

RANCANG BANGUN GAME MENGGUNAKAN TEKNOLOGI J2ME

**Laporan skripsi ini diajukan untuk
memenuhi salah satu persyaratan kurikulum
dalam menyelesaikan program strata-satu (S-1)**



Disusun Oleh:

AKHMAD ALFIAN ZUHRO SAPUTRO

08.18.186

**JURUSAN TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2012**

3045

INSTITUT TEKNOLOGI MASSACHUSETTS
UNIVERSITY OF CALIFORNIA
1-8 SAN FRANCISCO

08.12.1988

UNIVERSITY OF CALIFORNIA

San Francisco

San Francisco State University
Department of Biology
San Francisco, CA 94132

13NE

UNIVERSITY OF CALIFORNIA

RANCANG BANGUN GAME MENGGUNAKAN TEKNOLOGI J2ME

SKRIPSI

*Disusun dan Diajukan untuk melengkapi dan memenuhi persyaratan guna mencapai
Gelar Sarjana Teknik Informatika Strata Satu (S-1)*

Disusun Oleh :

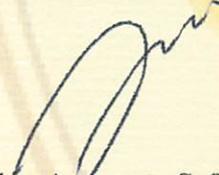
AKHMAD ALFIAN ZUHRO SAPUTRO

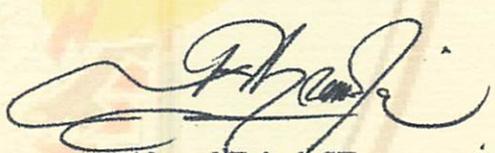
08.18.186

Telah disetujui dan disahkan oleh :

Dosen Pembimbing I

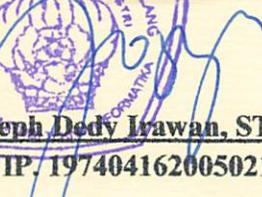
Dosen Pembimbing II


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


Ahmad Faisal, ST
NIP. 1031000431

Mengetahui

Ketua Jurusan Teknik Informatika S-1


Joseph Dedy Irawan, ST, MT.
NIP. 197404162005021002

JURUSAN TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

MILIK
PERPUSTAKAAN
ITN MALANG

LEMBAR KEASLIAN



PROGRAM STUDI TEKNIK INFORMATIKA S1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL
MALANG

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertandatangan di bawah ini :

Nama : Akhmad Alfian Zuhro Saputro

NIM : 0818186

Program Studi : Teknik Informatika S1

Fakultas : Teknologi Industri

Menyatakan bahwa skripsi yang berjudul :

“RANCANG BANGUN GAME MENGGUNAKAN TEKNOLOGI J2ME”

Adalah skripsi saya sendiri, bukan duplikat, dan seluruh sumber yang dikutip maupun dirujuk telah saya nyatakan dengan sebenarnya.

Malang, 18 Februari 2013

Yang membuat pernyataan



RANCANG BANGUN GAME MENGGUNAKAN TEKNOLOGI J2ME

AKHMAD ALFIAN ZUHRO SAPUTRO

Jurusan Teknik Informatika S-1
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang
Email : stregone.scald@yahoo.com

Abstrak

Aplikasi game merupakan hasil dari perkembangan dunia software. Dimana semua umur dapat memainkannya, terutama bagi anak-anak. Walaupun game ditujukan untuk anak-anak, tidak sedikit pula orang dewasa kerap memainkannya bahkan tidak sedikit yang menjadikannya sebagai pekerjaan dan mendapatkan penghasilan dari bermain game. Pada penulisan ini akan membahas pembuatan aplikasi java game mobile. Namun pembahasan di batasi pada bagaimana aplikasi ini dapat berjalan dalam perangkat mobile.

Implementasi aplikasi ini dibuat menggunakan software Java(TM) ME platform SDK 3.0. Wireless Toolkit 2.5.2, adobe photoshop, sedangkan bahasa yang digunakan adalah bahasa java 2 Micro Edition.

Kata kunci : Java, mobile, j2me, Aplikasi permainan.

KATA PENGANTAR

Rasa syukur yang dalam kita panjatkan kehadirat Allah SWT Yang Maha Pengasih lagi Maha Penyayang, karena dengan rahmat, taufik, serta hidayah-Nya semata laporan skripsi yang berjudul **“RANCANG BANGUN GAME MENGGUNAKAN TEKNOLOGI J2ME”** dapat diselesaikan. Dalam proses pembangunan sistem hingga penyusunan laporan ini, tentunya kami mendapat bimbingan, penjelasan, arahan, koreksi, dan saran. Untuk itu rasa terimakasih yang sedalam-dalamnya kami sampaikan kepada :

1. Bapak Ir. Soeparno Djiwo, MT, selaku rector Institut Teknologi Nasional Malang.
2. Bapak Ir. Anang Subardi, MT, selaku ketua Fakultas Institut Teknologi Nasional Malang.
3. Bapak Josep Dedy Irawan, ST, MT, selaku ketua program studi Teknik Informatika S-1.
4. Bapak Dr.Eng Aryuanto S, ST, selaku dosen pembimbing I skripsi ini atas bimbingan dan petunjuk yang telah diberikan kepada penyusun.
5. Bapak Ahmad Faisol, ST, selaku dosen pembimbing II skripsi ini atas bimbingan dan petunjuk yang telah diberikan kepada penyusun.
6. Kedua orang tua penyusun yang selalu mendoakan dan mendukung dalam pengerjaan skripsi ini.
7. Sahabat serta rekan-rekan yang telah banyak memberikan masukan dalam pengerjaan laporan skripsi.

Demikianlah laporan skripsi ini kami buat, semoga mempunyai manfaat bagi penulis sendiri maupun yang membaca laporan ini.

Malang, Februari 2013

Penyusun

Daftar Isi

Lembar Judul	i
Lembar Persetujuan	ii
Lembar Keaslian.....	iii
Abstrak.....	iv
Kata Pengantar	v
Daftar Isi	vi
Daftar Gambar	viii
Daftar Tabel.....	viii
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah.....	2
1.5 Metode Penulisan	3
1.6 Sistematika Penulisan	4
BAB II LANDASAN TEORI	
2.1 Game.....	5
2.2 Genre Game	6
2.3 Java	11
2.4 Java 2 Micro Edition (J2ME).....	12
2.5 Connected Limited Device Configuration (CLCD)	15
2.6 Connected Device Configuration (CDC).....	15
2.7 JDK dan JVM.....	16
2.7.1 JDK (Java Development Kit).....	16
2.7.2 JVM (Java Virtual Machine).....	16
2.8 Netbeans.....	17
2.8.1 Class MIDlet	17
2.8.2 Class splash	17
2.8.3 Class Menu.....	17

2.8.4 Class Counter	17
2.8.5 Class Game	17
2.8.6 Class Engine.....	18
BAB III ANALISA DAN PERANCANGAN	
3.1 Skenario Game	19
3.2 Analisa Sistem.....	19
3.2.1 Spesifikasi aplikasi	19
3.2.2 Spesifikasi perangkat handphone	20
3.2.3 Spesifikasi perangkat keras dan lunak	20
3.3 Perancangan Sistem.....	21
3.3.1 Desain alur aplikasi	21
3.3.2 Flow Chart	22
3.3.3 Rancangan Menu.....	23
3.4 Perancangan Antar Muka.....	24
3.4.1 Halaman Splash Screen	24
3.4.2 Halaman Menu	25
3.4.3 Halaman Ruang Permainan.....	26
BAB IV IMPLEMENTASI DAN PENGUJIAN	
4.1 Kebutuhan Sistem.....	27
4.1.1 Perangkat keras	27
4.2.2 Perangkat Lunak.....	27
4.2 Penggunaan Aplikasi	28
4.2.1 Instalasi Wireless toolkit.....	28
4.2.2 Spesifikasi Perangkat Mobile.....	28
4.2.2 Hasil Implementasi.....	28
4.3 Pengujian Fungsional Sistem	31
BAB V PENUTUP	
5.1 Kesimpulan	32
5.2 Saran	32
DAFTAR PUSTAKA	
LAMPIRAN	

Daftar Gambar

Gambar 2.1 Arsitektur J2ME	13
Gambar 2.2 Lingkup Configuration	14
Gambar 2.3 Hubungan J2ME dan J2SE	14
Gambar 2.4 Arsitektur CLDC	15
Gambar 3.1 Alur Aplikasi	21
Gambar 3.2 Flow Chart.....	22
Gambar 3.4 Halaman Splash Screen	24
Gambar 3.5 Halaman Menu	25
Gambar 3.6 Halaman ruang karakter utama	26
Gambar 4.1 Tampilan Awal.....	29
Gambar 4.2 Tampilan Fungsi Pengaturan	29
Gambar 4.3 Tampilan Pengisian	30
Gambar 4.4 Tampilan Permainan.....	30

Daftar Tabel

Tabel 2.1 Tabel Perbandingan Antara CDC dan CLDC	16
Tabel 2.1 Tabel Pengujian Fungsional	31



BAB I

PENDAHULUAN

1.1 Latar belakang masalah

Perkembangan teknologi selular dapat dikatakan sangat memberi pengaruh besar terhadap masyarakat dunia. Hampir semua orang di dunia menggunakan perangkat cellular ini dalam melakukan kehidupan keseharian mereka. Tahun 1920 teknologi selular sudah dikenalkan. Tahun 1928 kepolisian detroit mulai menggunakan radio komunikasi satu arah para semua mobil patroli dengan frekuensi 2Mhz. Kemudian tahun 1973 telepon selular pertama diperkenalkan ke public. Beberapa kekurangan versi ini membuat perkembangan teknologi selular semakin maju. Untuk memenuhi kebutuhan masyarakat, dibuat aplikasi-aplikasi pendukung yang dipasang pada telpone selular untuk membantu kegiatan keseharian masyarakat atau sebagai hiburan yang dapat dimainkan. Salah satu aplikasi tersebut adalah aplikasi game.

Aplikasi game merupakan hasil dari perkembangan dunia software. Dimana semua umur dapat memainkannya, terutama bagi anak-anak. Walaupun game ditujukan untuk anak-anak, tidak sedikit pula orang dewasa kerap memainkannya bahkan tidak sedikit yang menjadikannya sebagai pekerjaan dan mendapatkan penghasilan dari bermain game.

Sebuah game Game memiliki sifat adiktif yang dapat membuat pemainnya menghabiskan waktu berjam-jam untuk memainkan sebuah permainan hingga lupa akan pentingnya belajar. Dengan adanya sifat adiktif tersebut dibuatnya aplikasi game yang dapat membantu anak-anak dalam belajar. Sehingga anak-anak dapat mendapatkan pembelajaran dari game yang dimainkan di handphone miliknya.

Dalam pembuatan aplikasi game, digunakan bahasa pemrograman Java2 Micro Edition (J2ME). Java2 Micro Edition (J2ME) adalah lingkungan pengembangan yang didesain untuk meletakkan perangkat lunak Java pada

media elektronik beserta perangkat pendukungnya. Java2 Micro Edition (J2ME) merupakan salah satu produk Sun Microsystems. Pada J2ME, jika perangkat lunak berfungsi baik pada sebuah perangkat maka belum tentu juga berfungsi baik pada perangkat yang lainnya.

1.2 Rumusan masalah

Berdasarkan latar belakang diatas, dapat diambil suatu rumusan masalah yaitu bagaimana membuat suatu aplikasi permainan yang menarik dan menyediakan pembelajaran di dalam sebuah aplikasi game mobile.

1.3 Tujuan

Penelitian ini bertujuan untuk :

1. Menyediakan sebuah media pembelajaran mengenai perangkat dasar jaringan.
2. Menyediakan hiburan yang dapat dimainkan di media telpon selular.

1.4 Batasan masalah

Batasan permasalahan dalam pembuatan aplikasi hanya dibatasi sebagai berikut:

1. Aplikasi game mobile dibuat menggunakan bahasa pemrograman JAVA
2. Menggunakan teknologi j2me dalam pembuatan aplikasi
3. Permainan dimainkan secara single player.

1.5 Metode penulisan

Pada pembuatan skripsi ini terdapat beberapa tahapan yang dilakukan yaitu tahap studi literatur, analisa dan desain sistem, pembangunan aplikasi, pengujian, dan kesimpulan.

a. Studi Literatur

Pengumpulan data yang dilakukan dengan mencari bahan-bahan keperpustakaan dan referensi dari berbagai sumber, baik dari buku, artikel, maupun internet sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian.

b. Representasi Pengetahuan

Merepresentasikan pengetahuan yaitu bagaimana cara untuk menyimpan pengetahuan yang dibutuhkan ke dalam komputer. Memilihkan pemrograman, menentukan cara dimana pengetahuan dapat ditunjukkan.

c. Merancang dan Membuat Aplikasi Game

Setelah representasi pengetahuan maka dilakukan perancangan dan pembuatan program aplikasi game yang akan digunakan. Pembuatan program aplikasi game dilakukan berdasarkan pengetahuan yang digambarkan pada tahap representasi pengetahuan dan dalam pembuatan aplikasi game ini akan menggunakan bahasa JAVA.

d. Uji Coba Dan Evaluasi Hasil Uji Coba Program

Aplikasi game yang telah dibuat dilakukan uji pada sebuah perangkat

mobile untuk mengetahui aplikasi bekerja dengan baik.

1.6 Sistematika penulisan

Adapun sistematika penulisan laporan Skripsi ini terdiri atas :

BAB I : PENDAHULUAN

Berisi penjelasan tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian serta sistematika penulisan.

BAB II : LANDASAN TEORI

Bab ini berisi tentang penjelasan teori - teori penunjang yang berhubungan dengan topik skripsi, meliputi hal-hal yang berhubungan dengan perancangan sistem, serta teori-teori dasar lainnya seperti *java*, *Hardware jaringan*, dan sebagainya.

BAB III : ANALISA DAN PERANCANGAN SISTEM

Bab ini berisi tentang deskripsi sistem serta desain rancangan sistem.

BAB IV : IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini menyajikan hasil implementasi serta pengujian mengenai cara kerja dari sistem.

BAB V : KESIMPULAN DAN SARAN

Bab ini menyajikan kesimpulan dan saran mengenai pembahasan yang telah ditempuh.

BAB II

LANDASAN TEORI

2.1 Game

Game adalah bermain yang terstruktur, yang biasanya dilakukan untuk bersenang-senang atau sebagai media pembelajaran. Game berbeda dengan pekerjaan yang mana dalam pekerjaan dilakukan untuk mendapatkan suatu hasil atau hadiah. Game juga berbeda dengan seni yang merupakan hasil sebuah ekspresi dan unsur estetika atau ideologi. Namun perbedaan ini tidak begitu jelas, banyak permainan yang dianggap sebagai pekerjaan atau seni.

Pada umumnya game melibatkan stimulasi mental atau fisik, dan keduanya. Banyak permainan yang membantu mengembangkan kemampuan, tersedia dalam bentuk latihan, edukasi, stimulasi, atau psikologi. Komponen kunci dalam sebuah permainan adalah tujuan, peraturan, tantangan dan interaksi. Sebuah game dapat diklasifikasikan dari apa yang pemain lakukan, misalnya [6]:

1. Peralatan

Game terkadang diklasifikasikan oleh komponen-komponen yang dibutuhkan dalam permainan, misalnya : bola, kartu, papan, komputer.

2. Peraturan

Peraturan digunakan untuk menentukan giliran pemain, hak dan keharusan masing-masing pemain, dan tujuan pemain.

3. Kemampuan, strategi dan keberuntungan

Game dengan skill, contohnya dengan kekuatan fisik, misal gulat, menembak dan kekuatan mental seperti catur. Game strategi contohnya catur dan tic tac toe. Game keberuntungan contohnya permainan dadu. Namun banyak game yang memiliki dua atau tiga dari element tersebut. Contohnya permainan American Football dan baseball melibatkan

kemampuan fisik, mental, dan strategi. Sedangkan poker dan monopoli menggabungkan strategi dan keberuntungan. Banyak permainan kartu dan papan menggabungkan kemampuan mental, strategi, dan keberuntungan.

4. Single player game (permainan satu orang)

Banyak permainan yang membutuhkan lebih dari satu pemain. Namun single player juga merupakan sesuatu yang unik. Tidak seperti permainan multi player yang berkompetisi melawan satu sama lain untuk mencapai tujuan, single player melakukan permainan melawan kemampuannya sendiri, waktu, atau sebuah keberuntungan. Permainan tidak sah jika game computer memberikan sebuah perlawanan. Jika computer hanya mencatat nilai atau waktu saja, game tersebut dapat dikatakan sebagai single player game.

Terdapat beberapa jenis game yang dapat dimainkan, salah satunya yaitu game edukatif. Game jenis ini dibuat dengan tujuan spesifik sebagai alat pendidikan. Seperti mengenal warna untuk balita, mengenal huruf dan angka, matematika atau belajar bahasa asing. Ada juga yang ditujukan ke anak sekolah, sebagai contoh game tentang pelajaran biologi dimana di dalam game tersebut menyediakan konten misalnya tentang fungsi organ tubuh manusia.[6]

2.2 Genre game

Mengkatagorikan jenis game dilakukan berdasarkan dari interaksi gameplay dari pada perbedaan tampilan atau narasi. Sebuah jenis game dapat didefinisikan dengan kumpulan dari sebuah tipe permainan dari game tersebut. Game-game tersebut dapat diklasifikasikan dengan berbagai pengaturan atau dari konten dan isi game tersebut, tidak seperti hasil fiksi, film ataupun buku. Sebagai contoh, sebuah game action tetap saja di katakan game action walaupun game tersebut mengambil tempat didunia fantasi ataupun di ruang angkasa. Adapun beberapa katagori game sebagai berikut.[1]

1. Arcade

Game bertipe ini dapat dikatakan tipe-tipe game klasik. Salah satu ciri yang biasanya ditemui untuk game-game bertipe seperti ini, pada umumnya memiliki tampilan 2 dimensi dan karakter-karakter di dalam game tersebut dapat bergerak ke atas, bawah, kiri, dan kanan. Biasanya diikuti juga dengan pergerakan latarbelakang game yang selalu berganti dari wilayah ke wilayah yang lain. Beberapa contoh game bertipe arcade ini adalah Sonic (SEGA), Mario Bros (Nitendo), dan Metal Slug (PS). Meskipun game-game setipe menggunakan sudut pandang 2 dimensi namun seiring dengan perkembangan console dunia game-game setipe tetap dipertahankan dengan memperbaiki darisegi tampilan yang dibuat dengan semi 3 dimensi sehingga menjadikan game-game tersebut tetap enak untuk dipandang dan dimainkan.[1]

2. Shooting

Game shooting adalah tipe game yang sebenarnya cukup sederhana, karena apa yang harus kita lakukan di dalam game tersebut, kita hanya cukup menembak musuh-musuh yang menghalangi kita selama bermain. Secara umum game shooting ini menggunakan tampilan 3 dimensi. Perkembangan game shooting ini cukup baik jika dilihat hingga tahun 2007 ini, begitu banyak inovasi yang diberikan untuk game jenis ini, dimulai dari performa grafis yang ditingkatkan hingga menawarkan sebuah game dengan tampilan yang menawan dan semakin sedap dipandang, pemberian *special effect* memukau dari tembakan dan ledakan, dan penyisipan alur cerita yang akan membuat *gamer* menjadi semakin penasaran untuk menuntaskan suatu game hingga final. Game bertipe *shooting* ini dibagi menjadi 2 kategori, yaitu *first person shooter (fps)* dan *third person shooter (tps)*. Hal yang membedakan diantara 2 katagori tersebut terletak pada sudut pandang pemain di dalam game tersebut. Seperti halnya *fps*, pada katagori ini sudut pandang pemain ditempatkan sebagai orang pertama dalam hal ini karakter utama. Secara umum pada game *fps* ini hanya ditampilkan senjata yang digunakan karakter utama atau juga untuk beberapa game hanya menampilkan penanda sasaran senjata saja. Game seperti DOOM, Far Cry, F.E.A.R, Stalker, dan Project IGI 1 dan 2 (PC),

dikategorikan ke dalam game *fps*. Lain hanya game kategori *tps*, pada kategori ini pemain ditempatkan sebagai orang ketiga yang mengendalikan karakter utama yang dimainkan di dalam game tersebut. Secara umum pada game bertipe *fps* ini ditampakkan wujud utama yang dimainkan di dalam game tersebut. Contoh game bertipe *tps* ini antara lain seperti, Tomb Rider series, Splinter Cell series, Cold Fear, dan Max Payne (PC).[1]

Permainan ini bisa berupa misi melumpuhkan penjahat maupun alien, kadang juga sejumlah pemain dibagi beberapa tim yang bertugas melumpuhkan tim lainnya, sebelum dilumpuhkan. Ciri utama lain adalah penggunaan senjata genggam jarak jauh.

Contoh game genre ini antara lain *Duke Nukem 3D*, *Quake*, *Blood*, *Unreal*, *Unreal Tournament*, seri *Half-Life*, *Counter-Strike*, seri *Halo*, *Perfect Dark*, *TimeSplitters*, *Call of Duty*, *System Shock*, dan *GoldenEye 007*.

3. RPG (role playing game)

(*role-playing game* disingkat RPG) adalah sebuah permainan yang para pemainnya memainkan peran tokoh-tokoh khayalan dan berkolaborasi untuk merajut sebuah cerita bersama. Para pemain memilih aksi tokoh-tokoh mereka berdasarkan karakteristik tokoh tersebut, dan keberhasilan aksi mereka tergantung dari sistem peraturan permainan yang telah ditentukan. Asal tetap mengikuti peraturan yang ditetapkan, para pemain bisa berimprovisasi membentuk arah dan hasil akhir permainan ini.

Dalam sebuah permainan RPG, jarang ada yang “kalah” atau “menang”. Ini membuat pemain RPG berbeda dari jenis permainan papan lainnya seperti Monopoli atau Ular Tangga, permainan kartu, olah raga, dan permainan lainnya. Seperti sebuah novel atau film, permainan RPG mempunyai daya tarik karena permainan-permainan ini mengajak para pemain untuk menggunakan imajinasi mereka. RPG biasa lebih mengarah ke kolaborasi sosial daripada kompetisi. Pada umumnya dalam RPG, para pemain tergabung dalam satu kelompok.

Permainan RPG rata-rata dimainkan seperti sebuah drama radio: ketika seorang pemain “berbicara”, dia berbicara sebagai tokohnya dan ketika si pemain ingin tokohnya melakukan sesuatu yang fisik (seperti menyerang sebuah monster atau membuka sebuah gembok) dia harus menggambarkannya secara lisan.

Ada pula sejenis permainan RPG di mana para pemain bisa melakukan gerakan fisik tokohnya oleh si pemain sendiri. Ini disebut *Live-Action Role-playing* atau LARP. Dalam permainan LARP, biasanya para pemain memakai kostum dan menggunakan alat-alat yang sesuai dengan tokoh, dunia dan cerita yang dia mainkan.

Adapun istilah lainnya adalah MMORPG (*Massively multiplayer online role-playing game*) adalah permainan role-playing game (RPG) yang melibatkan ribuan pemain untuk bermain bersama dalam dunia maya yang terus berkembang pada saat yang sama melalui media internet dan jaringan.[6]

Contoh dari genre permainan ini *World of Warcraft, The Lord of the Rings Online: Shadows of Angmar, Final Fantasy, Ragnarok, DOTTA*.

4. Simulation

Sama seperti artinya yang berarti simulasi, di dalam game ini diberikan penggambaran konsep permainan tentang segala suatu hal di dalam kehidupan nyata, sehingga hal realistik akan lebih sering ditemukan di dalam game bertipe ini. Game bertipe *simulation* ini dapat menggambarkan tentang kehidupan pribadi kita sehari-hari, kehidupan dalam pekerjaan, pengaturan suatu hal, dan pengoperasian alat-alat tertentu, misalnya seperti simulasi pembangunan tata kota di dalam game SIM CITY series, simulasi kehidupan seorang manager, seperti Football Manager series, Hospital Tycoon, dan Roller Coaster Tycoon, dan yang paling populer adalah game simulasi yang terlahirdari ide sederhana tentang kehidupan sehari-hari, yaitu game The SIMS yang meraup keuntungan besar karena kepopulerannya.[1]

5. Racing

Ini adalah salah satu tipe game yang pada tahun 2005 dan 2006 lalu begitu berkembang pesat. Hal ini dikarenakan, elemen-elemen yang disertakan di dalam game tersebut mengikuti perkembangan otomotif di dunia, sehingga tampilan yang begitu realistic akan kita dapatkan yang dapat dinikmati di dalam game bertipe *racing*. Namun, tujuan dari game bertipe ini tetaplah simple, yaitu melakukan balapan dan merebut posisi pertama, meski akhir-akhir ini game racing ditambahkan fitur-fitur baru seperti kita bias membeli kendaraan baru, memodifikasi kendaraan, dan bahkan penambahan elemen-elemen cerita yang semakin membuat game-game sejenis tidak menjemukan untuk dimainkan. Contoh-contoh game bertipe *racing* antara lain adalah, Grand Turismo, Need For Speed Underground series dan Need For Speed Carbon.[1]

6. Fighting

Game bertipe *fighting* ini merupakan salah satu tipe game yang cukup digandrungi di kalangan remaja. Di dalam game bertipe ini, kita akan memainkan satu karakter atau lebih yang kemudian karakter tersebut harus bertarung dengan karakter-karakter lainnya hingga mencapai pertarungan terakhir. Pada awalnya game bertipe *fighting* ini menggunakan tampilan 2 dimensi, namun pada saat ini game-game setipe telah dikembangkan hingga memiliki tampilan 3 dimensi yang cukup realistic dan efek-efek yang bagus dalam setiap aksi yang dilakukan oleh setiap karakter. Game bertipe *fighting* ini tidak hanya untuk dinikmati sendiri tetapi juga dapat dinikmati oleh 2 pemain, sehingga game bertipe *fighting* menjadi salah satu tipe game yang banyak dinikmati pada lingkup komunitas. Game-game seperti Street Fighter series, Mortal Combat Series, Tekken, dan King of Fighter bias di katagorikan di dalam jenis game *fighting*. [1]

7. RTS (Real Time Strategy)

Pada game bertipe RTS ini, kita sebagai pemain melakukan control terhadap satu atau lebih karakter untuk melakukan tindakan-tindakan tertentu

guna menyelesaikan masalah (misi) di dalam game tersebut. Begitu banyak perkembangan yang telah dilakukan oleh para *developer* game-game bertipe ini. Dimulai dari penyajian grafis yang ditampilkan secara 2 dimensi hingga kini telah dikembangkan hingga menggunakan grafis 3 dimensi yang diberikan pada elemen-elemen game, baik karakter, bangunan, kendaraan, senjata, dan lingkungan. Serta tidak kurang pula, trend game bertipe RTS antara lain seperti, *Commandos series*, *Command Conquer series*, *War Craft series*, dan *Age of empire series*. [1]

2.3 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi kedalam p-code (bytecode) dan dapat dijalankan pada berbagai mesin Virtual Java (JVM). [8]

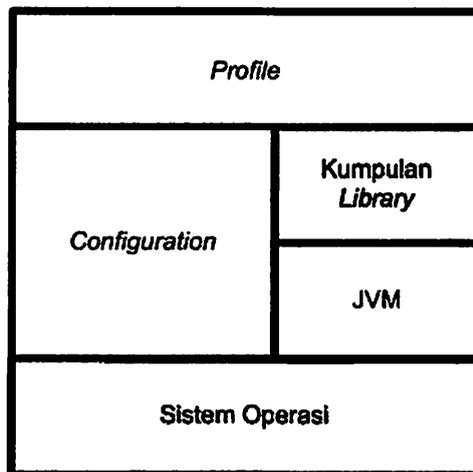
Java merupakan bahasa pemrograman yang bersifat umum dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda. Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak atau aplikasi berbasis web. Bahasa pemrograman java lahir dari The Green Project, yang berjalan selama 1 bulan, dari awal tahun 1991 hingga musim panas 1992. Project tersebut belum menggunakan versi yang dinamakan Oak. Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, James Gosling dan Bill Joy, beserta sembilan pemrogram lainnya dari Sun Microsystems. Salah satu hasil proyek ini adalah maskot Duke yang dibuat oleh Joe Palrang. [8]

Pada sekitar bulan Maret 1995, untuk pertama kali kode sumber Java versi 1.0a2 dibuka. Kesuksesan mereka diikuti dengan pemberitaan pertama kali pada surat kabar San Jose Mercury News pada tanggal 23 Mei 1995. Sayangnya terjadi perpecahan di antara mereka. Nama Oak diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja "Bapak Java", James Gosling. Nama Oak ini tidak dipakai untuk versi release Java karena sebuah perangkat lunak lain sudah terdaftar dengan merek dagang tersebut, sehingga di ambil nama penggantinya menjadi "Java". Nama ini diambil dari kopi murni yang digiling langsung dari biji kesukaan Gosling. Kopi ini berasal dari pulau Jawa. Jadi nama bahasa pemrograman Java tidak lain berasal dari kata Jawa (bahasa Inggris untuk Jawa adalah Java).[8]

2.4 Java 2 Micro Edition (J2ME)

Java2 Micro Edition (J2ME) adalah lingkungan pengembangan yang didesain untuk meletakkan perangkat lunak Java pada perangkat elektronik beserta perangkat pendukungnya. Java programming language. Bahasa pemrograman yang digunakan mirip dengan C++ tetapi secara fundamental berbeda. C++ menggunakan pointer-pointer yang kurang aman dan mengharuskan programmer untuk mengalokasikan dan mengkosongkan memori. Sedangkan Java menggunakan typesafe object reference dan setiap memori yang tidak digunakan akan dikosongkan secara otomatis. Java juga mendukung multi inheritance dengan konstruksi yang lebih baik, yaitu interface.[10]

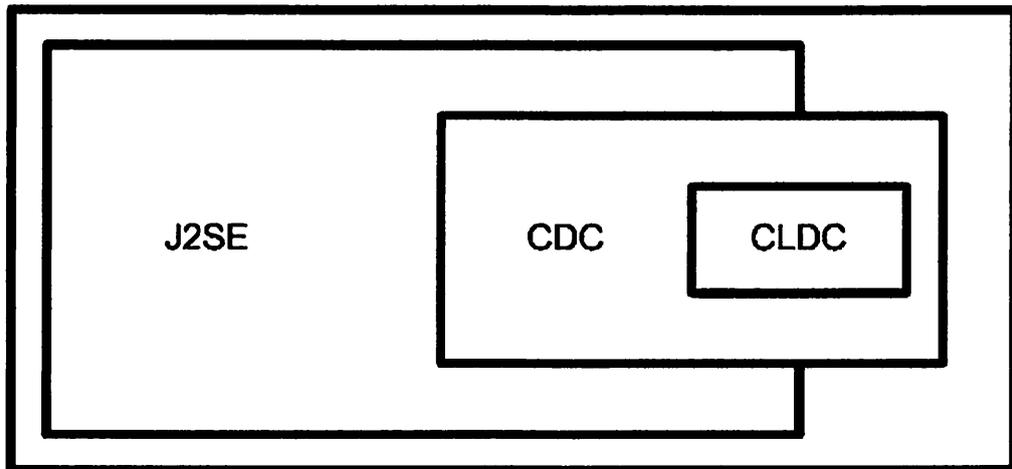
J2ME adalah bagian dari J2SE, karena itu tidak semua *library* yang ada pada J2SE dalam digunakan pada J2ME. Tetapi J2ME mempunyai beberapa *library* khusus yang tidak dimiliki J2SE. Arsitektur J2ME dapat dilihat pada gambar berikut [10] :



Gambar 2.1 Arsitektur J2ME

Teknologi J2ME juga memiliki beberapa keterbatasan, terutama jika diaplikasikan pada ponsel. J2ME sangat tergantung pada perangkat yang digunakan, bisa dari segi merk ponsel, maupun kemampuan ponsel dan dukungannya terhadap teknologi J2ME. Misalnya, jika sebuah ponsel tidak memiliki kamera, maka jelas J2ME pada ponsel tersebut tidak dapat mengakses kamera. Keterbatasan lainnya adalah pada ukuran aplikasi, karena memori pada ponsel sangat terbatas. Sebaian besar ponsel tidak mengijinkan aplikasi J2ME menulis pada file karena alasan keamanan.[10]

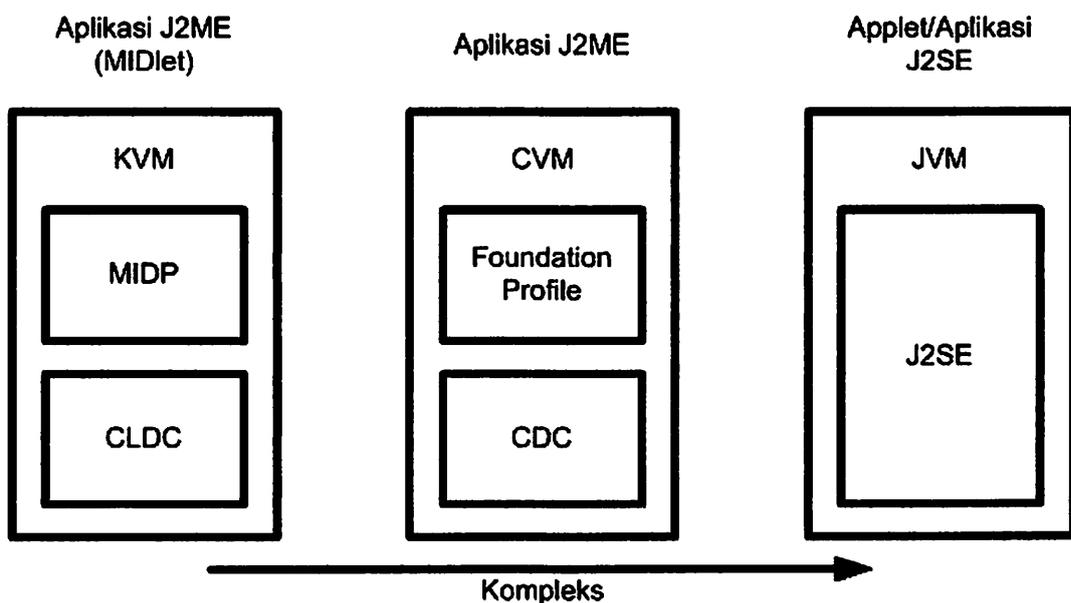
Configuration merupakan *Java library* minimum dan kapasitas yang dipunya oleh para pengembang JME, yang maksudnya sebuah *mobile device* dengan kemampuan *Java* akan dioptimalkan untuk menjadi sesuai. *Configuration* hanyalah mengatur hal-hal tentang kesamaan sehingga dapat dijadikan ukuran kesesuaian antar *device*. Misalnya sebuah lampu sepeda dirancang sedemikian sehingga dapat digunakan oleh berjenis-jenis sepeda. Dalam J2ME telah didefinisikan dua buah konfigurasi yaitu CLDC (*Connected Limited Device Configuration*) untuk perangkat kecil dan CDC (*Connected Device Configuration*) untuk perangkat yang lebih besar. Lingkup CLDC dan CDC dapat dilihat dari gambar dibawah ini [10]:



Gambar 2.2 lingkup Configuration

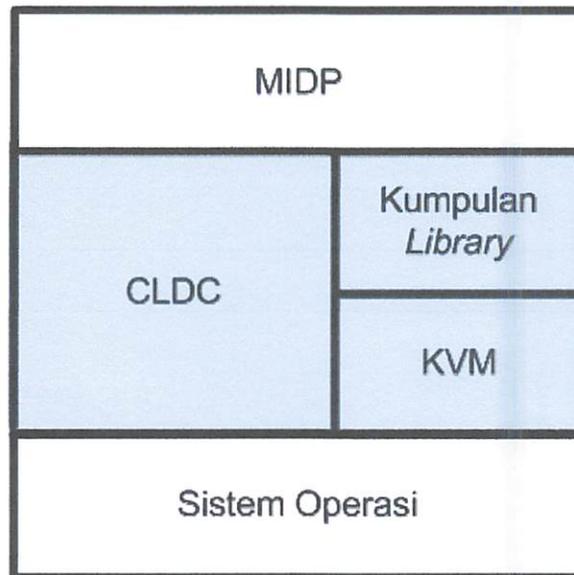
Profile berbeda dengan *configuration*, *profile* membahas sesuatu yang spesifik untuk sebuah perangkat. Sebagai contoh misalnya, sebuah sepeda dengan merk tertentu, tentunya mempunyai ciri spesifik dengan sepeda lain.

Dalam gambar J2ME terdapat dua buah *profile* yaitu MIDP dan *Foundation Profile*. Keterhubungan antara *configuration* dan *profile* yang ada pada J2ME beserta jenis mesin virtualnya dapat dilihat pada gambar berikut [10]:



Gambar 2.3 Hubungan J2ME dan J2SE

2.5 *Connected Limited Device Configuration (CLDC)*



Gambar 2.4 Arsitektur CLDC

CLDC atau *Connected Limited Device Configuration* adalah perangkat dasar dari J2ME, spesifikasi dasar yang berupa *library* dan API yang diimplementasikan pada J2ME, seperti yang digunakan pada telepon selular, pager dan PDA. Perangkat tersebut dibatasi dengan keterbatasan memori, sumber daya dan kemampuan memproses. Spesifikasi CLDC pada J2ME adalah spesifikasi minimal dari *package*, kelas dan sebagian fungsi *Java Virtual Machine* yang dikurangi agar dapat diimplementasikan dengan keterbatasan sumber daya pada alat-alat tersebut, JVM yang digunakan disebut KVM (*Kilobyte Virtual Machine*). Posisi CLDC pada arsitektur J2ME dapat dilihat pada gambar di atas.[10]

2.6 *Connected Device Configuration (CDC)*

CDC atau *Connected Device Configuration* adalah spesifikasi dari konfigurasi J2ME. CDC merupakan komunitas proses pada *Java* yang mempunyai standarisasi. CDC terdiri dari virtual machine dan kumpulan *library* dasar untuk dipergunakan pada profile industri. Implementasi CDC pada J2ME adalah *source code* yang menyediakan sambungan dengan macam-macam *platform*.[10]

Perbedaan antara CDC dan CLDC dapat dilihat dari tabel dibawah ini [10]:

Table 2.1 Tabel Perbandingan Antara CDC dan CLDC

CLDC	CDC
Mengimplementasikan subset dari J2SE	Mengimplementasikan seluruh fitur pada J2SE
JVM yang digunakan dikenal dengan nama KVM	JVM yang digunakan dikenal dengan nama CVM
Digunakan pada perangkat handled dengan ukuran memori terbatas (160 – 512 Kbytes)	Digunakan pada perangkat handled dengan ukuran memori minimal 2 Mbytes
Prosesor : 16 Bit atau 32 Bit	Prosesor : 32 Bit

2.7 JDK (*Java Development Kit*) dan JVM (*Java Virtual Machine*)

2.7.1 JDK (*Java Development Kit*)

JDK merupakan kotak peralatan (kit) yang digunakan untuk men-develop. JDK berguna saat menulis code program. JDK memiliki JVM didalamnya[5]

2.7.2 JVM (*Java Virtual Machine*)

JVM adalah sebuah mesin imajiner (maya) yang bekerja dengan menyerupai aplikasi pada sebuah mesin nyata. JVM menyediakan spesifikasi hardware dan platform dimana kompilasi kode java terjadi. Spesifikasi inilah yang membuat aplikasi berbasis java menjadi bebas dari platform manapun karena proses kompilasi diselesaikan oleh JVM. Aplikasi program java diciptakan dengan *file* teks berektensi *.java*. Program ini dikompilasi menghasilkan satu berkas *bytecode* berektensi *.class* atau lebih. *Bytecode* adalah serangkaian instruksi serupa instruksi kode mesin. Perbedaannya adalah kode mesin harus dijalankan pada system komputer dimana kompilasi ditujukan, sementara *bytecode* berjalan pada *java interpreter* yang tersedia di semua *platform* system komputer dan system operasi.[5]

2.8 NetBeans

NetBeans adalah integrated Development Environment (IDE) berbasis Java dari Sun Microsystem yang berjalan diatas Swing. Swing sebuah teknologi Java untuk pengembangan aplikasi Desktop yang dapat berjalan di berbagai macam platform seperti Windows, Linux, Mac OS X dan Solaris. Netbeans merupakan software development yang Open Source, dengan kata lain software di bawah pengembangan bersama, bebas biaya. Didalam netbeans tersedia fungsi dalam pembuatan aplikasi game mobile yang dikenal dengan nama J2ME. Dalam pembuatan aplikasi J2ME memiliki beberapa prinsip kerja, yaitu [5]:

2.8.1 Class MIDlet

MIDlet merupakan sebuah kelas yang pertama kali penulis buat dalam membangun sebuah game, karena midled adalah kelas utama atau yang sering dikenal dengan main class yang digunakan untuk memanggil kelas-kelas lainnya.[5]

2.8.2 Class splash

Splash adalah sebuah tampilan pembuka dalam sebuah game mobile. Splash ditampilkan sesaat dengan durasi waktu yang sangat singkat. Biasanya splash digunakan untuk menampilkan informasi vendor pembuat game atau kata-kata selamat datang.[5]

2.8.3 Class Menu

Pada kelas ini penulis menggunakan sebagai media optional yang didalamnya terdapat informasi, konfigurasi, serta keluar dari game. Kelas ini akan muncul setelah splash screen dan untuk memunculkannya penulis melakukan konfigurasi pada main class.[5]

2.8.4 Class Counter

Kelas yang digunakan untuk melakukan suatu perhitungan yang terprogram. Yang biasa digunakan untuk countdown.[5]

2.8.5 Class Game

Kelas yang terdapat pada netbeans guna memudahkan dalam memanipulasi tampilan grafis yang akan digunakan dalam sebuah game

mobile. Pada kelas ini semua source grafis dapat di panggil yang nantinya dipanggil melalui class engine.[5]

2.8.6 Class engine

Klas yang digunakan penilus untuk membuat rule atau ketentuan game yang dituangkan dalam bahasa pemrograman java. Pada kelas ini juga memanggil source grafis yang sebelumnya telah di buat dengan kelas game design. Kelas ini juga bertugas sebagai perantara perintah yang terprogram dengan gerakan player pada layar, sehingga kelas ini adalah kelas yang menunjang permainan dapat berjalan atau tidak.[5]

BAB III

ANALISA DAN PERANCANGAN

Proses analisa yang dilakukan pada sistem meliputi perangkat lunak dan perangkat keras yang digunakan untuk membangun aplikasi game. Dalam proses analisa juga menggambarkan alur kerja aplikasi dalam bentuk flow chart.

3.1 Skenario Game

Game ini merupakan game edukasi yang bertujuan untuk memperkenalkan beberapa perangkat jaringan yang digunakan dalam kehidupan sehari-hari. Dalam perancangan *game* edukasi ini dibuat skenario game terlebih dahulu.

Didalam *game* ini terdapat sebuah karakter yang dapat digunakan oleh pemain. Awalmula permainan karakter akan berdiri dalam sebuah ruangan. Tujuan karakter adalah untuk membuat jaringan sederhana. Untuk membuat sebuah jaringan sederhana tersebut, karakter harus memperbaiki permasalahan-permasalahan yang terdapat pada permainan. Dengan menyelesaikan permasalahan yang terdapat pada permainan maka jaringan sederhana dapat digunakan.

3.2 Analisis Sistem

Spesifikasi sistem yang digunakan untuk membangun aplikasi yang akan dibuat antara lain perangkat keras (*hardware*) dan perangkat lunak (*software*), dan dalam implementasi dan pengujian sistem ini ada beberapa perangkat keras dan perangkat lunak yang digunakan.

3.2.1 Spesifikasi Aplikasi

Aplikasi game edukatif merupakan suatu aplikasi yang digunakan sebagai media pembelajaran dengan cara memainkan aplikasi yang telah terinstall di dalam perangkat handphone. Setiap adegan dalam permainan

menyajikan materi pembelajaran, khususnya pelajaran dasar-dasar jaringan.

3.2.2 Spesifikasi perangkat handphone

Untuk memainkan aplikasi permainan berbasis java diperlukan beberapa perangkat pendukung supaya aplikasikasi game dapat berjalan dengan baik. Perangkat handphone harus memiliki spesifikasi minimal berupa CLCD 1.0 dan MIDP 2.0.

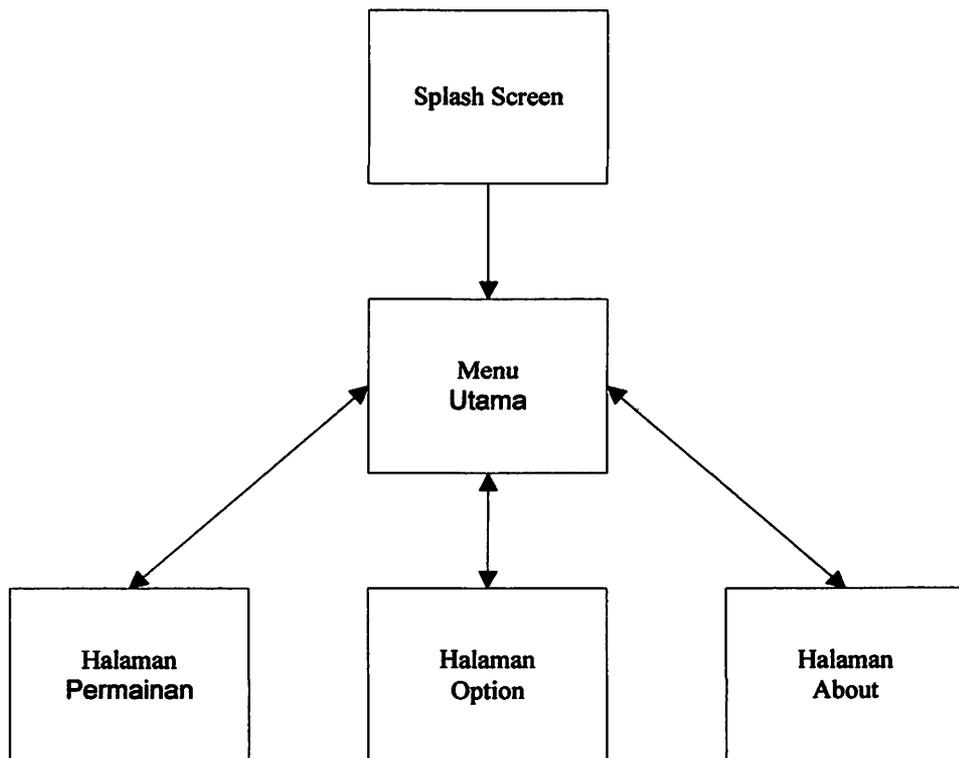
3.2.3 Spesifikasi perangkat keras (*hardware*) dan lunak (*Software*)

Berikut ini adalah spesifikasi perangkat baik *hardware* maupun *software* yang digunakan dalam pembuatan aplikasi game :

1. Perangkat keras (*hardware*)
 - a. CPU : *Intel Core i3 processor*
 - b. *Hardisk* : kapasitas 360 Gb.
 - c. *Memory* : 2Gb.
2. Perangkat lunak (*software*)
 - a. Sistem operasi *Windows 7*
 - b. *JDK 1.6.0_29*
 - c. *Netbeans 7.0.1*, sebagai java editor
 - d. *WTK 2.5.1*, sebagai emulator perangkat *mobile*

3.3 Perancangan sistem

3.3.1 Disain alur aplikasi



Gambar 3.1 Alur Aplikasi

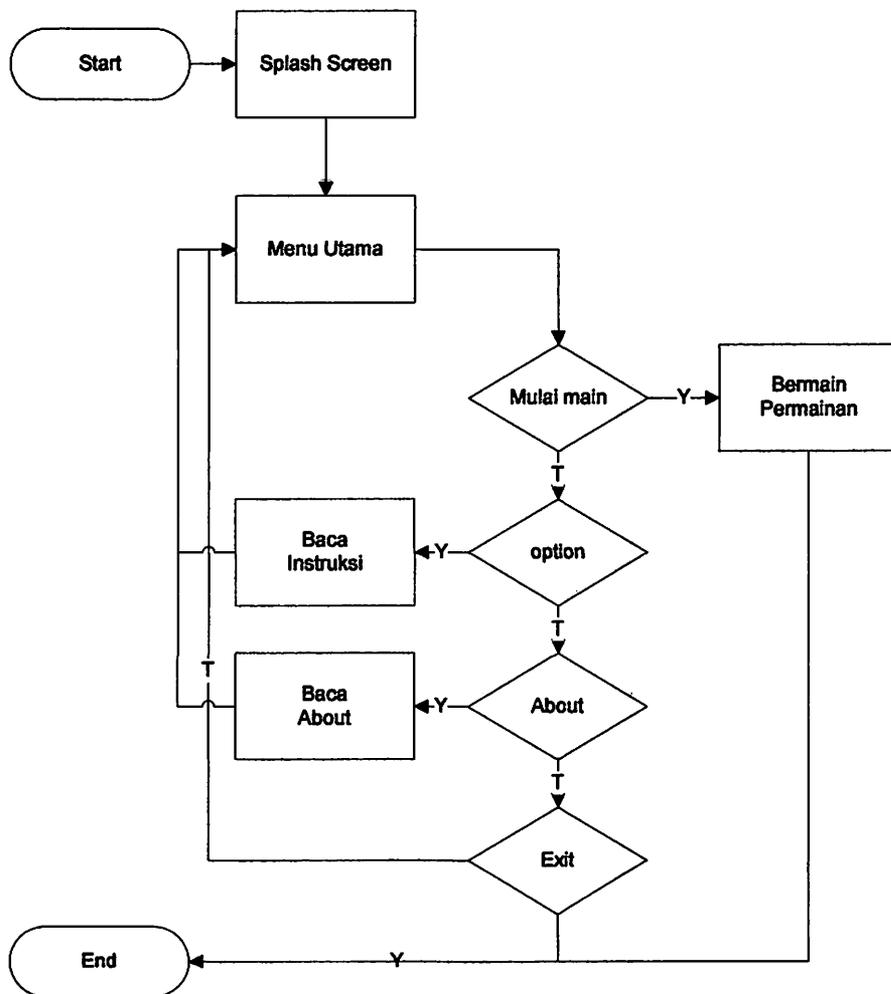
Penjelasan desain :

Halaman dibagi menjadi lima (5) bagian, *Splash Screen*, halaman utama (Main Menu), halaman permainan, halaman *instruction*, dan halaman *about*.

- a. *Splash Screen* : halaman pembuka yang ditunjukkan pertama kali kepada pengguna.
- b. Halaman utama (*Main Menu*) : halaman menu untuk navigasi utama aplikasi.

- c. Halaman Permainan : halaman inti pada aplikasi. Pengguna dapat menggerakkan objek yang telah ditunjuk sebelumnya menggunakan tombol pada perangkat handphone.
- d. Halaman *Option* : pada halaman ini pemain dapat menseting tingkat kesulitan dan suara pada permainan
- e. Halaman *about* : halaman ini berisi identitas atau keterangan pembuat aplikasi.

3.3.2 Flow chart



Gambar 3.2 Flow Chart

Berikut ini penjelasan tiap-tiap tahap dari flow chart :

1. Awal dari program dimulai dengan muncul *splash screen* di ikuti dengan sebuah *progress bar*.
2. Setelah itu user masuk ke dalam menu utama yang meliputi 4 menu yaitu :
 - a. Menu main baru.
 - b. Menu *instruction*.
 - c. Menu *about*.
 - d. Menu *exit*.
3. Jika menu yang dipilih bukan menu main baru, maka 3 menu yaitu *instructon*, *about*, dan *exit* dapat dipilih.
4. Jika user memilih menu main baru, user akan langsung memainkan *scene* permainan. yang tersedia dalam aplikasi.
5. Semua *scene* yang telah diselesaikan oleh user merupakan tanda dari selesainya permainan.
6. Jika user tidak ingin mengulang permainan, user dapat keluar dari aplikasi.

3.3.3 Rancangan menu

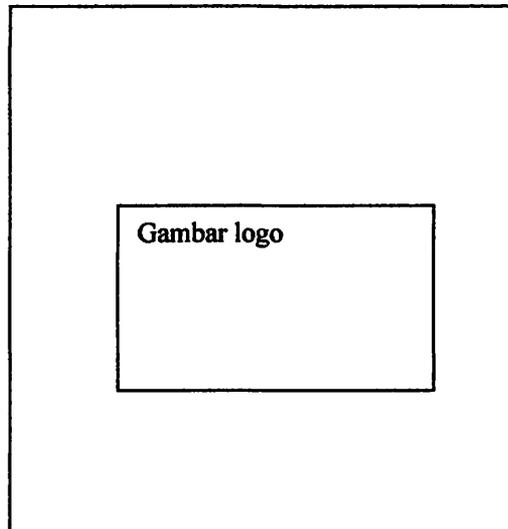
Model menu yang digunakan adlaah model *list* dengan *item-item* :

- a. Mulai Baru : menuju halaman permainan untuk memulai permainan.
- b. *Instruction* : menuju halaman *Instructon*.
- c. *About* : menuju halaman *about*.
- d. *Exit* : keluar dari permainan.

Item menu adalah kombinasi antara gambar *icon*, nama *item*, dan deskripsi singkat.

3.4 Perancangan antar muka

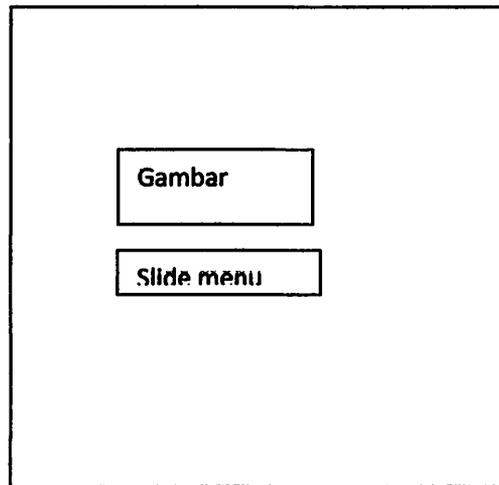
3.4.1 Halaman *Splash Screen*



Gambar 3.4 Halaman *Splash Screen*

Seperti tampak pada gambar 3.4. Halaman *splash screen* akan muncul pada awal user membuka aplikasi. Pada halaman *splash screen* terdapat objek gambar logo. Gambar logo merupakan logo dari permainan.

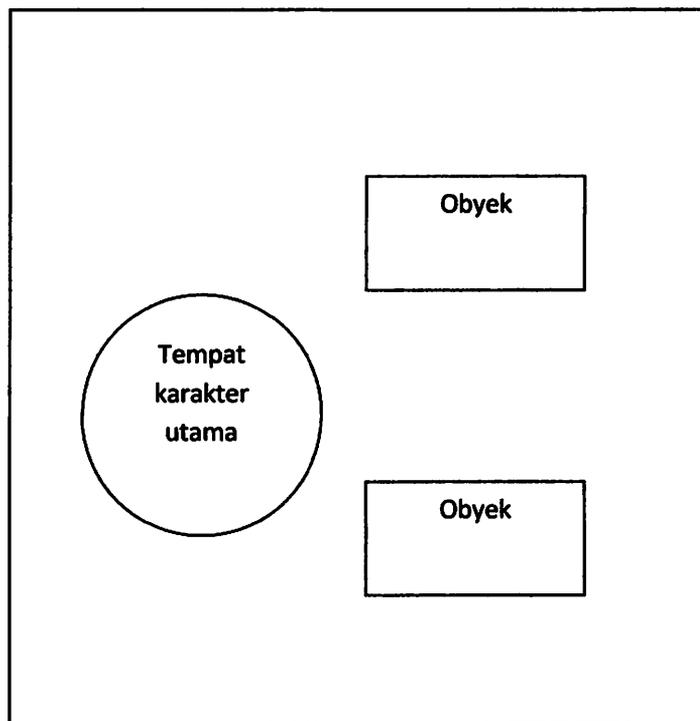
3.4.2 Halaman Menu



Gambar 3.5 Halaman Menu

Halaman menu akan keluar setelah proses *splash screen* selesai ditampilkan. Terdapat 4 menu yang dapat dipilih oleh user. Menu mulai permainan dapat digunakan user untuk memulai permainan baru. Menu *instruction* dan menu *about* berisikan teks penjelasan mengenai game dan penjelasan mengenai pembuat permainan. Menu keluar digunakan untuk keluar dari aplikasi permainan. Disamping 4 menu tersebut terdapat gambar logo untuk tiap-tiap menu yang tersedia.

3.4.3 Halaman ruang permainan



Gambar 3.6 Halaman ruang karakter utama

Seperti yang terlihat pada gambar 3.6. Halaman permainan merupakan awal permainan dimulai. Dalam ruangan tersebut karakter utama dapat menggunakan beberapa obyek yang telah ditunjuk sebelumnya untuk di gunakan user dalam permainan berlangsung.

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Kebutuhan Sistem

Spesifikasi sistem yang digunakan untuk membangun aplikasi yang akan dibuat antara lain perangkat keras (*hardware*) dan perangkat lunak (*software*), dan dalam implementasi dan pengujian system ini ada beberapa perangkat keras dan perangkat lunak yang digunakan antara lain:

4.1.1 Perangkat keras

Perangkat keras adalah bagian dari computer untuk menginputkan data dan menghasilkan output berupa informasi, serta komponen lainnya yang membantu system computer untuk menghasilkan bentuk-bentuk informasi. Perangkat keras yang digunakan dalam membuat aplikasi antara lain:

1. CPU : *Intel Core i3 processor*
2. *Harddisk* : kapasitas 360 Gb
3. *Memory* : 2 Gb
4. Modem GSM : K3520

4.1.2 Perangkat lunak

Perangkat lunak adalah program yang berfungsi untuk memproses inputan dari perangkat keras dan diolah dengan program computer untuk kemudian memberikan hasil informasi ke perangkat keras untuk ditampilkan. Program yang digunakan dalam implementasi dan pengujian antara lain:

1. Sistem operasi *windows 7 32 bit*
2. *JDK 1.6.0_29*
3. *Netbean 7.0.1*

4. *WTK 2.5.1*

5. *Java (TM) ME Platform SDK 3.0*

4.2 Penggunaan Aplikasi

Menjelaskan tampilan aplikasi yang mencakup tampilan menu, fungsi dan bagaimana menggunakannya agar dapat berjalan menghasilkan informasi dari input yang dihasilkan dari perangkat keras dan di proses inputan tersebut pada perangkat lunak.

4.2.1 Instalasi Wireless Toolkit

WTK digunakan sebagai tool utama untuk membuat aplikasi mobile. WTK dalam proyek ini menggunakan WTK versi 2.5.1. Cara penginstalan dilakukan sesuai petunjuk yang telah diberikan dalam aplikasi WTK tersebut.

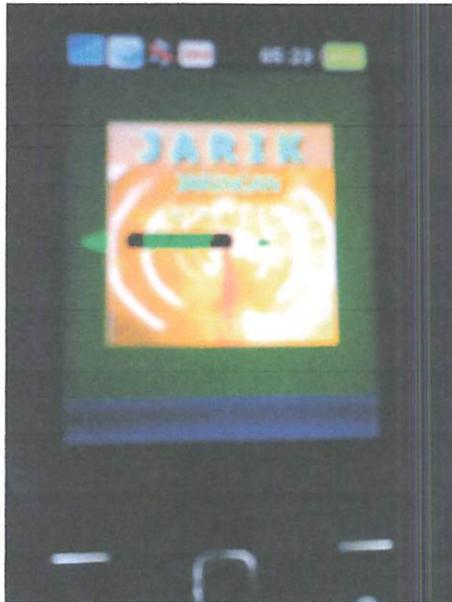
4.2.2 Spesifikasi Perangkat Mobile

Spesifikasi perangkat mobile yang digunakan dalam pengujian :

1. Sony Ericsson model k770i
2. Memory slot stick Micro, Space kosong 100MB
3. Bluetooth V2.0 with A2DP
4. Dimensi 105 x 47 x 14.5 mm
5. Layar 240 x 320 pixel, 1.9 inches
6. Warna layar TFT, 256K colors

4.2.2 Hasil Implementasi

Adapun tahap-tahap yang dilakukan untuk pengujian game pada perangkat mobil sebagai berikut :



Gambar 4.1 Tampilan Awal

Gambar diatas merupakan tampilan menu awal pada game. Untuk pemilihan menu, dapat digunakan tombol arah pada perangkat handphone.



Gambar 4.2 Tampilan Fungsi Pengaturan

Pada tampilan ini, terdapat fungsi pengaturan yang terdapat pada game. Pemain dapat mengaktifkan atau menonaktifkan suara pada permainan



Figure 1: [Illegible text]

[Illegible text]



Figure 2: [Illegible text]

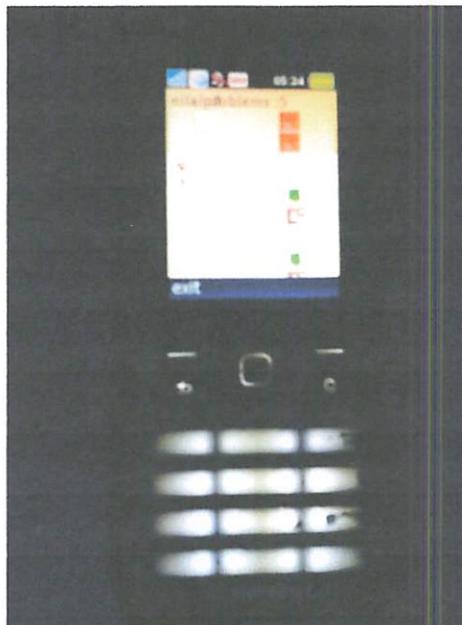
[Illegible text]

dengan cara memilih salah satu fungsi yang telah terdapat pada menu tersebut.



Gambar 4.3 Tampilan Pengisian

Pada tampilan diatas terdapat sebuah kota kosong dapat diisi sebuah kata oleh pemain. Kata tersebut berupa nama pemain. Pengisian nama dapat dilakukan menggunakan tombol yang terdapat pada perangkat handphone seperti menggunakan seperti biasa.

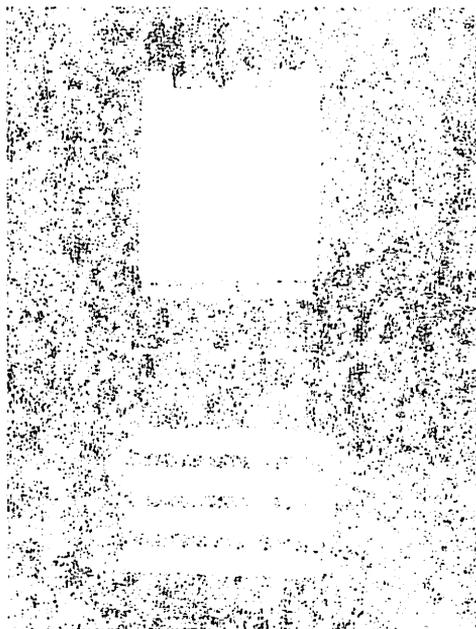


... ..



... ..

... ..



Gambar 4.4 Tampilan Halaman Permainan

Tampilan diatas merupakan tampilan dari permainan. Terdapat karakter, background, dan beberapa item pendukung dalam permainan. Karakter dapat dijalankan oleh pemain untuk menyelesaikan misi yang terdapat pada permainan.

4.3 Pengujian Fungsional Sistem

Pengujian dilakukan dengan menjalankan aplikasi pada perangkat selular. Perangkat selular yang digunakan dalam pengujian adalah Sony Ericsson model k770i dengan meliputi beberapa bagian yang terdapat pada aplikasi yang telah dibuat. Hasil pengujian fungsional dapat dilihat pada Tabel 4.1.

No.	Fungsi yang Diuji	Hasil	
		Berjalan	Tidak Berjalan
1.	Tampilan awal aplikasi	√	
2.	Pengaturan aplikasi	√	
3.	Tampilan pengisian	√	
4.	Tampilan permainan	√	

Kesimpulan dari hasil pengujian adalah aplikasi dapat dijalankan para perangkat seluar. Pada pengujian 4 bagian aplikasi pada perangkat selular Sony Ericson k770i dapat berjalan.

BAB V

PENUTUP

5.1 Kesimpulan

Setelah melakukan perancangan dan pengujian pada aplikasi game dapat diambil beberapa kesimpulan antara lain :

1. Hasil uji aplikasi pada perangkat selular Sony Ericson k770i, beberapa gambar masih terlihat terpotong dari layar.
2. Penggunaan perangkat yang memiliki ukuran layar lebih besar dari perangkat selular Sony Ericson k770i akan mengalami kerusakan pada gambar, gambar menjadi buram.

5.2 Saran

1. Aplikasi *game* yang telah dibuat ini dapat dikembangkan dari segi kesulitan permainan. Dalam misi yang dijalankan bias ditambahkan sebuah pertanyaan untuk menambah kesulitan bermain.
2. Gambar pada *game* dapat menyesuaikan dengan pengguna perangkat selain Sony ericson k770i.
3. Memperbaiki tampilan supaya gambar tidak terlihat seperti terpotong.

DAFTAR PUSTAKA

- [1] Anggara. 2008. *Memahami Teknik Dasar Pembuatan Game Berbasis Flash*. Yogyakarta: Penerbit Gava Media.
- [2] Anuff, Ed. 1996. *Java Sourcebook – Pemuntun Pemrograman Java*. Terjemahan oleh Bambang Wisudawan. 1997. Yogyakarta : Penerbit Andi Offset.
- [3] Purnama, Rangsang. 2007. *Pemrograman GUI Menggunakan Java*. Cetakan ke1. Jakarta : Penerbit Perstasi Pustaka Publisher.
- [4] Taufiq, Andik. 2010. *Pemrograman Grafik Java dengan Java*. Badung : Penerbit Informatika.
- [5] Artikel non-personal , 24 Januari 2014, “Game”, <http://en.wikipedia.org/wiki/Game> (diakses pada tanggal 24 Januari 2014).
- [6] Artikel non-personal , 24 Januari 2014, “Role Playing Game”, http://en.wikipedia.org/wiki/Role-playing_game (diakses pada tanggal 24 Januari 2014).
- [7] Artikel non-personal , 4 September 2013, “Java”, <http://en.wikipedia.org/wiki/Game> (diakses pada tanggal 24 Desember 2013).
- [8] Artikel non-personal, 27 Januari 2014, “J2ME”, <http://en.wikipedia.org/wiki/J2me> (diakses pada tanggal 27 Januari 2014)
- [9] *Java 2 Micro edition (J2ME)*. <http://elib.unikom.ac.id/files/disk1/435/jbptunikompp-gdl-hasbyfachr-21744-7-12.unik-i.pdf>, (diakses pada tanggal 2 Desember 2013).
- [10] Satria, Romi Wahono. “Tips-tips J2ME”. <http://ilmukomputer.org/2006/09/28/tips-tips-j2me> (diakses pada tanggal 26 April 2012).

LAMPIRAN



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Raya Karanglo Km. 2 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

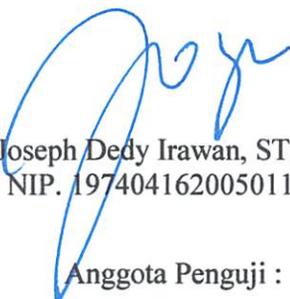
Nama : Akhmad Alfian Z.S
NIM : 0818186
Jurusan : Teknik Informatika S-1
Judul : Rancang Bangun Game Menggunakan Teknologi J2ME

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

Hari : Senin
Tanggal : 18 Februari 2013
Nilai :

Panitia Ujian Skripsi :

Ketua Majelis Penguji



Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002

Anggota Penguji :

Penguji Pertama



Sandy Nataly Mantja, SKom
NIP.P. 1030800418

Penguji Kedua



Karina Auliasari, ST, M.Eng
NIP.P. 1031000426



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Raya Karanglo Km. 2 Malang

FORMULIR PERBAIKAN SKRIPSI

Nama : Akhmad Alfian Z.S
NIM : 0818186
Jurusan : Teknik Informatika S-1
Judul : Rancang Bangun Game Menggunakan Teknologi J2ME

Tanggal	Penguji	Uraian	Paraf
18 Februari 2013	I	-.Batasan Masalah -.Hapus sejarah game -.Menganti buton dengan angka/huruf -.Menjalankan program pada perangkat hp -.Perbaiki kesimpulan dan saran -.Pengurutan penulisan laporan -.Menambahkan daftar pustaka	
18 Februari 2013	II	-.Perbaiki BAB I -.Perbaiki BAB II -.Perbaiki BAB III -.Perbaiki BAB IV -.Menjalankan program pada perangkat hp	

Anggota Penguji :

Penguji Pertama

Sandy Nalaty Mantja, SKom
NIP.P. 1030800418

Penguji Kedua

Karina Auliasari, ST, M.Eng
NIP.P. 1031000426

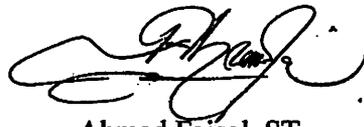
Mengetahui

Dosen Pembimbing I



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

Dosen Pembimbing II —



Ahmad Faisol, ST
NIP. 1031000413



FORMULIR BIMBINGAN SKRIPSI

Nama : Akhmad Alfian Zuhro Saputro
Nim : 0818186
Masa Bimbingan : 18 April 2012 s/d 18 Oktober 2012
19 Oktober 2012 s/d 19 April 2013
Judul Skripsi : Rancang Bangun Game Menggunakan Teknologi J2ME

No	Tanggal	Uraian	Paraf Pembimbing
1	9-6-2012	Revisi Bab I Sempurnakan rumusan masalah, tujuan, batasan masalah	
2	16-6-2012	Tambah materi tentang game	
3	25-6-2012	Rubah rancangan sistem	
4	22-10-2012	Pertajam rumusan rumusan masalah dan tujuan	
5	22-10-2012	Tambahkan scenario game	
6	4-2-2013	Acc laporan skripsi	
7			
8			
9			
10			

Malang, 14 Februari 2013
Dosen Pembimbing I

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



FORMULIR BIMBINGAN SKRIPSI

Nama : Akhmad Alfian Zuhro Saputro
Nim : 0818186
Masa Bimbingan : 18 April 2012 s/d 18 Oktober 2012
19 Oktober 2012 s/d 19 April 2013
Judul Skripsi : Rancang Banging Game Menggunakan Teknologi J2ME

No	Tanggal	Uraian	Paraf Pembimbing
1	11-5-2012	Bab I, dilakukan sedikit cek ulang Bab II & III	
2	22-1-2013	Acc makalah seminar hasil	
3	4-2-2013	Acc laporan skripsi	
4			
5			
6			
7			
8			
9			
10			

Malang, 14 Februari 2013
Dosen Pembimbing II

Ahmad Faisol, ST
NIP.P. 1031000431

Form S-4b

```
import binary.Midlet;
import java.util.TimerTask;
import javax.microedition.lcdui.game.Sprite;
```

```
public final class a extends TimerTask
{
```

```
    private final n a;
```

```
    public a(n n1)
```

```
    {
```

```
        a = n1;
```

```
    }
```

```
    public final void run()
```

```
    {
```

```
        n.a(a).a("/outdoor.mid", -1);
```

```
        n.a(a).addScore(n.a(a));
```

```
        n.b(a, false);
```

```
        n.a(a).setPosition(19, 126);
```

```
        n.a(a, false);
```

```
        return;
```

```
        JVM INSTR dup ;
```

```
        Exception exception;
```

```
        exception;
        printStackTrace();
    }
}

import binary.Midlet;
import java.util.TimerTask;

public final class b extends TimerTask
{

    private final n a;

    public b(n n1)
    {
        a = n1;
    }

    public final void run()
    {
        n.a(a, false);
        n.a(a).addScore(n.a(a));
        n.b(a, false);
    }
}
```

```
import binary.Midlet;
import java.util.TimerTask;

public final class c extends TimerTask
{

    private final n a;

    public c(n n1)
    {
        a = n1;
    }

    public final void run()
    {
        n.a(a, false);
        n.a(a).addScore(n.a(a));
        n.b(a, false);
    }
}

import java.io.ByteArrayInputStream;
import java.io.DataInputStream;
import javax.microedition.rms.RecordComparator;
```

```
public final class d
    implements RecordComparator
{

    private byte a_byte_array1d_fld[];
    private ByteArrayInputStream a_java_io_ByteArrayInputStream_fld;
    private DataInputStream a_java_io_DataInputStream_fld;

    public d()
    {
        a_byte_array1d_fld = new byte[10];
        a_java_io_ByteArrayInputStream_fld = null;
        a_java_io_DataInputStream_fld = null;
    }

    public final void a()
    {
        try
        {
            if(a_java_io_ByteArrayInputStream_fld != null)
            {
                a_java_io_ByteArrayInputStream_fld.close();
            }

            if(a_java_io_DataInputStream_fld != null)
```

```
    {  
        a_java_io_DataInputStream_fld.close();  
    }  
    return;  
}  
catch(Exception _ex)  
{  
    return;  
}  
}
```

```
public final int compare(byte abyte0[], byte abyte1[])
```

```
{  
    int i;  
    int j;  
    int k;  
    if((k = Math.max(abyte0.length, abyte1.length)) > a_byte_array1d_fld.length)  
    {  
        a_byte_array1d_fld = new byte[k];  
    }  
    a_java_io_ByteArrayInputStream_fld = new ByteArrayInputStream(abyte0);  
    a_java_io_DataInputStream_fld = new  
DataInputStream(a_java_io_ByteArrayInputStream_fld);  
    i = a_java_io_DataInputStream_fld.readInt();
```

```
a_java_io_ByteArrayInputStream_fld = new ByteArrayInputStream(a_byte1);
a_java_io_DataInputStream_fld = new
DataInputStream(a_java_io_ByteArrayInputStream_fld);
j = a_java_io_DataInputStream_fld.readInt();
if(i == j)
{
    return 0;
}
if(i > j)
{
    return -1;
}
return 1;
JVM INSTR pop ;
return 0;
}
}
import java.io.PrintStream;
import java.util.Vector;
import javax.microedition.rms.RecordStore;
import javax.microedition.rms.RecordStoreException;

public final class e
{
```

```
public RecordStore a_javax_microedition_rms_RecordStore_fld;
private static String a_java_lang_String_static_fld = "highscore";
private RecordStore b;

public e()
{
    new Vector();
}

public final void a(int i)
{
    try
    {
        byte abyte0[] = a(i);

        a_javax_microedition_rms_RecordStore_fld =
RecordStore.openRecordStore("kjgame", false);

        a_javax_microedition_rms_RecordStore_fld.setRecord(1, abyte0, 0,
abyte0.length);

        a_javax_microedition_rms_RecordStore_fld.closeRecordStore();

        return;
    }
    catch(RecordStoreException _ex) { }
    try
    {
```

```
        byte abyte1[] = a(i);

        a_javax_microedition_rms_RecordStore_fld =
RecordStore.openRecordStore("kjgame", true);

        a_javax_microedition_rms_RecordStore_fld.addRecord(abyte1, 0,
abyte1.length);

        a_javax_microedition_rms_RecordStore_fld.closeRecordStore();

        return;
    }

    catch(RecordStoreException recordstoreexception)
    {
        System.out.println(recordstoreexception);
    }
}
```

```
public final void b(int i)
{
    try
    {
        byte abyte0[] = a(i);

        a_javax_microedition_rms_RecordStore_fld =
RecordStore.openRecordStore("kjgame", false);

        a_javax_microedition_rms_RecordStore_fld.setRecord(2, abyte0, 0,
abyte0.length);

        a_javax_microedition_rms_RecordStore_fld.closeRecordStore();

        return;
    }
}
```

```
catch(RecordStoreException _ex) { }

try
{
    byte abyte1[] = a(i);

    a_javax_microedition_rms_RecordStore_fld =
RecordStore.openRecordStore("kjgame", true);

    a_javax_microedition_rms_RecordStore_fld.addRecord(abyte1, 0,
abyte1.length);

    a_javax_microedition_rms_RecordStore_fld.closeRecordStore();

    return;
}

catch(RecordStoreException recordstoreexception)
{
    System.out.println(recordstoreexception);
}
}
```

```
private static byte[] a(int i)
{
    String s;

    return (s = Integer.toString(i)).getBytes();
}
```

```
public final int a()
{
```

```

String s;

a_javax_microedition_rms_RecordStore_fld =
RecordStore.openRecordStore("kjgame", false);

byte abyte0[] = a_javax_microedition_rms_RecordStore_fld.getRecord(1);
s = new String(abyte0, 0, abyte0.length);
a_javax_microedition_rms_RecordStore_fld.closeRecordStore();

int i;

return i = Integer.parseInt(s);

JVM INSTR dup ;

RecordStoreException recordstoreexception;
recordstoreexception;
printStackTrace();
System.out.println("exception rms volum");
a(1);
return 1;
}

```

```

public final int b()
{
String s;

a_javax_microedition_rms_RecordStore_fld =
RecordStore.openRecordStore("kjgame", false);

byte abyte0[] = a_javax_microedition_rms_RecordStore_fld.getRecord(2);
s = new String(abyte0, 0, abyte0.length);
a_javax_microedition_rms_RecordStore_fld.closeRecordStore();

```

```
int i;  
return i = Integer.parseInt(s);  
JVM INSTR dup ;  
RecordStoreException recordstoreexception;  
recordstoreexception;  
printStackTrace();  
System.out.println("exception di rms getLevel");  
b(2);  
return 2;  
}
```

```
public final void c(int i)  
{  
    Vector vector = a();  
    int j = 0;  
    do  
    {  
        if(j >= 5)  
        {  
            break;  
        }  
        int l = Integer.parseInt(vector.elementAt(j).toString());  
        int k1;  
        if((k1 = i) > l)
```

```

    {
        vector.insertElementAt(k1 + "", j);
        break;
    }
    j++;
} while(true);
b = RecordStore.openRecordStore(a_java_lang_String_static_fld, true);
try
{
    for(int k = 0; k < vector.size(); k++)
    {
        int i1;
        byte abyte0[] = a(i1 = Integer.parseInt(vector.elementAt(k).toString()));
        b.setRecord(k, abyte0, 0, abyte0.length);
    }
}
catch(Exception _ex)
{
    for(int j1 = 0; j1 < vector.size(); j1++)
    {
        int l1;
        byte abyte1[] = a(l1 = Integer.parseInt(vector.elementAt(j1).toString()));
        b.addRecord(abyte1, 0, abyte1.length);
    }
}

```

```

        System.out.println("added = " + j1 + " = " + 11);
    }

}

b.closeRecordStore();

return;

JVM INSTR dup ;

RecordStoreException recordstoreexception;

recordstoreexception;

printStackTrace();

}

private Vector a()
{
    int ai[] = a();
    Vector vector = new Vector();
    for(int i = 0; i < ai.length; i++)
    {
        System.out.println("di get rekord store" + ai[i]);
        vector.addElement(ai[i] + "");
    }

    return vector;
}

```

```
private int[] a()
{
    int ai[];

    b = RecordStore.openRecordStore(a_java_lang_String_static_fld, true);
    ai = new int[5];
    for(int i = 1; i <= 5; i++)
    {
        try
        {
            byte abyte0[] = b.getRecord(i);
            String s;
            int j = Integer.parseInt(s = new String(abyte0, 0, abyte0.length));
            ai[i - 1] = j;
        }
        catch(RecordStoreException _ex)
        {
            ai[i - 1] = 0;
        }
    }

    b.closeRecordStore();

    return ai;
}
JVM INSTR pop ;
```

```
        return null;
    }

}

import java.util.TimerTask;
import javax.microedition.lcdui.game.Sprite;

public final class f extends TimerTask
{

    public Sprite a_javax_microedition_lcdui_game_Sprite_fld;
    public boolean a_boolean_fld;

    private f(o o1, Sprite sprite)
    {
        a_javax_microedition_lcdui_game_Sprite_fld = sprite;
        a_boolean_fld = true;
    }

    public final void run()
    {
        if(a_boolean_fld)
        {
            a_javax_microedition_lcdui_game_Sprite_fld.nextFrame();
        }
    }
}
```

```
    }  
}
```

```
// Unreferenced inner class i
```

```
/* anonymous class */
```

```
final class i extends TimerTask
```

```
{
```

```
    private final o a;
```

```
    public final void run()
```

```
    {
```

```
        a.a(true);
```

```
    }
```

```
    public
```

```
    {
```

```
        a = final_o1;
```

```
    }
```

```
}
```

```
}
```

```
import binary.Midlet;
import java.util.TimerTask;

public final class g extends TimerTask
{

    private final q a;

    public g(q q1)
    {
        a = q1;
    }

    public final void run()
    {
        a.a.exit();
    }
}
```