

MEMBANGUN APLIKASI GAME FN-BALL PADA PONSEL MENGUNAKAN TEKNOLOGI *JAVA 2 MICRO EDITION* (*J2ME*) *MIDP 2.0*

Wahyu Pujiyono¹⁾, Fajar Silalahi

Program Studi Teknik Informatika

Fakultas Teknologi Industri Universitas Ahmad Dahlan Yogyakarta

Email¹⁾ : yywahyup@yahoo.com

ABSTRAK

Pengembangan game pada ponsel memerlukan berbagai pertimbangan menyangkut keterbatasan kinerja prosesor, memori, layar dan keypad. Permainan (game) yang banyak dimainkan di antaranya adalah yang ber-genre paddle games. Makalah ini akan membahas teknik pengembangan aplikasi game FN-Ball pada ponsel. Tahapan pengembangan meliputi analisis dan pemodelan kebutuhan, perancangan sistem dan implementasinya. Dari hasil pengujian dapat disimpulkan bahwa aplikasi game FN-Ball yang dibangun dapat menjadi salah satu alternatif pilihan game yang menghibur dan sekaligus dapat menguji kecepatan dan ketepatan.

Kata kunci : *Game, Ponsel, Paddle, FN-Ball.*

1. PENDAHULUAN

Generasi pertama *game console* adalah Nintendo milik perusahaan Nintendo dan Sega milik perusahaan Sega. Keduanya merupakan *game console* yang ada (sebelum munculnya Play Station, Wii, Game Cube dan Xbox) yang memiliki kemampuan yang berbeda-beda dari sudut gambar misalnya Nintendo menggunakan teknologi 8 bit dan Sega menggunakan teknologi 16 bit, tetapi dari sudut media penyimpanan *game* keduanya sama-sama menggunakan *cartridge* (disebut *gamer* sebagai kaset) dan tergolong mahal untuk mendapatkannya. Dengan teknologi gambar yang dimiliki sangat minimum, maka gambar yang ditampilkan oleh kedua *game console* ini tidak terlihat natural dan cenderung patah-patah serta suara yang masih tergolong primitif. Karena kendala-kendala tersebut maka kedua *game console* ini sudah jarang kita jumpai dan telah lama ditinggalkan para *gamer*. [1]

Generasi terbaru dari *game console* adalah Play Station milik perusahaan Sony yang diluncurkan pada tahun 1999 dan Xbox. Milik perusahaan Microsoft pada tahun 2001. *Game console* ini mengadopsi teknologi 32 bit untuk gambar, untuk *processor* Play Station menggunakan *processor* Cell dari IBM berkecepatan 3,2 *Gigahertz* dengan *processor grafis* RSX dan chip didukung dari Nvidia, khusus untuk Play Station 3. Untuk media penyimpanan *game* Play Station menggunakan CD, DVD, dan *blue-ray disc* (untuk Play Station 3 yang mampu menyimpan data 50 *Gigabyte*). Sedangkan sedangkan Xbox berjalan menggunakan *procesor* IBM Power PC 3,2 *Gigahertz* dengan *processor grafis* ATI Radeon 500 *Megahertz* dan *Random Access Memory* (RAM) 512. *Megabyte*. selain DVD, juga dilengkapi dengan *hardisk* 20 *Gigabyte* untuk penyimpanan *game*. Selain itu keunggulan lain yang dimiliki kedua *game console* ini adalah kemampuan untuk

bermain *game* bersama para *gamer* di seluruh dunia karena menggunakan internet berbasis jalur pita lebar (*broadband*).[2]

Perkembangan teknologi saat ini, menjadikan *game* tidak lagi hanya dijalankan atau dimainkan pada *platform console* maupun komputer, melainkan juga dapat dijalankan pada perangkat *mobile*. Contoh dari perangkat *mobile* antaralain :

1. **Ponsel/handphone** merupakan perangkat *mobile* yang fungsi utamanya adalah berkomunikasi suara sambil bergerak selama masih berada pada jangkauan jaringan seluler. Komunikasi juga bisa dilakukan melalui pesan teks atau SMS yang memiliki tarif yang lebih murah. Ponsel generasi baru lazimnya sudah berteknologi GPRS, yang mampu mentransfer data sampai kecepatan 112 kbps. Contohnya : Nokia 3310, Nokia 3315, dll.[3]
2. **SmartPhone** merupakan perangkat *mobile* yang pada dasarnya adalah alat komunikasi atau ponsel, tapi para produsen mengembangkannya sehingga mempunyai kemampuan yang lebih seperti menjalankan aplikasi bisnis, *multitasking*, dan lain sebagainya. Contohnya : Samsung i500, Samsung i600, Samsung i700, dll.[4]
3. **Notebook** pada prinsipnya memiliki fungsi yang sama dengan *desktop* PC. Perbedaan yang mendasar antara keduanya adalah pada segi bentuk dan ukuran dari perangkat keras yang dipakai. Untuk *Notebook*, segala peripheral yang ada telah menyatu menjadi satu kesatuan sehingga para pemakai dapat membawa kemana saja mereka pergi didukung pula dengan ukurannya yang tidak terlalu besar. Sedangkan untuk sumber daya pada umumnya digunakan baterai. Contohnya : ECS G320+, Toshiba Cosmo M500, Axioo NBV 728NC, dll.[3]
4. **Pocket PC/PDA** dikenal sebagai komputer saku yang memiliki banyak kegunaan. Sebut saja untuk melakukan kalkulasi, menerima *e-mail*, merekam video, membuat dan menulis *spreadsheets*, menggunakannya sebagai radio atau *stereo*, hingga menfunksikannya sebagai *Global positioning system* (GPS). Seiring perkembangannya, PDA sekarang ini dikenal juga dengan *PDA phone* yang pada dasarnya adalah PDA tetapi dilengkapi dengan fasilitas telekomunikasi.[4]

Perbedaan yang mendasar dari keempat perangkat *mobile* di atas terletak pada *processor* yang dipakai, bentuk dan ukuran, *pixel* untuk gambar, media penyimpanan dan daya tahan baterai. *Processor*, *pixel* untuk gambar dan media adalah faktor yang mempengaruhi jenis *game* apa saja yang dapat dimainkan pada masing-masing perangkat *mobile* tersebut.

Beberapa tahun belakangan ini, *game* yang dimainkan pada ponsel mendapatkan prioritas yang sangat tinggi. Ini diakibatkan penjualan ponsel melesat jauh dibanding dengan penjualan perangkat *mobile* lainnya. Di samping harganya terjangkau oleh berbagai lapisan masyarakat, perangkat tersebut juga mudah didapat.

2. METODE PENELITIAN

2.1 Analisis dan Pemodelan Kebutuhan Sistem

Analisis merupakan langkah yang dilakukan sebelum melakukan perancangan sistem. Dari analisis ini akan didapatkan gambaran dasar mengenai sistem secara keseluruhan dan gambaran mengenai kebutuhan sistem.

Dalam pengembangan sistem berorientasi objek, pemodelan kebutuhan dilakukan dengan menggunakan *use case diagram*.

2.2 Perancangan Sistem

Setelah melakukan analisis terhadap sistem yang akan dibangun, maka langkah selanjutnya adalah melakukan perancangan sistem. Perancangan sistem berorientasi objek dilakukan dengan membuat *class diagram* dan *sequence diagram*.

2.3 Perancangan Antarmuka (*Interface Design*)

Perancangan antarmuka ini meliputi perancangan *splash screen*, *menu screen*, objek, dan *level-level*.

2.4 Implementasi Sistem

Setelah tahap perancangan sistem, maka tahap selanjutnya yang dilakukan adalah mengimplementasikan rancangan tersebut menjadi sebuah *game* FN-Ball yang akan dibangun menggunakan NetBeans 5.0 Beta 2 with Mobility Pack, Sun Java Wireless Toolkit (versi 1.0.3 dan 2.3 Beta) dan Java 2 SDK versi 1.4.2.

3. HASIL DAN PEMBAHASAN

3.1 Skenario game FN-Ball

Pertama-tama, *player* menekan sebuah tombol untuk menjalankan *game*. Kemudian akan ditampilkan *splashscreen* yang merupakan intro dari *game*. Setelah itu, *player* masuk pada tampilan menu utama untuk memilih menu yang telah ada yaitu mulai permainan, nilai tertinggi, bantuan, informasi dan dan keluar.

Menu **mulai main** berfungsi untuk memulai permainan dari *level* awal. Menu **nilai tertinggi** berfungsi untuk menampilkan *score/nilai* yang tertinggi pada permainan. Menu **bantuan** berfungsi untuk menampilkan informasi cara bermain kepada *player*. Menu **informasi** berfungsi untuk menampilkan informasi singkat tentang pengembang *game*. Menu **keluar** berfungsi untuk keluar dari permainan.

Pada tiap-tiap *level*, terdiri dari *box*, *paddle* dan *ball*. Tujuan dari permainan ini adalah *player* hanya diperintahkan untuk menjaga bola selalu di udara menggunakan pemantul dan menghancurkan kotak-kotak yang ada menggunakan bola. Ketika memulai permainan dari awal, *player* diberi kesempatan sebanyak tiga kali untuk menjaga bola agar tidak jatuh atau bola tidak dapat mengenai pemantul, kesempatan ini pada *frame* permainan dapat dilihat di kanan atas dari *frame* yang ditulis juga dengan *ball*. Jika kesempatan yang diberikan habis, maka permainan akan berakhir.

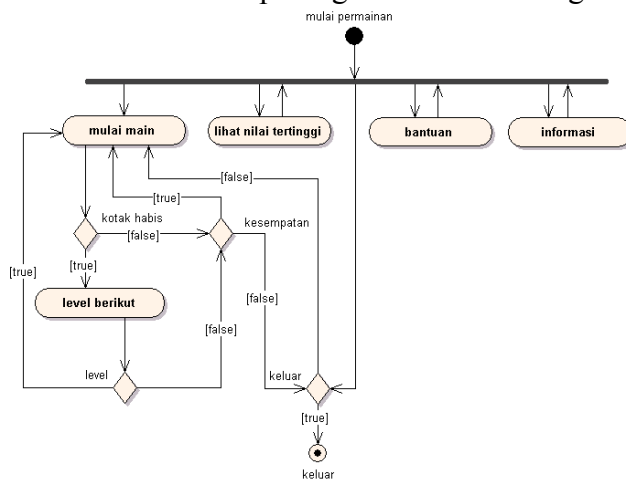
3.2 ANALISIS DAN PEMODELAN KEBUTUHAN SISTEM

1. Aktivitas Bermain *Game* FN-Ball

Berdasarkan skenario di atas, berikut ini adalah aktivitas yang dilakukan oleh *player* dalam bermain *game* FN-Ball :

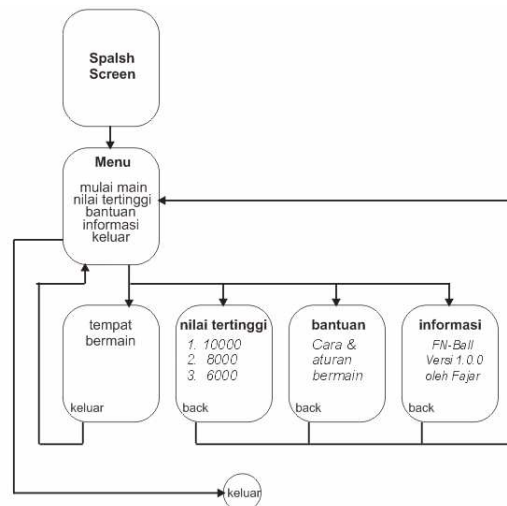
- a. *Player* memulai *game* FN-Ball.
- b. Kemudian *player* masuk pada tampilan menu utama untuk memilih menu yang sudah ada.

- c. Jika *player* memilih menu mulai permainan, *player* bisa memulai permainan dari *level* awal.
 - d. Di dalam *frame* permainan terdapat tiga menu lagi yaitu menu *exit*, *pause*, *resume*.
 - e. Menu *exit* terletak pada kiri bawah *frame* permainan yang berfungsi untuk keluar dari permainan dan kembali ke menu pemilihan.
 - f. Menu *pause* terletak pada kanan bawah *frame* permainan yang berfungsi untuk menghentikan permainan sementara.
 - g. Menu *resume* terletak pada kanan bawah *frame* permainan yang berfungsi untuk melanjutkan permainan yang terhenti ketika *player* memilih menu *pause* dan menu *resume* akan muncul apabila *player* terlebih dahulu memilih menu *pause*.
 - h. Jika semua *box* di dalam *frame* dapat dihancurkan menggunakan bola, maka permainan pada *level* tersebut berakhir.
 - i. *Player* bisa melihat *score*/nilai permainan pada kiri atas *frame* permainan.
 - j. *Player* bisa melihat jumlah kesempatan/*ball* pada kanan atas *frame* permainan.
 - k. Kemudian *player* bermain pada *level* selanjutnya.
 - l. Jika *player* memilih menu nilai tertinggi, *player* dapat melihat *score*/nilai tertinggi dalam permainan.
 - m. Jika *player* memilih menu bantuan, *player* dapat melihat informasi tentang cara memainkan *gamenya*.
 - n. Jika *player* memilih menu informasi, *player* dapat melihat informasi singkat tentang pengembang *game*.
 - o. Jika *player* memilih menu keluar, *player* keluar dari permainan.
- Aktivitas *game* FN-Ball di atas dapat digambarkan dalam gambar 1 berikut ini :



Gambar 1. Activity diagram game FN-Ball

Selain itu, kita juga dapat menggambarkan *diagram* menu dan submenunya dalam gambar 2 berikut ini :



Gambar 2. Main menu with sub menu flow diagram

Berdasarkan aktivitas *player* dalam bermain *game* FN-Ball di atas, diperoleh kebutuhan yang harus dipenuhi oleh sistem yang disajikan tabel 1 di bawah ini:

Tabel 1. Tabel kebutuhan *user* (*user requirement*)

No	Requirement	Aktor	Use Case
1	Player memulai <i>game</i> FN-Ball	Player	Mulai permainan
2	Player bisa melihat informasi nilai tertinggi dari permainan	Player	Lihat nilai tertinggi
3	Player bisa melihat informasi singkat cara bermain	Player	Lihat Bantuan
4	Player bisa melihat informasi singkat tentang pengembang	Player	Lihat informasi

3.3 Implementasi Sistem

1) FN-BallMidlet Class

FN-BallMidlet merupakan kelas utama (*MIDlet*) yang berfungsi untuk berhubungan langsung dengan AMS dari perangkat *handheld*. Kelas ini menginstansiasi kelas *SplashScreen*, *MenuScreen*, *High*, *Nilai tertinggi*, *LayoutCanvas*, *Help* dan *About*. Implementasi class ini adalah sebagai berikut :

```

public class FNBallMidlet extends MIDlet{
    private static Display display;
    private MenuScreen menuScreen;
    private High high;
    private Score sc;
    private LayoutCanvas gamenya;
    private About about;
    private Help help;

    private static
    final String scoreRMSName
    = "IronBalzScore";
    
```

Listing 1. Implementasi FN-BallMidlet Class

2) *SplashScreen class*

SplashScreen merupakan kelas yang digunakan untuk menampilkan *splashscreen* atau intro permainan. Tampilan Splash Screen diperlihatkan dalam gambar 3 di bawah ini.



Gambar 3. Splash Screen

Listing 2 di bawah ini adalah implementasi dari Splash Screen di atas.

```
public SplashScreen(Display display, Displayable next) {
    this.display = display;
    this.next = next;
    if(cov == null){
        try{
            cov = Image.createImage(cover);
            fH = cov.getHeight();
            fW = cov.getHeight();
        }catch(Exception ioe){
            return;
        }
    }
    maxSizeW = getWidth()/2;
    maxSizeH = getHeight()/2;
    leftRect = maxSizeW/10;
    upRect = getHeight()/10;
    display.setCurrent(this);
}
```

Listing 2. Implementasi *SplashScreen class*

MenuScreen merupakan Kelas yang digunakan untuk menampilkan menu yang dapat dipilih oleh *player*. Menu pilihan ini terdiri dari main baru, nilai tertinggi, bantuan, informasi, dan keluar. Menu Screen diperlihatkan dalam gambar 4 di bawah ini.



Gambar 4. Menu permainan

Sistem menu Permainan disimpan dalam array mainMenu1 dalam listing 3 di bawah ini :

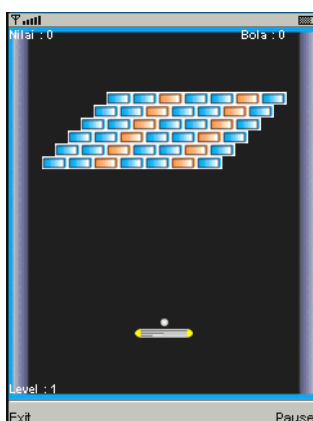
```
static final String[] mainMenu1 =  
{"mulai_main", "nilai_tertinggi", "bantuan", "informasi", "keluar"};
```

Listing 3. Implementasi mainMenu1 dalam array

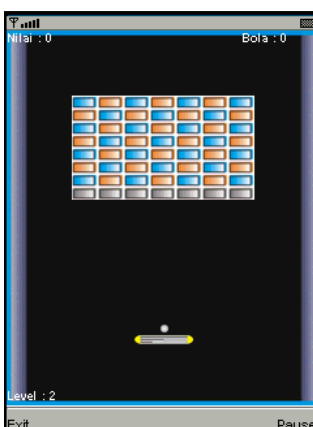
3) *LayoutCanvas Class*

LayoutCanvas merupakan kelas yang mengumpulkan semua objek yang diinginkan dalam satu tempat, yaitu *Canvas*. Kelas ini menangani pembacaan *level* permainan dan menterjemahkan objek-objek, yaitu *box*, *paddle*, *ball*. Kelas ini juga menangani aksi-aksi yang berkaitan dengan pergerakan dari objek *paddle* dan *ball* serta benturan-benturan yang terjadi apabila objek *ball* mengenai *paddle*, objek *ball* mengenai *box*, objek *ball* mengenai dinding pembatas, dan ketika *paddle* mengenai dinding pembatas.

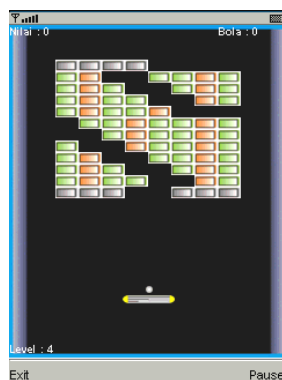
Gambar 5, 6 dan 7 di bawah ini, tampilan berbagai level yang dibuat dalam Game FN-Ball.



Gambar 5. Tampilan Level 1



Gambar 6. Tampilan Level 2



Gambar 7. Tampilan Level 4

Listing 4 berikut ini merupakan implementasi *LayoutCanvas Class* yang digunakan untuk berbagai tampilan level permainan.

```
public LayoutCanvas(IronBallz midlet,Score sc)
throws IOException{
    super(true);
    this.midlet = midlet;
    this.sc = sc;
    isPause = false;
    end = false;
    isWin = false;
    isBegin = false;
    isGameOver = false;
    widthScreen = getWidth();
    heightScreen = getHeight();
    divCurrentX = 2;

    divCurrentY = 3;
    initGame();
}
```

Listing 4. Implementasi *LavoutCanvas Class*

4) *SpriteKotak Class*

SpriteKotak merupakan *class* yang digunakan untuk menciptakan objek kotak yang ada pada *game FN-Ball*. Diperlukan 4 frame untuk membuat kotak yaitu sisi kiri, kanan, atas dan bawah. Implementasi dari tiap bagian kotak diperlihatkan pada program di bawah ini :

```
public Sprite leftFrame(Image image,int x,int y){
    Sprite left = new Sprite (image,1,10);
    left.setFrame(0);
    left.setPosition(x,y);
    return left ;
}
public Sprite rightFrame(Image image,int x,int y){
    Sprite right = new Sprite(image,1,10);
    right.setFrame(19);
    right.setPosition(x,y);
    return right;
}
public Sprite upFrame(Image image,int x,int y){
    Sprite up = new Sprite(image,20,1);
    up.setFrame(0);
    up.setPosition(x,y);
    return up;
}
public Sprite downFrane(Image image,int x,int y){
    Sprite down = new Sprite (image,20,1);
    down.setFrame(9);
}
```

Listing 5. Implementasi *SpriteKotak Class*

5) *Sliding Class*

Sliding merupakan kelas yang berfungsi untuk menciptakan objek *sliding* sebagai alat untuk pemantul bola dalam *game FN-Ball* ini. Objek didahului dengan posisi awal *startPosition*, kemudian bola di gambar dengan *Draw*. Bola dapat digeser ke kiri dengan *moveLeft* dan digeser ke kanan dengan *moveRight*. Untuk mengidentifikasi letak bola menggunakan *getXY*. Implementasinya disajikan dalam listing program di bawah ini :

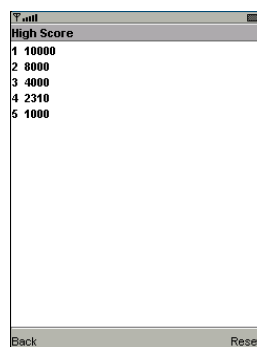
```

public void startPosition(){
    setPosition((scnWidth/2)-22,(8*scnHeight/10)+10);
}
public void draw(Graphics g){
    getXY();
    this.setPosition(x,y);
    super.paint(g);
    nextFrame();
}
public void moveLeft() {
    getXY();
    if (x - MOVE > -3)
        move(MOVE * -1,0);
}
public void moveRight() {
    getXY();
    if (x + MOVE + frameWidth < scnWidth+3)
        move(MOVE,0);
}
private void getXY() {
    x = getX();
    y = getY();
}
    
```

Listing 6. Implementasi *Sliding Class*

6) *Score Class*

Score merupakan kelas yang digunakan untuk menyimpan nilai yang disimpan pada *record store*. Gambar 8 berikut menampilkan inisialisasi High Score :



Gambar 8. Tampilan Score/Nilai

Listing 7 di bawah merupakan inisialisasi nilai permulaan permainan.

```

private void initValues() {
    values[0] = 10000;
    values[1] = 8000;
    values[2] = 4000;
    values[3] = 2310;
    values[4] = 1000;
}
    
```

Listing 7. Implementasi inisialisasi *Score*

4 Kesimpulan

Aplikasi permainan (*game*) FN-Ball ber-*genre paddle games* yang menarik menggunakan *Java 2 Micro Edition* dengan MIDP 2.0 dan terdiri dari 4 level permainan yang bisa dijalankan pada komputer dengan menggunakan emulator ponsel dan telah diimplementasikan langsung pada ponsel-ponsel yang telah mendukung teknologi *Java* dengan ukuran layar yang lebih lebar.

Berdasarkan hasil pengujian, terdapat perbedaan ketika menjalankan *game* FN-Ball ini pada emulator ponsel di komputer dengan ponsel/*handphone* dan antara ponsel yang satu dengan yang lainnya, tergantung tipe dan merk dari ponsel itu sendiri. Contohnya adalah dan efek suara yang dihasilkan, yang mana pada ponsel Motorola E680 efek suara dapat didengar tetapi tidak menutup kemungkinan efek suara tidak dapat didengar oleh ponsel yang berbeda merk dan tipe ponsel lain. Hal ini bisa terjadi karena perbedaan *hardware* dari masing-masing merk dan tipe ponsel.

5 Daftar Pustaka

- [1] <http://www.kompas.commetronews/020531003623.htm\003623.htm>
- [2] <http://www.kompas.comteknologinews050518172924.htm\172924.htm>
- [3] <http://www.ebizasia.com/0212-2003/focus,0212,02.htm\Mobile Warrior - eBizzAsia Nopember 2003.mht>
- [4] <http://www.menerimapesanan.wordpress.com200708\2007 August « menerima pesanan.mht>
- [5] Kadir, A., 2004, *Dasar Pemrograman Java 2*, Andi Offset, Yogyakarta.
- [6] Munawar, *Pemodelan Visual dengan UML*, Edisi Pertama, Graha Ilmu, Yogyakarta, 2005.
- [7] Wicaksono, A., *Pemrograman Aplikasi Wireless dengan Java*, PT. Elex Media Komputindo, Jakarta, 2002.
- [8] <http://www.java.sun.com/jdc/technicalArticles/wireless>
- [9] <http://www.java.sun.com/J2ME>
- [10] <http://www.java.sun.com/product/midp/>