

## IMPLEMENTASI METODE RAY-CASTING UNTUK GAMES SEMI-3D

Fariz Zikrulhaq Yulfinov<sup>1</sup>, Sri Widowati<sup>2</sup>, Bedy Purnama<sup>3</sup>

<sup>1</sup>Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

---

### Abstrak

Aplikasi berupa game dapat dibagi berdasarkan dimensi, yaitu game 2 dimensi dan 3 dimensi. Game 2 dimensi hanya memiliki 2 ukuran, yaitu panjang dan lebar, sedangkan game 3 dimensi memiliki ukuran tambahan yaitu tinggi, layaknya dunia nyata. Sampai sekarang pasaran game masih didominasi oleh game berbentuk 2 dimensi karena keterbatasan hardware.

Salah satu metode yang dapat memanipulasi game 2 dimensi menjadi seolah-olah game 3 dimensi, atau dapat disebut juga dengan game semi-3D adalah ray-casting. Metode ini mengubah sebuah map 2 dimensi (seperti maze) menjadi seolah-olah 3 dimensi. Ray-casting dapat digunakan untuk membuat sebuah game 3 dimensi yang real time dengan mensimulasikan sebuah cahaya yang terpantul dari sebuah benda menuju mata pengamat yang diimplementasikan kedalam 2 dimensi.

Dengan menggunakan variabel-variabel yang didapat dari map 2 dimensi tersebut, metode ini membuat beberapa variabel baru lain yang semu yang akan memberikan atribut terhadap objek 3 dimensi yang akan dimunculkan. Jumlah cahaya yang dibuat akan mempengaruhi hasil dari gambar 3 dimensi yang dihasilkan beserta performansi.

Kata Kunci : ray-casting, game, 2 dimensi, 3 dimensi, maze, real time.

---

### Abstract

Games can be divided based on the dimensions, ie the 2-dimensional and 3 dimensional game. 2 dimensional game only have 2 sizes, ie length and width, while the three-dimensional game has added size is high, like the real world. Until now the game market is still dominated by two-dimensional form of the game due to hardware limitations.

One method that can manipulate two-dimensional game to be as if the game 3-dimensional, or can be referred to as a semi-3D games are the ray-casting. This method changes a two-dimensional map (like the maze) to be as if the three dimensions. Ray-casting can be used to create a 3 dimensional game that simulates a real-time with the light reflected from an object toward the eye of the observer that is implemented into two dimensions.

By using variables obtained from the 2-dimensional map, this method makes some other new variables are apparent which will provide three-dimensional attributes of objects that will be builded. The amount of light that is made will affect the outcome of the resulting 3-dimensional image and its performance.

Keywords : ray-casting, game, 2 dimensional, 3 dimensional, maze, real time.

---

## BAB I PENDAHULUAN

Pada bab ini dibahas mengenai pendahuluan penelitian yang meliputi latar belakang permasalahan, perumusan masalah, batasan penelitian, tujuan penelitian, serta sistematika penulisan tugas akhir.

### I.1 Latar Belakang

Dewasa ini, perkembangan *game* komputer sudah sangat pesat. Hal ini dapat dilihat dari banyaknya *genre game* yang dapat dimainkan, mulai dari *game* sederhana seperti catur hingga *game* kompleks seperti Red Alert. Selain berdasarkan *genre*, *game* juga dapat dibagi berdasarkan dimensi, yaitu *game* 2 dimensi dan 3 dimensi. *Game* 2 dimensi hanya memiliki 2 ukuran, yaitu panjang dan lebar, sedangkan *game* 3 dimensi memiliki ukuran tambahan yaitu tinggi, seperti dunia nyata.

Walaupun perkembangan *game* sangat pesat, terdapat hal yang membatasi perkembangan *game* tersebut, yaitu *hardware* komputer yang digunakan. Misalnya, untuk dapat memainkan *game* yang berbentuk 3 dimensi diperlukan kebutuhan *hardware* minimum, setidaknya menggunakan *Graphic Processing Unit* (GPU) tersendiri. Sehingga untuk komputer yang tidak memiliki GPU diskrit akan mengalami performansi *game* yang kurang baik, seperti memiliki kualitas gambar yang tidak detail dan *frame per second* yang rendah (dapat dilihat dari animasi *game* tersebut yang tidak mulus). Hal ini menyebabkan pasaran *game* masih didominasi oleh *game* berbentuk 2 dimensi.

Dikarenakan batasan *hardware* tersebut, para *developer game* memikirkan cara membangun sebuah *game* yang memiliki kualitas 3 dimensi dengan kebutuhan *hardware* 2 dimensi. Dari pemikiran ini, muncullah metode-metode yang memanipulasi *game* 2 dimensi menjadi seolah-olah *game* 3 dimensi, atau dapat disebut juga dengan *game* semi-3D. Metode ini pertama kali diterapkan pada *game* Wolfenstein-3D (iD Software) tahun 1992. Pada *game* ini pemain memainkan karakter utama dari sudut pandang orang pertama (yang hanya dapat dicapai oleh *game* 3 dimensi). Bentuk asli dari *game* Wolfenstein-3D ini adalah 2 dimensi, dimana lingkungan merupakan sebuah map dan karakter dapat di-representasikan dengan titik yang berada dalam map tersebut.

Metode yang umum digunakan untuk membangun *game* semi-3D adalah *ray-casting* dan *ray-tracing*. Pada kasus Wolfenstein-3D digunakan metode *ray-casting*. *Ray-casting* merupakan metode yang mengubah bentuk data yang terbatas (sebuah map yang simpel) menjadi proyeksi 3 dimensi dengan menyebarkan cahaya dari sudut pandang karakter.

### I.2 Perumusan Masalah

Berdasarkan uraian diatas, maka permasalahan yang menjadi objek penelitian pada tugas akhir ini adalah:

1. Berapa besar elemen/pixel yang dibutuhkan untuk mendapatkan nilai *frame per second* yang optimal dengan metode *ray-casting*?

2. Bagaimana pengaruh posisi awal dari *player* terhadap kecepatan *load time* sistem?

### I.3 Batasan Masalah

Dalam perancangan tugas akhir ini dibatasi oleh beberapa hal, sebagai berikut :

1. Markup language yang digunakan adalah HTML5.
2. Hanya berbentuk galeri *game* yang bertipe *first-person shooter* (FPS) .
3. Instruksi pengujian hanya *movement*.
4. Waktu komputasi yang dihitung akan direpresentasikan dalam bentuk persentase penggunaan CPU.
5. Perhitungan waktu komputasi hanya pada satu komputer dengan satu kartu grafik dan prosesor dengan satu hingga empat inti.
6. Penggunaan RAM yang dihitung merupakan penggunaan memori oleh sistem itu sendiri tanpa dengan *software* pendukung (*browser*).
7. Bentuk dinding hanya vertikal dan horizontal.

### I.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah :

1. Mengimplementasikan metode *ray-casting* untuk mengetahui nilai elemen/pixel yang dibutuhkan agar mendapatkan nilai *frame per second* yang optimal.
2. Menganalisis pengaruh posisi awal dari *player* terhadap kecepatan *load time* sistem.

### I.5 Metodologi Penelitian

Metode penyelesaian masalah dalam tugas akhir ini adalah:

#### 1. Identifikasi Masalah

Identifikasi masalah-masalah yang berada di ruang lingkup tugas akhir antara lain:

- a. Konsep *game* semi-3D.
- b. Konsep *Ray-casting*.
- c. Konsep *intersection checking*.

#### 2. Perumusan Masalah.

Pendalaman materi yang digunakan untuk mendefinisikan masalah yaitu:

- a. Pembentukan *hex-map* bentuk 2 dimensi.
- b. *Ray-casting* pada bidang map.
- c. Pendeteksian terjadinya *intersection* terhadap dinding.
- d. Proyeksi dinding.

#### 3. Desain Sistem.

- a. Requirement

- Spesifikasi dari berbagai macam hal yang diperlukan untuk mengimplementasikan *ray-casting*.
- b. Entitas  
Pendefinisian berbagai entitas yang terlibat dalam implementasi *ray-casting*.
  - c. Metode  
Analisis dari metode yang akan digunakan.
4. Implementasi.
- Mengimplementasikan aplikasi game yang sesuai dengan rancangan yang telah dibuat dengan spesifikasi minimum sebagai berikut:
- a. Bahasa pemrograman HTML5
  - b. Macromedia Dreamweaver CS3
  - c. RAM 1GB
  - d. Harddisk 250GB
  - e. Processor Intel Atom N550 1,6GHz
  - f. Kartu Grafis Intel GMA2000 32MB
5. Pengujian.
- a. Mengimplementasikan *ray-casting* pada suatu galeri *game*.
  - b. Menghitung performansi penggunaan *ray-casting* terhadap *framerate*, penggunaan resource, dan *load time* pada berbagai *browser*.
  - c. Menganalisis pengaruh posisi awal *player* terhadap kecepatan *load time* sistem.

## I.6 Sistematika Penulisan

Tugas Akhir ini disusun dengan sistematika penulisan sebagai berikut:

### 1. Bab I – Pendahuluan

Pada bab ini dibahas tentang latar belakang penelitian, tujuan penelitian, perumusan masalah, pembatasan masalah, metode penelitian, dan sistematika penulisan tugas akhir.

### 2. Bab II – Landasan Teori

Pada bab ini memuat berbagai dasar teori yang mendukung dan mendasari penulisan tugas akhir ini.

### 3. Bab III – Perancangan

Pada bab ini dijelaskan mengenai tahap-tahap perancangan sistem, dimulai dari tahap pengambilan serta analisa kebutuhan sistem, tahap desain sistem, sampai ke tahap implementasi sistem.

**4. Bab IV – Analisis dan Implementasi**

Pada bab ini dijelaskan analisis data – data yang diperoleh dari hasil percobaan yang menunjukkan kemampuan dan efektifitas implementasi.

**5. Bab V – Kesimpulan dan Saran**

Pada bab ini diberikan kesimpulan dari serangkaian penelitian yang dilakukan dan saran pengembangan selanjutnya.



## BAB V KESIMPULAN DAN SARAN

### V.1 Kesimpulan

Berdasarkan pengujian dan analisis yang telah dilakukan maka didapatkan kesimpulan sebagai berikut:

- a. Jumlah *pixel per element* paling optimal adalah 4 pixel/element, dalam jumlah tersebut, sistem dapat mencapai nilai frame per second yang baik, dengan kualitas gambar yang tidak memiliki aliasing cukup terlihat.
- b. Untuk posisi awal player tidak mempengaruhi *load time*, setelah dilakukan pengujian yang meletakkan karakter pada beberapa tempat dan arah pandang berbeda, ternyata nilai *load time* yang dihasilkan tidak menentu. Ini dikarenakan pada sistem ray-casting tidak perlu me-load elemen dari game, melainkan langsung mengeksekusi fungsi-fungsi yang kemudian akan menggambarkan view 3 dimensi.
- c. Semakin besar nilai *pixel per element* maka nilai rata-rata FPS akan meningkat, rata-rata persentase penggunaan CPU menurun dan penggunaan RAM akan berkurang, dikarenakan pada nilai pixel per element yang semakin besar, maka tingkat detail dari gambar yang dihasilkan akan berkurang sehingga untuk penggunaan RAM akan berkurang diiringi dengan penggunaan CPU.
- d. Semakin besar nilai *pixel per element* akan mengakibatkan munculnya *aliasing*, karena ketika setiap elemen pembentuk dinding menggunakan jumlah pixel yang semakin besar, maka gambar akhir yang dihasilkan akan terlihat bergerigi atau tidak detail.

### V.2 Saran

Berikut ini merupakan saran penelitian yang bisa dilakukan lebih lanjut, yaitu:

- a. *Ray-casting* untuk *first person shooter game* dengan membandingkan dari segi *hardware* untuk mendapatkan penggunaan CPU yang terbaik.
- b. Untuk dinding berbentuk kurva atau melengkung dapat menggunakan rasterization.

Telkom  
University

## DAFTAR PUSTAKA

- [1]. Anonymous. 2010. <http://dev.opera.com/articles/view/creating-pseudo-3d-games-with-html-5-can-1/> [diakses pada 20 September 2011]
- [2]. Anonymous. 2010. <http://dev.w3.org/html5/html-author/> [diakses pada 19 Januari 2011]
- [3]. Anonymous. 2010. <http://diveintohtml5.org/> [diakses pada 19 Januari 2011]
- [4]. Anonymous. 2011. <http://en.wikipedia.org/wiki/2.5D> [diakses pada 8 Oktober 2011]
- [5]. Anonymous. 2011. [http://en.wikipedia.org/wiki/Casual\\_game](http://en.wikipedia.org/wiki/Casual_game) [diakses pada 8 Oktober 2011]
- [6]. Anonymous. 2010. [http://en.wikipedia.org/wiki/Doom\\_engine](http://en.wikipedia.org/wiki/Doom_engine) [diakses pada 27 September 2011]
- [7]. Anonymous. 2010. <http://en.wikipedia.org/wiki/HTML5> [diakses pada 20 Oktober 2010]
- [8]. Anonymous. 2010. [http://en.wikipedia.org/wiki/Ray\\_casting](http://en.wikipedia.org/wiki/Ray_casting) [diakses pada 22 Oktober 2010]
- [9]. Anonymous. 2011. [http://en.wikipedia.org/wiki/Video\\_game](http://en.wikipedia.org/wiki/Video_game) [diakses pada 8 Oktober 2011]
- [10]. Anonymous. 2010. <http://tigcc.ticalc.org/tut/ray-casting.html> [diakses pada 19 Januari 2011]
- [11]. Anonymous. 2010. <http://www.student.kuleuven.be/~m0216922/CG/ray-casting.html> [diakses pada 19 Januari 2011]
- [12]. Anonymous. 2010. <http://www.w3schools.com/html5/default.asp> [diakses pada 19 Januari 2011]
- [13]. Keith, Jeremy. 2010. HTML5 FOR WEB DESIGNERS. A Book Apart
- [14]. Lubbers, P., Albers, B., Salim, F. 2010. Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development. APRESS®
- [15]. Permadi, F. 2010. <http://www.permadi.com/tutorial/raycast/index.html> [diakses pada 19 Januari 2011]

- [16]. Petros. 2011. <http://blog.finalpromise.com/author/petros/> [diakses pada 8 Oktober 2011]
- [17]. Pilgrim, Mark. 2010. HTML5: Up and Running. O'Reilly Media, Inc
- [18]. Purcell, Edwin. J, Varberg, Dale. 1997. Kalkulus dan Geometri Analitis Jilid 1 Edisi Kelima. Penerbit Erlangga
- [19]. S. Haron, Nazleeni, K. Zaima, Mohd, A. Aziz, Izzatdin, H. Hasan, Mohd. 2009. "Performance Evaluation of an Online Text-Based Strategy Game"
- [20]. Vander Veer, Emily. 2005. Javascript for Dummies 4th Edition. Wiley Publishing, Inc

