

**ANALISIS DAN PERANCANGAN GAME “LABIRINTH BALL” DENGAN
PEMANFAATAN SENSOR ACCELEROMETER
BERBASIS JAVA PADA ANDROID**

NASKAH PUBLIKASI



diajukan oleh:

Rochman Zaelani

10.11.3687

kepada
**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
AMIKOM YOGYAKARTA
YOGYAKARTA
2013**

NASKAH PUBLIKASI

**ANALISIS DAN PERANCANGAN GAME “LABIRINTH BALL”
DENGAN PEMANFAATAN SENSOR ACCELEROMETER
BERBASIS JAVA PADA ANDROID**

disusun oleh

**Rochman Zaelani
10.11.3687**

Dosen Pembimbing



**Kusnawi, S.Kom, M. Eng.
NIK. 190302112**

Tanggal, 07 Desember 2013

**Ketua Jurusan
Teknik Informatika**



**Sudarmawan, MT
NIK. 190302035**

**ANALYSIS DESIGN GAMES LABYRINTH BALL WITH UTILIZATION OF
ACCELEROMETER AND USING ENGINE ANDENGINE
THE JAVA-BASED ON ANDROID**

**ANALISIS DAN PERANCANGAN GAME LABIRINTH BALL MENGGUNAKAN ENGINE
ANDENGINE DENGAN PEMANFAATAN ACELOMETER BERBASIS JAVA
PADA ANDROID**

Rochman Zaelani
Kusnawi
Jurusan Teknik Informatika
STMIK AMIKOM YOGYAKARTA

ABSTRACT

Game is one of the major industries in the world today . The development of the game is so fast with various types , such as action , adventure , role playing game (RPG) , education , strategies , and others . One of them is an example of a maze game simple game aimed at determining the right track to achieve its intended purpose . This maze game where memerlelukan accuracy to find the right path and quickly to achieve the goal . The number of gadgets that are developing now makes many of its applications and games running on the phone .

Various games have been played a lot , from children to adults most of the types of games that give the impression of entertaining and leisure. In addition to the many benefits of playing games that train the neural motor coordination , sensory and reflex . Along the development era and the technology maze games can be played on mobile android based smartphones are equipped with accelerometer sensor gravity as a control .

In the making of this game will be using the Java programming language and the Eclipse IDE with the help of AndEngine library . So that will facilitate the making and this game becomes more interesting to play .

Keywords : *game, android, labirinth, accelerometer*

1. Pendahuluan

1.1. Latar Belakang Masalah

Perkembangan teknologi begitu pesat khususnya didunia IT. Hampir setiap hari tercipta penemuan-penemuan baru, mulai dari *hardware* sampai *software*. Salah satunya dibidang *game*. *Game* untuk saat ini masih rame peminatnya mulai dari anak-anak hingga dewasa. Dimana *game* selain bisa dimanfaatkan untuk melepaskan rasa jenuh, juga dapat membantu merangsang orang untuk berpikir kritis dan mengasah otak. Teknologi *game* sangat cepat perkembangannya, hal ini ditandai dengan semakin banyaknya *platform* dan jenis *game* yang diciptakan. Berbagai bentuk *game* banyak bermunculan, baik yang sederhana maupun yang bersifat kompleks dari segi aturan permainan, tampilan, maupun peralatan pendukungnya.

Selain *game* konsol dan *game* PC, sekarang juga banyak yang mengembangkan *game* untuk perangkat *mobile*. Hal ini dipicu dengan semakin pesatnya perkembangan teknologi *mobile* salah satunya Smartphone dan Tablet PC. Salah satu platform yang banyak digunakan yaitu pada sistem operasi android. Android sendiri merupakan sistem operasi untuk perangkat *mobile* yang berbasis linux. Salah satu faktor para developer *game* memilih android, karena android sendiri merupakan sistem operasi *open source* sehingga bebas dikembangkan dan gratis.

Game sendiri juga memiliki beberapa genre berdasarkan cara bermain. Mulai genre action, adventure, role playing (RPG), simulation, dan strategy. Beberapa aplikasi *game* tidak hanya membutuhkan perhitungan-perhitungan numerik di dalam penyelesaiannya, namun juga memerlukan penalaran-penalaran akan ketidakpastian, selayaknya proses pemecahan masalah yang dilakukan secara alami oleh manusia. Salah satunya *game* labirin merupakan sebuah contoh *game* sederhana yang bertujuan menentukan jalur yang tepat dan tepat untuk mencapai tujuan yang telah ditetapkan. Hal ini yang dijadikan faktor dan dasar pemikiran penulis untuk pembuatan *game* "LABYRINTH BALL" dengan pemanfaatan sensor accelerometer berbasis java pada android. *Game* ini diharapkan bisa menjadi *game* yang menarik dan disukai, dimana bukan hanya untuk media hiburan semata, tetapi dapat membantu merangsang orang untuk berpikir kritis dan mengasah otak.

2. Landasan Teori

2.1. Game

2.1.1 Pengertian Game

Game adalah sesuatu yang bersifat interaktif, memiliki peraturan sendiri dimana mengandung tantangan dan suatu kondisi menang yang mendefinisikan suatu realitas dengan tujuan untuk menghibur.¹

2.1.2 Genre Game

Game memiliki banyak *genre* dalam perkembangannya, diantaranya diklasifikasikan berdasarkan perbedaan cara bermain dari pada visual atau naratif. Dalam pengklasifikasian ini sebuah game dapat memiliki beberapa *genre*. Berikut beberapa *genre* game yang ada untuk saat ini:

a. Action

Action adalah *Genre Game* yang mengharuskan pemain memanfaatkan reflek, keakuratan, dan kecepatan yang baik dalam menyelesaikan suatu misi. Pada *genre* tipe ini *gameplay* cenderung menekankan pada pertempuran. *Genre Action* memiliki beberapa *Sub-genre* seperti *Fighting Game*, *Shooter Game*, *Action Adventure*, dan lain-lain.

b. Adventure

Adventure adalah *Genre Game* yang umumnya mengharuskan pemain untuk menjelajahi suatu tempat di dalam *game* untuk memenuhi suatu objektif. Pada *Genre Game* ini, urutan cerita dan objektif disusun sedemikian rupa sehingga *game* memiliki alur yang menarik.

c. Role-Playing Game (RPG)

Role-Playing Game atau RPG adalah jenis *game* yang membuat seolah-olah pemain adalah karakter yang dimainkannya. Karakter yang dikendalikan bisa lebih dari satu. Inti dari *Genre Game* ini adalah mengendalikan karakter dan mengembangkannya ke arah yang diinginkan..

d. Simulation

Simulation adalah *Genre Game* yang mensimulasikan pengalaman bermain yang sedekat mungkin mirip dengan aslinya, meskipun objek yang disimulasikan tidak benar-benar ada. pengalaman yang mendekati realitas adalah hal yang ingin disampaikan melalui *game* ber-*genre* ini.

e. Strategy

Strategy adalah *Genre Game* mengharuskan pemain menggunakan taktik dan strategi dalam mengelola sumber daya untuk memenangkan suatu pertempuran atau permainan.

¹ <http://www.gamedev.net>, diakses pada : minggu 9 Maret 2013 ; 14.00 WIB

f. Casual

Game kasual dapat dibedakan dengan aturan sederhana mereka dan tidak diperlukan kompleksitas seperti pada game lain. Mereka tidak memerlukan komitmen jangka panjang waktu atau keahlian khusus untuk bermain, dan ada relatif produksi rendah dan biaya distribusi bagi produsen .

g. Education

Game pendidikan adalah permainan yang telah dirancang untuk mengajar orang tentang topik tertentu, memperluas konsep, memperkuat pembangunan, memahami sebuah peristiwa sejarah atau budaya, atau membantu mereka dalam belajar keterampilan karena mereka bermain.

h. Puzzle

Game ini berfokus pada logika dan tantangan konseptual, yang mempunyai variasi dari satu tema Pemain harus menemukan petunjuk untuk memenuhi kondisi menang dimana akan mendapatkan hak untuk melanjutkan ke *level* selanjutnya yang lebih sulit.

i. Other

Seiring berkembangnya *game*, *genre-genre* baru pun bermunculan seperti *music*, *Party*, *sosial game*. Dan yang baru-baru muncul adalah *Sosial game* yang berkembang sejalan dengan *sosial networking*.

3. Analisis dan Perancangan Sistem

3.1. Analisis

Pada tahapan analisis menentukan apakah semua kebutuhan dalam pengembangan game tersebut telah memenuhi kebutuhan baik dari segi fungsi dan fitur serta apakah *hardware* dan *software* cukup untuk mengembangkan game tersebut. Pada kebutuhan fitur, Selain fitur-fitur standar yang ada di game jenis ini. Fitur-fitur tambahan harus diberikan agar *game* menjadi lebih menarik.

3.1.1 Analisis SWOT

Tabel 3.1. Analisis SWOT

<p style="text-align: center;">Internal</p> <p style="text-align: center;">Eksternal</p>	<p>Strengt</p> <ul style="list-style-type: none"> a. Sistem kendali menggunakan accelerometer. b. Membantu merangsang daya pikir(kecermatan). c. Membantu mengembangkan keterampilan motorik. 	<p>Weakness</p> <ul style="list-style-type: none"> a. Jika accelerometer pada device tidak berfungsi, maka tidak bisa memainkan game ini. b. Cepat bosan untuk memainkan game ini. c. Hanya dapat dimainkan single player.
	<p>Opportunity</p> <ul style="list-style-type: none"> a. Semakin banyaknya pengguna <i>smartphone</i> android di kalangan masyarakat. b. Banyak peminat game berbasis mobile c. Masih sedikit game yang memiliki konsep sama. 	<p>SO</p> <ul style="list-style-type: none"> a. Membuat game labirin ball berbasis mobile android dan dapat di unduh secara gratis di googleplay setelah di rilis. b. Dengan menggunakan sistem kendali accelerometer akan melatih keterampilan motorik dan menarik pengguna untuk memainkan.
<p>Threats</p> <ul style="list-style-type: none"> a. Semakin banyaknnya <i>game</i> yang sejenis dengan grafis yang lebih bagus menjadikan banyak persaingan. 	<p>ST</p> <ul style="list-style-type: none"> a. Game ini akan berbeda dibandingkan dengan game lain yang sejenis, karena masih jarang game sejenis dengan menggunakan sistem kendali accelerometer. 	<p>WT</p> <ul style="list-style-type: none"> a. Dengan memodifikasi <i>game</i> yang sudah rilis dimungkinkan penambahan fitur penambahan karakter dan map baru. b. Dengan memodifikasi

b. Banyaknya game dengan menawarkan fitur lebih yang lebih menarik		<i>game</i> yang sudah rilis dimungkinkan game ini akan di buat dengan grafis 3D.
--	--	---

3.1.2 Analisis Kebutuhan Sistem

a. Kebutuhan Fungsional

Kebutuhan fungsional merupakan jenis kebutuhan yang berisikan proses-proses apa saja yang diberikan oleh sistem. Kebutuhan fungsional berhubungan dengan informasi-informasi dan fitur-fitur apa saja yang harus ada dalam perangkat lunak. Adapun beberapa kebutuhan fungsional yang harus terpenuhi pada aplikasi ini adalah :

1. User dapat memilih level selanjutnya dengan syarat sudah berhasil menyelesaikan level sebelumnya.
2. Permainan akan dianggap berhasil jika user sudah menumpulkan semua koin yang ada di map dan masuk ke hole clear.
3. Permainan akan dianggap game over apabila user tidak dapat menyelesaikan permainan dengan batas waktu yang ditentukan dan apabila user masuk ke hole fail.
4. Sistem akan menyimpan score tertinggi pada setiap levelnya.
5. Sistem akan mengupdate secara otomatis jika core yang didapat pada level tersebut lebih tinggi dari sebelumnya pada setiap levelnya.
6. *User* dapat melihat score tertinggi yang didapat di setiap level.

b. Analisis Kebutuhan Non Fungsional

Kebutuhan Non-Funtional pada *game* umumnya membahas apa saja yang dibutuhkan sistem itu sendiri untuk menjalankan seluruh kebutuhan fungsional yang ada. Serta membahas kebutuhan sistem dalam pengembangan dan menjalankan *game* tersebut.

c. Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras bertujuan untuk mengetahui spesifikasi pernagkat keras yang digunakan untuk proses pengembangan maupun implementasi aplikasi tersebut. Berikut spesifikasi perangkat keras yang digunakan untuk membangun aplikasi game labyrinth ball ini.

Tabel 3.2. Kebutuhan Perangkat Keras (Pembuatan)

No	Jenis	Spesifikasi
1	<i>Processor</i>	Intel(R) Core(TM) 2 Duo CPU T6500 @ 2.10 GHz
2	<i>Memory</i>	2048MB

3	Hard Drive	TOSHIBA ST9250410S ATA 500 GB
4	Graphic Card	Intel Graphics Media Accelerator 4500MHD

Tabel 3.3. Kebutuhan Perangkat Keras (Implementasi)

No	Jenis	Keterangan	Spesifikasi
1	DISPLAY	Type	TFT capacitive touchscreen, 256K colors
		Size	320x480 pixels
		Multitouch	Yes
			TouchWiz UI Touch sensitive controls
2	CAMERA	Primary	2 MP, 1600x1200 pixels
		Features	Geo-tagging
		Video	Yes, QVGA@15fps
		Secondary	No
3	MEMORY	Card slot	microSD, up to 32 GB
		Internal	180 MB, 290 MB RAM
4	BATTERY		Li-Ion 1200 mAh battery
		Stand-by	Up to 850 h (2G) / Up to 540 h (3G)
		Talk time	Up to 17 h (2G) / Up to 6 h 20 min (3G)

d. Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak bertujuan untuk mengetahui spesifikasi perangkat lunak yang digunakan untuk proses pengembangan maupun implementasi aplikasi tersebut.

Tabel 3.4. Kebutuhan Perangkat Lunak(Pembuatan)

No	Jenis	Spesifikasi
1	Sistem Operasi	Windows 7 Ultimate 32-Bit
2	Tool Editor	Eclipse SDK Version: 3.7.1
3	Database	SQLite
4	Graphics Design	Adobe Photoshop CS3 Corel Draw Graphics Suite X4
		Tiled Map Editor Version 0.8.1
5	Sound Effect	FL Studio 11

Tabel 3.5. Kebutuhan Perangkat Lunak(Implementasi)

No	Jenis	Spesifikasi
1	Versi Android	2.3.6
2	Versi Pita dasar	S5360DXLF1
3	Versi Kernel	2.6.35.7 dpi@DELL169#1
4	Nomor Versi	Ussrv9

4. Implementasi dan Pembahasan

Tahapan ini dilakukan setelah melakukan perancangan sistem dan program sudah siap digunakan. Tahap implementasi sistem ini digunakan untuk mengetahui apakah aplikasi yang telah dirancang dan dibangun sesuai dengan tujuan awal. Mulai dari desain-desain interface hingga game play apakah sudah sesuai pada saat perancangan sistem tersebut.

4.1. Implementasi code program

4.1.1 Implementasi code fungsi tombol back

```
@Override
public boolean onKeyDown(int keyCode, KeyEvent event)
{
    switch(State_Menu_Current)
    {
        case STATE_LOGO:
            break;
        case STATE_MENU_MAIN:
            if (keyCode == KeyEvent.KEYCODE_BACK)
                dialog.show();
            break;
        case STATE_MENU_LEVEL:
            if (keyCode == KeyEvent.KEYCODE_BACK)
            {
                final EngineLock engineLock = mEngine.getEngineLock();
                engineLock.lock();

                Grafik_Detach.menuLevelBg();
                Grafik_Detach.menuLevelBtn();

                engineLock.unlock();

                switchState(STATE_MENU_MAIN);
            }
    }
}
```

Gambar 1.1 Source Code Method onKeyDown

4.1.2 Implementasi code menampilkan Jdialog

```
void dialogClose(){
    dialog = new AlertDialog.Builder(this);
    dialog.setTitle("EXIT ?");
    dialog.setNegativeButton("Tidak", new OnClickListener()
    {
        public void onClick(DialogInterface dialog, int which)
        {
            dialog.cancel();
        }
    });
    dialog.setPositiveButton("Ya", new OnClickListener()
    {
        public void onClick(DialogInterface dialog, int which)
        {
            finish();
        }
    });
}
```

Gambar 1.2 Source Code Jdialog

4.1.3 Implementasi code menampilkan animasi, grafik font

```
public static void menuMainBtn()
{
    for (int i = 0; i < MENU_MAIN_BTN.length; i++)
    {
        tex_Menu_Main_Btn[i] = new BitmapTextureAtlas(
            getTexMan(),
            MENU_MAIN_BTN_TEX_WIDTH,
            MENU_MAIN_BTN_TEX_HEIGHT,
            TextureOptions.BILINEAR);

        reg_Menu_Main_Btn[i] = BitmapTextureAtlasTextureRegionFactory
            .createFromAsset(tex_Menu_Main_Btn[i], activity,
                MENU_MAIN_BTN[i], 0, 0);

        tex_Menu_Main_Btn[i].load();
    }
}

public static void menuLevelBg()
{
    tex_Menu_Level_Bg = new BitmapTextureAtlas(
        getTexMan(),
        MENU_LEVEL_BG_TEX_WIDTH,
        MENU_LEVEL_BG_TEX_HEIGHT,
        TextureOptions.BILINEAR);

    reg_Menu_Level_Bg = BitmapTextureAtlasTextureRegionFactory
        .createFromAsset(tex_Menu_Level_Bg, activity,
            MENU_LEVEL_BG, 0, 0);

    tex_Menu_Level_Bg.load();
}
```

Gambar 1.3 Source Code Load Texture

```
public static void attachMenuMainBg() {
    spr_Menu_Main_Bg = new Sprite(0, 0, activity.RES_WIDTH,
        activity.RES_HEIGHT, reg_Menu_Main_Bg, getVertex());

    m_Scene.attachChild(spr_Menu_Main_Bg);
}
```

Gambar 1.4 Source Code Attach Grafik

4.1.4 Implementasi Code Font

```
public static Font font;

public static void font()
{
    font = FontFactory.createFromAsset(activity.getFontManager(), new BitmapTextureAtlas(
        getTexMan(), 256, 256), activity.getAssets(), FONT_STYLE, 30, true,
        android.graphics.Color.BLACK);
    font.load();
}
```

Gambar 1.5 Source Code Load Font

```
public static void attachGameFontWaktu() {
    time = new Text(320, 0, font, "
        activity.getVertexBufferObjectManager());
    m_Scene.attachChild(time);
}
```

Gambar 1.6 Source Code Attach Font

4.1.5 Implmentasi code dettach grafik

```
public static void dettachMenuMainBg()
{
    spr_Menu_Main_Bg.detachSelf();
}
```

Gambar 1.7 Source Code Detach Grafik

4.1.6 Implementasi code untuk play dan stop sound

```
public static void menuBgm()
{
    try
    {
        bgm_Menu = MusicFactory.createMusicFromAsset(
            getMscMan(), activity, MENU_BGM);

        bgm_Menu.setLooping(true);
    }
    catch (Exception e)
    {
        .
    }
}
```

Gambar 1.8 Source Code Load Sound

```

public static void playMenuBgm()
{
    if (Profil.isSoundEnable())
    {
        bgm_Menu.play();
    }
    return;
}

```

Gambar 1.9 Source Code Play Sound

Gambar 1.10 Implementasi code untuk fungsi touch

```

@Override
public void onClick(ClickDetector pClickDetector, int pPointerID,
    float pSceneX, float pSceneY) {
    // TODO Auto-generated method stub
    for (int i = 0; i < Grafik_Load.MENU_MAIN_BTN.length; i++)
    {
        if(Utils.isOnArea(
            pSceneX, pSceneY,
            Grafik_Load.spr_Menu_Main_Bg,
            Grafik_Load.spr_Menu_Main_Btn[i]))
        {
            switch (i)
            {
                case MENU_MAIN_BTN_PLAY:
                    Grafik_Detach.menuMainBg();
                    Grafik_Detach.dettachMenuMainBtn();
                    State_Activity.switchState(State_Activity.STATE_MENU_LEVEL);
            }
        }
    }
}

```

Gambar 1.11 Source Code Fungsi Touch

4.1.7 Membangun Gameplay

```

private FixtureDef m_Mc_FixtureDef = PhysicsFactory.createFixtureDef(1,
    0.5f, 0.5f);
private FixtureDef m_Wall_FixtureDef = PhysicsFactory.createFixtureDef(0,
    0.5f, 0.5f);
private PhysicsWorld m_PhysicsWorld = new PhysicsWorld(new Vector2(0,
    0), false);

@Override
public void onAccelerationChanged(AccelerationData pAccelerationData) {
    if (State_Menu_Current == STATE_GAME_PLAY) {
        Vector2 gravity = Vector2Pool.obtain(pAccelerationData.getX(),
            pAccelerationData.getY());

        m_PhysicsWorld.setGravity(gravity);

        Vector2Pool.recycle(gravity);
    }
}

```

Gambar 1.12 Souce Code kendali gravitasi Accelerometer

4.1.8 Implementasi code time

```
@Override
public void onUpdate(float pSecondsElapsed) {

    if (State_Activity.State_Menu_Current == State_Activity.STATE_GAME_PLAY) {
        time -= pSecondsElapsed;
        Grafik_Attach.time.setText((int) time / 60 + " : " + (int) time
            % 60);

        if(time+timemasuk < System.currentTimeMillis()){
            State_Activity.switchState(State_Activity.STATE_GAME_OVER);
        }
    }
}
```

Gambar 1.13 Source Code Konversi Time

4.1.9 Implementasi event tabrakan antar object

```
for (int i = 0; i < Grafik_Attach.lst_Game_Play_Hole_Fail
    .size(); i++) {
    if(Grafik_Attach.spr_Game_Play_Mc.collidesWith(Grafik_Attach.lst_Game_Play_Hole_Fail.get(i)))
    {
        detach();
        State_Activity.switchState(State_Activity.STATE_GAME_OVER);
    }
}
}
```

Gambar 1.14 Source Code Event Game Play

4.1.10 Implementasi Highscore

```
@SuppressWarnings("NewApi")
public class ScoreDbHelper extends SQLiteOpenHelper{
```

Gambar 1.15 Source Code SQLiteHelper

```
public ScoreDbHelper(Context context, String name, CursorFactory factory,
    int version) {
    super(context, DBName, null, 1);
}
```

Gambar 1.16 Source Code Create Database

```
public void createTableHighScore()
{
    myDb = this.getWritableDatabase();
    myDb.execSQL(CREATE_TABLE_SCORE);
    myDb.close();
}
```

Gambar 1.17 Source Code Create Table

```
private final String CREATE_TABLE_SCORE = "CREATE TABLE IF NOT EXISTS " +TableHighScore+
    "(Id Integer, Sisa_Waktu Integer)";
```

Gambar 1.18 Source Code String SQL

```

        final String[] DB_INSERT_SCORE = {
            "INSERT INTO " + TableHighScore + " VALUES (1,0)",
            "INSERT INTO " + TableHighScore + " VALUES (2,0)",
            "INSERT INTO " + TableHighScore + " VALUES (3,0)",
            "INSERT INTO " + TableHighScore + " VALUES (4,0)",
            "INSERT INTO " + TableHighScore + " VALUES (5,0)",
            "INSERT INTO " + TableHighScore + " VALUES (6,0)",
            "INSERT INTO " + TableHighScore + " VALUES (7,0)",
            "INSERT INTO " + TableHighScore + " VALUES (8,0)",
        };

        public void insertAwalHighScore(){
            myDb = this.getReadableDatabase();
            for(int i = 0; i<8; i++){
                myDb.compileStatement(DB_INSERT_SCORE[i]).executeInsert();
            }
            myDb.close();
        }
    }

```

Gambar 1.19 Source Code Insert Table

```

        public void updateScore(int level,int score){
            String DB_UPDATE_LEVEL = "UPDATE " + TableHighScore + " SET sisa_waktu=" + score+ " Where Id = " +level;
            myDb = this.getReadableDatabase();
            myDb.compileStatement(DB_UPDATE_LEVEL).executeInsert();
            myDb.close();
        }

        public int cekScore(int index){
            myDb = this.getReadableDatabase();
            Cursor cur = myDb.rawQuery("SELECT sisa_waktu FROM "
            + TableHighScore+ " Where Id = " +index, null);
            while (cur.moveToNext()) {
                myDb.close();
                return cur.getInt(0);
            }
            return 0;
        }
    }

```

Gambar 1.20 Source Code Update dan Select

```

        public void score(){
            float posX = 50;
            float posY = 0;
            for(int i = 0; i<8; i++){
                if(db.cekScore(i+1) != 0){
                    Grafik_Attach.txtHighScore[i].setText("Level " + (i+1) + " " + (int) db.cekScore(i+1)/ 60 +
                    " : " + (int) db.cekScore(i+1) % 60);
                    Grafik_Attach.txtHighScore[i].setPosition( (State_Activity.RES_WIDTH/4)-posX,
                    ((State_Activity.RES_HEIGHT/5))+posY);
                    posY+=20;
                }else{
                    Grafik_Attach.txtHighScore[i].setVisible(false);
                }
            }
        }
    }

```

Gambar 1.21 Source Code Konversi Time

4.2. Instalasi Program

Setelah semua di buat langkah berikutnya instalasi ke *gadget*. *Gadget* yang di gunakan dalam proses instalasi ini adalah Samsung Galaxy Young.

4.3. Pembahasan

Setelah seluruh komponen tersebut di implemmentasikan, maka dibahas apakah semua komponen sudah berjalan sesuai dengan yang di harapkan dan sesuai dengan perancangan

4.4. Manual Program

4.4.1 Tampilan Menu Utama



Gambar 1.22 Tampilan Menu Main

4.4.2 Tampilan Menu Level



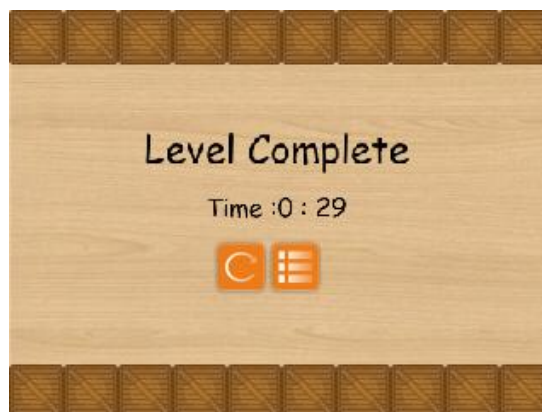
Gambar 1.23 Tampilan Menu Level

4.4.3 Tampilan Game Pause



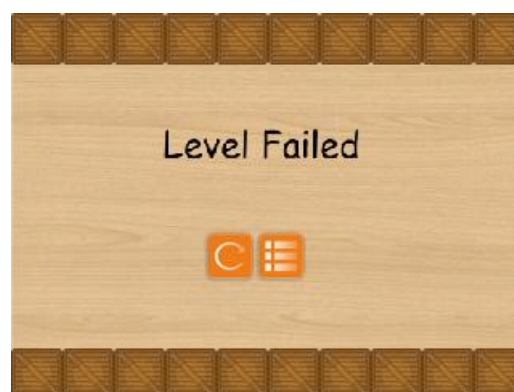
Gambar 1.24 Tampilan Game Pause

4.4.4 Tampilan Game Win



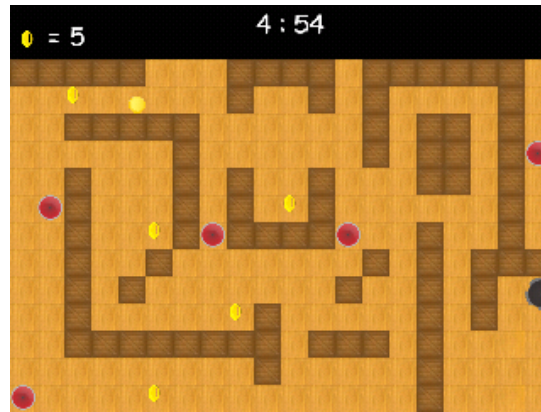
Gambar 1.25 Tampilan Game Win

4.4.5 Tampilan Game Over



Gambar 1.26 Tampilan Game Over

4.4.6 Tampilan Play



Gambar 1.27 Tampilan Help

4.4.7 Tampilan Credit



Gambar 1.28 Tampilan Credit

5. Kesimpulan dan Saran

5.1. Kesimpulan

1. Pembuatan game berbasis java pada android dengan memanfaatkan sensor accelerometer bisa di implementasikan dengan bantuan *library engine* AndEngine dengan extension tambahan TMXTiledMap dan PhysicsBox2D.
2. *Game* ini bertujuan untuk melatih ketangkasan untuk berpikir lebih cepat dan ketelitian
3. Dalam tahapan analisi dan rancangan program menggunakan bantuan uml dapat membantu dan memudahkan saat implementasi ke coding programnya dan struktur program jadi lebih terstruktur.
4. Dari awal perancangan sampai ke tahap implementasi, *game* ini berjalan sesuai dengan yang direncanakan.
5. Berdasarkan pengujian, *game Labirinth Ball* ini sesuai dengan perancangan aplikasi dan tidak ada *error* yang terdeteksi.

5.2. Saran

Mengingat kesempurnaan hanya milik yang maha kuasa, disadari sepenuhnya skripsi ini masih banyak kekurangan sehingga kritik dan saran dari pembaca sangat diharapkan. Kiranya skripsi ini dengan segala kelebihan dan kekurangan dapat diterima serta memberi manfaat yang sebesar-besarnya bagi para pembaca. *Game* ini masih memiliki potensi pengembangan lebih lanjut. Berikut adalah kumpulan saran-saran untuk pengembangan game ini lebih lanjut.

1. Pengembangan *game* ini dapat dilakukan dengan menambahkan level lebih banyak dengan map yang lebih bervariasi lagi.
2. Terdapat *fitur highscore* online agar lebih menarik pengguna berlomba-lomba mendapatkan *score* tertinggi secara *online*.

DAFTAR PUSTAKA

- Al Hanif, Fatta. 2007. *Analisis dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern*. Yogyakarta: Penerbit Andi.
- Nugroho, Adi. 2009. *Rekayasa Perangkat Lunak menggunakan UML dan JAVA*, Yogyakarta: Penerbit Andi.
- Rangkuti, Freddy. 2005. *Analisa SWOT Teknik Membedah Kasus Bisnis*. Jakarta: PT.Gramedia Pustaka.
- Safaat, Nazaruddin. 2011. *Pemrograman android aplikasi mobile smartphone dan tablet berbasis android*. Bandung: H. Informatika
- Winarno, Edy. 2012. *Hacking & programming dengan Android SDK untuk advanced*. Jakarta: SmitDev Community, Elex Media Komputindo.
- Anonim. 2013. *Pengertian JDK JRE dan JVM*. <http://astaq.web.id/2013/04/pengertian-jdk-jre-dan-jvm/>. diakses tanggal 30 November 2013.
- Anonim. 2010. *Eclipse ADT*. <http://developer.android.com/tools/sdk/eclipse-adt.html>, diakses tanggal 10 Maret 2013.