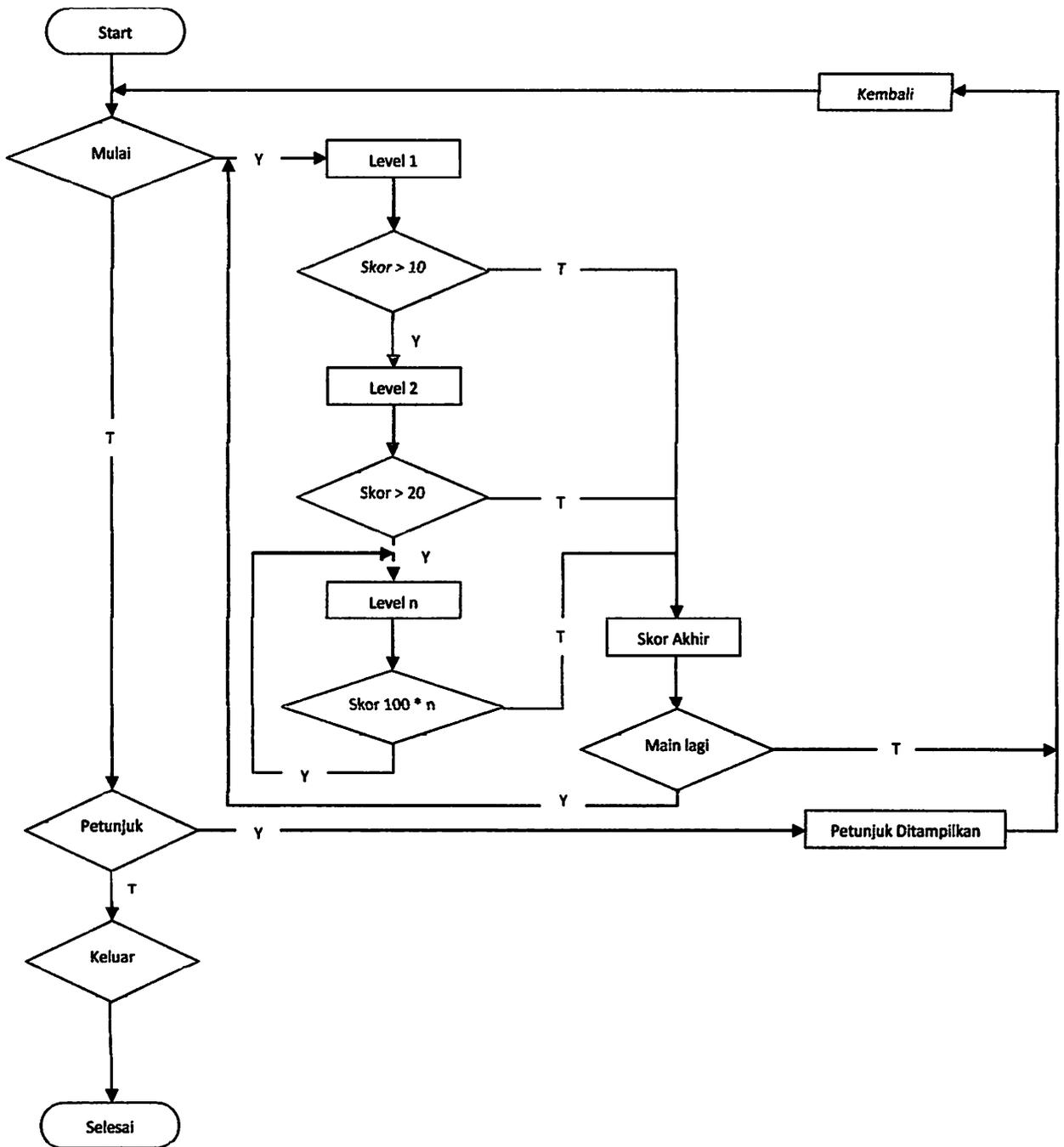
3.3 Alur Game

Saat memulai *game*, yang pertama kali muncul adalah tampilan menu utama yang terdiri dari tiga menu utama, yaitu *start game*, *petunjuk permainan*, dan tombol options pada ponsel untuk *quit game*. Bila menu *start game* dipilih, maka pemain akan diarahkan ke halaman pembuka yang berisi *game* yang langsung bisa dimainkan dengan menggunakan Accelerometer sensor pada ponsel, yaitu *level 1*. Bila pemain sudah menghancurkan *Space UFO* (Musuh) pada *level 1* dan skor = 10 maka pemain akan otomatis naik ke-*level* selanjutnya, demikian seterusnya sampai nyawa *Spaceship* (Player) habis. Sedangkan, bila menu *petunjuk permainan* dipilih, pemain akan diarahkan ke halaman instruksi bermain *game* yang berisi spesifikasi yang diperlukan pada ponsel untuk memainkan *game*, pengenalan fitur-

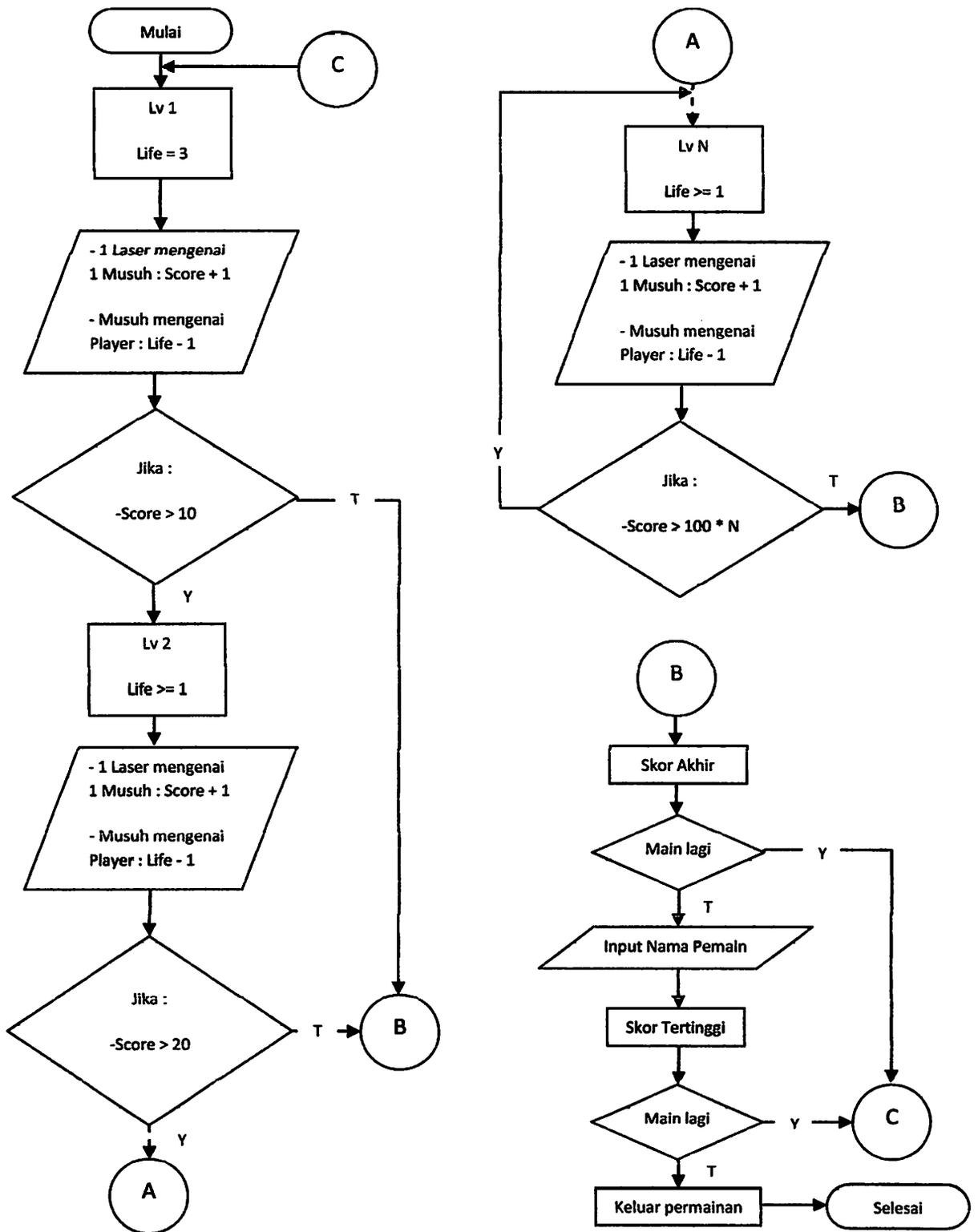
fitur *game*, dan juga keterangan atau ketentuan dalam *game* tersebut. Tombol *options* sebagai *quit game* digunakan untuk keluar dari *game*.

Ketika memulai *level*, ruang gerak *player* dibatasi pada *stage* yang ada yaitu berukuran 854 x 480 dengan *background static view*. Di beberapa *level player* diharuskan mencapai 10 skor disetiap *level* dan secara otomatis menuju *level* berikutnya dengan jumlah *Space UFO* yang jumlahnya bertambah setiap naik ke *level* yang lebih tinggi. Sampai nyawa *Player* habis baru akan ditampilkan *menu skor* yang didalamnya terdapat kolom yang bisa diisi dengan nama inisial pemain untuk dicantumkan pada *menu skor*, hanya 1 nama dengan *skor* tertinggi saja yang ditampilkan didalam *menu skor* dan akan digantikan hanya jika *skor* lebih tinggi.

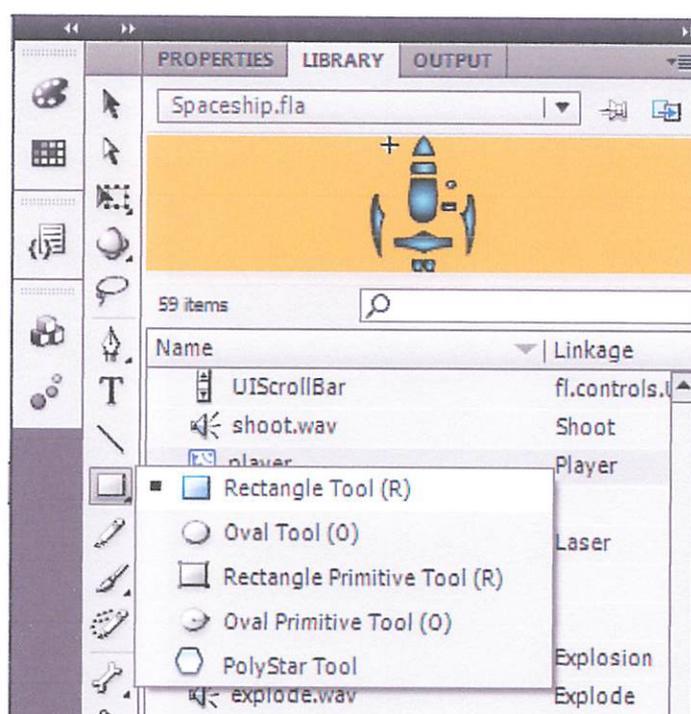
Di dalam *game* pemain hanya bisa menggerakkan ke-kanan atau ke-kiri dengan menggunakan *Accelerometer* sensor dan menghindari dari musuh *Space UFO* sekaligus mengarahkan *laser* yang secara otomatis ditembak-kan dari *Spaceship*.



Gambar 3.3 Alur Game

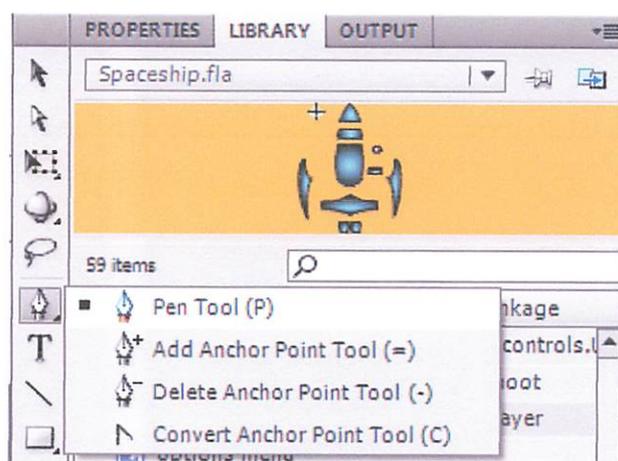


Gambar 3.4 Alur Score dan Level

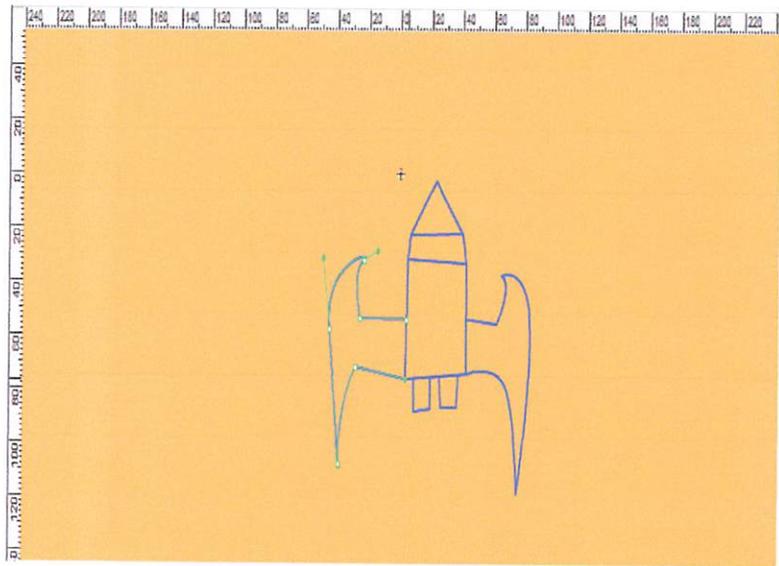


Gambar 3.6 Penggunaan Tools *Rectangle*, *Oval*, dan *PolyStar*

Atau bisa juga menggunakan *Pen Tools* dimana bisa membuat karakter dengan menarik titik 1 dan menghubungkan dengan titik – titik yang lain sampai membentuk sebuah pola yang diinginkan.



Gambar 3.7 Penggunaan *Pen Tools*

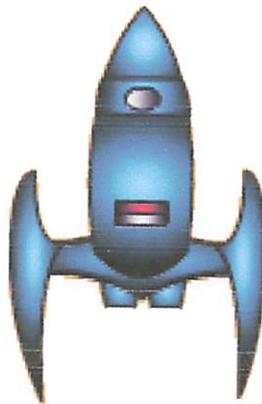


Gambar 3.8 Contoh Penggunaan *Pen Tools*

3.4.2 Hasil Pembuatan Karakter

Karakter di dalam *game Spaceship Impact* terdiri dari dua karakter, yaitu :

1. *Spaceship*, sebagai karakter utama yang bisa menembakkan senjata laser.



2. *Space UFO*, sebagai karakter musuh.



Gambar 3.9 Hasil Desain Karakter 2D

3.5 Pembuatan Background

Pembuatan *background* dilakukan dengan menggunakan *software* Adobe Flash Professional CS5. *Background* yang digunakan berukuran 854 x 480 *pixel*. Hasil desain *background* kemudian disimpan dalam bentuk *Movie Clip* dengan nama *background*.



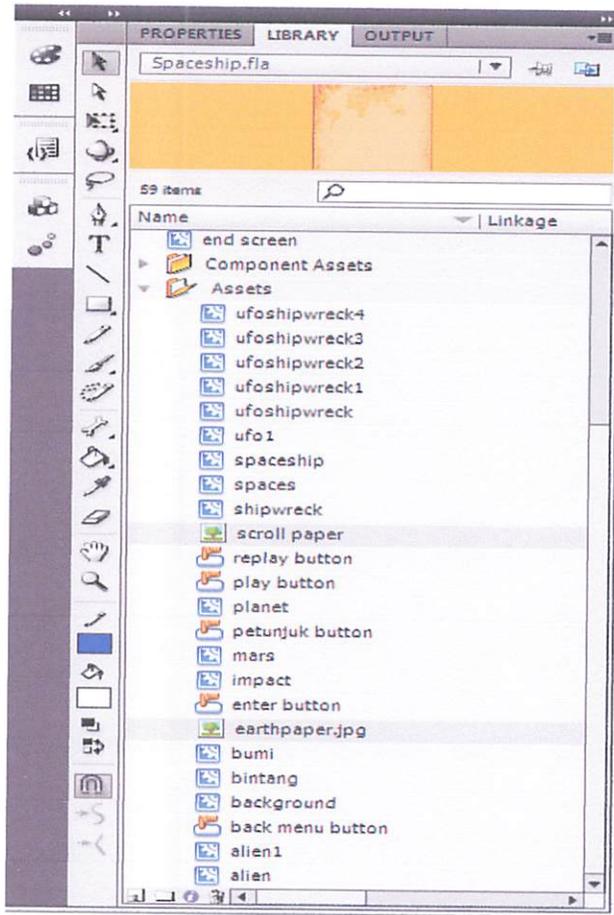
Gambar 3.10 Background Game

3.6 Pembuatan Movie Clip

Pembuatan *Movie Clip* dilakukan setelah semua desain karakter dan *background* yang dibutuhkan selesai dibuat. Secara garis besar *Movie Clip* yang dibutuhkan antara lain meliputi :

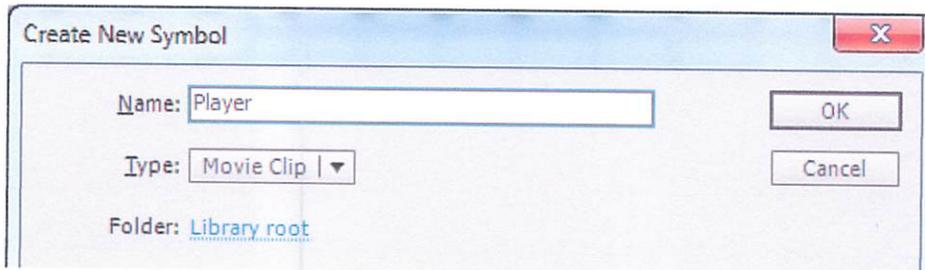
1. *Movie Clip Player (Spaceship)* sebagai karakter utama.
2. *Movie Clip Enemy (Space UFO)* sebagai musuh atau lawan dalam permainan.
3. *Movie Clip background*, terdiri dari gambar latar, tampilan *score*, *lives*, dan *level*.
4. *Movie Clip Intro screen* sebagai tampilan menu awal saat aplikasi dijalankan, yang didalamnya terdapat tombol *petunjuk permainan* atau *button help_btn* untuk menampilkan menu *petunjuk* atau *Movie Clip helpMenu*, dan tombol *Start Game* atau *button play_btn* untuk memulai permainan.
5. *Movie Clip end screen* sebagai tampilan menu skor setelah permainan berakhir, didalamnya terdapat kolom teks *highscorename* yang digunakan untuk input nama pemain dan untuk memasukkan teks ke dalam menu menggunakan tombol *enter button*, tombol *coba lagi* atau *playagain button* untuk memulai lagi permainan dari awal (*level 1*) dan tombol *kembali* atau *back button* untuk kembali ke menu awal (*Intro screen*).

Pembuatan *Movie Clip* bisa dibentuk langsung dengan menggunakan Toolbox atau dengan menggabungkan gambar dengan cara mengimport semua gambar yang dibutuhkan dalam bentuk *png* ataupun *jpeg* ke dalam *library* Adobe Flash Professional CS5.



Gambar 3.11 Mengimport File ke Library Adobe Flash

Movie Clip yang dibuat pertama kali adalah *Movie Clip* karakter utama. Pembuatannya dimulai dengan menekan **Ctrl+F8** pada *keyboard* untuk eksekusi menu *insert > new symbol* dan diberi nama *Movie Clip* pertama dengan nama *player*.



Gambar 3.12 Membuat Movie Clip Karakter Utama

3.7 Logika Dasar

Sebelum memasuki tahapan *scripting* perlu dirumuskan logika dasar untuk menggerakkan karakter di dalam *game*. Logika yang perlu dirumuskan yaitu logika pergerakan karakter, logika tumbukan karakter dengan objek.

Pergerakan *Movie Clip* karakter menggunakan fungsi *accelMove*. Agar *Movie Clip* karakter bergerak menggunakan sensor *Accelerometer* pada ponsel, yang terjadi sebenarnya adalah adanya perubahan logika *properties* (*x*) *Movie Clip* karakter dengan nilai tertentu yang diinginkan. Contoh penulisan *script*-nya:

```

if (player.x < 0)
{
    player.x = 0;
}
else if (player.x > (stage.stageWidth - player.width) )
{
    player.x = stage.stageWidth - player.width;
}

```

Script di atas berarti, jika posisi awal *player* *x* lebih pendek dari posisi 0 maka posisi *player* akan berada posisi 0 pada saat game dimulai, hanya jika posisi *player* *x* lebih lebar dari lebar *stage* maka lebar posisi *player* sama dengan lebar *stage*.

Sementara untuk deteksi tumbukan antara *Movie Clip* karakter dengan objek lain menggunakan fungsi *hitTestObject* dimana *Movie Clip Enemy* mengenai *Movie Clip Player*, dan *Movie Clip Laser* mengenai *Movie Clip Enemy*. Contoh penulisan *script* :

```

/** tumbukan antara Movie Clip Enemy dengan Player
If (tempEnemy.y > stage.stageHeight || tempEnemy.hitTestObject(player))
{
    makeExplosion(tempEnemy.x, tempEnemy.y);
    removeEnemy(i);
    lives--;
    lives_txt.text = String(lives);
}

```

```

    }
    /** tumbukan antara Movie Clip Laser dengan Enemy
        if (tempEnemy.hitTestObject(tempLaser))
        {
            makeExplosion(tempEnemy.x, tempEnemy.y);
            removeEnemy(i);
            removeLaser(j);
            score++;
            score_txt.text = String(score);
            break enemy;
        }

```

3.8 Penerapan Action Script

Setelah merumuskan logika dasar yang akan dipakai di dalam pembuatan *game* dan semua *Movie Clip* telah di masukkan dan diatur di dalam *stage*, selanjutnya tahapan pembuatan *game* memasuki tahapan penerapan *action script*. Di dalam tahapan ini penulisan *action script* bisa diletakkan di dalam *Movie Clip* atau di *frame-frame* yang dikehendaki sesuai dengan kebutuhan *game* yang akan dibuat.

Berikut adalah beberapa contoh penerapan *action script* di dalam pembuatan *game* “*Spaceship Impact*” :

3.8.1 Action Script Persiapan Permainan

```

function initGame():void
{
    Level = 1;//level mulai
    level_txt.text = String(level);
    score = 0;//skor awal
    score_txt.text = String(score);
    lives = 3;//set nyawa
    lives_txt.text = String(lives);
    player = new Player();
    enemies = new Array();
    Lasers = new Array();
    explosions = new Array();
    gameState = STATE_START_PLAYER;
    trace(gameState);
}

```

3.8.2 Action Script Persiapan Karakter Utama (Spaceship)

```

function startPlayer():void
{
    player.y = stage.stageHeight - player.height;//set posisi Player
    addChild(player);
    laserTimer.start();
    accel = new Accelerometer();//set accelerometer
    if (Accelerometer.isSupported)
    {
        accel.addEventListener(AccelerometerEvent.UPDATE, accelMove);
    }
    else
    {
        addEventListener(Event.ENTER_FRAME, movePlayer);
    }
}

```

```

gameState = STATE_PLAY_GAME;
trace(gameState);
}

```

3.8.3 Action Script Menggerakkan Karakter Utama

```

function accelMove(event:AccelerometerEvent):void
{
player.x -= event.accelerationX * 240;//set sensivity sensor
if (player.x < 0)
{
    player.x = 0;
}
else if (player.x > (stage.stageWidth - player.width) )
{
    player.x = stage.stageWidth - player.width;
}
}

```

3.8.4 Action Script Persiapan Tembakan (Laser)

```

function timerListener(e:TimerEvent):void
{
var tempLaser:Movie Clip = new Laser();
tempLaser.x = player.x +(player.width/2); //set posisi Laser
tempLaser.y = player.y; //set posisi Laser pada Player
tempLaser.speed = 10;//set kecepatan Laser
Lasers.push(tempLaser);
addChild(tempLaser);
var music:Sound = new Shoot();
var sc:SoundChannel = music.play();
}

```

3.8.5 Action Script Gerakan Laser

```

function moveLaser():void
{
    var tempLaser:Movie Clip;
    for (var i=Lasers.length-1; i>=0; i--)
    {
        tempLaser = Lasers[i];
        tempLaser.y -= tempLaser.speed;
        if (tempLaser.y < 0)
        {
            removeLaser(i);
        }
    }
    var tempExplosion:Movie Clip;
    for (i=explosions.length-1; i>=0; i--)
    {
        tempExplosion = explosions[i];
        if (tempExplosion.currentFrame >= tempExplosion.totalFrames)
        {
            removeExplosion(i);
        }
    }
}

```

3.8.6 Action Script Persiapan Musuh (Space UFO)

```

function makeEnemies():void
{
    //set kecepatan Enemy muncul
    var chance:Number = Math.floor(Math.random() * 100);
    if (chance <= 1 + level)
    {
        var tempEnemy:Movie Clip;

```

```

    tempEnemy = new Enemy();
    tempEnemy.speed = 2;//set kecepatan gerak
    //set jumlah Enemy yang dikeluarkan
    tempEnemy.x = Math.round(Math.random() * 1000);
    trace("tempEnemy");
    addChild(tempEnemy);
    enemies.push(tempEnemy);
}
}

```

3.8.7 Action Script Gerakan Musuh

```

function moveEnemies():void
{
    var tempEnemy:Movie Clip;
    for (var i:int =enemies.length-1; i>=0; i--)
    {
        tempEnemy = enemies[i];
        //set gerak rotasi Enemy
        tempEnemy.rotation += (Math.round(Math.random()*10-5));
        tempEnemy.x -=
        (Math.sin((Math.PI/180)*tempEnemy.rotation))*tempEnemy.speed;
        tempEnemy.y +=
        (Math.cos((Math.PI/180)*tempEnemy.rotation))*tempEnemy.speed;

        if (tempEnemy.x < 0)
        {
            tempEnemy.x = 0;
        }
        if (tempEnemy.x > (stage.stageWidth – tempEnemy.width))
        {
            tempEnemy.x = stage.stageWidth – tempEnemy.width;
        }
    }
}

```

```

//set tumbukan Enemy dengan Player
if (tempEnemy.y > stage.stageHeight || tempEnemy.hitTestObject(player))
{
    makeExplosion(tempEnemy.x, tempEnemy.y);
    removeEnemy(i);
    lives--;
    lives_txt.text = String(lives);
}
}
}

```

3.8.8 Action Script Tumbukan Laser Dengan Enemy

```

function testCollisions():void
{
    var tempEnemy:Movie Clip;
    var tempLaser:Movie Clip;

    enemy:for (var i:int=enemies.length-1; i>=0; i--)
    {
        tempEnemy = enemies[i];
        for (var j:int=Lasers.length-1; j>=0; j--)
        {
            tempLaser = Lasers[j];
            //set tumbukan Enemy dengan Laser
            if (tempEnemy.hitTestObject(tempLaser))
            {
                //set ledakan saat terjadi tumbukan Laser dengan Enemy
                makeExplosion(tempEnemy.x, tempEnemy.y);
                removeEnemy(i);
                removeLaser(j);
                score++; //tambah skor
                score_txt.text = String(score);
            }
        }
    }
}

```

```

        break enemy;
    }
}
}
}

```

3.8.9 Action Script Membuat Ledakan

```

function makeExplosion(ex:Number, ey:Number):void
{
    var tempExplosion:Movie Clip;
    tempExplosion = new Explosion();
    tempExplosion.x = ex;
    tempExplosion.y = ey;
    addChild(tempExplosion);
    explosions.push(tempExplosion);
    var music:Sound = new Explode();
    var sc:SoundChannel = music.play();
}

```

3.8.10 Action Script Menu Skor (End Screen)

```

function endGame():void
{
    removeGame();
    endScreen.visible = true;
    endScreen.x = 0;
    removeEventListener(Event.ENTER_FRAME, gameLoop);
    showResults();
}

//set untuk tombol kembali ke menu utama ( intro screen )
endScreen.backmenu_btn.addEventListener(MouseEvent.CLICK, backtoMenu);

function backtoMenu(event:MouseEvent):void
{

```

```

    moveScreenOn(introScreen);
}

```

3.8.11 Action Script Menampilkan Hasil Akhir Permainan

```

function showResults():void
{
    trace("Show Results");
    endScreen.enter_btn.visible = false;
    endScreen.highScoreName.visible = false;
    //merekam semua skor dan menampilkan yang paling tertinggi
    var so:SharedObject = SharedObject.getLocal("alltimeHighScore");
    if (so.data.score == undefined || score > so.data.score)
    {
        //set untuk kolom nama pemain
        endScreen.highScore.text = "Kamu mencapai Level " + level + " dengan
Score " + score + "! \nMasukkan nama kamu pada kolom dibawah :";
        endScreen.enter_btn.visible = true;
        endScreen.highScoreName.visible = true;
    }
    else
    {
        //hanya satu skor tertinggi yang ditampilkan
        endScreen.highScore.text = "Score kamu adalah " + score + " ! \nBelum
mengalahkan Score " + so.data.score + " dari " + so.data.name;
    }
    endScreen.enter_btn.addEventListener(MouseEvent.CLICK, clickEnter);
    function clickEnter(event:MouseEvent):void
    {
        trace("clickEnter");
        //jika skor terpecahkan maka akan keluar kolom nama pemain baru
    }
}

```

```

endScreen.highScore.text = "Selamat kepada : " +
endScreen.highScoreName.text + " Kamu mencapai Level " + level + "
dengan Score " + score;
    so.data.score = score;
    so.data.level = level;
    so.data.name = endScreen.highScoreName.text;
    so.flush();
    endScreen.enter_btn.visible = false;
    endScreen.highScoreName.visible = false;
}
endScreen.play_btn.addEventListener(MouseEvent.CLICK,
clickFinalAway);
function clickFinalAway(event:MouseEvent):void
{
    trace("clickFinalAway");
    moveScreenOff(endScreen);
}
}

```

3.8.12 Action Script Menampilkan Option Menu

```

function optionsKey(event:KeyboardEvent):void
{
    //set kode ASCII pada ponsel dan pada keyboard komputer
    if (event.keyCode == 16777234 || event.keyCode == 13)
    {
        if (hiddenOptions)
        {
            setChildIndex(optionsMenu,numChildren-1);
            optionsMenu.visible = true;
            optionsMenu.addEventListener(MouseEvent.CLICK, exitApp);
            pauseGame();
        }
    }
}

```

```

        else
        {
            optionsMenu.visible = false;
optionsMenu.removeListener(MouseEvent.CLICK, exitApp);
            resumeGame();
        }
        hiddenOptions = ! hiddenOptions;
    }
}

```

3.8.13 Action Script Keluar Aplikasi Pada Ponsel

```

function exitApp(event:MouseEvent)
{
    trace("exitApp");
NativeApplication.nativeApplication.exit(0);
}
//Layar ponsel tetap menyala
NativeApplication.nativeApplication.systemIdleMode =
SystemIdleMode.KEEP_AWAKE;

```

3.9 Finishing Game

Setelah semua *action script* diterapkan dan dilakukan uji coba, maka selanjutnya dilakukan tahap penyempurnaan yang meliputi memasukkan suara pada *game*.

Penambahan suara dilakukan dengan memanggil file *sound* yang diinginkan yang sudah diimport ke library dengan menggunakan metode:

```

var music:Sound = new "nama sound yang ada pada Library";
var sc:SoundChannel = music.play();

```

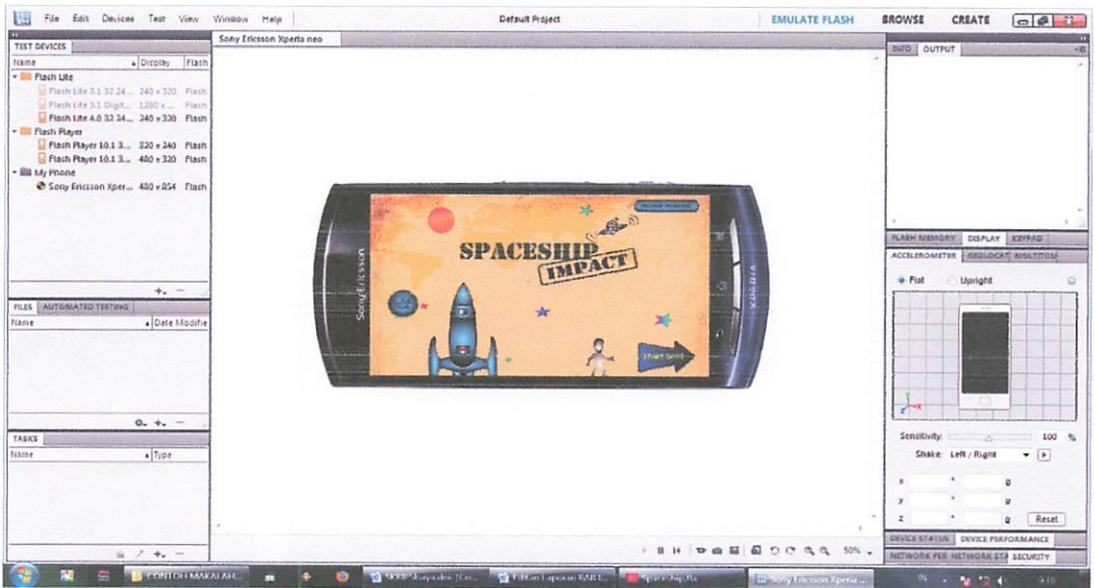
Pada tahapan paling akhir kemudian dilakukan *publishing game* dari *Adobe Flash Pro CS5* menjadi format *.swf*, agar *game* dapat dijalankan pada ponsel ber-OS Android maka harus di rubah kedalam format *.apk*.

BAB IV IMPLEMENTASI DAN PENGUJIAN GAME

4.1 Implementasi Game "Spaceship Impact"

Pembuatan game flash Spaceship Impact ini dilakukan dengan menerapkan hasil desain yang telah dibuat kedalam bahasa pemrograman flash yakni Action Script sehingga prosedur – prosedur yang telah dibuat dapat dimengerti oleh mesin dan menghasilkan keluaran seperti dengan yang diharapkan.

Berikut ini adalah implementasi game "Spaceship Impact" pada software Adobe Device Central CS5:



Gambar 4.1 Tampilan Game Pada Adobe Device Central

Dalam implementasi aplikasi diatas adalah implementasi menggunakan emulator Adobe Device Central CS5 yang sudah terintegrasi dengan software Adobe Flash Profesional CS5, dimana *game* harus dicoba dahulu agar setting *game* pada ponsel aslinya nanti bisa berjalan atau tidak.

4.2 Pengujian Game "Spaceship Impact"

Dalam pembuatan *game* "Spaceship Impact" dilakukan juga tahapan pengujian aplikasi untuk mengetahui fungsi dari setiap menu dan fitur yang digunakan untuk memainkan *game*.

Untuk dapat menjalankan aplikasi *game* ini dengan baik, maka diperlukan perangkat keras dan perangkat lunak yang memadai. Adapun spesifikasi perangkat keras dan perangkat lunak yang diperlukan antara lain :

1. Ponsel dengan kecepatan CPU minimal 1 GHz.
2. Ukuran Layar 854 x 480 pixel
3. O.S. Android minimal versi 2.2 (Froyo).
4. Support Adobe Flash Player.
5. Support Adobe AIR.
6. Support fitur Accelerometer sensor.

Dari ketentuan spesifikasi diatas dapat digunakan sebagai acuan dasar untuk pengujian maka dibutuhkan perangkat keras dan perangkat lunaknya menurut spesifikasi karena pertimbangan sebagai berikut :

1. Dengan penggunaan kecepatan CPU pada ponsel dengan minimal 1 GHz akan sangat dianjurkan, karena CPU pada ponsel dengan kecepatan dibawah atau kurang dari 1 GHz akan mempengaruhi kinerja *game* pada ponsel seperti pada saat *game* dijalankan maka *game* akan menjadi lambat karena kecepatan CPU ponsel yang kurang memadai.
2. Pada ukuran layar diatas ditentukan 854 x 480 pixel karena pada ukuran layar yang lebih besar tampilan *game* tetap pada ukuran saat *game* dibuat, maka tampilan *game* akan lebih kecil dari lebar pixel pada layar jika diuji pada ponsel dengan layar yang lebih besar.
3. Sistem operasi yang digunakan minimal adalah versi 2.2 (Froyo) seperti penjelasan pada BAB II, Sub-BAB 2.3.5 dijelaskan bahwa dari versi – versi Android sebelumnya belum mendukung Adobe Flash dan Adobe AIR sebagai perangkat lunak pendukung pada pengujian *game* "Spaceship Impact" ini.

4.3 Tampilan Aplikasi Game "Spaceship Impact"

Dalam pengujian aplikasi game "Spaceship Impact" dapat dilihat beberapa tampilan game yang digunakan dalam pembuatan game.

4.3.1 Icon Game Pada Ponsel

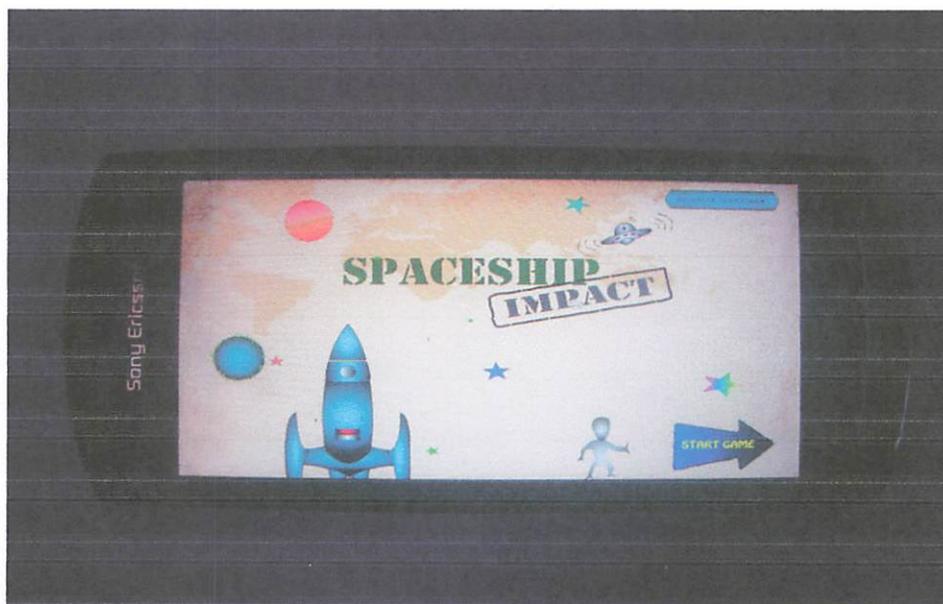
Icon yang dilingkari dengan warna merah adalah contoh tampilan icon game "Spaceship Impact" setelah diinstal pada salah satu ponsel Android.



Gambar 4.2 Tampilan Icon Game

4.3.2 Menu Utama

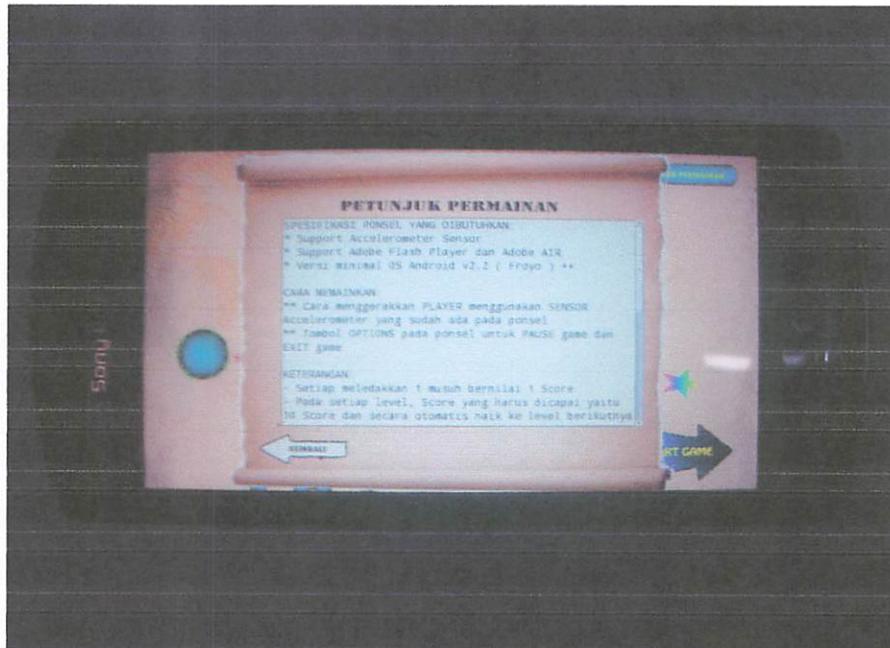
Di dalam menu utama terdapat tiga menu yaitu menu *Start Game* untuk memulai *game*, menu *petunjuk permainan* untuk melihat petunjuk memainkan *game*.



Gambar 4.3 Tampilan Menu Utama

4.3.3 Halaman Petunjuk Permainan

Halaman petunjuk permainan terdapat petunjuk tentang *spesifikasi* untuk ponsel, Cara memainkan *game*, dan sekaligus keterangan dan ketentuan didalam *game* didalam menu ini juga terdapat tombol *kembali* untuk menutup halaman petunjuk permainan dan kembali ke menu utama.



Gambar 4.4 Tampilan Petunjuk Permainan

Pada tampilan Petunjuk Permainan didalamnya terdapat Spesifikasi dasar perangkat keras dan perangkat lunak yang dibutuhkan untuk pengujian *game*-nya, bagaimana cara memainkan *game*-nya dan keterangan yang didalamnya terdapat ketentuan permainan seperti perolehan skor, naik level, & jumlah nyawa yang sudah ditentukan.

4.3.4 Eksekusi Menu Exit Game

Jika ditekan tombol *Options* pada ponsel maka terdapat menu *exit* untuk keluar dari *game*.

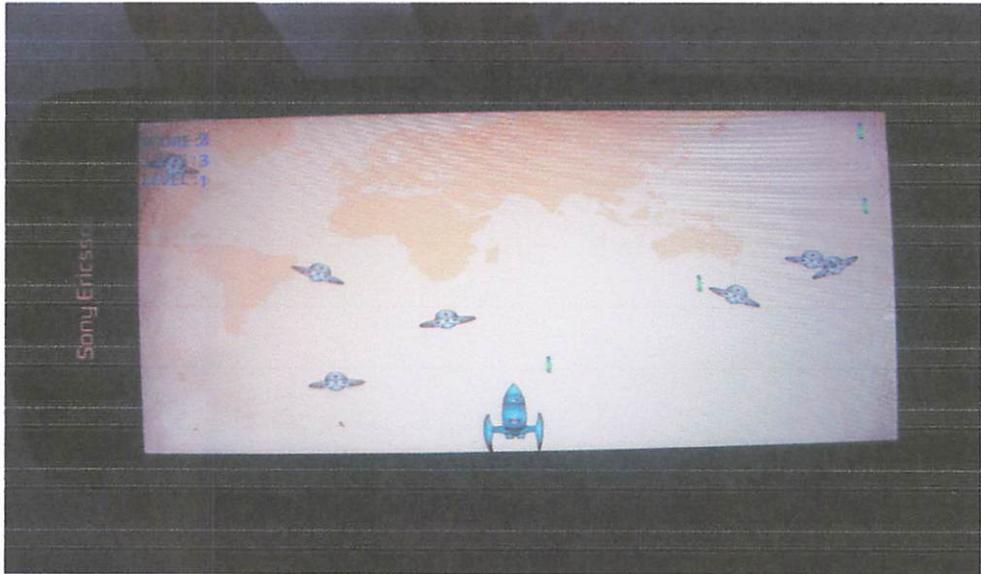


Gambar 4.5 Tampilan Eksekusi Menu Exit

Pada tampilan Menu Exit kali ini berguna jika pada saat permainan, user mendapat panggilan telepon atau notifikasi lain seperti sms, baterai habis maka secara otomatis Menu Exit akan keluar yang sekaligus berfungsi sebagai *Pause game*.

4.3.5 Eksekusi Permainan

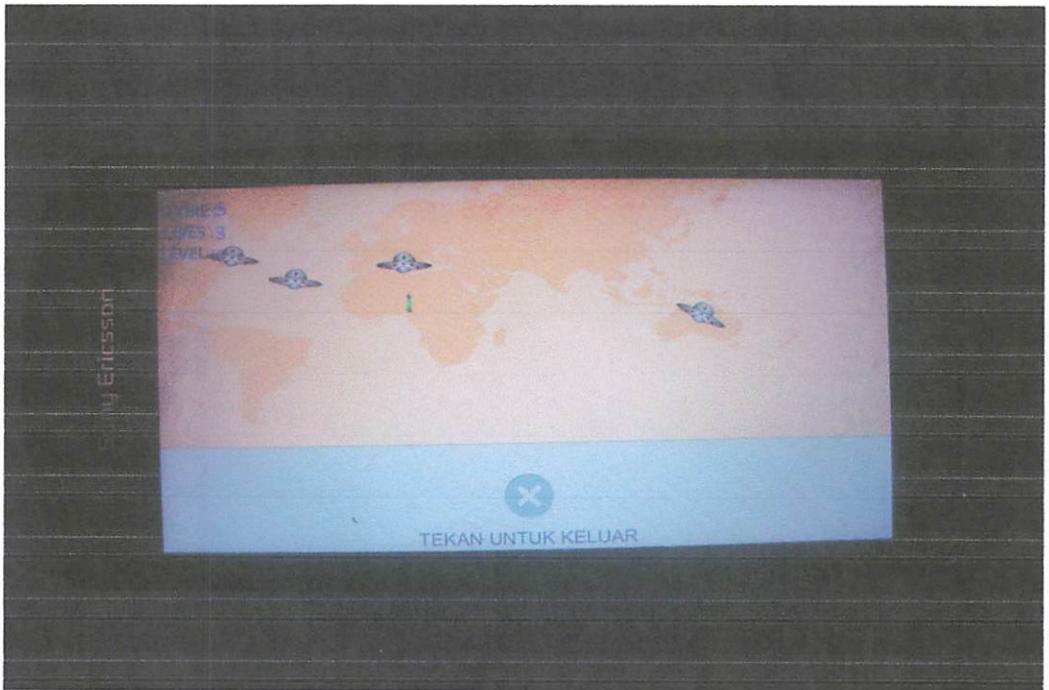
Berikut adalah beberapa contoh tampilan *level game Spacehip Impact*. Pada setiap *level* akan tampil meteran *Lives*, *Level*, dan Skor *player* pada pojok kiri atas.



Gambar 4.6 Tampilan Eksekusi Permainan

Cara memainkan *game* ini sangat mudah hanya menggunakan *sensor Accelerometer* yang sudah tersedia pada ponsel dengan memiringkan ponsel kekanan atau kekiri pada posisi *landscape* seperti pada gambar diatas.

Menu *pause* muncul ketika *user* menekan tombol *Options* pada keypad ponsel ketika *game* sedang berjalan dan *player* berada di *level* aktif. Jika tombol *Options* ditekan kembali maka *game* akan *resume* atau kembali berjalan, dan jika saat menu *pause* masih ada dan ditekan maka akan terjadi proses *exit game* atau keluar dari *game*.



Gambar 4.7 Tampilan Menu Pause

Pada eksekusi permainan kali ini dijelaskan sistem penambahan skor yaitu seperti pada *script* :

```

if (tempEnemy.hitTestObject(tempLaser))
{
//set ledakan saat terjadi tumbukan Laser dengan Enemy
makeExplosion(tempEnemy.x, tempEnemy.y);
removeEnemy(i);
removeLaser(j);
score++; //tambah skor
score_txt.text = String(score);
break enemy;

```

Bahwa jika tembakan (*Laser*) mengenai musuh (*Enemy*) if (*tempEnemy.hitTestObject(tempLaser)*) maka skor akan bertambah *score++*, begitu juga untuk naik *Level* yaitu :

```
if (score > level * 10)
{
    level++;
    level_txt.text = String(level);
}
```

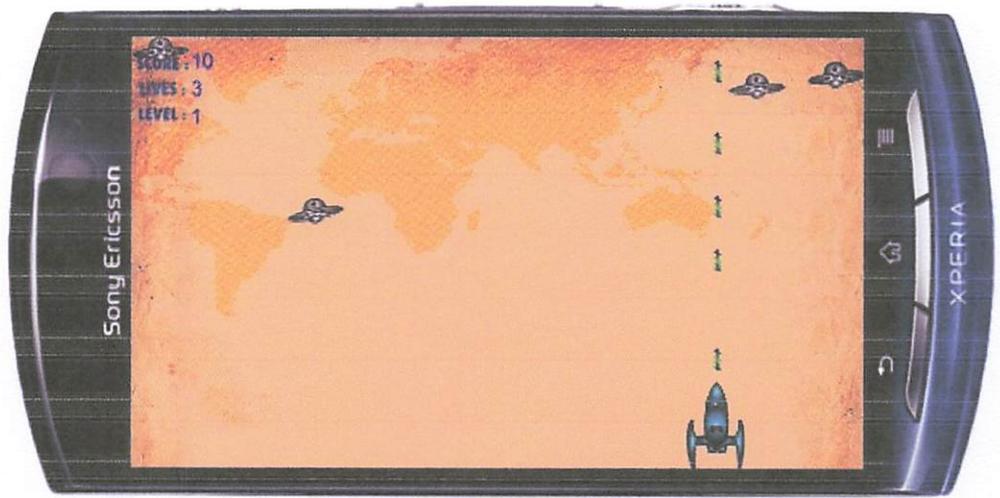
Pada tahap ini dimana setiap skor kelipatan 10 maka *Level* akan bertambah 1 secara otomatis.

Berikut ini adalah pengujian sistem penilaian (*Score*) dan naik *Level* pada *game* menggunakan emulator Adobe Device Central CS5 :



4.8 Tampilan *Level* awal

Dimana *Score* akan bertambah 1 saat *Laser* mengenai musuh (*SpaceUFO*).



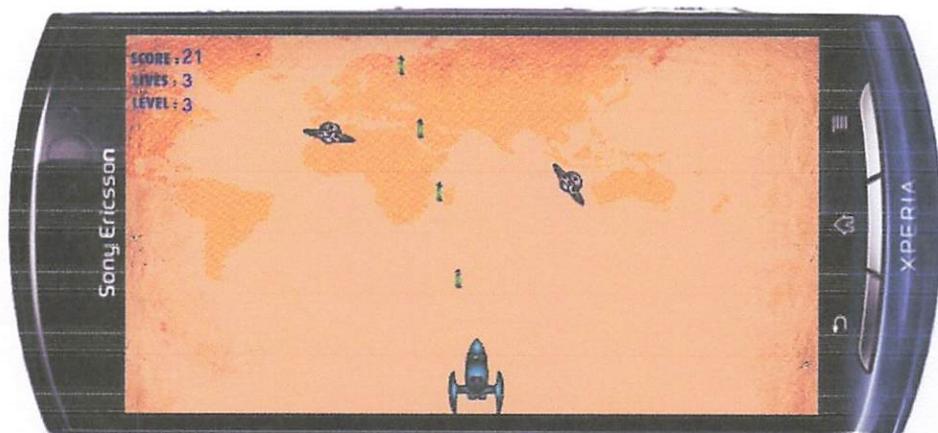
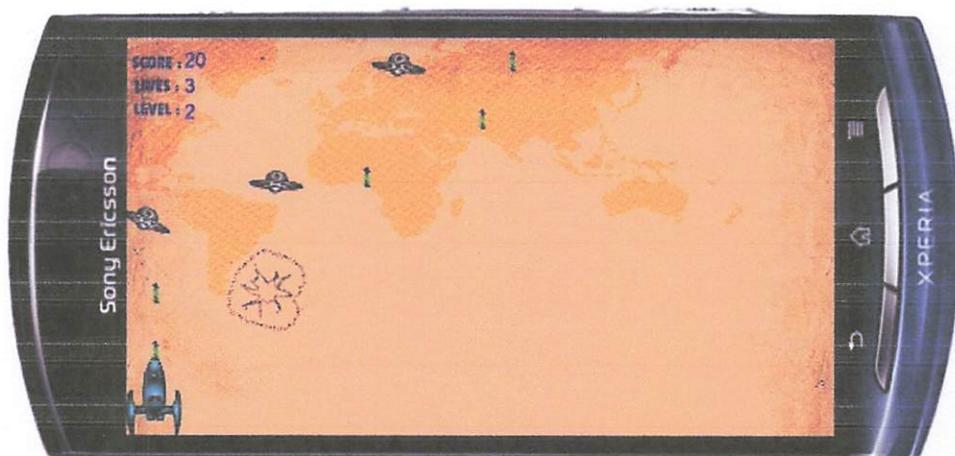
4.9 Tampilan sebelum naik Level

Diatas adalah tampilan pada saat *game* berada pada *level* 1 dengan *score* 10.

Tampilan dibawah ini adalah saat *game* sudah naik ke *level* 2 setelah *Score* mencapai 11 ($Score > 10$).



4.10 Tampilan setelah naik Level



4.11 Tampilan *Level 2* dan setelah naik ke *Level 3*

Dan dibawah ini adalah tampilan hasil akhir setelah nyawa *Player* habis, sekaligus menampilkan *Score* akhir yang diperoleh oleh pemain.



4.12 Tampilan Skor Akhir

Pada tampilan Skor Akhir seperti gambar 4.12 didalamnya terdapat tampilan *Level* dan *Score* yang sudah dicapai dan terdapat kolom pengisian nama untuk pemain, terdapat juga tombol kembali ke-*Menu* awal dan tombol *Main lagi* untuk bermain kembali ke level awal.

4.3.6 Halaman End Screen

Halaman *end screen* atau saat nyawa atau *lives player* habis dan permainan berakhir maka muncul menu *end screen* dimana akan menampilkan skor terakhir dalam permainan, dan terdapat kolom untuk nama pemain, tombol *coba lagi* untuk bermain kembali dan tombol *kembali* untuk kembali ke menu utama (*intro screen*). Pada halaman ini juga berlaku tombol *Options* pada keypad ponsel untuk *exit game*.





Gambar 4.13 Tampilan Halaman End Screen

Pada halaman *End Screen* didalamnya terdapat skor dan *level* terakhir yang sudah dicapai *user* (pemain) yang juga terdapat kolom untuk mengisi nama (inisial) dan dapat disikan nama sesuai dengan keinginan pemain.

4.3.7 Halaman Skor

Pada halaman ini yang ditampilkan hanya satu skor tertinggi dari skor keseluruhan, dan seperti gambar dibawah skor baru yang belum mengalahkan skor tertinggi “ Score kamu adalah 6!, Belum mengalahkan score 13 dari angka “.



Gambar 4.14 Tampilan Halaman Skor Tertinggi

BAB V PENUTUP

5.1 Kesimpulan

1. Desain karakter dua dimensi (2D) yang terdapat di dalam *game* “*Spaceship Impact*” dibuat dengan aplikasi pendukung agar dapat berjalan dengan baik menggunakan aplikasi utama yaitu Adobe Flash Pro CS5 yang berupa software desain animasi dengan teknik *tween animation*.
2. Dalam Pengujian Game “*Spaceship Impact*” terdapat penjelasan tentang spesifikasi ponsel yang digunakan dalam pengujian bahwa spesifikasi tersebut sangat diperlukan agar pada saat pengujian *game* dapat bekerja dengan maksimal.
3. Pada Tampilan Skor Akhir menggunakan emulator Adobe Device Central CS5 menampilkan sementara *Level* yang bisa ditempuh pemain dengan menggunakan Simulator, dan tidak perlu khawatir karena *Level* yang disediakan dalam *game* tidak terbatas maka pemain bisa berkompetisi untuk menggantikan *Score* tertinggi sebelumnya dengan *Score* yang lebih tinggi.

5.2 Saran

Adapun saran-saran yang diberikan untuk pengembangan *game* “*Spaceship Impact*” adalah sebagai berikut :

1. Diharapkan pada pengembangan selanjutnya terdapat beberapa pilihan karakter utama yang bisa digunakan dalam permainan.
2. Game bisa dilengkapi dengan fitur penyimpanan posisi *level* yang telah ditempuh disetiap *level* sehingga pemain tidak harus mengulang dari *level* awal ketika pemain ingin melanjutkan permainan dilain waktu.

DAFTAR PUSTAKA

- [1]. Madcoms, "Kupas tuntas Adobe flash profesional CS5", penerbit Andi publihser, Yogyakarta.
- [2]. Madcoms, "Adobe flash profesional CS5 untuk pemula", Penerbit Andi Publisher, Yogyakarta.
- [3]. MADCOMS , "Belajar Sendiri Adobe Flash Pro CS5", Penerbit Andi Publisher, Yogyakarta.
- [4]. Priyanto Hidayatullah dkk, "Membuat Mobile Game Edukatif Dengan Flash", Penerbit INFORMATIKA, Bandung.
- [5]. Ronald Budi, "Having Fun with Adobe Flash Professional CS5", Penerbit SKRIPTA, Yogyakarta.
- [6]. [Http://rulyardiansyah.wordpress.com/2010/07/15/tutorial-membangun-aplikasi-android-via-adobe-flash-cs5-professional-dengan-teknologi-adobe-air/](http://rulyardiansyah.wordpress.com/2010/07/15/tutorial-membangun-aplikasi-android-via-adobe-flash-cs5-professional-dengan-teknologi-adobe-air/), diakses (22/11/2011).
- [7]. [Http://www.lynda.com/Flash-CS5-tutorials/flash-professional-cs5-creating-a-simple-game-for-android-devices](http://www.lynda.com/Flash-CS5-tutorials/flash-professional-cs5-creating-a-simple-game-for-android-devices), diakses (22/11/2011).



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

NAMA : ANGGA PUTRA KUSUMA
NIM : 06.12.643
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer dan Informatika
MASA BIMBINGAN: 5 Desember 2011 s/d 5 Juni 2012
JUDUL : **PENGEMBANGAN APLIKASI GAME FLASH "SPACESHIP
IMPACT" MENGGUNAKAN ACCELEROMETER SENSOR
PADA PONSEL ANDROID**

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Selasa
Tanggal : 21 Februari 2012
Dengan Nilai : 82,25 (A) *✓*

PANITIA UJIAN SKRIPSI

KETUA,

SEKRETARIS,



Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189



Dr. Eng. Aryuanto S, ST, MT
NIP.P.1030800417

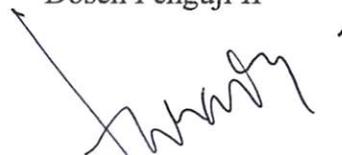
ANGGOTA PENGUJI

Dosen Penguji I

Dosen Penguji II



Sandy Nataly Mantja, S.Kom
NIP.P.1030800418



Irmalia Suryani Faradisa, ST, MT
NIP.P.1030000365



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer dan Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : ANGGA PUTRA KUSUMA
NIM : 06.12.643
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer dan Informatika
MASA BIMBINGAN: 5 Desember 2011 s/d 5 Juni 2012
JUDUL : **PENGEMBANGAN APLIKASI GAME FLASH “SPACESHIP IMPACT” MENGGUNAKAN ACCELEROMETER SENSOR PADA PONSEL ANDROID**

Tanggal	Uraian	Paraf
Penguji I 21 - 02 - 2012	Abstrak + tulisan pada BAB harus Times New Roman 12 pt	
	Batasan masalah tujuan, tidak perlu ada manfaat, metodologi	
	BAB II tidak perlu desain	
	Tidak ada yang dikembangkan harusnya dari 2D ke 3D	
	Tidak ada penjelasan teori flash, Accelerometer sensor, Android	
	Script Action Script letakkan dilampiran	
	Tidak ada pengujian sudah diujicoba pada resolusi berapa, Ujicoba user bagaimana hasilnya, dijalankan diFroyo kenapa?	
	Kesimpulan & saran (saran tidak boleh lebih baik dari kesimpulan)	
	Daftar Pustaka	



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Tanggal	Uraian	Paraf
Penguji II 21 - 02 - 2012	Margin & Font disesuaikan dengan silabus	
	Susunan diBAB III, berdasarkan perencanaan	
	Flowchart tentang Score	
	BAB IV pengujiannya bukan hanya sekedar tampilan tapi dari perancangan tersebut diuji, misal : ke penilaian, ke naik level	
	Dari hasil pengujian tersebut dibuat kesimpulan	

Mengetahui,

Dosen Pembimbing I

Dr. Eng. Aryuanto Soetedjo, ST, MT
NIP.Y.1030800417

Dosen Pembimbing II

Michael Ardita, ST, MT
NIP.P. 1031000434

Disetujui,

Dosen Penguji I

Sandy Nataly Mantja, S.Kom
NIP.P.1030800418

Dosen Penguji II

Irmalia Suryani Faradisa, ST, MT
NIP.P.1030000365



PERMOHONAN PERSETUJUAN SKRIPSI

Yang betanda tangan dibawah ini :

Nama : ANGETA PUTRA KUSUMA
N I M : 06.12.643
Semester : XI
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro S-1
Konsentrasi : ~~TEKNIK ELEKTRONIKA-~~
~~TEKNIK ENERGI LISTRIK~~
~~TEKNIK KOMPUTER DAN INFORMATIKA~~
~~TEKNIK KOMPUTER-~~
~~TEKNIK TELEKOMUNIKASI~~
Alamat : JL. SAKO POME NO. 5 MALANG

Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat **SKRIPSI Tingkat Sarjana**. Untuk melengkapi permohonan tersebut, bersama kami lampirkan persyaratan-persyaratan yang harus dipenuhi.

Adapun persyaratan-persyaratan pengambilan **SKRIPSI** adalah sebagai berikut :

1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (.....)
2. Telah lulus dan menyerahkan Laporan Praktek Kerja (.....)
3. Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (.....)
4. Telah menempuh mata kuliah ≥ 134 sks dengan IPK ≥ 2 dan tidak ada nilai E (.....)
5. Telah mengikuti secara aktif kegiatan seminar skripsi yang diadakan Jurusan (.....)
6. Memenuhi persyaratan administrasi (.....)

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Telah diteliti kebenaran data tersebut diatas
Recording Teknik Elektro


Angeta Putra Kusuma

Malang,201

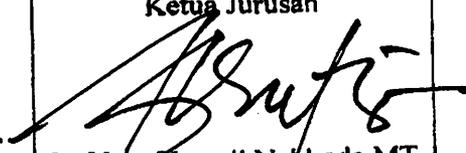
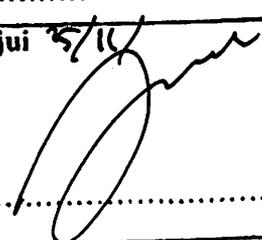
Pemohon


ANGETA PUTRA K.



LEMBAR PENGAJUAN JUDUL SKRIPSI JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik-Energi Listrik / Teknik-Elektronika / Teknik Komputer &
Informatika / ~~Teknik Komputer~~ / Teknik Telekomunikasi*)

1.	Nama Mahasiswa: <u>ANGGA PUTRA KURUMA</u>	Nim: <u>06.12.643</u>
2.	Waktu Pengajuan	Tanggal: _____ Bulan: _____ Tahun: _____
3.	Spesifikasi Judul (berilah tanda silang)**)	
	a. Sistem Tenaga Elektrik b. Energi & Konversi Energi c. Tegangan Tinggi & Pengukuran d. Sistem Kendali Industri	e. Elektronika & Komponen <input checked="" type="checkbox"/> f. Elektronika Digital & Komputer g. Elektronika Komunikasi h. lainnya
4.	Konsultasikan judul sesuai materi bidang ilmu kepada Dosen*) <u>Dr. Aryuanto, ST, MT</u>	Ketua Jurusan  <u>Ir. Yusuf Ismail Nakhoda, MT</u> NIP. Y. 1018800189
5.	Judul yang diajukan mahasiswa:	<u>Pengaruh harmonik Arus pada sistem tenaga listrik yang mempunyai beban ac dengan beban induktif.</u>
6.	Perubahan judul yang disetujui Dosen sesuai materi bidang ilmu
7.	Catatan:	
	Persetujuan Judul skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu	Disetujui Dosen <u>5/11</u> 201 <u>6</u> 

Perhatian:

1. Formulir pengajuan ini harap dikembalikan kepada jurusan paling lambat satu minggu setelah disetujui kelompok dosen keahlian dengan dilampirkan proposal skripsi beserta persyaratan skripsi sesuai form.S-1
2. Keterangan: *) Coret yang tidak perlu
**) dilingkari a, b, c,atau g sesuai bidang keahlian



INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo, Km 2
MALANG

Lampiran : 1(satu) berkas

Pembimbing skripsi

Kepada : Yth. Bapak Dr. Eng. Aryuanto Soetedjo, ST, MT
Dosen Institut Teknologi Nasional
MALANG

Yang bertanda tangan di bawah ini :

Nama : Angga Putra Kusuma
Nim : 06.12.643
Jurusan : Teknik Elektro S-1
Konsentrasi : Komputer & Informatika

Dengan ini mengajukan permohonan, kiranya Bapak bersedia menjadi Dosen pembimbing Utama / Pendamping *), untuk penyusunan skripsi dengan judul (proposals terlampir)

**PENGEMBANGAN APLIKASI GAME FLASH "SPACESHIP IMPACT"
MENGUNAKAN ACCELEROMETER SENSOR PADA PONSEL ANDROID**

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik.

Demikian permohonan kami dan atas kesediaan bapak kami ucapkan terima kasih.

Mengetahui
Ketua Jurusan T.Elektro S-1

Malang
Hormat kami


Ir. Yusuf Ismail Nakhoda, MT
NIP. X.1018800189


ANGGA PUTRA K.
06.12.643

*) coret yang tidak perlu

Form S - 3a



INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo, Km 2
MALANG

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa :

Nama : Angga Putra Kusuma
Nim : 06.12.643
Semester : 11 (sebelas)
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & informatika

Dengan ini menyatakan bersedia/~~tidak bersedia~~ *) membimbing skripsidari mahasiswa tersebut,dengan judul :

PENGEMBANGAN APLIKASI GAME FLASH “SPACESHIP IMPACT” MENGUNAKAN ACCELEROMETER SENSOR PADA PONSEL ANDROID

Demikian pernyataan ini kami buat agar dapat digunakan seperlunya.

Malang, 29 – November – 2011
Kami yang membuat pernyataan

Dr. Eng. Aryanto Soetedjo, ST, MT
NIP.Y.1030800417

Catatan :

Setelah disetujui agar formulir ini
Diserahkan mahasiswa yang bersangkutan
Kepada jurusan untuk diproses lebih lanjut
*) Coret yang tidak perlu

Form S – 3b



INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo, Km 2
MALANG

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa :

Nama : Angga Putra Kusuma
Nim : 06.12.643
Semester : 11 (sebelas)
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & informatika

Dengan ini menyatakan bersedia/tidak bersedia *) membimbing skripsidari mahasiswa tersebut,dengan judul :

PENGEMBANGAN APLIKASI GAME FLASH "SPACESHIP IMPACT" MENGUNAKAN ACCELEROMETER SENSOR PADA PONSEL ANDROID

Demikian pernyataan ini kami buat agar dapat digunakan seperlunya.

Malang, 29 – November – 2011
Kami yang membuat pernyataan

Michael Ardhita, ST, MT
NIP.

Catatan :

Setelah disetujui agar formulir ini
Diserahkan mahasiswa yang bersangkutan
Kepada jurusan untuk diproses lebih lanjut
*) Coret yang tidak perlu



BERITA ACARA SEMINAR PROPOSAL SKRIPSI JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik-Elektronika/ Teknik Komputer & Informatika*)

1.	Nama Mahasiswa: <u>ANGEA PUTRA KUSUMA</u>		Nim: <u>06.12.693</u>	
2.	Keterangan	Tanggal	Waktu	Tempat
	Pelaksanaan	<u>25-12-2011</u>	<u>09.00</u>	Ruang:
Spesifikasi Judul (berilah tanda silang)**)				
3.	a. Sistem Tenaga Elektrik		e. Elektronika & Komponen	
	b. Energi & Konversi Energi		<input checked="" type="checkbox"/> Elektronika Digital & Komputer	
	c. Tegangan Tinggi & Pengukuran		g. Elektronika Komunikasi	
	d. Sistem Kendali Industri		h. lainnya	
4.	Judul Proposal yang diseminarkan Mahasiswa	<u>Perencanaan... Aplikasi... Game Flash "SpaceShip... Impact! Menggunakan accelerometer sensor pada ponsel android</u>		
5.	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian		
6.	Catatan:	<u>- Batasan</u> <u>- Program</u> <u>- Manfaat</u> <u>- Dikirim oleh</u>		
Catatan:				
Persetujuan Judul Skripsi				
Disetujui, Dosen Keahlian I		Disetujui, Dosen Keahlian II		Disetujui, Dosen Keahlian III



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPII. DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 01 Februari 2012

Nomor : ITN- 097/EL-FTI/2012
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr/I. **DR. ENG. ARYUANTO S, ST, MT**
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama : ANGG PUTRA. K
Nim : 0612643
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik **Komputer & Informatika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya
kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai
tanggal :

05 Desember 2011 s/d 05 Juni 2012

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1,
Demikian atas perhatian serta bantuannya kami sampaikan terima kasih



Ketua Jurusan
Teknik Elektro S-1

(Signature)
Ir. Yusuf Ismail Nakhoda, MT
Nip. Y.1018800189

Tembusan Kepada Yth :

1. Mahasiswa Yang Berangkutan
2. Arsip

Form. S 4a



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI RI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigurgura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 01 Februari 2012

Nomor : ITN- 098/EL-FTI/2012
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr/I. **MICHAEL ARDITA, ST, MT**
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama : ANGGA PUTRA. K
Nim : 0612643
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik **Komputer & Informatika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya
kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai
tanggal :

05 Desember 2011 s/d 05 Juni 2012

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1,
Demikian atas perhatian serta bantuannya kami sampaikan terima kasih



Ketua Jurusan
Teknik Elektro S-1

(Signature)
Ir. Yusuf Ismail Nakhoda, MT
Nip. Y.1018800/89

Tembusan Kepada Yth :

1. Mahasiswa Yang Berangkutan
2. Arsip

Form. S 4a



FORMULIR BIMBINGAN SKRIPSI

Nama : ANGGA PUTRA KUSUMA
Nim : 06.12.643
Masa Bimbingan : 05 DESEMBER 2011 s/d 05 JUNI 2012
Judul Skripsi : PENGEMBANGAN APLIKASI GAME FLASH "SPACESHIP
IMPACT" MENGGUNAKAN ACCELEROMETER SENSOR PADA
PONSEL ANDROID

No.	Tanggal	Uraian	Paraf Pembimbing
1.	06 - 02 - 2012	Demo Program	
2.	09 - 02 - 2012	Revisi BAB IV	
3.	10 - 02 - 2012	Acc BAB I, IV & V	
4.	12 - 02 - 2012	Acc BAB II & III	
5.			
6.			
7.			
8.			
9.			
10.			

Malang,
Dosen Pembimbing I

Dr. Eng. Aryanto Soetedjo, ST, MT
NIP. 1030800417



FORMULIR BIMBINGAN SKRIPSI

Nama : ANGGA PUTRA KUSUMA
Nim : 06.12.643
Masa Bimbingan : 05 DESEMBER 2011 s/d 05 JUNI 2012
Judul Skripsi : PENGEMBANGAN APLIKASI GAME FLASH "SPACESHIP
IMPACT" MENGGUNAKAN ACCELEROMETER SENSOR PADA
PONSEL ANDROID

No.	Tanggal	Uraian	Paraf Pembimbing
1.	06 - 02 - 2012	Demo Program	
2.	07 - 02 - 2012	Acc BAB I & II	
3.	10 - 02 - 2012	Revisi BAB III & IV	
4.	12 - 02 - 2012	Acc BAB III, IV, & V	
5.			
6.			
7.			
8.			
9.			
10.			

Malang, 17 April 2012
Dosen Pembimbing II

MICHAEL ARDITA, ST, MT
NIP.P. 1031000434

Source Code :

```
import fl.transitions.Tween;
import fl.transitions.easing.*;
import fl.transitions.TweenEvent;
import flash.media.Sound;
import flash.net.SharedObject;
import flash.ui.Mouse;
import flash.media.SoundChannel;
import flash.events.MouseEvent;
import flash.display.MovieClip;
import flash.events.Event;
import fl.motion.MotionEvent;

/*****VARIABLES*****/
var STATE_INIT_GAME:String = "STATE_INIT_GAME";
var STATE_START_PLAYER:String = "STATE_START_PLAYER";
var STATE_PLAY_GAME:String = "STATE_PLAY_GAME";
var STATE_END_GAME:String = "STATE_END_GAME";
var gameState:String;

var player:MovieClip;
var enemies:Array;
var level:Number;
var score:Number;
var lives:Number;
var accel:Accelerometer;
var Lasers:Array;
var explosions:Array;
var hiddenOptions:Boolean = true;
var sc:SoundChannel;
```

```
/******SETUP******/
```

```
endScreen.visible = false;  
optionsMenu.visible = false;  
helpMenu.visible = false;
```

```
/******INTRO SCREEN******/
```

```
introScreen.play_btn.addEventListener(MouseEvent.CLICK, clickAway);  
function clickAway(event:MouseEvent):void  
{  
    moveScreenOff(introScreen);  
    helpMenu.visible = false;  
}  
introScreen.help_btn.addEventListener(MouseEvent.MOUSE_DOWN, showHelp);  
function showHelp(event:MouseEvent):void  
{  
    setChildIndex(helpMenu,numChildren-1);  
    helpMenu.visible = true;  
    helpMenu.back_btn.addEventListener(MouseEvent.MOUSE_DOWN,  
backMenu);  
}  
function backMenu(event:MouseEvent):void  
{  
    helpMenu.visible = false;  
}  
//Gesture Swipe  
Multitouch.inputMode = MultitouchInputMode.GESTURE;  
introScreen.addEventListener(TransformGestureEvent.GESTURE_SWIPE,  
swipeAway);  
function swipeAway(event:TransformGestureEvent):void  
{  
    //Swipe Left  
    if (event.offsetX == 1)
```

```

    {
        moveScreenOff(introScreen);
    }
}

```

```
function moveScreenOff(screen:MovieClip):void
```

```

{
    var          introTween          =          new
Tween(screen,"x",Strong.easeInOut,screen.x,(screen.width)*1,1,true);
    introTween.addEventListener(TweenEvent.MOTION_FINISH, tweenFinish);
    function tweenFinish(e:TweenEvent):void
    {
        trace("tweenFinish");
        gameState = STATE_INIT_GAME;
        trace(gameState);
        addEventListener(Event.ENTER_FRAME, gameLoop);
    }
}

```

```
function moveScreenOn(screen:MovieClip):void
```

```

{
    var          introTween          =          new
Tween(screen,"x",Strong.easeInOut,screen.x,(screen.width)*0,1,true);
    introTween.addEventListener(TweenEvent.MOTION_FINISH, tweenBack);
    function tweenBack(e:TweenEvent):void
    {
        trace("tweenBack");
        endScreen.visible = false;
    }
}

```

```
/******GAME STATES******/
```

```
function gameLoop(e:Event):void
```

```

{
    //Tetap pada gameState
    switch (gameState)
    {
        case STATE_INIT_GAME :
            initGame();
            break;
        case STATE_START_PLAYER :
            startPlayer();
            break;
        case STATE_PLAY_GAME :
            playGame();
            break;
        case STATE_END_GAME :
            endGame();
            break;
    }
}

/*****STATE_INIT_GAME*****/
function initGame():void
{
    level = 1;
    level_txt.text = String(level);
    score = 0;
    score_txt.text = String(score);
    lives = 3;
    lives_txt.text = String(lives);

    //Graphic
    player = new Player();
    //Enemy
    enemies = new Array();
}

```

```

//Laser
Lasers = new Array();
//Explode
explosions = new Array();
gameState = STATE_START_PLAYER;
trace(gameState);
}

/*****STATE_START_PLAYER*****/
function startPlayer():void
{
    //Create player
    player.y = stage.stageHeight - player.height;
    addChild(player);
    laserTimer.start();
    //Create accelerometer;
    accel = new Accelerometer();
    if (Accelerometer.isSupported)
    {
        accel.addEventListener(AccelerometerEvent.UPDATE, accelMove);
    }
    else
    {
        addEventListener(Event.ENTER_FRAME, movePlayer);
    }
    gameState = STATE_PLAY_GAME;
    trace(gameState);
}

function accelMove(event:AccelerometerEvent):void
{
    player.x -= event.accelerationX * 240;
    if (player.x < 0)

```

```

    {
        player.x = 0;
    }
    else if (player.x > (stage.stageWidth - player.width) )
    {
        player.x = stage.stageWidth - player.width;
    }
}

```

```

function movePlayer(e:Event):void
{
    player.x = stage.mouseX;
    if (player.x < 0)
    {
        player.x = 0;
    }
    else if (player.x > (stage.stageWidth - player.width) )
    {
        player.x = stage.stageWidth - player.width;
    }
}

```

```

var laserTimer:Timer = new Timer(500);
laserTimer.addEventListener(TimerEvent.TIMER, timerListener);
function timerListener(e:TimerEvent):void
{
    var tempLaser:MovieClip = new Laser();
    tempLaser.x = player.x +(player.width/2);
    tempLaser.y = player.y;
    tempLaser.speed = 10;
    Lasers.push(tempLaser);
    addChild(tempLaser);
    var music:Sound = new Shoot();
    var sc:SoundChannel = music.play();
}

```

```
}
```

```
function moveLaser():void
```

```
{
```

```
    //Move Laser
```

```
    var tempLaser:MovieClip;
```

```
    for (var i=Lasers.length-1; i>=0; i--)
```

```
    {
```

```
        tempLaser = Lasers[i];
```

```
        tempLaser.y -= tempLaser.speed;
```

```
        if (tempLaser.y < 0)
```

```
        {
```

```
            removeLaser(i);
```

```
        }
```

```
    }
```

```
    var tempExplosion:MovieClip;
```

```
    for (i=explosions.length-1; i>=0; i--)
```

```
    {
```

```
        tempExplosion = explosions[i];
```

```
        if (tempExplosion.currentFrame >= tempExplosion.totalFrames)
```

```
        {
```

```
            removeExplosion(i);
```

```
        }
```

```
    }
```

```
}
```

```
/******STATE_PLAY_GAME******/
```

```
function playGame():void
```

```
{
```

```
    //Set up the enemies, explosions etc.
```

```
    makeEnemies();
```

```
    moveEnemies();
    moveLaser();
    testCollisions();
    testForEnd();
}
```

```
function makeEnemies():void
{
    var chance:Number = Math.floor(Math.random() * 100);
    if (chance <= 1 + level)
    {
        var tempEnemy:MovieClip;
        tempEnemy = new Enemy();
        tempEnemy.speed = 2;
        tempEnemy.x = Math.round(Math.random() * 1000);
        trace("tempEnemy");
        addChild(tempEnemy);
        enemies.push(tempEnemy);
    }
}
```

```
function moveEnemies():void
{
    var tempEnemy:MovieClip;
    for (var i:int =enemies.length-1; i>=0; i--)
    {
        tempEnemy = enemies[i];

        tempEnemy.rotation += (Math.round(Math.random()*10-5));
        tempEnemy.x -=
(Math.sin((Math.PI/180)*tempEnemy.rotation))*tempEnemy.speed;
        tempEnemy.y +=
(Math.cos((Math.PI/180)*tempEnemy.rotation))*tempEnemy.speed;
```

```

    if (tempEnemy.x < 0)
    {
        tempEnemy.x = 0;
    }
    if (tempEnemy.x > (stage.stageWidth - tempEnemy.width))
    {
        tempEnemy.x = stage.stageWidth - tempEnemy.width;
    }
    if (tempEnemy.y > stage.stageHeight) //
tempEnemy.hitTestObject(player))
    {
        makeExplosion(tempEnemy.x, tempEnemy.y);
        removeEnemy(i);
        lives--;
        lives_txt.text = String(lives);
    }
}
}

```

```

function testCollisions():void
{
    var tempEnemy:MovieClip;
    var tempLaser:MovieClip;

    enemy:for (var i:int=enemies.length-1; i>=0; i--)
    {
        tempEnemy = enemies[i];
        for (var j:int=Lasers.length-1; j>=0; j--)
        {
            tempLaser = Lasers[j];
            if (tempEnemy.hitTestObject(tempLaser))
            {
                makeExplosion(tempEnemy.x, tempEnemy.y);
                removeEnemy(i);
            }
        }
    }
}

```

```

        removeLaser(j);
        score++;
        score_txt.text = String(score);
        break enemy;
    }
}
}
}
}

```

```

function makeExplosion(ex:Number, ey:Number):void
{
    var tempExplosion:MovieClip;
    tempExplosion = new Explosion();
    tempExplosion.x = ex;
    tempExplosion.y = ey;
    addChild(tempExplosion);
    explosions.push(tempExplosion);
    var music:Sound = new Explode();
    var sc:SoundChannel = music.play();
}

```

```

function testForEnd():void
{
    if (score > level * 10)
    {
        level++;
        level_txt.text = String(level);
    }
    if (lives == 0)
    {
        gameState = STATE_END_GAME;
    }
}

```

```

        trace(gameState);
    }
}

function removeEnemy(idx:int)
{
    removeChild(enemies[idx]);
    enemies.splice(idx,1);
}

function removeLaser(idx:int)
{
    removeChild(Lasers[idx]);
    Lasers.splice(idx,1);
}

function removeExplosion(idx:int)
{
    removeChild(explosions[idx]);
    explosions.splice(idx,1);
}

/*****END SCREEN*****/
function endGame():void
{
    removeGame();
    endScreen.visible = true;
    endScreen.x = 0;
    removeEventListener(Event.ENTER_FRAME, gameLoop);
    showResults();
}
endScreen.backmenu_btn.addEventListener(MouseEvent.CLICK, backtoMenu);

```

```
function backtoMenu(event:MouseEvent):void
```

```
{  
    moveScreenOn(introScreen);  
}
```

```
/******REMOVE GAME******/
```

```
function removeGame():void
```

```
{  
    trace("Remove Game");  
    for (var i:int = enemies.length-1; i >=0; i--)  
    {  
        removeEnemy(i);  
    }  
    for (var j:int = Lasers.length-1; j >=0; j--)  
    {  
        removeLaser(j);  
    }  
    for (var k:int = explosions.length-1; k >=0; k--)  
    {  
        removeExplosion(k);  
    }  
    removeChild(player);  
    laserTimer.stop();  
}
```

```
function showResults():void
```

```
{  
    trace("Show Results");  
    //HIGH SCORE  
    endScreen.enter_btn.visible = false;  
    endScreen.highScoreName.visible = false;  
    var so:SharedObject = SharedObject.getLocal("alltimeHighScore");
```

```

if (so.data.score == undefined || score > so.data.score)
{
    endScreen.highScore.text = "Kamu mencapai Level " + level + " dengan
Score " + score + "!\nMasukkan nama kamu pada kolom dibawah :";
    endScreen.enter_btn.visible = true;
    endScreen.highScoreName.visible = true;
}
else
{
    endScreen.highScore.text = "Score kamu adalah " + score + "!\nBelum
mengalahkan Score " + so.data.score + " dari " + so.data.name;
}
endScreen.enter_btn.addEventListener(MouseEvent.CLICK, clickEnter);
function clickEnter(event:MouseEvent):void
{
    trace("clickEnter");
    endScreen.highScore.text = "Selamat kepada : " +
endScreen.highScoreName.text + " Kamu mencapai Level " + level + " dengan Score "
+ score;

    so.data.score = score;
    so.data.level = level;
    so.data.name = endScreen.highScoreName.text;
    so.flush();
    endScreen.enter_btn.visible = false;
    endScreen.highScoreName.visible = false;
}
endScreen.play_btn.addEventListener(MouseEvent.CLICK, clickFinalAway);
function clickFinalAway(event:MouseEvent):void
{
    trace("clickFinalAway");
    moveScreenOff(endScreen);
}
}

```

```

/*****OPTIONS MENU*****/
//Menu key
stage.addEventListener(KeyboardEvent.KEY_UP, optionsKey);
function optionsKey(event:KeyboardEvent):void
{
    if (event.keyCode == 16777234 || event.keyCode == 13)
    {
        if (hiddenOptions)
        {
            setChildIndex(optionsMenu,numChildren-1);
            optionsMenu.visible = true;
            optionsMenu.addEventListener(MouseEvent.CLICK, exitApp);
            pauseGame();
        }
        else
        {
            optionsMenu.visible = false;
            optionsMenu.removeEventListener(MouseEvent.CLICK,
exitApp);
            resumeGame();
        }
        hiddenOptions = ! hiddenOptions;
    }
}

function exitApp(event:MouseEvent)
{
    trace("exitApp");
    //NativeApplication.nativeApplication.exit(0);
}
//Keep screen awake
//NativeApplication.nativeApplication.systemIdleMode
SystemIdleMode.KEEP_AWAKE;
=

```

```
stage.addEventListener(Event.DEACTIVATE, Deactivate);
function Deactivate(event:Event):void
{
    pauseGame();
}
stage.addEventListener(Event.ACTIVATE, Activate);
function Activate(event:Event):void
{
    resumeGame();
}

function pauseGame():void
{
    if (gameState == STATE_PLAY_GAME)
    {
        removeEventListener(Event.ENTER_FRAME, gameLoop);
        laserTimer.stop();
        accel.removeEventListener(AccelerometerEvent.UPDATE, accelMove);
    }
}

function resumeGame():void
{
    if (gameState == STATE_PLAY_GAME)
    {
        addEventListener(Event.ENTER_FRAME, gameLoop);
        laserTimer.start();
        accel.addEventListener(AccelerometerEvent.UPDATE, accelMove);
    }
}
```