

Penerapan Graf dan Pohon dalam Dragon Nest

Ichwan Haryo Sembodo / 13512008
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13512008@std.stei.itb.ac.id

Abstract—Makalah ini membahas tentang penerapan graf dan pohon dalam permainan Dragon Nest, dalam makalah ini akan membahas tentang berbagai fitur dalam game Dragon Nest yang menerapkan graf dan pohon. Dragonnest sendiri adalah game ber-genre *massively multiplayer online role playing game* (MMORPG).

Index Terms—Dragon Nest, Online Game, Graf, Pohon.

I. PENDAHULUAN

Dragon Nest adalah sebuah *game online* ber-genre *massive multiplayer online role playing game* (MMORPG) fantasi. *Game* ini memiliki sistem non-target sehingga pemain memegang kendali penuh atas setiap gerakan karakternya. Ciri khas dari *game* ini adalah memiliki grafik dan pola bermain yang cepat sehingga membutuhkan sedikit konsentrasi dalam memainkannya.

Dragon Nest sendiri sudah dirilis di berbagai negara seperti Korea, China, Amerika dan lain-lain. Di Indonesia sendiri Dragonnest atau yang lebih dikenal Dragon Nest Indonesia rilis pada tanggal 14 Juni 2012.



Gambar 1.1 logo Dragon Nest.

Dengan menggabungkan sistem permainan non-target, pemain dapat memainkan gam ini dengan pengalaman yang banyak dan gerakan serba cepat. Untuk masalah peralatan dan senjata, tidak perlu khawatir dalam game ini sudah dilengkapi dengan persenjataan yang lengkap dan sesuai. Selain itu, pemain juga dapat melatih kecapakan dalam menjalankan game ini. Mempelajari skill menjadi hal penting untuk meningkatkan kekuatan pemain, dengan semakin kuat pemain maka dapat mengalahkan pemain lain dan juga monster di *dungeon* dengan cepat.

Dalam game ini semuanya diukur dengan ketrampilan dan kecakapan seorang pemain. Meskipun level rendah

jika memiliki ketrampilan yang cukup maka kemungkinan besar dapat mengalahkan pemain dengan level lebih tinggi.

Adapun beberapa fitur dalam Dragon Nest yaitu 3D action RPG, ini adalah fitur utama dan fitur khas dari *game* ini. Dengan adanya fitur ini pemain dapat memainkan karakternya dengan leluasa karena semua hal dalam game ini sudah menggunakan fitur 3D. Fitur lainnya adalah penempatan channel server otomatis, jika kita memasuki *game* ini maka kita akan otomatis di tempatkan di server yang tidak padat dengan menggunakan fitur ini maka kemungkinan terjadi *lag* berkurang.



Gambar 1.2 *combat in dungeon*

Seperti permainan online lainnya disini terdapat *quest-quest* untuk dijalankan, jika *quest* tersebut berhasil maka pemain akan mendapat sejumlah *exp* dan beberapa barang untuk dijual atau pun digunakan sendiri. Dalam permainan ini pemain diwajibkan mengumpulkan beberapa *exp* tertentu agar dapat melanjutkan ke level selanjutnya. *Quest* disini terbagi menjadi dua yaitu *main quest* dan *side quest*, pemain dapat mengikuti *main story* dengan menjalankan *main quest* ataupun pemain juga dapat menjalankan *side quest* jika pemain merasa bosan dengan *main story*-nya. Keunikan lain dari *game* ini memiliki *story* yang sangat menarik, cerita-cerita ini di masukan dalam berbagai *chapter*, setiap chapter memiliki *main character* berbeda sehingga pemain tidak akan cepat bosan dalam memainkan *game* ini.

II. LANDASAN TEORI

A. Matematika Diskrit

Matematika Diskrit adalah cabang matematika yang membahas objek-objek diskrit. Diskrit adalah keadaan dimana benda yang terdiri dari sejumlah elemen yang berbeda dan berhingga atau elemen yang tidak bersambungan. Lawan dari diskrit adalah kontinu. Hal yang dibahas pada matematika diskrit antara lain logika, teori himpunan, probabilitas, graf, pohon, dan lain-lain.

B. Graf

Graf merupakan salah satu topik yang dibahas dalam Matematika Diskrit. Graf digunakan untuk merepresentasikan objek – objek diskrit dan hubungan antara satu objek dengan objek lainnya. Representasi dari graf dengan menyatakan objek sebagai bulatan atau titik, dan hubungan objek dengan garis.

Graf $G = (V, E)$, yang dalam hal ini:

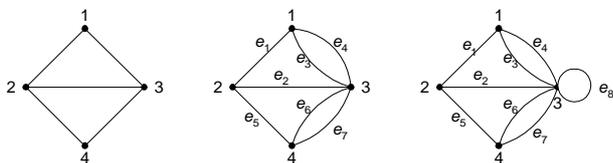
V = himpunan tidak-kosong dari simpul-simpul (*vertices*)

$$= \{ v_1, v_2, \dots, v_n \}$$

E = himpunan sisi (*edges*) yang menghubungkan sepasang

simpul

$$= \{ e_1, e_2, \dots, e_n \}$$



Gambar 2.b.1 (a) graf sederhana, (b) graf ganda, dan (c) graf semu

Contoh 1. Pada Gambar 2, G_1 adalah graf dengan

$$V = \{ 1, 2, 3, 4 \} \quad E = \{ (1, 2), (1, 3), (2, 3), (2, 4), (3, 4) \}$$

G_2 adalah graf dengan

$$V = \{ 1, 2, 3, 4 \}$$

$$E = \{ (1, 2), (2, 3), (1, 3), (1, 3), (2, 4), (3, 4), (3, 4) \}$$

$$= \{ e_1, e_2, e_3, e_4, e_5, e_6, e_7 \}$$

G_3 adalah graf dengan

$$V = \{ 1, 2, 3, 4 \}$$

$$E = \{ (1, 2), (2, 3), (1, 3), (1, 3), (2, 4), (3, 4), (3, 4), (3, 3) \}$$

$$= \{ e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8 \}$$

Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, maka graf digolongkan menjadi dua jenis:

1. Graf sederhana (*simple graph*).

Graf yang tidak mengandung gelang maupun sisi-ganda dinamakan graf sederhana. G_1 pada Gambar 2 adalah contoh graf sederhana

2. Graf tak-sederhana (*unsimple-graph*).

Graf yang mengandung sisi ganda atau gelang dinamakan graf tak-sederhana (*unsimple graph*). G_2 dan G_3 pada Gambar 2 adalah contoh graf tak-sederhana

Berdasarkan orientasi arah pada sisi, maka secara umum graf dibedakan atas 2 jenis:

1. Graf tak-berarah (*undirected graph*)

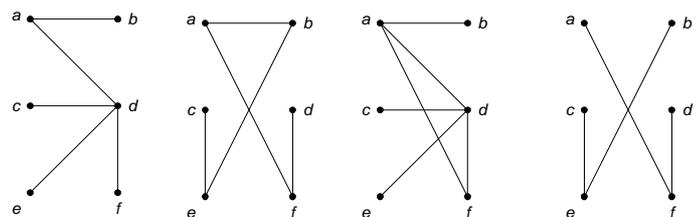
Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah. Tiga buah graf pada Gambar 2 adalah graf tak-berarah.

2. Graf berarah (*directed graph* atau *digraph*)

Graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah. Dua buah graf pada Gambar 3 adalah graf berarah.

C. Pohon

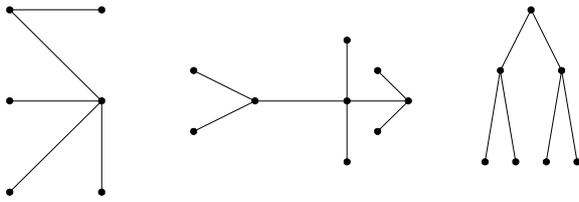
Pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit.



Gambar 2.c.1 (a) pohon (b) pohon (c) bukan pohon (d) bukan pohon

Hutan (*forest*) adalah

- kumpulan pohon yang saling lepas, atau
- graf tidak terhubung yang tidak mengandung sirkuit. Setiap komponen di dalam graf terhubung tersebut adalah pohon.



Hutan yang terdiri dari tiga buah pohon

C.1 Sifat-sifat (properti) pohon

Misalkan $G = (V, E)$ adalah graf tak-berarah sederhana dan jumlah simpulnya n . Maka, semua pernyataan di bawah ini adalah ekuivalen:

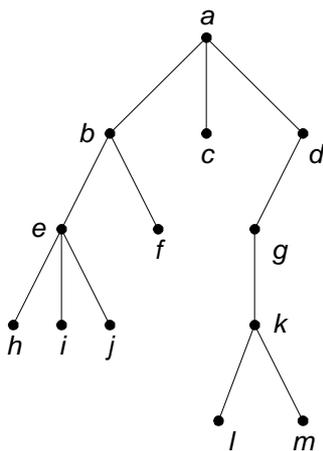
1. G adalah pohon.
2. Setiap pasang simpul di dalam G terhubung dengan lintasan tunggal.
3. G terhubung dan memiliki $m = n - 1$ buah sisi.
4. G tidak mengandung sirkuit dan memiliki $m = n - 1$ buah sisi.
5. G tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
6. G terhubung dan semua sisinya adalah jembatan.

Teorema di atas dapat dikatakan sebagai definisi lain dari pohon.

C.2 Terminologi pada pohon berakar

1. Anak (*child* atau *children*) dan Orangtua (*parent*)

$b, c,$ dan d adalah anak-anak simpul a , a adalah orangtua dari anak-anak itu.



2. Lintasan (*path*)

Lintasan dari a ke j adalah a, b, e, j .

Panjang lintasan dari a ke j adalah 3.

3. Saudara kandung (*sibling*)

f adalah saudara kandung e , tetapi g bukan

saudara kandung e , karena orangtua mereka berbeda.

4. Upapohon (subtree)

Yaitu simpul b dan e lalu daun f, h, i, j

5. Derajat (*degree*)

Derajat sebuah simpul adalah jumlah upapohon (atau jumlah anak) pada simpul tersebut.

Derajat a adalah 3, derajat b adalah 2,

Derajat d adalah satu dan derajat c adalah 0.

Jadi, derajat yang dimaksudkan di sini adalah derajat-keluar.

Derajat maksimum dari semua simpul merupakan derajat pohon itu sendiri. Pohon di atas berderajat 3.

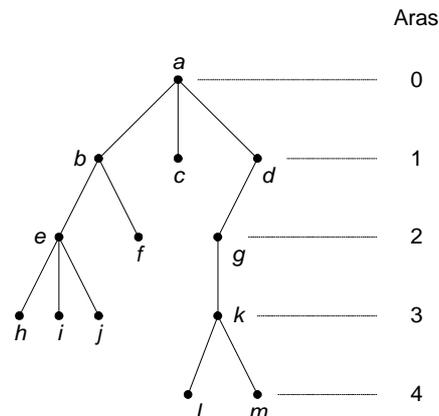
6. Daun (*leaf*)

Simpul yang berderajat nol (atau tidak mempunyai anak) disebut daun. Simpul $h, i, j, f, c, l,$ dan m adalah daun.

7. Simpul Dalam (internal nodes)

Simpul yang mempunyai anak disebut simpul dalam. Simpul $b, d, e, g,$ dan k adalah simpul dalam.

8. Aras (*level*) atau Tingkat



9. Tinggi (*height*) atau Kedalaman (*depth*)

Aras maksimum dari suatu pohon disebut tinggi atau kedalaman pohon tersebut. Pohon di atas mempunyai tinggi 4.

C.3 Pohon n -ary

Pohon berakar yang setiap simpul cabangnya mempunyai paling banyak n buah anak disebut pohon n -ary.

Pohon n -ary dikatakan teratur atau penuh (*full*) jika setiap simpul cabangnya mempunyai tepat n anak.

C.4 Pohon keputusan

Pohon keputusan berguna untuk memodelkan persoalan yang terdiri dari serangkaian keputusan yang mengarah ke solusi.

III. PENGGUNAAN MATEMATIKA DISKRIT PADA DRAGON NEST

A. Job Tree

Job Tree merupakan salah satu dari fitur Dragon Nest, dengan adanya fitur ini pemain dapat berganti *job* sehingga permainan tidak monoton dan pemain pun tidak cepat merasa bosan. Keuntungan dari berganti *job* adalah pemain bisa menggunakan *skill-skill* baru yang belum pernah mereka coba. Untuk berganti *job* sendiri pemain diwajibkan memenuhi beberapa syarat, yaitu *level* pemain harus mencukupi dan menjalankan *quest* khusus yang dinamakan *spezialitation*, setelah memenuhi kedua syarat tersebut maka karakter pemain pun dapat berganti *job*.



Gambar 3.a Job Tree

Di Dragon Nest sendiri pertama kali saat pemain memainkan *game* ini diwajibkan memilih 1 *job* dasar. *Job* tersebut nantinya akan berganti menjadi *job* lainnya atau bisa disebut *upgrade* sehingga *job* karakter pemain menjadi lebih kuat. Dragon Nest menggunakan jenis pohon 2-ary untuk *job tree*-nya karena hanya terdapat 2 percabangan disetiap pemilihan *job* baru. *Job tree* Dragon

Nest sendiri menggunakan sistem pohon karena untuk mencapai *job* terakhir kita harus memilih jalan yang diinginkan.

B. Skill Tree

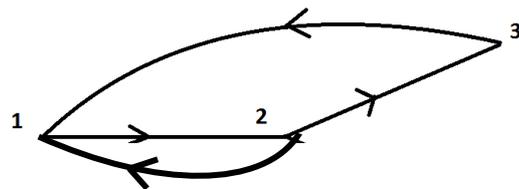
Skill Tree dalam Dragon Nest sangat menentukan seberapa kuatnya sebuah karakter, karena disini kita diwajibkan mengisi berapa jumlah *skill* yang diambil, jumlah tersebut mempengaruhi *damage* terhadap lawan yang dihadapi. Untuk mengambil *skill* pun ada beberapa syarat yaitu harus mengambil *skill* dasarnya terdahulu sebelum mengambil *skill* selanjutnya atau pun harus mengambil *job* baru terlebih dahulu, dengan sistem seperti ini maka sistem pohon lah yang sesuai untuk *skill tree*.



Gambar 3.b Skill Tree

Dalam mengambil *skill* pemain tidak bisa sembarangan mengisi *skill* tersebut dengan sesuka hati. Dragon Nest menggunakan SP (*skill point*) yang berguna untuk membatasi berapa jumlah *skill* yang dapat diambil pemain untuk mengisi *skill tree* karakternya.

C. Skill



Penggunaan *skill* dalam Dragon Nest menggunakan sistem graf, seperti pada gambar diatas. 1 adalah saat dimana meminta *input-an skill* dari pemain, lalu 2 adalah pengecekan apakah *skill* tersebut sedang *delay* atau sedang aktif, jika aktif maka lanjut ke 3 yaitu melakukan aksi dari *skill* tersebut dan kembali meminta *input-an skill* selanjutnya. Jika *skill* sedang *delay* maka graf kembali ke 1 dan meminta *input-an skill* kembali.

IV. BEBERAPA KESALAHAN UMUM

Dragon Nest merupakan *game* yang cukup berat dan membutuhkan *spesifikasi* yang tinggi untuk dapat memainkannya, sehingga sering terjadi *crash* ataupun *critical error*. Biasanya ini terjadi ketika pemain meng-*input*-kan suatu *skill* karena ketika terjadi peng-*input*-an komputer akan bekerja lebih. Dragon Nest sendiri sudah cukup memakan tempat di memori sehingga tidak ada sisa memori yang digunakan untuk proses lainnya. Kesalahan lainnya adalah pada *update* beberapa bulan lalu *skill tree* dan *skill* di Dragon Nest tidak berjalan dengan semestinya, ketika melakukan peng-*input*-an *skill* karakter dalam *game* tersebut tidak melakukan aksi apapun dan *damage* yang seharusnya dimiliki karakter karena *build skill tree*-nya tidak sesuai dengan yang tertera pada diskripsi *skill tree*.

V. KESIMPULAN

Teori graf dan pohon banyak digunakan dalam Dragon Nest. Dengan menggunakan teori graf dan pohon permainan menjadi menarik seperti saat pemain diharuskan cerdas dalam memilih *skill tree* maupun *job tree*.

DAFTAR PUSTAKA

- [1] <http://www.ign.com/articles/2011/10/08/dragon-nest-review>, 15 Desember 2013. 23:56 WIB..
- [2] <http://portal.paseban.com/review/12137/dragon-nest-indonesia>, 16 Desember 2013. 00:10 WIB.
- [3] <http://ggftw.com/forum/dragon-nest-fandom/99624-dragon-nest-class-tree.html>, 17 Desember 2013. 06:25 WIB.
- [4] Munir, Rinaldi, Slide Perkuliahan IF2120 Graf 2013, 17 Desember 2013. 09:52 WIB.
- [5] Munir, Rinaldi, Slide Perkuliahan IF2120 Pohon 2013, 17 Desember 2013. 10:09 WIB.

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, 17 Desember 2013



Ichwan Haryo Sembodo / 13512008