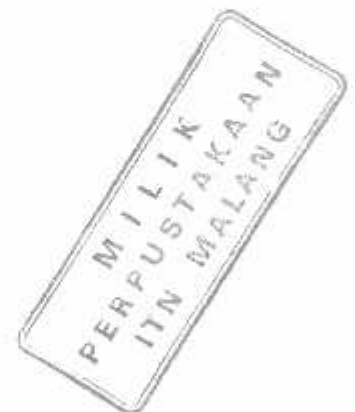


**RANCANG BANGUN MULTIPLAYER GAME PUZZLE GAMBAR
VIA BLUETOOTH BERBASIS MOBILE**

SKRIPSI



Disusun Oleh :
PETRUS SILALAH
06.12.582



**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2011**

LEMBAR PERSETUJUAN
RANCANG BANGUN MULTIPLAYER GAME PUZZLE GAMBAR
VIA BLUETOOTH BERBASIS MOBILE

SKRIPSI

*Disusun dan Diajukan Untuk Melengkapi dan Memenuhi Syarat
Guna Mencapai Gelar Sarjana Teknik*

Disusun oleh :

PETRUS SILALAH

NIM: 06.12.582


Mengetahui,
Ketua Jurusan T. Elektro S-1



Ir. Yusuf Ismail Nakhoda, MT
NIP. Y.1018890189

Diperiksa dan Disetujui,

Dosen Pembimbing I

Dosen Pembimbing II


Ir. Eko Nurcahyo
NIP. Y.1028700172


Michael Ardita ST, MT
NIP. Y.1031000434

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2011

ABSTRAKSI

RANCANG BANGUN MULTIPLAYER GAME PUZZLE GAMBAR VIA BLUETOOTH BERBASIS MOBILE

Petrus Silalahi, 06.12.582

Email : petrus137@gmail.com

Jurusan Teknik Elektro

Konsentrasi Teknik Komputer & Informatika S-1

Fakultas Teknologi Industri

Institut Teknologi Nasional Malang

Jl. Raya Karanglo Km 2 Malang

Dosen pembimbing I : Ir. Eko Nurcahyo

Dosen pembimbing II : Michael Ardita ST, MT

Tugas akhir ini membahas tentang cara membuat aplikasi dengan menggunakan salah satu subset dari Java, yaitu Java 2 Micro Edition. J2ME ini memungkinkan bagi seorang programmer untuk membuat aplikasi dengan jumlah memory yang terbatas dan biasa diimplementasikan pada handheld device seperti Handphone, PDA dan sebagainya. Untuk menggunakan fasilitas ini, seorang programmer hanya diharuskan memasukkan unit-unit J2ME pada aplikasi yang akan dibuatnya. Fasilitas ini dapat diaktifkan dengan menuliskan kelas dan fungsi yang telah ada sebelumnya dalam J2ME. Dalam Contoh aplikasi yang disertakan adalah GAME PUZZLE, yaitu sebuah game sederhana yang terdiri dari beberapa potongan gambar dan nantinya potongan-potongan gambar tersebut harus disusun agar menjadi sebuah gambar yang utuh dan game puzzle dapat dimainkan secara bersamaan dengan menggunakan fungsi bluetooth yang ada pada handphone.

Kata Kunci : Multiplayer puzzle, bluetooth, mobile phone.

KATA PENGANTAR

Dengan mengucapkan syukur kepada Tuhan Yang Maha Esa yang dengan segala rahmat dan anugerah – Nya, telah memberikan kekuatan, kesabaran, bimbingan dan perlindungan sehingga penulis dapat menyelesaikan laporan skripsi dengan judul :

RANCANG BANGUN MULTIPLAYER GAME PUZZLE GAMBAR VIA BLUETOOTH BERBASIS MOBILE

Pembuatan skripsi ini disusun guna memenuhi syarat akhir kelulusan pendidikan jenjang Strata I di Institut Teknologi Nasional Malang. Dalam penyusunan skripsi ini penulis banyak mendapat bantuan baik moril maupun materiil, saran dan dorongan semangat dari berbagai pihak, untuk itu penulis mengucapkan terima kasih kepada :

1. Bapak Ir. Soeparno Djiwo, MT selaku Rektor ITN Malang
2. Bapak Ir. Sidik Noertjahjono, MT selaku Dekan Fakultas Teknologi Industri.
3. Bapak Ir. Yusuf Ismail Nakhoda, MT selaku Ketua Jurusan Teknik Elektro S-1 ITN Malang.
4. Bapak Ir. Eko Nurcahyo & Bapak Michael Ardita, ST, MT selaku Dosen Pembimbing.
5. Kedua orangtua yang telah memberikan dukungan yang tiada hentinya.
6. Semua teman-teman dan semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih banyak yang perlu disempurnakan. Oleh sebab itu kritik dan saran yang membangun sangat diharapkan.

Akhir kata, penulis mohon maaf kepada semua pihak bilamana selama penyusunan skripsi ini penyusun membuat kesalahan secara tidak sengaja atau menyinggung pihak lain. Semoga skripsi ini dapat bermanfaat bagi kita semua.

Malang, Agustus 2011

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAK	ii
KATA PENGANTAR	iii
DAFTAR ISI	iv
DAFTAR GAMBAR	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	1
1.3 Batasan Masalah.....	2
1.4 Tujuan Masalah.....	2
1.5 Metodologi Penelitian.....	2
1.6 Sistematika Penulisan.....	2
BAB II TINJAUAN PUSTAKA	4
2.1 <i>Puzzle</i>	4
2.2 <i>Java</i>	4
2.3 <i>Java 2 Micro Edition (J2ME)</i>	5
2.3.1 <i>J2ME configuration (konfigurasi)</i>	6
2.3.2 Profil J2ME.....	7
2.3.3 MDIP dan MIDlet.....	7
2.3.3.1 Daur Hidup MIDlet.....	7
2.3.3.2 AntarMuka Pemakai.....	8
2.3.4 Pemaketan Aplikasi MIDlet.....	9

2.3.5 Perangkat Perkembangan J2ME	9
2.4 <i>Bluetooth</i>	10
2.4.1 Teknologi <i>Bluetooth</i>	10
2.4.2 <i>Bluetooth Protocol Stack</i>	11
2.4.2.1 Jaringan <i>Bluetooth</i>	13
2.4.2.2 <i>Profile Bluetooth</i>	14
2.4.3 <i>Spread Spectrum dan Frequency Hopping</i>	14
2.4.4 Aplikasi dan Layanan.....	15
2.5 <i>Mobile (handphone)</i>	16
2.6 Metode SWAP	16
2.7 <i>Flowchart</i>	17
2.7.1 <i>Terminator</i>	18
2.7.2 <i>Prosesing</i>	18
2.7.3 <i>Decision</i>	18
2.7.4 <i>Preparation Symbol</i>	18
2.7.5 <i>Connector Symbol</i>	19
2.7.6 <i>Input/Output Symbol</i>	19
2.7.7 <i>Flow Line Symbol</i>	19
BAB III PERANCANGAN DAN DESAIN SISTEM	20
3.1 Perancangan Game	20
3.1.1 Bahasa pemograman yang digunakan	21
3.1.2 <i>Mobile</i> Teknologi yang Digunakan.....	21
3.2 Cara Kerja J2ME (<i>Java 2 mobile edition</i>).....	21
3.2.1 Penggunaan MIDlet	21

3.3 <i>Request</i> Konektivitas <i>Bluetooth</i>	22
3.3.1 Contoh Potongan Program untuk <i>sending Bluetooth</i>	22
3.3.2 Contoh Potongan Program untuk <i>sending Bluetooth</i>	22
3.4 Analisa	23
3.4.1 Proses menu	24
3.4.2 Proses <i>Single Player</i>	24
3.4.3 Proses <i>MultiPlayer</i>	24
3.5 Perancangan Sistem	25
3.5.1 Desain Perangkat Lunak	25
3.6 Proses Penentuan Penempatan Gambar	29
3.7 Proses Penentuan <i>Score</i>	34
3.8 Pemotongan Gambar.....	35
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM	37
4.1 Implementasi Sistem.....	37
4.2 Pengujian	47
4.2.1 Lingkungan Pengujian	47
4.2.2 Perangkat Lunak.....	48
4.3 Pengujian Aplikasi.....	48
4.3.1 Sony Ericsoon Z_520i.....	48
4.3.2 Sony Ericsson K_530i	49
4.3.3 Pengujian Aplikasi <i>MultiPlayer via Bluetooth</i>	50
4.3.4 Pengujian Aplikasi Pengiriman <i>Score</i>	51
4.3.5 Pengujian Pada <i>Handphone BlackBerry</i>	52
4.4 Pengujian Pada Jarak 1 Meter Fungsi <i>Bluetooth</i>	52
4.5 Pengujian Pada Jarak 2 Meter Fungsi <i>Bluetooth</i>	53

4.6 Pengujian Pada Jarak 3 Meter Fungsi <i>Bluetooth</i>	53
4.7 Pengujian Pada Jarak 4 Meter Fungsi <i>Bluetooth</i>	53
4.8 Pengujian Pada Jarak 5 Meter Fungsi <i>Bluetooth</i>	53
4.9 Pengujian <i>MultiPlayer Bluetooth</i> dengan terhalang Tembok.....	54
BAB V KESIMPULAN DAN SARAN	55
5.1 Kesimpulan	56
5.2 Saran	57
DAFTAR PUSTAKA	58
LAMPIRAN	59

DAFTAR GAMBAR

BAB II TINJAUAN PUSTAKA

2.1 Arsitektur Java	5
2.2 Daur Hidup MIDlet.....	8
2.3 <i>Bluetooth Protocol Stack</i>	12
2.4 Topologi Jaringan <i>Bluetooth</i>	14
2.5 Metode SWAP	17
2.6 <i>Terminator</i>	18
2.7 <i>Prosesing</i>	18
2.8 <i>Decision</i>	18
2.9 <i>Prepation Symbol</i>	18
2.10 <i>Connector Symbol</i>	19
2.11 <i>Input-Output Symbol</i>	19
2.12 <i>Flowline Syimbol</i>	19

BAB III RANCANGAN DAN DESAIN SISTEM

3.1 Jaringan <i>Bluetooth</i> Pada <i>Mobile</i>	20
3.2 Proses Perancangan <i>Game Puzzle</i>	23
3.3 Menu Utama.....	25
3.4 Menu Pilihan Permainan.....	26
3.5 Tampilan <i>Score Tertionggi</i>	26
3.6 Tampilan <i>Menu Options</i>	27
3.7 Tampilan <i>Single Player</i>	27
3.8 Tampilan Utuh	28
3.9 Menu <i>Play Options</i>	28

3.11 Menu Tampil <i>Score</i>	29
3.12 Potongan Gambar Utuh Sebelum Proses <i>Swap</i>	30
3.13 <i>Coding</i> Metode SWAP pada <i>Game Puzzle</i>	31
3.14 Proses Peengacakan Potongan Gambar	32
3.15 Pengacakan Dengan Menghilangkan Potongan p8.....	33
3.16 <i>Coding</i> Percobaan Menghilangkan Potongan p8.....	34
3.17 Ukuran <i>Pixel</i> yang Akan Dipotong.....	35
3.18 Potongan Gambar <i>Puzzle</i> Menggunakan <i>Photoshop</i>	36
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM	
4.1 Tampilan <i>Splash Screen Puzzle</i>	37
4.2 Tampilan Menu Utama	38
4.3 Tampilan <i>Menu Play</i>	39
4.4 Tampilan <i>High Score</i>	40
4.5 Tampilan <i>Menu Options</i>	41
4.6 Tampilan Permainan <i>Single Player</i>	42
4.7 Tampilan <i>Menu Multiplayer</i>	43
4.8 Tampilan Pengoneksian <i>Bluetooth</i>	44
4.9 Tampilan Gambar 1	45
4.10 Tampilan Gambar 2	46
4.11 Tampilan Gambar 3	47
4.12 Sony Ericsson Z 520i	48
4.13 Sony Ericsson K 530i.....	49
4.14 Tampilan Proses Koneksi <i>Bluetooth Multiplayer</i>	50
4.15 Tampilan Proses <i>Request</i> Penerimaan Koneksi <i>Game</i>	51
4.16 Tampilan Hasil Pengiriman <i>Score</i> Pada <i>Player II</i>	51

4.17 Tampilan Hasil Pengiriman <i>Score</i> Pada <i>Player I</i>	52
4.18 Tampilan Pengujian <i>Bluetooth</i> Dengan Jarak 1 Meter	53
4.17 Tampilan Pengujian <i>Bluetooth</i> Terhalang Tembok	54

DAFTAR TABEL

BAB II TINJAUAN PUSTAKA

Tabel 2.1 Perbandingan CDLD dan CDC.....	7
Tabel 2.2 Kelas-Kelas <i>Bluetooth</i>	11

BAB I

PENDAHULUAN

1.1 Latar Belakang

Game saat ini telah banyak dimainkan oleh banyak orang dari usia muda sampai tua. Ada banyak genre dalam *game*, salah satunya adalah *game puzzle*. *Puzzle* yang merupakan permainan asah otak yang menantang ketrampilan pemainnya, sepertinya tidak pernah kehilangan popularitasnya dan tidak pernah termakan usia. *Sliding puzzle* merupakan salah satu jenis *game puzzle* yang cukup memeras otak untuk menyelesaikannya. Pemain ditantang untuk berpikir kreatif bagaimana untuk membuat semua bagian *puzzle* terletak pada posisi sebenarnya.

Cara memainkannya cukup mudah, pemain hanya menggeser *puzzle* satu demi satu sampai akhirnya semua *puzzle* terletak pada posisi sebenarnya. Permainan ini terlihat cukup sederhana namun untuk menempatkan semua *puzzle* pada tempat sebenarnya adalah kendala besar. Pemain harus mengerahkan segala kemampuan otaknya dalam berfikir untuk membuat *puzzle* tersebut terletak pada posisi sebenarnya.

Program permainan (*game*) yang menarik dalam *handphone*, membuat penulis tertarik untuk membuat sebuah aplikasi pembuatan program permainan (*game*). Khususnya dalam membuat sebuah aplikasi pembuatan *Game "Puzzle"* dengan Menggunakan Bahasa Pemrograman JAVA khususnya *Java 2 Mobile Edition (J2ME)*.

Dengan adanya fasilitas *bluetooth* pada *handphone* penulis juga menambahkan fungsi *bluetooth* tersebut sebagai suatu sistem untuk pengoneksian multiplayer pada aplikasi *game puzzle* yang akan dibuat. Dimana fungsi *bluetooth* sangatlah penting dalam *game puzzle* karena sesuai dengan permasalahan yang akan dirancang.

1.2 Rumusan Masalah

Adapun rumusan masalah dalam tugas akhir ini adalah:

- 1) Pembahasan tentang bahasa pemrograman *Java 2 Mobile Edition (J2ME)* pada *game puzzle*.
- 2) Bagaimana merancang permainan dengan menggunakan *Bluetooth* pada 2 HP atau lebih (*multipalyer*), dan penggunaan secara bersamaan.

1.3 Batasan Masalah

Batasan masalah dimaksud agar pembahasan dapat dilakukan secara terarah dan tercapai sesuai dengan yang diharapkan, maka perlu ditetapkan batasan-batasan permasalahan yang akan dibahas, yaitu :

- 1) Menggunakan 2 buah *Hand-Phone* sony ericsson Z-520i dan K-530i sebagai bahan pengujian aplikasi game puzzle.
- 2) *Handphone* yang digunakan berfasilitas *bluetooth* dan resolusi *handphone* minimal 128x160 pixel.
- 3) Pemotongan pada gambar puzzle hanya 9 pecahan.

1.4 Tujuan Masalah

Membangun aplikasi berjenis game yang akan di implementasikan pada perangkat *mobile* dan dapat dimainkan secara *multiplayer*.

1.5 Metodologi Penelitian

Metodologi yang digunakan pada penelitian Tugas Akhir ini adalah:

- 1) Studi Literatur
Penelaahan studi dengan referensi dari jurnal, buku dan artikel yang tersebut dalam daftar pustaka.
- 2) Perancangan Sistem
Dalam perancangan sistem ini penulis merancang sistem dengan menggunakan *flowchart*, yang mana *flowchart* tersebut menggambarkan suatu proses berlangsungnya aplikasi game yang akan dibuat.
- 3) Pembelajaran mengenai materi-materi terkait perancangan *J2ME scripe*.
- 4) Uji coba sistem pengujian dilakukan untuk mengetahui keberhasilan program aplikasi yang telah dibuat.

1.6 Sistematika Penulisan

Sistematika penulisan pada tugas akhir ini adalah :

BAB I : PENDAHULUAN

Dalam bab ini dibahas latar belakang tugas akhir yang dilaksanakan, maksud dan tujuan, batasan masalah, metodologi penelitian dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Dalam bab ini diuraikan teori yang dibutuhkan dalam tugas akhir, yaitu teori mengenai bahasa java, J2ME (*java 2 mobile edition*) dan *bluetooth*.

BAB III : PERANCANGAN DAN DESAIN SISTEM

Dalam bab ini penulis membahas tentang pembuatan *game Puzzle* menggunakan J2ME dan penggunaan pada handphone yang sistem yang akan dibuat menggunakan program *java 2 mobile editon*.

BAB IV : IMPLEMENTASI DAN PENGUJIAN SISTEM

Dalam bab ini penulis akan mengimplementasikan dan menguji sistem aplikasi yang dijeleaskan pada bab sebelumnya.

BAB V : KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan dan saran yang didapat selama melakukan perancang pembuatan aplikasi.

BAB II TINJAUAN PUSTAKA

2.1 Puzzle

Puzzle merupakan salah satu jenis permainan yang biasanya menggunakan potongan (*piece*), dan pemain diminta memindahkan potongan (*piece*) secara bebas. Umumnya *puzzle* ini memiliki potongan (*piece*) dengan bentuk yang kompleks sehingga selama kita berusaha mencocokkan gambar, kita juga harus mencocokkan bentuk sambungannya.

2.2 JAVA

Java menurut definisi dari sun adalah mana untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *stand alone* atau pada lingkungan jaringan. Java 2 adalah generasi kedua dari *java platform*. Kata berdiri di atas sebuah mesin interpreter yang diberi nama *Java virtual machine* (JVM). JVM inilah yang akan membaca *bytecode* dalam *file .Class* dari suatu program sebagai representasi langsung dari program yang berisi bahasa mesin. Oleh karena itu, bahasa java disebut sebagai bahasa pemrograman yang portable karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM.

Agar sebuah program java dapat dijalankan, maka file dengan ekstensi *.java* harus dikompilasi menjadi *file bytecode*. Untuk menjalankan *bytecode* tersebut dibutuhkan JRE (*java runtime environment*) yang memungkinkan pemakai untuk menjalankan program *java*, hanya menjalankan, tidak untuk membuat kode baru lagi. JRE berisi JVM dan *library java* yang digunakan.

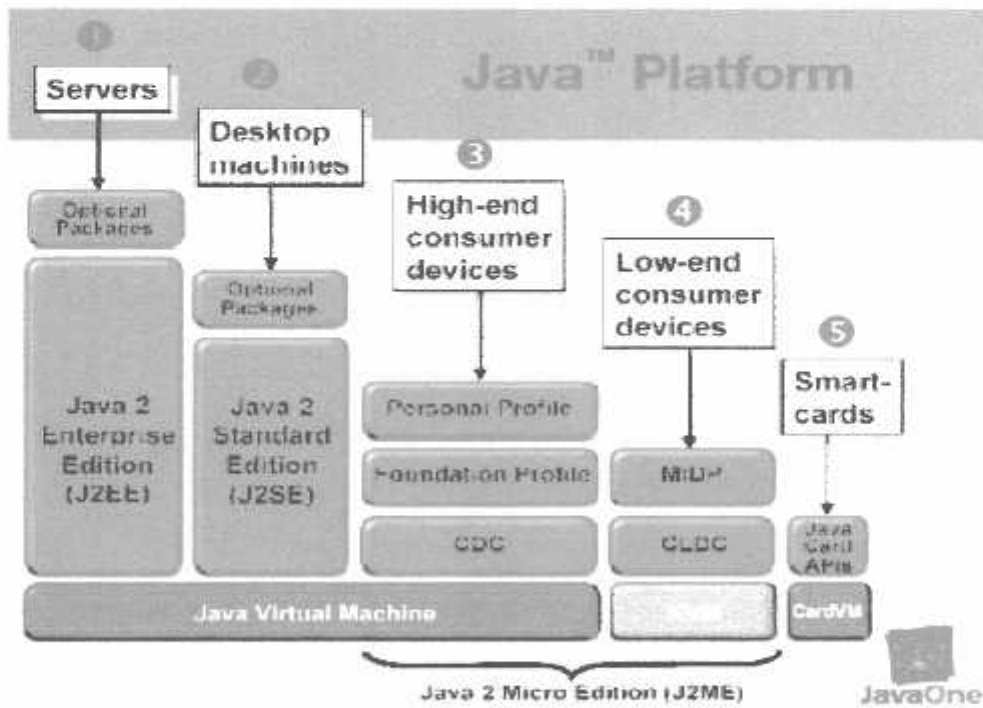
Java memiliki beberapa versi *library* atau teknologi yang disebut juga sebagai edisi dari bahasa pemrograman java. Tiga edisi utama dari *library* tersebut adalah *micro*, *standard*, dan *enterprise*.

J2ME (*java 2 micro edition*) merupakan edisi *library* yang dirancang untuk digunakan pada device tertentu seperti pager dan mobile phone. J2SE (Java2 Standard Edition) merupakan edisi *library* yang dirancang untuk membuat aplikasi *desktop* atau applet pada *web browser*. J2EE (*java 2 enterprise edition*) merupakan edisi *library* java yang dirancang untuk membuat sebuah aplikasi *enterprise* yang memerlukan antarmuka dengan sumber data (*data source*) atau dapat pula dikatakan bahwa J2EE adalah kelompok yang lebih besar dengan J2SE di dalamnya.

2.3 Java 2 Micro Edition (J2ME)

Java 2 Micro Edition merupakan subset dari J2SE yang ditujukan untuk implementasi pada peralatan *embeded* sistem dan *handled* yang tidak mampu mendukung secara penuh implementasi menggunakan J2SE. *Embeded* sistem adalah produk-produk dengan komputer kecil berada didalamnya, namun aplikasi yang bisa dimanfaatkan dari peralatan tersebut sangatlah spesifik. J2ME menyediakan lingkungan *runtime* yang optimal untuk *embeded* sistem, seperti: *pager*, *handphone*, PDA, *3G handset*, *webphone*, iTV, sistem navigasi mobil, dan sistem keamanan gedung. J2ME digunakan pada perangkat yang memiliki memory kecil.

Seperti aplikasi Java pada umumnya yang menggunakan JVM, dalam J2ME digunakan pula *Java Virtual Machine* yang disebut *K virtual machine*. *K virtual machine* adalah *virtual machine* yang sangat kecil dalam kebutuhan memorinya. Huruf K disini adalah singkatan dari kilobyte. *K virtual machine* ini berjalan pada memori 128 sampai dengan 512 *klobyte*. Pada gambar 2.1 akan ditunjukkan lingkungan kerja J2ME pada pemrograman Java.



Gambar 2.1. Arsitektur Java

J2ME terdiri atas komponen-komponen sebagai berikut :

- *JVM (Java Virtual Machine)*
Java Virtual Machine adalah software yang berfungsi untuk menjalankan program Java supaya dapat dimengerti oleh komputer. Kode program java ditulis menggunakan editor teks seperti *notepad*, *textpad*, *editplus*, *jcreator* dan lainnya. *Java compiler* yang digunakan untuk mengkompilasi kode program java dirancang untuk menghasilkan kode yang netral terhadap semua arsitektur perangkat keras (*hardware*) yang disebut sebagai *java bytecode (*.class)*. Dan JVM merupakan basis dari *java platform* dan menjembatani antara *bytecode* dengan *hardware*.
- *Java API (Application Programming Interface)*
Komponen ini merupakan kumpulan *library* untuk menjalankan dan mengembangkan program *java* pada *handheld devices*.
- *Tools* untuk mengembangkan aplikasi *java*, semacam *emulator java phone*.

2.3.1 J2ME Configuration (konfigurasi)

Konfigurasi J2ME adalah spesifikasi yang mendefinisikan sebuah *virtual machine* dari kumpulan API-API dasar yang dapat digunakan dalam kelas tertentu dari sebuah peralatan. Perbandingan ditunjukkan pada table 2.1.

Ada 2 konfigurasi pada J2ME :

1. *CLDC (Connected Limited Device Configuration)*
CLDC merupakan perangkat atau konfigurasi dasar dari J2ME. CLDC sebenarnya berupa *library* dan API (*Application Programming Interface*) yang diimplementasikan pada J2ME. Konfigurasi ini biasanya untuk alat kecil seperti telepon seluler (*handphone*), *pager* dan PDA. Peralatan tersebut biasanya mempunyai keterbatasan memori (RAM), sumber daya, dan kemampuan memproses.
 2. *CDC (Connected Device Configuration)*
CDC merupakan perangkat atau konfigurasi superset dari CLDC. Konfigurasi ini biasanya dipakai untuk alat seperti *internet TV*, *nokia communicator* dan *car TV*.
-

Tabel 2.1
Perbandingan antara CDLD dan CDC

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 <i>handheld</i> dengan ukuran memori terbatas (160-512 Kilobyte)	Digunakan pada perangkat <i>handheld</i> dengan ukuran memori minimal 2 Megabyte
Prosesor : 16 atau 32 bit	Prosesor : 32 bit

2.3.2 Profil J2ME

Sebuah profil dibangun dalam sebuah konfigurasi, namun ditambahkan beberapa API khusus agar dihasilkan sebuah lingkungan yang lengkap untuk membangun aplikasi. Profil berisi daur hidup (*life cycle*), antarmuka pemakai (*user interface*), serta penyimpanan. Salah satu profil J2ME adalah *Mobile Information Device Profile* (MIDP). Profil MIDP menyediakan sebuah *platform* standar untuk peralatan komunikasi bergerak yang memiliki kapasitas memori terbatas sehingga cocok untuk pengembangan aplikasi pada ponsel.

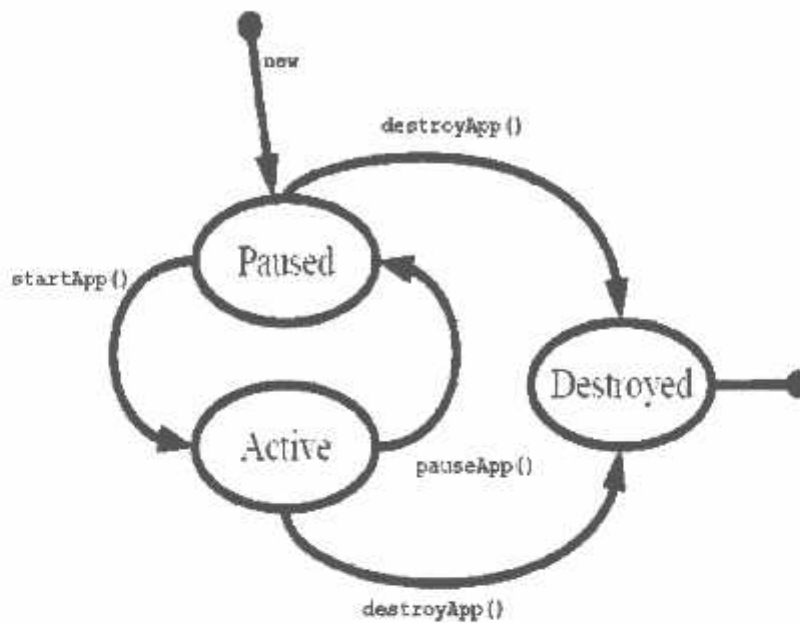
2.3.3 MIDP dan MIDlet

MIDP merupakan profil yang banyak digunakan dan populer dari J2ME dan MIDlet merupakan aplikasi-aplikasi yang dibuat di dalam handphone menggunakan profil MIDP.

2.3.3.1 Daur Hidup MIDlet

Daur hidup dari sebuah MIDlet ditangani oleh *Application Management Software* (AMS). AMS adalah sebuah lingkungan tempat siklus dari sebuah MIDlet diciptakan, dijalankan, dihentikan, maupun dihilangkan. AMS sering disebut dengan *Java Application Manager* (JAM). Dalam daur hidupnya MIDlet memiliki tiga status, yaitu

Pause, *Active* dan, *Destroy*. Ketika masing-masing status dipanggil, beberapa fungsi standar yang bersesuaian akan dipanggil.



Gambar 2.2. Daur Hidup MIDlet

Dari gambar 2.2 dapat dijelaskan sebagai berikut :

- Ketika MIDlet pertama kali diciptakan dan diinisialisasi, MIDlet berada dalam status *pause*.
- Jika terjadi kesalahan selama konstruksi MIDlet, MIDlet akan berpindah ke status *destroyed* dan MIDlet batal diciptakan dengan memanggil fungsi *destroyApp()*
- Saat MIDlet dijalankan, MIDlet akan berada pada status *active* dan memanggil fungsi *startApp()*.
- Jika saat dijalankan MIDlet dihentikan sementara, maka MIDlet berada pada status *pause* dan memanggil fungsi *pauseApp()*.

2.3.3.2 Antarmuka Pemakai

Untuk membuat suatu antarmuka bagi pemakai, program MIDlet harus mengimpor paket *javac.microedition.lcdui*. Kelas yang dipergunakan untuk membuat dan memanipulasi antarmuka tersebut adalah kelas yang diturunkan dari kelas *Displayable*.

Melalui kelas-kelas inilah sebuah aplikasi dapat berinteraksi dengan pemakai. Pada MIDP, antarmuka terdiri dari API tingkat tinggi (*High-level*) dan API tingkat rendah (*Low-level*). API tingkat tinggi berbasis pada kelas *Screen*, sedangkan API tingkat rendah berbasis pada kelas *Canvas*.

2.3.4 Pemaketan aplikasi MIDlet

Hasil dari kompilasi program sumber Java adalah satu atau lebih berkas *bytecode* yang dikenali dengan akhiran *“.class”*. Pada aplikasi MIDlet, semua berkas *byte-code* dipaketkan menjadi suatu berkas terkompresi yang disebut *Java Archive* (JAR) yang dikenali dengan ekstensi *“.jar”*. Selain berkas JAR, terdapat juga berkas *Java Application Descriptor* (JAD) yang berekstensi *“.jad”*, yaitu berkas yang berisi informasi mengenai suatu berkas JAR. Kedua berkas inilah yang harus di-*upload* ke perangkat ponsel agar aplikasi dapat dijalankan. Untuk meng-*upload* berkas tersebut ke perangkat ponsel dapat dilakukan dengan cara transfer data antara komputer dengan perangkat komunikasi bergerak melalui media sinar infra merah, koneksi kabel data atau *bluetooth*.

2.3.5 Perangkat Perkembangan J2ME

Kakas / *tool* untuk mengembangkan aplikasi J2ME disediakan oleh *sun microsystem*. *Tool* ini diberi nama *Wireless Toolkit* (WTK). Versi terbaru sampai buku ini ditulis adalah WTK 2.5 untuk windows sedangkan untuk unix masih 2.0. Jumlah *Java Specification Requirement* (JSR) yang diimplementasikan pada masing-masing versi WTK berbeda-beda. Versi terbaru dari WTK biasanya mengimplementasikan semua JSR yang diimplementasikan pada versi sebelumnya ditambah dengan JSR terbaru lainnya.

Implementasi aplikasi pada WTK ini mengacu pada standar J2ME, CLDC dan MIDP dan tidak mengacu pada perangkat mobile vendor manapun. Hal ini perlu diwaspadai karena biasanya vendor perangkat *mobile* melakukan penyesuaian implementasi standar pada perangkatnya. Aplikasi yang berjalan baik di WTK Emulator belum tentu dapat berjalan dengan baik di perangkat aslinya. WTK mempunyai library-library yang dapat digunakan untuk pembuatan aplikasi J2ME. Seringkali penggunaan WTK ini dijadikan plug-in untuk *development* environment seperti *Net Beans* dan *eclipse*. Dengan demikian dimungkinkan pengembangan aplikasi J2ME pada

lingkungan yang lebih baik seperti halnya pengembangan aplikasi J2SE maupun J2EE. Library-library yang dicakup dalam WTK antara lain:

- Java Technology for Wireless Industry/JTWI (JSR 185)
- Wireless Messaging API/WMAPI (JSR 120)
- Java API Bluetooth Wireless ToolKit/JABWT (JSR 82)
- Java Web Service API (JSR 172)
- Mobile Media API (JSR 135)

Selain WTK dari Sun Microsystems terdapat juga perangkat pengembangan yang lain untuk J2ME. Kebanyakan perangkat pengembangan ini dikeluarkan oleh vendor perangkat mobile. Dengan menggunakan perangkat pengembangan dari suatu vendor perangkat mobile diharapkan nantinya aplikasi kita akan berjalan dengan baik pada perangkat mobile vendor tersebut tanpa penyesuaian lagi. Perangkat-perangkat pengembangan itu antara lain:

- Nokia developer Suite for J2ME
- Motorola J2ME SDK
- Sony Ericsson J2ME SDK
- BenQ-Siemens Mobility Toolkit

2.4 *Bluetooth*

Bluetooth adalah spesifikasi industri untuk jaringan kawasan pribadi (*personal area networks* atau PAN) tanpa kabel. *Bluetooth* menghubungkan dan dapat dipakai untuk melakukan tukar-menukar informasi di antara peralatan-peralatan. Spesifikasi dari peralatan *bluetooth* ini dikembangkan dan didistribusikan oleh kelompok *Bluetooth Special Interest Group*. *Bluetooth* beroperasi dalam pita frekuensi 2,4 Ghz dengan menggunakan sebuah *frequency hopping traceiver* yang mampu menyediakan layanan komunikasi data dan suara secara real time antara host-host *bluetooth* dengan jarak terbatas. Kelemahan teknologi ini adalah jangkauannya yang pendek dan kemampuan transfer data yang rendah.

2.4.1 *Teknologi Bluetooth*

Teknologi *Bluetooth* dikembangkan oleh *Bluetooth Special Interest Group (SIG)* yang diprakarsai oleh perusahaan-perusahaan besar yaitu companies 3Com, Ericsson,

Intel, IBM, Agere, Microsoft, Motorola, Nokia dan Toshiba. *Bluetooth* telah didesain bagi alat komunikasi personal yang mendukung komunikasi tanpa kabel (*wireless*) seperti pada *mobile phone* maupun *PDA* dengan jarak sampai 10 kilometer. *Bluetooth* adalah sebuah protokol komunikasi yang beroperasi pada frekuensi 2.4 GHz. Sinyal dari *bluetooth* adalah *omni-directional* serta dapat menerima data dan juga suara.

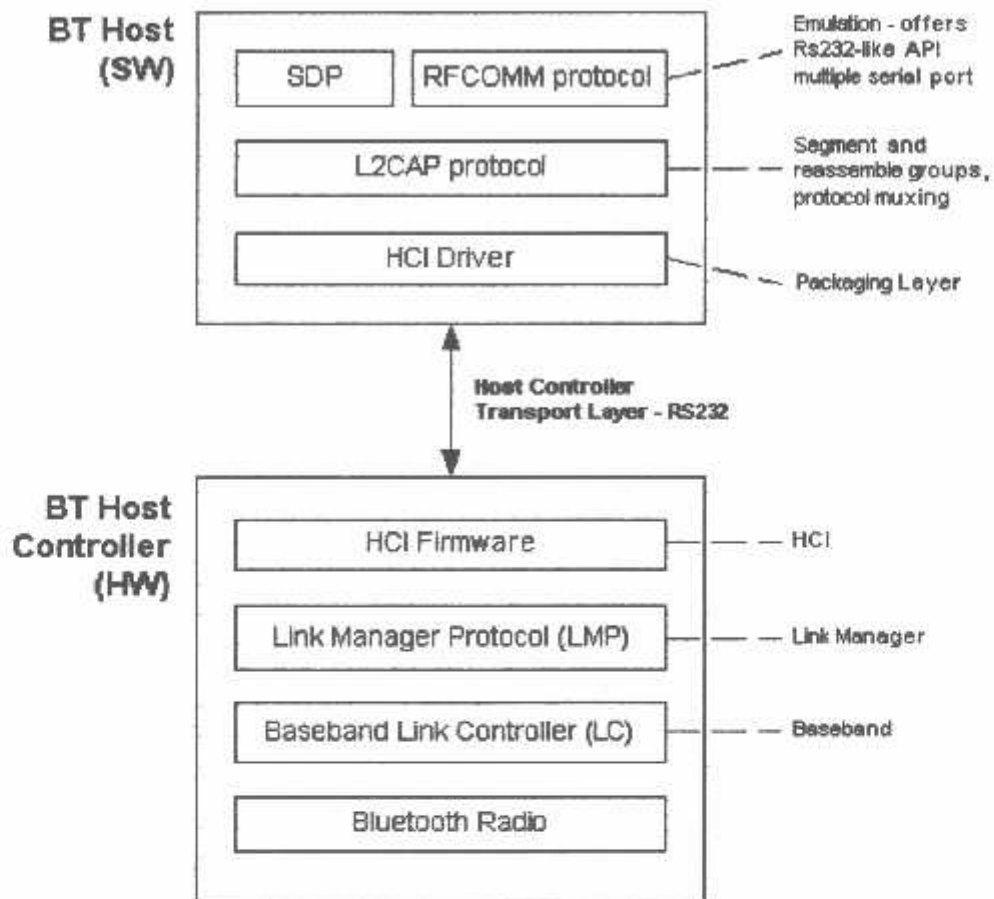
Device yang mendukung komunikasi *bluetooth* dapat dengan mudah menemukan koneksi dan juga berkomunikasi dengan *device* yang lain secara otomatis. *Bluetooth* dapat digunakan untuk mengirim *file*, membangun sebuah jaringan tertentu, sinkronisasi data, mengkoneksikan sekitar misalnya dengan *hands-free kits* dan juga digunakan untuk *gaming*. Sesuai dengan *power* serta jangkauannya, *bluetooth* terbagi menjadi 3 kelas:

Tabel 2.2 Kelas-kelas Bluetooth

Kelas	Maksimum Power yang diijinkan	Jangkauan Maksimal
Kelas 1	100 mW (20 dBm)	100 meter
Kelas 2	2.5 mW (-4dBm)	10 meter
Kelas 3	1 mW (0 dBm)	1 meter

2.4.2 *Bluetooth Protocol Stack*

Bluetooth Protocol Stack secara umum dapat dibagi menjadi dua komponen yaitu: *Bluetooth Host* dan *Bluetooth Controller* yang disebut juga *Bluetooth Radio*. *Bluetooth Protocol Stack* terdiri dari protokol-protokol yang sejak dikembangkan sudah dimilikinya yaitu protokol L2CAP dan SDP, serta protokol lain hasil dari adopsi yaitu OBEX. Selain itu *Bluetooth Protocol Stack* dapat dibagi menjadi empat layer sesuai dengan peruntukannya, sesuai dengan yang ditunjukkan didalam gambar.



Gambar 2.3 Bluetooth Protocol Stack

- *Baseband* yang berfungsi untuk mengaktifkan *physical RF* sehingga *Bluetooth* dapat membuat koneksi.
- *Link Manager Protocol (LMP)* adalah protokol yang bertanggung jawab untuk membangun koneksi antara *Bluetooth device* serta mengatur sistem keamanan didalam koneksi, seperti *authentication* dan *encryption*.
- *Logical Link Control and Adaptation Protocol (L2CAP)* adalah protokol yang mengadaptasikan layer protokol yang ada di atas kepada *baseband*. Hal ini melibatkan banyak bagian, yaitu bermacam-macam koneksi *logical* yang berada pada layer di atasnya.
- *Service Discovery Protocol (SDP)* digunakan untuk mendapatkan informasi tentang atribut *device*, *service* yang disediakan, dan berikut karakteristik dari *service*.

- *RFCOMM* mengimplementasikan *RS-232* kontrol dan data *signal* melalui *Bluetooth baseband*, yang menyediakan kemampuan transportasi untuk layer yang berada di atasnya dengan menggunakan antar muka serial sebagai mekanisme transportasi.
- *TCS Binary* mendefinisikan pemanggilan sinyal control untuk membuat panggilan data dan suara antara *Bluetooth device*.

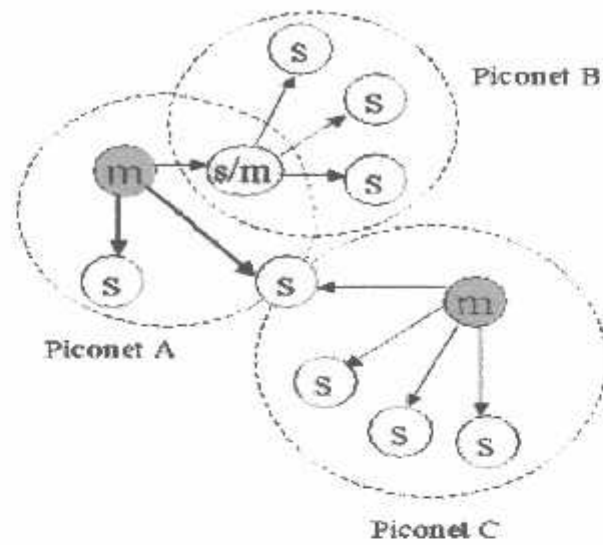
2.4.2.1 Jaringan *Bluetooth*

Bluetooth devices dibagi lagi menjadi group-group kecil yang disebut *piconet*. Didalam *piconet*, ada sebuah *master* dan satu atau lebih *slaves*. Sampai 7 buah *slaves* bisa diterima didalam sebuah *piconet*. Sedangkan *master unit* adalah sesuatu yang memulai proses komunikasi. Ia akan menggunakan komunikasi *point-to-multipoint*.

Sebuah *device* didalam sebuah *piconet* mampu berkomunikasi dengan *bluetooth device* didalam *piconet* yang lain. Sebuah *slave* didalam *piconet* tertentu mungkin dapat menjadi *slave* didalam *piconet* yang berbeda. Sebuah *master* didalam sebuah *piconet* dapat menjadi *slave* pada *piconet* yang lain. Komunikasi antar *piconet* tersebut dapat berasal dari jaringan manapun.

Dalam rangka untuk pemeliharaan baterai; *Bluetooth* memiliki tiga *low modus operasi* yang hemat energi, Pada *modus sniff*, sebuah *slave device* akan memperhatikan berkurangnya level energi, dimana ia tidak berpengaruh terhadap *piconet*.

Pada *modus hold*, sebuah *device* tidak hanya mengirimkan data tetapi ia juga melakukan sinkronisasi secara konstan dengan *master*. Ia bukanlah member aktif dari *piconet*, tetapi ia menyimpan alamat member yang aktif. Sebuah *device* pada modus *park* berlaku seperti *device* pada modus *hold*, akan tetapi ia tidak menyimpan alamat member yang aktif.



Gambar 2.4 Topologi Jaringan Bluetooth

2.4.2.2 Profile Bluetooth

Profile dari bluetooth telah ditentukan dengan memperhatikan *interoperability* antara *device* dan aplikasi dari beberapa manufaktur. Sebuah *profile* mendefinisikan *roles* dan kemampuan yang dimiliki oleh tipe aplikasi tertentu. Sebuah *device* hanya dapat berkomunikasi dengan *device* yang lain apabila mereka memiliki *profile* yang sama. Semua *bluetooth device* harus menggunakan *Generic Access Profile*. Profile ini menentukan prosedur koneksi, *device discovery*, dan *management link*.

Sebuah *Service Discovery Profile* mendefinisikan fitur dan prosedur bagi aplikasi *bluetooth* untuk mengenali segala servis yang telah ditentukan pada *bluetooth device* yang lain.

Sebuah *profile* sinkronisasi menentukan hal-hal apa saja yang dibutuhkan pada dua atau lebih *devices* untuk mensinkronisasikan data.

Profile bluetooth yang lain telah didefinisikan juga pada spesifikasi *bluetooth*. Akan tetapi tidak akan dibicarakan dalam materi ini.

2.4.3 Spread Spectrum dan Frequency Hopping

Spread spectrum dengan *frequency hopping* adalah proses *spread* atau penyebaran spektrum yang dilakukan pemancar dengan frekuensi pembawa informasi yang merupakan deretan pulsa termodulasi acak semu (*pseudorandom*) yang dilompatlompatkan dari satu nilai frekuensi ke nilai frekuensi yang lain dalam lebar

spectrum frekuensi yang telah ditetapkan sebelumnya dan berulang kali dengan pola kode yang dapat dimodifikasi secara saling bebas, sehingga dapat menempatkan sejumlah pemakai dalam lebar spektrum frekuensi tersebut dengan berbeda pola acak kode generatornya. Teknik penyebaran spektrum (*spread spectrum*) digunakan, karena :

- Kemampuannya membatasi interferensi internal akibat padatnya lalu lintas komunikasi yang menggunakan frekuensi radio.
- Kemampuan menolak terhadap penyadapan informasi oleh penerima yang tidak dikenal. Dapat dioperasikan dengan kerapatan spektral berenergi rendah.
- Penggunaan yang lebih aman. Frekuensi ini dapat melakukan lompatan gelombang hingga 1600 lompatan per detik. Hal ini mempersulit dilakukan penyadapan data, karena lompatan sinyal data yang cepat dan tidak beraturan sulit ditangkap oleh transceiver lain, kecuali *transceiver* penerimanya.
- Penggunaan yang lebih aman. Frekuensi ini dapat melakukan lompatan gelombang hingga 1600 lompatan per detik. Hal ini mempersulit dilakukan penyadapan data, karena lompatan sinyal data yang cepat dan tidak beraturan sulit ditangkap oleh *transceiver* lain, kecuali *transceiver* penerimanya.
- *Noise* yang lebih kecil dan jarak pita gelombang yang sempit dapat menolak interferensi.

2.4.4 Aplikasi dan Layanan

Protokol bluetooth menggunakan sebuah kombinasi antara circuit switching dan packet switching. Bluetooth dapat mendukung sebuah kanal data asinkron, tiga kanal suara sinkron simultan atau sebuah kanal dimana secara bersamaan mendukung layanan data asinkron dan suara sinkron. Setiap kanal suara mendukung sebuah kanal suara sinkron 64 kb/s.

Kanal asinkron dapat mendukung kecepatan maksimal 723,2 kb/s asimetris, dimana untuk arah sebaliknya dapat mendukung sampai dengan kecepatan 57,6 kb/s. Sedangkan untuk mode simetris dapat mendukung sampai dengan kecepatan 433,9 kb/s.

Range yang dapat dijangkau oleh Bluetooth adalah 10 meter atau 30 feet. Sistem Bluetooth juga menyediakan layanan komunikasi *point to point* maupun komunikasi *point to multipoint*. Produk bluetooth dapat berupa PC card atau USB adapter yang dimasukkan ke dalam perangkat. Sedangkan perangkat yang dapat dikombinasikan dengan Bluetooth diantaranya: handphone, kamera, personal computer (PC), printer, headset, Personal Digital Assistant (PDA), dan lainnya. Aplikasi-aplikasi yang dapat disediakan oleh layanan bluetooth ini antara lain : PC to PC file transfer, PC to PC file synch (notebook to desktop), PC to mobile phone, PC to PDA, wireless headset, LAN connection via *ethernet access point* dan sebagainya.

2.5 Mobile (Hand Phone)

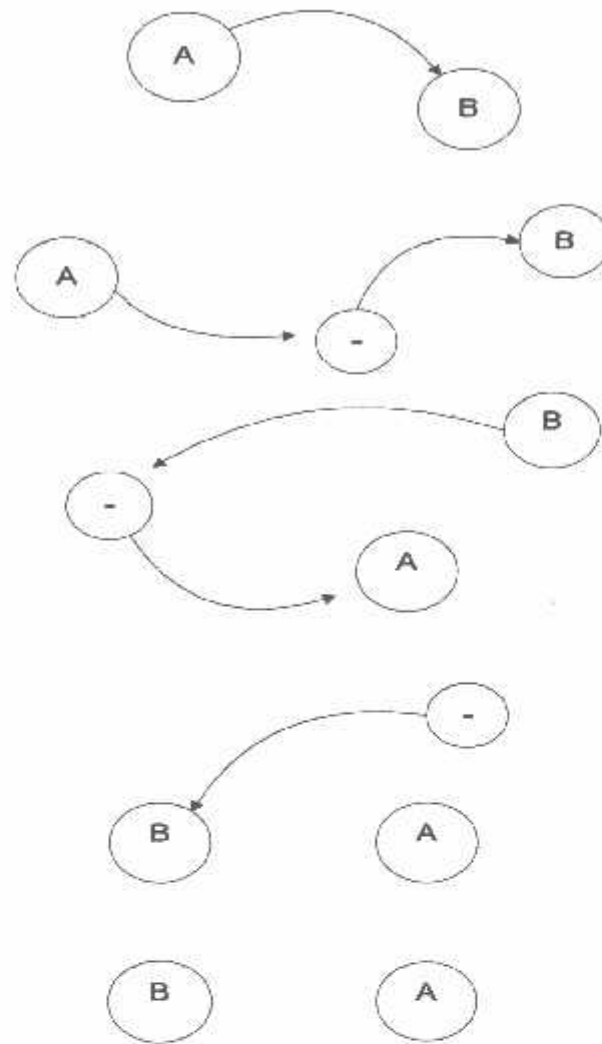
Mobile adalah perangkat elektronik yang mempunyai kemampuan dasar yang sama dengan saluran konvensional saluran tetap, namun dapat dibawa kemana-mana (portable, mobile) dan tidak perlu disalurkan dengan jaringan telpon menggunakan kabel (nirkabel; wireless):(wikipedia.org).

2.6 Metode SWAP

Pada aplikasi game *puzzle* ini metode yang digunakan untuk pengacakan gambar adalah metode SWAP, dimana metode SWAP berfungsi untuk mengacak *gride* (kolom) antara kolom A dan kolom B dari gambar yang telah dipotong.

Untuk pengacakan gambar dilakukan secara manual dengan menentukan algoritma pada potongan yang telah dipilih dan pada saat player memulai game gambar sudah teracak secara otomatis.

Dimana kolom A melakukan perpindahan pada posisi kolom B, maka adanya kolom kosong yang menyediakan tempat sementara untuk penyimpanan kolom yang akan diisi tersebut dan proses ini terjadi terus menerus, sehingga kembali pada posisi pertama. Proses metode ini bisa dilihat pada gambar 2.5.



Gambar 2.5 Metode SWAP

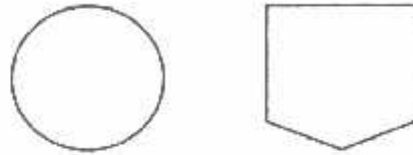
2.7 Flowchart

Flowchart adalah gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Gambaran ini dinyatakan dengan simbol, dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan hubungan antar proses digambarkan dengan garis penghubung.

Flowchart ini merupakan langkah awal pembuatan program. Dengan adanya *flowchart* urutan proses kegiatan menjadi lebih jelas dan jika ada penambahan proses maka dapat dilakukan lebih mudah.

2.7.5 Connector symbol

Digunakan untuk menunjukkan sambungan dari bagan alir yang terputus dihalaman yang masih sama atau dihalaman lainnya.



Gambar 2.10 Connector Symbol

2.7.6 Input/Output symbol

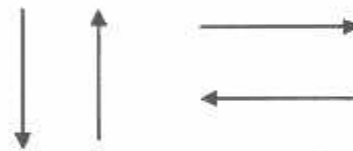
Digunakan untuk mewakili data *input/output*.



Gambar 2.11 Input/output symbol

2.7.7 Flow line symbol

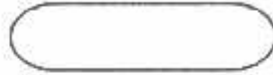
Digunakan untuk menunjukkan arus dari suatu proses.



Gambar 2.12 Flow line symbol

2.7.1 Terminator

Terminator adalah tampilan pada awal flowchart (berisi kata "Start") atau pada akhir proses (berisi kata "End")



Gambar 2.6 Terminator

2.7.2 Prosesing

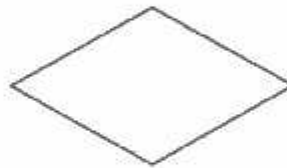
Prosesing ialah satu atau beberapa himpunan penugasan yang akan dilaksanakan secara berurutan.



Gambar 2.7 Prosesing

2.7.3 Decision

Decision merupakan suatu simbol keputusan dalam menyeleksi suatu kondisi didalam program.



Gambar 2.8 Decision

2.7.4 Preparation Symbol

Digunakan untuk member nilai awal suatu besaran.



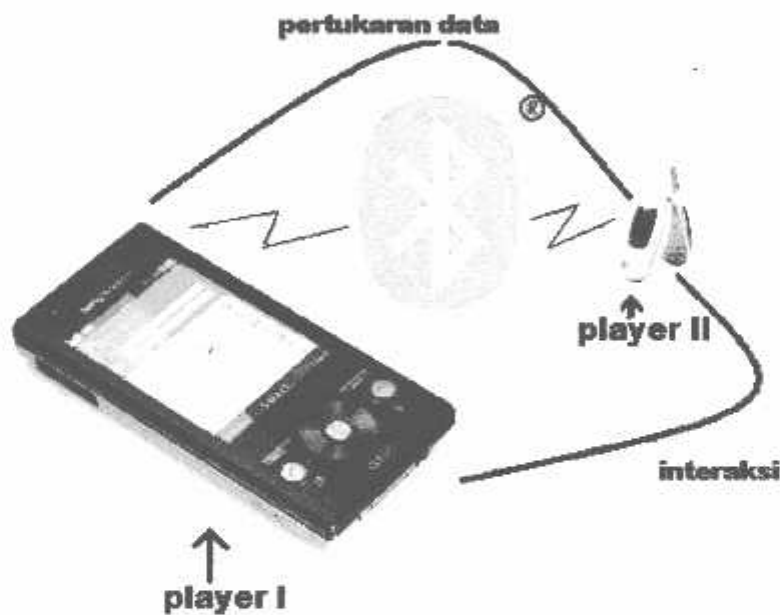
Gambar 2.9 Preparation Symbol

BAB III

PERANCANGAN DAN DESAIN SISTEM

3.1. Perancangan Game

Pada bab ini akan dilakukan Perancangan sistem untuk aplikasi multiplayer game puzzle menggunakan ponsel, yang juga memanfaatkan teknologi bluetooth untuk pertukaran data. Dalam pembuatan sistem ini dibutuhkan *software* J2ME dan telepon genggam yang support Java. Untuk alur sistem atau gambaran sistem secara umum dapat dilihat pada gambar 3.1.



Gambar 3.1 Contoh Jaringan *Bluetooth* Pada *Mobile*.

Dalam pembuatan game adanya sistem yang digunakan pada aplikasi game, yaitu menghubungkan 2 player dengan menggunakan *bluetooth*, agar dapat berinteraksi antara player I dan player II.

3.1.1. Bahasa Pemrograman Yang Digunakan

Dalam penelitian ini penulis akan mengimplementasikan hasil rancangan dalam bahasa pemrograman java, karena didasarkan atas beberapa hal :

1. Perkembangan bahasa pemrograman java sangatlah pesat karena dapat dikembangkan oleh komunitas-komunitas tertentu.
2. Bahasa pemrograman java adalah bahasa pemrograman yang *multi platform* karena dapat berjalan di beberapa komputer berbeda.

3.1.2. Mobile Technology Yang Digunakan

Dalam pemilihan *mobile technology*, penulis mengembangkan dengan teknologi J2ME. Hal ini didasari oleh :

1. J2ME adalah pengembangan bahasa pemrograman java yang ditujukan untuk aplikasi pada ponsel.
2. Sebagian besar ponsel – ponsel yang beredar dipasaran mampu dan bisa menjalankan aplikasi J2ME.
3. Aplikasi J2ME merupakan aplikasi *multi platform*.

3.2. Cara Kerja J2ME (*java 2 mobile edition*)

3.2.1 Penggunaan MIDlet

J2ME adalah satu set spesifikasi dan teknologi yang fokus kepada perangkat konsumen. Perangkat ini memiliki jumlah memori yang terbatas, menghabiskan sedikit daya dari baterai, layar yang kecil dan *bandwith* jaringan yang rendah.

Konsep program J2ME adalah sebagai berikut; dalam J2ME kita akan membuat *main class* turunan dari *class* MIDlet, *main class* turunan dari MIDlet tersebut yang nantinya akan dipanggil pertama kali saat aplikasi game puzzle ini berjalan di *handphone*, dalam *main class* ada 3 *method* yang nantinya berfungsi sebagai *trigger even* dari *handphone* yang akan digunakan Seperti:

startApp(), *method* yang dipanggil apabila aplikasi kita pertama kali jalan.
pauseApp(), *method* yang dipanggil apabila pengguna *handphone* mempause aplikasi kita.
destroyApp(), *method* yang dipanggil apabila pengguna *handphone* menutup aplikasi kita.

3.3. Request Konektivitas *Bluetooth*

Request konektivitas *bluetooth* merupakan suatu proses pengoneksian antara dua *handphone* yang berbeda dalam fungsi pertukaran atau pengiriman data. Dalam hal proses pengiriman dan penerimaan data, perlunya listing khusus pada proses konektivitas tersebut.

3.3.1 Contoh potongan program untuk *sending Bluetooth*

```
BluePacket s = endpt.getString();

if ( s != null ) {

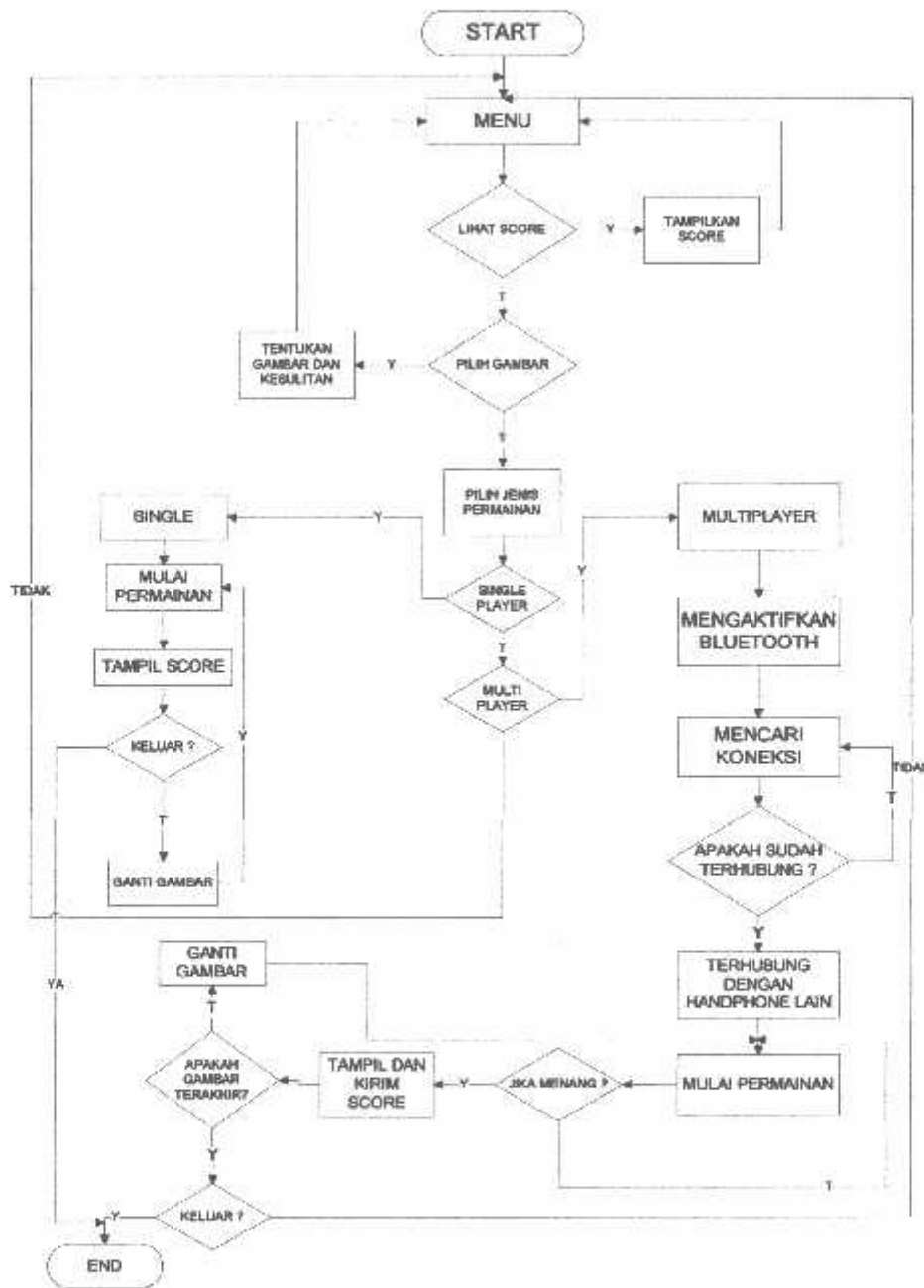
log("sending signal "+s.signal+" string "+s.msg+" to
"+endpt.remoteName);
dataout.writeInt(s.signal);
dataout.writeUTF(s.msg );
dataout.flush();
```

3.3.2 Contoh potongan program untuk *reading Bluetooth*

```
BluePacket packet = new BluePacket
BlueLayer.SIGNAL_MESSAGE, endpt.remoteName, s );
log("read in MESSAGE string "+s+" from "+endpt.remoteName);
if (BlueControl.instance.isReady==true){
if (s.indexOf("READY")!=-1 && BlueControl.instance.BCon.cPlaye
==null){
BlueControl.instance.btnet.sendString(BlueControl.instance.Names+"
READY");
if ("Player-I".equals(BlueControl.instance.Names)){
int level= BlueZzle.random(3);
```

3.4 Analisa

Analisa proses dilakukan dengan menggunakan alir diagram (*flowchart*) tujuannya agar memudahkan pada proses perancangan dan pada saat pengembangan aplikasi seperti pada Gambar 3.2.



Gambar 3.2 Proses Perancangan *Game Puzzle*

Adapun penjelasan proses *flowchart* diatas sebagai berikut :

3.4.1 Proses *menu*

Dalam proses ini menu tersebut menampilkan beberapa pilihan yang sesuai dengan pilihan yang sudah ditentukan pembuat, yaitu :

- a) Pilih jenis permainan (*play*), dimana pengguna dapat memilih jenis permainan yang diinginkan.
- b) Nilai tertinggi (*high score*), dimana pengguna dapat melihat hasil tertinggi dari permainan tersebut.
- c) Pilih gambar, dimana pemain dapat memilih gambar dan tingkat kesulitan yang sudah dibuat pada sistem tersebut.

3.4.2 Proses *Single player*

Dalam proses *single player* ini dapat difungsikan untuk bermain sendiri dan didalam menu *single player* mempunyai 2 sistem, yaitu ;

- a) Mulai permainan (*play*), dimana pada saat pengguna menekan tombol pilih maka langsung berhadapan dengan *game*, dan *game* dapat langsung dimainkan.
- b) Tampil *score* dimana pengguna dapat langsung melihat *score* jika pemain berhasil menyelesaikan *game puzzle*.

3.4.3 Proses *Multiplayer*

Sedangkan dalam proses *multiplayer* kurang lebih sama dengan proses *single player*, tetapi proses ini ditambahkan dengan 2 pengguna atau 2 *player* yang bermain secara bersamaan. Adapun sistem yang digunakan dalam *multiplayer* ini,yaitu ;

- a) Mengaktifkan *bluetooth*, dimana sebelum membuka aplikasi *game puzzle* tersebut, 2 pengguna yang berbeda harus mengaktifkan koneksi *bluetooth*, agar pada saat pencarian koneksi dapat terdeteksi.
 - b) Mencari koneksi, yang mana sistem ini melakukan proses secara otomatis dan saling interaksi antara *bluetooth player 1* dan *Bluetooth player 2*. Dan jika *player 1* sudah terhubung dengan *player 2* maka tampilan *game puzzle* akan muncul secara bersamaan dan permainan dapat segera diumulai.
-

- c) Tampil dan kirim *score*, jika salah satu *player* 1 menang maka *score* akan tampil dan terkirim secara otomatis ke *player* 2, dan pada saat itu juga *player* 1 dapat melanjutkan gambar berikutnya, sampai dengan gambar terakhir.

3.5 Perancangan Sistem

Perancangan sistem merupakan pembuatan sistem yang berasal dari data yang telah ada dan penelitian yang telah dilakukan oleh pembuat. Perancangan sistem ini adalah inti dari program tersebut, karena mulai awal dari pembuatan sampai dengan selesai program akan dibuat.

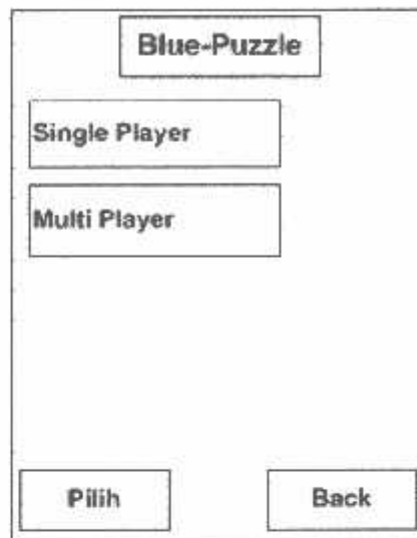
3.5.1 Desain Perangkat Lunak

Dalam bab ini akan dijelaskan tentang desain perangkat lunak dan pembahasan tentang program. Dimana pembuat akan menampilkan menu utama dalam pembuatan game *puzzle* dan dalam tampilan menu utama ini sudah tersusun sesuai dengan sistem dan proses yang sudah ditentukan oleh pembuat. Tampilan menu pada *game* bisa dilihat pada gambar 3.3.



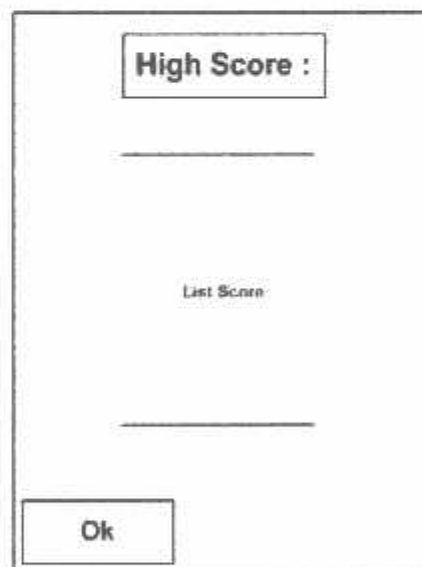
Gambar 3.3 Menu Utama

Dalam *game puzzle* kita menggunakan 2 pilihan dimana pilihan tersebut yaitu *player* I dan *player* II, adapun tampilan pada saat kita memilih permainan yang akan kita mainkan, tergantung pada pengguna atau pemainnya. Dapat kita lihat pada gambar 3.4.



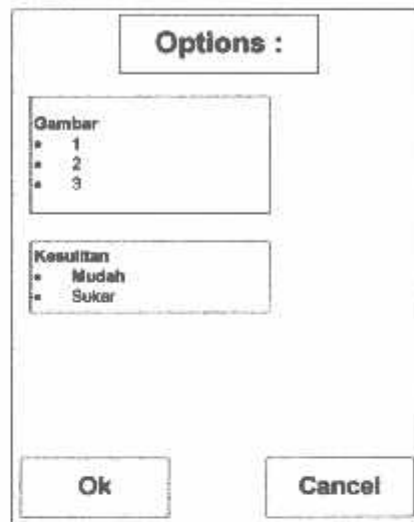
Gambar 3.4 Menu pilihan permainan

Disini pembuat akan menampilkan menu-menu lain pada game sesuai dengan proses dan sistem yang sudah tersusun. Pada gambar 3.5 berikut dapat kita lihat bahwa ini adalah proses menampilkan *score*.



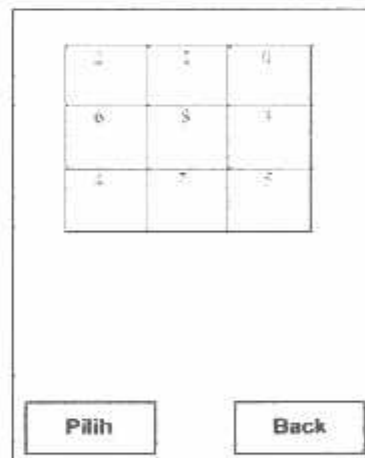
Gambar 3.5 Tampilan Score Tertinggi

Dalam desain ini pembuat menampilkan menu *options* yang mana fungsi dari menu tersebut untuk menentukan pengacakan atau pertukaran potongan gambar serta tingkat kesulitan dalam bermain game *puzzle*, dapat kita lihat pada gambar 3.6.



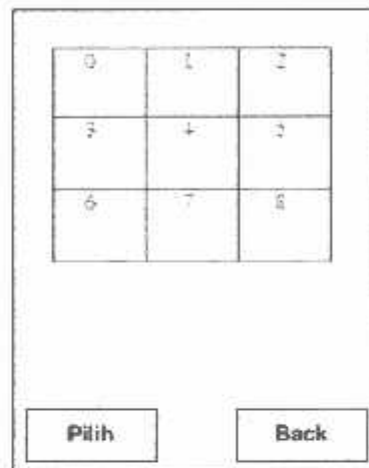
Gambar 3.6 Tampilan Menu *Options*

Untuk tampilan permainan *single player* bisa dilihat pada gambar 3.7, ketika pemain memilih *singleplayer* pada menu maka akan menampilkan gambar-gambar yang sudah tersusun secara acak.



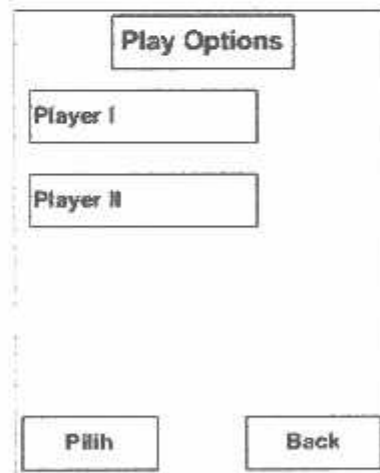
Gambar 3.7 Tampilan Single Player

Pada gambar diatas player dapat langsung bermain dan jika player ingin mengingat atau melihat sejenak gambar yang utuh maka player cukup menekan angka 0 pada *keypad handphone*, maka akan berbentuk gambar sesuai dengan urutan awalnya, seperti gambar 3.8.



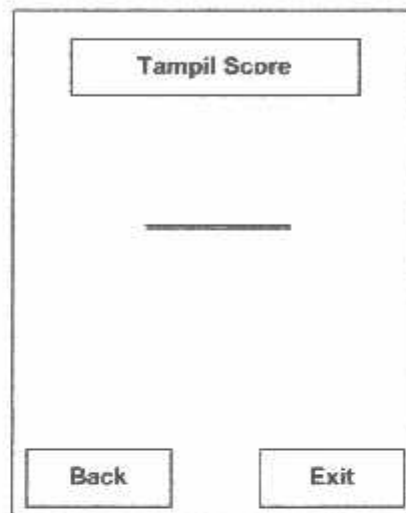
Gambar 3.8 Tampilan Utuh

Untuk perlakuan *multiplayer* sendiri mempunyai hak khusus pada proses ini, yaitu pada saat player 1 dan player 2 sudah dalam kondisi *bluetooth* aktif, maka secara langsung player 1 dapat mengirimkan isyarat atau sinyal khusus pada player 2 dan setelah player 2 mengizinkan atau mengkonfrimasi *sinyal* yang diberikan player 1, maka player 2 dapat bermain secara bersamaan. Pada gambar 3.9 dapat kita lihat menu *play options*.



Gambar 3.9 Menu *Play Options*

Setelah permainan selesai dan jika player 1 menang, maka player 1 akan mengirimkan *score* atau nilai kepada player 2 dan player 1 dapat melanjutkan permainan dengan gambar berikutnya. Dalam hal ini kedua pemain dapat melakukan permainan terus-menerus sampai pada gambar terakhir yang sudah ditentukan oleh sistem. Dapat dilihat pada gambar 3.10.



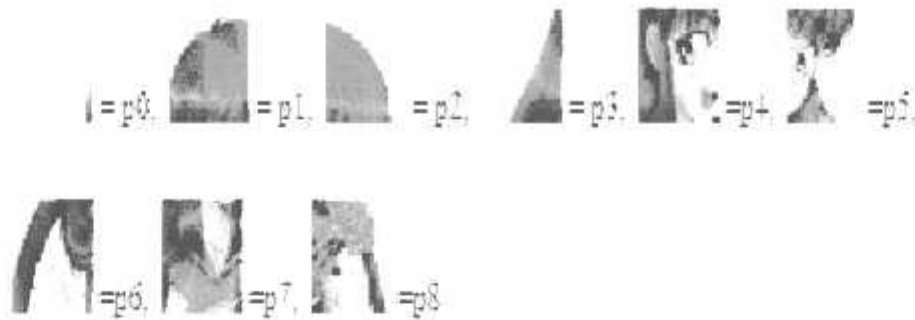
Gambar 3.10 Tampil *score*

3.5.2 Proses Penentuan / Penempatan Gambar

Untuk penentuan gambar kita akan menentukan proses pengacakan atau pertukaran potongan gambar, dengan menggunakan metode SWAP. Pada gambar pertama proses pengacakan potongan tidak terlalu rumit, sedangkan pada gambar berikutnya dan sampai pada gambar terakhir proses pertukaran atau pengacakannya lebih susah dari gambar-gambar sebelumnya.

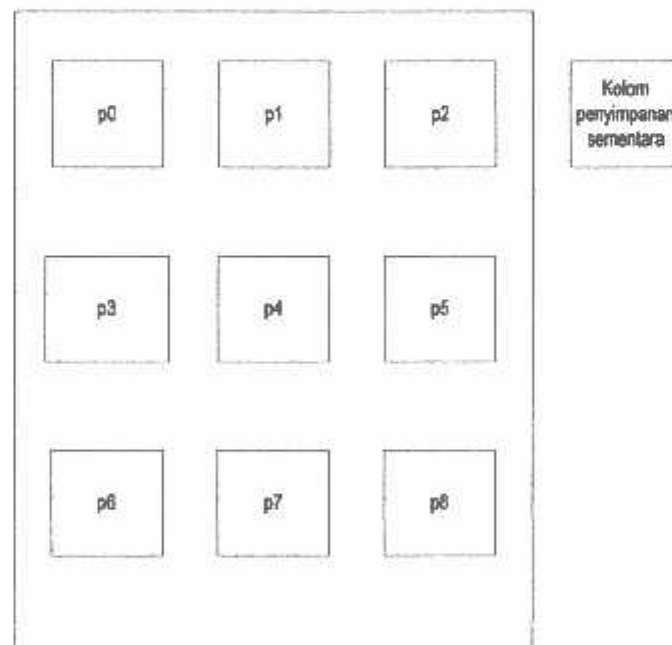
Pada masing-masing potongan gambar sudah diberi kode atau inisial tertentu dan jika pada saat pemanggilan kode tersebut, maka potongan gambar dapat berpindah secara otomatis sesuai sistem yang dibuat pada aplikasi game puzzle ini.

Dalam pengkodean atau penginisialan potongan gambar dapat kita lihat pada gambar 3.11.



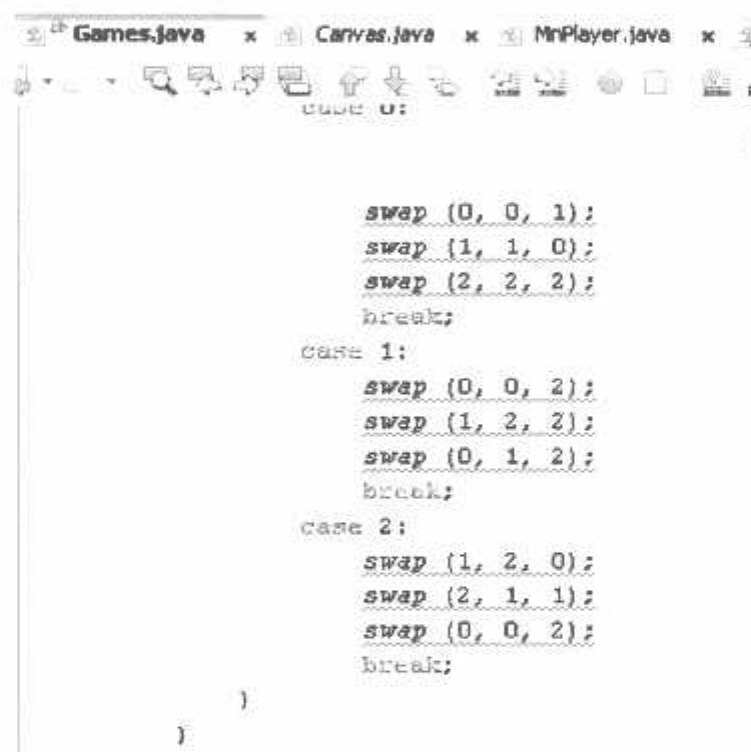
Gambar 3.11 potongan gambar dan inisial potongan

Adapun potongan gambar utuh sebelum proses penempatan dilakukan. Dalam gambar 3.12 dapat kita lihat bahwa adanya kolom penyimpanan potongan gambar sementara yang berfungsi pada saat pengacakan potongan gambar, potongan gambar mana yang akan dikosongkan dan potongan gambar tersebut berpindah pada kolom penyimpanan potongan gambar sementara.



Gambar 3.12 Potongan Gambar Utuh Sebelum Proses SWAP

Setelah adanya proses penempatan menggunakan metode SWAP maka posisi potongan gambar secara otomatis sudah teracak dan proses ini dilakukan pada saat pengkodean dalam pembuatan aplikasi game puzzle ini. Dengan memasukkan inisial-inisial potongan gambar tersebut penulis melakukan pengacakan dalam pembuatan program yang nantinya proses pengacakan ini berjalan secara otomatis dan penggunaan aplikasi game puzzle dalam berjalan dengan lancar. Sedangkan pengacakan tersebut sudah dilakukan atau ditentukan pada saat pembuatan program. Coding pengacakan menggunakan metode SWAP dapat dilihat pada gambar 3.13.



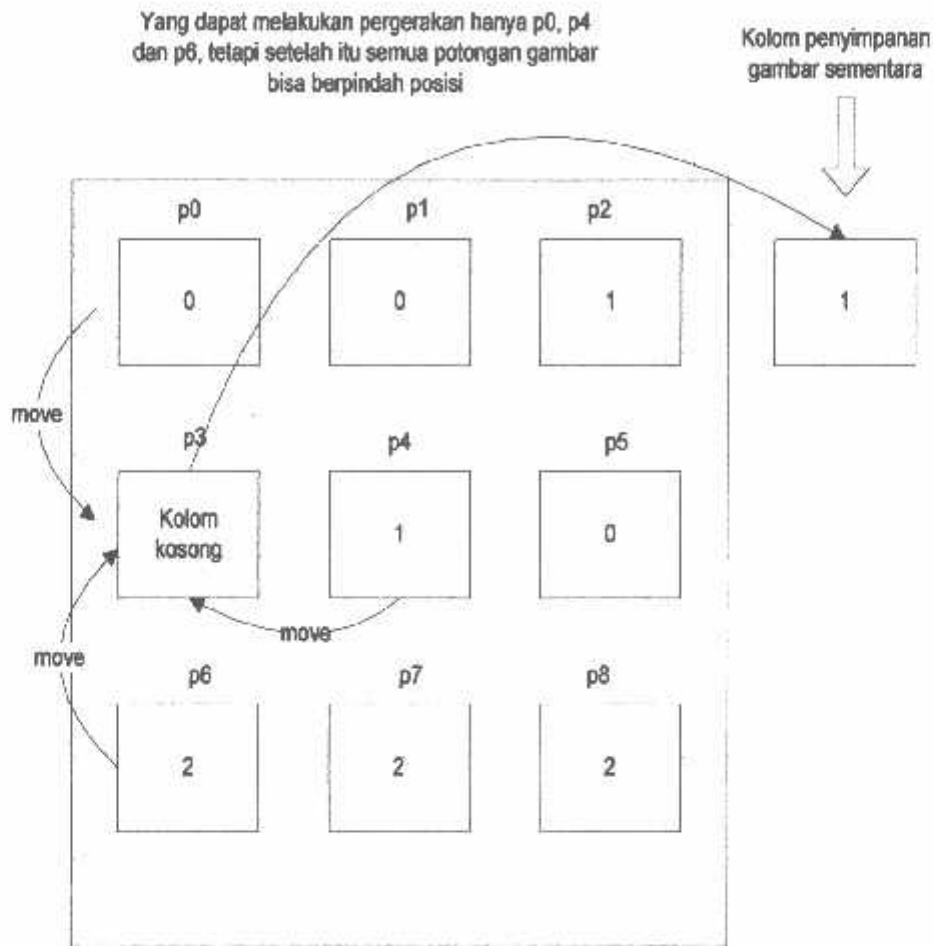
```

Games.java x Canvas.java x MrPlayer.java x
case 0:
    swap (0, 0, 1);
    swap (1, 1, 0);
    swap (2, 2, 2);
    break;
case 1:
    swap (0, 0, 2);
    swap (1, 2, 2);
    swap (0, 1, 2);
    break;
case 2:
    swap (1, 2, 0);
    swap (2, 1, 1);
    swap (0, 0, 2);
    break;
    )
)

```

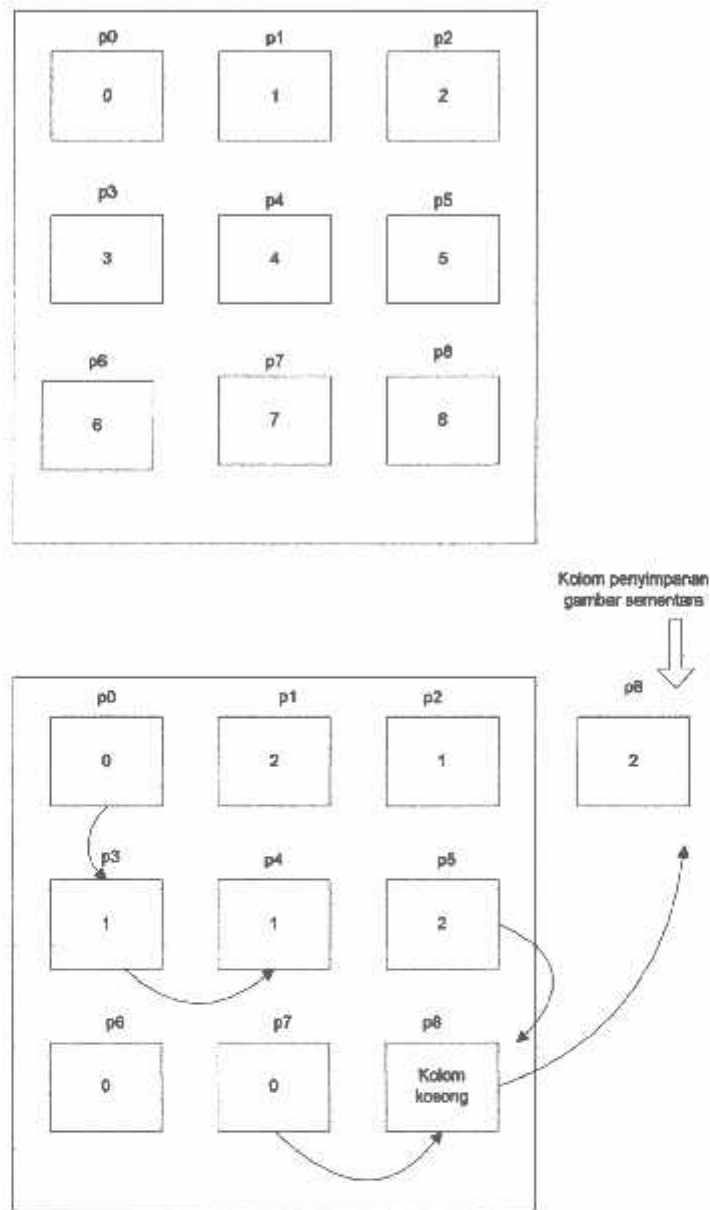
Gambar 3.13 Coding Metode SWAP pada Game Puzzle

Setelah melakukan proses pengacakan pada aplikasi game puzzle, maka nantinya akan teracak secara otomatis sesuai dengan perintah yang dibuat pada saat pengkodean sebelumnya. Gambar 3.13 menunjukkan bahwa proses pengacakan potongan gambar sudah terjadi.



Gambar 3.14 Proses Pengacakan Potongan Gambar

Gambar 3.14 menunjukkan adanya metode *swap* yang digunakan dengan menghilangkan atau mengosongkan salah satu kolom yaitu kolom p3, maka terjadilah pengacakan yang sesuai dengan *coding case 0*, dan jika pembuat menghilangkan p8 maka akan terjadi pengacakan seperti gambar 3.15.



Gambar 3.15 Pengacakan Dengan Menghilangkan Potongan p8

Proses pengacakan menghilangkan potongan pecahan gambar p8 dapat kita tentukan coding pengacakannya, dan dalam proses ini dimana ketika p8 ditarik maka p5, p7, dan p1 dapat melakukan pergerakan atau perpindahan tempat dan apa bila potongan pecahan sebelumnya dapat bergerak maka selanjutnya pecahan berikutnya dapat ditentukan penempatannya atau dapat berpindah terus-menerus. Dari percobaan pengacakan didapatkan *coding swap* sebagai berikut :

Format View Help

```
//kelem,baris,kelem,baris
```

```
case 1:
    swap (0, 2, 1);
    swap (1, 1, 2);
    swap (0, 0, 2);
    break;
```

```
)
```

Gambar 3.16 Coding Percobaan Menghilangkan Potongan p8

3.5.3 Proses Penentuan *Score*

Sedangkan pada proses penentuan *score* ini mengambil hitungan/acuan pada waktu (time), dimana waktu tersebut yang akan menentukan *score* dan mengakumulasikan hasil *score* keseluruhan.

Alasan dalam perhitungan *score* ini, agar dapat meningkatkan kemampuan seseorang dalam berfikir dan memanfaatkan waktu pada saat bermain.

Dalam game *puzzle* gambar, ditentukan batas waktu (*limited*), dimana fungsi pembatasan waktu tersebut pada menit ke-10, yang berarti jika player 1 dan player 2 tidak dapat menyelesaikan permainan *puzzle* dalam waktu 10 menit, maka *score* akan menjadi 0 (nol). Untuk proses perhitungan *score* sudah ditentukan dan diatur oleh sistem. Pada perhitungan dibawah dapat kita lihat pembuktian hasil *score* sebagai berikut :

Tabel 3.1 Rumus Perhitungan *Score* Pada *Game Puzzle* Gambar

Hasil *Score* x 6 detik / potongan *score* = total detik

$$1000 \times 6 / 10 = 600 \text{ detik}$$

Jika 10 menit sama dengan 600 detik, maka hasilnya ketika pada menit ke 10, *score* akan menjadi 0, dikarenakan adanya pengurangan 10 *score* per 6 detik.

Adapun contoh pada perhitungan score ini, jika hasil score 720 maka tentukan berapa detik waktu yang akan kita dapatkan.

Hasil Score x 6 detik / potongan score = total detik

$$720 \times 6 / 10 = 432 \text{ detik}$$

Rumus Pembuktiannya :

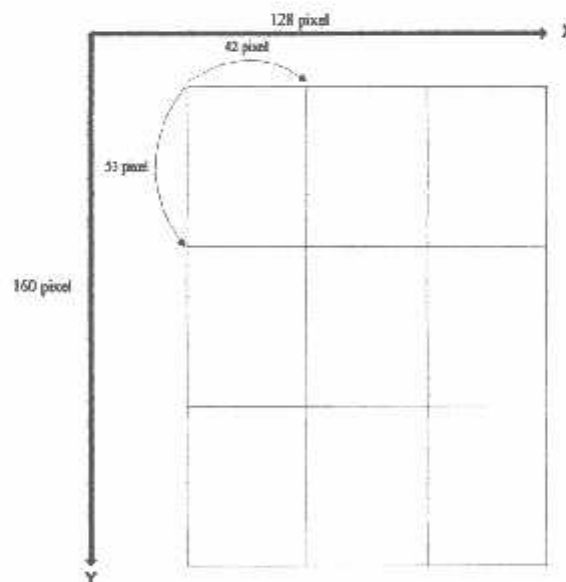
Total detik / 6 detik x Potongan score = Hasil score

Jadi :

$$432 / 6 \times 10 = 720 \text{ score}$$

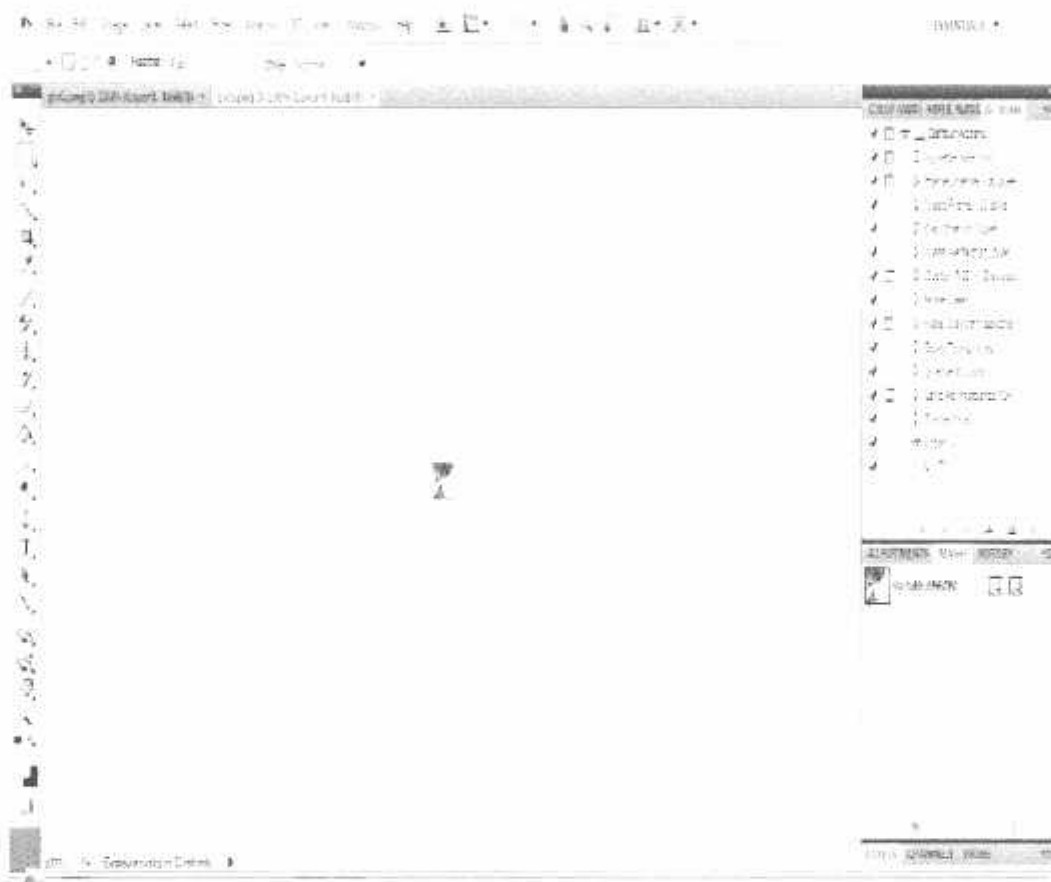
3.5.4 Pemotongan Gambar

Pemotongan gambar sendiri dilakukan dengan menggunakan *photoshop*, pada pemotongan tersebut akan dibagi sesuai dengan gambar yang sudah ditentukan, yaitu 42x53 pixel sebanyak 3x3 maka menjadi 9 pecahan yang akan diacak. Pemotongan gambar dilakukan sesuai dengan *pixel hand phone* apa yang akan kita gunakan maka dari situ kita dapat menentukan berapa kali berapa potongan yang diperlukan sehingga sesuai dengan *hand phone* tersebut. Dapat dilihat pada gambar 3.15 dibawah.



Gambar 3.17 Ukuran *Pixel* yang akan Dipotong

Pada pemotongan ini penulis membagi 3 pecahan tiap resolusi *handphone* yang digunakan, karna kotak potongan pecahan gambar yang dibutuhkan pembuatan game ini hanya 9 potongan gambar. Pada *resolusi handphone* yang kita gunakan 128 x 160 pixel maka $128 / 3$ dan $160 / 3$, dan hasilnya akan disesuaikan pada saat pemotongan gambar tersebut. Pemotongan gambar sendiri akan menggunakan aplikasi *adobe photoshop* yang mana pada saat dipotong akan diukur sesuai dengan ketentuan yang sudah kita buat. Contoh pemotongan pecahan gambar dapat dilihat pada gambar 3.16.



Gambar 3.18 Potongan GambarPuzzle Menggunakan *Photoshop*

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

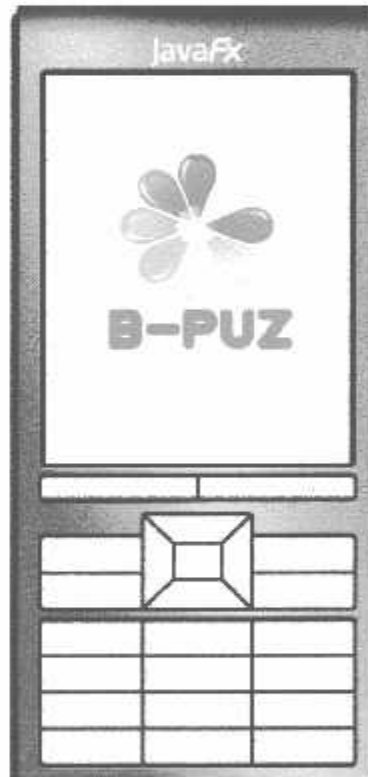
4.1. Implementasi sistem

Tahap implementasi merupakan tahap akhir dari proses membangun sebuah sistem. Pada tahap ini semua rancangan yang sudah dihasilkan dan diterjemahkan dengan menggunakan bahasa pemrograman sehingga menjadi sebuah sistem.

Pada implementasi program aplikasi game puzzle menggunakan emulator dari netbeans IDE 6.9 memiliki tampilan sebagai berikut :

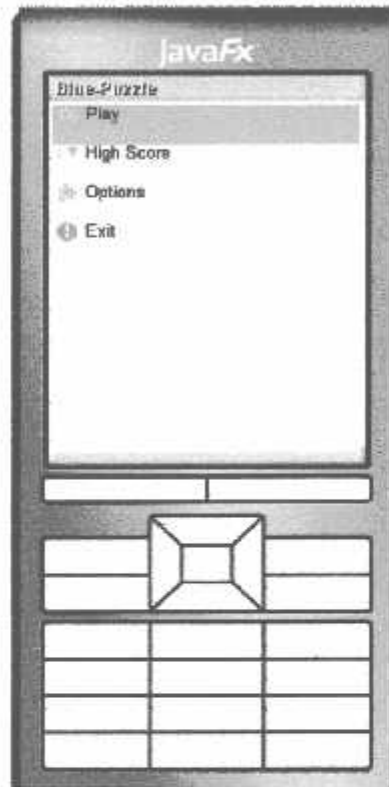
1) Tampilan splash screen

Tampilan gambar 4.1 adalah tampilan awal pada game ini itu untuk pembukaan sebelum masuk kemenu utama.



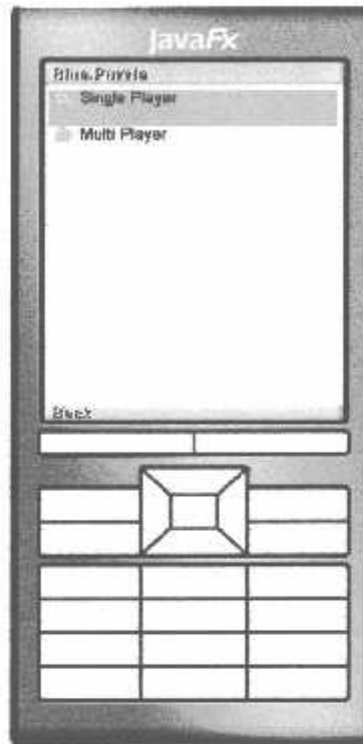
Gambar 4.1 Tampilan *splash screen* puzzle

- 2) Tampilan menu utama
pada gambar 4.2. menjelaskan dimana tampilan menu utama mempunyai optins-option atau ada 4 menu pilihan, yaitu ; *play*, *high score*, *options*, *exit*, dan tampilan game *puzzle* yang sudah dirancang dan dijalankan oleh sistem.



Gambar 4.2. Tampilan menu utama

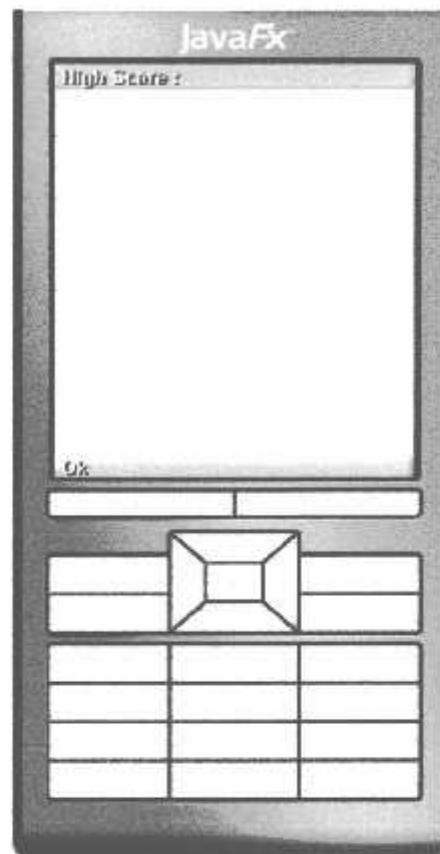
- 3) Tampilan *Play*
Tampilan *play* adalah sistem untuk memulai permainan dalam tampilan ini mempunyai menu pilihan, yaitu ; *single player* atau *multi player*, dan fungsi tampilan tersebut menjelaskan adanya pemilihan jenis permainan. Bisa dilihat pada gambar 4.3.



Gambar 4.3. Tampilan menu *Play*

4) Tampilan *High Score*

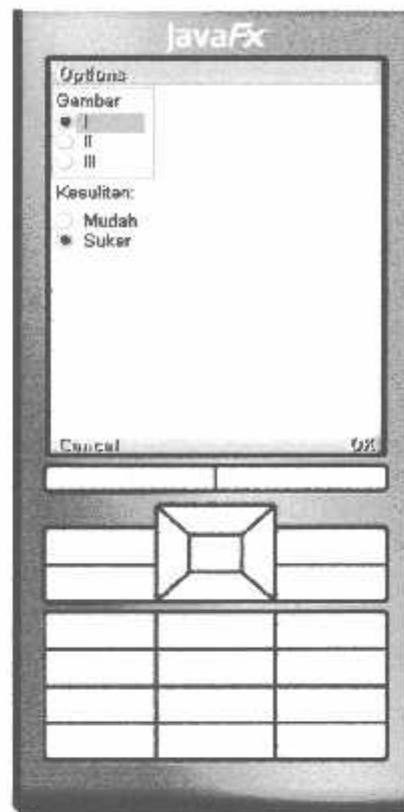
Tampilan *high score* ini menjelaskan tentang adanya *score* tertinggi dalam permainan *puzzle*. Fungsi tampilan ini agar pemain merasa penasaran sehingga akan mencoba terus-menerus dan mempunyai *score* tertinggi. Bisa dilihat pada gambar 4.4.



Gambar 4.4. Tampilan High Score

5) Tampilan *Options*

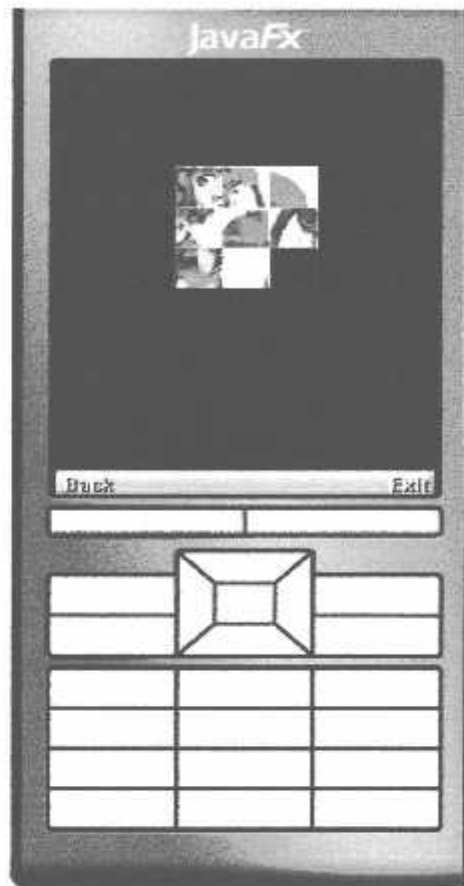
Tampilan pada gambar 4.5 menyatakan bahwa adanya pilihan dalam menu *options*, yang mana pemain dapat memilih gambar dan tingkat kesulitan sesuai keinginan pengguna (*player*).



Gambar 4.5 Tampilan *menu Options*

6) Tampilan *single player*

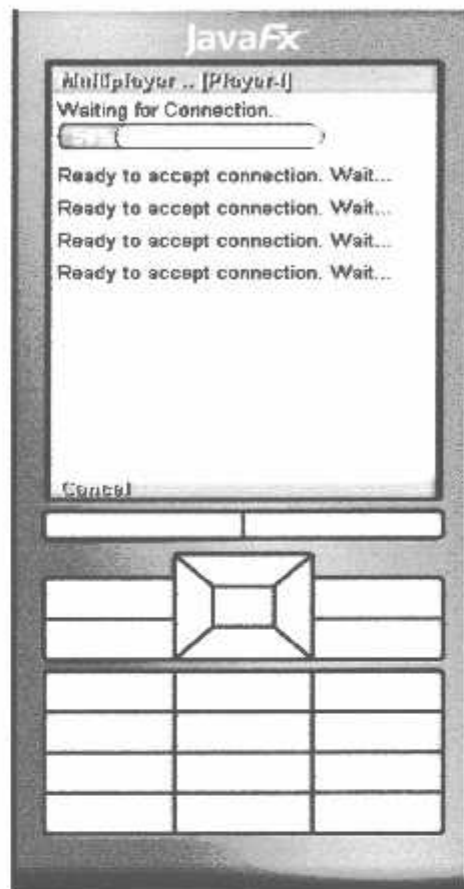
Dimana tampilan menu *single player* ini menyatakan mulainya permainan *puzzle*. Pada tampilan tersebut sudah teracak secara *random* dan siap untuk dimainkan.



Gambar 4.6. Tampilan permainan *single player*

7) Tampilan *multi player*

Tampilan *multiplayer* mempunyai pilihan sendiri dimana pilihan tersebut berisi 2 options, yaitu ; *player 1* dan *player 2*. Fungsinya untuk pengkoneksian *bluetooth* agar dapat bermain secara bersamaan. Dapat dilihat pada gambar 4.7.



Gambar 4.8. Tampilan pengoneksian *bluetooth*

9) Tampilan gambar 1

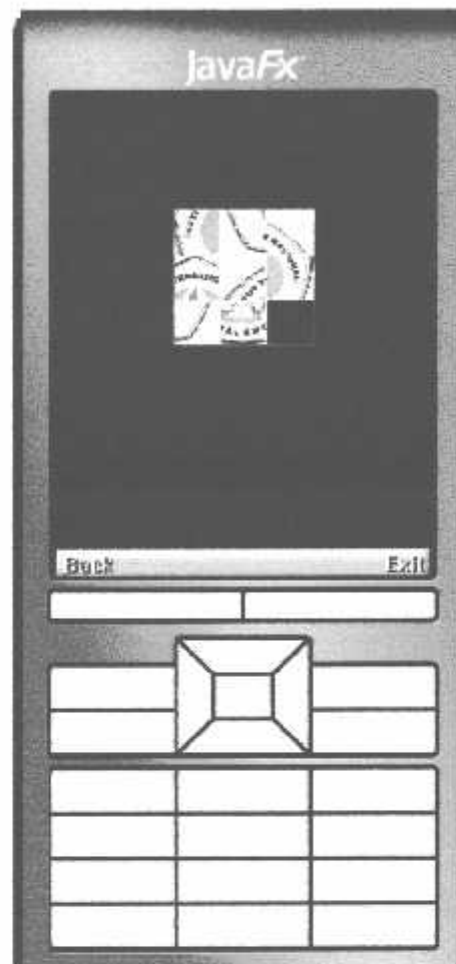
Tampilan ini merupakan gambar pertama pada *multiplayer game*, dimana dalam *games* ini ada 2 *player* yang berlomba. Dengan tingkat kesulitan yang mudah, tetapi tergantung pilihan pengguna atau pemain tersebut untuk memilih tingkat kesulitannya, bisa dilihat pada gambar 4.9.



Gambar 4.9. Tampilan gambar 1

10) Tampilan gambar 2

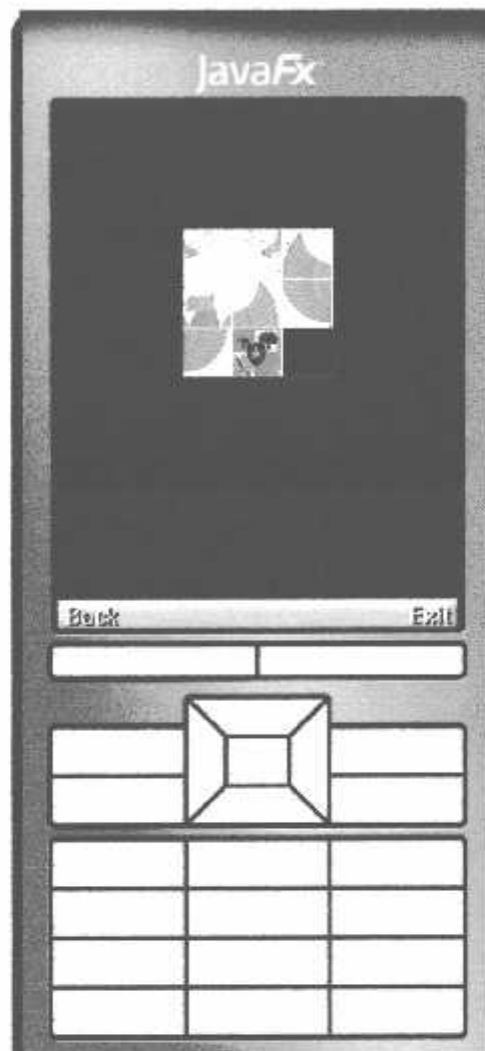
Untuk tampilan gambar 2, pembuat memasukkan gambar logo ITN malang, tingkat kesulitannya agak begitu rumit dikarenakan pengacakan gambar lebih susah dari pada gambar 1, dimana pemain dituntut agar lebih kreatif dalam menyelesaikan *game puzzle* tersebut. Dapat dilihat pada gambar 4.10.



Gambar 4.10. Tampilan gambar 2

11) Tampilan gambar 3

Pada tampilan gambar 3, pembuat merancang tingkat kesusahan yang lebih rumit (sulit) dari pada gambar 1 dan gambar 2. Dimana gambar 3 ini adalah gambar terakhir dalam *game puzzle* ini. Dapat dilihat pada gambar 4.11.



Gambar 4.11. Tampilan gambar 3

4.2. Pengujian

Pengujian merupakan tahap yang utama dalam pembuatan suatu aplikasi perangkat lunak. Hasil pengujian yang didapat, akan dijadikan sebagai tolak ukur dalam proses pengembangan selanjutnya. Pengujian ini dilakukan untuk mengetahui hasil yang didapat dari perangkat lunak yang telah dibuat.

4.2.1 Lingkungan Pengujian

Lingkungan pengujian merupakan penjelasan alat bantu apa saja yang digunakan dalam proses pembuatan aplikasi perangkat lunak game puzzle.

4.2.2 Perangkat Lunak

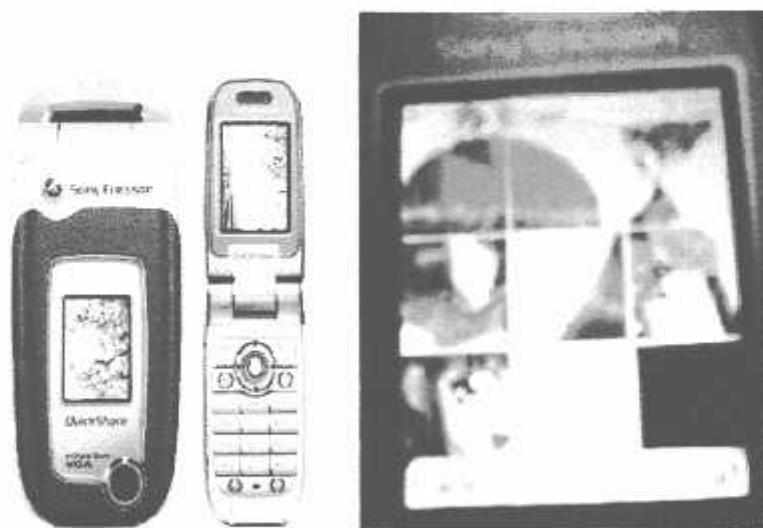
Aplikasi yang telah dibuat, diujikan dalam lingkungan perangkat lunak dengan spesifikasi sebagai berikut :

- *Operating System* yang digunakan adalah *Microsoft Windows XP 2*.
- *Software* yang digunakan untuk membuat aplikasi ini adalah *J2ME Wireless Toolkit 2.2*.
- *NetBeans IDE 6.9*, digunakan untuk pengetikan *source code* program.

4.3. Pengujian Aplikasi

Untuk pengujian aplikasi pembuat menggunakan 2 *hand phone* (ponsel), dimana *hand phone* tersebut *support* aplikasi JAVA dan tentunya ada aplikasi *bluetooth* pada *handphone* yang akan diuji. Dalam pengujian ini pembuat menguji aplikasi dengan menggunakan *handphone* merk :

4.3.1 Sony Ericsson Z_520i



Gambar 4.12. Sony ericsson Z_520i

Spesifikasi *handphone sony ericsson Z_520i* sebagai berikut :

- Type TFT, 65k colors
- Resolusi 128x160 pixel
- Memory handphone 16 MB
- Support java MIDP 2.0
- Bluetooth V2.0

Dari hasil pengujian *handphone sony ericsson Z_520i* aplikasi game puzzle dapat dimainkan dengan lancar dan semua fungsi keypad berjalan dengan normal dan tampilan menu sesuai dengan sistem yang dibuat.

4.3.2 Sony ericsson K_530i



Gambar 4.13. Sony ericsson K_530i

Dari hasil pengujian *handphone sony ericsson K-530i* aplikasi *game puzzle* dapat dimainkan dengan lancar dan baik. Tetapi adanya perbedaan hasil tampilan pada layar sony ericsson K_530i, dimana layar tidak *fullscreen* melainkan hanya berbentuk *center* (ditengah) dan meninggalkan sisa layar hitam disisi kanan/kiri serta sisi atas bawah. Hal ini disebabkan karena tingginya resolusi pada *handphone K_530i*.

Spesifikasi *handphone sony ericsson K_530i* sebagai berikut :

- Type TFT 65k colors

- Resolusi 176x220 pixel
- Memori internal 16 Mb, card slot MsD up to 2Gb
- Support java MIDP 2.0
- Bluetooth V2.0, with A2DP

4.3.3 Pengujian aplikasi multiplayer via Bluetooth

Gambar 4.14 menjelaskan proses koneksi *multiplayer* dengan *via bluetooth*, dimana player 1 akan mencari player 2 sebagai koneksinya, jika saling terkoneksi maka *palyer* 1 akan menjadi *server* dan *player* 2 akan menjadi *client* proses ini dapat dilihat pada gambar 4.15. Pada saat koneksi terhubung maka permainan dapat dimainkan secara bersamaan.



Gambar 4.14 Tampilan proses koneksi *bluetooth multiplayer*



Gambar 4.15 Tampilan Proses *Request* Penerimaan Koncksi Game

Pada gambar 4.15 dapat kita lihat adanya proses pengoneksian antara player 1 dan player jika player 2 meginzinkan (merespon) permintaan player maka terhubunglah dengan perangkat *bluetooth* yang sudah aktif sehingga terbentuklah jaringan kecil tersebut dan game dapat dimainkan.

4.3.4 Pengujian Aplikasi Pengiriman *Score*

Dalam hal pengujian pengiriman *score* yang dilakukan secara multiplayer pada player 1 dan player 2 sudah terbukti karena munculnya *score* pada player 2 jika player 1 menang. Hasil pengujian dapat dilihat pada gambar 4.16.



Gambar 4.16 Tampilan hasil pengiriman *score* pada player 2



Gambar 4.17 Tampilan hasil pengiriman *score* pada player 1

4.3.5 Pengujian Pada Handphone BB

Pengujian aplikasi game *puzzle* pada *handphone blackberry* sangatlah penting, dikarenakan sebagai alat pembanding dengan *handphone* merk lainnya.

Setelah dilakukan pengujian ada perbedaan pada tampilan layar yang sangat kecil dan tampilan gambar *puzzle* tepat berada ditengah-tengah lcd dan menyisakan ruang sisi kanan-kiri, atas-bawah. Hal tersebut disebabkan adanya tingkat resolusi yang terlalu tinggi maka tidak sesuai dengan apa yang diharapkan dan dalam pengujian ini juga didapati adanya fungsi *bluetooth* yang tidak bisa digunakan karena adanya perbedaan servis *bugs bluetooth* yang berbeda pada setiap merk *handphone* tetapi untuk fungsi keypad yang digunakan berjalan dengan normal.

4.4. Pengujian pada jarak 1 meter fungsi bluetooth

Pada saat pembuat menguji aplikasi game *puzzle* dengan jarak 1 meter, hasilnya sangat baik dalam waktu kurang dari 1 menit jaringan *bluetooth* sudah terhubung dan game dapat dimainkan langsung. Untuk pengiriman *score* dapat terkirim pada player 2. Pada gambar 4.18 terlihat contoh pengujian yang telah dilakukan oleh penulis.



Gambar 4.18 Pengujian Bluetooth dengan jarak 1 meter

4.5. Pengujian pada jarak 2 meter fungsi bluetooth

Pada saat pengujian dilakukan dengan jarak 2 meter hasil yang didapatkan kurang lebih sama dengan pengujian 1 meter, hal ini disebabkan jarak tidak terlalu jauh dan tidak ada gangguan pada saat pengujian.

4.6. Pengujian pada jarak 3 meter fungsi Bluetooth

Dari hasil pengujian 3 meter tidak ada perbedaan dengan hasil pengujian-pengujian sebelumnya. Jadi proses penggunaan perangkat bluetooth berjalan dengan baik.

4.7. Pengujian pada 4 meter fungsi Bluetooth

Dari hasil pengujian dengan jarak 4 meter fungsi *bluetooth* berjalan lebih lama dari pengujian-pengujian sebelumnya sekitar 2-3 menit dan hasil yang didapatkan kurang begitu baik. Kekkita player 1 selesai pada gambar pertama hasil dari player 1 tersebut tidak terkirim pada player 2 hal ini disebabkan pada saat proses pengiriman score adanya gangguan atau terputusnya jaringan *bluetooth*.

4.8. Pengujian pada jarak 5 meter fungsi bluetooth

Pada saat pengujian bluetooth dengan jarak 5 meter kedua perangkat bluetooth sudah aktif dan kedua device tersebut sudah saling mengizinkan tetapi tidak adanya respon dari antara player 1 dan player 2. Hal ini disebabkan jarak terlalu jauh dan tidak sampainya piconet pada device tersebut.

4.9. Pengujian multiplayer bluetooth dengan terhalang tembok

Dari hasil pengujian terhalang dengan tembok setebal 20 cm dan dengan jarak 1-3 meter tidak ada gangguan hasil yang didapatkan sama halnya dengan pengujian 1 -3 meter. Hal ini disebabkan sangat kecil pengaruh dari penghalang tersebut. Dapat dilihat pada gambar 4.19.



Gambar 4.19 Pengujian Bluetooth terhalang tembok

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan data dan hasil pengujian aplikasi yang di buat maka dapat diberikan kesimpulan dan saran sebagai berikut :

1. Dari hasil pengujian menggunakan *handphone sony ericsson Z-520i* tampilan layar gambar game *puzzle* utuh sesuai dengan sistem yang dirancang dan fungsi keypad berjalan dengan normal.
2. Sedangkan pada *handphone sony ericsson K-530i* tampilan layar gambar game *puzzle* tidak utuh tetapi fungsi keypad berjalan dengan normal.
3. Perbedaan servis *bug bluetooth* pada serial dan merk handpone mempengaruhi koneksi *device bluetooth*.
4. Dari hasil pengujian pada jarak 1-4 meter fungsi pengoneksian bluetooth dan pengiriman *score* berjalan dengan normal.
5. Pengujian pada jarak 5 meter tidak berhasil dikarenakan jarak terlalu jauh.
6. Pengujian pada jarak 1-3 meter dan terhalang tembok setebal 20cm, pengoneksian *bluetooth* dan pengiriman *score* berjalan dengan normal.
7. Pengujian yang dilakukan pada handpone dengan merk *blackberry 8250* hasilnya tidak sesuai yang diinginkan dikarenakan adanya servis bug *bluetooth* dan *support java* yang berbeda antara *handphone* tersebut.
8. Minimal menggunakan *hand phone support java MDIP 2.0* dengan resolusi minimal 128x160 pixel maka hasil tampilan memuaskan.

5.2. Saran

Saran yang dapat diberikan untuk pengembangan yang mungkin dapat dilakukan pada game *puzzle* gambar sebagai berikut :

1. Game ini dapat dikembangkan dengan tingkat kesulitan penambahan kotak dan membuat suatu sistem pengacakan yang lebih rumit.
 2. Lebih pengguna dalam bermain atau bukan hanya 2 *player* yang diperuntukan pada game *puzzle* ini.
 3. Pengembangan game bukan hanya dengan *bluetooth* tetapi menggunakan jaringan internet pada *mobile* sehingga jumlah pemain lebih banyak.
-

DAFTAR PUSTAKA

- [1] Beny Hermawan, 2004, *JAVA 2 & Object Oriented Programming*, Penerbit Andi Yogyakarta
- [2] **Ir. Yuniar Supardi**, 2008. *Pemrograman Handphone dengan J2ME*. Penerbit Elek Media Komputindo
- [3] M. Shalahuddin, Rosa A.S, 2009. *Pemrograman J2ME: Belajar Cepat Pemrograman Perangkat Telekomunikasi Mobile*. Penerbit Informatika.
- [4] **M. Shalahuddin & Rosa A.S.**, 2008. *Pemrograman J2ME (Belajar Cepat Pemrograman Perangkat Telekomunikasi Mobile)*, Penerbit Informatika.
- [5] Sri Widayati, MT. Budi Darytamo, MT.2004. *Pemrograman Berorientasi Obyek dengan Java 2 Platform*, Penerbit Ghanesa.
- [6] <http://zakibabyhouse.com/promo/manfaat-bermain-puzzle/>, terbit 12 juni 2008, diakses tanggal 20 juni 2011
- [7] <http://ilmukomputer.org/2006/09/28/tips-tips-j2me/> (Published: September 28, 2006 · Category: Pemrograman Java) diakses tanggal 15 juni 2011
- [8] <http://mtamim.wordpress.com/2010/05/20/download-ebook-pemrograman-j2me/>. Terbit Mar 17 2009, diakses tanggal 14 juni 2011
- [9] <http://id.wikipedia.org/wiki/Bluetooth> terbit 30 mei 2011, diakses tanggal 14 juni 2011
- [10] <http://www.forum-handphone.org/t236/pengertian-hardware-dan-software-pada-handphone-8465.html>. terbit 22 november 2008, diakses tanggal 20 juni 2011.
- [11] <http://irma14.blogspot.com/2008/09/pengertian-dasar-dan-simbol-flowchart.html>, diakses tanggal 14 juni 2011.



SOURCE CODE

- *Code MIDlet game puzzle*

```
import java.util.Random;

import javax.microedition.lcdui.Command;

import javax.microedition.lcdui.CommandListener;

import javax.microedition.lcdui.Display;

import javax.microedition.lcdui.Displayable;

import javax.microedition.lcdui.Image;

import javax.microedition.lcdui.List;

import javax.microedition.midlet.*;

public class BlueZzle extends MIDlet implements CommandListener {

    public Splash fSplash;

    private String[] strMenu={"Play","High Score","Options","Exit"};

    public Display display;

    public List listMenu,listPlayer;

    public gCanvas cPlayer;

    public Options foptions;

    public ScoreView fScore;

    public MnPlayer fMenu;

    public BlueControl BleuConn;

    public Score rms_score;
```

```

private static Random randLevel= new Random();

public String Status;

public String Score1;

public boolean lanjut;

public void startApp() {

    display = Display.getDisplay(this);

    fSplash = new Splash(this);

    fSplash.show();

    foptions = new Options ( this);

    rms_score =new Score();

}

static int random(int size)

{

    return (randLevel.nextInt() & 0x7FFFFFFF) % size;

}

public void tampilMenu(){

    if (listMenu!=null ) listMenu=null;

    listMenu =new List("Blue-Puzzle",List.IMPLICIT);

    try{

        for (int i =0;i<strMenu.length ;i++){

            listMenu.append(strMenu[i]+"\\n",

Image.createImage("/images/"+String.valueOf(i)+".png"));

        }

    }catch(Exception cx){}

```

```

listMenu.setCommandListener(this);

this.display.setCurrent(listMenu);
}

public boolean tampilMenuPlayer()

if (listPlayer!=null ) listPlayer=null;

listPlayer =new List("Blue-Puzzle",List.IMPLICIT);

try{

listPlayer.append("SinglePlayer\n",
Image.createImage("/images/"+String.valueOf(0)+".png"));

listPlayer.append("Multi Player\n",
Image.createImage("/images/"+String.valueOf(2)+".png"));

} catch(Exception ex){}

listPlayer.setSelectedIndex(0, false);

listPlayer.setCommandListener(this);

listPlayer.addCommand(new Command("Back", Command.BACK,0));

this.display.setCurrent(listPlayer);

return true;

}

public void pauseApp() {

}

public void destroyApp(boolean unconditional) {

if (cPlayer!=null) cPlayer.stop();

```

```
}  
  
public void commandAction(Command c, Displayable d) {  
  
    if (c==List.SELECT_COMMAND){  
  
        if (d.equals(listMenu)) {  
  
            switch (listMenu.getSelectedIndex()){  
  
                case 0:  
  
                    tampilMenuPlayer();  
  
                    break;  
  
                case 1:  
  
                    fScore =new ScoreView(this);  
  
                    fScore.show();  
  
                    break;  
  
                case 2:  
  
                    display.setCurrent(foptions);  
  
                    break;  
  
                case 3:  
  
                    notifyDestroyed();  
  
                    break;  
  
            }  
  
        }  
  
        if (d.equals(listPlayer)) {  
  
            switch (listPlayer.getSelectedIndex()){  
  
                case 0:  
  
                    int level = random(3);
```

```

        cPlayer = new
gCanvas(this,foptions.cgType.getSelectedIndex()+1,foptions.cgLevel.getSelectedIndex()+1);

        // cGames – new Games(this,level);

        display.setCurrent(cPlayer);

        break;

    case 1:

        fMenu = new MnPlayer(this);

        fMenu.show();

    }

}

} else{

    tampilMenu();

}

}

}

```

- ***Code render bluetooth***

```

public void run()
{
    try
    {
        DataInputStream datain = endpt.con.openDataInputStream();

        while ( !done )
        {
            log("waiting for next signal from "+endpt.remoteName);

            int signal = datain.readInt();

            if ( signal == BlueLayer.SIGNAL_MESSAGE )
            {
                String s = datain.readUTF();
            }
        }
    }
}

```

```

        BluePacket packet = new BluePacket( BlueLayer.SIGNAL_MESSAGE,
endpt.remoteName, s );

        log("read in MESSAGE string '"+s+"' from "+endpt.remoteName);
        if (BlueControl.instance.isReady==true){
            if (s.indexOf("READY")!=-1 && BlueControl.instance.BCon.cPlayer
==null){

BlueControl.instance.btnet.sendString(BlueControl.instance.Names+":READY");
                if ("Player-1".equals(BlueControl.instance.Names)){
                    int level= BlueZzle.random(3);

BlueControl.instance.btnet.sendString(BlueControl.instance.Names+":LEVEL,"+level)
;
                    Thread.sleep(10000);
                    BlueControl.instance.BCon.cPlayer = new
gCanvas(BlueControl.instance.BCon , 1, 1);

BlueControl.instance.BCon.display.setCurrent(BlueControl.instance.BCon.cPlayer);
                }
            }
            if (s.indexOf("LEVEL")!=-1){
                int level= Integer.parseInt(s.substring(s.indexOf(";")+1));

                BlueControl.instance.BCon.cPlayer = new
gCanvas(BlueControl.instance.BCon , 1, 1);

BlueControl.instance.BCon.display.setCurrent(BlueControl.instance.BCon.cPlayer);
            }
            if (s.indexOf("SCORE")!=-1){
                String name = s.substring(0,s.indexOf(':'));
                int dur= Integer.parseInt(s.substring(s.indexOf(";")+1));
                BlueControl.instance.BCon.rms_score.setScore(name, dur);
            }
            if (s.indexOf("FINISH")!=-1){
                String name = s.substring(0,s.indexOf(':'));
                Ticker t= new Ticker(name+ " Finish");
                BlueControl.instance.setTicker(t);
            }
        }
        endpt.callback.handleAction( BlueAdapter.EVENT_RECEIVED, endpt, packet
);

    } else if ( signal == BlueLayer.SIGNAL_HANDSHAKE )
    {
        String s = datain.readUTF();
        log("read in HANDSHAKE name '"+s+"' from "+endpt.remoteName);
    }
}

```

```

endpt.remoteName = s;

endpt.putString( BlueLayer.SIGNAL_HANDSHAKE_ACK, endpt.localName );

endpt.callback.handleAction( BlueAdapter.EVENT_JOIN, endpt, null );

} else if ( signal == BlueLayer.SIGNAL_TERMINATE )
{
log("read in TERMINATE from "+endpt.remoteName);

endpt.putString( BlueLayer.SIGNAL_TERMINATE_ACK, "end" );

endpt.callback.handleAction( BlueAdapter.EVENT_LEAVE, endpt, null );

endpt.btnet.cleanupRemoteEndPoint( endpt );

stop();

} else if ( signal == BlueLayer.SIGNAL_HANDSHAKE_ACK )
{

String s = datain.readUTF();
log("read in HANDSHAKE_ACK name "+s+" from "+endpt.remoteName);

endpt.remoteName = s;

} else if ( signal == BlueLayer.SIGNAL_TERMINATE_ACK )
{

System.out.println("read in TERMINATE_ACK from "+endpt.remoteName);

} else
{
log("Unkonwn signal, probably connection closed");
}

}

datain.close();
} catch (Exception e)
{
log(e.getClass().getName()+" "+e.getMessage());
}
}

```

```

        log("reader thread exit for "+endpt.remoteName);
    }
    private static void log( String s)
    {
        System.out.println("Reader: "+s);
    }
}

```

- *Code sender bluetooth*

```

public void run()
{
    try
    {
        DataOutputStream dataout = endpt.con.openDataOutputStream();
        while( !done )
        {

            if ( ! endpt.peekString() )
            {
                synchronized (this) {
                    this.wait(5000);
                }
            }
            BluePacket s = endpt.getString();

            if( s != null )
            {

                log("sending signal "+s.signal+" string '"+s.msg+"' to "+endpt.remoteName);
                dataout.writeInt(s.signal);
                dataout.writeUTF(s.msg );
                dataout.flush();
            }

            if ( s != null && s.signal == BlueLayer.SIGNAL_TERMINATE )
            {
                stop();
            }

        }

        dataout.close();
    } catch (Exception e)
    {

```

```

    log(e.getClass().getName()+" "+e.getMessage());
}
log("sender thread exit for "+endpt.remoteName);
}
private static void log( String s)
{
    System.out.println("Sender: "+s);
}
}

```

- **Code pengiriman score**

```

public class Score {
    private RecordStore rs = null;
    private int maxRecord=5;

    public Score()
    {
    }

    private void close()
    {
        try {
            rs.closeRecordStore();
            rs=null;
        } catch (RecordStoreNotOpenException ex) {
        } catch (RecordStoreException ex) {
        }
    }
}
private String[] sort(Vector v)
{
    String[] arraySort=new String[v.size()];
    v.copyInto(arraySort);

    int a=0;
    int b=0;
    String abTemp=null;
    for(int i=0;i<arraySort.length-1;i++)
    {
        for(int j=1;j<arraySort.length;j++)
        {
            a=Integer.parseInt(arraySort[j].substring(1+arraySort[j].indexOf(":")));
            b=Integer.parseInt(arraySort[j-1].substring(1+arraySort[j-1].indexOf(":")));
            if(a<b)
            {
                abTemp=arraySort[j];
                arraySort[j]=arraySort[j-1];
            }
        }
    }
}

```

```

        arraySort[j-1]=abTemp;
    }
}

return arraySort;
}

public void setScore(String name, int score)
{

    try {
        rs = RecordStore.openRecordStore("myscore", true);
    } catch (RecordStoreException ex) {
    }

    String data=name+"."+score;
    try {
        rs.addRecord(data.getBytes(), 0, data.getBytes().length);
    } catch (RecordStoreNotOpenException ex) {
    } catch (RecordStoreException ex) {
    }
    this.close();

}

public String[] getScore()
{
    Vector vData = new Vector();
    try {

        rs = RecordStore.openRecordStore("myscore", true);

        RecordEnumeration re = rs.enumerateRecords(null, null, false);

        while (re.hasNextElement()) {
            vData.addElement(new String(re.nextRecord()));
        }
        re = rs.enumerateRecords(null, null, false);

        while (re.hasNextElement()) {
            rs.deleteRecord(re.nextRecordId());
        }

    } catch (RecordStoreException ex) {
    }

    this.close();
}

```

```

String[] dataScore=sort(vData);
for (int i = 0; i < dataScore.length; i++) {
    if(i<maxRecord)
    {
        this.setScore(dataScore[i].substring(0,dataScore[i].indexOf(":")),
Integer.parseInt(dataScore[i].substring(1+dataScore[i].indexOf(":"))));
    }
    else
    {
        dataScore[i]="";
    }
}
for (int i = 0; i < dataScore.length; i++) {
    dataScore[i]= toTime(
Integer.parseInt(dataScore[i].substring(1+dataScore[i].indexOf(":")))+ " " +
dataScore[i].substring(0,dataScore[i].indexOf(":")) :

}
return dataScore;
}
private String toTime(int valtime){
    int mi = (int) valtime/60;
    int sec = valtime -mi;
    return mi+" ":"+sec ;
}
}
}

```

- *Code gcanvas game puzzle*

```

public class gCanvas extends Canvas implements Runnable,CommandListener{
    public Image m_almg[];
    public int m_anCur[][];
    public int m_nHidX, m_nHidY;
    public int m_nSta;
    private Command ExitCommand,BackCommand;
    private BlueZzle midlet;
    int picutes;
    int levels;
    public gCanvas( BlueZzle mMidlet,int picutes, int levels )
    {
        midlet = mMidlet;
        this.levels = levels;
        this.picutes =picutes;
        try
        {
            ExitCommand = new Command("Exit", Command.EXIT, 0);

```

```

BackCommand = new Command("Back", Command.BACK, 1);
    addCommand(ExitCommand);
addCommand(BackCommand);
    setCommandListener(this);
    m_nHidX = 2;
    m_nHidY = 2;
    m_nSta = 0;
    InitCurrent();
    m_aImg = new Image[9];
    StringBuffer temp = null;
    for( int i = 0; i < 9; i ++ )
    {
        temp=new StringBuffer();
        temp.append("/image/puzzle/_")
        .append(picutes).append("/pic");
        temp.append(i);
        temp.append(".png");
        m_aImg[i] = Image.createImage(temp.toString());
    }
}
catch (Exception ex)
{
}
Thread thread = new Thread(this);
thread.start();
}
public void InitCurrent()
{
    Random random = new Random();
    m_anCur = new int[3][3] { {0,1,2}, {3,4,5}, {6,7,8} };
    int Rx, Ry, k, nTemp;
    for( int x = 0; x < 3; x ++ )
    {
        for( int y = 0; y < 3; y ++ )
        {
            k = random.nextInt();
            Rx = Math.abs(k % 3);
            k = random.nextInt();
            Ry = Math.abs(k % 3);
            if( Rx != x || Ry != y )
            {
                nTemp = m_anCur[y][x];
                m_anCur[y][x] = m_anCur[Ry][Rx];
                m_anCur[Ry][Rx] = nTemp;
            }
        }
    }
}
}

```

```

}
protected void paint(Graphics g)
{
    g.setColor(0);
    g.fillRect( 0, 0, getWidth(), getHeight() );
    int nImg = 0;

    switch( m_nSta )
    {
    case 0:
        for( int x = 0; x < 3; x++ )
        {
            for( int y = 0; y < 3; y++ )
            {
                if( x == m_nHidX && y == m_nHidY )
                    continue;
                nImg = m_anCur[y][x];
                if( nImg >= 0 && nImg < 9 )
                {
                    if( m_aImg[nImg] != null )
                        g.drawImage( m_aImg[nImg], x * 30, y * 30,
Graphics.LEFT|Graphics.TOP);
                }
            }
        }

        g.setColor(0xffffffff);
        for(int i = 0; i <= 3; i++)
        {
            g.drawLine( 0, i*30, 3*30, i*30);
            g.drawLine( i*30, 0, i*30, 3*30);
        }
        break;
    case 1:
        for( int x = 0; x < 3; x++ )
        {
            for( int y = 0; y < 3; y++ )
            {
                nImg = y * 3 + x;
                if( m_aImg[nImg] != null )
                    g.drawImage( m_aImg[nImg], x * 30, y * 30,
Graphics.LEFT|Graphics.TOP);
            }
        }
        break;
    case 2:
    default:
        g.drawString( "Anda Berhasil!", 10, 45, 0 );
    }
}

```

```

    }
}
public boolean CheckMove( int nX, int nY )
{
    if( nX < 0 || nX >= 3 || nY < 0 || nY >= 3 )
        return false;
    if(m_nHidX==nX && m_nHidY==nY)
return false;
    if( nX > 0 && nX - 1 == m_nHidX && nY == m_nHidY )
        return true;
    if( nX < 2 && nX + 1 == m_nHidX && nY == m_nHidY )
        return true;
    if( nY > 0 && nY - 1 == m_nHidY && nX == m_nHidX )
        return true;
    if( nY < 2 && nY + 1 == m_nHidY && nX == m_nHidX )
        return true;
    return false;
}
private void sweep(int x, int y)
{
    if( !CheckMove( x, y ) )
        return;
    int temp = m_anCur[y][x];
    m_anCur[y][x] = m_anCur[m_nHidY][m_nHidX];
    m_anCur[m_nHidY][m_nHidX] = temp;
    m_nHidX = x;
    m_nHidY = y;
}

protected void keyPressed(int keyCode)
{
    if( m_nSta == 2 )
        return;
    switch(keyCode)
    {
case KEY_NUM1:
    sweep( 0, 0 );
    break;
case KEY_NUM2:
    sweep( 1, 0 );
    break;
case KEY_NUM3:
    sweep( 2, 0 );
    break;
case KEY_NUM4:
    sweep( 0, 1 );
    break;
case KEY_NUM5:

```

```

        sweep( 1, 1 );
        break;
    case KEY_NUM6:
        sweep( 2, 1 );
        break;
    case KEY_NUM7:
        sweep( 0, 2 );
        break;
    case KEY_NUM8:
        sweep( 1, 2 );
        break;
    case KEY_NUM9:
        sweep( 2, 2 );
        break;
    case KEY_NUM0:
        if( m_nSta == 0 )
            m_nSta = 1;
        else
            m_nSta = 0;
        break;
        }
        if( isFinish() )
            m_nSta = 2;

    }
    public boolean isFinish()
    {
    for( int x = 0; x < 3; x++ )
    {
        for( int y = 0; y < 3; y++ )
        {
            if( m_anCur[y][x] != y * 3 + x )
                return false;
        }
    }
    }
    return true;
}

    public void commandAction(Command c, Displayable s)
    {
        if( c.getCommandType() == Command.EXIT )
        {
            midlet.notifyDestroyed();
        }
        else if (c.getCommandType() == Command.BACK){
            midlet.tampilMenu();
            midlet.cPlayer.stop();
            midlet.cPlayer = null;
        }
    }
}

```

```
public void run() {  
    try{  
        while(true){  
            repaint(0, 0, getWidth(), getHeight());  
        }  
    }  
    catch(Exception e){}  
}  
public void stop()  
{  
}  
}
```



FORMULIR BIMBINGAN SKRIPSI

Nama : Petrus Silalahi
Nim : 06.12.582
Masa Bimbingan : 22 Juni s/d 23 Desember 2011
Judul Skripsi : Rancang Bangun Multiplayer Game Puzzle Gambar Via Bluetooth Berbasis Mobile

No	Tanggal	Uraian	Paraf Pembimbing
1	21/06/2011	Konsultasi Makalah	
2	05/07/2011	Demo Program	
3	09/07/2011	Konsultasi BAB I & BAB II	
4	14/07/2011	Konsultasi Makalah Seminar Hasil	
5	15/07/2011	Revisi Batasan Masalah	
6	16/07/2011	ACC Makalah Seminar Hasil	
7	07/08/2011	Revisi Bab III, IV, dan V	
8	18/09/2011	ACC Laporan BAB I,II,III,IV dan V	
9			
10			

Malang, Agustus 2011

Dosen pembimbing I

Ir. Eko Nurcahyo.
NIP. Y.1028700172



FORMULIR BIMBINGAN SKRIPSI

Nama : Petrus Silalahi
Nim : 06.12.582
Masa Bimbingan : 22 Juni s/d 23 Desember 2011
Judul Skripsi : Rancang bangun multiplayer game puzzle gambar via bluetothh berbasis mobile

No	Tanggal	Uraian	Paraf Pembimbing
1	05/07/2011	Flowcart di Perjelas	
2	08/07/2011	Revisi Penulisan Pada Laporan	
3	09/07/2011	Revisi Pada Program	
4	11/07/2011	ACC Program	
5	16/07/2011	ACC Makalah Seminar Hasil	
6	02/08/2011	Revisi Halaman Laporan Skripsi	
7	03/08/2011	ACC Laporan BAB I,II,III,IV dan V	
8			
9			
10			

Malang, Agustus 2011
Dosen pembimbing II

Michael Ardita ST, MT
NIP. Y.1031000434

Form S-4b



Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa

NAMA

Muhammad Sulalawa

NIM

10611322

Perbaikan meliputi

1. Metode Sampling with penulisan / proses
 pada gambar

2. Proses pada Form Solid

1. penulisan
2. cara keluas

3. proses penulisan gambar

BAB II, III, IV, V

Daftar Isi dan Gambar

Malang,

[Handwritten signature]



WILAYAH PERSEKUTUAN MALANG
JALAN KEMERDEKAAN
KABUPATEN MALANG

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

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

Kampus I : J. Bendungan Sigura-gura No.2 Telp. (0341) 561431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : J. Raya Karanglo, Km.2 Telp. (0341) 417638 Fax. (0341) 417634 Malang

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

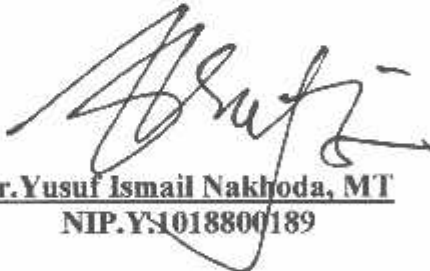
NAMA : PETRUS SILALAH
NIM : 06.12.582
JURUSAN : Teknik Elektro S-1
JUDUL : **RANCANG BANGUN MULTIPLAYER GAME PUZZLE
GAMBAR VIA BLUETOOTH BERBASIS MOBILE**

Dipertahankan dihadapan Tim Penguji Ujian Skripsi Jenjang Program Strata Satu (S-1)

Pada Hari : Kamis
Tanggal : 18 Agustus 2011
Dengan Nilai : 77,20 (B+) *ff*

PANITIA UJIAN SKRIPSI

KETUA



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

SEKRETARIS



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

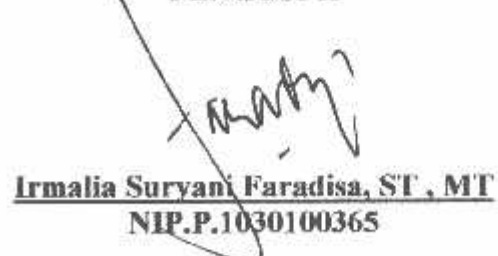
ANGGOTA PENGUJI

PENGUJI I



Sotyo Hadi, ST
NIP.Y.1039700309

PENGUJI II



Irmalia Suryani Faradisa, ST, MT
NIP.P.1030100365



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata 1 Jurusan Teknik Elektro Konsentrasi Teknik Komputer & Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama : PETRUS SILALAH
Nim : 06.12.582
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Komputer & Informatika S-1
Judul : **RANCANG BANGUN MULTIPLAYER GAME PUZZLE
GAMBAR VIA BLUETOOTH BERBASIS MOBILE**

Tanggal	Uraian	Paraf
Penguji I 18 Agustus 2011		
Penguji II 18 Agustus 2011	1. Metode SWAP untuk penentuan gambar 2. Proses penentuan score 3. Proses pemotongan gambar, dijelaskan secara teknis	

Disetujui :

Dosen Penguji I

Sotyohadi, ST
NIP.Y.1039700309

Dosen Penguji II

Irmalia Suryani Faradisa, ST, MT
NIP.P.1030100365

Mengetahui :

Dosen Pembimbing I

Ir. Eko Nurcahyo
NIP. Y.1028700172

Dosen Pembimbing II

Michael Ardita, ST, MT
NIP.Y.1031000434

Teriring Ucapan Terima Kasih Kepada :

**JESUS MY GOD, My father in heaven,
My mom Kristina br Ginting,**

**Abang-abang_Q dan kaka_Q tercinta yang slalu
memberiku semangat.....**

**Wanita yang _Q sayang ' Baiq Tresnawati Lativha' yang
slalu mendukung-_Q.....**

**Semua Teman-_Q kalian adalah Jospirasi bagi-_Q untuk
terus maju.....**

Scooter pinky-_Q, tanpamu aku tidak berangkat kekampus,

I very love you.....

Dosen Pembimbing-ku

'Bpk. Ir. Eko Murchyo.'

dan

'Bpk. Michael Ardita, ST.MT.'

Jasa bapak tidak akan terlupakan bagi_Q.....

