

PEMBUATAN PERANGKAT LUNAK UNTUK MEMVISUALISASIKAN BENDA TEMBUS PANDANG DENGAN METODE RAY TRACING

Liliana

Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra
e-mail : lilian@peter.petra.ac.id

ABSTRAK: Dewasa ini grafika komputer semakin banyak digunakan di berbagai bidang terutama untuk pembuatan film animasi, iklan dan pembuatan game. Diharapkan teknologi grafika komputer mampu menghasilkan gambar-gambar realistik yang kualitasnya sama dengan kualitas foto. Salah satu metode yang digunakan untuk menghasilkan gambar 3D yang realistik tersebut adalah metode *ray tracing*.

Dalam penelitian ini dibuat perangkat lunak yang mampu menghasilkan gambar-gambar 3D yang realistik terutama untuk benda-benda yang mengkilap dan benda-benda transparan. Benda mengkilap yang dimodelkan bisa memantulkan bayangan benda lain yang berada di sekitarnya. Benda transparan yang dimodelkan adalah benda transparan yang menghasilkan efek kaustik, yaitu pembiasan sinar dari sumber cahaya yang mengumpul di suatu daerah sehingga pada daerah tersebut akan tampak lebih terang daripada daerah sekitarnya.

Kata kunci: efek kaustik, *ray tracing*.

ABSTRACT: Today computer graphics is used in many aspects, especially to make animation, advertisement and game. We hope this technology can produce realistic pictures which same quality with photo. Metode to get the realistic 3D image is ray tracing.

In this journal, make a software which can produce realistic 3D image, especially for reflective and transparent object. Reflective object will modeled can reflect another object surrounding it. And transparent object will modeled can produce caustic effect, that's rays which refract in one area. So that area will appear brighter than area surround it.

Keywords: caustic effect, *ray tracing*.

1. PENDAHULUAN

Untuk menghasilkan gambar yang realistik perlu memodelkan pencerminan dan pembiasan cahaya. Dengan memodelkan pencerminan untuk benda yang reflektif seperti cermin akan dihasilkan pantulan benda lain. Dan efek pembiasan cahaya dapat dimodelkan pada benda yang transparan untuk menghasilkan penampakan obyek lain yang berada di belakang obyek transparan tersebut serta efek pengumpulan cahaya bias.

Pemodelan obyek tembus pandang kebanyakan tidak menampakkan efek kaustiknya. Efek kaustik adalah daerah pengumpulan cahaya yang dibiaskan oleh benda tembus pandang tersebut ke suatu daerah sehingga daerah tersebut nampak lebih

terang dari pada daerah sekitarnya. *Software* ini akan menghasilkan obyek tembus pandang dengan efek kaustik yang dihasilkannya.

2. PEMANTULAN DAN PEMBIASAN CAHAYA

Berikut ini dijelaskan lebih lanjut mengenai pemantulan dan pembiasan terutama cara untuk mendapat arah pantul dan arah bias sinar:

- **Pemantulan Cahaya**

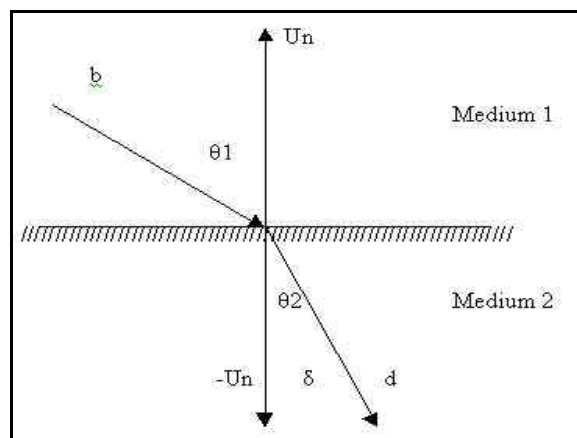
Sudut yang dibentuk antara sinar pantul dengan normal sama dengan sudut yang terbentuk antara sinar datang dengan normal seperti terlihat pada gambar 1.

(4)

$$\ddot{a} = \frac{1}{\sqrt{\frac{c_1^2}{c_2^2} * |c|^2 - |c + U_n|^2}} \quad (5)$$

$$c = \frac{b}{|b \cdot U_n|} \quad (6)$$

Dimana \ddot{a} menunjukkan jarak antara cahaya yang dibiaskan dengan garis normal. Pada persamaan 5, Gambar pembiasan cahaya menurut hukum Snell terlihat pada gambar 2.



Gambar 2. Pembiasan sinar dari medium 1

3. RAY TRACING

Ray tracing adalah suatu metode merender gambar dengan cara menelusuri sinar yang mengenai obyek. Jika sinar yang ditelusuri tersebut mengenai suatu obyek maka diperhitungkan intensitas warna obyek tersebut.

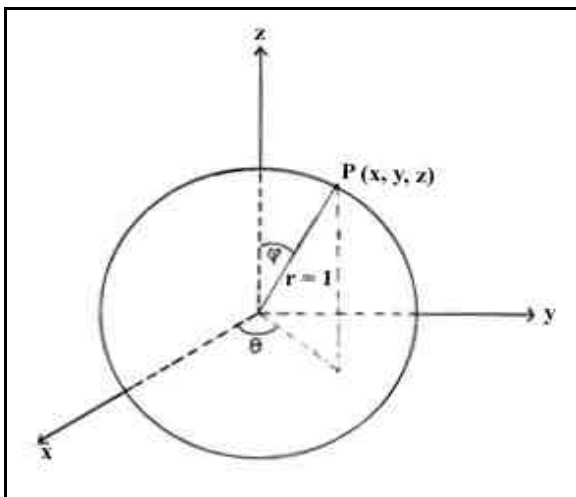
Metode *ray tracing* dibagi menjadi dua jenis, yaitu forward ray tracing dan backward ray tracing. Forward ray tracing digunakan untuk mendapatkan intensitas obyek yang banyak dikenai sinar. Dengan metode ini didapatkan efek caustik dari

obyek – obyek yang transparan. Sedangkan pada backward ray tracing, sinar yang ditelusuri adalah sinar yang mengenai mata.

3.1 Forward Ray Tracing

Pada forward ray tracing, sinar yang ditelusuri adalah sinar yang dipancarkan dari sumber cahaya. Forward ray tracing untuk fase illumination, fase penghitungan sinar yang mengenai obyek. Supaya penelusuran tidak lama maka sinar yang ditelusuri dibatasi jumlahnya. Sumber cahaya yang digunakan pada penelitian ini adalah sumber cahaya berupa titik.

Modelkan sumber cahaya sebagai bola dengan radius satu dan titik pusat posisi sumber cahaya. Dari titik pusat dicari arah pancaran sinar ke seluruh permukaan bola. Hal ini dapat dilihat pada gambar 3.



Gambar 3. Mencari arah pancaran sinar dari sumber cahaya.

Sedangkan cara untuk mendapatkan titik pada permukaan bola digunakan persamaan 7:

$$P(u, v) = (r * \cos(v) * \cos(u), r * \cos(v) * \sin(u), r * \sin(v)) \quad (7)$$

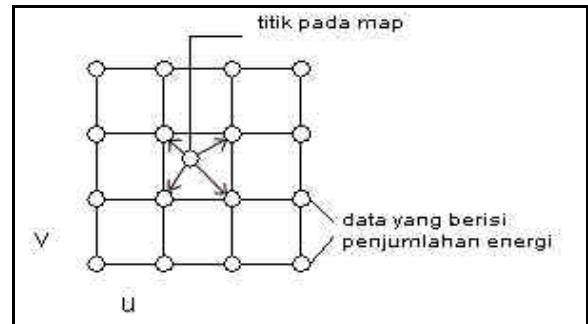
dimana;

r = jari-jari bola (bernilai 1)

u = sudut searah sumbu mendatar bernilai antara 0 s/d 2π

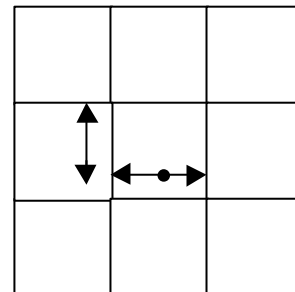
v = sudut searah sumbu tegak bernilai antara $-\pi/2$ s/d $\pi/2$.

Pemetaan (Arvo 1986) menyimpan energi dari sinar dalam suatu array 2D. Karena obyek berupa 3D, maka diubah dahulu ke bentuk $P(u,v)$ nya. Setelah itu distribusikan energi tersebut ke titik – titik disekitarnya, secara bilinear, seperti terlihat pada gambar 4.



Gambar 4. Pemetaan sinar pada permukaan obyek.

Untuk didistribusikan ke empat titik sekitarnya maka distribusikan dulu ke dua titik pada garis yang masing-masing sejajar dengan dua titik ujung, kemudian distribusikan lagi ke dua titik. Untuk lebih jelasnya perhatikan gambar 5:



Gambar 5. Distribusi intensitas ke 4 titik terdekat.

Peta yang menyimpan energi pada permukaan obyek disebut illumination map. Intensitas didapat dengan membagi energi pada illumination map dengan luas daerah yang diwakili. Luas permukaan bola bisa dicari dengan rumus 8.

$$L = 4 * \pi * r^2 \quad (8)$$

dimana :

L = luas permukaan bola

π = konstanta bernilai 3,141596 ($22/7$)

r = radius bola

3.2 Backward Ray Tracing

Backward *ray tracing* menggunakan penelusuran sinar dari mata. Sinar dipancarkan dari mata ke setiap pixel yang membentuk layar gambar dan diteruskan ke obyek-obyek yang akan digambar. Jika sinar yang melalui suatu pixel mengenai obyek maka hitung intensitasnya untuk mewarnai pixel tersebut. Perhitungan intensitas juga memperhitungkan efek pencahayaan.

Untuk mencari vektor dari mata ke tiap pixel, digunakan PO sebagai vektor acuan, yaitu vektor dari mata ke pixel pertama dari kiri atas. Untuk mendapatkan arah vektor ke pixel pertama digunakan persamaan 9.

$$PO = \mathbf{c} + (S / (2 * \text{pix } x)) * \mathbf{x} - (S / (2 * \text{pix } y)) * \mathbf{u} \quad (9)$$

Dimana $\text{pix } x$ adalah lebar layar gambar sedangkan $\text{pix } y$ adalah tinggi layar gambar. S adalah ukuran layar kamera. Vektor \mathbf{u} adalah vektor yang sejajar dengan arah atas kamera. Vektor \mathbf{x} adalah hasil perkalian antara vektor \mathbf{u} dengan vektor arah pandang mata, vektor \mathbf{d} , yang merupakan unit vektor. Dari hasil cross tersebut akan didapatkan vektor acuan untuk arah horisontal. Vektor \mathbf{c} merupakan vektor arah ke ujung kiri atas layar gambar, didapatkan dengan rumus 10.

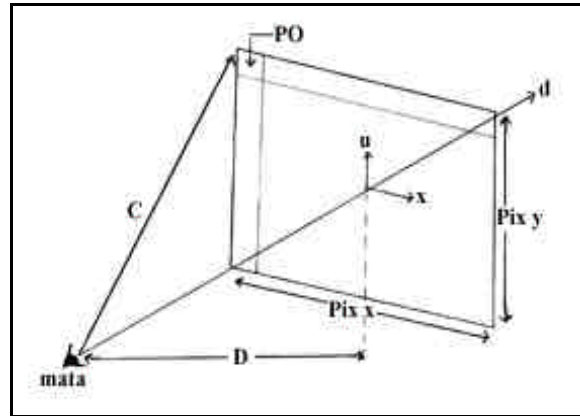
$$\mathbf{c} = D * \mathbf{d} - (S / 2) * \mathbf{x} + (S / 2) * \mathbf{u} \quad (10)$$

dimana D adalah jarak antara mata dengan layar gambar, yang mana hal ini ditentukan oleh sudut pandang mata, yang dapat dicari dengan rumus 11.

$$D = S / 2 / \tan (FOV * \pi / 180) \quad (11)$$

dimana FOV adalah sudut pandang mata yang dinyatakan dalam satuan derajat

Pencarian vektor-vektor untuk lebih jelasnya dapat dilihat pada Gambar 6.



Gambar 6. Vektor-vektor dalam *ray tracing*.

4. DESAIN ALGORITMA

Desain algoritma dibuat berdasarkan proses yang terjadi. Proses ray tracing akan dibagi menjadi dua fase, yaitu fase illumination dan fase rendering. Fase illumination akan dijabarkan pada algoritma 1.

Algoritma 1. Algoritma untuk fase illumination

1. For ($u = 0 ; u < \delta ; u += 2\delta/n$)
2. For ($v = 0 ; v < \delta ; v += \delta/n$)
3. Untuk setiap sumber cahaya
4. Cari arah pancaran sinar
5. Cari titik tabrak sinar dengan semua obyek
6. Jika menabrak obyek maka
7. Jika mengenai benda reflektif maka
8. Cari sinar pantulannya
9. Petakan dengan parameter sinar pantulnya
10. Jika mengenai benda transparan maka
11. Cari sinar biasanya
12. Petakan dengan parameter sinar biasanya

Obyek reflektif jika koefisien reflektifitasnya lebih besar dari 0. Obyek transparan yaitu jika koefisien transparansinya lebih besar dari 0.

Setelah itu barulah fase rendering, dengan backward ray tracing. Untuk pemetaan intensitas pada fase illumination, digunakan algoritma 2.

Algoritma 2. Algoritma untuk memetakan energi sinar pada obyek

1. Jika energi belum mencapai batas tertentu maka
2. Untuk semua obyek yang lain
3. Jika sinar yang dipassing menabrak obyek
4. Cari titik tabraknya
5. Jika benda reflektif maka
6. Cari sinar pantulannya
7. Ulangi proses pemetaan (rekursi) dengan sinar pantulnya dan nilai energi yang dikalikan dengan koefisien reflektifitasnya sebagai parameter
8. Jika benda transparan maka
9. Cari sinar biasnya
10. Ulangi proses pemetaan (rekursi) dengan sinar biasnya dan nilai energi yang dikalikan dengan koefisien transparansinya sebagai parameter.
11. Jika tidak maka petakan energi ke obyek yang ditabrak oleh sinar

Sinar hasil pantulan dan biasan dipetakan ke obyek terus sampai energinya cukup kecil. Setiap kali memantul atau membias maka energinya berkurang dengan mengalikan nilai energi dengan koefisien transparansinya.

5. UJI COBA

Uji coba terhadap perangkat lunak yang telah dibuat dengan menggunakan visual C++ 6.0 dan OpenGL ini bertujuan untuk mengetahui apakah *output* nya sudah sesuai. Untuk uji coba, data – data diatur sebagai berikut:

- Kamera diletakkan di posisi (17, 3, 0) dengan arah pandang sejajar sumbu x negatif dan arah atas sejajar sumbu y positif. Sudut pandang mata adalah 90° .
- Layar pada kamera diset dengan perbandingan 1, layar gambar 300 x 300.
- Warna *background* diatur dengan kombinasi merah 0.3, hijau 0.3 dan biru 0.8
- Lampu dengan koefisien ambient (0.5, 0.5, 0.5), koefisien diffuse dan specular

(0.8 0.8 0.8). Lampu diletakkan di posisi (0, 19, 0). Energinya 20.

- Bidang datar dengan normal (0, 1, 0), jarak -5 dari bidang $y = 0$. Untuk koefisien ambient 0.7, diffuse 0.2, specular 0.65, reflektifitas 0 dan transparansinya 0, tidak bertekstur dan warnanya (0.7, 0.7, 0.7).
- Bola dengan titik pusat (2, 1, -1), jari-jari 4. Koefisien ambient 0.3, diffuse 0.9, specular 0.2, reflektifitas 0.2, transparansinya 0.7 dan index biasnya 1.25. Bola tidak bertekstur dan berwarna (1, 1, 0).
- Bola dengan titik pusat (-12, 2, 4), jari-jari 6.5. Untuk koefisien ambient 0.5, diffuse 0.8, specular 0.2, reflektifitas 0.2 dan transparansinya 0 dan index biasnya 0. Bola tidak bertekstur dan diset dengan warna (1, 0, 0).
- *Skala map* (ukuran luas pada dunia nyata yang diwakili oleh setiap satuan luas pada illumination map) pada bidang datar adalah 2.

Dengan data – data tersebut di atas maka didapatkan hasil seperti pada gambar 7.



Gambar 7. Daerah efek caustik terlalu luas

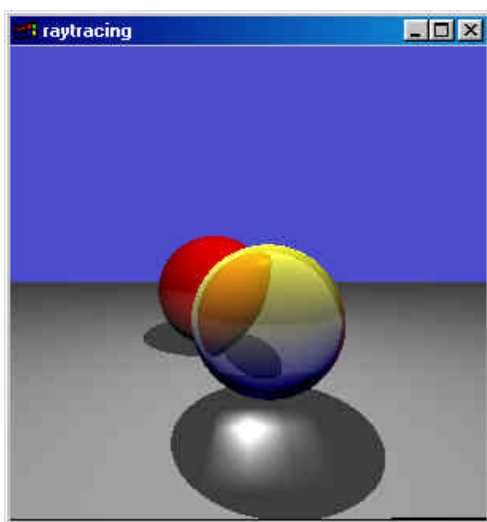
Bola merah yang berada di belakang terlihat dari bola kuning di depannya sehingga menampilkan transparansi pada bola kuning. Daerah berwarna putih yang menunjukkan adanya pengumpulan sinar yang dibiarkan mengumpul sehingga daerah tersebut terang. Daerah yang berwarna terang tampak terlalu luas dan berbentuk segiempat. Hal ini terjadi karena energi yang

mengumpul terlalu besar dan peta yang berbentuk segiempat. Jika pembagi energinya diperbesar maka akan terlihat seperti pada gambar 8.



Gambar 8. Daerah efek caustik semakin menyempit

Dari gambar di atas didapati efek caustik yang dihasilkan semakin melemah namun tampak lebih baik karena daerah efek caustiknya tidak terlalu luas dan tidak terlalu berbentuk segiempat. Jika index biasanya dikurangi menjadi 1.15 dan jika posisi obyek diubah, titik pusatnya menjadi (4, 1, -1) maka gambar yang dihasilkan seperti pada gambar 9.



Gambar 9. Penampakan bola merah semakin berkurang.

6. KESIMPULAN

Dari hasil uji coba yang sudah dilakukan dan evaluasi yang dijelaskan pada bagian uji coba sebelumnya maka dapat diambil kesimpulan sebagai berikut:

1. Obyek tembus pandang yang dimodelkan tidak hanya berhasil menampilkan obyek lain yang ada di belakangnya tapi juga menghasilkan efek caustik seperti yang diharapkan.
2. Efek caustik yang dihasilkan hanya mampu menghasilkan efek caustik yang berwarna putih, bukannya sesuai dengan warna dari obyek transparan. Hal ini karena yang dipetakan bukan intensitasnya tetapi energinya.

DAFTAR PUSTAKA

1. Arvo, J . 1986. *Backward Ray Tracing*. Apollo Computer, Inc. Chelmsfort, MA didownload dari situs: <http://graphics.stanford.edu/courses/cs34-8b-97/basics/intersection/slides/>. 16 Mei 2001
2. Cheah, Shu Chiun. 1996. *An Implementation Of A Recursive Ray Tracer That Renders Caustic Lighting Effects*. Dept. of Computer Science, University of Maryland. didownload dari situs: <http://www.cs.umd.edu/~mount/Indep/Scheah/causticpaper.html>. 7 Juni 2001.
3. Halliday; Resnick; Krane. *Physics 4th Edition Volume 2*. John Willey & sons, Inc. 1960. pp 1020.
4. Hill, F. *Computer Graphics*. Macmillan Publishing Company, New York. 1990. pp.754.
5. Purcell; Varberg. *Calculus with Analytic Geometry 5th Edition*. Prentice-Hall, Inc. 1987. pp 503.
6. Whitted. *Basics of Ray Tracing*. Lecture notes for Spring Quarter. 1997.
7. Woo, M.; Neider, J.; Davis, T. *OpenGL Programming Guide Second Edition*. Silicon Graphics, Inc. 1997. pp.650.