

Implementasi Algoritme *Binary Space Partition* untuk Pembuatan Peta Prosedural pada Aplikasi Permainan *Roguelike Guns Blazing*

Ignatius Dwiki Iskandar, Imam Kuswardayan, dan Hadziq Fabroyir
Departemen Informatika, Institut Teknologi Sepuluh Nopember (ITS)
e-mail: imam@its.ac.id

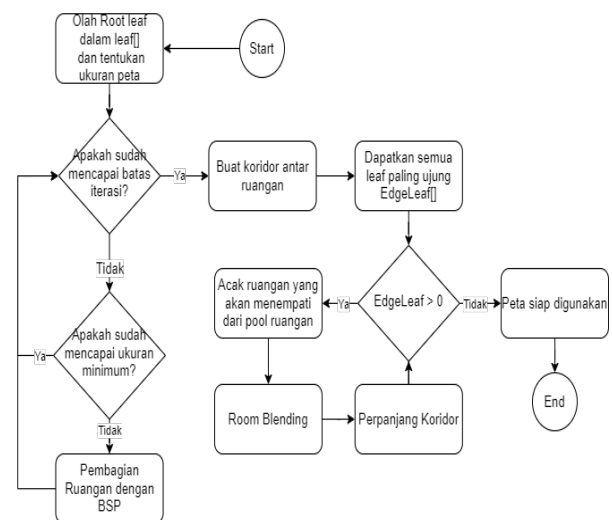
Abstrak—Roguelike adalah sebuah genre game yang merupakan turunan dari genre role-playing games atau yang biasa disebut RPG. Roguelike biasanya memiliki sistem level yang diolah secara prosedural dan sistem permanent death yang berarti pemain harus mengulang dari titik awal yang telah ditentukan sebelumnya secara fresh. Maka dari itu, pemain bisa dibilang dipaksa untuk mempelajari sistem dari game ini melalui kegagalan dan berkembang menjadi pemain yang lebih baik dari sebelumnya. Sistem pembangkit peta dari sebuah permainan ada bermacam-macam, tergantung dengan kebutuhannya. Salah satu metode yang digunakan dalam pembangkitan peta ini adalah Binary Space Partition. Metode ini bekerja untuk membelah sebuah ruangan menjadi ruangan-ruangan yang lebih kecil lagi, dimana pembelahan yang dilakukan ini adalah secara acak. Maka dari itu ruangan tersebut akan memiliki tempat yang terdistribusi dan tidak tumpang tindih antar ruangan. Game yang ingin dibangun ini bernama Guns Blazing, yaitu game roguelike yang pembangkit petanya menggunakan metode Binary Space Partition ini. Namun, sebelum bisa dimainkan, game harus memilih ruangan yang akan diisi kedalam tempat yang sudah disediakan oleh algoritme dan melakukan beberapa perubahan kepada interior ruangan agar koneksi antar ruangan dapat terhubung dengan baik. Uji coba yang dilakukan pada pembangkit peta didalam game Guns Blazing ini bisa dibilang cepat. Dalam satu kali pembuatan, waktu yang diperlukan hanya sekitar 31 milidetik saja. Waktu ini sangat penting agar pemain tidak lama menunggu loading dari game. Dalam uji coba lainnya, didapati bahwa setiap kali pemain memainkan game, peta yang ada selalu berubah yang sangat penting untuk keperluan replayability dari game ini.

Kata Kunci—Binary Space Partition, Game, Procedural Generation, Roguelike.

I. PENDAHULUAN

PADA akhir tahun 2019, virus baru bernama COVID-19 muncul dan menyerang kesehatan manusia di seluruh dunia, dengan kasus yang terus bertambah. Karena itu, banyak negara yang melakukan pembatasan kontak antar warganya, dan menerapkan kerja dari rumah, yang biasa disebut Work From Home (WFH). Semua kegiatan, termasuk pekerjaan kantor hingga dilakukan dari rumah, dengan didukung dengan teknologi internet. Hal ini kemudian memicu kebosanan dan kejenuhan orang karena tidak bisa mendapatkan sosialisasi serta entertainment yang biasa didapatkan jika bisa keluar rumah dengan bebas. Karena itu, banyak orang yang beralih memainkan game untuk kebutuhan entertainment-nya itu [1].

Hal diatas bisa dilihat dengan penjualan video game AS ke rekor \$56,9 miliar pada tahun 2020, meningkat 27% dibandingkan tahun sebelumnya. Dalam hal peningkatan user engagement, di Asia-Pasifik mengalami peningkatan 42%



Gambar 1. Flowchart Pembangkitan Peta Secara Umum.

waktu bermain game. Hal ini bisa menjadi kesempatan yang baik untuk melakukan pembuatan permainan, dimana pasarnya sedang naik daun. Salah satu genre yang sedang populer beberapa tahun ini adalah roguelike. Dimana biasanya pemain harus belajar bermain lebih baik untuk memenangkan permainan.

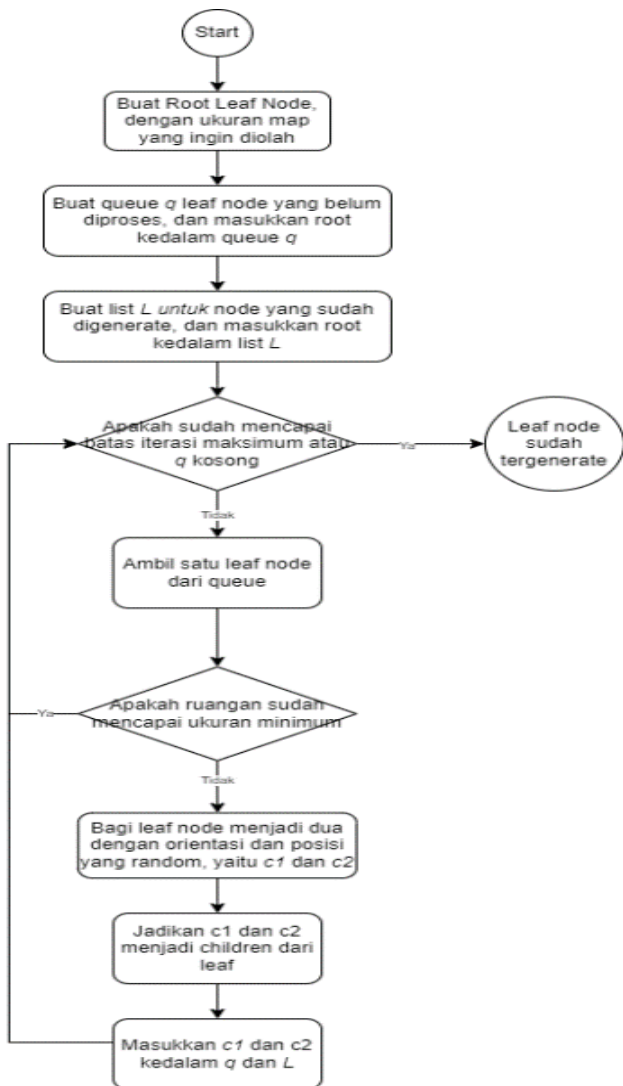
Salah satu tantangan dari permainan roguelike adalah pembuatan peta proseduralnya, yang merupakan keunggulan genre ini untuk meningkatkan nilai replayability. Banyak algoritme yang bisa membantu dalam pembuatan peta di dalam permainan dan bahkan bisa digabungkan untuk membuat peta yang lebih bagus lagi. Tetapi tidak semua permainan memiliki algoritme yang sama dan pemakaiannya tergantung dari bagaimana peta yang ingin dicapai. Algoritme tersebut antara lain adalah Binary Space Partition, Cellular Automata, Drunkard Walk, Dijkstra Map, Perlin Noise, dan lain-lain.

Topik tugas akhir ini mengacu pada penggunaan algoritme Binary Space Partition dalam implementasi pembuatan peta secara prosedural pada aplikasi permainan dengan genre roguelike ini, agar nilai dari replayability dari permainan ini tetap bisa menjadi tinggi, dan menjaga retention dari pemain agar mereka tetap terus kembali kepada permainan ini, serta mendesain permainan sesuai dengan teori Game Design agar dapat memuaskan “fun” yang dicari oleh pemain ketika memainkan permainan ini.

II. TINJAUAN PUSTAKA

A. Procedural Generation

Procedural generation merupakan sebutan untuk



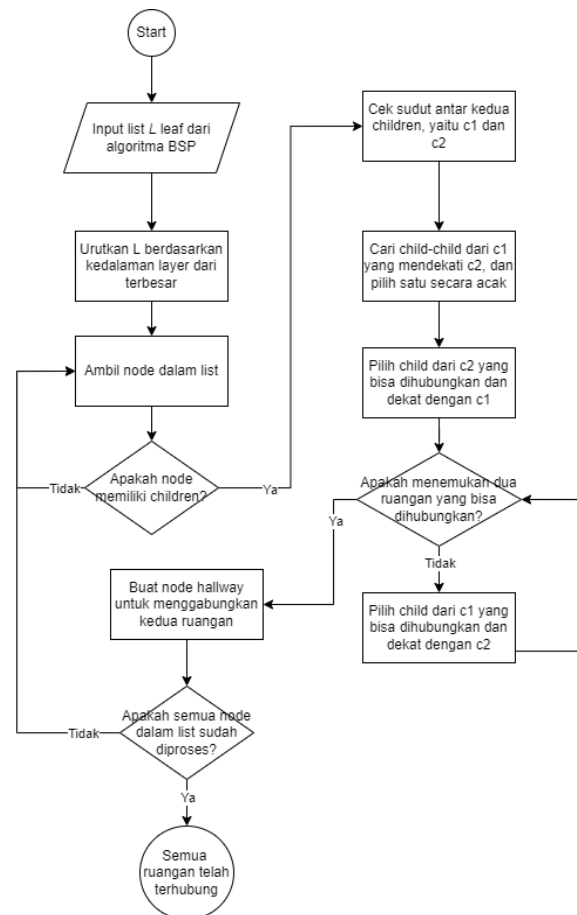
Gambar 2. Flowchart Pembagian Ruang dengan BSP.

pembuatan data oleh komputer. Penggunaannya biasanya adalah untuk membuat konten untuk aplikasi permainan atau film animasi. Procedural generation dapat memberikan banyak kelebihan, misalnya visi yang tak terbayangkan, level yang dapat diputar ulang (replayable), musik reaktif, dan narasi unik yang dipandu pemain. Pemain dapat memiliki perjalanan pribadi mereka sendiri tetapi masih memiliki cukup pengalaman umum untuk berbagi cerita dengan orang lain[2].

B. Binary Space Partition

Binary space partitioning tree adalah data struktur geometrik yang diperoleh dari sebuah skema partisi rekursif, yang disebut binary space partition (BSP) di atas satu set input objek, yaitu ruang dipartisi sepanjang hyperlane menjadi dua setengah ruang, dan kedua setengah ruang tersebut dipartisi secara rekursif sampai setiap subproblem hanya berisi bagian kecil dari input objek[3].

Binary Space Partition mengambil ruang tertentu dan membaginya menjadi dua, dan kemudian mengambil dua area tersebut dan membaginya menjadi dua, dan mengulanginya hingga threshold yang ditentukan telah tercapai. Beberapa threshold yang umum digunakan adalah jumlah pembagian ruang yang dilakukan dan ukuran rata-rata area yang dihasilkan.



Gambar 3. Flowchart Pembuatan Koridor.

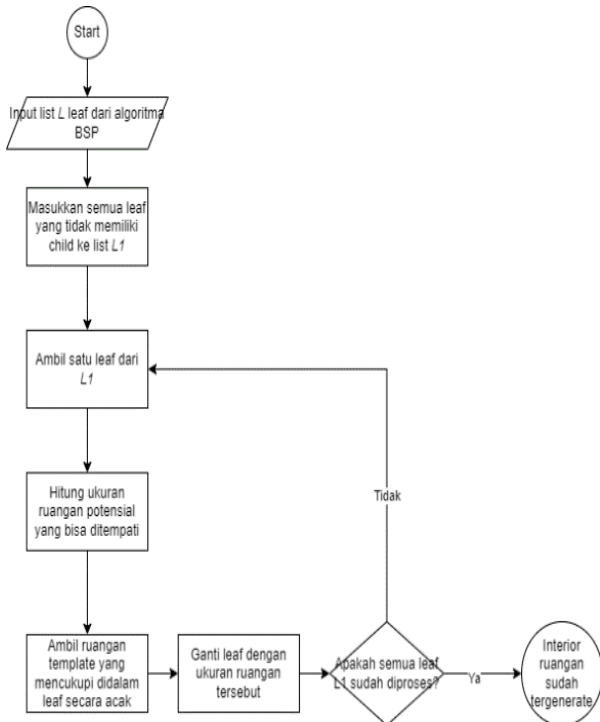
C. Game

Permainan muncul dari keinginan manusia untuk bermain dan dari kemampuan kita untuk berpura-pura. Bermain adalah kategori luas dari aktivitas manusia yang tidak penting, dan biasanya rekreasional, yang seringkali juga signifikan secara sosial. Berpura-pura adalah kemampuan mental untuk membangun realitas imajiner yang orang yang berpura-pura tahu berbeda dari dunia nyata dan yang dapat dibuat, ditinggalkan, atau diubah sesuka hati oleh orang yang berpura-pura. Bermain dan berpura-pura adalah elemen penting dari bermain game. Keduanya telah dipelajari secara ekstensif sebagai fenomena budaya dan psikologis.

Jadi, definisi dari game adalah jenis aktivitas bermain, yang dilakukan dalam konteks realitas pura-pura, di mana para peserta mencoba untuk mencapai setidaknya satu tujuan nontrivial yang sewenang-wenang dengan bertindak sesuai dengan aturan. Elemen penting dari permainan adalah bermain, berpura-pura, tujuan, dan aturan[4].

D. Roguelike

Roguelike adalah salah satu sub-genre dari role-playing games (RPG). Inti dari yang membuat sebuah roguelike adalah game tersebut memiliki level yang dihasilkan secara prosedural. Karakter pemain dalam roguelike mengalami permadeath, yaitu pemain harus mengulang dari titik awal yang telah ditentukan sebelumnya secara fresh. Dalam roguelike, pemain dipaksa untuk mempelajari sistem dari permainan melalui kegagalan. Contoh permainan Roguelike yang terkenal adalah Dead Cells, Hades, Vampire Survivor.



Gambar 4. Flowchart Pengacakan Ruang.



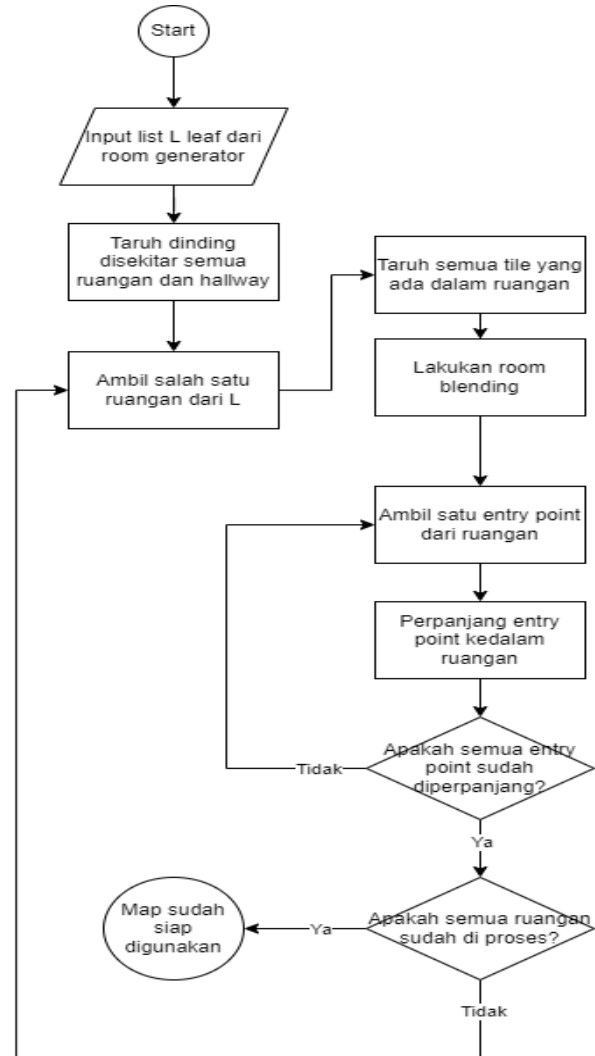
Gambar 5. Tampilan Pemain Menembak Musuh.

E. Game Engine

Konsep mesin permainan cukup sederhana, yaitu ada untuk mengabstraksi detail dalam melakukan tugas-tugas umum yang terkait dengan permainan, seperti rendering, fisika, dan input, sehingga pengembang (artis, desainer, bahkan programmer lain) dapat fokus pada detail yang membuat game mereka unik. Game Engine menawarkan komponen yang dapat digunakan kembali yang dapat dimanipulasi untuk menghidupkan game. Memuat, menampilkan, dan menghidupkan model, deteksi tabrakan antara objek, fisika, input, antarmuka pengguna grafis, dan bahkan bagian dari kecerdasan buatan game semuanya dapat menjadi komponen yang membentuk engine.

F. Unity

Unity adalah mesin game 3D/2D dan IDE lintas platform yang kuat untuk pengembang. Sebagai Game Engine, Unity mampu menyediakan banyak fitur bawaan yang penting yang membuat game berfungsi, seperti fisika, rendering 3D, dan deteksi tabrakan (collision). Developer game dapat menggunakan fitur-fitur tersebut daripada harus memulai proyek baru dengan membuat mesin fisika baru dari awal—menghitung setiap gerakan terakhir dari setiap material, atau cara cahaya memantul dari permukaan yang berbeda.



Gambar 6. Flowchart Pembuatan Environment.

G. C#

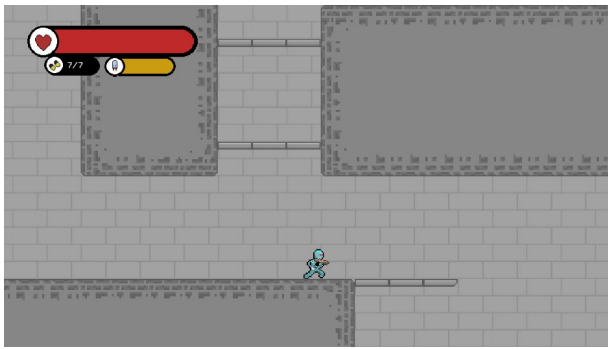
C# adalah bahasa pemrograman berorientasi objek, berorientasi komponen. Programmer akan mendefinisikan jenis dan perilaku mereka. C# menyediakan konstruksi bahasa untuk secara langsung mendukung konsep-konsep ini, menjadikan C# sebagai bahasa alami untuk membuat dan menggunakan komponen perangkat lunak. C# telah menambahkan fitur untuk mendukung beban kerja baru dan praktik desain perangkat lunak yang muncul.

III. PERANCANGAN

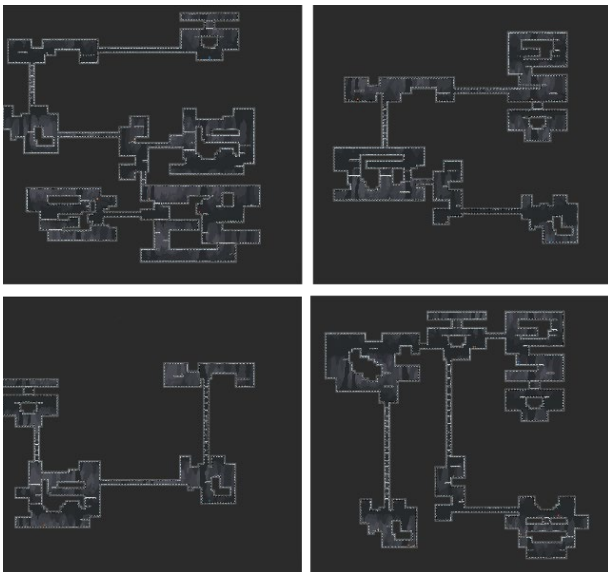
A. Perancangan Algoritme Pembangkitan Peta

Algoritme pembangkitan peta dalam game ini terdiri dari beberapa tahap, yaitu pembagian ruangan dengan Binary Space Partition, pembuatan koridor, pengacakan ruangan, dan pembuatan environment. Tahap-tahap secara umum ini dapat dilihat pada Gambar 1. Flowchart Pembangkitan Peta Secara Umum.

Pada umumnya, metode Binary Space Partition bekerja dengan membelah dua area yang telah disediakan sampai batas tertentu. Prosedur dari algoritme Binary space partition terdiri dari 4 langkah. Langkah pertama yaitu mengambil area, yang dikenal sebagai Leaf, lalu bagi menjadi dua secara horizontal atau vertikal pada titik acak, yang berarti sekarang



Gambar 7. Tampilan Permainan Guns Blazing.



Gambar 8. Beberapa Hasil Pembangkitan Peta.

memiliki dua leaf child. Langkah berikutnya, ulangi proses pembagian untuk masing-masing dari dua leaf child yang baru sampai setiap area cukup kecil, yaitu sub-area terendah memiliki kira-kira ukuran ruangan yang ingin dibuat. Setelah itu, jika se-mua leaf tidak dapat dibelah lagi, tambahkan ruangan secara acak di setiap leaf. Terakhir, gabungkan ruangan-ruangan di leaf yang bersaudara (memiliki parent yang sama) dengan koridor, lalu naik ke leaf induk dan gabungkan salah satu ruangan ke leaf saudara, dan seterusnya sampai mencapai akar dari tree. Flowchart dari proses diatas dapat dilihat pada Gambar 2. Flowchart Pembangian Ruang dengan BSP

Setelah pembagian ruangan, maka ruangan tersebut harus dihubungkan antar satu dengan lainnya. Proses penghubungan tersebut dapat dilihat pada flowchart pada Gambar 3. Flowchart Pembuatan Koridor. Untuk meminimalisir terjadinya duplikasi ruangan yang sama ketika peta dibangkitkan, maka harus ada sistem yang mengaturnya. Pengacakan ruangan dilakukan dengan prioritas, yang berarti ruangan yang paling sedikit muncul akan dicek terlebih dahulu apakah bisa diletakkan dalam leaf tersebut. Flowchart pengacakan ruangan ini dapat dilihat pada Gambar 4. Flowchart Pengacakan Ruangan.

Setelah itu akan dilakukan pembuatan dinding dari ruangan, room blending, serta pemanjangan koridor. Pada pembuatan dinding, ruangan akan dikelilingi dengan dinding agar membentuk sebuah ruangan kotak, setelah itu akan dimasukkan tile dari template ruangan yang telah ditentukan sebelumnya. Hal ini akan menimbulkan konflik antar dinding

Tabel 1. Variabel Uji Fungsi Waktu

Variabel	Isi
Ukuran Peta	100x100 unit
Ukuran minimal leaf	30x30 unit
Lebar koridor	1 unit

Tabel 2. Hasil Pengujian Fungsi Waktu

No	Durasi Pembangkitan (ms)
1	27.76527
2	26.13831
3	28.96118
4	53.26843

Tabel 3. Hasil Pengujian Index Keunikan

No	Skala Likert										Persentase
	1	2	3	4	5	6	7	8	9	10	
1							1	3	3	2	87%
2							4	3		2	80%
3						1		1	5	2	88%
4								1	4	4	93%
5						1	2	1	4	1	82%
Rata-rata persentase											86%

bagian luar dan dinding interior dari ruangan, maka dari itu diperlukan room blending. Room blending akan membuat dinding menjadi tersambung dengan interior nya sehingga akan terlihat seperti ruangan yang utuh. Selanjutnya akan dilakukan pemanjangan koridor, karena kemungkinan terdapat koridor yang terhalang oleh interior dari ruangan itu, sehingga harus membuka jalan untuk mencapai ruangan itu. Untuk pemanjangan dilakukan dengan pengeboran interior sampai menemui ruangan terbuka yang bisa dilewati. Flowchart dari proses diatas dapat dilihat pada Gambar 5. Flowchart Pembuatan Environment.

B. Perancangan Fitur dan Mekanik Game

Game ini memiliki empat level yang harus ditempuh pemain untuk mencapai kondisi akhir. Sebelum masuk kedalam area, diawal memulai permainan, pemain akan dipindahkan ke dalam ruangan persiapan. Di dalamnya ada dua NPC yaitu pembuka senjata dan penjual upgrade permanen, sehingga pemain dapat menjadi lebih kuat untuk gameplay loop selanjutnya. Pemain memiliki senjata yang dapat digunakan untuk menembak dan melukai musuh. Proses menembak ini dapat dilihat pada Gambar 6. Tampilan Pemain Menembak Musuh.

Didalam sebuah area, akan ada kemungkinan sebuah ruangan untuk menjadi ruangan treasure. Ruangan ini akan berisi peti (chest) yang jika dibuka akan memberikan pemain sebuah upgrade sementara. Untuk melanjutkan ke area dan level selanjutnya, pemain perlu untuk mencari portal yang terletak diruangan yang jauh dari tempat munculnya pemain. Tampilan permainan ini dapat dilihat pada Gambar 7. Tampilan Permainan Guns Blazing. Setelah itu, pemain dapat berinteraksi dengan portal dan akan dipindahkan ke dalam ruangan penghubung (connector). Didalam ruangan penghubung, pemain dapat memulihkan darahnya dengan berinteraksi dengan healing station dan NPC penjual upgrade sementara untuk memperkuat karakter. Ketika pemain sudah mencapai area terakhir dari sebuah level, maka akan dihadapkan dengan boss pada level tersebut. Setiap level memiliki boss yang berbeda dengan kemampuan yang berbeda pula. Boss ini memiliki darah yang sangat banyak

Tabel 4.
Hasil Uji Coba User

No	Pertanyaan	Rata-rata	Total Rata-Rata
Pertanyaan Asset			
1	Tampilan dari game Guns Blazing menarik untuk dilihat	4.56	4.56
2	Musik dan efek suara yang dimiliki game Guns Blazing cocok dengan game nya	4.56	
Pertanyaan Ketertarikan Dengan Mekanik			
3	Menjelajahi setiap area dan ruangan yang ada	3.78	4.15
4	Mengalahkan setiap musuh dan boss dalam setiap level	4.44	
5	Mendapatkan upgrade, currency, dan menyelesaikan permainan	4.44	
6	Mencoba atau mengkoleksi setiap senjata yang ada	3.78	
7	Penasaran ingin mengetahui kelanjutan level	4.22	
8	Ingin membalas kekalahan dengan melawan kembali	4.22	
Pertanyaan Layout Peta			
9	Peta dalam setiap level dalam game Guns Blazing selalu berubah-ubah	4.89	4.48
10	Ruangan yang ada didalam game Guns Blazing tidak sulit untuk ditelusuri (tidak stuck pada suatu ruangan)	3.89	
11	Pembangkitan layout peta dan ruangan yang bervariasi tiap level membuat game tidak membosankan, lebih menantang dan penasaran	4.67	
Pertanyaan Umum			
12	Game Guns Blazing berjalan dengan baik, tidak menimbulkan lag atau hang	4.89	4.67
13	Tingkat kesulitan dari game Guns Blazing tidak terlalu sulit dan tidak terlalu mudah	4.56	
14	Kontrol dari Game Guns Blazing tidak sulit untuk dimainkan dan tidak membingungkan	4.89	
15	Saat mengakhiri permainan, kalah atau berhasil mengalahkan boss, saya tertarik untuk memainkannya kembali karena adanya skenario dan layout peta yang berbeda dari sebelumnya	4.33	

sehingga lebih sulit untuk dikalahkan dibanding musuh biasa.

Didalam *game* ini, tidak memiliki kondisi menang untuk pemain. Jika karakter pemain mati yang ditunjukkan dengan darahnya yang habis atau berhasil mengalahkan *boss level* terakhir, maka pemain akan langsung dikembalikan ke ruangan fase persiapan. Mata uang koin yang tidak dibelanjakan dan *upgrade* sementara yang telah didapatkan pemain juga akan hilang, namun mata uang *scrap part* dan *upgrade* permanen akan tetap ada walaupun karakter pemain mati berulang kali.

IV. UJI COBA EVALUASI

Gambar 8. Beberapa Hasil Pembangkitan Peta Menunjukkan beberapa hasil dari pembangkitan peta yang dihasilkan oleh algoritme, dengan variabel yang sama. Dapat dilihat bahwa setiap kali membangkitkan sebuah peta, maka akan berbeda dari yang lainnya. Untuk memperlihatkan kinerja dan evaluasi hasil pembangkitan peta tersebut, maka diadakan beberapa uji coba. Uji coba sistem game ini akan dibagi menjadi dua pengujian, yaitu uji coba objektif dan subjektif sistem.

A. Uji Coba Objektif Sistem

Untuk skenario pengujian objektif sistem, akan dilakukan uji coba dengan menggunakan fungsi waktu dan penilaian index keunikan. Untuk pengujian fungsi waktu, variabel yang digunakan dapat dilihat pada Tabel 1. Variabel Uji Fungsi Waktu. Uji coba ini akan dijalankan sebanyak 10 kali dengan hasil yang dapat dilihat pada Tabel 2. Hasil Pengujian Fungsi Waktu.

Uji coba index keunikan akan dilakukan dengan membandingkan 2 buah peta yang dibangkitkan dengan variabel dan lingkungan yang sama sebanyak 5 kali. Peta tersebut kemudian akan dinilai oleh user yang akan memainkan game. Penilaian ini memiliki penilaian skala likert dengan jangkauan dari 1 sampai dengan 10, dengan 1 yaitu sangat mirip dan 10 yaitu sangat berbeda. Hasil pengujian index keunikan ini dapat dilihat pada Tabel 3.

Hasil Pengujian Index Keunikan.

Setelah hasil tersebut dijumlah dan dirata-rata, ternyata hasil yang didapat cukup tinggi, yaitu 86%. Dari hasil uji coba index keunikan tersebut, dapat diketahui bahwa *user* yang telah mencoba *game* dan memberikan penilaian menganggap peta-peta yang dibangkitkan oleh algoritme cukup berbeda satu dengan yang lainnya. Dengan ini, dapat dikatakan bahwa pembangkitan peta dengan algoritme *binary space partition* dapat membuat peta yang acak setiap kali membangkitkan peta untuk setiap sesi permainan.

B. Uji Coba Subjektif Sistem

Untuk pengujian sistem secara subjektif adalah pengambilan pendapat oleh para user yang akan memainkan game ini. Uji coba ini akan dilakukan oleh sembilan user yang dipilih, yang akan memainkan game dan kemudian mengisi kuisioner yang sudah disediakan. User yang dipilih merupakan seorang pemain game, sudah pernah memainkan game roguelike, atau seorang pengembang game. Hasil pengujian ini dapat dilihat pada Tabel 4. Hasil Uji Coba User

Dari hasil uji coba ke beberapa user yang sudah mengisi kuisioner, dapat dilihat bahwa skor yang didapatkan cukup tinggi dari semua aspek dari game. Namun ada beberapa poin yang perlu diperhatikan disini, yaitu pertanyaan nomor 3, 6, dan 10. Setelah dianalisis dan bertanya kepada user kembali, ternyata mereka tidak ingat mana ruangan yang sudah pernah dimasuki, dan mereka kesulitan untuk menemukan ruangan keluar. Hal ini yang membuat mereka menjadi tidak tertarik kepada poin tersebut. Kebanyakan dari mereka memberi saran untuk menambahkan indikator terhadap exit atau memberitabeltabelkan minimap yang menandakan ruangan mana yang sudah pernah dimasuki.

V. KESIMPULAN/RINGKASAN

Kesimpulan yang didapatkan berdasarkan hasil uji coba yang telah didapat adalah sebagai berikut: (1)Rancangan desain game yang diimplementasikan, dimana pemain harus mengalahkan musuh untuk menghadapi boss, konsep

upgrade dan mengumpulkan senjata beserta konsep perma-death, dapat dikatakan diterima oleh para user yang tertarik kepada game ini. Hal ini ditunjukkan oleh respon dari yang tersedia pada hasil percobaan dengan user:(2)Skenario pada masing-masing level berbeda satu dengan yang lainnya. Peta yang dibangkitkan menggunakan Binary Space Partition ini bisa dibidang cukup acak untuk penggunaan pembangkit peta prosedural, terlihat dari pengisian uji coba index keunikan dengan rata-rata sebesar 86% berbeda.(3)Pembuatan ruangan pada setiap level dilakukan secara manual atau hand-made. Hal ini dilakukan untuk menjaga konsistensi dan kemudahan pemain dalam menelusuri peta yang dibuat. Namun layout peta dan pengacakan ruangan tetap dilakukan sehingga setiap kali bermain peta dapat berganti-ganti, sehingga mendapatkan nilai replayability yang tinggi.(4)Peta dari Game Guns Blazing dapat dibangkitkan dengan menggunakan algoritme Binary Space Partition sebagai algoritme utamanya, dengan memakan waktu yang sangat singkat hanya dengan rata-rata waktu 31ms saja per area;(5)Pembangkitan peta dengan Binary Space Partition dapat dibentuk dengan dibantu dengan beberapa tahapan,

yaitu pengacakan ruangan, room blending, serta penghubungan ruangan dengan koridor. Saran yang diberikan dari hasil uji coba, evaluasi, serta fitur yang belum sempat terimplementasi pada tugas akhir ini adalah sebagai berikut(1)Menambahkan minimap atau desain kamera yang lebih baik, agar memudahkan pemain dalam permainan.;(2)Menambahkan variasi upgrade yang lebih banyak lagi;(3)Menambahkan lebih banyak variasi senjata dan objek yang terdapat pada suatu ruangan.

DAFTAR PUSTAKA

- [1] D. Sener, T. Yalcin, and O. Gulseven, "The Impact of COVID-19 on the video game industry," *SSRN Electronic Journal*, pp. 1–13, Jan. 2021, doi: 10.2139/ssrn.3766147.
- [2] T. X. Short and T. Adams, *Procedural Generation in Game Design*. Boca Raton: aylor & Francis, CRC Press, 2017.
- [3] H. Wolverson, *Hands-On Rust: Effective Learning Through 2D Game Development and Play*. Raleigh, North Carolina : Pragmatic Bookshelf, 2021.
- [4] E. Adams, *Fundamentals of Game Design*, Third edition., vol. 77, no. 22. Berkeley California: New Riders, 2014.