

# TRANSFORMASI LINIER DALAM GAME ANIMASI UNTUK PEMBELAJARAN PERSAMAAN KURVA

Nugroho Agus Haryono, Hendro Setiadi, Setiawan

Fakultas Teknik Program Studi Teknik Informatika

Universitas Kristen Duta Wacana Yogyakarta

Email: cnug@ukdw.ac.id, setiadi.hendro@gmail.com, setiawan.aliong@gmail.com

## **Abstrak :**

Media visual menjadi salah satu daya tarik tersendiri sebagai media pembelajaran. Dengan adanya media visual, suasana belajar menjadi menyenangkan dan pelajar tentunya akan lebih mudah mengerti. Salah satu media visual yang dapat digunakan adalah permainan berbasis komputer atau yang lebih dikenal dengan istilah *game*. *Game* sebagai media visual memiliki kelebihan dibanding dengan media visual yang lain karena *game* mengajak pemainnya untuk turut serta dan andil dalam menentukan hasil akhir dari *game* tersebut.

Dalam tulisan ini dibahas hasil pembuatan *game* 2 dimensi yang dapat menampilkan visualisasi pergerakan objek mengikuti sebuah persamaan kurva dengan menggunakan metode transformasi linier. *Game* yang dibuat terdiri dari 3 level yaitu level pertama berupa *game* penentuan persamaan garis lurus, level kedua berupa *game* penentuan persamaan kurva sinus, dan level ketiga berupa *game* penentuan persamaan kurva cosinus. Masing-masing tahap mempunyai 10 sublevel yang harus diselesaikan untuk dapat menuju level berikutnya.

Dari hasil penerapan metode transformasi linier dalam *game* ini diketahui bahwa kecekungan persamaan kurva mempengaruhi kecepatan pergerakan animasi objek yang mengikuti persamaan kurva tersebut. Semakin cekung persamaan kurva, semakin bertambah cepat pergerakan objek tersebut.

**Kata Kunci :** *Game, transformasi linier, animasi, persamaan kurva*

## **1. Pendahuluan**

Saat ini, perkembangan teknologi terutama di bidang teknologi informasi sangat pesat. Berkembangnya teknologi informasi mendorong perkembangan sarana pembelajaran di berbagai bidang ilmu pengetahuan. Media yang digunakan sebagai sarana pembelajaran beraneka ragam, misalnya pembelajaran dengan metode konvensional seperti membaca buku dan mengerjakan latihan, sampai dengan metode pembelajaran yang interaktif dan tidak membosankan, yakni menggunakan media visual. Media visual menjadi salah satu daya tarik tersendiri sebagai media pembelajaran. Dengan adanya media visual, suasana belajar menjadi

menyenangkan dan pelajar tentunya akan lebih mudah mengerti.

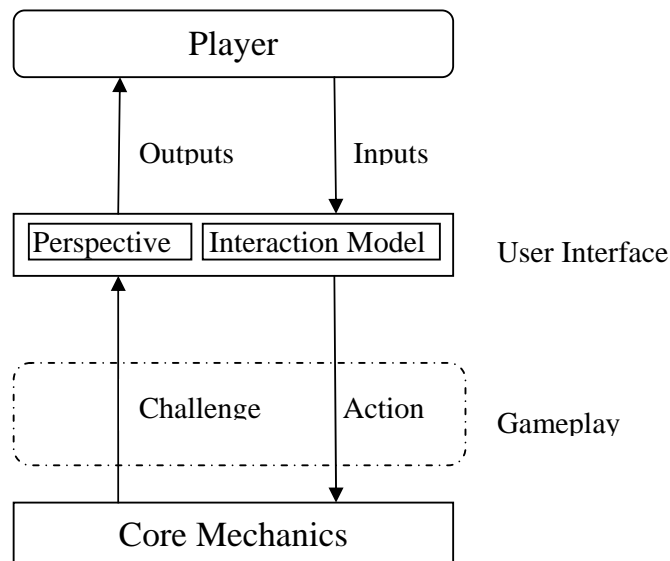
Salah satu media visual yang dapat digunakan adalah permainan berbasis komputer atau yang lebih dikenal dengan istilah *game*. *Game* sebagai media visual memiliki kelebihan dibanding dengan media visual yang lain karena *game* mengajak pemainnya untuk turut serta dan andil dalam menentukan hasil akhir dari *game* tersebut.

## 2. Game

Menurut Ernest Adams dan Andrew Rollings (2007) dalam buku *Fundamentals of Game Design*, *game* merupakan salah satu jenis kegiatan bermain dengan pemainnya berusaha meraih tujuan dari *game* tersebut dengan melakukan aksi sesuai dengan aturan dari *game* tersebut.

### 2.1 Game Design

Menurut Ernest Adams dan Andrew Rollings (2007) dalam buku *Fundamentals of Game Design*, sebuah *game* memiliki beberapa elemen dasar yaitu *core mechanics*, *user interface*, dan *gameplay*. Skema komponen *game design* diberikan dalam Gambar 1.



Gambar 1. Komponen *Game Design* (Adams dan Rollings, 2007:46)

#### **Core Mechanics**

*Core mechanics* menurut Adams dan Rollings (2006:43), adalah suatu model yang menjelaskan rules dari *game* secara spesifik dengan menggunakan simbol-simbol matematika. *Core mechanics* menjelaskan bagaimana cara kerja *game*, *rules* dari *game*, bagaimana pemain

berinteraksi dengan *game*, bagaimana kondisi untuk mencapai *goal* dari sebuah *game*. Secara lebih mendalam, beberapa fungsi dari core mechanics adalah sebagai berikut :

- a. Core mechanics menjelaskan mengenai syarat dan kondisi mengenai keberadaan pemain, apakah pemain masih bisa meneruskan permainan, ataukah pemain tidak dapat melanjutkan permainan.
- b. Memberikan *challenge* (tantangan) kepada pemain
- c. Menerima input dari pemain
- d. Mengubah mode *game* dari satu mode ke mode yang lain. Saat terjadi perubahan mode, *game* akan memberi informasi kepada pemain. Informasi yang dapat diterima antara lain terjadinya perubahan *user interface* dan suara latar.

### **User Interface**

*User interface* menghubungkan antara pemain dengan core mechanics. *User interface* pada *game* berbeda dengan user interface pada software-software komputer pada umumnya. *User interface* pada *game* dapat juga menjadi *challenges* dari *game* tersebut.

Beberapa konsep yang digunakan untuk mendesain *user interface* antara lain:

- a. Perlu adanya konsistensi.
- b. *User interface* memberikan *feedback* yang tepat kepada pemain.
- c. Pemain adalah pemegang kontrol utama dalam *game*. Pemain mengharapkan apa yang dikirim ke sistem memberikan *feedback* sesuai apa yang diharapkan.
- d. Pembatasan pada *user interface* sangat diperlukan karena *game* mempunyai banya aturan (*rule*).

### **Gameplay**

*Gameplay* adalah rangkaian dari *challenges* dan *actions* yang menghubungkan antara user interface dengan core mechanics dari *game*.

### **Challenges (Tantangan)**

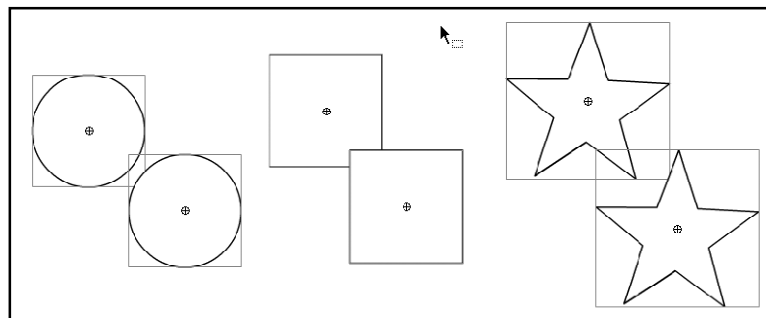
Tantangan adalah serangkaian tantangan yang harus diselesaikan oleh pemain untuk mencapai tujuan dari *game* tersebut. Tantangan yang ada dalam *game* dibagi menjadi dua jenis yaitu tantangan implisit dan tantangan eksplisit.

### **Actions (Aksi)**

Aksi adalah serangkai inputan dari user melalui *user interface* untuk menghadapi *challenges* yang diberikan oleh *game* agar mencapai tujuan dari *game* tersebut. Aksi ini yang menghubungkan antara *user interface* dengan *core mechanics*.

## 2.2 Deteksi Tumbukan (*Collision Detection*)

Menurut Keith Peters (2007) dalam buku **Foundation ActionScript 3.0 Animation, Making Things Move!**, setiap objek memiliki kotak yang mengelilinginya, itulah yang disebut dengan *bounding box*. Dalam lingkungan flash, terdapat dua macam tumbukan yaitu tumbukan berdasarkan *bounding box* dan tumbukan secara titik koordinat. Pada tumbukan berdasarkan *bounding box*, ketika *bounding box* suatu objek bertumbukan dengan *bounding box* objek lain maka kedua objek tersebut dikatakan bertumbukan. Dalam hal ini, penulis menggunakan *bounding box* untuk pembuatan game. Contoh tumbukan objek menggunakan *bounding box* diberikan dalam Gambar 2.



**Gambar 2. Objek Bertumbukan**

## 2.4 ActionScript 3.0

Dalam pembuatan game, penulis menggunakan ActionScript 3.0 sebagai bahasa scripting untuk pengembangan aplikasi dalam Flash. Keuntungan menggunakan ActionScript adalah kita dapat menambah fungsi yang kompleks yang tidak dapat dilakukan jika menggunakan timeline sehingga aplikasi kita tidak statik.

## 3. Implementasi dan Pembahasan




### 3.1 Tampilan Menu Utama

Tampilan menu utama dalam *game* ini berupa halaman yang bagian atas seperti papan tulis dan bagian bawah seperti meja belajar. Di atas meja belajar terdapat tulisan **Game Edukasi Persamaan Garis** dan 3 tombol untuk memilih menu. Tampilan awal game diberikan dalam Gambar 3.



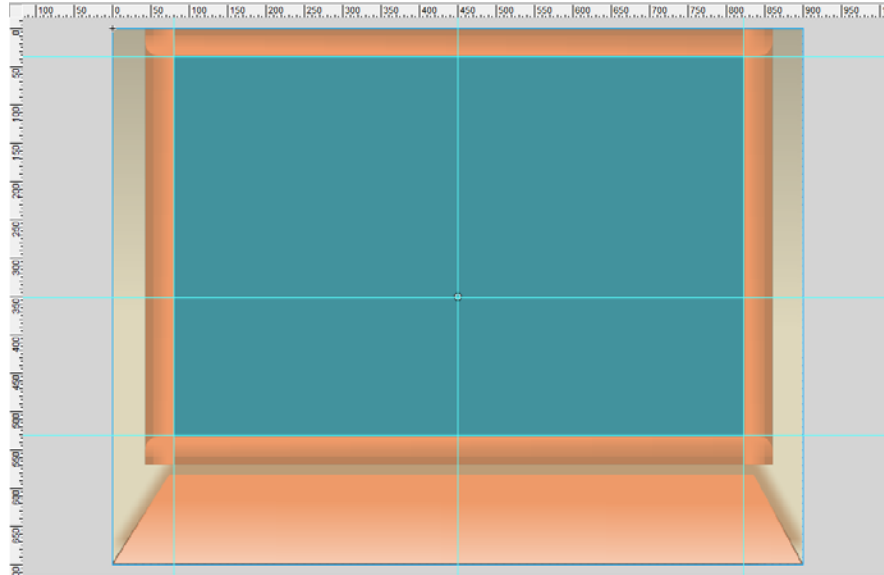
Gambar 3. Halaman Utama

Berikut ini diberikan fungsi-fungsi Ketiga tombol yang ada dalam menu utama.

- Tombol  digunakan untuk memulai permainan.
- Tombol  sebagai fungsi bantuan, digunakan untuk melihat cara main, belajar persamaan garis menggunakan simulasi, belajar persamaan garis menggunakan visualisasi, dan belajar persamaan garis melalui tutorial yang disediakan.
- Tombol  digunakan untuk keluar dari sistem.

### 3.2 Area Utama Permainan

Persamaan garis terletak pada koordinat kartesius yang terdiri dari 4 kuadran yaitu kuadran I, II, III, dan IV. Permainan ini menitikberatkan pada visualisasi persamaan garis tersebut sehingga perlu dibuat koordinat kartesius 4 kuadran. Titik pusat koordinat (0,0) pada Koordinat Kartesius diletakkan pada koordinat (451,286) pada lingkungan flash. Skala jarak dalam sumbu-X dibuat sama dengan skala jarak dalam sumbu-Y. Batas daerah utama *game* (berupa papan tulis) memiliki batas atas kiri (80,36) dan batas bawah kanan (822,532). Posisi-posisi koordinat tersebut dibuat tetap dalam lingkungan flash. Pembuatan koordinat kartesian dalam daerah utama game diberikan dalam Gambar 4.



**Gambar 4. Gambar Penentuan Daerah Utama Game**

### 3.3 Penerapan Transformasi Linier

#### Translasi

Dalam *game* ini, penerapan translasi terlihat pada pergerakan objek saat mengikuti visualisasi dari persamaan garis. Objek selalu bergerak mengikuti koordinat (x,y) dengan x dibuat dari kiri bergerak ke kanan (semakin besar), dan y menyesuaikan hasil fungsi terhadap nilai x. Proses translasi objek diberikan dalam *script* berikut ini.

```
xLama = objek.x;  
xBaru = xLama+2;  
yBaru = fungsi(xBaru);  
objek.x = xBaru;  
objek.y = yBaru;
```

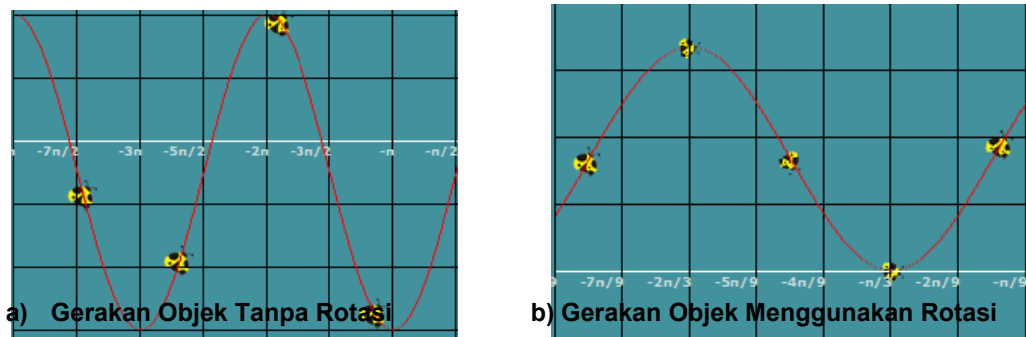
#### Dilatasi

Dalam *game* ini, penerapan dilatasi terlihat pada animasi tembakan saat mengenai dinding batas flash. Tembakan yang berupa bola saat mengenai dinding flash, tembakan tidak akan bergerak ke kanan maupun ke kiri, tembakan tersebut hanya akan mengecil sampai pada batas tertentu, tembakan tersebut menghilang. Proses dilatasi objek diberikan dalam *script* berikut ini.

```
while(skala > 0){
    shoot.scaleX = skala;
    shoot.scaleY = skala;
    skala = skala-0.1; }
```

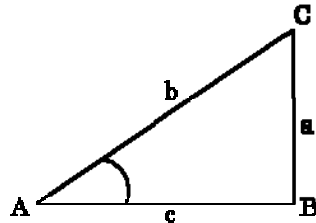
### Rotasi

Dalam *game* ini, rotasi diterapkan ketika objek berjalan mengikuti persamaan sebuah kurva yang melengkung sehingga animasi pergerakan objek akan tampak seperti mengikuti lintasan. Contoh penggunaan rotasi pada objek diberikan dalam Gambar 5.



Gambar 5. Penggunaan rotasi pada objek

Mencari sudut pergerakan objek dari sebuah titik ke titik lain dapat digambarkan sebagai berikut :



Objek mula-mula berada di A kemudian bergerak menuju C. Sudut A adalah sudut pergerakan objek tersebut.  $\tan \alpha = a/c$  atau jika A terletak pada  $(x_0, y_0)$  dan C terletak pada  $(x_1, y_1)$  maka  $\tan A = (y_1 - y_0) / (x_1 - x_0)$ . Untuk mencari sudut  $\alpha$  dapat dihitung dengan menggunakan rumus  $\alpha = \arctan (y_1 - y_0) / (x_1 - x_0)$ . Proses rotasi objek diberikan dalam script berikut ini.

```
sudut = Math.atan2((posisiY-yLama),(posisiX-xLama));
objek.rotation=PUSAT_ROTASI+sudut*180/Math.PI;
```

### 3.4 Model level permainan

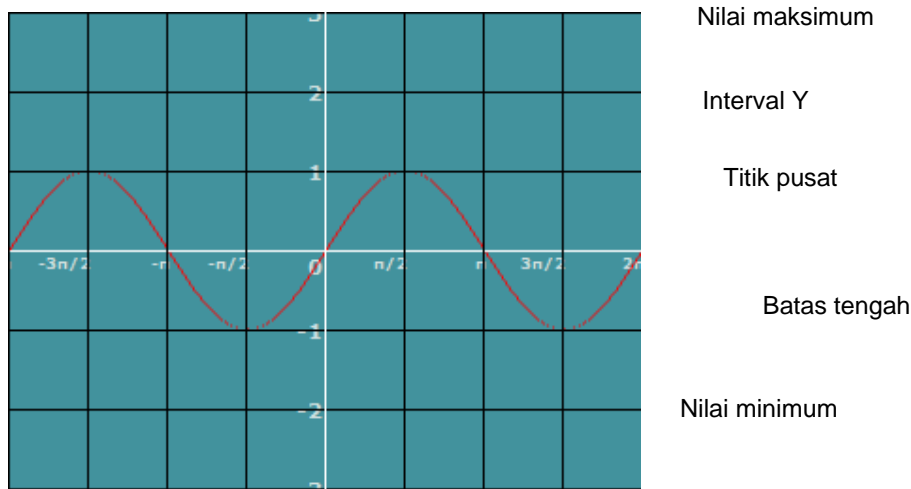
*Game* yang dibuat terdiri dari 3 level yaitu level pertama berupa *game* penentuan persamaan garis lurus, level kedua berupa *game* penentuan persamaan kurva sinus, dan level ketiga berupa *game* penentuan persamaan kurva cosinus. Masing-masing tahap mempunyai 10 sublevel yang harus diselesaikan untuk dapat menuju level berikutnya.

Pada level satu, dibuat sebuah fungsi yang dapat menerima dua masukan dua parameter  $m$  (gradien) dan  $c$  (perpotongan dengan sumbu  $y$ ). Berikutnya dibuat visualisasi garis persamaan fungsi  $y = mx + c$ , yang diikuti oleh pergerakan objek-objek (berupa serangga) sesuai garis tersebut. Objek-objek akan bergerak dari kiri ke kanan sampai keluar batas area utama *game*.

Pada level dua, dibuat sebuah fungsi yang dapat menerima tiga masukan parameter  $a, b$ , dan  $c$  untuk membentuk persamaan  $y = a + b \cdot \sin cx$ . Berikutnya dibuat visualisasi kurva dari persamaan tersebut. yang diikuti oleh pergerakan objek-objek (berupa serangga) sesuai kurva tersebut. Objek-objek akan bergerak dari kiri ke kanan sampai keluar batas area utama *game*.

Pada level dua, dibuat sebuah fungsi yang dapat menerima tiga masukan parameter  $a, b$ , dan  $c$  untuk membentuk persamaan  $y = a + b \cdot \cos cx$ . Berikutnya dibuat visualisasi kurva dari persamaan tersebut. yang diikuti oleh pergerakan objek-objek (berupa serangga) sesuai kurva tersebut. Objek-objek akan bergerak dari kiri ke kanan sampai keluar batas area utama *game*.

Dalam visualisasi kurva sinus dan kurva cosinus digunakan Gambar 6 dan perhitungan berikut ini.



Gambar 6. Kurva  $y = \sin x$

Persamaan  $y = a + b \sin cx$  mempunyai nilai minimum pada  $\min = a - b$  dan akan mencapai nilai maksimum pada  $\max = a + b$ . Batas tengah dari kurva diambil dari  $\text{batasTengah} = \frac{(\min + \max)}{2} = \frac{(a - b) + (a + b)}{2} = \frac{2a}{2} = a$ . Batas tengah pada lingkungan flash dapat diperoleh dengan menggunakan rumus:



$$\text{batasTengahY} = \text{titikPusatY} - (a * \text{intervalY})$$

Nilai **b** digunakan untuk menentukan besarnya skala visualisasi pada sumbu-Y. Nilai **c** digunakan untuk menentukan besarnya skala visualisasi pada sumbu-X

### 3.5 Deteksi Tumbukan (*Collision Detection*)

Dalam *game* ini terdapat 3 tumbukan yaitu :

a. Tumbukan tembakan dengan objek

Untuk mendeteksi tumbukan antara objek serangga dengan objek tembakan (peluru kendali) digunakan fungsi **hitTestObject()**, bawaan ActionScript 3.0. Setiap kali tembakan berhasil mengenai objek serangga, maka objek serangga tersebut akan menghilang dan nilai pemain bertambah. Proses deteksi tumbukan antar objek diberikan berikut ini.

```
for(count=0;count<jumlahObjek;count++){
    if(shoot.hitTestObject(objek[count])){
        objek[count].x=INVISIBLE_X;
        objek[count].y=INVISIBLE_Y;
        nilai+=NILAI_KENA;
        showLabelLevelNilai();
    }
}
```

b. Tumbukan objek dengan batas area permainan

Tumbukan antara objek serangga atau pun tembakan dengan batas area permainan tidak digunakan fungsi **hitTestObject()**, namun hanya menggunakan deteksi koordinat objek. Jika koordinat objek sudah melebihi batas area game, maka objek akan menghilang. Apabila objek serangga mencapai batas area game sebelum tertembak maka nilai pemain akan berkurang. Proses pendeteksian pencapaian batas area diberikan berikut ini.

```
if((objek[count].x>X_MAX||objek[count].y<Y_MIN)&&
objek[count].x != INVISIBLE_X){
    objek[count].x = INVISIBLE_X;
    objek[count].y = INVISIBLE_Y;
    nilai-=NILAI_HANCUR;
    showLabelLevelNilai();
}
```

#### 4. Analisis Kelebihan dan Kekurangan Game

Setelah menerapkan transformasi linier, ditemukan beberapa kelebihan game yang dibuat sebagai berikut:

- *Game* dapat mengacak nilai-nilai koefisien dan konstanta dari persamaan garis yang akan ditampilkan sehingga pemain akan merasa tidak bosan.
- *Game* dapat mengubah nilai pada interval titik koordinat sesuai dengan persamaan garis yang dibangkitkan oleh sistem sehingga pemain tidak akan merasa kesulitan menghitung persamaan yang dimaksud oleh sistem.
- *Game* dapat mengacak nilai pada interval titik koordinat namun tetap sesuai dengan persamaan yang dibangkitkan oleh sistem sehingga pemain akan jarang menemukan jenis persamaan yang sama.
- *Game* dapat memberikan waktu yang tidak jauh berbeda meskipun persamaan yang ditampilkan berbeda.
- *Game* dapat memberikan nilai jika pemain dapat menyelesaikan permainan sebelum batas waktu yang ditentukan selesai.
- *Game* memberikan pembelajaran melalui visualisasi sehingga pemain dapat melatih menebak persamaan sebelum memainkan *game*.
- *Game* memberikan pembelajaran melalui simulasi sehingga pemain dapat mencoba-coba memasukkan input nilai ke dalam persamaan dan mengamati grafik persamaan yang terbentuk.
- *Game* memberikan pembelajaran melalui tutorial secara tertulis sehingga pemain dapat mempelajari materi persamaan yang akan dimainkan.

Sedangkan kekurangan dalam *game* ini adalah sebagai berikut :

- *Game* hanya dapat menentukan waktu pergerakan animasi dengan tingkat akurasi 1 detik sehingga terdapat perbedaan kecepatan pergerakan animasi saat objek yang muncul sudah banyak.
- Beberapa pergerakan animasi objek terlihat kurang halus terutama jika waktu delay objek dibuat terlalu tinggi.

#### 5. Kesimpulan dan Saran

Setelah dilakukan penelitian berdasarkan program yang dibuat, disimpulkan bahwa metode transformasi linier dapat diterapkan dalam pembuatan *game* animasi persamaan kurva. *Game* dapat mengacak nilai-nilai koefisien dan konstanta dari persamaan garis serta mengubah nilai pada interval titik-titik koordinat sesuai dengan persamaan garis tersebut.

Dari penelitian ini juga diketahui bahwa kecekungan persamaan kurva mempengaruhi

kecepatan animasi objek yang mengikuti persamaan kurva tersebut. Semakin cekung persamaan kurva, semakin bertambah cepat pergerakan objek tersebut meskipun penulis telah meneliti bahwa waktu yang ditempuh oleh objek akan selalu sama dari awal objek muncul sampai objek menumbuk batas area permainan.

Pembuatan *game* edukasi ini juga menyertakan cara edukasi lain yaitu pembuatan visualisasi persamaan garis, pembuatan simulasi persamaan garis, dan pemberian tutorial sederhana bagi pemain sebelum memainkan *game* ini.

Rumusan sistem ini dapat dikembangkan lebih lanjut dengan menerapkan konsep desain interaksi yang lebih baik melalui survey kepada pemain. Sistem juga akan menjadi lebih kompleks jika sistem memberikan tantangan yang lebih banyak. Pengembangan sistem ini juga dapat dilakukan dengan menambah materi pembelajaran pada *game* misalnya dengan menambahkan pembelajaran persamaan kuadrat atau persamaan parabola.

#### Daftar Pustaka

- [1] Adams, Ernest., & Andrew Rollings. (2007). ***Fundamentals of Game Design***. New Jersey: Pearson Education, Inc.
- [2] Alexander, Bryan. (2008, July/August). ***Games for Education: 2008***. Educause Review. hlm 64.
- [3] Andre Lamothe. (2004). ***Mathematics For Game Developer***. Thomson Course Technology.
- [4] Peters, Keith. (2006). ***Foundation ActionScript Animation: Making Things Move!***. New York: Springer-Verlag New York, Inc.
- [5] Virvou, M., Katsionis, G., & Manos, K. (2005). ***Combining Software Games with Education: Evaluation of its Educational Effectiveness***. Educational Technology & Society, 8 (2), 54-65.