

teknologi Java dapat digunakan untuk menjalankan aplikasi-aplikasi yang dibuat dengan Java, seperti aplikasi *game*.

1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan di atas, maka dalam tugas akhir ini masalah yang dirumuskan adalah membuat sebuah aplikasi *game* Serangan Alien yang dapat digunakan oleh para pengguna perangkat telepon selular yang telah mendukung teknologi Java.

1.3 Batasan Masalah

Pembatasan masalah pada penulisan tugas akhir ini yaitu :

1. Mode grafis yang digunakan adalah mode grafis dua dimensi.
2. Program aplikasi *game* ini hanya membutuhkan satu pemain tunggal dan tidak mendukung pemain dalam jumlah banyak.
3. Program aplikasi *game* ini tidak mendukung permainan dengan menggunakan jaringan.

1.4 Tujuan Penelitian

Adapun tujuan dari penulisan tugas akhir ini adalah merancang dan membangun aplikasi *game* Serangan Alien pada perangkat telepon selular dengan menggunakan J2ME.

1.5 Manfaat Penelitian

Dengan adanya penelitian yang dilakukan ini diharapkan dapat memberi manfaat, yaitu membantu para pengguna perangkat telepon selular untuk mengisi waktu luangnya dengan bermain aplikasi *game* ini.

1.6 Metodologi Penelitian

1.6.1 Metode Pengumpulan Data

Pada penelitian ini, metode pengumpulan data yang digunakan adalah:

a. Metode Kepustakaan (*Library Research*)

Metode Kepustakaan (*Library Research*) dengan mempelajari berbagai literatur baik berupa buku-buku referensi, skripsi, dan artikel-artikel dari situs internet.

b. Telaah Dokumen

Mempelajari dokumen, artikel, dan catatan lain yang masih berkaitan

Working/Heap Memory merupakan area dimana informasi *game* disimpan pada saat eksekusi *game* sedang berlangsung dan akan dilepaskan ketika *game* diakhiri.

b. Memori penyimpanan

Memori penyimpanan atau yang lebih dikenal dengan *Record Management System (RMS)* digunakan untuk menyimpan data RMS ke dalam basis data lokal.

c. Memori aplikasi

Memori aplikasi merupakan tempat dimana perangkat menyimpan *game* atau aplikasi-aplikasi lainnya.

2. Fragmentasi

Fragmentasi terjadi karena banyak lubang memori yang kecil tetapi ketika dimasukkan objek yang besar, objek tersebut tidak bisa masuk karena tidak tersedianya ruang memori yang besar.

3. Ukuran *screen*

Resolusi *screen* merupakan salah satu masalah dalam pengembangan *game* pada perangkat *mobile*, karena adanya perbedaan ukuran *screen*.

4. Antarmuka telepon

Perubahan bentuk telepon tradisional menjadi bentuk yang *user-friendly*.

Pengembangan *game* untuk perangkat *mobile* berbeda dari pengembangan *game* yang konvensional dalam sejumlah hal (<http://www.forum.nokia.com>), antara lain:

1. Ukuran tim

Ukuran tim yang kecil, biasanya terdiri dari 3-5 orang.

2. Anggaran

Anggaran yang dikeluarkan jauh lebih murah.

3. Siklus pengembangan

Game konvensional biasanya dikerjakan selama 2-3 tahun, sedangkan *mobile game* dapat dilakukan dengan tim dan anggaran yang kecil.

4. Perangkat Jaringan

dihasilkan oleh pemain *game*. Nilai yang dihasilkan dalam satu permainan akan diproses. Jika nilai yang dihasilkan bernilai negatif, maka tidak akan dimasukkan ke dalam *Record Store* Terbaik. Jika nilai yang dihasilkan bernilai lebih dari atau sama dengan 0, maka akan diproses dalam *Record Store*. Pada awalnya RMS akan mengumpulkan dan menyimpan 5 buah nilai yang dihasilkan dari 5 kali permainan. Untuk yang berikutnya RMS akan membandingkan nilai yang dihasilkan oleh pemain dengan nilai yang tersimpan dalam *Record Store*. Jika nilai yang dihasilkan oleh pemain lebih besar daripada nilai yang tersimpan dalam *Record Store*, maka nilai tersebut dimasukkan ke dalam *Record Store*. Akan tetapi jika lebih kecil daripada nilai yang tersimpan dalam *Record Store*, maka tidak akan dimasukkan ke dalam *Record Store*. RMS akan mengurutkan nilai yang tersimpan di dalam *Record Store* menurut nilai yang terbesar lalu diikuti oleh nilai- nilai berikutnya yang lebih kecil.

3.3.3 Rancangan Menu

Gambar 3.3 di bawah ini merupakan rancangan menu pada *game* Serangan Alien yang menunjukkan menu-menu dan hubungan antar menu yang terdapat *game* Serangan Alien.

yang terdiri dari satu *frame* tunggal dan *sprite* animasi. *Sprite* mobil (pemain) merupakan *sprite* biasa yang berukuran 18 x 9 piksel. Gambar 3.12 di bawah ini merupakan rancangan *sprite* mobil.



Gambar 3.12 Rancangan *sprite* mobil

Sprite animasi yang digunakan dalam *game* ini terdiri dari beberapa *set frame* yang berasal dari satu gambar. Masing-masing *frame* mempunyai lebar dan tinggi yang sama. *Sprite* alien merupakan *sprite* yang berukuran 20 x 21 piksel sehingga dengan ukuran gambar alien sebesar 100 x 21 piksel, maka pada *sprite* alien terdapat 5 buah *set frame*. Gambar 3.13 di bawah ini merupakan rancangan *sprite* alien.



Gambar 3.13 Rancangan *sprite* alien

Sprite UFO berukuran 63 x 21 piksel. Sedangkan *sprite* monster berukuran 63 x 11 piksel. Gambar 3.14 di bawah ini merupakan rancangan *sprite* UFO dan *sprite* monster.



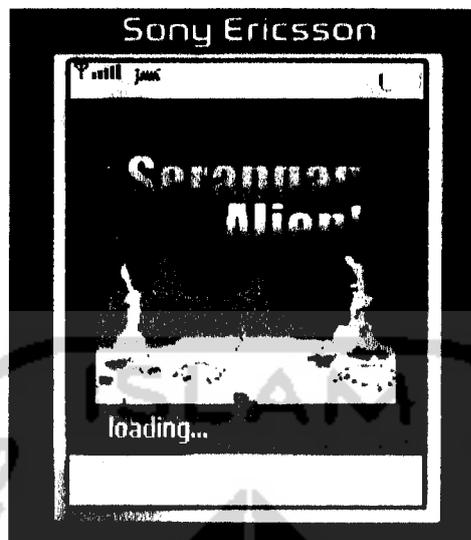
Gambar 3.14 Rancangan *sprite* UFO dan *sprite* monster

Sprite ledakan berukuran 84 x 21 piksel. Sedangkan *sprite* peluru berukuran 44 x 11 piksel. Gambar 3.15 di bawah ini merupakan rancangan *sprite* ledakan dan *sprite* peluru.



Gambar 3.15 Rancangan *sprite* ledakan dan *sprite* peluru

Medan tempur yang akan digunakan dalam aplikasi *game* Serangan Alien terdiri dari 1 gambar tunggal. Gambar tersebut berukuran 180 x 177 piksel. Gambar 3.16 di bawah ini merupakan rancangan medan tempur.



Gambar 3.17 Halaman layar *splash*

Kelas *SeranganAlien* yang merupakan kelas *MIDlet* akan membuat objek kelas *LayarSplash* agar bisa menampilkan layar tersebut. Gambar 3.18 di bawah ini merupakan gambar instantiasi kelas *LayarSplash* pada kelas *SeranganAlien*.

```
public void startApp() {
    .....
    new LayarSplash(display, new MenuUtama(this));
    .....
}
```

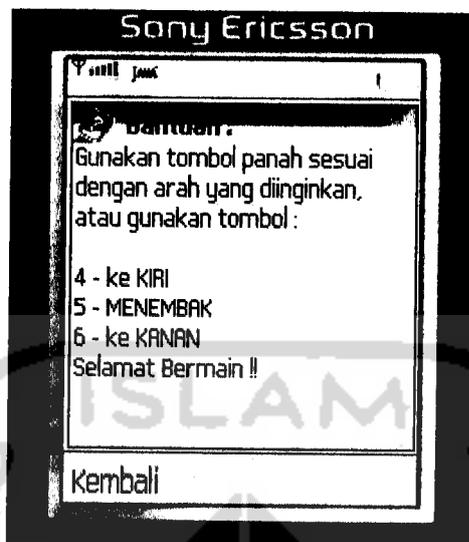
Gambar 3.18 Cuplikan program instans *splash*

Layar *splash* dapat saja diabaikan dengan cara melakukan penekanan sembarang tombol pada telepon selular agar langsung menuju ke halaman utama. Gambar 3.19 di bawah ini merupakan cuplikan program agar layar *splash* dapat langsung menuju ke halaman menu utama melalui penekanan sembarang tombol pada telepon selular.

```
private void lanjut(){
    timer.cancel();
    display.setCurrent(next);
}
```

Gambar 3.19 Cuplikan program loncat *splash*

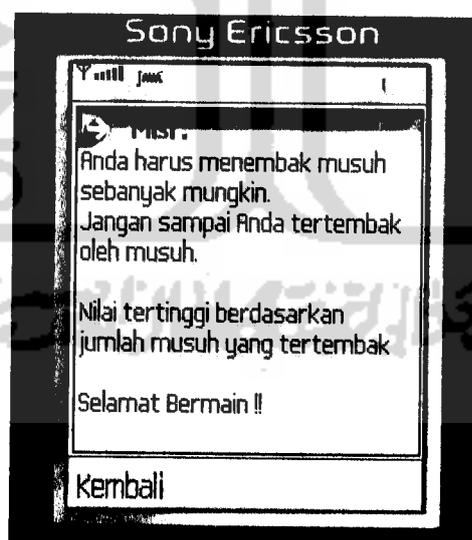
Metode *lanjut()* pada gambar 3.19 di atas digunakan untuk meloncat menuju halaman berikutnya yaitu halaman menu utama. Jika pengguna melakukan penekanan sembarang tombol maka layar *splash* akan dilewatkan



Gambar 3.22 Menu Bantuan

3.4.5 Implementasi Halaman Menu Misi

Menu "Misi" terdiri dari 1 buah pilihan yaitu pilihan Kembali. Jika pilihan tersebut dipilih maka aplikasi akan kembali ke menu utama. Gambar 3.23 di bawah ini merupakan implementasi halaman menu misi.



Gambar 3.23 Menu Misi

3.4.6 Implementasi Halaman Menu Option

Menu "Option" terdiri dari 2 buah pilihan, yaitu pilihan Hilangkan musik dan Hilangkan getar. Jika pilihan Hilangkan musik diberi tanda centang, maka aplikasi permainan tidak menggunakan musik. Jika pilihan Hilangkan getar diberi tanda centang, maka aplikasi permainan tidak menggunakan efek getar. Jika



Gambar 3.25 Mode permainan

Mobil dikendalikan oleh pemain *game* Serangan Alien. Pasukan lawan merupakan sejumlah pasukan musuh yang harus dihancurkan. Nilai menyatakan nilai yang dimiliki oleh mobil (pemain). Nilai mobil akan bertambah 10 angka setiap menembak 1 alien, bertambah 15 angka setiap menembak UFO, dan bertambah 20 angka setiap menembak monster. Nyawa merupakan nyawa mobil yang tersisa. Pada awal permainan, mobil akan diberi nyawa sebanyak 3 nyawa. Nyawa mobil akan berkurang sebanyak 1 nyawa apabila tertembak musuh. Pada awal mode permainan posisi mobil berada di sebelah kiri layar. Posisi pasukan musuh telah ditempatkan secara acak.

3.4.8 Medan tempur

Medan tempur merupakan daerah yang dijadikan arena pertempuran. Medan tempur dibuat dengan menggunakan 1 buah gambar yang berukuran 180 x 177 piksel. Gambar 3.26 di bawah ini merupakan gambar medan tempur yang digunakan dalam permainan.

dapat dihidupkan atau dimatikan dengan melakukan perubahan pada kelas Option. Gambar 3.32 di bawah ini merupakan metode *start()* untuk memainkan sebuah musik yang terdapat pada kelas Layar.

```
public void start()
{
    try {
        mediaPlayer.start();
    }
    catch(MediaException mediaexception) { }
}
```

Gambar 3.32 Metode *start()*

Metode *start()* di atas hanya akan dipanggil jika menginginkan mendengarkan musik dengan tidak mencentang pilihan Hilangkan musik pada submenu "Option". Metode *start()* di atas akan mencari *path* musik yang berjenis midi. Jika ditemukan maka musik akan dibuat dan dimainkan secara berulang-ulang. Gambar 3.33 di bawah ini merupakan metode *stop()* yang bisa menghentikan musik yang berulang-ulang tadi.

```
public void stop()
{
    try {
        mediaPlayer.stop();
    }
    catch(MediaException mediaexception) { }
}
```

Gambar 3.33 Metode *stop()*

3.4.13 Game Over

Mode *game over* merupakan sebuah mode dimana permainan berada dalam tahap akhir. Pada tahap ini semua jenis suara akan dinonaktifkan dan akan ditampilkan sebuah gambar baru yang berada di atas medan tempur. *Game over* akan terjadi jika nyawa mobil (pemain) telah habis. Pada layar *game over* akan ditampilkan perintah "Mulai" yang akan membawa pemain ke permainan baru dan "Keluar" yang akan membawa pengguna aplikasi ke menu utama. Musik akan dimatikan pada mode ini. Pada *game over* ini juga akan ditampilkan daftar nilai terbaik. Nilai yang ditampilkan hanya 5 buah nilai terbaik. Jika nilai yang dihasilkan lebih kecil daripada 5 nilai terbaik atau nilai kurang dari nol, maka nilai

4.2.3 Pengujian oleh pengguna

Dari kuisisioner tersebut kemudian dirangkum dan dikonversikan ke dalam beberapa nilai untuk memudahkan penghitungan hasil analisisnya, yaitu :

Nilai 1 untuk jawaban sangat kurang.

Nilai 2 untuk jawaban kurang.

Nilai 3 untuk jawaban sedang.

Nilai 4 untuk jawaban baik.

Nilai 5 untuk jawaban sangat baik.

Dari nilai tersebut dapat digunakan untuk menghitung nilai rata-rata dari jawaban pengguna. Rumus untuk menghitung nilai rata-ratanya adalah :

$$\text{Rata-rata} : \frac{\sum \text{nilai jawaban (jumlah nilai jawaban)}}{\sum \text{jumlah pengguna (jumlah pengguna)}}$$

Tabel 4.1 di bawah ini merupakan hasil pengujian aplikasi *game* Serangan Alien yang dilakukan pada 10 orang.

Tabel 4.1 Hasil pengujian aplikasi *game* Serangan Alien

Pertanyaan	Jawaban responden										Rata-rata
	A	B	C	D	E	F	G	H	I	J	
Tampilan Antarmuka	4	5	4	4	4	3	4	4	4	4	4,0
Tingkat Kesulitan	3	2	2	3	3	3	2	2	2	3	2,5
Menghibur Tidaknya	5	4	5	3	3	4	4	4	5	4	4,1

Dari tabel pengujian terhadap 10 orang di atas dapat disimpulkan bahwa aplikasi *game* Serangan Alien mempunyai tampilan antarmuka yang baik, memiliki tingkat kesulitan yang kurang, dan menghibur.

- [SET06] Setianto, Dwi. *Panduan Lengkap Pemrograman Celluler Device dengan J2ME*. Yogyakarta: Ardana Media, 2006.
- [SHA06] Shalahuddin, M. & Rosa A. S. *Pemrograman J2ME-Belajar Cepat Pemrograman Perangkat Telekomunikasi Mobile*. Bandung: Informatika, 2006.
- [SIN05] Sinaga, Benjamin, L. *Pemrograman Berorientasi Objek dengan Java*. Yogyakarta: Gava Media, 2005.
- [SUY05] Suyoto, Dr. *Membuat Sendiri Aplikasi Ponsel*. Yogyakarta: Gava Media, 2005.
- [TOP02] Topley, K. *J2ME in a Nutshell: A Desktop Quick Reference*. California: O'Reilly, 2002.
- [WEL04] Wells, Martin, J. *J2ME Game Programming*. Boston: Premier Press, 2004.

