KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)
Terakreditasi Nomor 204/E/KPT/2022 | Vol. 5, No. 3, Juli (2024), pp. 1174-1182

Analisis Performa Raytracing dan MCMC Pada Realisme Visualisasi Obyek 3D Dengan Terintegrasi MIPMapping

Vincensa Woytimena Budet[1], Agustinus Rudatyo Himamunanto[2], Haeni Budiati[3]*
[1,2,3]Universitas Kristen Immanuel, Yogyakarta, Indonesia
E-mail: vincensawb@gmail.com[1], rudatyo@ukrim.ac.id[2], heni@ukrim.ac.id[3]

Abstract

The development of computer graphics has resulted in an increasingly realistic and immersive digital world, especially in the field of 3D object representation. One of the techniques for image presentation is ray tracing, however, regular ray tracing requires long computation time. To achieve high realism in 3D objects, complex computational operations and the use of appropriate algorithms are required. In this research, Markov chain Monte carlo (MCMC) algorithm has the potential to achieve realism on a 3D object. This research analyzes the performance comparison between ordinary ray tracing and MCMC algorithm in achieving realism on 3D objects and integrating Mipmapping technology to improve the visual quality of 3D objects. The results are measured by calculating the PSNR value on the rendered object and comparing the noise level of a 3D object rendered with ordinary ray tracing, and ray tracing using the Monte carlo algorithm. The number of samples used were 50 samples of 3D objects tested with Monte Carlo and obtained a result of 94%, and with ordinary ray tracing of 6% which is indicated by the level of distortion or error that occurs in the processed object. This shows that by rendering using the MCMC algorithm the image quality of the rendered object is better than rendering using ordinary ray tracing.

Keywords: *Computer graphics, ray tracing, Markov Chain Monte Carlo (MCMC), Mipmapping, realism, efficiency*

Abstrak

Perkembangan grafika komputer telah menghasilkan dunia digital yang semakin realistis dan imersif, terutama pada bidang representasi objek 3D. Salah satu teknik untuk presentasi citra adalah ray tracing, namun, ray tracing biasa membutuhkan waktu komputasi yang lama. Untuk mencapai realisme yang tinggi pada objek 3D, diperlukan operasi komputasi rumit dan penggunaan algoritma yang tepat. Dalam penelitian ini algortima Markov chain Monte carlo (MCMC) berpotensi dalam mencapai realisme pada suatu objek 3D. Penelitian ini menganalisis perbandingan kinerja antara ray tracing biasa dan algoritma MCMC dalam mencapai realisme pada objek 3D dan mengintegrasikan teknologi Mipmapping untuk meningkatkan kualitas visual objek 3D. Hasil penelitian diukur dengan menghitung nilai PSNR pada objek hasil rendering dan membandingkan tingkat noise atau kebisingan dari suatu objek 3D hasil rendering dengan ray tracing biasa, dan ray tracing yang menggunakan algoritma Monte carlo. Jumlah sampel yang digunakan masing-masing sebanyak 50 sampel objek 3D yang diuji dengan monte carlo dan mendapatkan hasil sebesar 94%, dan dengan ray tracing biasa sebesar 6% yang ditunjukkan oleh tingkat distorsi atau kesalahan yang terjadi pada objek yang diproses. Hal ini menunjukkan bahwa dengan rendering menggunakan algoritma MCMC kualitas Gambar dari objek hasil rendering lebih baik dari rendering menggunakan ray tracing biasa.

Kata kunci: *Grafika Komputer, Ray Tracing, Markov Chain Monte Carlo (MCMC), Mipmapping, Realisme, Efisiensi*

ISSN: 2720-992X

1. Pendahuluan

Grafika komputer telah berkembang pesat dalam beberapa dekade terakhir, memberikan dunia digital yang semakin realistis dan imersif, hal ini sejalan dengan peningkatan pemanfaatan teknologi informasi diberbagai bidang. Salah satu pemanfaatan teknologi informasi adalah dalam mendukung pengolahan citra dalam bidang desain dan multimedia seperti rendering. Proses grafis 3D pada komputer yang mengkonversi model 3D ke dalam bentuk Gambar 2D pada komputer disebut sebagai 3D Rendering [1]. Untuk mencapai tingkat realisme yang tinggi dalam representasi objek 3D diperlukan metode rendering yang mendukung. Dalam rendering membutuhkan waktu komputasi yang lama untuk menghasilkan hasil yang memuaskan.

Ray tracing dan algoritma MCMC ditambah dengan pengintegrasian mip mapping berpotensi memberikan solusi dalam proses rendering. MCMC dapat membantu meningkatkan kinerja ray tracing dengan menangani distribusi probabilitas yang kompleks, sehingga menghasilkan visualisasi yang lebih realistis dalam waktu yang lebih efisien [2]. Ray tracing dan algoritma MCMC ditambah dengan pengintegrasian mip mapping berpotensi memberikan solusi dalam proses rendering. MCMC dapat membantu meningkatkan kinerja ray tracing dengan menangani distribusi probabilitas yang kompleks, sehingga menghasilkan visualisasi yang lebih realistis dalam waktu yang lebih efisien [3].

Mengintegrasikan teknologi mip mapping dengan metode ray tracing dan algoritma MCMC untuk meningkatkan kualitas visual objek 3D [4]. Mengembangkan metode dan algoritma yang lebih efisien untuk mengurangi waktu komputasi yang dibutuhkan oleh ray tracing. Melakukan analisis perbandingan kinerja antara ray tracing biasa dan algoritma MCMC untuk mencapai realisme pada objek 3D. Dengan menggunakan teknologi mip mapping untuk memaksimalkan penggunaan tekstur pada objek 3D dan mengintegrasikan metode ray tracing dengan algoritma MCMC untuk meningkatkan realisme visual objek 3D.

Penelitian ini akan membuktikan bahwa integrasi antara ray tracing, MCMC, dan Mipmapping dapat menghasilkan visualisasi obyek 3D yang lebih realistis dalam waktu komputasi yang lebih efisien, membuka potensi baru dalam pengembangan aplikasi grafis dan industri animasi.

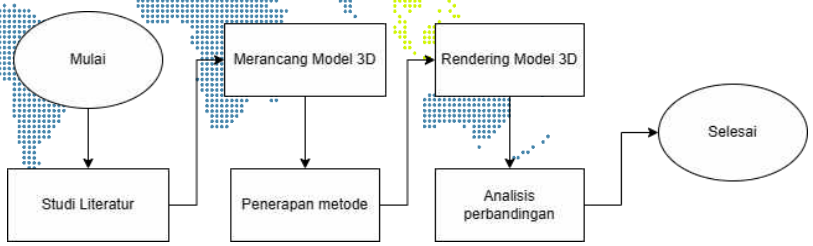
Beberapa penelitian telah dilakukan oleh para peneliti dalam meneliti performa Ray tracing, algoritma monte carlo, dan teknologi Mip Mapping. Penelitian [5] yang berjudul **"Real-Time Ray Tracing for Interactive Global Illumination"** membahas penggunaan teknik ray tracing secara real-time untuk menghasilkan iluminasi global interaktif pada grafika komputer. Peneliti menerapkan pendekatan yang efisien untuk mengatasi tantangan komputasi yang kompleks dalam real-time ray tracing, dengan fokus pada iluminasi global. Hasil yang mereka dapatkan adalah pengembangan teknik ray tracing yang memungkinkan visualisasi real-time yang lebih realistis dengan iluminasi global yang lebih baik.

Penelitian yang berjudul **"Efficient Ray Tracing of Volume Data Using Multi-Dimensional MIP Mapping"** [6] mengusulkan metode efisien untuk melakukan ray tracing pada data volume dengan menggunakan teknik Multi-Dimensional MIP Mapping. Hasil penelitian menunjukkan bahwa penggunaan Multi-Dimensional MIP Mapping dapat menghasilkan ray tracing yang lebih efisien pada data volume. Pada penelitian ini akan digunakan pendekatan yang berbeda dalam mengintegrasikan teknik raytracing, MCMC, dan MIP Mapping dalam visualisasi objek 3D yang diharapkan mampu memberikan perbandingan antara penggunaan metode rendering dalam menghasilkan objek 3D yang lebih realistis.

2. Metodologi Penelitian

Pada penelitian ini terdapat beberapa proses atau tahapan didalam eksperimen, yaitu dimulai dari studi literatur, perancangan model 3D, penerapan metode, rendering model

3D, dan analisis perbandingan yang bisa dilihat pada Gambar 1.

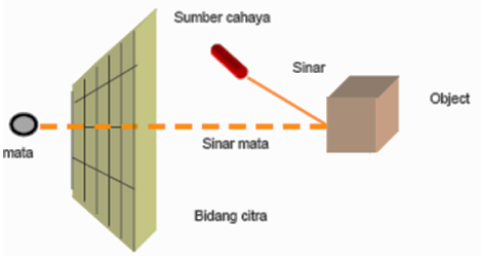


Gambar 1. Diagram Alir Pengembangan *Ray Tracing*

a) Studi Literatur
Pada tahap ini, peneliti akan melakukan studi terkait dengan penelitian yang akan dilakukan untuk mendapatkan pemahaman lebih mendalam mengenai Ray Tracing, Algoritma Monte Carlo dan MIP Mapping, serta uji PSNR.
b) Merancang Model 3D
Pada tahap ini peneliti akan memilih objek 3D disekitar dan merancang model objek 3D kedalam software blender, yang akan digunakan untuk eksperimen penelitian. Tanpa adanya perancangan model, tidak akan ada atau tidak akan bisa melakukan proses rendering.
c) Penerapan Metode
Pada tahap ini, setelah model 3D dibuat, maka Ray tracing, algoritma monte carlo, dan Mip mapping akan diimplementasikan kedalam model objek 3D. Fitur didalam software blender sendiri secara tidak langsung mendukung ray tracing dan algoritma monte carlo didalamnya,
d) Rendering Model 3D
Pada tahap ini, peneliti akan melakukan renderering pada model objek 3D yang sebelumnya sudah dirancang di software blender. Objek 3D yang sudah direnderer akan digunakan sebagai bahan analisis.
e) Analisis Perbandingan
Pada tahap ini, akan dilakukan analisis terhadap kinerja motode ray tracing biada dan ray tracing yang menggunakan algoritma monte carlo, yaitu dengan membandingkan waktu komputasi, jumlah memori yang digunakan untuk memproses rendering, serta nilai PSNR dari kedua metode.

2.1. Ray Tracing

Ray tracing merupakan salah satu metode pencahayaan global yang sudah lama digunakan dalam dunia grafika komputer [7]. Ray Tracing melibatkan pelacakan setiap piksel cahaya, dimulai dari mata atau titik pandang dan bergerak melalui piksel tersebut keposisi objek yang sedang dilihat [8]. Cahaya merambat melalui sekelilingnya dalam pola refleksi dan perpotongan dengan mengabaikan objek yang jauh [9].



Gambar 2. Pemodelan Ray Tracing

Terlihat pada Gambar 2, sinar bergerak menelusuri tiap sumber cahaya dari lokasi sinar mata dan objek perpotongan. Sinar bayangan dapat digunakan untuk mengukur

intensitas dan warna dari titik perpotongan objek atau sinar mata. Proses ini akan menghasilkan warna dan intensitas untuk setiap piksel. Proses ini dilakukan pada piksel dalam bidang citra. Setelah selesai, bidang citra akan menampilkan Gambar dari peristiwa dengan resolusi yang ditentukan oleh pikselisasi bidang citra. Ketika kondisi tersebut terpenuhi, bidang citra akan menampilkan Gambar dari peristiwa dengan resolusi yang telah ditentukan oleh pikselisasi bidang citra. Dengan kata lain, setiap piksel dalam bidang Gambar mengandung sebagian informasi visual tentang peristiwa yang diambil, dan jumlah piksel dalam bidang Gambar menentukan seberapa rinci atau halus Gambar yang direproduksi [10].

Resolusi pikselisasi bidang Gambar juga mempengaruhi seberapa jelas dan detail Gambar yang dapat ditampilkan. Energi cahaya yang datang ke permukaan yang bukan kaca sempurna dipantulkan secara acak (pemantulan yang tidak sederhana) [11]. Untuk permukaan spekular, arah penyebaran cahaya dapat dianggap sebagai variabel acak yang mengikuti distribusi tertentu, tergantung pada sudut datang cahaya dan sifat permukaan tersebut. Pada setiap perpotongan, sinar pantul dipantulkan dalam arah yang ditentukan melalui sampling distribusi, yang digunakan untuk memodelkan penyebaran sinar.

2.2. Algoritma Markov Chain Monte Carlo (MCMC)

Algoritma monte carlo merupakan metode komputasi yang digunakan untuk menyimulasikan ulang berbagai perilaku sistem fisika dan matematika. Dalam konteks grafika komputer, algoritma monte carlo biasanya digunakan untuk menghasilkan efek pencahayaan yang realistis, contohnya seperti bayangan, refleksi, dan penyebaran cahaya. Pada dasarnya, algoritma monte carlo bekerja dengan menghasilkan sejumlah sampel acak yang mewakili peristiwa yang tidak pasti [12]. Kemudian sampel ini dievaluasi untuk mendapatkan hasil akhir yang mendekati dengan solusi atau yang diharapkan sebenarnya. Dalam konteks grafika komputer, sampel-sampel ini dapat mewakili sinar Cahaya yang dipancarkan dari sumber Cahaya dan berinteraksi dengan objek didalam
adegan.

Manfaat utama yang diberikan oleh algoritma Monte Carlo dalam grafika komputer adalah kemampuannya untuk menciptakan efek pencahayaan yang terlihat nyata dengan biaya komputasi yang cukup efisien[13]. Meskipun demikian, kelemahan utamanya muncul dalam bentuk noise atau penurunan kualitas Gambar karena sifat acak dari metode ini. Untuk mengatasi tantangan ini, seringkali diterapkan teknik seperti penggabungan sampel (sample merging) dan pengurangan noise (noise reduction) [14].

2.3. MIP Mapping

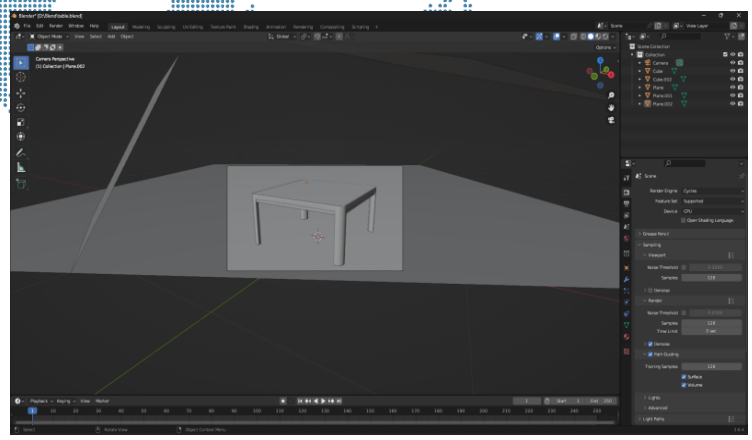
Dalam dunia komputer grafik, MIP Mapping atau yang biasa disebut dengan pemetaan MIP. Mipmapping adalah teknik dalam pemrosesan Gambar di mana Gambar atau peta tekstur asli yang memiliki resolusi tinggi diolah dengan filter, lalu diturunkan skalanya menjadi beberapa peta tekstur dengan resolusi yang lebih rendah. Kumpulan peta tekstur dengan resolusi yang berbeda-beda ini disimpan dalam satu berkas tekstur yang sama. [15]. Peta tekstur yang lebih kecil dibuat berdasarkan Gambar asli, dengan setiap tekstur lebih kecil dari "level" sebelumnya, biasanya setengah dari ukuran resolusi. Jadi, jika tekstur asli (Level 0) berukuran 128×128, Level 1 akan berukuran 64×64, Level 2 akan berukuran 32×32, dan seterusnya.

Teknik pemetaan mip banyak dipakai dalam game komputer 3D, simulator penerbangan, dan sistem pencitraan 3D lainnya untuk penyaringan tekstur, dan perangkat lunak GIS 2D dan 3D [16].

2.4. Pengerjaan Eksperimen

Pada penelitian ini digunakan metode penelitian eksperimen, data input yang digunakan pada penelitian ini yaitu berupa file Gambar objek 3D yaitu dengan

menggunakan objek 3D yang terdapat di lingkungan sekitar, kemudian objek 3D akan dibuat di sebuah software bernama Blender, di software blender akan diterapkan settingan Ray tracing dan algoritma MC MC serta Mipmapping yang terlihat pada Gambar 3.



Gambar 3. Proses Perancangan Objek 3D Yang Akan Digunakan Dalam Eksperimen

Gambar 3 menunjukkan proses perancangan objek 3D yang akan digunakan dalam eksperimen, sebagai contoh kali ini objek 3D yang akan dirancang adalah meja. Pada software blender objek akan dibuat sebisa mungkin menyerupai aslinya, baik dari warna maupun bentuk objek. Setelah objek 3D dibuat, pengaturan Ray Tracing dan Algoritma monte carlo diterapkan, di blender terdapat metode rendering cycle yang mendukung ray tracing dan algoritma monte carlo, seperti jumlah sampel yang digunakan dalam efek pencahayaan, jumlah pantulan sinar, arah cahaya dan kamera serta jumlah kontribusi cahaya yang diterima pada setiap pixel objek.

2.5. Peak Signal Noise Rasio (PSNR)

Untuk membandingkan efektifitas dari rendering menggunakan ray tracing biasa dan rendering dengan algoritma monte carlo dapat dilakukan dengan melihat hasil rendering secara visual, waktu yang digunakan untuk merender objek, dan memori yang digunakan saat komputasi. Namun itu saja tidak cukup oleh karena itu, digunakan tools yaitu PSNR atau Peak Signal to noise rasio [17]. Peak Signal to Noise Ratio atau PSNR merupakan metode yang digunakan dalam penelitian ini untuk mengukur perbandingan kualitas citra antara sebelum dan setelah proses steganografi[18]. Semakin besar nilai PSNR mengindikasikan bahwa kualitas sebuah Gambar baik dan sebaliknya. PSNR adalah salah satu metrik yang digunakan untuk mengukur kualitas visual dari sebuah sinyal atau Gambar yang telah mengalami kompresi atau pemrosesan [19]. Pada eksperimen akan digunakan program kompresi perbandingan Gambar, yang bisa diakses pada website alma-technologies.com.

Gambar 4. Proses Input File objek 3D kedalam program PSNR

Bisa dilihat pada Gambar 4, file objek 3D yang sudah dibuat akan diinputkan ke program aplikasi PSNR, sebagai contoh diambil file MejadenganMCMC.png, selanjutnya menentukan rasio kompresi pada Gambar, rasio yang digunakan adalah 5.0, sebelum menjalankan program akan diminta untuk mengisi kode verifikasi terlebih dahulu, kemudian program akan menghitung jumlah noise yang terdapat pada Gambar 3D hasil rendering dengan Ray tracing biasa dan Ray tracing dengan menggunakan algoritma monte carlo.

Gambar yang direkonstruksi setelah dekompresi dikembalikan untuk pemeriksaan visual, serta rasio kompresi yang dicapai, metrik PSNR akan muncul sebagai output programnya. Output program berupa nilai PSNR bisa dilihat pada Gambar 5.

Gambar 5. Output Program PSNR

Pada Gambar 5 menunjukkan bahwa nilai PSNR yang dihasilkan dari program sebesar 53,33 dB, nilai ini nantinya akan digunakan untuk membandingkan efektivitas antara Ray tracing biasa dan Ray tracing yang menggunakan algoritma monte carlo, skala abu-abu digunakan sebagai output program unutk mempermudah kompresi sambil mengurangi kebutuhan penyimpanan dan waktu komputasi.

3. Hasil Dan Pembahasan

3.1. Hasil Rendering

Berikut merupakan hasil rendering software blender menggunakan Ray tracing biasadan Ray tracing yang menggunakan algoritma monte carlo.

Tabel 1. Hasil pengujian rendering

Wakt u Rend ering ray tracing biasa	Waktu Render ing Ray tracing dengan MCMC	Memori yang digunakan MCMC	Memori yang digunakan MCMC	Ukuran Gambar (Panjang x Lebar)	Jumlah sampel MCMC
0.33 detik	23,37 detik	33.13 mb	13.35 mb	1920px x 1080px	32 Sampel
2,18 detik	2 menit 29 detik	120.05 mb	40,95 mb	1920px x 1080px	32 Sampel
2,37 detik	1 menit 6 detik	145.1 mb	131,25 mb	1920px x 1080px	32 Sampel
1,95 detik	1 menit 21 detik	125,37 mb	46,27 mb	1920px x 1080px	32 Sampel
43,5 detik	2 menit 35 detik	121,3 mb	45,2 mb	1920px x 1080px	32 Sampel
1,68 detik	1 menit 48 detik	92.10 mb	41.47 mb	1080px x 1080px	32 Sampel
5,15 detik	12 menit 6 detik	506 mb	275 mb	1920px x 1080px	32 Sampel
2,70 detik	3 menit 18 detik	132,4 mb	44,6 mb	1920px x 1080px	32 Sampel
1,16 detik	1 menit 25 detik	104,1 mb	54,2 mb	1080px x 1080px	32 Sampel
0,66 detik	43,17 detik	57,78 mb	46,8 mb	1920px x 1080px	32 Sampel
0,53 detik	31,47 detik	42,6 mb	22,7 mb	1920px x 1080px	32 Sampel
2,7 detik	1 menit 38 detik	76,3 mb	40,95 mb	1920px x 1080px	32 Sampel
34,3 detik	3 menit 2 detik	106,2 mb	67.56 mb	1920px x 1080px	32 Sampel
1,45 detik	2 menit 47 detik	121,7 mb	42.17 mb	1920px x 1080px	32 Sampel
6,54 detik	55,9 detik	107,5 mb	104,6 mb	1080px x 1080px	32 Sampel

Hasil rendering bisa dilihat pada Tabel 1. Gambar hasil rendering menunjukkan bahwa secara kasat mata kedua Gambar hasil rendering dengan ray tracing biasa dan ray tracing dengan MCMC terlihat sama, akan tetapi ada beberapa perbedaan yang terlihat jika dilihat dengan teliti, jika dilihat dengan seksama, Gambar yang dihasilkan dari rendering ray tracing dengan MCMC lebih baik dibandingkan dengan ray treacing biasa, walaupun waktu rendering dan memori yang digunakan dalam rendering untuk MCMC lebih banyak dibandingkan dengan rendering ray tracing biasa, tanpa mengubah resolusi Gambar, penggunaan jumlah sampel dengan MCMC membuat hasil rendering dengan MCMC menghasilkan Gambar yang lebih baik dibandingkan dengan rendering ray tracingbiasa.

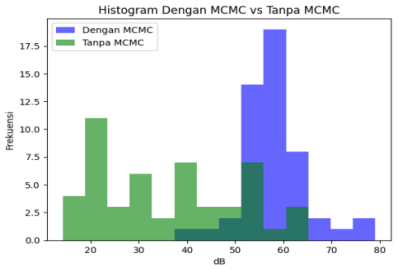
3.2. Hasil Pengujian PSNR

Hasil output program penghitung nilai PSNR dari 50 sampel objek 3D yang sudah direnderdapat dilihat melalui Tabel 1.

Tabel 2. Hasil Pengujian PSNR

Objek	Dengan MCMC	Tanpa MCMC	Objek	Dengan MCMC	Tanpa MCMC
Batu bata	60.40 dB	22.35 dB	Rubik	57.62 dB	39.54 dB
Vas	58.80 dB	21.40 dB	Bowling	58.35 dB	31.89 dB
Piala	59.65 dB	57.49 dB	Handsanitizer	60.03 dB	19.47 dB
Lemari	52.85 dB	42.35 dB	Gelas bening	59.26 dB	20.81 dB
Kacamata	53.16 dB	14.46 dB	Gramophone	57.73 dB	27.10 dB
Kaleng	62.26 dB	44.82 dB	Balon Udara	66.24 dB	63.44 dB
Kursi	58.51 dB	14.92 dB	Buku	59.40 dB	35.58 dB
Mug	53.39 dB	39.20 dB	TV	53.23 dB	53.28 dB
Meja	53.33 dB	21.52 dB	Pohon	41.83 dB	19.19 dB
Piring	79.08 dB	51.44 dB	Garpu	62.94 dB	63.99 dB
Lampu Tidur	62.18 dB	40.74 dB	Mata	52.04 dB	37.38 dB
Donat	51.04 dB	29.51 dB	Catur	57.97 dB	46.80 dB
Dadu	59.00 dB	31.78 dB	Teko	59.78 dB	37.48 dB
Mug set	53.31 dB	34.61 dB	Microphone	56.93 dB	20.52 dB
Sofa	53.31 dB	40.42 dB	Water tank	53.52 dB	53.05 dB
Kardus	61.24 dB	37.44 dB	Roket	64.30 dB	55.37 dB
Kotak Kado	61.97 dB	22.27 dB	Peluit	58.24 dB	20.54 dB
Bunga Kaktus	53.18 dB	21.65 dB	Wastafel	76.59 dB	32.16 dB
Speaker	60.56 dB	21.39 dB	Patung	63.76 dB	61.10 dB
Boneka	50.09 dB	24.23 dB	Sepeda	53.12 dB	53.23 dB
Bola basket	55.64 dB	29.16 dB	Daging	57.30 dB	23.85 dB
Apel	53.75 dB	53.74 dB	Bola Sepak	45.10 dB	29.97 dB
Ikan Hiu	59.64 dB	45.47 dB	Kompas	73.82 dB	50.56 dB
Lampu jalan	65.49 dB	48.18 dB	Lentera	57.91 dB	14.21 dB
Springbed	53.24 dB	53.52 dB	Kapak	59.60 dB	17.29 dB

Sebagai contoh objek 3D yaitu kapak dari hasil rendering dengan algortima monte carlo memiliki nilai PSNR sebesar 59.60 dB, sedangkan ray tracing biasa memiliki nilai PSNR sebesar 17.29 dB. Dari 50 sampel objek 3D yang diuji, 47 pengujian penggunakan MCMC berhasil dan 3 diantaranya gagal, dengan ini presentase keberhasilan pengujian yang didapat sebesar 94%. Hal ini menunjukkan bahwa Gambar hasil rendering ray tracing dengan mengggunakan algoritma monte carlo memiliki tingkat distorsi yang sangat rendah atau kesalahan yang sangat kecil dibandingkan dengan Gambar hasil rendering ray tracing biasa. Dalam konteks PSNR, semakin tinggi nilai dB, semakin kecil distorsi atau kesalahan antara Gambar yang diproses dengan Gambar asli [20]. Perbandingan antara ray tracing biasa dan ray tracing yang menggunakan algortima montecarlo bisa terlihat pada histogram 1 berikut.



Gambar 6. Histogram perbandingan jumlah noise dengan MCMC dan tanpa MCMC

Gambar 6 merupakan histogram yang merepresentasikan visual dari distribusi frekuensi dari dua set data yang berbeda, yaitu "Dengan MCMC" dan "Tanpa MCMC". Sumbu-x (sumbu horizontal) menunjukkan rentang nilai dari data dalam dB. Sumbu-y (sumbu vertikal) menunjukkan frekuensi jumlah data dalam setiap rentang nilai tersebut. Tinggi setiap bar pada histogram mewakili jumlah data dalam rentang nilai tertentu. Semakin tinggi bar, semakin banyak data yang berada dalam rentang nilai tersebut.

4. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dari 50 sampel objek 3D yang diuji, 47 pengujian penggunakan MCMC berhasil dan 3 diantaranya gagal, dengan ini presentase keberhasilan pengujian yang didapat sebesar 94%. hasil penelitian menunjukkan bahwa dengan menggunakan algoritma MCMC, Gambar hasil rendering objek 3D memiliki kualitas yang lebih baik dibandingkan dengan ray tracing biasa. Hal ini ditunjukkan oleh tingkat distorsi atau kesalahan yang rendah pada objek yang diproses. Penggunaan algoritma MCMC membantu meningkatkan kinerja ray tracing dengan menangani distribusi probabilitas yang kompleks, sehingga menghasilkan visualisasi yang lebih realistis dalam waktu yang lebih efisien.

Selain itu, integrasi teknologi Mipmapping dengan metode ray tracing dan algoritma MCMC juga berkontribusi dalam meningkatkan kualitas visual objek 3D. Mipmapping memaksimalkan penggunaan tekstur pada objek 3D, sehingga menghasilkan visualisasi yang lebih baik. Dengan demikian, penelitian ini memperlihatkan bahwa integrasi antara ray tracing, MCMC, dan Mipmapping dapat menghasilkan visualisasi objek 3D yanglebih realistis.

Berdasarkan penelitian ini juga diketahui ada beberapa hal yang belum diteliti lebih jauh, seperti pengaruh mesh poligon dalam menghasilkan design 3D, dan pengaruh resolusi Gambar terhadap kualitas atau hasil rendering, oleh karena itu perlu dingat bahwa penelitian ini hanya sebagai pedoman dan tidak bisa dijadikan acuan sepenuhnya dalam mengambil suatu keputusan, dan masih perlu pengembangan dan penelitian lebih lanjut mengenai Ray tracing dan MCMC.

Daftar Pustaka

[1] I. N. B. Hartawan and A. M. Dirgayusari, “Analisis Rendering Video Animasi 3D Menggunakan Aplikasi Blender Berbasis Network Render,” *J. Resist. (Rekayasa Sist. Komputer)*, vol. 1, no. 1, pp. 25–33, 2020, doi: 10.31598/jurnalresistor.v1i1.223.

[2] B. E. Tarazona-Romero, N. Y. Castillo Leon, J. G. Ascanio Villabona, M. A. Duran Sarmiento, P. Hulse, and C. G. Cardenas Arias, “Performance evaluation of a parabolic cylinder collector applying the Monte Carlo ray tracing method,” *Sustain. Eng. Innov.*, vol. 6, no. 1, pp. 3–16, 2024, doi: 10.37868/sei.v6i1.id226.

[3] A. Keller, C. Wächter, and N. Binder, “Quasi-Monte Carlo Algorithms (not only) for Graphics Software,” pp. 1–18, 2023, [Online]. Available: http://arxiv.org/abs/2307.15584

[4] M. Ula, “Realistic Texturing pada Objek 3-dimensi Menggunakan Model Tehnik Texture Mapping,” *J. Arsitekno*, vol. 6, no. 6, p. 12, 2021, doi: 10.29103/arj.v6i6.1217.

[5] I. Wald, “Realtime Ray Tracing and Interactive Global Illumination,” *IT - Inf. Technol.*, vol. 48, no. 4, pp. 242–245, 2020, doi: 10.1524/itit.2006.48.4.242.

[6] R. W. Jens Krüger, “‘Efficient Ray Tracing of Volume Data Using Multi-Dimensional MIP Mapping,’” *IEEE Trans. Vis. Comput. Graph.*, vol. 4, pp. 147–173, 2023.

[7] R. Ghaffari, P. Abdi, A. Moghaddasi, S. Heidarzadeh, H. Ghahvhechian, and M. Kasiri, “Ray Tracing versus Thin-Lens Formulas for IOL Power Calculation Using

Swept-Source Optical Coherence Tomography Biometry," *J. Ophthalmic Vis. Res.*, vol. 17, no. 2, pp. 176–185, 2022, doi: 10.18502/jovr.v17i2.10788.

[8] N. I. Ziedan, "Urban positioning accuracy enhancement utilizing 3D buildings model and accelerated ray tracing algorithm," *30th Int. Tech. Meet. Satell. Div. Inst. Navig. ION GNSS 2017*, vol. 5, no. September, pp. 3253–3268, 2017, doi: 10.33012/2017.15366.

[9] V. Giangaspero, V. Sharma, J. Laur, J. Thoemel, S. Poedts, and A. Lani, "Ray Tracing Analysis of High Frequency Communication System in Inductively Coupled Plasma Facility," 2022, doi: 10.13009/EUCASS2022-6181.

[10] N. Hikmah, M. Fachri, and R. Darmawan, "Visualization of Real-World 3D Reconstructed Objects with Real-Time Ray Tracing on Ampere Architecture Graphic Processing Unit," vol. 5, no. 36, pp. 2039–2045, 2021.

[11] M. A. Bin Misran, A. Bilgaiyan, and R. Hattori, "Optical Ray Tracing Simulation by Using Monte Carlo Method for Reflectance-based Photoplethysmography Sensor in Human Skin and Fingertip Model," *Comput. Exp. Res. Mater. Renew. Energy*, vol. 5, no. 2, p. 78, 2022, doi: 10.19184/cerimre.v5i2.31668.

[12] M. I. Disney, P. Lewis, and P. R. J. North, "Monte Carlo ray tracing in optical canopy reflectance modelling," *Remote Sens. Rev.*, vol. 18, no. 2, pp. 163–196, 2020, doi: 10.1080/02757250009532389.

[13] M. A. Bin Misran, A. Bilgaiyan, and R. Hattori, "Optical Ray Tracing Simulation by Using Monte Carlo Method for Reflectance-based Photoplethysmography Sensor in Human Skin and Fingertip Model," *Comput. Exp. Res. Mater. Renew. Energy*, vol. 5, no. 2, p. 78, 2022, doi: 10.19184/cerimre.v5i2.31668.

[14] O. González, S. Rodríguez, R. Pérez-Jiménez, B. R. Mendoza, and A. Ayala, "Comparison of Monte Carlo ray-tracing and photon-tracing methods for calculation of the impulse response on indoor wireless optical channels," *Opt. Express*, vol. 19, no. 3, p. 1997, 2021, doi: 10.1364/oe.19.001997.

[15] P. Mykhaylov, R. Y. Chekhmestruk, O. N. Romanyuk, and S. V. Kotlyk, "MIP mapping the virtual environment for computer games," *Autom. Technol. Bus. Process.*, vol. 13, no. 4, pp. 34–39, 2022, doi: 10.15673/atbp.v13i4.2220.

[16] W. C. Park, D. S. Kim, J. S. Park, S. D. Kim, H. S. Kim, and T. D. Han, "The design of a texture mapping unit with effective MIP-map level selection for real-time ray tracing," *IEICE Electron. Express*, vol. 8, no. 13, pp. 1064–1070, 2021, doi: 10.1587/elex.8.1064.

[17] R. Humayrah, A. M. Elhanafi, and M. T. Batubara, "Analisa Histogram dan PSNR Pada Citra True Color Dalam Pengamanan Teks Menggunakan Spread Spectrum dan LSB Histogram and PSNR Analysis on True Color Image in Text Security Using Spread Spectrum and LSB," *J. Ilmu Komput. dan Sist. Inf.*, vol. 2, no. 1, pp. 188–200, 2022.

[18] H. Sajati, "The Effect of Peak Signal to Noise Ratio (PSNR) Values on Object Detection Accuracy in Viola Jones Method," *Conf. Senat. STT Adisutjipto Yogyakarta*, vol. 4, 2022, doi: 10.28989/senatik.v4i0.139.

[19] U. Sara, M. Akter, and M. S. Uddin, "Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study," *J. Comput. Commun.*, vol. 07, no. 03, pp. 8–18, 2020, doi: 10.4236/jcc.2019.73002.

[20] V. I. Ungureanu, P. Negirla, and A. Korodi, "Image-Compression Techniques: Classical and 'Region-of-Interest-Based' Approaches Presented in Recent Papers," *Sensors*, vol. 24, no. 3, 2024, doi: 10.3390/s24030791.