

**IMPLEMENTASI J2ME PADA *GAME*  
PERTEMPURAN PESAWAT “STEALTH”**

**TUGAS AKHIR**

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



oleh:

Nama : Fajar Wahyudin

No. Mahasiswa : 01523024

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

**2006**

**LEMBAR PERNYATAAN KEASLIAN  
HASIL TUGAS AKHIR**



Saya yang bertanda tangan di bawah ini,

Nama : Fajar Wahyudin

No. Mahasiswa : 01523024

Menyatakan bahwa seluruh komponen dan isi dalam Laporan Kerja Praktek ini adalah hasil karya saya sendiri.

Apabila di kemudian hari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya saya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.

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

Yogyakarta, 15 Juni 2006

A handwritten signature in black ink, appearing to read 'Fajar Wahyudin'.

( Fajar Wahyudin )

**LEMBAR PENGESAHAN PEMBIMBING**

**IMPLEMENTASI J2ME PADA GAME  
PERTEMPURAN PESAWAT “STEALTH”**

**TUGAS AKHIR**

oleh:

Nama : Fajar Wahyudin

No. Mahasiswa : 01523024

Yogyakarta, 15 Juni 2006

Pembimbing,

A handwritten signature in black ink, appearing to read 'Taufiq Hidayat', with a stylized flourish at the end.

Taufiq Hidayat, ST, MCS.

**LEMBAR PENGESAHAN PENGUJI**  
**IMPLEMENTASI J2ME PADA GAME**  
**PERTEMPURAN PESAWAT "STEALTH"**

**TUGAS AKHIR**

oleh:

Nama : Fajar Wahyudin

No. Mahasiswa : 01523024

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat  
untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika  
Fakultas Teknologi Industri Universitas Islam Indonesia

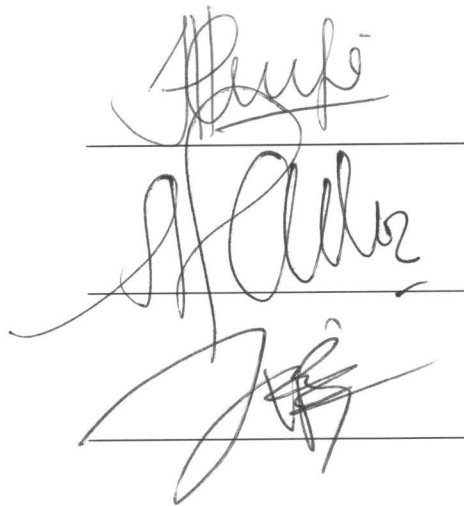
Yogyakarta, 29 Juni 2006

**Tim Penguji**

Taufiq Hidayat, ST, MCS.  
Ketua

Sri Kusuma Dewi, S.Si, MT.  
Anggota I

Irving Vitra Paputungan, ST.  
Anggota II



Mengetahui,  
Dekan Fakultas Teknologi Industri  
Universitas Islam Indonesia



Muhammad Wahid, ST, M.Sc.

Insyah Allah dengan ikhlas kupersembahkan kepada-Nya

Semoga memberikan kebaikan di dunia dan di kehidupan sesudahnya

Maha Suci Allah yang telah menyempurnakan ikhtiar insan-Nya

Kepada kedua orangtua dan keluargaku..

Untuk Cintaku..

Dan seluruh pihak yang telah berjasa besar

Hanya ucapan terima kasih yang dapat terucap

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

“Bacalah dengan (menyebut) nama Tuhanmu Yang menciptakan”

(Q.S. Al ‘Alaq : 1)

“...Apa yang diberikan Rasul kepadamu, maka terimalah.

Dan apa yang dilarangnya bagimu, maka tinggalkanlah...”

(Q.S. Al Hasyr : 7)



## KATA PENGANTAR

Assalamu'alaikum wa rahmatullaahi wa barakaatuh

Segala puji dan syukur hanyalah milik Allah Subhanahu Wa Ta'ala yang Maha Pengasih dan Maha Penyayang, yang telah menjadikan manusia beriman dan berilmu setelah sebelumnya berada dalam kondisi yang lemah dan diliputi kebodohan. Atas izin dan kehendak-Nya, tugas akhir berjudul "IMPLEMENTASI J2ME PADA GAME PERTEMPURAN PESAWAT STEALTH" ini akhirnya dapat terselesaikan.

Tugas Akhir ini merupakan salah satu syarat untuk memperoleh gelar sarjana Teknik Informatika di Universitas Islam Indonesia.

Penulis dalam kesempatan ini mengucapkan terima kasih kepada :

1. Papa dan mama serta keluarga tercinta yang telah memberikan banyak semangat dan dorongan.
2. Bapak Fathul Wahid, ST, M.Sc. selaku Dekan Fakultas Teknologi Industri.
3. Bapak Yudi Prayudi, S.Si, M.Kom. selaku Ketua Jurusan Teknik Informatika.
4. Bapak Taufiq Hidayat, ST, MCS. selaku Pembimbing Tugas Akhir. Terima kasih atas segala kesabarannya, bantuan, dukungan, pengetahuan dan semua kemudahan yang selalu diberikan.
5. Inda, yang selalu memberi motivasi dalam hidup ini.
6. Adit, Wahyu, Ujang, dan Towie yang selalu bercanda dan menyebarkan.
7. Teman-teman "Biner" angkatan 2001 Teknik Informatika yang menjadi sahabat dalam studiku.

## SARI

Makin berkembangnya kebutuhan manusia akan teknologi terutama yang tidak terikat oleh tempat dan waktu, membuat fungsi ponsel tidak lagi hanya sebagai alat komunikasi tetapi dapat sebagai sebuah sarana hiburan. Salah satu bentuk hiburan adalah permainan (*game*). Hadirnya teknologi J2ME (*Java 2 Micro Edition*) yang memungkinkan diimplementasikannya aplikasi *game* pada terminal bermemori kecil dapat menjadi sebuah hiburan yang dapat dimainkan kapan saja dan dimana saja.

*Game* pertempuran pesawat "Stealth" adalah *game* dengan tampilan 2D, *game* ini termasuk jenis *Skill and Action Games* yang melibatkan kemampuan persepsi dan motorik pemain. *Game* ini terdiri dari tiga level dan merupakan *game* pertempuran bersifat konfrontasi langsung dengan lingkungan atau dengan pemain lain untuk dapat mencapai kemenangan. Setiap levelnya memiliki tingkat kesulitan yang berbeda, sehingga pemain mendapatkan variasi dalam bermain. Pemain akan kalah jika life bar dan jumlah nyawa telah habis, dan *score* tertinggi yang belum pernah tercapai akan tersimpan pada form *high score*.

Studi kasus diimplementasikan pada telepon seluler dengan java *platform* yang mendukung MIDP 2.0. Dengan berbagai keterbatasan pada lingkungan pengembangannya, teknologi J2ME tetap menjadi satu teknologi yang menarik untuk meningkatkan dan menambah fitur-fitur yang ada dalam sebuah perangkat telepon seluler.

Kata Kunci: *J2ME, Combat Game*



## TAKARIR

<i>game</i>	permainan
<i>platform</i>	lingkungan sistem
<i>high score</i>	nilai tertinggi
<i>background</i>	latar belakang
<i>input</i>	masukan
<i>output</i>	keluaran
<i>level</i>	tingkat
<i>goal</i>	tujuan
<i>combat flight</i>	pertempuran pesawat
<i>software</i>	perangkat lunak
<i>single player</i>	game dengan mode pemain tunggal
<i>game worlds</i>	Elemen utama yang terdapat dalam suatu permainan
<i>game board</i>	layanan aplikasi game berupa bentuk tampilan game beserta latar belakangnya
<i>option</i>	opsi
<i>structural approach</i>	pendekatan terstruktur

## DAFTAR ISI

JUDUL .....	i
LEMBAR PERNYATAAN KEASLIAN .....	ii
LEMBAR PENGESAHAN PEMBIMBING .....	iii
LEMBAR PENGESAHAN PENGUJI .....	iv
HALAMAN PERSEMBAHAN .....	v
HALAMAN MOTTO .....	vi
KATA PENGANTAR .....	vii
SARI .....	ix
TAKARIR .....	x
DAFTAR ISI .....	xi
DAFTAR GAMBAR .....	xiv
DAFTAR TABEL .....	xv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan Penelitian .....	3
1.5 Manfaat Penelitian .....	3
1.6 Metode Penelitian .....	3
1.6.1 Metode Pengumpulan Data .....	3
1.6.2 Metode Analisis .....	4
1.6.3 Tahapan Pengembangan .....	4
1.7 Sistematika Penulisan .....	5
BAB II LANDASAN TEORI .....	7
2.1 Pendekatan Berorientasi Objek .....	7
2.1.1 Kelas .....	7
2.1.1.1 Kegunaan Kelas .....	8
2.1.1.2 Karakteristik Kelas .....	8
2.1.1.3 Properti Kelas .....	9
2.2 Teknologi Java .....	10
2.2.1 Java 2 .....	11
2.2.2 Java 2 Micro Edition (J2ME) .....	12
2.2.2.1 J2ME Configuration .....	13
2.2.2.2 J2ME Profile .....	14
2.2.2.3 API (Application Programming Interface) .....	15
2.2.2.4 MIDlet .....	16
2.2.2.4.1 Life Cycle .....	16
2.2.2.4.2 User Interface .....	18
2.2.3 Pemrograman Java pada Perangkat Lunak .....	19
2.2.3.1 Pemrograman GUI .....	19
2.3 Teori <i>Game</i> .....	21
2.3.1 Klasifikasi Permainan dan Definisi Permainan Pertempuran .....	21
2.3.2 Elemen <i>Game</i> .....	22
2.3.3 Teori Dalam Pembuatan Program Permainan .....	24

BAB III ANALISIS KEBUTUHAN .....	26
3.1 Metode Analisis.....	26
3.2 Hasil Analisis .....	26
3.2.1 Analisis Masukan.....	26
3.2.2 Analisis Proses.....	27
3.2.3 Analisis Keluaran.....	29
3.2.4 Analisis User Interface.....	29
BAB IV PERANCANGAN .....	30
4.1 Metode Perancangan.....	30
4.2 Hasil Perancangan .....	30
4.2.1 Rancangan Umum Aplikasi.....	30
4.2.1.1 Use Case Diagram .....	31
4.2.1.2 Class Diagram .....	32
4.2.1.3 Sequence Diagram.....	34
4.2.1.4 Diagram Aktivitas .....	41
4.3 Rancangan Interface .....	43
4.3.1 Rancangan Antarmuka Splash Screen .....	43
4.3.2 Rancangan Antarmuka Menu Utama.....	43
4.3.3 Rancangan antarmuka Permainan.....	44
4.3.4 Rancangan Antarmuka Masukan Nama.....	45
4.3.5 Rancangan Antarmuka <i>High Score</i> .....	45
4.3.6 Rancangan Antarmuka <i>Option</i> .....	46
4.3.7 Rancangan Antarmuka <i>Help</i> .....	46
4.3.8 Rancangan Antarmuka <i>About</i> .....	47
BAB V IMPLEMENTASI .....	48
5.1 Batasan Implementasi.....	48
5.1.1 Lingkungan Pengembangan .....	48
5.1.2 Lingkungan Target.....	49
5.1.3 Asumsi – asumsi Baru.....	50
5.2 Implementasi.....	50
5.2.1 Implementasi Sistem.....	50
5.2.2 Implementasi Antarmuka .....	52
5.2.2.1 Antarmuka Splash Screen.....	52
5.2.2.2 Antarmuka Menu Utama .....	53
5.2.2.3 Antarmuka Permainan .....	53
5.2.2.4 Antarmuka Masukan Nama .....	56
5.2.2.5 Antarmuka <i>High Score</i> .....	57
5.2.2.6 Antarmuka <i>Option</i> .....	57
5.2.2.7 Antarmuka <i>Help</i> .....	58
5.2.2.8 Antarmuka <i>About</i> .....	58
BAB VI ANALISIS KINERJA .....	59
6.1 Pengujian Perangkat Lunak .....	59
6.2 Hasil Pengujian .....	59
6.3 Analisis Kesesuaian.....	60
6.4 Gambaran Umum Sistem.....	62
BAB VII SIMPULAN DAN SARAN .....	65

7.1	Simpulan .....	65
7.2	Saran .....	66
DAFTAR PUSTAKA .....		68

## DAFTAR GAMBAR

Gambar 2.1 Pembagian lingkungan kerja pada teknologi Java .....	12
Gambar 2.2 J2ME profile .....	15
Gambar 2.3 Daur hidup MIDlet .....	17
Gambar 2.4 Proses pembuatan aplikasi MIDlet.....	19
Gambar 4.1 Use Case Diagram .....	31
Gambar 4.2 Class Diagram .....	33
Gambar 4.3 Diagram sekuen pemanggilan <i>game</i> .....	34
Gambar 4.4 Diagram sekuen <i>game</i> baru .....	35
Gambar 4.5 Diagram sekuen <i>High Score</i> .....	36
Gambar 4.6 Diagram sekuen setting <i>suara</i> .....	37
Gambar 4.7 Diagram sekuen <i>help</i> .....	37
Gambar 4.8 Diagram sekuen <i>about</i> .....	38
Gambar 4.9 Diagram sekuen memulai <i>game</i> .....	39
Gambar 4.10 Diagram sekuen menjalankan pesawat.....	40
Gambar 4.11 Diagram sekuen <i>input score</i> .....	41
Gambar 4.12 Diagram aktivitas .....	42
Gambar 4.13 Rancangan antarmuka <i>splash screen</i> .....	43
Gambar 4.14 Rancangan antarmuka menu utama.....	44
Gambar 4.15 Rancangan antarmuka permainan .....	44
Gambar 4.16 Rancangan antarmuka masukan nama.....	45
Gambar 4.17 Rancangan antarmuka <i>high sroce</i> .....	45
Gambar 4.18 Rancangan antarmuka <i>option</i> .....	46
Gambar 4.19 Rancangan antarmuka form <i>help</i> .....	46
Gambar 4.20 Rancangan antarmuka <i>about</i> .....	47
Gambar 5.1 Antarmuka <i>splash screen</i> .....	52
Gambar 5.2 Antarmuka menu utama .....	53
Gambar 5.3 Antarmuka permainan <i>level 1</i> .....	54
Gambar 5.4 Antarmuka permainan <i>level 2</i> .....	54
Gambar 5.5 Antarmuka permainan <i>level 3</i> .....	55
Gambar 5.6 Antarmuka informasi permainan saat menang .....	55
Gambar 5.7 Antarmuka informasi permainan saat kalah .....	56
Gambar 5.8 Antarmuka Masukan Nama .....	56
Gambar 5.9 Antarmuka <i>High Score</i> .....	57
Gambar 5.10 Antarmuka form <i>Option</i> .....	57
Gambar 5.11 Antarmuka <i>Help</i> .....	58
Gambar 5.12 Antarmuka <i>About</i> .....	58
Gambar 6.1 Tampilan Form Menu game “Stealth” .....	63
Gambar 6.2 Tampilan FormMenu game “Aleste” .....	64
Gambar 6.3 Antarmuka game “Stealth” .....	64
Gambar 6.4 Antarmuka game “Aleste” .....	64

## DAFTAR TABEL

Tabel 6.1 Tabel hasil analisis.....	60
Tabel 6.2 Tabel perbandingan <i>game</i> .....	63

# BABI

## PENDAHULUAN



### 1.1 Latar Belakang

Meningkatnya kebutuhan manusia untuk berkomunikasi dan bekerja tanpa dibatasi oleh tempat dan waktu membutuhkan teknologi yang memiliki tingkat mobilitas yang tinggi. Salah satu hasil teknologi itu adalah telepon seluler (*handphone*). Dengan adanya telepon seluler tingkat mobilitas manusia menjadi meningkat sehingga membantu dalam pelaksanaan kegiatan menjadi lebih efisien.

Makin berkembangnya kebutuhan manusia akan teknologi terutama yang tidak terikat oleh tempat dan waktu membuat fungsi telepon seluler tidak lagi hanya sebagai alat komunikasi. Manusia juga membutuhkan hiburan yang dapat dimainkan kapan dan di mana saja. Salah satu bentuk hiburan adalah permainan (*game*). Karena itu *game* menjadi bagian penting dari telepon seluler. Ada beberapa jenis *game*, salah satunya adalah *game combat flight* (pertempuran pesawat). Jenis *game* ini cukup populer karena permainannya membutuhkan ketangkasan dan pemikiran dalam penyelesaian tiap misi dari pemainnya dan cukup menghibur untuk mengisi waktu.

Kebutuhan manusia akan *game mobile* yang meningkat mendorong pembuatan *game-game* yang makin beragam jenisnya dengan bahasa pemrograman yang beragam pula. Salah satunya adalah Java 2 Micro Edition (J2ME). Teknologi ini adalah salah satu dari bahasa pemrograman yang populer

digunakan untuk membuat *software* dan *game mobile*. Dan salah satu keuntungannya adalah dapat berjalan di berbagai *platform* yang mendukung teknologi Java.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disebutkan, maka dapat dibuat rumusan masalah sebagai berikut :

“Bagaimana mengimplementasikan teknologi J2ME dalam pembuatan *game* pertempuran pesawat pada telepon seluler ?”.

## 1.3 Batasan Masalah

Untuk menghindari meluasnya ruang lingkup yang dibahas pada penelitian ini, maka diberikan batasan masalah sebagai berikut :

1. Perangkat lunak ini didesain dengan menggunakan teknologi J2ME.
2. Perangkat lunak ini merupakan *game* pertempuran pesawat.
3. Perangkat lunak ini didesain dengan ukuran layar untuk *handphone* series 60 yaitu 176 x 208.
4. *Game* ini memiliki 3 *level* yang ditentukan oleh jumlah musuh.
5. Dalam uji coba digunakan emulator dan ponsel dengan Java platform yang mendukung MIDP 2.0.
6. Hanya ada 1 mode permainan, yaitu mode misi. Mode misi dimainkan dengan menyelesaikan misi tiap *level* untuk menuju *level* berikutnya.
7. *Game* ini merupakan *singleplayer game*.



#### **1.4 Tujuan Penelitian**

Tujuan penelitian adalah :

1. Mengimplementasikan teknologi J2ME pada *game mobile*.
2. Membuat *game* pertempuran pesawat.

#### **1.5 Manfaat Penelitian**

Manfaat penelitian ini adalah :

1. Menghasilkan aplikasi *game* pertempuran pesawat dengan menggunakan teknologi J2ME
2. Memberikan alternatif hiburan dalam bentuk *game* pertempuran pesawat kepada para pengguna telepon seluler.
3. Mendapatkan pembelajaran lebih dalam tentang bahasa Java terutama J2ME.

#### **1.6 Metode Penelitian**

Beberapa metode yang digunakan untuk menyelesaikan penelitian ini adalah sebagai berikut :

##### **1.6.1 Metode Pengumpulan Data**

Pengumpulan data pada penelitian ini menggunakan metode studi literatur dengan mengumpulkan literatur yang layak dijadikan landasan, melalui buku-buku referensi, internet, dan forum diskusi, terutama yang banyak diterapkan oleh para praktisi pengembang aplikasi *mobile* dan praktisi pengembang *game*.

### 1.6.2 Metode Analisis

Untuk melakukan analisis digunakan analisis sistem berorientasi obyek dengan menggunakan model perancangan berbasis UML (*Unified Modelling Language*).

### 1.6.3 Tahapan Pengembangan

Tahapan pengembangan yang dilalui pada penelitian ini adalah sebagai berikut:

1. Observasi dan Pengumpulan Data.

Pada tahap ini dilakukan pencarian dan pengumpulan data-data, buku-buku dan berbagai literatur yang dapat digunakan sebagai referensi dan acuan ketika melakukan penelitian.

2. Analisis Kebutuhan dan Perancangan.

Pada tahap ini dilakukan proses analisis terhadap berbagai kebutuhan yang mungkin diperlukan oleh sistem yang akan dibangun dan dilanjutkan dengan proses perancangan aplikasi perangkat lunak.

3. Proses Implementasi dan Pengujian.

Implementasi dilakukan setelah semua bagian dalam tahap perancangan sudah layak dilanjutkan menuju proses implementasi. Selama implementasi, pada tiap-tiap bagian tertentu dilakukan proses pengujian secara bertahap hingga pada akhirnya seluruh hasil implementasi telah mengalami pengujian dengan baik.

#### 4. Analisis Kinerja.

Tahapan ini dilakukan untuk menguji dan mengevaluasi secara keseluruhan kinerja perangkat lunak yang dibuat. Dari analisis kinerja dapat dilihat kesesuaian rancangan dan hasil akhir yang dihasilkan.

### 1.7 Sistematika Penulisan

Penulisan laporan berguna untuk memberi gambaran umum dari keseluruhan isi laporan. Penulisan laporan ini dibagi dalam beberapa bab, yaitu :

Bab I berisi pendahuluan yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan atika penulisan.

Bab II berisi landasan teori yang meliputi gambaran umum tentang teori yang diterapkan dalam pembuatan *software* ini. Dalam bab ini dijelaskan tentang teori-teori pemrograman berorientasi objek, Teknologi java, java2, J2ME, pemrograman java pada perangkat lunak dan teori pembuatan program permainan.

Bab III berisi analisis kebutuhan perangkat lunak yang meliputi metode analisis dan hasil analisis. Dalam bab ini dijelaskan tentang kebutuhan *software* yang diperlukan, seperti masukan, keluaran, proses, serta spesifikasi *software game* yang digunakan.

Bab IV berisi perancangan perangkat lunak yang meliputi metode perancangan dan hasil perancangan. Dalam bab ini dijelaskan tentang perancangan dari *software game* ini dengan menggunakan *Unified Modelling Language* (UML), rancangan tampilan yang akan dibangun *software game* ini dan

spesifikasi *hardware* yang digunakan. Dalam perancangan ini, digunakan empat diagram untuk memodelkan *software game*, yaitu diagram *use-case*, diagram aktivitas, diagram sekuen, dan diagram kelas. Keempat diagram ini mendokumentasikan aspek statis dan aspek dinamis dari *software game* ini.

## BAB II

### LANDASAN TEORI

#### 2.1 Pendekatan Berorientasi Objek

Pendekatan berorientasi objek adalah cara memandang persoalan menggunakan model-model yang diorganisasikan seputar konsep objek yang mengkombinasikan struktur data dan perilaku entitas.

Pada pendekatan berorientasi objek, blok pembangun utama perangkat lunak adalah objek dan/atau kelas. Kandidat-kandidat kelas umumnya merupakan sesuatu yang diperoleh dari kosakata di ruang persoalan dan/atau ruang solusi. Kelas merupakan deskripsi himpunan objek yang serupa. Setiap objek memiliki identitas (sesuatu yang membedakan dengan objek lain), *state* (data yang diasosiasikan dengannya), dan perilaku.

Adanya pewarisan dalam pendekatan berorientasi objek memungkinkan struktur dan perilaku dipakai bersama dalam beberapa subklas serupa tanpa mengalami redundansi dan penggunaan ulang (*reuse*) rancangan dan kode pada proyek-proyek berikutnya [HAR03a].

##### 2.1.1 Kelas

Kelas merupakan deskripsi satu objek atau lebih dengan sekumpulan atribut dan layanan seragam, termasuk deskripsi penciptaan objek baru di kelas itu. Kelas objek mendeskripsikan kelompok objek dengan properti-properti serupa, perilaku

yang umum, kelompok struktur data yang dibentuk oleh kelas, keterhubungan terhadap himpunan objek-objek lain dan semantik yang umum.

#### 2.1.1.1 Kegunaan Kelas

Kegunaan kelas adalah sebagai berikut :

- a. Kelas merupakan abstraksi dari masalah yang dihadapi. Abstraksi memberikan kemampuan generalisasi beberapa kasus spesifik menjadi induk kasus-kasus yang serupa.
- b. Definisi nama kelas dan atribut, dan operasi-operasi cukup disimpan dan ditulis sekali per kelas. Semua objek dalam kelas mendapatkan manfaat penggunaan kembali (*reuse*) kode itu.
- c. Kelas merupakan penerapan prinsip pengkapsulan yang memadukan properti-properti unit komputasi (data dan operasi) untuk mengembangkan satu tanggungjawab tertentu yang diembannya.

#### 2.1.1.2 Karakteristik Kelas

Kelas memiliki karakteristik-karakteristik yang unik yaitu :

##### 1. *Tangibility*

Kemampuan kelas untuk merepresentasikan sesuatu yang terdapat entitas fisiknya atau informasi abstrak.

##### 2. *Inclusiveness*

Kelas dapat berisi kelas-kelas lain dalam satu kelas (kelas agregat) atau kelas tidak berisi kelas-kelas lain (kelas atomik).

### 3. *Sequentially*

Kelas dapat memiliki banyak alur kendali (kelas konkuren) atau kelas dikendalikan dari sumberdaya eksternal (kelas sekuen).

### 4. *Persistence*

Sifat kelas berdasar *persistence* terbagi sebagai berikut :

- (i). *Transient* , yaitu diciptakan dan dihilangkan semua operasi program.
- (ii). *Temporary*, yaitu diciptakan selama operasi dan dihilangkan ketika program berakhir.
- (iii). *Permanent*, yaitu disimpan di basisdata.

### 5. *Integrity*

Kelas dapat dikorupsi, yaitu sumber dayanya tidak dijaga dari pengaruh luar, atau kelas *guarded*, yaitu kelas memaksakan pengendalian pengaksesan sumberdaya di objek.

#### 2.1.1.3 Properti Kelas

Kelas memiliki properti-properti sebagai berikut :

##### a. Nama

Nama digunakan untuk membedakan dengan kelas-kelas lain.

##### b. Atribut

Atribut adalah properti bernama di kelas yang mendeskripsikan satu *range* nilai yang dapat dimiliki instan kelas dan merepresentasikan suatu properti dari sesuatu yang dimodelkan. Kelas dapat memiliki beberapa atribut atau tidak sama sekali.

c. Operasi/Metode

Operasi adalah implementasi layanan yang dapat diminta dari suatu objek. Kelas dapat memiliki beberapa objek atau tidak sama sekali.

d. Tanggungjawab (*responsibilities*)

Tanggungjawab adalah kontrak atau kewajiban kelas. Kelas yang berisi sejumlah atribut dan operasi merupakan manifestasi tanggungjawab yang diembannya untuk dilaksanakan.

## 2.2 Teknologi Java

Java adalah bahasa pemrograman berorientasi objek yang dikembangkan oleh *Sun Microsystems*. Sebagai bahasa pemrograman, Java dikenal sebagai bahasa pemrograman tingkat tinggi dengan fitur-fitur utama sebagai berikut [WIC02] :

a. *Simple*

Java dirancang untuk mudah dipelajari, terutama bagi yang telah mengenal C/C++ akan mudah untuk berpindah ke Java.

b. *Object Oriented*

Java adalah bahasa pemrograman berorientasi objek yang merupakan pemrograman masa depan.

c. *Robust and Secure*

Java dirancang sebagai bahasa pemrograman yang handal dan aman. Manajemen memori dan pengaturan keamanan yang bagus merupakan keunggulan bahasa ini.



d. *Architecture Neutral and Portable*

Java dirancang untuk dapat dijalankan di semua platform, tanpa peduli apakah arsitektur perangkat Intel, AMD atau Machintos. Java memungkinkan untuk berjalan pada operasi apapun, baik di Windows, Linux, Solaris, MacOS dan sebagainya tanpa harus melakukan kompilasi ulang pada program Java yang dibuat,

e. *High Perfomance*

Java dirancang untuk menghasilkan aplikasi-aplikasi yang terbaik.

### 2.2.1 Java 2

Setelah mengalami pengembangan-pengembangan, Sun Microsystem akhirnya mengenalkan Java versi terbaru yaitu Java versi 1.2 atau lebih dikenal dengan nama Java 2 yang terdiri atas JDK dan JRE versi 1.2. Aplikasi-aplikasi yang mendukung dan kompatibel dengan Java 2 dikenal sebagai Java 2 Compliant. Terdapat tiga kategori dalam Java 2 ini, yaitu [WIC02] :

1. Java 2 Standard Edition (J2SE)

Kategori ini digunakan untuk menjalankan dan mengembangkan aplikasi-aplikasi Java pada *level* PC (Personal Computer).

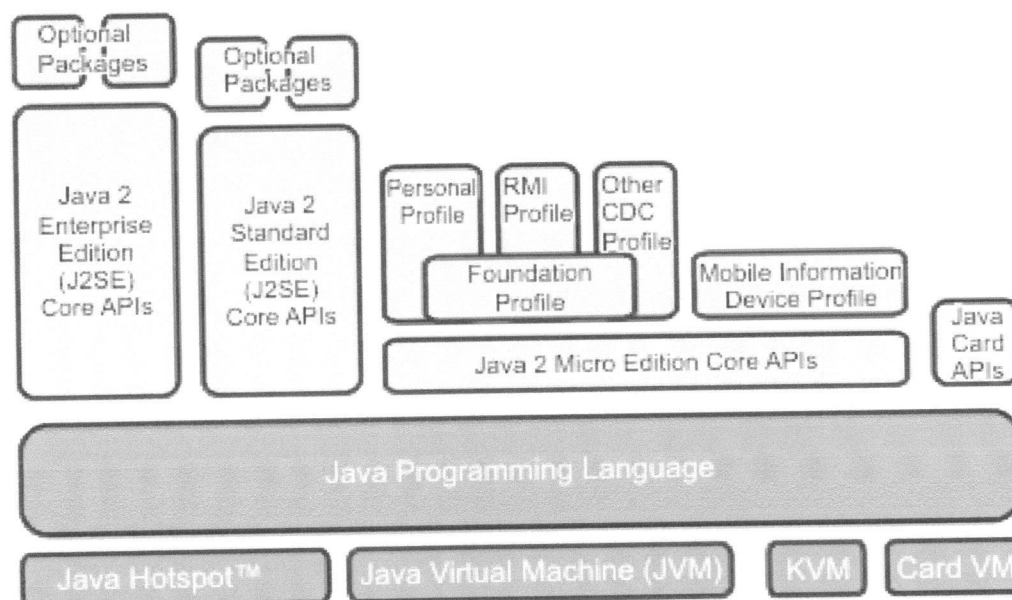
2. Java 2 Enterprise Edition (J2EE)

Kategori ini digunakan untuk menjalankan dan mengembangkan aplikasi-aplikasi Java pada lingkungan enterprise dengan menambah fungsionalitas-fungsionalitas Java semacam EJB (Enterprise Java Bean), Java CORBA, Servlet dan JSP, serta Java XML (Extensible Markup Language).

### 3. Java 2 Micro Edition (J2ME)

Kategori ini digunakan untuk menjalankan dan mengembangkan aplikasi-aplikasi Java pada *handheld devices* atau perangkat-perangkat mobile lainnya semacam *handphone*, Palm, PDA dan Pocket PC.

Pembagian lingkungan kerja pada teknologi Java terlihat pada Gambar 2.1.



**Gambar 2.1** Pembagian lingkungan kerja pada teknologi Java

#### 2.2.2 Java 2 Micro Edition (J2ME)

J2ME (Java 2 Micro Edition) merupakan subset dari J2SE yang ditujukan untuk implementasi pada peralatan implementasi pada peralatan *embedded system* dan *handheld* yang tidak mampu mendukung secara penuh implementasi menggunakan J2SE.

Yang dimaksud dengan *embedded system* sendiri adalah produk-produk dengan komputer kecil berada di dalamnya, namun aplikasi yang bisa

dimanfaatkan dari peralatan tersebut sangatlah spesifik. Hal ini tentu saja berbeda dengan komputer PC yang kita kenal sehari-hari, yang mampu digunakan untuk berbagai aplikasi. Contoh *embedded system* adalah aplikasi-aplikasi yang memanfaatkan mikroprosesor seperti televisi, keamanan gedung dan sebagainya.

J2ME dirancang untuk dapat menjalankan program Java pada perangkat-perangkat semacam *handphone* dan PDA, yang mempunyai karakteristik berbeda dengan sebuah komputer biasa, misal kecilnya jumlah memori pada *handphone* dan PDA.

#### 2.2.2.1 J2ME Configuration

*J2ME Configuration* mendefinisikan lingkungan kerja J2ME *runtime* dan menyediakan *library* standar yang mengimplementasikan fitur standar sebuah perangkat genggam. Lingkungan Kerja yang dimaksud meliputi JVM (*Java Virtual Machine*). J2ME Configuration dikategorikan menjadi dua yaitu [WIC02]:

a. CLDC (Connected Limited Devices Configuration)

Kategori ini umum digunakan untuk aplikasi Java pada *handphone*, oraganizer, atau PDA. CLDC digunakan untuk mengimplementasikan program Java pada perangkat-perangkat keras dengan ukuran memori sangat terbatas, yakni 160-512 Kbytes. Akibatnya fitur-fitur yang kurang penting untuk diimplementasikan dalam *handheld devive* yang bersangkutan dari Java 2 harus dibuang.

b. CDC (Connected Device Configuration)

CDC digunakan untuk peralatan dengan kapasitas memori yang lebih besar dengan koneksi jaringan yang lebih kokoh misalnya pada peralatan internet. Configuration ini diperuntukkan bagi peralatan *wireless* dengan *resource* memori yang lebih besar minimal beberapa Megabyte dengan kemampuan untuk melakukan koneksi internet.

#### 2.2.2.2 J2ME Profile

J2ME Profile menyediakan implementasi tambahan yang sangat spesifik dari sebuah perangkat genggam. J2ME Profile yang dikategorikan menjadi lima yaitu :

a. *MIDP (Mobile Information Device Profile)*

MIDP menyediakan *library-library* Java untuk implementasi dasar antarmuka (GUI), implementasi jaringan (*networking*), database dan *timer*. MIDP dirancang untuk *wireless phone* dan *pager*.

b. *Foundation Profile*

Profile dasar untuk non-GUI *network devices* pada CDC.

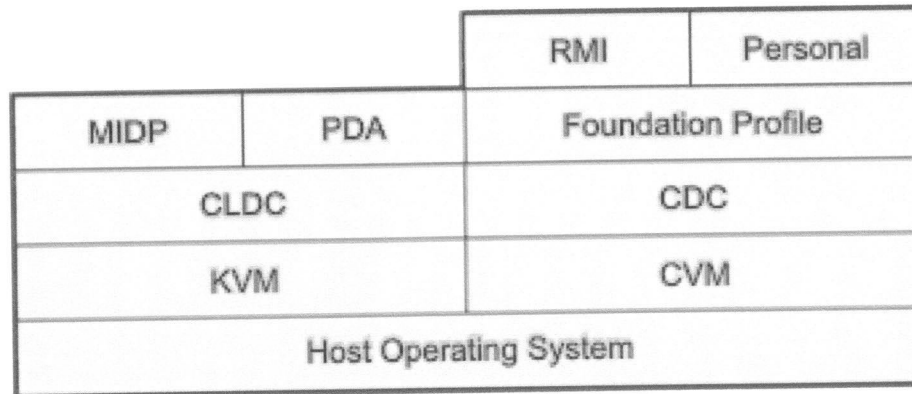
c. *Personal Profile*

d. *PDA Profile*

e. *RMI Profile*

MIDP dirancang khususnya untuk *wireless phone* dan *pager*. MIDP menyediakan *library* Java untuk implementasi dasar antarmuka (GUI), jaringan database dan *timer*. Aplikasi MIDP yang dijalankan pada sebuah perangkat genggam disebut dengan Java MIDP atau MIDlet.

Secara arsitektural, J2ME Profile dapat digambarkan seperti terlihat pada Gambar 2.2.



**Gambar 2.2** J2ME profile

### 2.2.2.3 API (Application Programming Interface)

API paling menentukan kemampuan dari *mobile devices*. Sebagai tambahan untuk API CLDC, terdapat dua komponen utama API yang berpotensi pada platform J2ME :

a. Mobile Information Devices Profile (MIDP) APIs

Sebuah standarisasi dari sejumlah API yang ditentukan oleh Mobile Information Devices Profile Expert Group (MIDPG).

b. OEM-specific APIs

Disediakan oleh OEM untuk mengakses fungsi tertentu yang dapat diberikan pada *handheld devices*. Aplikasi yang menggunakan APIs ini mungkin tidak akan cocok bila digunakan oleh *handheld* lain.

#### 2.2.2.4 MIDlet

Midlet adalah aplikasi yang dibuat menggunakan Java 2 Micro Edition dengan profile MIDP. MIDP dikhususkan untuk digunakan pada handset dengan kemampuan CPU, memori, keyboard, dan layar yang terbatas misalnya pada *handphone*, PDA dan sebagainya.

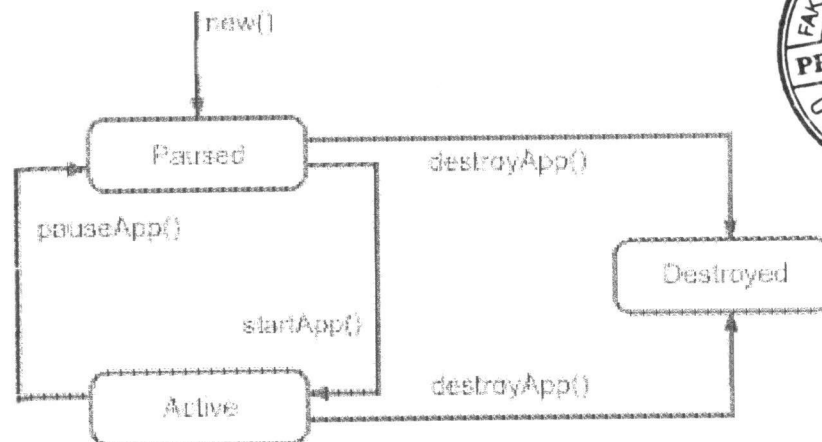
Pengaplikasian API ditentukan oleh interaksi antara aplikasi dan MIDP serta bagaimana MIDlet ini dikontrol. Penanganan eksekusi aplikasi dilakukan oleh JAM (Java Application Manager). JAM akan mengatur instalasi, eksekusi, manajemen versi dan perhitungan MIDlet.

Secara umum terdapat beberapa hal penting dalam membuat sebuah aplikasi MIDlet yaitu *lifecycle* dan *user interface*.

##### 2.2.2.4.1 Life Cycle

*Lifecycle* dari sebuah MIDlet ditangani oleh AMS (Application Management Software). AMS ini adalah sebuah lingkungan tempat siklus dari sebuah MIDlet mampu diciptakan, dijalankan, dihentikan maupun dihilangkan. AMS juga sering dinamakan dengan JAM (Java Application Manager).

MIDlet memiliki beberapa *state* yaitu *pause*, *active*, dan *destroy*. Ketika masing-masing *state* dipanggil, beberapa *method* standar yang bersesuaian dipanggil. *Method-method* ini merupakan bawaan dari J2ME. Jika digambarkan status daur hidup MIDlet dapat dilihat pada Gambar 2.3.



**Gambar 2.3** Daur hidup MIDlet

Dari gambar tersebut dapat dijelaskan sebagai berikut :

1. Ketika MIDlet pertama kali diciptakan dan diinisialisasi, MIDlet berada dalam *state pause*.
2. Apabila terjadi kesalahan selama konstruksi MIDlet, MIDlet akan berpindah ke *state destroy* dan MIDlet batal diciptakan dengan jalan memanggil fungsi standar `destroyApp()`.
3. Selanjutnya ketika MIDlet dijalankan MIDlet akan berada pada *state active*, dalam hal ini fungsi standar yang dipanggil adalah `startApp()`.
4. Akan tetapi, jika ditengah jalan MIDlet dihentikan sementara, MIDlet akan berada dalam *state pause* dengan jalan memanggil fungsi standar `pauseApp()`. Pada *state* ini diperlukan proses *cleanup* terhadap *garbage collector* yang dihasilkan.

#### 2.2.2.4.2 User Interface

*User interface* dari MIDP terdiri atas API-API yang *High Level* dan *Low Level*. *Low Level* API berbasis pada class Canvas, sedangkan *High Level* API berbasis pada *screen*.

Sebuah aplikasi MIDlet akan mengalami beberapa proses sebelum dapat digunakan dalam sebuah perangkat *handphone*, proses tersebut meliputi :

1. *Editing*

Proses bagaimana mulai menulis atau memperbaiki program. Program disimpan dalam ekstensi .java.

2. *Compiling*

Proses dimana program yang dibuat dengan ekstensi .java diubah dalam bentuk *byte code* menjadi file yang berekstensi .class.

3. *Preverify*

Proses penyusunan kembali *byte code* dari file berekstensi .class tadi untuk verifikasi akhir pada CLDC *virtual machine*. Di samping itu diperiksa juga apakah ada fitur dari *virtual machine* yang tidak di-support oleh DLDC.

4. *Packaging*

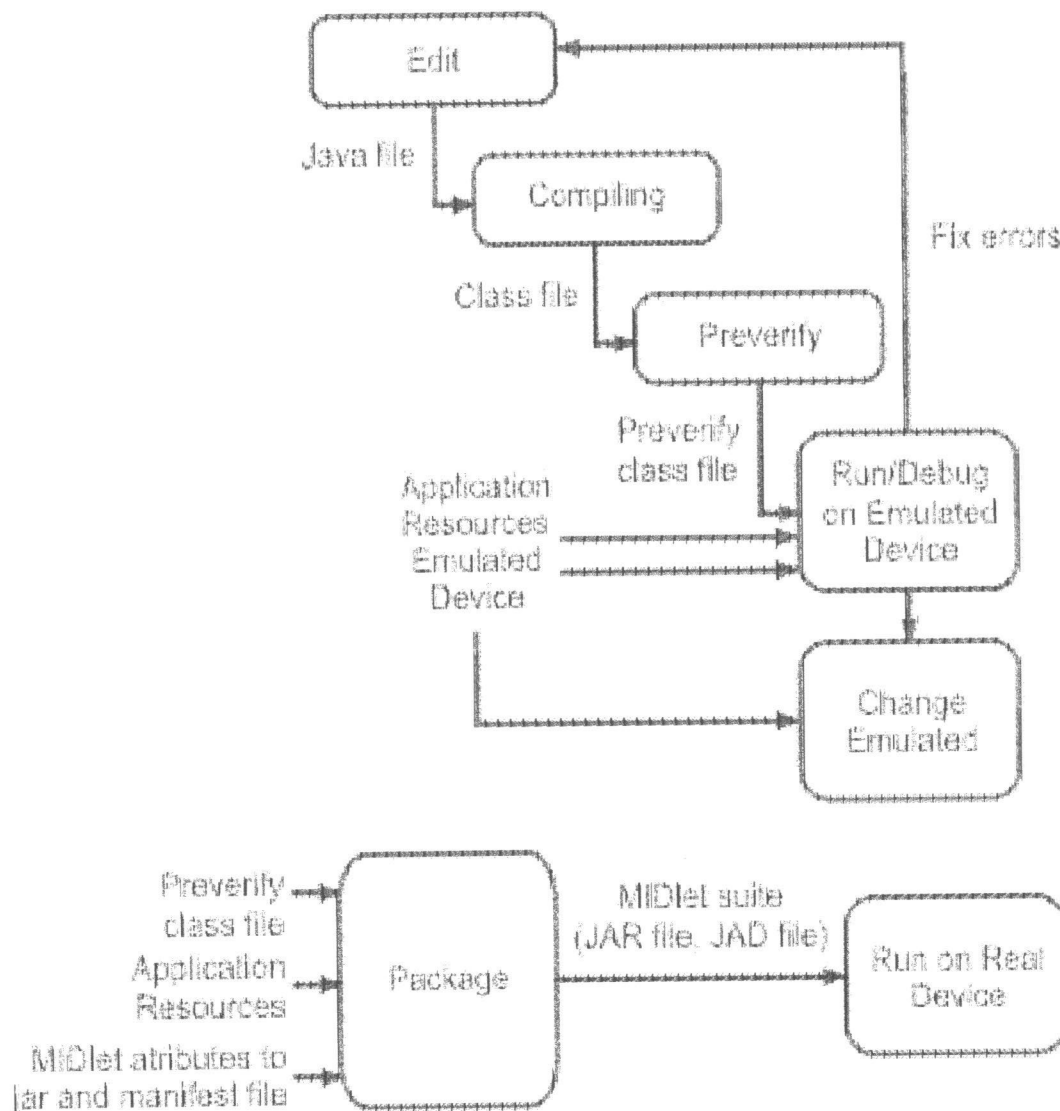
Proses ini kurang lebih adalah pengemasan semua file yang diperlukan termasuk di dalamnya adalah file-file tambahan seperti teks atau gambar ke dalam sebuah file berekstensi .jar.

5. *Run* atau *Debug*

Proses untuk memeriksa hasil kode program dalam emulator.

Untuk proses pembuatan aplikasi MIDlet dapat dilihat pada Gambar 2.4.





**Gambar 2.4** Proses pembuatan aplikasi MIDlet

## 2.2.3 Pemrograman Java pada Perangkat Lunak

### 2.2.3.1 Pemrograman GUI

Fungsi-fungsi untuk pembuatan antarmuka berbasis window (GUI) disediakan oleh MIDP. Fungsi-fungsi ini ditangani oleh paket `javax.microedition.lcdui`. Pemrograman GUI dibagi menjadi dua *level*, yaitu

*highlevel* dan *lowlevel*. Pada *highlevel*, aplikasi MIDlet yang dibuat akan lebih portabel dengan banyak perangkat genggam yang berbeda. Sedangkan pada *lowlevel* akan didapatkan fungsionalitas yang lebih spesifik ke jenis perangkat genggam yang digunakan.

Paket `javax.microedition.lcdui` untuk pembuatan GUI pada MIDlet mempunyai dua obyek yaitu *display* dan *screen*. *Display* menyediakan fungsi untuk manajemen layar, menampilkan *object screen* dan menyediakan informasi tentang properti pada perangkat genggam yang digunakan. Sedangkan *screen* menyediakan fungsi untuk interaksi antara pengguna dengan perangkat genggam.

Obyek *screen* memiliki empat jenis obyek turunan yang umum digunakan, yaitu :

1. Text Box, menyediakan media untuk menerima masukan teks dari keypad pengguna *handphone*.
2. Alert, menyediakan informasi kecil kepada pengguna yang ditampilkan ke layar sebelum kemudian berpindah ke *object screen* lain.
3. List, menyediakan masukan pilihan layar.
4. Form, memungkinkan untuk menampilkan beberapa komponen GUI semacam daftar pilihan serta masukan teks dalam satu layar.

J2ME terdiri atas komponen-komponen sebagai berikut :

1. JVM (Java Virtual Machine)

Komponen ini untuk menjalankan program-program Java pada emulator atau perangkat genggam.

## 2. Java API (Application Programming Interface)

Komponen ini merupakan kumpulan *library* untuk menjalankan dan mengembangkan program-program Java pada perangkat genggam.

## 3. Tools

Komponen ini untuk pengembangan aplikasi Java, semacam emulator Java Phone.

## 2.3 Teori Game

### 2.3.1 Klasifikasi Permainan dan Definisi Permainan Pertempuran

Secara umum permainan dapat dibagi jenisnya berdasarkan kategori-kategori tertentu [ROL03], yaitu “*Skill and Action Games*” dan “*Strategi Games*”. *Skill and Action Games* adalah permainan yang melibatkan kemampuan persepsi dan motorik pemain, sedangkan *Strategy Games* adalah permainan yang melibatkan kemampuan cognitive pemain. Berdasarkan kemampuan persepsi dan motorik pemain yang terlibat, permainan dapat dibagi menjadi enam, yaitu *Fighting Games*, *Combat Games*, *Maze Games*, *Sport Games*, *Paddle Games*, dan *Race Games*.

Berdasarkan pengelompokan di atas, *Combat Games* (permainan pertempuran) diartikan sebagai permainan yang bersifat konfrontasi langsung dengan lingkungan atau dengan pemain lain untuk dapat mencapai kemenangan. Dalam permainan ini biasanya berisikan misi untuk menghancurkan karakter lain sehingga hanya tersisa karakter dari pemain saja.

### 2.3.2 Elemen Game

Di dalam sebuah game terdapat beberapa elemen-elemen yang akan membentuk sebuah game itu sendiri, yaitu [ROL03]:

1. *Rules* (aturan-aturan)

Sebuah game mengambil tempat atau seting di dunia buatan yang diatur oleh aturan-aturan (*rules*). *Rules* inilah yang menentukan aksi dan gerakan pemain dalam sebuah game.

2. *Victory condition* (kondisi menang) dan *Lose condition* (kondisi kalah)

Karena adanya kondisi kemenangan, maka game juga memiliki kondisi kekalahan (*lose condition*). Kadang kondisi kekalahan ini bersifat implisit, yaitu jika pemain tidak mendapatkan kondisi kemenangan maka dinyatakan kalah. Sedangkan kondisi yang eksplisit misalnya tokoh telah mati karena kehabisan nyawa.

3. *Setting* (seting)

Sebuah game mengambil tempat pada suatu dunia atau seting tertentu. Contohnya pada permainan bola yang bersejing pada sebuah lapangan dengan batasan-batasan tertentu.

4. *Interaction model* (model interaksi)

Yaitu cara pemain berinteraksi dengan game dan melakukan aksi untuk menghadapi tantangan dari game tersebut.

5. *Perspective* (sudut pandang)

Yaitu menjelaskan bagaimana pemain melihat *gameworld* dari suatu game pada layar.

6. *Role* (peran)

Yaitu tokoh yang dimainkan oleh pemain dalam suatu game. Dengan adanya peran ini maka pemain akan lebih mudah untuk memahami tujuan apa yang sebenarnya ingin dicapai dan aturan apa yang dimainkan.

7. *Mode*

Pemain selalu ingin mencapai atau menyelesaikan hal yang sama dengan cara yang sama pula. Namun ada juga game yang memiliki mode yang nyata, yaitu dimana *gameplay*-nya berubah dari satu mode ke mode lainnya.

8. *Structure* (struktur)

Hubungan antara mode dan aturan menentukan kapan dan mengapa game berubah secara bersama-sama membentuk struktur permainan.

9. *Realism* (realisme)

Game menggambarkan sebuah dunia, bahkan mungkin dunia yang khayal. Sebuah game yang menerapkan akal sehat dan logika pada aturan permainannya dapat dikatakan sebagai game yang realistis.

10. *Story* (cerita)

Beberapa game memiliki alur cerita yang linear atau hanya satu jalan cerita dan bersifat non-interaktif, namun ada juga yang bersifat interaktif dimana cerita akan berbeda atau bercabang tergantung pada pilihan atau tindakan yang diambil pemain. Sehingga cerita akhir (*ending*) yang didapat akan berbeda-beda pula, atau biasa disebut dengan *multiple ending*.

### 2.3.3 Teori Dalam Pembuatan Program Permainan

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

a. Tentukan tipe permainan yang ingin dibuat

Penentuan ini sebagai dasar mulai bekerja sampai mendapatkan ide yang bagus untuk dibuat program permainan.

b. Definisikan model permainan dan tujuannya

Pada tahap ini sebaiknya model permainannya ditulis secara jelas sehingga jika terjadi perubahan maka permainannya tetap konsisten dan tidak membingungkan.

c. Definiskan secara jelas *Game Worlds*-nya

*Game worlds* adalah elemen-elemen utama yang terdapat dalam suatu program permainan yang terdiri dari :

(i). *Game board*

*Game board* merupakan bentuk tampilan, latar belakang, dan lainnya.

(ii). Instruksi untuk permainan

Instruksi untuk permainan harus jelas supaya tidak membingungkan dan pemain dapat menentukan strategi dari permainannya.

(iii). Informasi untuk pemain

Informasi ini penting ditampilkan dalam program permainan ketika sedang berjalan.

(iv). Penghargaan

Penghargaan digunakan sebagai rangsangan untuk pemain ketika menyelesaikan *level* tertentu dan mendorong untuk melanjutkan ke *level* berikutnya.

(v). Variasi

Variasi digunakan agar pemain tidak cepat merasa bosan tetapi variasi tidak boleh berlebihan karena dapat membuat alur cerita menjadi tidak konsisten dan membingungkan.

(vi). Tingkat Kesulitan

Tingkat kesulitan dari program permainan akan membuat pemain menjadi bergairah jika melewati tingkat kesulitan yang diberikan.

d. Pastikan permainan bisa dimainkan

Permainan yang tidak mungkin diselesaikan akan membuat program permainan tidak dapat dimainkan.

e. Rancang program sebaik mungkin

Gunakan teknik pemrograman yang sesuai dalam membuatnya dan buat program yang mudah dimodifikasi dan dikembangkan.

f. Pengujian program

Program yang selesai harus diuji. Pertama, program diuji oleh perancangnya sendiri (*alpha test*) untuk menentukan kesalahan logika yang mungkin terjadi. Kedua, program diuji oleh orang lain (*beta test*) untuk mengetahui kesalahan logika yang tidak terlihat oleh perancangnya sehingga memberi masukan, saran atau ide dalam pengembangan program permainan.

## **BAB III**

### **ANALISIS KEBUTUHAN**

#### **3.1 Metode Analisis**

Metode analisis yang digunakan adalah metode analisis dengan pendekatan terstruktur (*structural approach*) yang lengkap dengan alat dan teknik (*techniques*) yang disesuaikan dengan kebutuhan. Arsitektur berusaha didefinisikan dengan sebaik mungkin agar pengembangan perangkat lunak dapat berjalan dengan terstruktur dan jelas meskipun mengalami peningkatan kompleksitas yang signifikan.

Pada tahap analisis ini, digunakan suatu alat untuk melakukan pemodelan agar pengembangan perangkat lunak dapat memenuhi semua kebutuhan pengguna dengan lengkap dan tepat. Pemodelan yang dilakukan menggunakan notasi-notasi UML (*Unified Modelling Language*) yang merupakan standar dalam dunia industri perangkat lunak untuk melakukan visualisasi, perancangan dan pendokumentasian.

#### **3.2 Hasil Analisis**

##### **3.2.1 Analisis Masukan**

yang akan dibangun nantinya diusahakan agar dapat menangani hal-hal yang sudah lazim terdapat pada sebuah *game* pertempuran pesawat. Sehingga terdapat beberapa komponen masukan utama yang diperlukan, yaitu :



1. Masukan pada seleksi menu

Pemain dapat memilih beberapa menu yang disediakan, antara lain : menu *new game*, menu *high score*, menu *setting*, dan menu *help*. Menu *new game* akan menampilkan *game* pada *level 1*. Menu *high score* akan menampilkan daftar score tertinggi yang pernah dicapai, dan disertakan nama pemainnya. Menu *setting* akan menampilkan setting *sound*. Menu *help* akan menampilkan halaman bantuan.

2. Masukan penentuan setting

Pemain dapat melakukan pengaturan terhadap *sound*, dimana pemain dapat mematikan atau menghidupkan suara selama permainan berlangsung.

3. Maukan kendali pesawat

Pemain dapat mengarahkan pesawat sesuai dengan arah yang diinputkan oleh pemain.

4. Masukan untuk menembak

Pemain dapat melakukan penyerangan terhadap musuh sesuai dengan senjata yang digunakan.

### 3.2.2 Analisis Proses

Beberapa proses yang dapat dilakukan oleh adalah sebagai berikut :

1. Proses menampilkan *level*

Pada proses ini, *game* akan menampilkan *level* tertentu dengan elemen-elemen pembangunnya, seperti gambar *background*, maupun penampilan pesawat musuh.

2. Proses menampilkan pesawat

Pada proses ini, *game* akan menampilkan pesawat pemain yang akan ditampilkan pada layar.

3. Proses pergerakan pesawat

Proses ini dilakukan untuk menggerakkan pesawat pemain, sesuai dengan data yang diinputkan oleh pemain.

4. Proses menggerakkan gambar background

Proses ini dilakukan untuk menggerakkan gambar *background* untuk menimbulkan kesan pesawat berjalan.

5. Proses menyerang

Proses ini dilakukan untuk menyerang musuh maupun pemain, dimana *health status* musuh atau pemain akan berkurang.

6. Proses ledakan

Pada proses ini dilakukan apabila *health status* pemain atau musuh habis, yang dikarenakan oleh tembakan ataupun tabrakan.

7. Proses pergerakan musuh

Proses ini dilakukan untuk memilih karakter musuh yang akan bergerak, menentukan aksinya dengan karakter pemain.

8. Proses penghitungan dan penyimpanan nilai

Proses ini dilakukan untuk menyimpan nama pemain yang masuk daftar *high score*.

### 3.2.3 Analisis Keluaran

Keluaran pada sistem ini adalah status permainan pada suatu kondisi tertentu ketika permainan sedang berlangsung.

### 3.2.4 Analisis User Interface

Suatu sistem yang memiliki *interface* menarik dan informatif pada tampilan saat bermain, seperti :

1. Adanya indikasi untuk informasi *life bar* dan jumlah nyawa.
2. Adanya indikasi untuk informasi jumlah bomb.
3. Adanya indikasi untuk informasi jumlah *score*.

## **BAB IV**

### **PERANCANGAN**

#### **4.1 Metode Perancangan**

Metode perancangan perangkat lunak yang digunakan dalam pembuatan implementasi J2ME pada *game* pertempuran pesawat “Stealth” ini menggunakan *Unified Modeling Language* (UML). Penggunaan UML (*Unified Modelling Language*) untuk melakukan analisis berorientasi obyek lebih sesuai dan lebih tepat daripada penggunaan model perancangan lainnya seperti DFD (*Data Flow Diagram*). Ketika membangun model dengan menggunakan UML, tidak hanya diciptakan melalui satu model tunggal. Perlu didekati melalui sekumpulan pandangan model yang hampir independent, karena tidak ada satu pandangan tunggal yang memadai untuk sistem yang kompleks.

#### **4.2 Hasil Perancangan**

##### **4.2.1 Rancangan Umum Aplikasi**

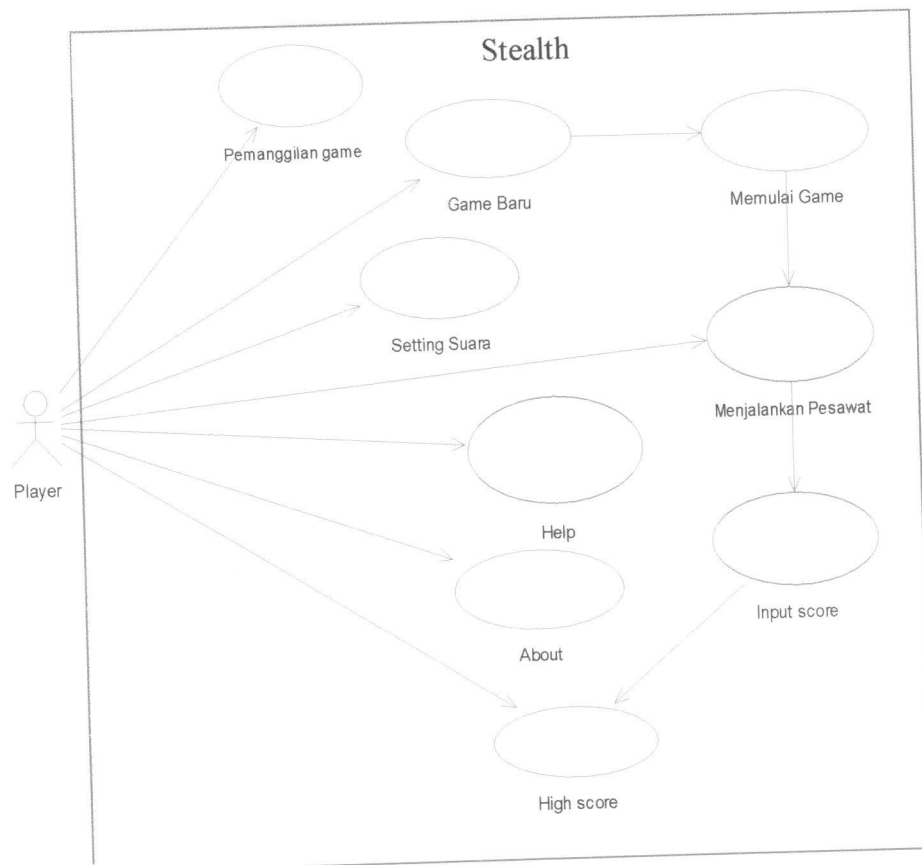
UML memiliki beberapa konsep dasar yang diabstraksikan dalam bentuk *structural classification*, *dynamic behavior*, dan *model management*. Hal terpenting dalam penggunaan UML adalah pembuatan diagram yang sesuai dengan analisis dan pengembangan sistem. Dalam tahap perancangan ini, dibatasi pada pembuatan 4 (empat) diagram saja, yaitu *use case diagram*, *class diagram*, *sequence diagram*, dan *activity diagram*.

#### 4.2.1.1 Use Case Diagram

*Use case diagram* berisi gambaran fungsionalitas yang diharapkan dari sebuah sistem dengan fokus penekanan pada apa yang dilakukan oleh sistem, bukan bagaimana sistem melakukan sesuatu. Dalam *use case diagram*, ada dua hal yang saling berhubungan, yaitu *actor* dan *use case*.

Dalam perancangan sistem dengan diagram *use-case* ini, hanya terdapat satu *actor* saja, yaitu pemain (*player*). Didalam diagram *use-case*, digambarkan kebutuhan sistem dan interaksi antara *actor* dengan sistem.

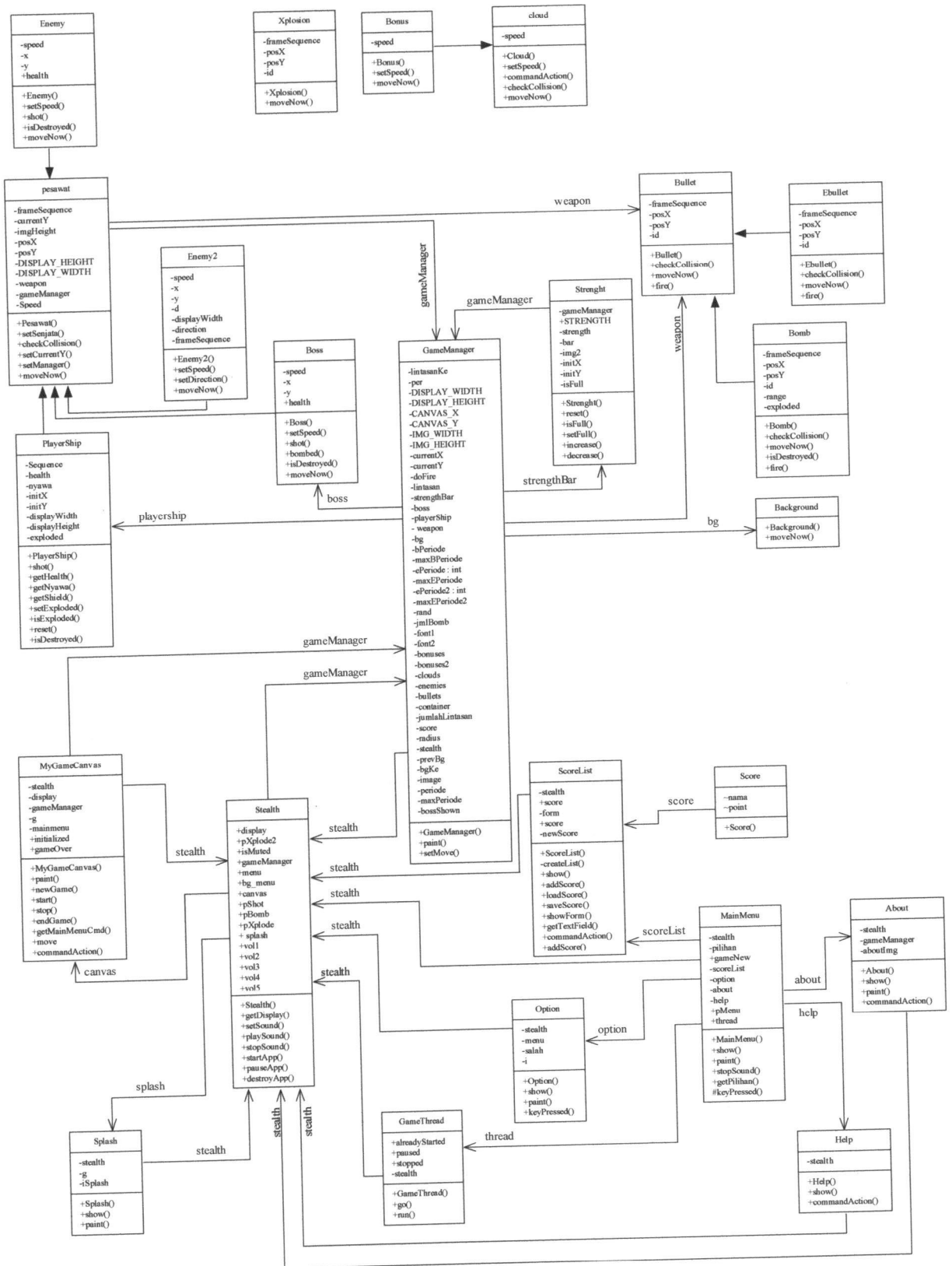
Hubungan antara sistem dan pemain ditunjukkan oleh gambar 4.1.



**Gambar 4.1** Use Case Diagram

#### 4.2.1.2 Class Diagram

Aplikasi yang menggunakan perancangan berorientasi obyek dapat diilustrasikan dalam struktur kelas-kelas dan hubungan antar kelas yang ada. Dalam UML digunakan notasi *class diagram* untuk menggambarkan hal tersebut. Lebih jelasnya dapat dilihat pada gambar 4.2.

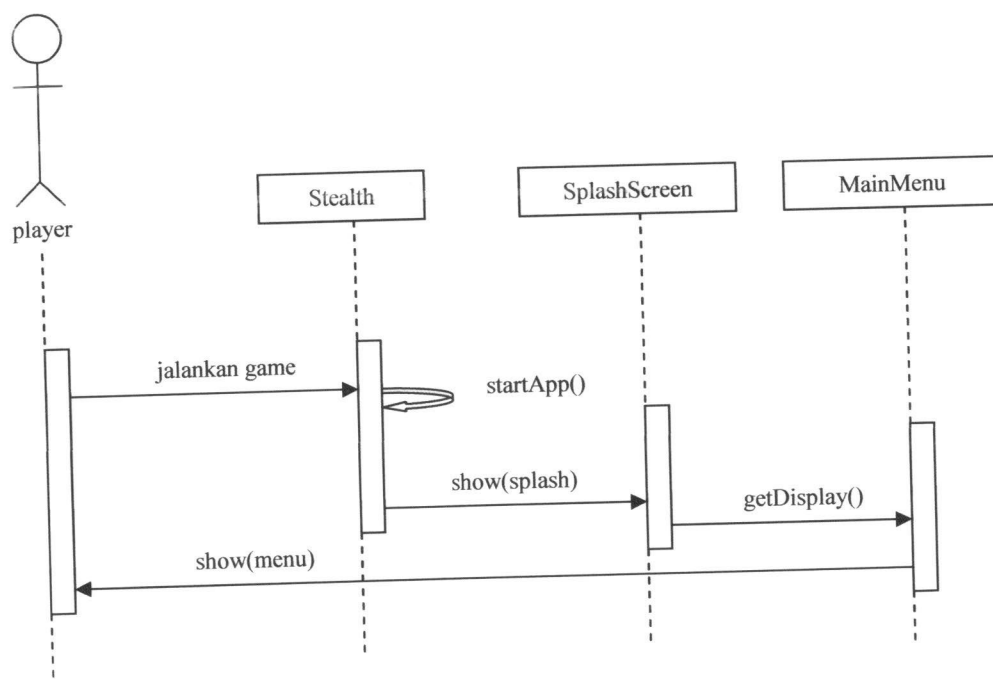


Gambar 4.2 Class Diagram

### 4.2.1.3 Sequence Diagram

Diagram sekuen menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa message yang digambarkan terhadap waktu. Diagram sekuen terdiri atas dimensi vertical (waktu) dan dimensi horizontal (objek-objek yang terkait).

Diagram sekuen digunakan untuk memodelkan skenario penggunaan. Skenario penggunaan adalah barisan kejadian yang terjadi selama satu eksekusi sistem. Dalam perancangan sistem terdapat sembilan diagram sekuen, yaitu panggil *game*, *game* baru, setting suara, help, about, memulai *game*, menjalankan pesawat, *input score*, dan *high score*. Diagram sekuen pemanggilan *game* ditunjukkan oleh gambar 4.3.

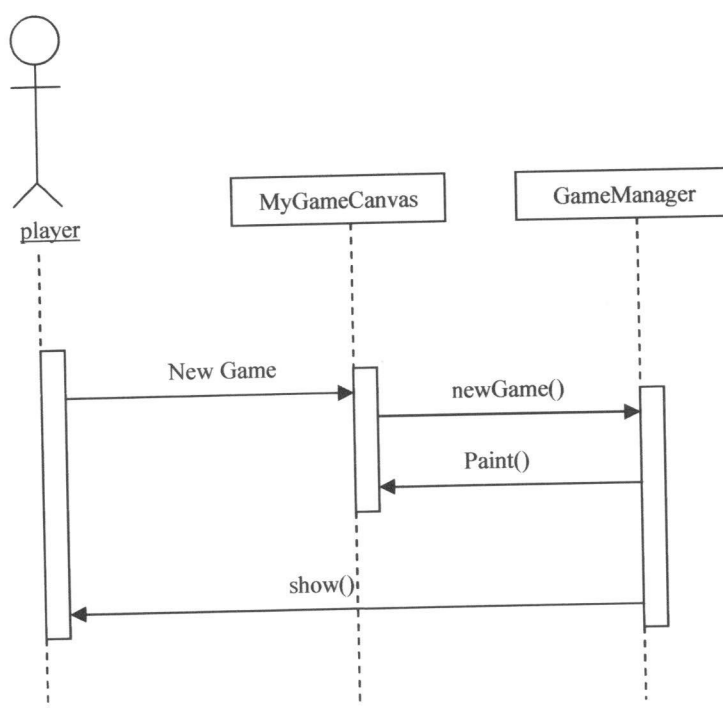


**Gambar 4.3** Diagram sekuen pemanggilan *game*



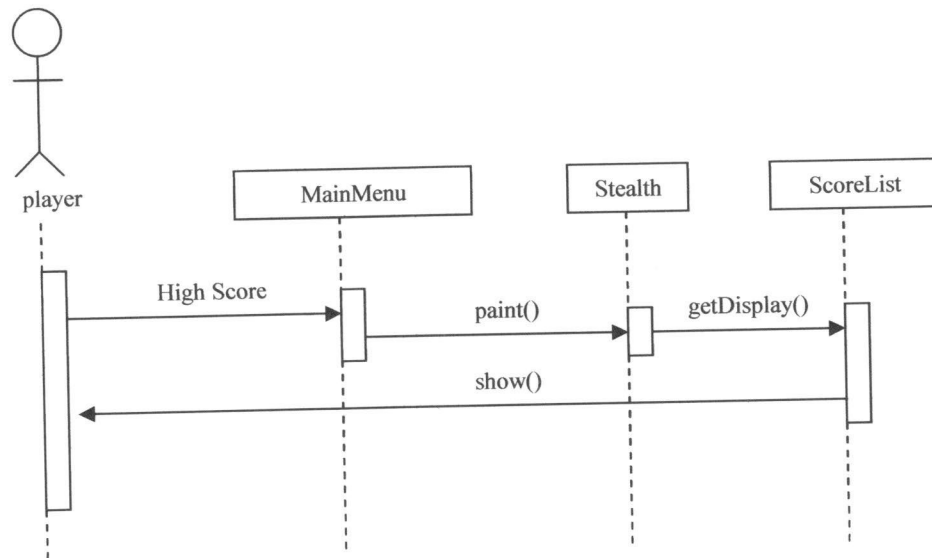
Ketika pemain menjalankan *game* pertama kali, maka kelas yang pertama kali dipanggil adalah kelas yang mewariskan Midlet. Kemudian akan dipanggil kelas Splash untuk menampilkan *splash screen* sebagai intro *game* selama waktu yang ditentukan atau jika ada masukan dari pemain. Kemudian akan dipanggil kelas MainMenu untuk memunculkan menu-menu permainan yang terdiri dari *New Game*, *High Score*, *Option*, dan *Help*.

Apabila pemain memilih New Game, maka permainan akan dimulai dari *level 1*. Kelas menu akan memanggil kelas yang mengatur pemanggilan *level 1*, yaitu kelas MyGameCanvas. Di dalam kelas MyGameCanvas berisi aturan pada *level 1* yang kemudian akan diterapkan pada kelas GameManager, seperti jumlah musuh, jumlah bonus, perputaran background, maupun kemampuan boss. Diagram sekuen *game* baru ditunjukkan oleh gambar 4.4.



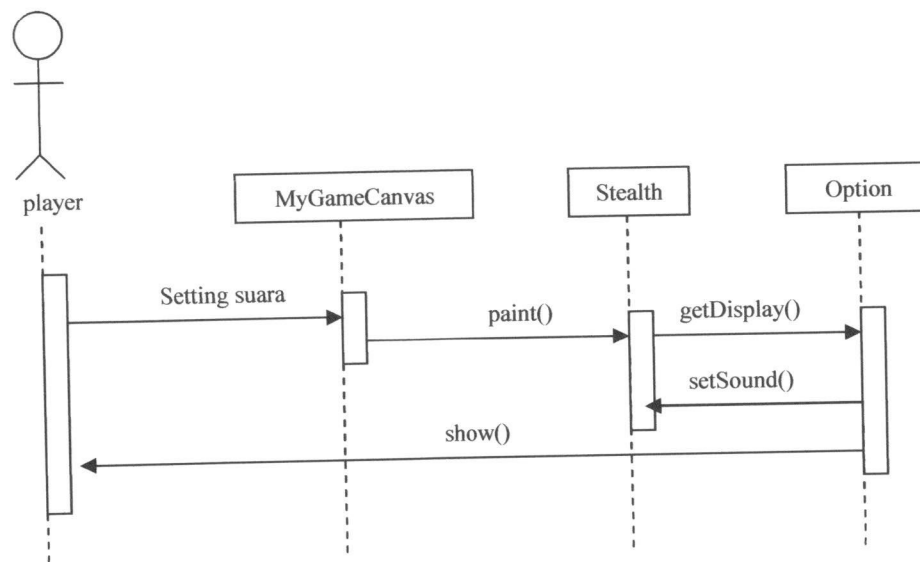
**Gambar 4.4** Diagram sekuen *game* baru

Diagram sekuen *high score*, yang berfungsi untuk menampilkan isi daftar pemain yang pernah bermain game ini. Dengan ini, pemain akan mengetahui berapa catatan *score* tertinggi yang pernah dicatatkan pemain. Diagram sekuen yang mendeskripsikan tampilan *high score* ditunjukkan oleh gambar 4.5.



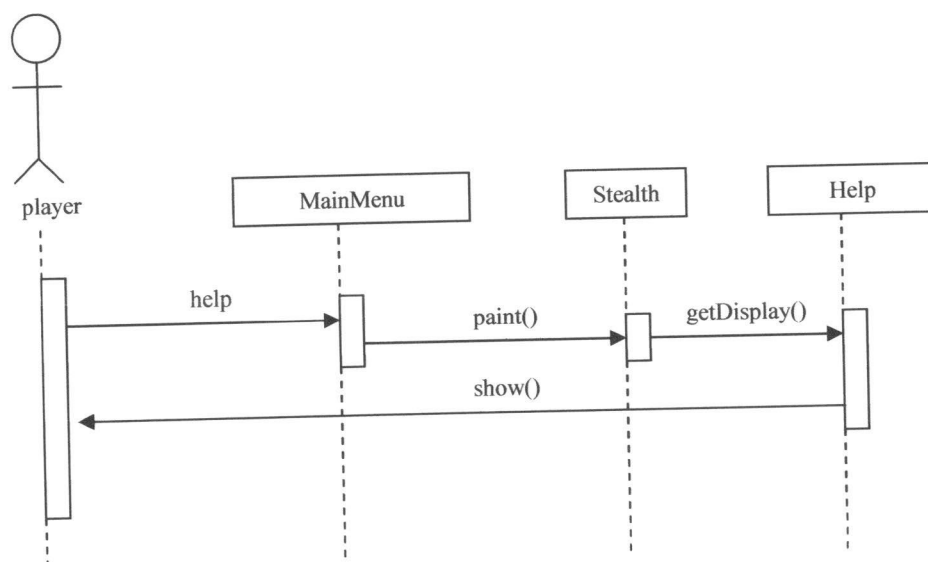
**Gambar 4.5** Diagram sekuen *High Score*

Diagram sekuen setting suara, yang berfungsi untuk mengaktifkan atau tidak mengaktifkan suara. Masukan dari pemain akan diatur oleh *method* `setSound()`. Diagram sekuen yang mendeskripsikan option *game* ditunjukkan oleh gambar 4.6.



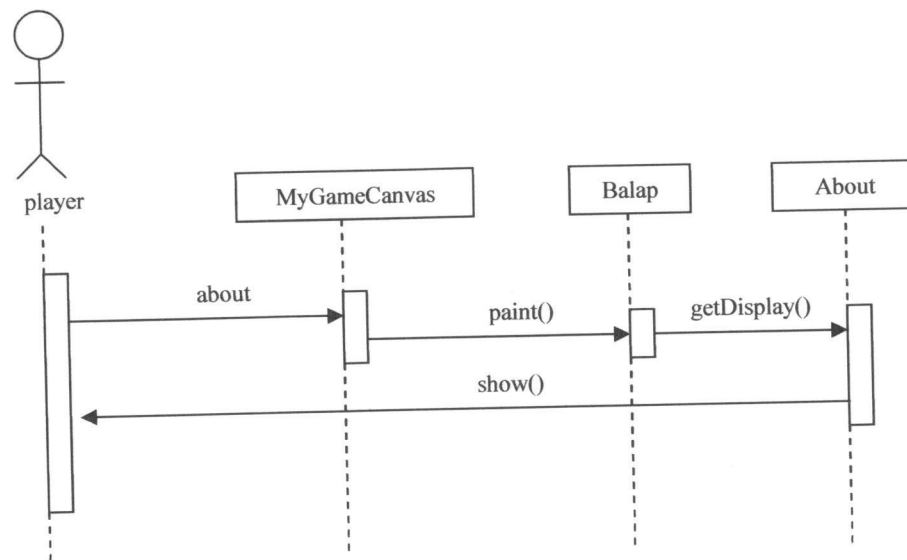
**Gambar 4.6** Diagram sekuen setting *suara*

Diagram sekuen help, yang berfungsi untuk menampilkan informasi cara bermain. Dengan ini, pemain akan mengetahui cara bermain. Diagram sekuen yang mendeskripsikan bantuan ditunjukkan oleh gambar 4.7.



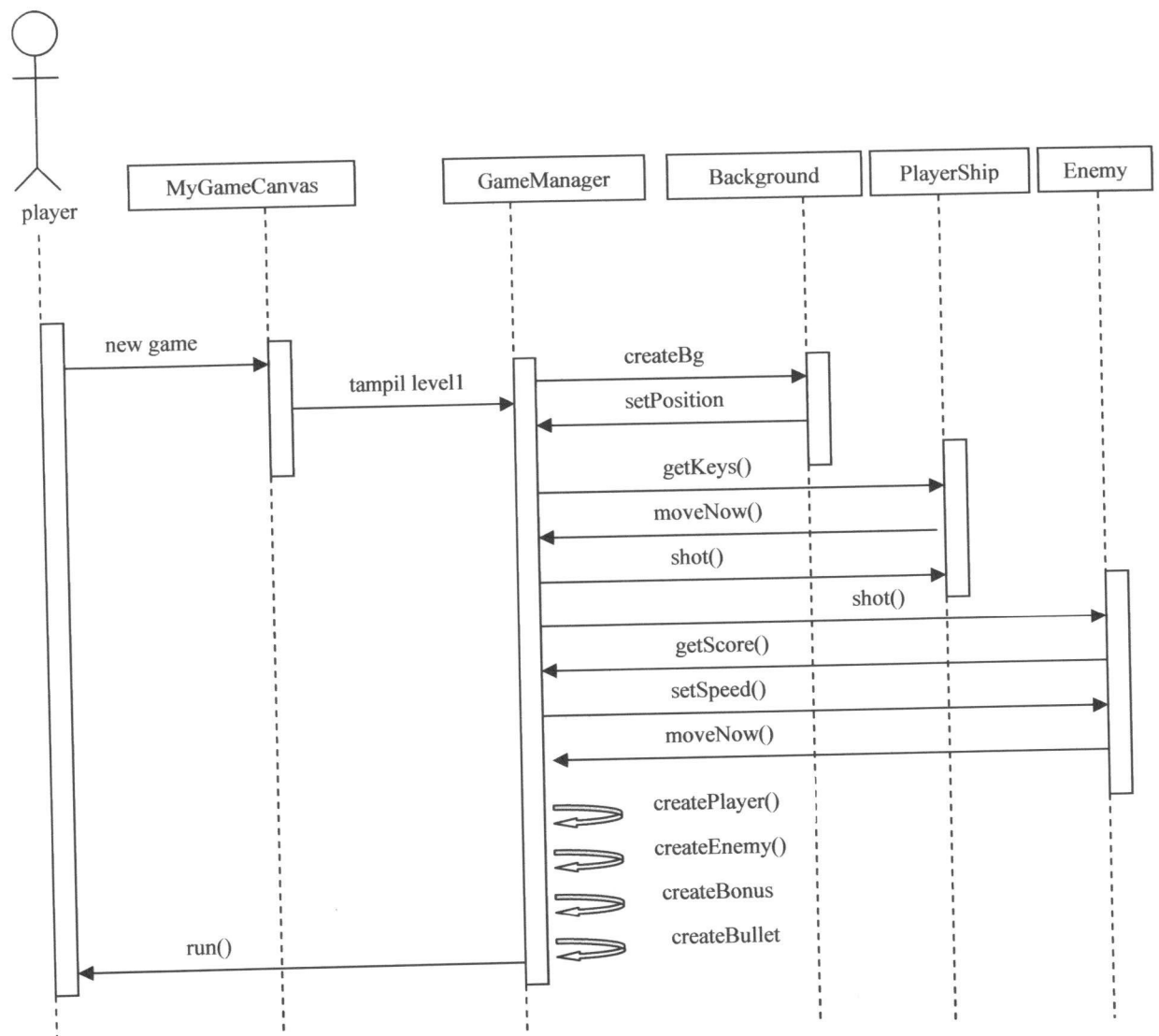
**Gambar 4.7** Diagram sekuen *help*

Diagram sekuen *about*, yang berfungsi untuk melihat informasi tentang pembuat game ini. Diagram sekuen yang mendeskripsikan *about game* ditunjukkan oleh gambar 4.8.



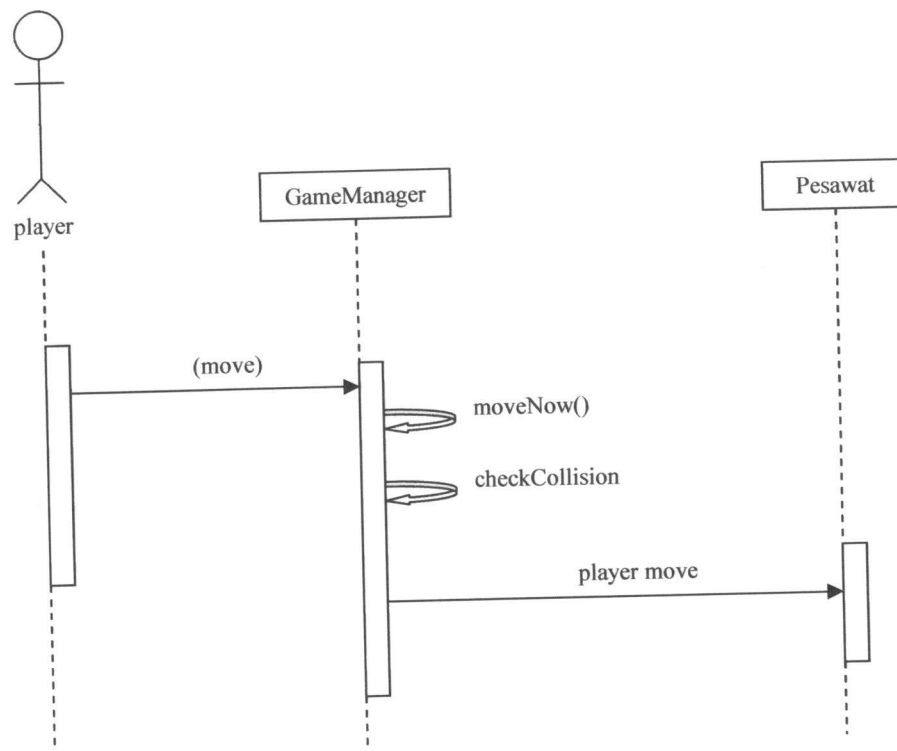
**Gambar 4.8** Diagram sekuen *about*

Diagram sekuen memulai *game*, yang berfungsi untuk menampilkan semua karakter yang dipanggil dalam game ini khususnya *level 1*. Dengan ini, pemain akan mengetahui tampilan awal saat game akan dimainkan. Diagram sekuen yang mendeskripsikan memulai game ditunjukkan oleh gambar 4.9.



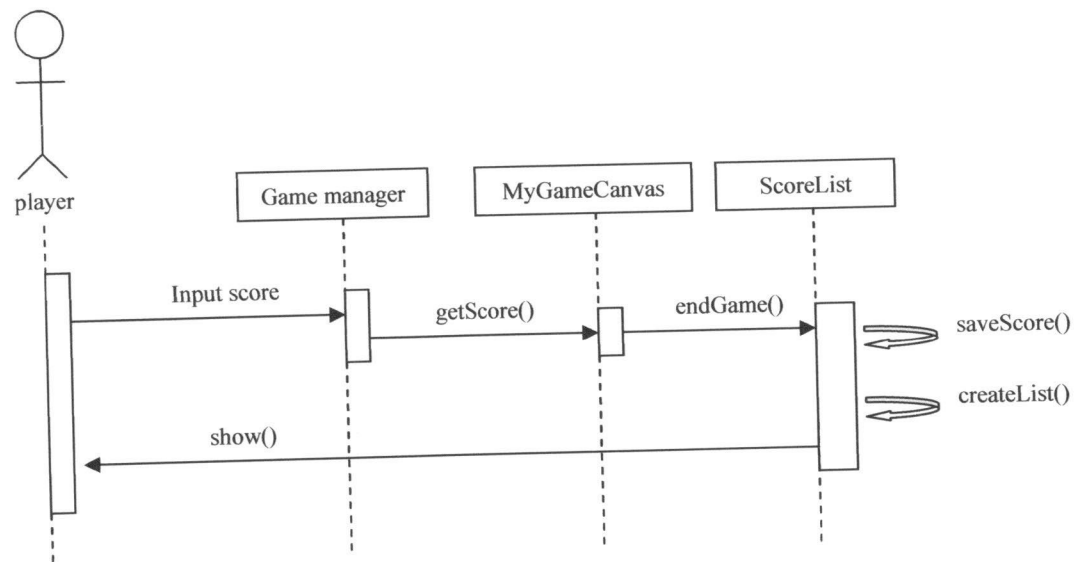
**Gambar 4.9** Diagram sekuen memulai *game*

Diagram sekuen menjalankan pesawat pemain, yang berfungsi untuk menggerakkan pesawat dan juga melakukan pengecekan strength bar. Dengan ini, pemain dapat menjalankan pesawat maju, mundur, kekanan ataupun kekiri sesuai keinginan. Diagram sekuen yang mendeskripsikan menjalankan pesawat ditunjukkan oleh gambar 4.10.



**Gambar 4.10** Diagram sekuen menjalankan pesawat

Diagram sekuen *input score*, yang berfungsi untuk menginputkan *score* pada *form high score* setelah pemain menyelesaikan game atau pemain kalah. Pada proses ini pemain dapat menginputkan nama, dan nama tersebut akan terpampang pada *form high score*. Diagram sekuen yang mendeskripsikan *input score* ditunjukkan oleh gambar 4.11.

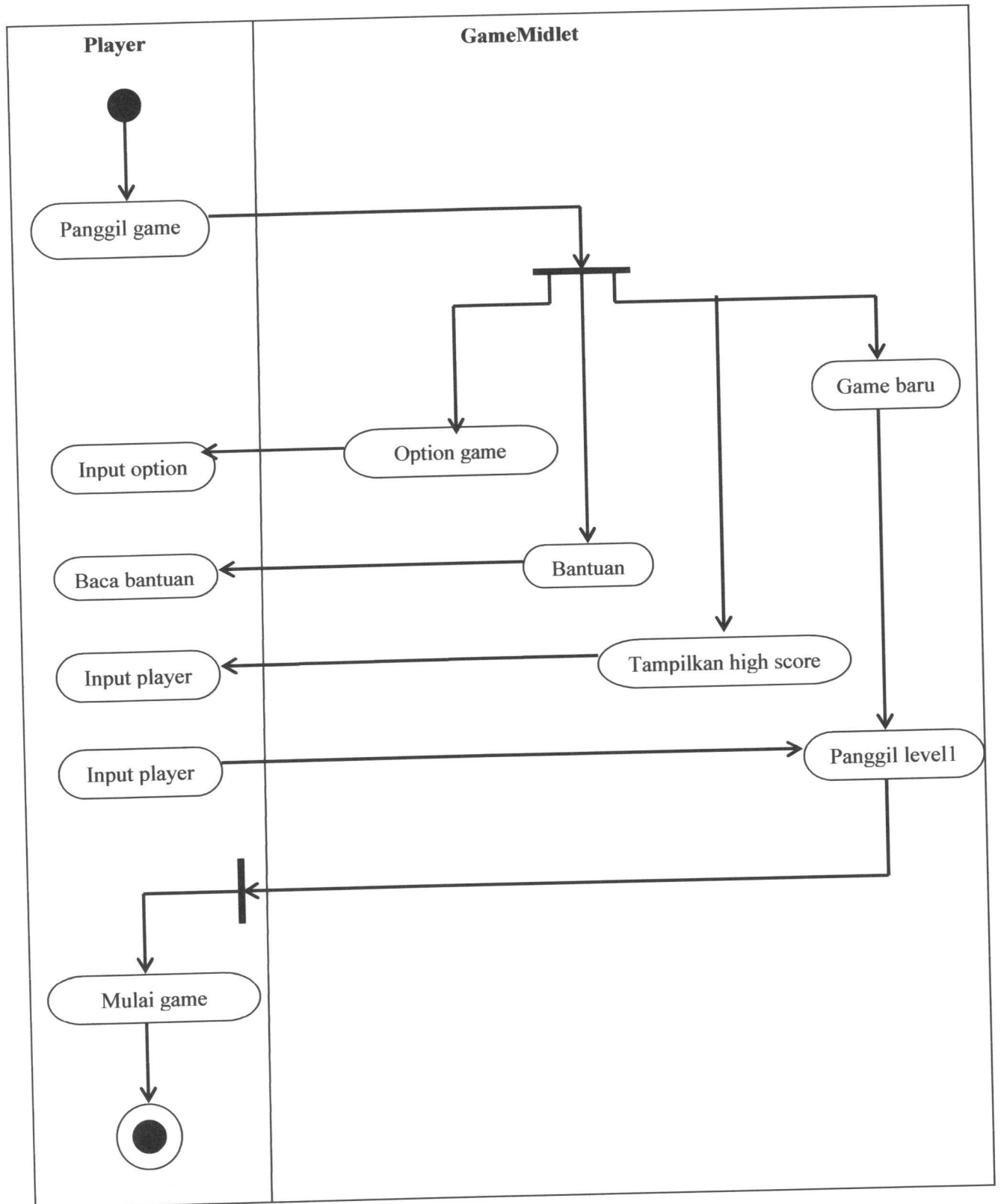


**Gambar 4.11** Diagram sekuen *input score*

#### 4.2.1.4 Diagram Aktivitas

Diagram aktivitas mendeskripsikan aksi-aksi dan hasilnya. Diagram aktivitas memiliki tiga elemen, yaitu *activity state*, transisi, objek.

Dalam perancangan *game* ini, diagram aktivitas digunakan untuk menggambarkan bagaimana objek-objek di dalam *game* ini bekerja. Aktivitas-aktivitas dalam *game* ini ditunjukkan oleh gambar 4.12.



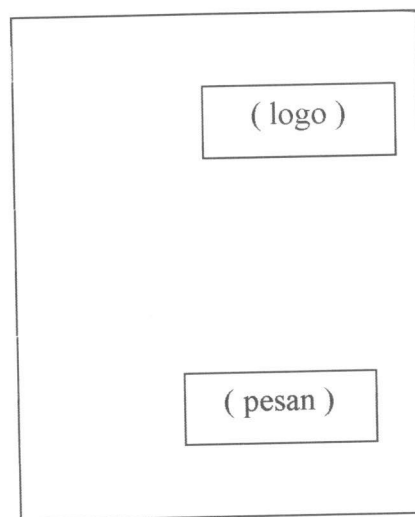
Gambar 4.12 Diagram aktivitas



### 4.3 Rancangan Interface

#### 4.3.1 Rancangan Antarmuka Splash Screen

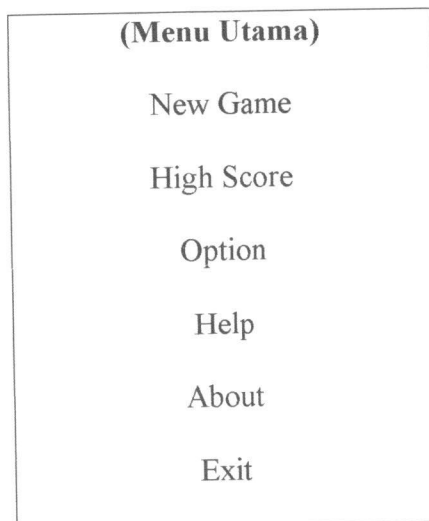
Saat game dijalankan, maka secara otomatis akan menampilkan *splash screen* sebagai intro. Intro game ini digunakan sebagai pengantar menuju interface main menu. Interface *splash screen* seperti ditunjukkan pada gambar 4.13.



Gambar 4.13 Rancangan antarmuka *splash screen*

#### 4.3.2 Rancangan Antarmuka Menu Utama

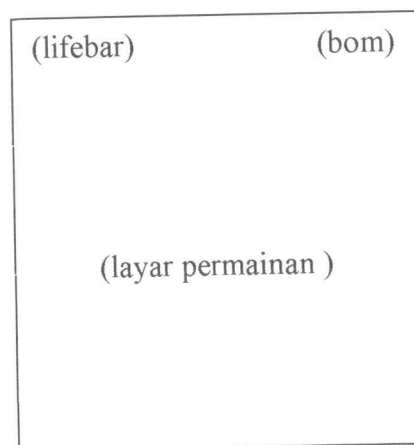
Antarmuka menu utama mempunyai beberapa menu, yaitu *new game* digunakan untuk memulai game baru, *high score* digunakan untuk melihat *record score* tertinggi dari pemain, *option* digunakan untuk menyetting suara *game*, *help* digunakan untuk memberi tahu cara bermain dalam *game* ini, *about* digunakan untuk melihat versi *game*, *exit* digunakan untuk keluar dari game. Rancangan antarmuka menu utamanya seperti yang ditunjukkan pada gambar 4.14.



**Gambar 4.14** Rancangan antarmuka menu utama

#### 4.3.3 Rancangan antarmuka Permainan

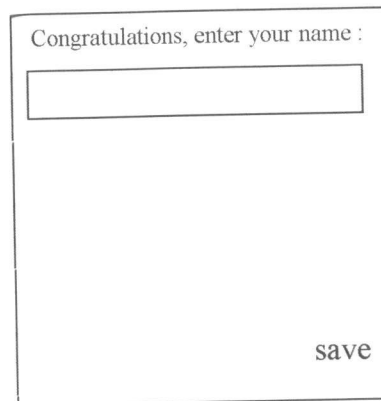
Antarmuka permainan adalah tampilan ketika permainan sedang berlangsung. Rancangan interface permainan dibagi menjadi dua bagian, yaitu *gameManager* dan *toolbar*. *GameManager* berisi tampilan *level* permainan, dan *toolbar* berisi informasi permainan, yaitu jumlah nyawa pemain, *score*, dan jumlah bom. Rancangan antarmuka permainan ditunjukkan oleh gambar 4.15.



**Gambar 4.15** Rancangan antarmuka permainan

#### 4.3.4 Rancangan Antarmuka Masukan Nama

Antarmuka masukan nama digunakan untuk menginputkan nama pemain untuk ditampilkan pada *form high score*. Rancangan antarmuka masukan nama ditunjukkan oleh gambar 4.16.

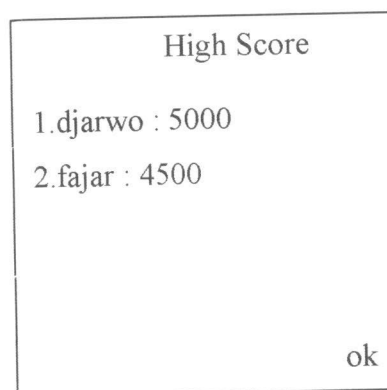


The image shows a rectangular window with a thin border. At the top, it contains the text "Congratulations, enter your name :". Below this text is a horizontal rectangular input field. In the bottom right corner of the window, there is a button labeled "save".

**Gambar 4.16 Rancangan antarmuka masukan nama**

#### 4.3.5 Rancangan Antarmuka *High Score*

Antarmuka *high score* digunakan pemain untuk melihat catatan *score* tertinggi yang pernah dibuat oleh pemain. Rancangan antarmuka *high score* ditunjukkan oleh gambar 4.17.

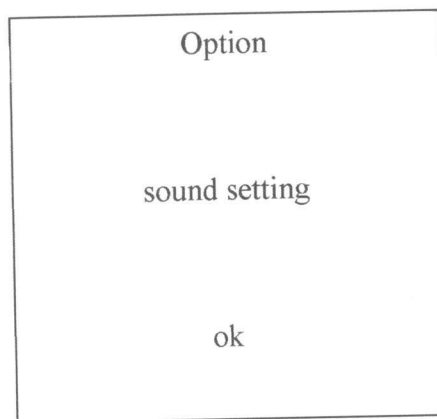


The image shows a rectangular window with a thin border. At the top center, it contains the text "High Score". Below this, there is a list of two entries: "1.djarwo : 5000" and "2.fajar : 4500". In the bottom right corner of the window, there is a button labeled "ok".

**Gambar 4.17 Rancangan antarmuka *high score***

#### 4.3.6 Rancangan Antarmuka *Option*

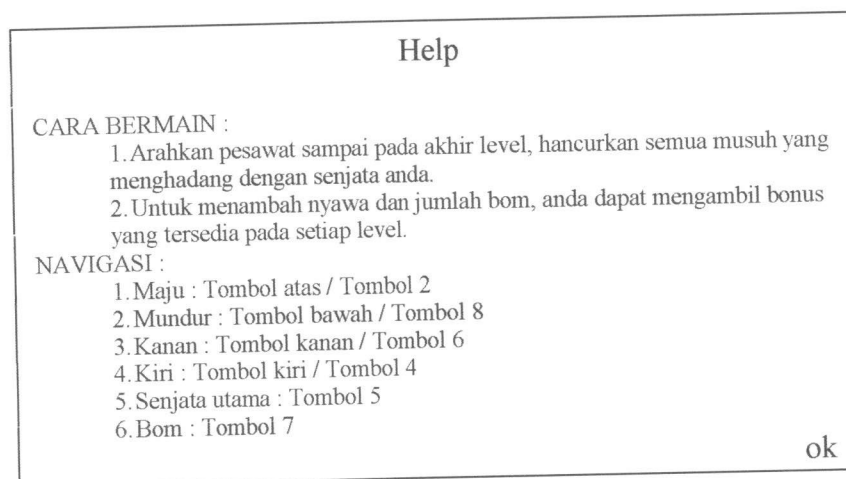
Antarmuka *option* digunakan untuk menyalakan atau mematikan suara. Rancangan antarmuka *option* ditunjukkan oleh gambar 4.18.



**Gambar 4.18** Rancangan antarmuka *option*

#### 4.3.7 Rancangan Antarmuka *Help*

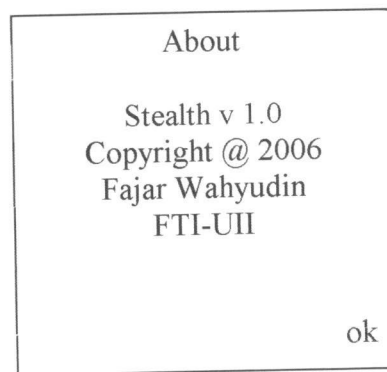
Antarmuka *help* digunakan untuk menampilkan informasi tentang cara bermain, kontrol, dan bonus. Rancangan antarmuka *help* ditunjukkan oleh gambar 4.19.



**Gambar 4.19** Rancangan antarmuka form *help*

#### 4.3.8 Rancangan Antarmuka *About*

Antarmuka *about* digunakan untuk menampilkan informasi tentang versi *game*. Rancangan antarmuka *about* ditunjukkan oleh gambar 4.20.



**Gambar 4.20** Rancangan antarmuka *about*

## **BAB V**

### **IMPLEMENTASI**

#### **5.1 Batasan Implementasi**

Untuk dapat mengimplementasikan perancangan sistem yang telah dilakukan diperlukan beberapa hal yaitu perangkat keras, perangkat lunak, dan antar muka.

##### **5.1.1 Lingkungan Pengembangan**

Untuk dapat melakukan pengembangan perangkat lunak dengan nyaman dan memadai, spesifikasi perangkat keras komputer juga harus diperhatikan. Berikut spesifikasi yang direkomendasikan:

- a. Processor berbasis Intel atau AMD dengan kecepatan 1 GHz atau lebih,
- b. Kapasitas memori (RAM) 256 MB atau lebih.
- c. Alokasi ruang harddisk 1 GB atau lebih.
- d. Perangkat konektivitas seperti IrDA, bluetooth, *card reader* atau kabel data disesuaikan dengan fasilitas ponsel.

Perangkat konektivitas seperti bluetooth atau kabel data sangat membantu dalam pengembangan perangkat lunak, karena digunakan sebagai perangkat untuk mendistribusikan aplikasi pada lingkungan target.

Lingkungan pengembangan dilakukan dengan spesifikasi sebagai berikut:

- a. Perangkat notebook dengan spesifikasi: Intel Pentium M IV 2,8 GHz,  
Memori 256 MB L2 Cache 512 KB.

- b. Sistem operasi Microsoft Windows XP SP2.
- c. Kabel data CA-53 untuk ponsel Nokia

Selain perangkat keras yang telah disebutkan, lingkungan pengembangan juga memerlukan perangkat lunak pendukung sebagai berikut:

- a. JDK 1.5 sebagai platform pengembangan sistem.
- b. NetBeans IDE 5.0 dengan NetBeans Mobility 5.0 sebagai editor penulisan kode.
- c. J2ME Wireless Toolkit 2.2 sebagai emulator yang digunakan pada proses testing sistem.
- d. Adobe Photoshop 7.0 dan Corel Draw 12.0 untuk merancang desain gambar.

### 5.1.2 Lingkungan Target

Perangkat ponsel yang merupakan *target device*, harus memenuhi spesifikasi minimum sebagai berikut:

- a. Ponsel dengan Java platform yang mendukung MIDP 2.0.
- b. Kapasitas memori 1 MB atau lebih.
- c. Layar warna, minimal 4096 warna.

Lingkungan target ini mengacu pada ponsel Nokia 3230 dan Sony Ericksson W550i yang telah memenuhi spesifikasi minimum tersebut di atas.

### 5.1.3 Asumsi – asumsi Baru

Dalam proses pembuatan dan pengembangan sistem ini telah banyak ditemui asumsi baru yang lebih beragam dari batasan yang telah direncanakan, asumsi-asumsi baru itu antara lain :

1. Tingkat kesulitan *game* ini ditentukan dengan jumlah musuh, panjang lintasan, dan kekuatan boss pada tiap *level*nya.
2. Permainan pada tiap *level* akan berhenti jika tujuan atau *goal* terpenuhi atau pemain tidak lagi memiliki nyawa. Tujuan atau *goal* tiap *level* adalah sama, yaitu menghancurkan semua karakter musuh.
3. Disediakkannya input nama untuk mencatat *score* tertinggi, dan hanya dibatasi untuk 5 karakter saja.
4. Pencatatan statistik *score* tertinggi hanya dibatasi untuk 10 nilai tertinggi.

## 5.2 Implementasi

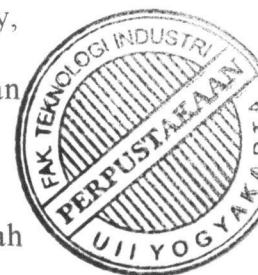
Implementasi aplikasi ini dibagi menjadi dua bagian, yaitu implementasi sistem dan implementasi antarmuka.

### 5.2.1 Implementasi Sistem

Implementasi sistem merupakan tahap dimana sistem mampu diaplikasikan dalam keadaan yang sesungguhnya. Dari implementasi ini akan diketahui apakah sistem yang dibuat dapat berjalan dengan baik atau tidak dan menghasilkan keluaran yang sesuai dengan perancangan yang ada. Pada tahap implementasi sistem ini dibagi menjadi beberapa bagian, yaitu:



1. Method `go` yang terdapat pada kelas `GameThread` adalah prosedur yang digunakan untuk menjalankan dan menghentikan thread utama.
2. Method `paint` yang terdapat pada kelas `GameManager` adalah prosedur yang digunakan untuk menampilkan gambar yang menyusun tiap *level*.
3. Method `moveNow` pada kelas `Background`, `Pesawat`, `Bullet`, `Ebullet`, `Enemy`, `Enemy2`, dan `Xplosion` adalah prosedur yang digunakan untuk menggerakkan objek yang ada pada kelas tersebut.
4. Method `checkCollision` pada kelas `Bomb`, `Bullet`, `Ebullet`, dan `Pesawat` adalah prosedur yang digunakan untuk pengecekan tabrakan setiap objek yang ada pada kelas tersebut.
5. Method `newGame` pada kelas `MyGameCanvas` adalah prosedur yang digunakan untuk memanggil objek penyusun *level*.
6. Method `endGame` pada kelas `MyGameCanvas` adalah prosedur yang digunakan untuk mengakhiri *game* dan menginput *score* kedalam kelas `ScoreList`.
7. Method `setSpeed` pada kelas `Bonus`, `Boss`, `Cloud`, `Enemy`, `Enemy2`, dan `Pesawat` adalah prosedur yang digunakan untuk mengatur kecepatan setiap objek yang ada pada kelas tersebut.
8. Method `shot` pada kelas `Boss`, `Enemy`, dan `PlayerShip` adalah prosedur yang digunakan untuk mengurangi jumlah health setelah tertembak.
9. Method `SetMove` pada kelas `GameManager` adalah prosedur yang digunakan untuk merespon input yang dilakukan pemain, baik menembak maupun melepaskan bom.



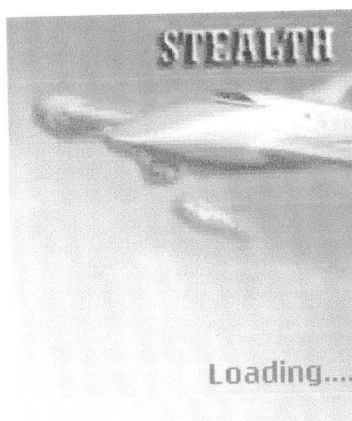
10. Method `SetSound` pada kelas `Stealth` adalah prosedur yang digunakan untuk melakukan setting suara pada *game* ini.

## 5.2.2 Implementasi Antarmuka

Pada bagian ini memuat gambaran penjelasan antarmuka yang terdapat pada sistem perangkat lunak *game* ini.

### 5.2.2.1 Antarmuka Splash Screen

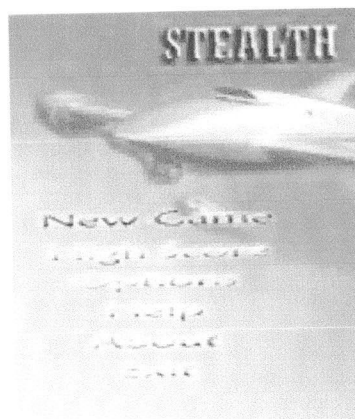
Pada form *splash screen* menampilkan *splash*, judul *game*, dan *loading*. Tampilan *form splash screen* akan hilang setelah beberapa detik dan secara otomatis aplikasi akan langsung menuju ke tampilan menu. Gambar 5.1 adalah gambar antarmuka form *splash screen*.



**Gambar 5.1** Antarmuka *splash screen*

### 5.2.2.2 Antarmuka Menu Utama

Pada tampilan menu terdapat enam menu utama, yaitu *New Game*, *HighScore*, *Option*, *Help*, *About*, dan *Exit*. Gambar 5.2 adalah gambar antarmuka menu utama.



Gambar 5.2 Antarmuka menu utama

### 5.2.2.3 Antarmuka Permainan

Antarmuka permainan memiliki tampilan-tampilan pada tiap *level*, yaitu *level 1*, *level 2*, dan *level 3*. Jika pemain dapat melewati *level 1*, maka akan masuk ke *level 2*, dan seterusnya. Indikasi lifebar dan jumlah nyawa terdapat pada bagian kiri atas layar, indikasi jumlah bomb terdapat pada bagian kanan atas layar, sedangkan pada bagian tengah terdapat indikasi jumlah *score*.

Berikut ini merupakan tampilan *game* pada *level 1*, *level 2*, *level 3*, dan tampilan ucapan "*level complete*" jika pemain mampu melewati *level* tertentu, tampilan ucapan *game over* jika pemain kalah.

Pada level 1, panjang lintasan yang dilewati hanya satu kali jumlah background, dan jumlah musuh yang dihadapi sedikit. Gambar 5.3 adalah gambar antarmuka permainan *level 1*.



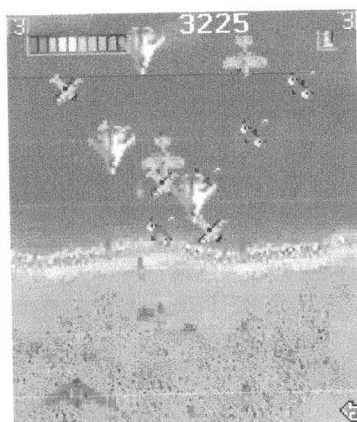
**Gambar 5.3** Antarmuka permainan *level 1*

Pada *level 2*, panjang lintasan yang dilewati hanya dua kali jumlah background, dan jumlah musuh yang dihadapi lebih banyak dari *level* sebelumnya. Gambar 5.4 adalah gambar antarmuka permainan *level 2*.



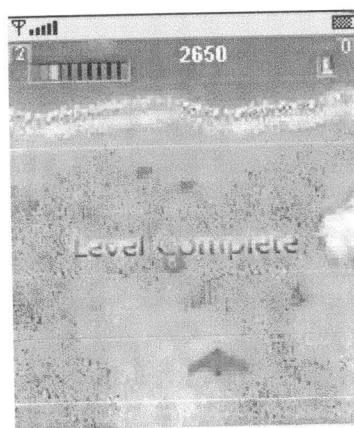
**Gambar 5.4** Antarmuka permainan *level 2*

Pada *level 3*, panjang lintasan yang dilewati hanya tiga kali jumlah background, dan jumlah musuh yang dihadapi lebih banyak dari *level* sebelumnya. Gambar 5.5 adalah gambar antarmuka permainan *level 3*.



**Gambar 5.5** Antarmuka permainan *level 3*

Indikasi tampilan *game* saat pemain mampu menyelesaikan *game* dari *level* tertentu adalah dengan munculnya informasi kemenangan pada layar. Gambar 5.6 adalah gambar antarmuka informasi permainan yang diwakili oleh *level 1* saat menang.



**Gambar 5.6** Antarmuka informasi permainan saat menang

Indikasi tampilan permainan saat pemain tidak dapat menyelesaikan *game* karena *life bar* dan jumlah nyawa habis adalah dengan munculnya informasi kekalahan. Gambar 5.7 adalah gambar antarmuka informasi permainan yang diwakili oleh *level 1* saat kalah.



**Gambar 5.7** Antarmuka informasi permainan saat kalah

#### 5.2.2.4 Antarmuka Masukan Nama

Jika pemain kalah atau dapat menyelesaikan permainan, maka akan muncul form untuk menginputkan nama pemain untuk ditampilkan pada *form high score*.

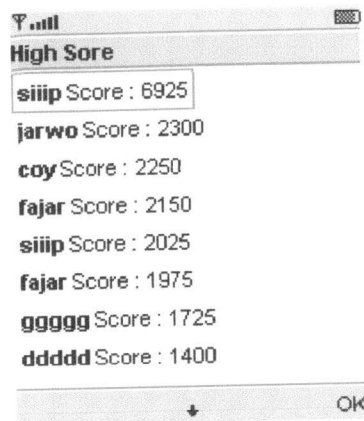
Gambar 5.8 adalah gambar antarmuka form input nama.



**Gambar 5.8** Antarmuka Masukan Nama

### 5.2.2.5 Antarmuka *High Score*

Antarmuka *High Score* digunakan untuk melihat catatan *score* tertinggi yang pernah dibuat oleh pemain. Gambar 5.9 adalah gambar antarmuka *High Score*.



Gambar 5.9 Antarmuka *High Score*

### 5.2.2.6 Antarmuka *Option*

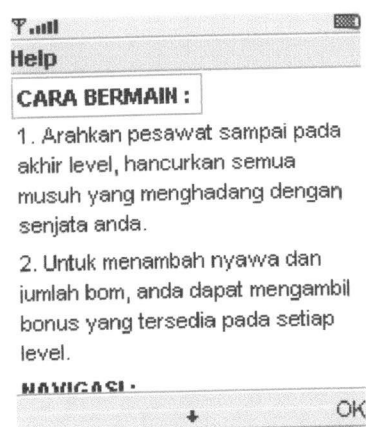
Antarmuka *Option* digunakan untuk menyalakan atau mematikan suara. Gambar 5.10 adalah gambar antarmuka *Option*.



Gambar 5.10 Antarmuka form *Option*

### 5.2.2.7 Antarmuka *Help*

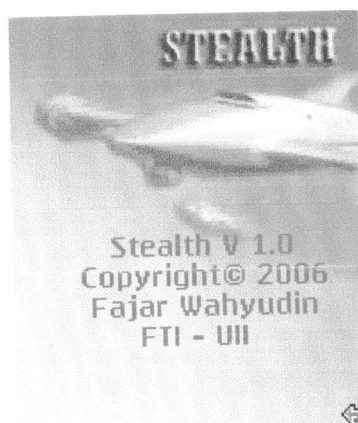
Antarmuka *Help* digunakan untuk menampilkan informasi tentang cara bermain, dan informasi navigasi. Gambar 5.11 adalah gambar antarmuka *Help*.



Gambar 5.11 Antarmuka *Help*

### 5.2.2.8 Antarmuka *About*

Antarmuka *About* digunakan untuk menampilkan informasi tentang *game*. Gambar 5.12 adalah gambar antarmuka *About*.



Gambar 5.12 Antarmuka *About*



## BAB VI

### ANALISIS KINERJA



#### 6.1 Pengujian Perangkat Lunak

Setelah *game* selesai dibuat, maka diperlukan pengujian pada *game* tersebut. Pengujian ini digunakan untuk mengetahui kekurangan pada *game* yang telah dibuat, dan untuk dapat mengetahui secara langsung apakah *game* telah bersifat *user friendly* atau tidak.

Untuk dilakukan pengujian secara langsung dan pemberian kuisisioner pada sepuluh orang responden dengan usia 15 sampai 25 tahun. Kuisisioner dibuat untuk mendapatkan penilaian langsung dari para responden tentang kekurangan-kekurangan yang ada pada *game* ini.

#### 6.2 Hasil Pengujian

Dalam kuisisioner tersebut diberikan beberapa range nilai untuk memudahkan penghitungannya, yaitu :

Nilai 1 untuk jawaban tidak baik.

Nilai 2 untuk jawaban kurang

Nilai 3 untuk jawaban sedang

Nilai 4 untuk jawaban baik

Nilai 5 untuk jawaban sangat baik

Dari nilai tersebut dapat digunakan untuk menghitung nilai rata-rata dari jawaban responden, rumus untuk menghitung nilai rata-ratanya adalah :

$$\text{Rata-rata : } \frac{\Sigma \text{ nilai jawaban (jumlah nilai jawaban)}}{\Sigma \text{ responden (jumlah responden)}}$$

Dari kuisisioner yang dibagikan kepada para responden, maka didapatkan hasil yang digambarkan pada tabel 6.1.

**Tabel 6.1** Tabel hasil analisis

Jawaban responden soal	1	2	3	4	5	6	7	8	9	10	Rata2
1. Bagaimana tampilan antarmuka game ini?	4	4	4	4	4	4	5	3	4	4	4,0
2. Apakah instruksi dan informasi pada game ini mudah dimengerti?	4	4	4	3	4	3	4	4	4	4	3,8
3. Bagaimana sound yang terdapat pada game ini?	4	4	3	4	3	3	4	3	3	4	3,5
4. Bagaimana susunan menu pada game ini?	4	4	4	3	4	4	4	3	4	4	3,8
5. Bagaimana tingkat kesulitan pada game ini?	4	4	3	4	3	4	4	4	3	4	3,7
6. Menurut anda sudah baikkah game ini?	4	4	4	3	4	4	4	4	4	3	3,8
7. Apakah anda terhibur dengan memainkan game ini?	4	4	3	4	3	4	4	5	4	4	3,9

### 6.3 Analisis Kesesuaian

Sesuai dengan teori dalam pembuatan *game* pada bab II, analisis yang dilakukan adalah :

1. *Game board*

*Game board* merupakan bentuk tampilan, dan untuk game ini menggunakan tampilan 2D yang *full colour*. Menurut responden tampilan antarmuka game ini sudah baik, hal itu dapat dilihat dari nilai rata-rata yang diberikan oleh responden pada pertanyaan nomor 1, yaitu dengan nilai 4,0.

2. Instruksi untuk pemain

*Game* telah dilengkapi dengan instruksi cara bermain yang dapat diakses pemain sebelum mulai bermain pada menu *help*. Menurut responden instruksi yang ada sudah baik, hal itu dapat dilihat dari nilai rata-rata yang diberikan oleh responden pada pertanyaan nomor 2, yaitu dengan nilai 3,8.

3. Informasi untuk pemain

Informasi pada *game* ini adalah informasi mengenai tombol yang akan digunakan dalam permainan yang dapat dilihat pada menu *help*. Menurut responden informasi yang ada sudah baik, hal itu dapat dilihat dari nilai rata-rata yang diberikan oleh responden pada pertanyaan nomor 2, yaitu dengan nilai 3,8.

4. Penghargaan

Ketika pemain berhasil memenangkan *game* ini maka akan muncul ucapan selamat dan pemain dapat mencatatkan namanya pada daftar high score, sedangkan untuk yang belum berhasil memenangkan *game* ini maka akan muncul dialog *game over*.

#### 5. Variasi

Disediakannya variasi sound berfungsi agar permainan tidak berjalan monoton. Menurut responden variasi sound yang ada dinilai cukup, hal itu dapat dilihat dari nilai rata-rata yang diberikan oleh responden pada pertanyaan nomor 3, yaitu dengan nilai 3,5.

#### 6. Tingkat kesulitan

Tingkat kesulitan dalam menyelesaikan *game* ini terletak pada panjang lintasan, jumlah musuh dan kekuatan boss, menurut responden tingkat kesulitan yang ada sudah baik, hal itu dapat dilihat dari nilai rata-rata yang diberikan oleh responden pada pertanyaan nomor 5, yaitu dengan nilai 3,7.

### 6.4 Gambaran Umum Sistem

Game “Stealth” merupakan game pertempuran pesawat yang ditujukan bagi para pengguna ponsel dengan Java platform yang telah mendukung MIDP 2.0. Game ini terdiri dari 3 level dimana tingkat kesulitan setiap level ditentukan oleh panjang lintasan, jumlah musuh, dan kekuatan boss. Setiap musuh yang dihancurkan akan menghasilkan *score* untuk pemain, dan jika *score* tersebut merupakan *score* tertinggi maka pemain akan mendapatkan penghargaan dengan menginputkan nama yang kemudian akan ditampilkan pada *form high score*.

Pada antarmuka saat game sedang berjalan disediakan tampilan yang informatif, seperti disediakannya indikasi jumlah nyawa, indikasi *life bar*, maupun indikasi jumlah bomb yang akan berguna bagi pemain. Selain itu disediakan juga bonus untuk menambah *life bar* dan jumlah bomb untuk pemain. Selain itu

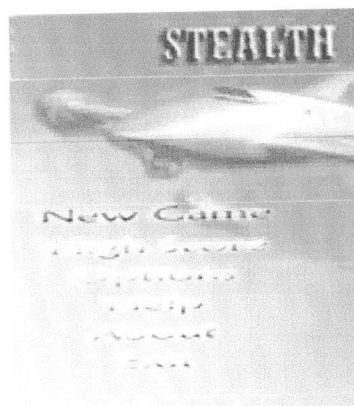
disediakan juga efek sound dan pengaturannya agar game tidak terlihat monoton, baik pada antarmuka menu utama maupun antarmuka game pada saat sedang berjalan.

Perbandingan antara *game* pertempuran “Stealth” dengan game “Aleste” yang merupakan game sejenis, dapat dilihat pada tabel 6.2.

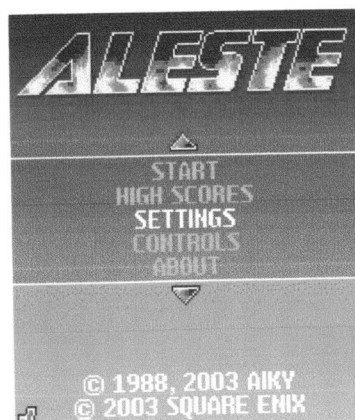
**Tabel 6.2** Tabel perbandingan *game*

No.	Pembanding	Game “Stealth”	Game “Aleste”
1.	Variasi background	Ada	Ada
2.	Indikasi score	Ada	Ada
3.	Indikasi life bar	Ada	Tidak ada
4.	Indikasi jumlah nyawa	Ada	Ada
5.	Konfigurasi suara	Ada	Ada
6.	Tampilan high score	Ada	Ada
7.	Tampilan help	Ada	Ada
8.	Variasi level	Ada (3 level)	Ada (7 level)
9.	Splash screen	Ada	Ada

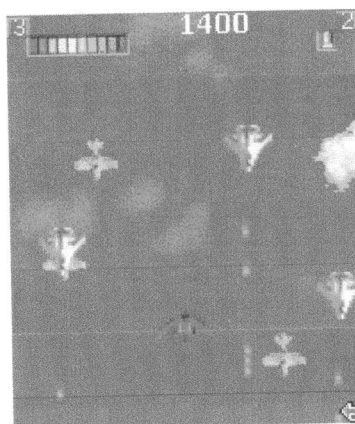
Untuk lebih jelasnya akan ditampilkan antarmuka dan menu pada game “Stealth” dan “Aleste” pada gambar 6.1, 6.2, 6.3, dan 6.4.



**Gambar 6.1** Tampilan Form Menu game “Stealth”



**Gambar 6.2** Tampilan FormMenu game “Aleste”



**Gambar 6.3** Antarmuka game “Stealth”



**Gambar 6.4** Antarmuka game “Aleste”

## BAB VII

### SIMPULAN DAN SARAN

Dalam bab ini akan disampaikan mengenai beberapa kesimpulan yang dapat diambil pada tugas akhir ini dan saran-saran untuk pengembangan tugas akhir ini selanjutnya.

#### 7.1 Simpulan

1. Dalam pembuatan *game*, yang harus dilakukan adalah pembuatan konsep *game* itu sendiri dan melakukan perancangan dengan menggunakan model yang mendukung pemrograman berorientasi objek, sehingga dalam pembuatannya sudah tersedia kerangka kerjanya. Hal ini dapat membantu dalam menentukan batasan dan fokus pengembangannya.
2. Penggunaan J2ME beserta JDK 1.5, NetBeans IDE 5.0 dalam implementasi *game* ini sangat membantu, karena :
  - a. Script pada bahasa pemrograman J2ME beserta JDK-nya menghasilkan file-file yang kecil sehingga memungkinkan untuk diakses melalui handphone.
  - b. Dengan didukung teknologi MIDP 2.0, pembuatan *game* dengan J2ME menjadi lebih mudah karena terdapat fitur-fitur baru yang dikhususkan untuk pembuatan *game*, yaitu *GameCanvas*, *Layer*, *LayerManage*, *Sprite*, *TiledLayer*. *GameCanvas* menyediakan fungsi-fungsi dasar dalam penanganan *user interface*. Kelas *Layer* digunakan sebagai elemen visual dari *game* dan kumpulan *layer-layer* tersebut diatur dalam kelas

LayerManager. Kelas Sprite digunakan untuk elemen visual yang dapat ditampilkan berdasarkan *frame*-nya. Dan kelas TiledLayer digunakan untuk membuat latar belakang dari sekumpulan potongan gambar-gambar.

- c. Penggunaan NetBeans IDE 5.0 untuk mendesain program dan penulisan kode program serta Java Wireless Toolkit 2.2 sebagai emulator ponsel memudahkan pengujian.
3. Aplikasi *game* pertempuran pesawat “Stealth” yang dibuat sudah *full colour* pada tampilannya, sehingga cukup menarik untuk dimainkan.
4. Tingkat kesulitan yang ada membuat orang merasa tertantang untuk memainkan game tersebut berkali-kali.
5. Aplikasi *game* pertempuran pesawat “Stealth” dapat digunakan sebagai sarana hiburan untuk mengisi waktu luang.

## 7.2 Saran

Berdasarkan pada pengujian yang telah dilakukan pada perangkat lunak yang dibuat, masih terdapat banyak kekurangan dan kelemahan sehingga perlu dikembangkan lagi agar menjadi lebih baik dan menarik, adapun hal tersebut adalah antara lain :

1. Membuat aplikasi *game* pertempuran pesawat “Stealth” berbentuk *game* 3D sehingga dapat menambah daya tarik tersendiri bagi pengguna.
2. Mengembangkan perencanaan *game* pertempuran pesawat “Stealth” dengan menambah tingkat kesulitan yang lebih tinggi, efek suara yang lebih beragam, dan pergerakan musuh yang lebih bervariasi.



3. Membuat *game* pertempuran pesawat “Stealth” dengan tampilan yang lebih dinamis,
4. Mengembangkan *game* pertempuran pesawat “Stealth” dengan menambah efek getar atau vibrasi pada saat tertembak atau meledak.
5. Pengembangan *game* pertempuran pesawat “Stealth” pada bahasa pemrograman lain maupun pada platform lain.



## DAFTAR PUSTAKA

- [HAR03a] Hariyanto, B. *Esensi-esensi Bahasa Pemrograman Java*. Bandung: Informatika Bandung, 2003
- [HAR03b] Hartanto, A. A. *Tip dan Trik Java 2 Micro Edition Tingkat Lanjut*. Jakarta: PT. Elex Media Komputindo, 2003
- [HAR04] Hartanto, A. A. *Pemrograman Mobile Java dengan MIDP 2.0*. Jogjakarta: ANDI, 2004
- [HEN89] Hendra. *Membuat Program Permainan*. Jakarta: Elex Media Komputindo, 1989.
- [ROL03] Rollings, A., dan Adams, E. *Andrew Rollings and Ernest Adams on Game Design*. USA : New Riders Publishing, 2003.
- [WIC02] Wicaksono, A., *Dasar-Dasar Pemrograman Java 2*, Jakarta : Penerbit PT Elex Media Komputindo, 2002.

## LAMPIRAN

## LAMPIRAN A

### KUISIONER

**Petunjuk pengisian :**

1. Berilah jawaban untuk setiap pertanyaan (jangan dikosongi)
2. Berikan tanda centang (  $\checkmark$  ) dalam menjawab pertanyaan sesuai dengan apa yang anda ketahui

Berapa usia anda?

( . . . -tahun)

Apakah anda suka bermain game?

( Ya / Tidak)\**coret salah satu*

No.	Pertanyaan	SB	B	S	K	TB
1.	Bagaimana mengenai tampilan antarmuka game ini?					
2.	Apakah instruksi dan informasi pada game ini mudah dimengerti?					
3.	Bagaimana sound yang terdapat pada game ini?					
4.	Bagaimana susunan menu pada game ini?					
5.	Bagaimana tingkat kesulitan pada game ini?					
6.	Menurut anda sudah baikkah game ini?					
7.	Apakah anda terhibur dengan memainkan game ini?					

**Keterangan :**

SB = Sangat Baik

B = Baik

S = Sedang

K = Kurang

TB = Tidak Baik

## KUISIONER

### Petunjuk pengisian :

1. Berilah jawaban untuk setiap pertanyaan (jangan dikosongi)
2. Berikan tanda centang ( ✓ ) dalam menjawab pertanyaan sesuai dengan apa yang anda ketahui

Berapa usia anda?

( 17-tahun)

Apakah anda suka bermain game?

( Ya / Tidak)\**coret salah satu*

No.	Pertanyaan	SB	B	S	K	TB
1.	Bagaimana mengenai tampilan antarmuka game ini?		✓			
2.	Apakah instruksi dan informasi pada game ini mudah dimengerti?		✓			
3.	Bagaimana sound yang terdapat pada game ini?		✓			
4.	Bagaimana susunan menu pada game ini?		✓			
5.	Bagaimana tingkat kesulitan pada game ini?		✓			
6.	Menurut anda sudah baikkah game ini?		✓			
7.	Apakah anda terhibur dengan memainkan game ini?		✓			

### Keterangan :

SB = Sangat Baik

B = Baik

S = Sedang

K = Kurang

TB = Tidak Baik

## KUISIONER

### Petunjuk pengisian :

1. Berilah jawaban untuk setiap pertanyaan (jangan dikosongi)
2. Berikan tanda centang ( ✓ ) dalam menjawab pertanyaan sesuai dengan apa yang anda ketahui

Berapa usia anda? ( ~~20~~ 23-tahun)

Apakah anda suka bermain game? ( Ya / Tidak)\**coret salah satu*

No.	Pertanyaan	SB	B	S	K	TB
1.	Bagaimana mengenai tampilan antarmuka game ini?		✓			
2.	Apakah instruksi dan informasi pada game ini mudah dimengerti?		✓			
3.	Bagaimana sound yang terdapat pada game ini?		✓			
4.	Bagaimana susunan menu pada game ini?		✓			
5.	Bagaimana tingkat kesulitan pada game ini?		✓			
6.	Menurut anda sudah baikkah game ini?		✓			
7.	Apakah anda terhibur dengan memainkan game ini?		✓			

### Keterangan :

SB = Sangat Baik

B = Baik

S = Sedang

K = Kurang

TB = Tidak Baik

## KUISIONER

### Petunjuk pengisian :

1. Berilah jawaban untuk setiap pertanyaan (jangan dikosongi)
2. Berikan tanda centang ( ✓ ) dalam menjawab pertanyaan sesuai dengan apa yang anda ketahui

Berapa usia anda?

( 22-tahun)

Apakah anda suka bermain game?

( Ya / Tidak)\**coret salah satu*

No.	Pertanyaan	SB	B	S	K	TB
1.	Bagaimana mengenai tampilan antarmuka game ini?		✓			
2.	Apakah instruksi dan informasi pada game ini mudah dimengerti?		✓			
3.	Bagaimana sound yang terdapat pada game ini?			✓		
4.	Bagaimana susunan menu pada game ini?		✓			
5.	Bagaimana tingkat kesulitan pada game ini?			✓		
6.	Menurut anda sudah baikkah game ini?		✓			
7.	Apakah anda terhibur dengan memainkan game ini?			✓		

### Keterangan :

SB = Sangat Baik

B = Baik

S = Sedang

K = Kurang

TB = Tidak Baik

## KUISIONER

### Petunjuk pengisian :

1. Berilah jawaban untuk setiap pertanyaan (jangan dikosongi)
2. Berikan tanda centang ( ✓ ) dalam menjawab pertanyaan sesuai dengan apa yang anda ketahui

Berapa usia anda? ( 12 ) -tahun)

Apakah anda suka bermain game? ( Ya / Tidak)\**coret salah satu*

No.	Pertanyaan	SB	B	S	K	TB
1.	Bagaimana mengenai tampilan antarmuka game ini?		✓			
2.	Apakah instruksi dan informasi pada game ini mudah dimengerti?			✓		
3.	Bagaimana sound yang terdapat pada game ini?		✓			
4.	Bagaimana susunan menu pada game ini?			✓		
5.	Bagaimana tingkat kesulitan pada game ini?		✓			
6.	Menurut anda sudah baikkah game ini?			✓		
7.	Apakah anda terhibur dengan memainkan game ini?		✓			

### Keterangan :

SB = Sangat Baik

B = Baik

S = Sedang

K = Kurang

TB = Tidak Baik



## KUISIONER

### Petunjuk pengisian :

1. Berilah jawaban untuk setiap pertanyaan (jangan dikosongi)
2. Berikan tanda centang ( ✓ ) dalam menjawab pertanyaan sesuai dengan apa yang anda ketahui

Berapa usia anda? ( 23-tahun)

Apakah anda suka bermain game? ( Ya / Tidak)\*coret salah satu

No.	Pertanyaan	SB	B	S	K	TB
1.	Bagaimana mengenai tampilan antarmuka game ini?		✓			
2.	Apakah instruksi dan informasi pada game ini mudah dimengerti?		✓			
3.	Bagaimana sound yang terdapat pada game ini?			✓		
4.	Bagaimana susunan menu pada game ini?		✓			
5.	Bagaimana tingkat kesulitan pada game ini?			✓		
6.	Menurut anda sudah baikkah game ini?		✓			
7.	Apakah anda terhibur dengan memainkan game ini?			✓		

### Keterangan :

SB = Sangat Baik

B = Baik

S = Sedang

K = Kurang

TB = Tidak Baik

## KUISIONER

### Petunjuk pengisian :

1. Berilah jawaban untuk setiap pertanyaan (jangan dikosongi)
2. Berikan tanda centang ( ✓ ) dalam menjawab pertanyaan sesuai dengan apa yang anda ketahui

Berapa usia anda? ( 21 -tahun)

Apakah anda suka bermain game? ( Ya / Tidak)\**coret salah satu*

No.	Pertanyaan	SB	B	S	K	TB
1.	Bagaimana mengenai tampilan antarmuka game ini?		✓			
2.	Apakah instruksi dan informasi pada game ini mudah dimengerti?			✓		
3.	Bagaimana sound yang terdapat pada game ini?			✓		
4.	Bagaimana susunan menu pada game ini?		✓			
5.	Bagaimana tingkat kesulitan pada game ini?		✓			
6.	Menurut anda sudah baikkah game ini?		✓			
7.	Apakah anda terhibur dengan memainkan game ini?		✓			

### Keterangan :

SB = Sangat Baik

B = Baik

S = Sedang

K = Kurang

TB = Tidak Baik

## KUISIONER



### Petunjuk pengisian :

1. Berilah jawaban untuk setiap pertanyaan (jangan dikosongi)
2. Berikan tanda centang ( ✓ ) dalam menjawab pertanyaan sesuai dengan apa yang anda ketahui

Berapa usia anda?

( 20-tahun)

Apakah anda suka bermain game?

( Ya / Tidak)\**coret salah satu*

No.	Pertanyaan	SB	B	S	K	TB
1.	Bagaimana mengenai tampilan antarmuka game ini?	✓				
2.	Apakah instruksi dan informasi pada game ini mudah dimengerti?		✓			
3.	Bagaimana sound yang terdapat pada game ini?		✓			
4.	Bagaimana susunan menu pada game ini?		✓			
5.	Bagaimana tingkat kesulitan pada game ini?		✓			
6.	Menurut anda sudah baikkah game ini?		✓			
7.	Apakah anda terhibur dengan memainkan game ini?		✓			

### Keterangan :

SB = Sangat Baik

B = Baik

S = Sedang

K = Kurang

TB = Tidak Baik

## KUISIONER

### Petunjuk pengisian :

1. Berilah jawaban untuk setiap pertanyaan (jangan dikosongi)
2. Berikan tanda centang ( ✓ ) dalam menjawab pertanyaan sesuai dengan apa yang anda ketahui

Berapa usia anda?

( 18. -tahun)

Apakah anda suka bermain game?

( Ya / Tidak)\**coret salah satu*

No.	Pertanyaan	SB	B	S	K	TB
1.	Bagaimana mengenai tampilan antarmuka game ini?			✓		
2.	Apakah instruksi dan informasi pada game ini mudah dimengerti?		✓			
3.	Bagaimana sound yang terdapat pada game ini?			✓		
4.	Bagaimana susunan menu pada game ini?			✓		
5.	Bagaimana tingkat kesulitan pada game ini?		✓			
6.	Menurut anda sudah baikkah game ini?		✓			
7.	Apakah anda terhibur dengan memainkan game ini?	✓				

### Keterangan :

SB = Sangat Baik

B = Baik

S = Sedang

K = Kurang

TB = Tidak Baik

## KUISIONER

### Petunjuk pengisian :

1. Berilah jawaban untuk setiap pertanyaan (jangan dikosongi)
2. Berikan tanda centang ( ✓ ) dalam menjawab pertanyaan sesuai dengan apa yang anda ketahui

Berapa usia anda? ( 24-tahun)

Apakah anda suka bermain game? ( Ya / Tidak)\**coret salah satu*

No.	Pertanyaan	SB	B	S	K	TB
1.	Bagaimana mengenai tampilan antarmuka game ini?		✓			
2.	Apakah instruksi dan informasi pada game ini mudah dimengerti?		✓			
3.	Bagaimana sound yang terdapat pada game ini?			✓		
4.	Bagaimana susunan menu pada game ini?		✓			
5.	Bagaimana tingkat kesulitan pada game ini?			✓		
6.	Menurut anda sudah baikkah game ini?		✓			
7.	Apakah anda terhibur dengan memainkan game ini?		✓			

### Keterangan :

SB = Sangat Baik

B = Baik

S = Sedang

K = Kurang

TB = Tidak Baik

## KUISIONER

### Petunjuk pengisian :

1. Berilah jawaban untuk setiap pertanyaan (jangan dikosongi)
2. Berikan tanda centang ( ✓ ) dalam menjawab pertanyaan sesuai dengan apa yang anda ketahui

Berapa usia anda?

( 12 - tahun)

Apakah anda suka bermain game?

( Ya / Tidak)\**coret salah satu*

No.	Pertanyaan	SB	B	S	K	TB
1.	Bagaimana mengenai tampilan antarmuka game ini?		✓	✓		
2.	Apakah instruksi dan informasi pada game ini mudah dimengerti?		✓			
3.	Bagaimana sound yang terdapat pada game ini?		✓			
4.	Bagaimana susunan menu pada game ini?		✓			
5.	Bagaimana tingkat kesulitan pada game ini?		✓			
6.	Menurut anda sudah baikkah game ini?			✓		
7.	Apakah anda terhibur dengan memainkan game ini?		✓			

### Keterangan :

SB = Sangat Baik

B = Baik

S = Sedang

K = Kurang

TB = Tidak Baik