

Penggunaan Pohon Biner dalam Binary Space Partition untuk Membuat Dungeon Game Roguelike RPG

Cliff Jonathan 13513044¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

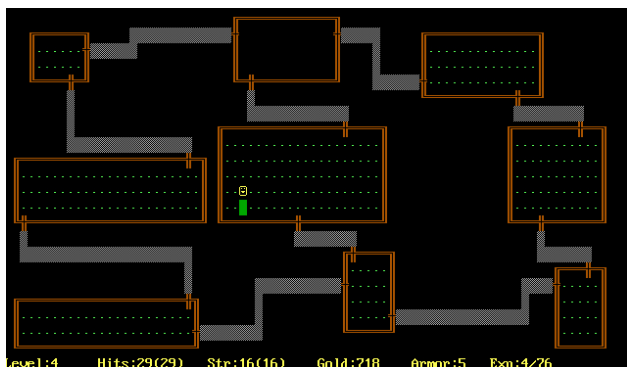
¹13513044@std.stei.itb.ac.id

Abstract—Dalam membuat sebuah *game*, diperlukan algoritma yang tepat untuk mekanik dari *game* yang ingin dibuat. Salah satunya adalah dalam pembuatan *game Roguelike RPG*. Supaya pembuatan *dungeon* di dalam *game Roguelike RPG* bisa dilakukan dengan mudah dan tepat, maka digunakanlah *Binary Space Partition* yang memanfaatkan pohon biner dalam pengerjaannya. Makalah ini membahas proses penggunaan pohon biner dalam metode *Binary Space Partition* untuk membuat *dungeon* dalam *game Roguelike RPG*.

Keywords—BSP, *dungeon*, generate, pohon biner, random, *Roguelike RPG*.

I. PENDAHULUAN

Pohon biner (*binary tree*) merupakan jenis pohon yang banyak sekali diaplikasikan dalam berbagai bidang. Sistem pakar makhluk hidup, kode Huffman untuk kompresi data, penentuan penyakit berdasarkan gejala-gejalanya, pohon sintaks, pohon pencarian biner, dan lain sebagainya. Dari banyak aplikasi pohon biner itu, yang akan dibahas di makalah ini adalah penggunaan pohon biner dalam *binary space partition* (BSP) untuk membuat peta (*map*) *dungeon* acak pada *game* bertipe *Roguelike RPG*.



Gambar 1.1 Contoh peta *dungeon game* bertipe *Roguelike RPG*^[1]

Peta *dungeon* dalam *game* bertipe *Roguelike RPG* bersifat acak, setiap kali main, peta *dungeon* akan

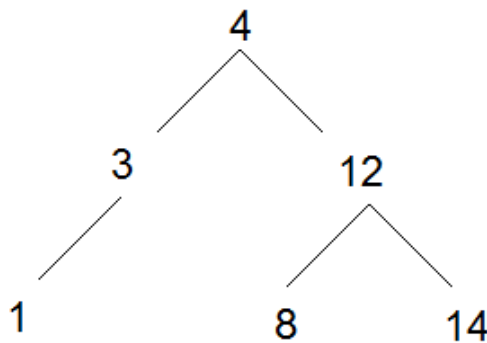
berubah secara acak sesuai dengan algoritma yang dibuat. Pengacakan yang dilakukan adalah pengacakan posisi ruangan, ukuran ruangan, jumlah ruangan, dan koridor yang menghubungkan antara satu ruangan dengan ruangan yang lainnya.

Pengacakan yang dilakukan adalah dengan mengimplementasikan *binary space partition* (BSP). Mengapa harus dengan *binary space partition*? Pembatan peta *dungeon* secara acak tidak boleh dilakukan dengan cara "murni mengacak". Pengacakan yang dilakukan murni akan mengakibatkan peta *dungeon* menjadi tidak baik, mungkin saja akan terbentuk hanya menjadi sebuah *maze* tanpa ruangan, hanya terbentuk hanya dua ruangan, atau malah terbentuk peta yang terlalu acak di mana beberapa ruangan bisa bertumpang tindih.

II. DASAR TEORI

A. Pohon Biner

Pohon adalah graf tak berarah terhubung dan tidak memiliki sirkuit.^[2] Pohon memiliki properti orangtua (*parent*) dan anak (*child*) di setiap simpulnya. Pohon memiliki berbagai variasi dan salah satu variasinya adalah pohon biner.

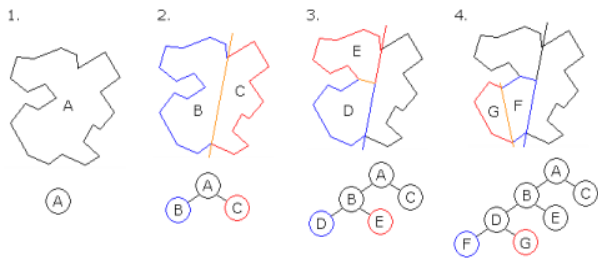


Gambar 2.1 Contoh pohon biner, yaitu pohon pencarian biner.

Pada pohon biner, setiap simpul (*node*) hanya memiliki maksimum dua anak, yaitu anak kiri dan anak kanan.

Pohon biner adalah pohon terurut karena urutan anak-anaknya dapat dibedakan antara anak kiri dan anak kanan.

B. Binary Space Partition Tree (BSP Tree)



Gambar 2.2 Contoh BSP Tree.^[3]

Binary Space Partition Tree (BSP Tree atau pohon BSP) merupakan salah satu terapan dari pohon biner. BSP membagi sebuah bidang menjadi dua dengan menggunakan bidang dengan dimensi kurang satu dari dimensi bidang yang ingin dibaginya secara rekursif. Misalnya, sebuah bidang datar (dimensi dua) dibagi menjadi dua dengan menggunakan garis (dimensi satu), sedangkan sebuah bangun ruang (dimensi tiga) dibagi menjadi dua dengan sebuah bidang datar (dimensi dua). Pembagian direpresentasikan dengan pohon biner.

Adapun hasil pembagiannya dibagi menjadi dua, yaitu yang berada di depan/atas/kiri dan belakang/bawah/kanan (tergantung dimensi dan arah pembagian). Hasil pembagian pertama menjadi anak kiri dari sebuah simpul di pohon biner. Misalnya sebuah bidang datar penuh dibagi menjadi dua, berarti simpul pohon biner paling atas (simpul pertama) memiliki dua anak.

Pembagian terus dilakukan sampai mencapai kondisi basis, atau dalam kata lain, pembagian dilakukan dengan cara rekursif. Kondisi basisnya ditentukan dari tujuan penggunaan BSP ini. Salah satu contoh kondisi basisnya adalah bidang datar yang ingin dibagi memiliki luas kurang dari atau sama dengan sebuah nilai yang ditentukan. Bidang yang memenuhi kondisi basis ini menjadi daun dari pohon BSP.

Pada gambar 3, dapat dilihat proses pembagian sebuah bidang datar (dimensi dua). Pada langkah 1, bidang diberi nama A, dan pohon BSP dibuat dengan simpul paling atas yaitu A. Kemudian, bidang A dibagi menjadi dua yaitu B dan C, sehingga anak B dan C ditambahkan ke simpul A, dengan B menjadi anak kiri A dan C menjadi anak kanan A. Selanjutnya, pada langkah 3 B dibagi menjadi D dan E, dan pada pohon BSP ditambahkan anak D dan E di simpul B. Proses ini berlanjut terus sampai mencapai basisnya, sehingga terbentuk sebuah bidang yang terbagi-bagi sesuai dengan keinginan.

C. Roguelike RPG



Gambar 2.3 Contoh game Roguelike RPG yaitu *Crypt of the NecroDancer*. Dapat dilihat bahwa dalam sebuah *dungeon* terdapat ruangan, koridor, dan monster.^[4]

Role-playing game (RPG) adalah suatu genre *video game* di mana pemain memainkan sebuah karakter dan seolah-olah masuk ke dunia *game* tersebut menjadi karakter yang dimainkannya. Karakter pemain memiliki pengukuran untuk tingkat kehebatannya yaitu pengukuran dari *level* karakter pemain. *Level* ini akan meningkat bila pemain mendapatkan *experience* (pengalaman) yang cukup dari membunuh monster, menyelesaikan misi, dan sebagainya. Peningkatan *level* mengakibatkan pemain menjadi tambah kuat, sehingga *stats* karakter pemain juga akan menjadi lebih baik. *Stats* karakter pemain yang dimaksud bisa berupa HP (*Hit Point/Health Point*) yaitu tingkat kesehatan karakter pemain, STR (*Strength*) yaitu tingkat kekuatan pemain, AGI (*Agility*) yaitu tingkat kelincahan pemain, INT (*Intelligence*) yaitu tingkat kecerdasan karakter pemain, dan lain-lain. Pemain bisa mendapatkan barang-barang untuk keperluannya dalam bertualang, termasuk senjata, pakaian, dan barang-barang lainnya seperti minuman, makanan, dan lain sebagainya yang memiliki fungsinya masing-masing. Setiap perlengkapan yang ada dapat ditentukan kualitasnya berdasarkan dari *stats* tambahannya.

Roguelike RPG adalah salah satu subgenre *role-playing game* (RPG) di mana pemain memulai petualangannya di sebuah *dungeon*, dimulai dari lantai paling atas menuju ke bawah. Peta dalam Roguelike RPG dibuat secara acak, sehingga setiap kali pemain memulai permainan yang baru, maka peta *dungeon* tidak akan sama seperti pada permainan yang sebelumnya. Peta *dungeon* di Roguelike RPG dibentuk dari ruangan-ruangan dan koridor-koridor. Koridor menghubungkan antara suatu ruangan dengan ruangan lainnya. Monster bisa berada di manapun di peta, tetapi barang-barang dan tangga menuju ke lantai berikutnya hanya berada di ruangan.

Hal yang unik dari Roguelike RPG adalah adanya fitur *permanent death*. Fitur ini berarti bila karakter pemain mati di tengah permainannya (status HP pemain bernilai 0), maka karakter tersebut mati untuk selamanya, dan pemain

akan memainkan karakter baru lagi ketika memulai permainan yang baru. Karakter yang baru berarti level karakter kembali menjadi 1, *stats* yang terdistribusi kembali seperti semula ketika karakter sebelumnya dibuat, dan tidak ada barang-barang yang dimilikinya. Karakter pemain yang baru juga akan kembali memulai petualangannya dari lantai atas lagi, dengan peta *dungeon* yang berbeda dengan yang sebelumnya.

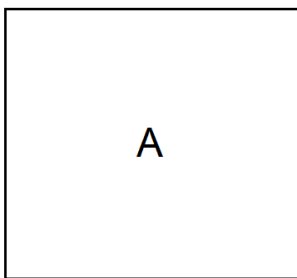
Dalam *Roguelike RPG*, pemain menjelajahi *dungeon* sedalam mungkin, membunuh monster-monster di perjalanannya, mengambil barang-barang yang dijatuhkan monster, melanjutkan ke lantai berikutnya, sampai akhirnya karakter pemain mati.

Sistem pertarungan dengan monster adalah *turn-based* (bergiliran). Menyerang, berjalan satu langkah, menggunakan barang, dan menggunakan sihir dianggap 1 kali aksi, kemudian monster akan melakukan aksinya, lalu pemain dapat melakukan aksinya kembali.^[5]

III. PEMBUATAN PETA DUNGEON

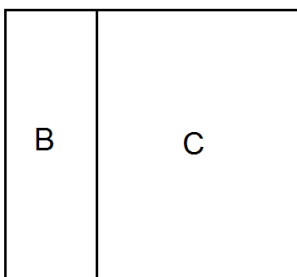
A. Ilustrasi Proses

1. Pembagian peta

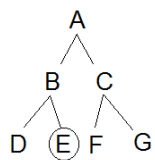
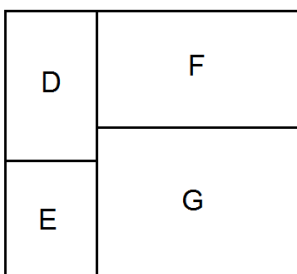


A

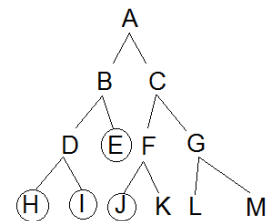
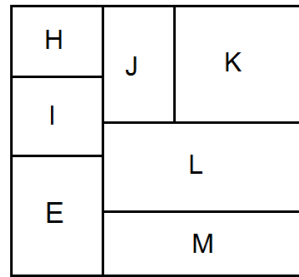
Pada langkah pertama, dibuat pohon BSP dengan akar bernama A, yaitu *dungeon* yang akan dibagi-bagi.



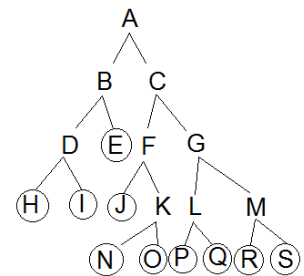
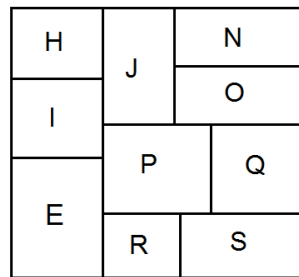
Kemudian, *dungeon* A dibagi menjadi dua *sub-dungeon* yaitu B dan C. A memiliki anak kiri B dan anak kanan C di pohon BSP.



Karena masih memungkinkan untuk dibagi, *sub-dungeon* B dan C masing-masing dibagi. *Sub-dungeon* B dibagi menjadi D dan E, sedangkan *sub-dungeon* C dibagi menjadi F dan G.



Selanjutnya, *sub-dungeon* D dibagi menjadi H dan I, *sub-dungeon* F menjadi J dan K, dan *sub-dungeon* G menjadi L dan M. Misal E tidak bisa dibagi lagi karena bila dibagi, ukuran bidang menjadi lebih kecil dari ukuran bidang minimum yang diinginkan, maka E tidak dibagi dan E merupakan daun (pada gambar dibuat lingkaran di simpul daun).

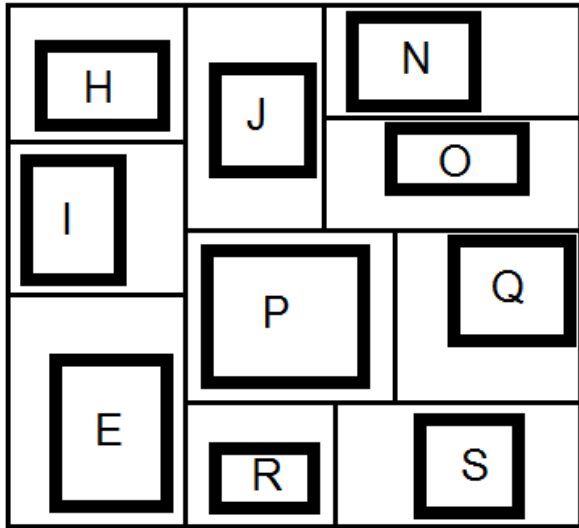


Sub-dungeon H dan I keduanya memiliki ukuran yang kecil dan tidak bisa dibagi lagi (sama kasusnya seperti E, di mana bila dibagi akan menghasilkan bidang yang ukurannya lebih kecil dari batas minimum ukuran bidang), sehingga H dan I tidak dibagi. *Sub-dungeon* yang masih dapat dibagi adalah K, L dan M. *Sub-dungeon* K dibagi menjadi N dan O, *sub-dungeon* L dibagi menjadi P dan Q, sedangkan *sub-dungeon* M dibagi menjadi R dan S.

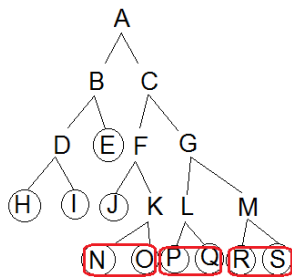
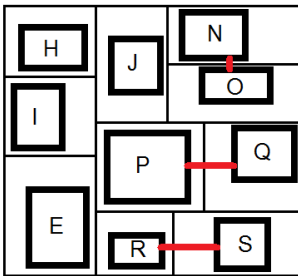
Semua bidang/*sub-dungeon* yang ada di peta sekarang sudah tidak bisa dibagi lagi karena ukurannya sudah cukup kecil, sehingga semua bidang yang tersisa saat ini adalah daun-daun pohon BSP.

2. Pembuatan ruangan

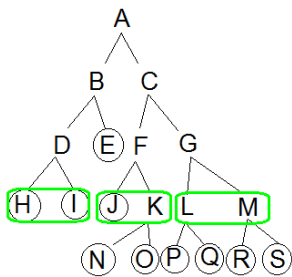
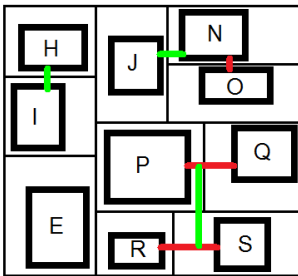
Pembuatan ruangan dilakukan di setiap bidang-bidang yang telah dibagi dengan BSP. Pada gambar, ruangan digambar dengan persegi panjang yang memiliki garis tebal.



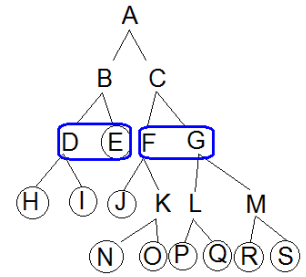
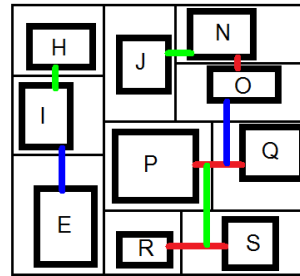
3. Pembuatan koridor



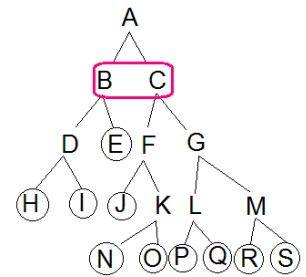
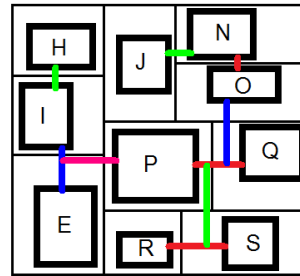
Pertama, pembuatan koridor dilakukan di simpul-simpul dengan *level* paling bawah, yaitu N dan O, P dan Q, serta R dan S. Pada gambar, koridor *sub-dungeon* paling bawah ini dihubungkan dengan garis merah.



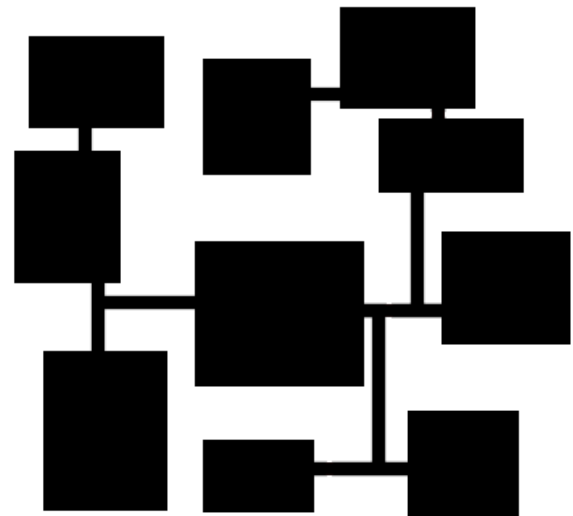
Kemudian, koridor dibuat di *sub-dungeon*/simpul yang memiliki *level* lebih atas dari *sub-dungeon* sebelumnya, sehingga *sub-dungeon* H dan I terhubung. Hal yang sama dilakukan juga di *sub-dungeon* lain yang memiliki *level* yang sama dengan H dan I, yaitu J dan K, serta L dan M. Koridor yang menghubungkan *level* ini ditunjukkan dengan garis berwarna hijau. Perhatikan bahwa koridor penghubung bisa menghubungkan antara salah satu ruangan di *sub-dungeon* yang ingin digabung dengan ruangan lain, dengan koridor lain, atau menghubungkan dua koridor penghubung dari *sub-dungeon* *level* sebelumnya.



Pembuatan koridor dilakukan lagi pada *sub-dungeon* dengan *level* yang lebih atas, yaitu *sub-dungeon* D dan E, serta *sub-dungeon* F dan G. Koridor penghubung digambar dengan garis biru.



Terakhir, dibuat koridor untuk menghubungkan *sub-dungeon* yang merupakan anak kiri dan kanan dari *dungeon* A, yaitu B dan C. Koridor penghubung digambar dengan garis berwarna magenta.



Gambar 3.1 Hasil peta *dungeon*.

B. Penjelasan detail

1. Pembagian peta

Proses pertama dalam pembuatan peta *dungeon* adalah membagi-bagi peta dengan BSP sehingga yang tersisa adalah bidang-bidang datar tempat membuat ruangan *dungeon*. Bidang-bidang datar inilah yang merupakan daun-daun dari pohon BSP tersebut. Setiap pembagian

menghasilkan bidang datar baru, yaitu *sub-dungeon* dari bidang yang dibagi (anak dari simpul pohon yang dibagi).

Setiap pembagian yang dilakukan adalah secara acak, baik posisi garis pembagian dan arah pembagian, namun ada aturan di mana bila pembagian dengan arah yang diinginkan tidak bisa dilakukan karena sudah memenuhi kondisi basis, maka pembagian akan dicoba untuk arah yang lainnya. Misalnya, pengacakan memberi hasil bahwa pembagian harus dilakukan dengan arah horizontal. Tetapi, ternyata pembagian horizontal menyebabkan ukuran bidang hasil pembagian lebih kecil dari atau sama dengan batas minimum ukuran bidang (kondisi basis), maka akan dicoba pembagian dengan arah vertikal. Bila pembagian arah vertikal dapat dilakukan, maka pembagian arah vertikal akan dilakukan. Tetapi, bila pembagian arah vertikal juga sudah tidak bisa dilakukan (hasil pembagian akan menghasilkan bidang yang memiliki ukuran lebih kecil dari batas minimum ukuran bidang), maka bidang datar itu akan menjadi daun pohon BSP.

2. Pembuatan ruangan

Pembuatan ruangan dilakukan dengan membuat sebuah persegi panjang di dalam setiap *sub-dungeon*/bidang-bidang hasil pembagian. Pembuatan ruangan ini meliputi penentuan posisi ruangan secara acak dan ukurannya secara acak. Pengacakan ukuran dan posisi juga tentu dilakukan dengan batasan tertentu. Misalnya, ukuran ruangan minimum memiliki panjang setengah dari panjang bidang hasil bagi, begitu juga dengan lebarnya. Batas ukuran maksimum ruangan tentu adalah ukuran bidang hasil bagi itu sendiri. Batasan pengacakan posisi adalah bahwa dengan ukuran hasil pengacakan, ruangan harus berada di dalam bidang hasil bagi (tidak ada bagian yang melewati garis bidang hasil bagi).

Pada aplikasi nyatanya, di tahap pembuatan ruangan ini juga diletakkan monster-monster, barang-barang, atau tangga menuju ke lantai berikutnya secara acak. Pada beberapa game *Roguelike RPG*, di dalam ruangan juga bisa diletakkan *Non-Playable Character* (NPC) yang berfungsi sebagai penjual barang, pemberi barang, dan lain sebagainya. Posisi peletakan objek-objek ini adalah acak, tetapi harus berada di dalam ruangan, bukan di luar ruangan walaupun berada di dalam bidang hasil pembagian.

3. Pembuatan koridor

Pembuatan koridor bisa dilakukan dengan cara menghubungkan simpul-simpul pohon BSP yang memiliki *level* yang sama secara rekursif. Basis dari fungsi rekursif ini adalah simpul pada level paling bawah, yaitu daun-daunnya (tidak memiliki anak/*sub-dungeon* lagi). Hasil penghubungan ini adalah *sub-dungeon* yang merupakan orangtua dari daun-daun. Kemudian, orangtua dari daun-daun ini juga dihubungkan, sehingga terbentuk *sub-dungeon* dengan *level* yang lebih tinggi. Hal ini terus dilakukan sampai akhirnya semua

Pada implementasinya, ada banyak cara untuk membuat koridor. Pembuatan bisa dengan membuat garis yang menghubungkan koordinat tengah-tengah dari *sub-*

dungeon/bidang yang dibagi, atau melakukan pengacakan terhadap posisi garis yang ingin dibuat.

IV. KESIMPULAN

Pohon biner bisa digunakan dalam membuat peta *dungeon* dalam game *Roguelike RPG* dengan metode BSP. Peta *dungeon* dibagi-bagi menjadi beberapa bidang datar yang lebih kecil untuk tempat meletakkan ruangan *dungeon*, kemudian setiap ruangan dihubungkan dengan koridor yang dibuat dengan menghubungkan simpul-simpul pohon BSP yang memiliki *level* sama.

Penggunaan pohon biner untuk metode BSP dalam membuat *dungeon game Roguelike RPG* memiliki kelebihan yaitu supaya tidak ada ruangan yang bertumpang tindih, atau pembentukan ruangan-ruangan secara acak yang tidak diinginkan.

V. UCAPAN TERIMA KASIH

Kami mengucapkan terima kasih yang sebesar-besarnya kepada Ir. Rinaldi Munir, M.T., selaku dosen mata kuliah Matematika Diskrit yang telah mendidik dan membimbing kami, sehingga kami bisa menyelesaikan makalah ini. Tidak lupa kami mengucapkan terima kasih kepada orangtua dan teman-teman yang telah membantu dan memberi kami dukungan.

REFERENSI

- [1] https://venturebeat.files.wordpress.com/2014/09/rogue_screen_shot_car.png. Tanggal Akses: 7 Desember 2014 pukul 23:30.
- [2] Munir, Rinaldi, "Matematika Diskrit", Informatika, Bandung: 2010.
- [3] <http://www.beyond3d.com/images/articles/ingenu-part-2/bsp.png>. Tanggal Akses: 9 Desember 2014 pukul 19:56.
- [4] http://www.mmoga.co.uk/images/screenshots/_p/1028116/af571a363d84e5bb51631a8840154765_crypt-of-the-necrodancer-steam-gift-key.jpg. Tanggal Akses: 9 Desember 2014 pukul 20:05.
- [5] <http://www.manapool.co.uk/mana-pool-guide-to-roguelikes/>. Tanggal Akses: 7 Desember 2014 pukul 23:41.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 10 Desember 2014



Cliff Jonathan
13513044