

REALTIME RAY TRACING PADA OBJEK STATIK MENGGUNAKAN POVRAY

Lu'lu Mawaddah Wisudawati^{1a}

¹Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Gunadarma

^alulu_mawadah@staff.gunadarma.ac.id

Abstrak

Perkembangan Teknologi komputer grafis semakin pesat menyebabkan pembuatan program-program aplikasi grafis yang mampu menghasilkan gambar-gambar realtime, terutama gambar tiga dimensi yang photorealistic (sekualitas foto). Untuk menghasilkan gambar tiga dimensi dapat dilakukan beberapa metode salah satunya adalah teknik pencahayaan. Teknik ini menggunakan Ray tracing yaitu Metode merender obyek dengan melakukan penelusuran sinar dari mata ke arah obyek dengan memperhitungkan semua efek pencahayaan yang mengenai obyek tersebut dan juga bayangan dari objek, sehingga mampu menghasilkan obyek 3D yang sesuai dengan aslinya. Untuk membuat Objek tersebut digunakan perangkat lunak Povray. Povray menggunakan algoritma Ray-tracing dalam proses renderingnya. Ray tracing akan membaca file text yang berisi informasi yang menggambarkan objek dan cahaya dalam scene dan menggenerate sebuah image dalam scene dari view point kamera. Ray tracing bukanlah proses yang cepat, tetapi proses ini dapat memproduksi image yang berkualitas tinggi dengan realistic reections, shading, perspective dan efek lain. Dengan menggunakan povray, objek-objek baru 3D bisa dibentuk, seperti piring, buku, kursi, lampu hanya dengan menggunakan objek-objek primitive yang ada dalam povray seperti balok, bola, cincin, text dan kerucut.

Kata Kunci: Realtime, Photorealistic, Pencahayaan, Ray tracing, Rendering, Povray.

Realtime Ray Tracing Pada Objek Statik Menggunakan Povray

Abstrak

Perkembangan Teknologi komputer grafis semakin pesat menyebabkan pembuatan program-program aplikasi grafis yang mampu menghasilkan gambar-gambar realtime, terutama gambar tiga dimensi yang photorealistic (sekualitas foto). Untuk menghasilkan gambar tiga dimensi dapat dilakukan beberapa metode salah satunya adalah teknik pencahayaan. Teknik ini menggunakan Ray tracing yaitu Metode merender obyek dengan melakukan penelusuran sinar dari mata ke arah obyek dengan memperhitungkan semua efek pencahayaan yang mengenai obyek tersebut dan juga bayangan dari objek, sehingga mampu menghasilkan obyek 3D yang sesuai dengan aslinya. Untuk membuat Objek tersebut digunakan perangkat lunak Povray. Povray menggunakan algoritma Ray-tracing dalam proses renderingnya. Ray tracing

akan membaca file text yang berisi informasi yang menggambarkan objek dan cahaya dalam scene dan menggenerate sebuah image dalam scene dari view point kamera. Ray tracing bukanlah proses yang cepat, tetapi proses ini dapat memproduksi image yang berkualitas tinggi dengan realistic reections, shading, perspective dan efek lain. Dengan menggunakan povray, objek-objek baru 3D bisa dibentuk, seperti piring, buku, kursi, lampu hanya dengan menggunakan objek-objek primitive yang ada dalam povray seperti balok, bola, cincin, text dan kerucut.

Kata Kunci: Realtime, Photorealistic, Pencahayaan, Ray tracing, Rendering, Povray.

PENDAHULUAN

Perkembangan teknologi komputer grafis yang pesat menyebabkan banyak pembuatan program-program aplikasi grafis yang mampu menghasilkan gambar-gambar realistis, terutama gambar tiga dimensi yang photorealistic (sekualitas foto). 3D graphics adalah grafik yang merepresentasi objek tiga dimensi dengan menggunakan data geometric (umumnya menggunakan kartesian) yang disimpan di dalam komputer untuk keperluan perhitungan tampilan serta rendering objek 2 Dimension. 3D graphics menggunakan representasi 3 dimensi (cartesian) dalam bentuk ruang (x,y dan z) yang ada dalam komputer dalam format data tertentu untuk keperluan rendering Graphic 2D atau untuk real-time viewing dalam game [Amran, 2010].

3D graphics sering mengacu pada 3D Models. Hal ini dikarenakan karena adanya persamaan dalam hal graphical data file. Namun sebenarnya dua hal ini adalah dua hal yang berbeda. 3D model lebih mengacu pada representasi secara matematis dari suatu objek. Sedangkan 3D graphics lebih mengacu pada representasi objek secara visual yang mendekati realitas. Untuk menghasilkan gambar tiga dimensi dapat dilakukan beberapa metode diantaranya yaitu Lighting atau Illumination, shadow, Difuse Color, ray tracing dan lain-lain. Di dalam paper ini, kita akan membahas tentang ray tracing. Ray tracing adalah suatu metode

untuk merender obyek 3D yang hasilnya realistik seperti foto. Metode ini dilakukan dengan cara menelusuri sinar mata atau sumber cahaya, kemudian diperiksa apakah sinar tersebut mengenai obyek atau tidak. Jika ternyata sinar yang ditelusuri tersebut mengenai suatu obyek maka selanjutnya diperhitungkan intensitas pada obyek tersebut, yaitu intensitas ambient, difuse dan specular. Hasil dari perhitungan intensitas inilah yang terlihat oleh mata. Ray tracing juga dapat didefinisikan sebagai suatu teknik memodelkan cahaya yang menyebar ke berbagai arah dan kemudian menghitung kuat cahaya pada saat cahaya itu mengenai mata. Ray tracing dibagi menjadi dua metode yaitu backward ray tracing dan forward ray tracing. Perbedaan forward dan backward ray tracing terletak pada sinar yang dipancarkan. Backward ray tracing sinar yang digunakan berasal dari penelusuran sinar dari mata, sedangkan forward ray tracing sinar yang dipancarkan berasal dari sumber cahaya.

Karena metode ray tracing akan menghasilkan obyek 3D yang realistik seperti foto, maka untuk memperoleh gambar yang semirip mungkin dengan aslinya, perlu ditambahkan efek pencahayaan, karena pada dunia nyata pun semua benda dapat terlihat karena adanya cahaya. Efek pencahayaan dapat dibedakan menjadi tiga, yaitu ambient, difuse dan specular [3]. Disini metode ray tracing diimplementasikan menggunakan perangkat lunak Persistence of Vision

Raytracer atau disingkat POVRay. Povray adalah suatu program untuk membuat gambar grafis dengan cara menuliskan sebuah code. Sama seperti membuat software, kita tulis dulu kodenya kemudian kita jalankan. Bedanya adalah, jika saat membuat software hasil akhirnya sebuah software jadi, sedangkan povray ini akan menghasilkan suatu gambar grafis [Novrianto, 2003].

Paper ini dibagi dalam beberapa bagian. Pertama-tama kami menjelaskan mengenai latar belakang penulisan paper. bagian kedua yaitu tinjauan pustaka yang menjelaskan mengenai pekerjaan yang terkait dengan masalah dalam paper, bagian ketiga yaitu metodologi dimana membahas mengenai bagaimana cara membuat objek menggunakan ray tracing dengan perangkat lunak pov ray, dan bagian 4 akan berisi simpulan.

TINJAUAN PUSTAKA

Dalam bagian ini akan dijelaskan beberapa pekerjaan terkait yang menjelaskan tentang ray tracing. Liliana dalam papernya yang berjudul [1] "Pembuatan Perangkat Lunak untuk Memvisualisasikan Benda Tembus Pandang dengan Metode Ray Tracing". Dalam paper ini menjelaskan tentang perangkat lunak yang mampu menghasilkan gambar-gambar 3D yang realistis 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 berkumpul di suatu daerah sehingga pada daerah tersebut akan tampak lebih terang daripada daerah sekitarnya.

Samuli Laine, Timo Aila, Ulf Assarsson, Jaakko Lehtinen, Tomas Akenine [Samuli, 2005] dalam papernya yang berjudul "Soft Shadow Volumes for Ray Tracing" menjelaskan tentang suatu algoritma yang cepat untuk rendering secara

fisik yang berbasis bayangan yang lembut dalam ray tracing berbasis render. Metode ini menggantikan ratusan sinar bayangan yang biasa digunakan dalam pelacak ray stokastik dengan sinar bayangan tunggal dan rekonstruksi lokal pada visibilitas fungsi. Dibandingkan dengan menelusuri sinar bayangan, algoritma ini memproduksi gambar yang sama persis ketika mengeksekusi satu ke dua order pada ukuran lebih cepat dalam pengujian uji yang digunakan. Kontribusinya adalah dua tahap metode untuk lebih cepat menentukan tepi siluet yang tumpang tindih di kawasan sumber cahaya, seperti yang terlihat dari titik yang akan diarsir. Kedua, kita menunjukkan bahwa siluet parsial occluders, sepanjang dengan sinar bayangan tunggal, cukup untuk merekonstruksi visibilitas fungsi antara titik dan sumber cahaya.

Meinrad Recheis [Recheis, 2007] dalam papernya yang berjudul "Realtime Ray Tracing" menjelaskan tentang gambar yang menghasilkan kualitas tinggi pada interaktif rata-rata tayangannya. Sementara domain render realtime masih didominasi dengan grafis raster, realtime ray tracing baru-baru ini menjadi jauh lebih kuat. paper ini membahas esensi realtime ray tracing yaitu teknik berbagai percepatan yang menghasilkan kinerja interaktif, bagaimana koherensi antara sinar dapat dimanfaatkan dan efektif paralelisasi. Selain itu, akan dijelaskan komprehensif perbandingan realtime ray tracing dengan grafis raster modern hardware. Dibandingkan dengan grafis raster, ray tracing memungkinkan gambar yang sangat realistis dan sintesis yang paling penting, baik skala untuk adegan massal yang kompleks. Mengingat keuntungan dan fakta bahwa kinerja perangkat lunak realtime ray tracing pada PC memiliki akhirnya melampaui kinerja dari perangkat keras yang didedikasikan Pengubahan pola Postscript untuk model yang sangat kompleks, semua indikasi adalah bahwa ray tracing dapat mengganti Pengubahan pola Postscript render realtime berbasis visualisasi ilmiah dan permainan dalam waktu dekat.

Pada ketiga paper tersebut menjelaskan tentang teknik pencahayaan menggunakan raytracing. Pada paper pertama menjelaskan bagaimana teknik ray tracing tersebut dapat menghasilkan Efek caustik yang berwarna putih, bukannya sesuai dengan warna dari obyek tranparan. Sedangkan paper kedua, mengenai dua tahap metode untuk lebih cepat untuk menghasilkan suatu algoritma yang cepat untuk rendering secara fisik yang berbasis bayangan yang lembut dalam ray tracing berbasis render. Dan untuk paper ketiga, menjelaskan teknik berbagai percepatan yang menghasilkan kinerja interaktif, bagaimana koherensi antara sinar dapat dimanfaatkan dan efektif paralelisasi serta komprehensif perbandingan realtime ray tracing dengan grafis raster modern hardware. Sehingga dalam paper ini akan di jelaskan bagaimana menghasilkan object dengan teknik pencahayaan menggunakan ray tracing dengan metode backward dan forward dan proses rendering.

METODOLOGI

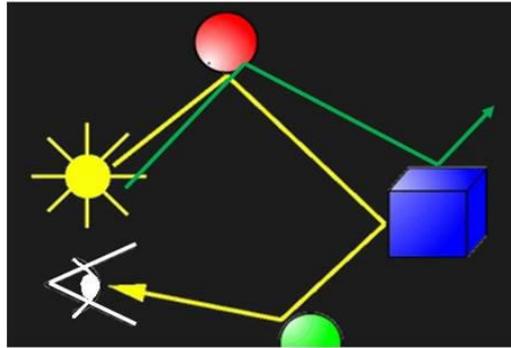
Pada bagian ini akan dijelaskan mengenai pendekatan pencahayaan menggunakan Ray Tracing. Ray Tracing memodelkan cahaya yang menyebar ke berbagai arah dan kemudian menghitung kuat cahaya pada saat cahaya tersebut mengenai mata. Kuatnya cahaya yang diterima oleh mata ditentukan oleh permukaan benda tersebut. Pencahayaan. Ray Tracing menggunakan menggunakan dua metode yaitu backward dan forward ray tracing. Backward ray tracing sinar yang digunakan berasal dari penelusuran sinar dari mata, sedangkan forward ray tracing sinar yang dipancarkan berasal dari sumber cahaya. Dengan dua metode tersebut akan menghasilkan tata pencahayaan dalam ojejek 3D akan menjadi realistis

RAY TRACING

Ray tracing adalah suatu metode untuk merender obyek 3D yang hasilnya realistik seperti foto. Metode ini dilakukan dengan cara menelusuri sinar mata atau sumber cahaya, kemudian diperiksa apakah sinar tersebut mengenai obyek atau tidak. Jika ternyata sinar yang ditelusuri tersebut mengenai suatu obyek maka selanjutnya diperhitungkan intensitas pada obyek tersebut, yaitu intensitas ambient, difuse dan specular. Hasil dari perhitungan intensitas inilah yang terlihat oleh mata. Ada dua konsep dasar yang harus di perhatikan dalam ray tracing ini, yaitu: kita dapat melihat benda karena benda tersebut memantulkan cahaya; jika sinar menabrak permukaan benda maka dapat terjadi 3 hal, yaitu penyerapan, pemantulan, dan pembiasan. Ada pula 3 efek umum yang terjadi pada proses ray tracing, yaitu penyerapan, pemantulan, dan pembiasan cahaya. Di sini pemahaman kita mengenai fisika optik harus digali lagi. Metode ray tracing dibagi menjadi dua jenis, yaitu forward ray tracing dan backward ray tracing.

Forward Ray Tracing

Pada forward ray tracing (lihat Gambar 1), sinar yang ditelusuri adalah sinar yang dipancarkan dari sumber cahaya. Satu hal yang harus diperhatikan adalah bahwa sinar yang dipancarkan oleh sumber cahaya tidak hanya berjumlah puluhan atau ratusan tetapi dapat berjumlah jutaan bahkan lebih. Semua sinar yang dipancarkan tersebut harus ditelusuri satu persatu. Bila setelah proses penelusuran dilakukan, sinar yang sedang ditelusuri tersebut tidak mengenai mata, maka sinar tersebut akan diabaikan, yang berarti akan banyak sekali perhitungan sia-sia yang dilakukan. Hal ini dikarenakan tidak semua sinar yang dipancarkan dari sumber cahaya akan mengenai mata. Dengan menggunakan cara ini, maka untuk menghasilkan gambar yang diinginkan akan membutuhkan banyak waktu.

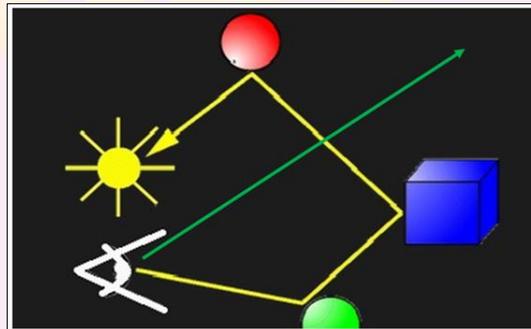


Gambar 1. Forward Ray Tracing

Backward Ray Tracing

Backward ray tracing (lihat Gambar 2) menggunakan penelusuran sinar dari mata. Sinar dipancarkan dari mata ke arah setiap pixel yang membentuk layar gambar dan kemudian diteruskan ke obyek-obyek yang akan digambar. Jika sinar yang melalui suatu pixel tersebut mengenai suatu obyek maka dilakukan perhitungan intensitas pada titik

tabrak obyek tersebut. Intensitas hasil perhitungan tersebut digunakan untuk memberi warna pada pixel tersebut. Perhitungan intensitas yang dilakukan adalah dengan memperhitungkan efek pencahayaan dan efek visual. Jika sinar yang dipancarkan tersebut tidak mengenai obyek sama sekali maka pixel diberi warna sama dengan warna latar belakangnya.



Gambar 2: Backward Ray Tracing

Efek Pencahayaan Pada Ray Tracing

Untuk memperoleh gambar yang semirip mungkin dengan aslinya, perlu ditambahkan efek pencahayaan, karena pada dunia nyata pun semua benda dapat terlihat karena adanya cahaya. Efek pencahayaan dapat dibedakan menjadi tiga, yaitu : Ambient, Difuse, Specular.

1. Ambient adalah efek pencahayaan yang telah membaur dengan lingkungan sehingga arah cahaya tidak dapat diketahui, seakan-akan cahaya datang dari segala arah. Efek ini akan mempengaruhi terang atau tidaknya suatu lingkungan yang terlihat oleh mata. Semakin banyak lampu maka ruangan

semakin terang, sebaliknya jika lampu sedikit maka ruangan remang-remang.

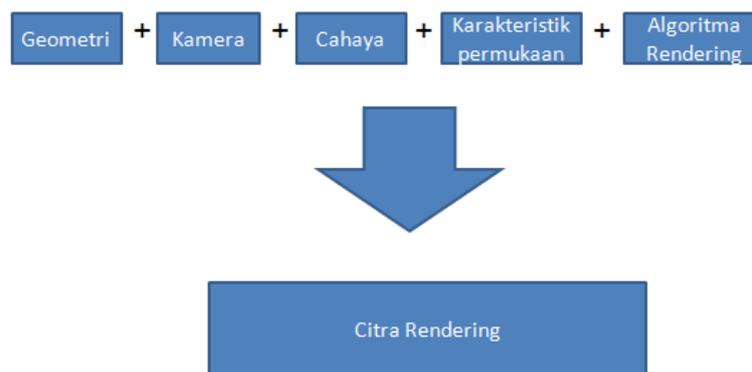
2. Difuse adalah pencahayaan yang tergantung dari besarnya sudut yang dibentuk antara sinar dari lampu ke titik tabrak pada obyek dengan normal obyek. Sehingga posisi lampu sangat mempengaruhi efek difuse ini. Intensitas di_use dapat dicari dengan hukum Lambertian.
3. Specular adalah efek pencahayaan dimana bayangan sumber cahaya terlihat pada permukaan obyek. Efek specular terlihat pada obyek yang mengkilap. Semakin mengkilap permukaan suatu obyek maka makin jelas bayangan sumber cahaya

yang terlihat pada permukaan obyek tersebut.

RENDERING

Rendering merupakan proses untuk menghasilkan sebuah citra 2D dari data 3D. Proses ini bertujuan untuk memberikan visualisasi user mengenai data 3D tersebut melalui monitor atau pencetak

yang hanya dapat menampilkan data 2D. Contoh Metode rendering yang paling sederhana dalam grafika 3D : *Wireframe rendering*, *Hidden Line Rendering*, *Shaded Rendering*. Proses untuk menghasilkan rendering dua dimensi dari objek-objek 3D yaitu dengan mengkombinasikan geometri, kamera, cahaya, karakteristik permukaan, dan algoritma rendering (lihat Gambar 3).



Gambar 3. Proses Rendering

Kamera

Dalam grafik 3D, sudut pandang (point of view) adalah bagian dari kamera. Kamera dalam grafik 3D biasanya tidak didefinisikan secara fisik, namun hanya untuk menentukan sudut pandang kita pada sebuah world, sehingga sering disebut virtual camera. Sebuah kamera dipengaruhi oleh dua buah faktor penting.

1. Faktor pertama adalah lokasi (camera location). Lokasi sebuah kamera ditentukan dengan sebuah titik (x,y,z) .
2. Faktor kedua adalah arah pandang kamera. Arah pandang kamera ditunjukkan dengan sebuah sistem yang disebut sistem koordinat acuan pandang atau system (U,N,V) . Arah pandang kamera sangat penting dalam membuat sebuah citra, karena letak dan arah pandang kamera menentukan apa yang terlihat oleh sebuah kamera. Penentuan apa yang dilihat oleh kamera biasanya ditentukan dengan sebuah titik (x,y,z) yang disebut *camera interest*.

Karakteristik Permukaan

Parameter tekstur direpresentasikan dengan sebuah nama file. File ini akan menjadi tekstur pada permukaan objek tersebut. Selain itu juga ada beberapa parameter dalam tekstur yang berguna untuk menentukan letak tekstur pada sebuah objek, sifat tekstur, perulangan tekstur, dan lain-lain.

Algoritma Rendering

Algoritma Rendering adalah prosedur yang digunakan oleh suatu program untuk mengerjakan perhitungan untuk menghasilkan citra 2D dari data 3D. Kebanyakan algoritma rendering yang ada saat ini menggunakan pendekatan yang disebut scan-line rendering berarti program melihat dari setiap pixel, satu per satu, secara horizontal dan menghitung warna di pixel tersebut.

IMPLEMENTASI PENGGUNAAN POV-Ray

Povray adalah suatu program untuk membuat gambar grafis dengan cara

menuliskan sebuah code. Sama seperti membuat software, kita tulis dulu kodenya kemudian kita jalankan. Bedanya adalah, jika saat membuat software hasil akhirnya sebuah software jadi, sedangkan povray ini akan menghasilkan suatu gambar grafis.

Pengaturan pada PovRay:

```
# povray skyvase.pov +v +d +ft -x
+a0.300 +r3 -q9 -w640 -h480 +b1000
```

Penjelasan opsi yg digunakan:

- +v untuk verbose (output ke layar) proses rendering yang dilakukan.
- +d untuk menampilkan gambar yang sedang di-render. Dengan demikian, Anda bisa melihat sebagian gambar yang telah diproses.
- +ft untuk menentukan output yang dihasilkan disimpan dalam format TGA.
- +a untuk antialiasing.
- +r untuk mengeset kedalaman maksimum rekursi dalam proses sub sampling pixel antialiased.
- +q untuk menentukan kualitas gambar hasil render.
- +w mengeset lebar gambar.
- +h mengeset tinggi gambar.
- +b mengeset besar buffer (dalam kilobyte).

Selanjutnya kita membuat spesifikasi untuk kamera:

```
camera
{
// camera position X, Y, Z
location <0.0, 0.5, -4.0>
// camera focus point X, Y, Z
look_at <0.0, 0.0, 0.0>
}
```

Dimana kita memposisikan kamera pada posisi $x=0.0$, $y=0.5$, $z=-4.0$ dan kita mengatur titik focus kamera pada posisi $x=0.0$, $y=0.0$, $z=0.0$.

Setelah itu melakukan pengaturan cahaya berdasarkan tipe, posisi dan warna. Pencahayaan paling umum adalah pada titik cahaya tersebut dimana cahaya tersebut dapat memancarkan cahaya di semua arah. Dibawah ini pengaturan untuk posisi pencahayaan.

```
// standard point light source
light_source
{
// light's location X, Y, Z
<-30, 30, -30>
// light's colour defined in terms
// of red, green, and blue
color red 1.0 green 1.0 blue 1.0
}
```

Yang ketiga adalah melakukan pengaturan untuk look at yaitu menggunakan sphere (sphere dipilih karena simple untuk di deskripsikan dan cepat untuk di render)

```
sphere {
// centre X, Y, Z and the radius or size
// of the sphere
<1, 0, 0>, 1
// The basic surface colour of the sphere
pigment {
// Another way of defining red, green,
// and blue.
color rgb <1, 0, 0>
}
}
```

Percobaan menggunakan Povray

```

// Real time raytracing demo for POV-Ray 3.7
// Render +rtr +k1a
// start with a small size image like +w160 +h120
#include "colors.inc"
#include "textures.inc"
#include "cube.inc"

global_settings {
  assumed_gamma 2.2
  max_trace_level 3
}
#declare ck = 0.0;
#while (ck < 2.0)

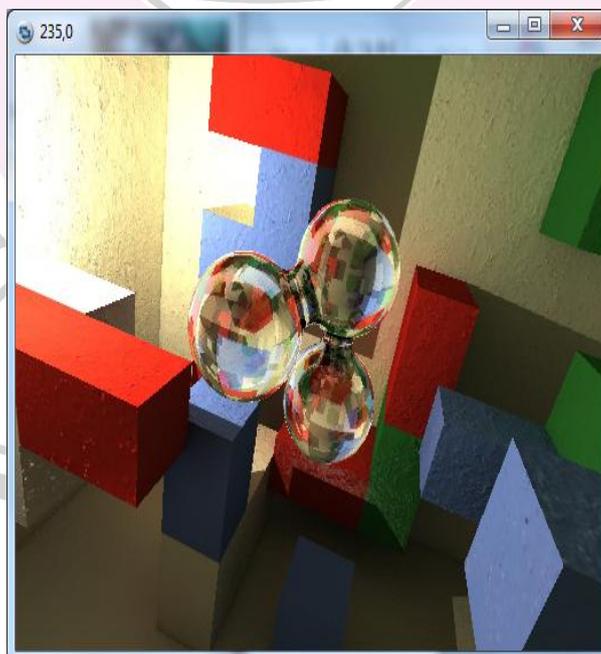
  camera{

    location <-1,1,90>
    direction z=0.8
    up y
    right image_width*x/image_height
    look_at <0,0,0>
    rotate y*360*ck/2
    rotate x*360*ck

    #declare ck = ck + 0.01;
  }
#end
object{Cube rotate <-25,-70,-50>}
blob{
  threshold 0.3
  sphere{<15,5,5>,20,1}
  sphere{<-10,16,-5>,20,1}
  sphere{<-5,-12,-15>,20,1}
  // texture{pigment{Blue} finish{ambient 0 diffuse 0 reflection{0.8, 1}}}
  texture{pigment{rgbf <1,1,1,1>} finish{ambient 0 specular 0.8
  roughness 0.01 diffuse 1.0 reflection{0.4 1}}}
  interior{ior 0.8}
}

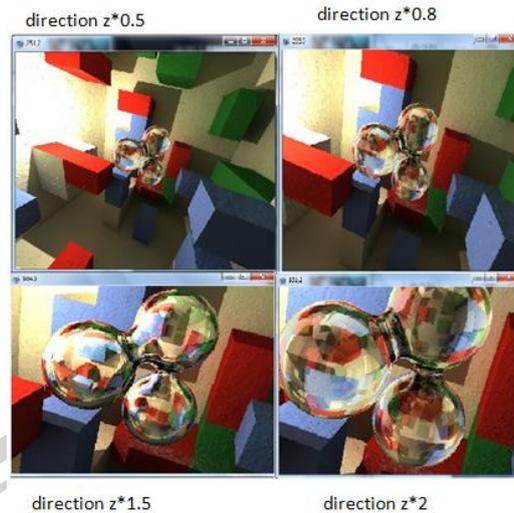
```

Output yang dihasilkan pada Gambar 4



Gambar 4. Hasil Ray Tracing menggunakan Povray

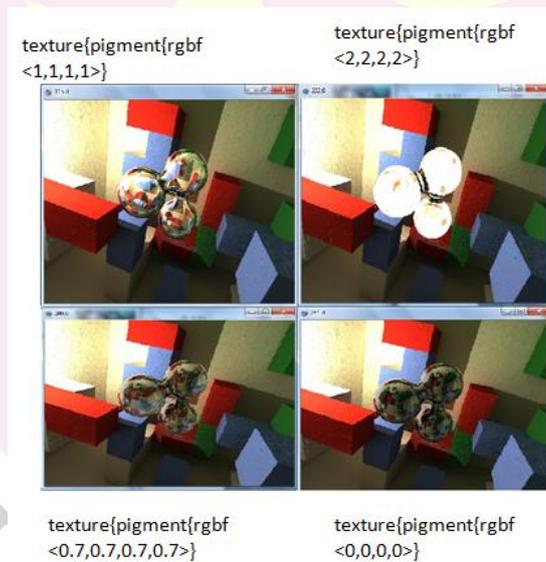
Jika dilakukan perubahan untuk direction
0.5, 0.8, 1.5, dan 2:



Gambar 5. Pengaturan Object Direction pada Povray.

Dari Gambar 5 tersebut maka didapat semakin besar direction, maka semakin dekat object mendekati kamera.

Dibawah ini output yang dihasilkan untuk parameter pigment $\langle 1,1,1,1 \rangle$, $\langle 2,2,2,2 \rangle$, $\langle 0.7,0.7,0.7,0.7 \rangle$ dan $\langle 0,0,0,0 \rangle$



Gambar 6. Perubahan Object texture Pigment

Dari hasil percobaan pada Gambar 6. dapat dilihat bahwa semakin besar nilai pigmentnya maka semakin terang(putih) warna yang dihasilkan, dan sebaliknya semakin

kecil parameter pigment yang diberikan maka semakin gelap object tersebut. Untuk percobaan kedua yaitu pada kaca merah:

```
#declare Kaca_Merah1 =
texture {
  finish {
    ambient 1
    diffuse 1
    reflection 1
    specular 2
    roughness 0.1
```

```

    }
    pigment { color rgbf <0.9, 0.1, 0.2, 0.8> }
}

```

Output:



Gambar 7. Percobaan Kaca Merah dengan Nilai Parameter 1

```

#declare Kaca_Merah2 =
texture {
  finish {
    ambient 0
    diffuse 0
    reflection 0
    specular 0
    roughness 0
  }
  pigment { color rgbf <0.9, 0.1, 0.2, 0.8> }
}

```

Output:



Gambar 8. Percobaan Kaca Merah dengan Nilai Parameter 0

Dari percobaan kaca merah (Gambar 7 dan Gambar 8) dapat disimpulkan bahwa semakin besar nilai koefisien ambient, diffuse, reflection, specular, roughness maka

object/hasil renderingnya semakin terang atau kasar.

SIMPULAN

Ray tracing menghasilkan gambar yang lebih realistik/fotorealistis dan dapat digunakan untuk out-door dan in-door scene. Komponen ray tracing terdiri dari: Plane, merupakan bidang data sebagai latar objek ray tracing; Shadow, pembentukan bayangan pada objek; Lighting, pencahayaan objek; Refleksi, pencerminan objek; Refraksi, pembiasan objek.

Dengan metode ray tracing, akan didapatkan citra hasil render dengan model pencahayaan yang realistis. Salah satu kelemahan ray tracing adalah munculnya alias pada citra hasil rendahnya. Alias timbul karena undersampling, yaitu pengambilan sample dengan jumlah yang kurang. Alias dalam ray tracing nampak dalam bentuk "jaggies" pada citra Hasil. Untuk menghilangkan alias, ada berbagai metode anti-aliasing di antaranya ordered grid supersampling (OGSS). Metode ini melakukan supersampling pada citra untuk meningkatkan jumlah sampel, kemudian melakukan downsampling kembali pada citra yang diinginkan.

DAFTAR PUSTAKA

- [Amran, 2010]T. Amran. Graphic 3d.<http://my.opera.com/taniadwy/blog/graphic-3d>, February 2010.
- [Novrianto, 2003] Noprianto. Console-console yang meriah.www.infolinux.web.id, October 2003.
- [Recheis, 2007] M. Recheis. Realtime ray tracing. June 2007.
- [Samuli , 2005] U.A.J.L.T.A. Samuli Laine, Timo Aila. Soft shadow volumes for ray tracing. volume 24, 2005.