

Penerapan Teori Graf dan Pohon dalam Game Dragon Nest

Aditya Pratama / 13515103
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13515103@std.stei.itb.ac.id

Abstract—Sekarang ini banyak orang yang bermain game MMORPG (Massively Multiplayer Online Role-Playing game) tetapi banyak juga orang yang tidak mengerti bagaimana konsep game tersebut bekerja, seperti bagaimanakah konsep pindah map bekerja, konsep mengenai skill tree, dan konsep lainnya. Dalam makalah ini kita menerapkan cabang Matematika Diskrit berupa pohon dan graf untuk menjelaskan konsep – konsep yang ada pada game MMORPG.

Keywords—graf, pohon, game, Dragon Nest, karakter

I. PENDAHULUAN

Dalam permainan yang bergenre Massively Multiplayer Online Role – Playing game, pemain harus mengerti konsep dari game tersebut agar dapat memainkan game tersebut secara efisien dan superior dibanding yang lain, contohnya seperti kita harus memiliki kemampuan untuk menentukan skill apa yang harus kita ambil, ini merupakan poin yang penting karena akan berpengaruh ke depannya nanti saat level karakter kita sudah tinggi, banyak orang yang harus mengulang atau membuat karakter baru lagi karena salah memilih skill, sehingga karakter mereka menjadi tidak sesuai yang mereka inginkan dan mereka menjadi bosan memainkannya. Pengambilan keputusan diawal merupakan hal yang penting karena game MMORPG ini merupakan game yang berjangka panjang, karakter akan terus berkembang seiring naiknya level karakter, maka dari itu kita harus berhati – hati dalam memilih skill kita di awal – awal sehingga nantinya kita dapat membuat karakter yang kita inginkan. Strategi tersebut dapat kita mengerti dari teori tree dan graf yang telah dipelajari dalam Mata Kuliah Matematika Diskrit yang dalam makalah ini akan dibahas.

II. DASAR TEORI

A. GRAF

Graf G memiliki fungsi untuk merepresentasikan objek - objek diskrit dan hubungan antara objek - objek tersebut. Graf G didefinisikan sebagai pasangan himpunan (V, E) , yang dalam hal ini V merupakan himpunan tidak kosong dari simpul-simpul dan E merupakan himpunan sisi yang menghubungkan sepasang simpul. Dan di makalah ini graf yang dipakai hanya ada 5 jenis, yang dibagi lagi menjadi 3,

yaitu graf sederhana, graf tidak sederhana, graf berarah, graf ganda berarah, dan graf berbobot.

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 sisiganda dinamakan graf sederhana. G_1 pada Gambar 1.1 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 1.1 adalah contoh graf tak-sederhana. Setiap graf sederhana merupakan graf ganda, namun tidak semua graf ganda merupakan graf sederhana.

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

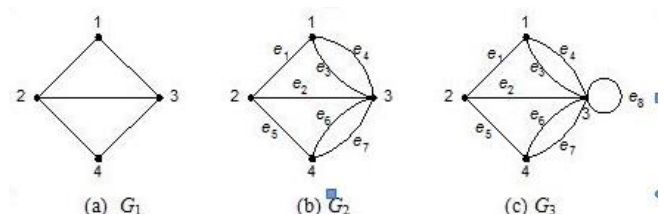
1. Graf tak-berarah (undirected graph)

Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah. Tiga buah graf pada Gambar 1.1 adalah graf tak-berarah..

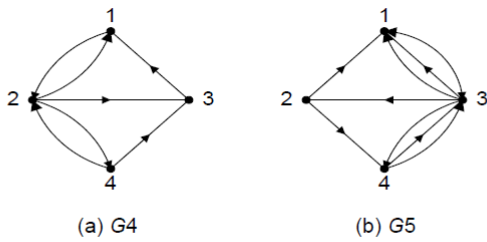
2. Graf berarah (directed graph atau digraph)

Graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah. Dua buah graf pada Gambar 1.2 adalah graf berarah.

Lalu mengenai graf berbobot, seperti pada gambar 1.3, anggap bobot dari sisi gambar tersebut adalah uang yang harus kita spend untuk dapat menyebrangi ke simpul selanjutnya, maka untuk menyebrang dari simpul a ke b, kita memerlukan uang sebanyak 12, yang berarti pada graf berbobot ini, setiap bobot dari sisi merupakan syarat agar dapat menyebrangi sisi tersebut.

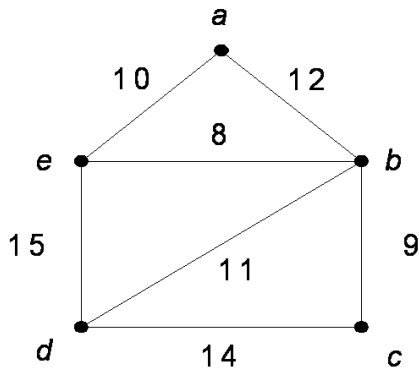


Gambar 1.1 (a) graf sederhana, (b) graf ganda, (c) graf semu



(a) graph berarah, (b) graph-ganda berarah

Gambar 1.2 (a) graf berarah, (b) graf ganda berarah [7]



Gambar 1.3 Graf berbobot [7]

B. Pohon

Pohon merupakan graf tak berarah terhubung yang tidak mengandung sirkuit, pada makalah ini ada 3 jenis pohon yang kita pakai, yaitu pohon berakar, pohon keputusan, dan pohon berbobot.

1. Pohon Berakar

Pohon yang sebuah simpulnya diberlakukan seperti akar dan sisi - sisinya diberi arah. Dan pada pohon berakar terdapat terminologi yang disebut anak (child) dan orangtua (parent), pada gambar 2.1, e merupakan child dari b dan b merupakan parent dari e.

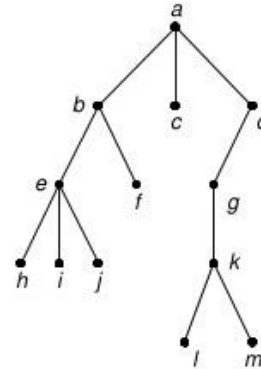
2. Pohon keputusan

Pohon keputusan adalah pohon yang berguna untuk analisis pemecahan masalah pengambilan keputusan. Pada gambar 2.2, pohon tersebut menampilkan kemungkinan /probabilitas yang akan mempengaruhi keputusan – keputusan yang akan kita ambil.

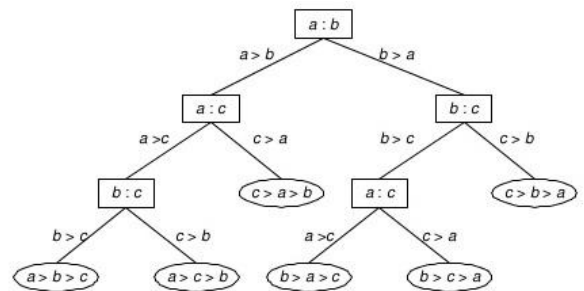
3. Pohon berbobot

Pada gambar 2.3, anggap bobot pada sisi pohon tersebut adalah level dan anggap nama simpul adalah

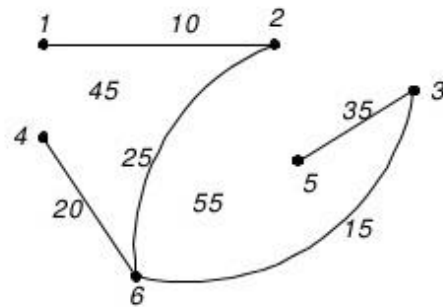
jenis job di game, maka untuk mengubah dari job 1 ke job 2, maka kita harus minimal harus mencapai level 10 baru bisa mengambil job tersebut, jadi setiap bobot disini merupakan syarat agar bisa ke simpul selanjutnya.



Gambar 2.1 Pohon berakar [7]



Gambar 2.2 Pohon keputusan [1]



Gambar 2.3 Pohon berbobot [7]

III. PENGGUNAAN GRAF PADA DRAGON NEST

Seperti yang dikatakan tadi bahwa di game MMORPG ini atau Dragon Nest ada konsep – konsep atau bisa disebut strategi yang harus kita tau agar kita bisa membuat karakter yang kita inginkan, salah satu kategori game yang dapat diaplikasikan dengan graf adalah saat kita pindah map, hal ini perlu kita tau juga karena jangan sampai saat kita memainkan game tersebut kita malah sering tersesat dan jadi tidak efisien menaikn level karakter game kita. Pada gambar 3.1, disini kita dapat menerapkan konsep dari graf, misalkan posisi kita sekarang berada di Kadena

Outpost, posisi tersebut mempunyai 2 sisi berarah yang berarti mempunyai akses ke map lain sebanyak 2, yaitu ke Crystal Valley dan Ashy Ruins, jadi pilihan untuk pindah map kita sekarang ada 2, yaitu pergi ke Crystal Valley atau Ashy Ruins, misalkan kita memilih untuk pindah map ke Crystal Valley, karena posisi kita sekarang (Crystal Valley) memiliki 3 sisi berarti pada kasus sekarang kita mempunyai pilihan pindah ke map lain sebanyak 3, yaitu ke Kadena Outpost (posisi sebelumnya), Thaw Forest, dan Whirlwind forest. Jadi kesimpulannya bahwa Game MMORPG ini juga memakai konsep graf untuk membuat map pada game tersebut, maka dari itu untuk bisa mengerti / memakai map pada game tersebut agar tidak tersesat, kita harus mengerti konsep dari graf.



Gambar 3.1 Sub Map pada Game Dragon Nest ^[2]

IV. PENGGUNAAN POHON PADA DRAGON NEST

Sama seperti SubBab III, bahwa kita juga harus mengerti konsep teori ini (pohon) agar bisa memainkan game dengan lebih efisien, seru, dan dapat membuat karakter sesuai yang kita inginkan.

A. Skill

Pada gambar 4.1, bisa dilihat bahwa skill tree dari swordmaster merupakan forest karena memiliki 3 pohon, dan misalkan kita mau mengambil skill aerial combo, maka ada 3 hal yang harus diperhatikan dan berkaitan dengan konsep pohon :

1. Mengenai parentnya

Karena skill aerial combo memiliki parent yang bernama triple slash maka jelas kita harus mengambil skill triple slash terlebih dahulu, jika kita belum mengambil skill triple slash atau parent dari aerial combo maka kita tidak bisa mengambil skill aerial combo, begitu juga dengan skill Deep Straight, jika kita ingin mengambil skill tersebut maka kita harus sudah mengambil parent dari skill

tersebut, dan parent dari parentnya skill tersebut juga, jadi intinya untuk mengambil skill kita harus sudah mengambil semua parent skillnya.

2. Pohon Berbobot

Pohon yang ada pada gambar 4.1 merupakan pohon berbobot, mengapa ? Karena syarat untuk mengambil skill tersebut bukan hanya parent skillnya saja, tetapi ada syarat lainnya yaitu level karakter kita, untuk mengambil skill Deep Straight misalkan kita perlu level 22, lalu misalkan kita sudah mengambil semua parent skill dari deep straight tetapi level kita baru 20, karena syarat belum cukup maka kita belum bisa mengambil skill tersebut

3. Skill Point

Jadi bisa disimpulkan syarat pada pohon berbobot tersebut ada 2 yaitu yang sebelumnya (level karakter) dan satu lagi yang akan kita bahas yaitu Skill Point (SP), jadi setiap karakter kita naik level, maka kita akan mendapatkan 1 Skill Point, untuk mengambil skill tertentu memerlukan Skill Point tertentu juga, misalkan untuk mempelajari skill Aerial Combo akan dibutuhkan 2 Skill Point, tetapi untuk skill lainnya misalkan Hacking Stance akan memerlukan 3 Skill Point, jadi setiap skillnya memiliki Skill Point yang bervariasi. Misalkan Skill Point sekarang kita 0 berarti jelas kita tidak bisa mengambil skill apapun karena paling sedikit Skill Point yang dibutuhkan adalah 1, begitu juga jika Skill Point kita 2 tetapi membutuhkan 3 Skill Point untuk mengambil skill tersebut, maka jelas kita tidak bisa mengambil skill tersebut.

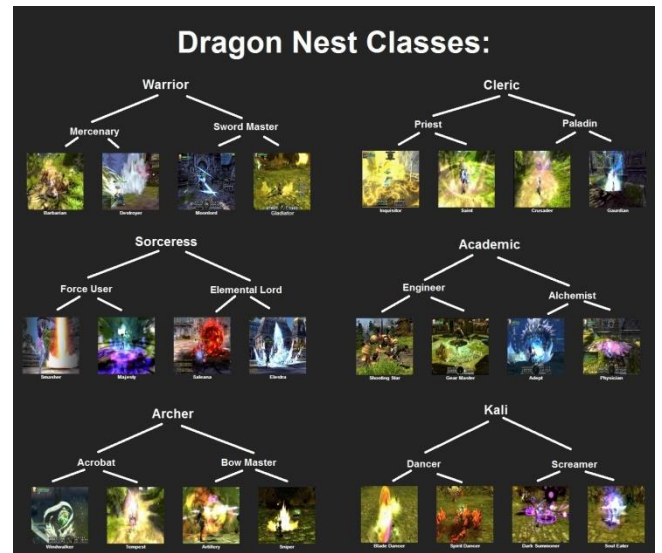
Jadi kesimpulannya bahwa skill tree dari swordmaster ini adalah forest yang memiliki 3 pohon berbobot, dan ada 2 syarat untuk pindah ke childnya (maksudnya mengambil skill baru) yaitu apakah level kita sudah mencukupi dan apakah skill point kita sudah cukup untuk mengambil skill selanjutnya. Mengapa kita harus mengerti konsep ini ? Karena skill point yang diberikan tidak bisa mengambil semua skill yang ada, maka kita harus calculate / menghitung – hitung agar kita bisa mendapatkan kombinasi skill yang terbaik dan sesuai dengan karakter yang kita inginkan, dan untuk bisa menghitung skill tersebut, kita harus mengerti konsep pohon terlebih dahulu, maka dari itulah kita harus mengerti konsep dari pohon.



Gambar 4.1 Skill Tree SwordMaster [3]

B. Class / Job

Pada Gambar 4.2 bisa dilihat bahwa itu adalah class / job dari karakter dragon nest, jadi misalkan kita ingin mengambil job WindWalker, maka ada 2 syarat yang harus diperhatikan yaitu level dan parent jobnya, untuk syarat level sama seperti pembahasan tadi pada Skill, yaitu untuk pindah job / class atau bisa dibilang berubah menjadi class baru, maka kita harus minimal mencapai syarat level yang sudah ditentukan, lalu mengenai parent skill juga sama seperti pada bagian Skill, yaitu untuk berubah menjadi job selanjutnya kita harus memiliki parent jobnya terlebih dahulu, misalkan mau menjadi WindWalker, berarti kita pertama harus jadi Archer terlebih dahulu, lalu mengambil Acrobat terlebih dahulu dan baru mengambil WindWalker setelah itu, kalau job kita Archer kita tidak bisa langsung merubah job kita menjadi WindWalker, karena kita belum mengambil parent job dari WindWalker. Jadi kesimpulannya adalah bahwa skill tree ini merupakan forest yang terdiri dari 6 pohon berbobot yang isi dari bobotnya hanya 1 yaitu harus memiliki level yang cukup untuk pindah ke childnya.



Gambar 4.2 Class / Job Dragon Nest [3]

C. Pilihan Skill

Disini barulah kita akan menerapkan konsep pohon keputusan, digambar 4.3 ditunjukkan kita ingin mengambil skill yang menghasilkan banyak damage, maka pertama kita harus mengambil skill buletan merah paling kiri atas, setelah itu dicek skill apakah yang mau kita ambil, karena kita ingin mengambil yang dapat membuat karakter kita dapat menghasilkan damage yang besar, maka kita memilih skill yang sebelah kanan dari sebelumnya, seterusnya sama, akan dicek lagi, dikondisinya akan ditanyakan, mau fokus damage (karakter yang gampang membunuh lawan) atau tanker (karakter yang susah dibunuh) atau support (karakter yang bisa membantu teman kita atau team dengan cara heal, atau boost damage dan lain - lain). Kemudian jika kita ingin fokus di damage, akan dicari, diantara semua kemungkinan manakah skill yang akan menghasilkan damage terbesar, maka dari itu untuk membuat karakter sesuai yang kita inginkan jika juga harus mengerti konsep pohon keputusan.



Gambar 4.3 Skill yang kita pilih agar damage karakter besar [4]

D. Chain Quest

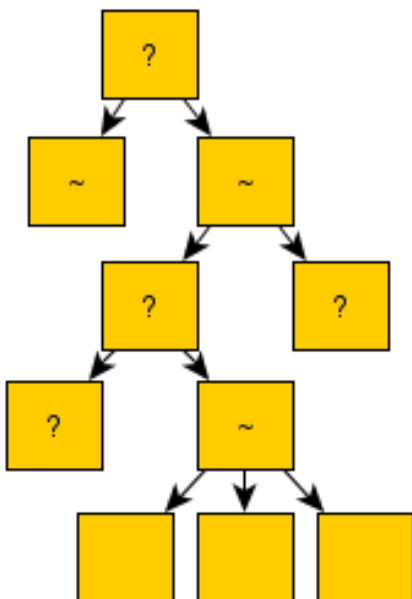
Pada gambar 4,4 bisa dilihat itu adalah quest yang harus dikerjakan oleh karakter kita, jadi chain quest tersebut berkaitan dengan konsep pohon juga karena untuk meneruskan quest tersebut ada 2 syarat yaitu sama seperti tadi bahwa parent questnya harus sudah dikerjakan dan levelnya harus mencukupi.



Gambar 4.4 Chain Quest karakter [6]

E. Enemy behavior

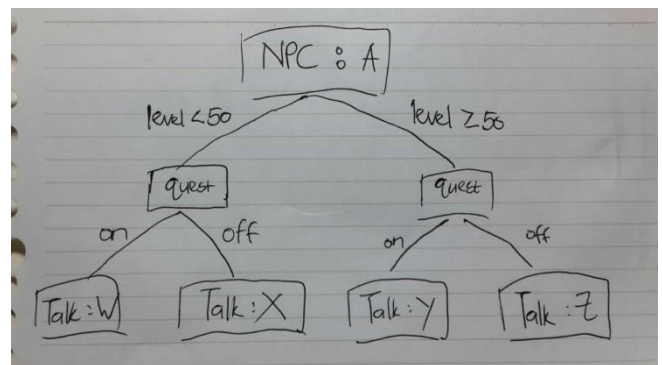
Jadi pada gambar 4.5, bisa dilihat itu adalah tree musuh, jadi maksudnya adalah bahwa dengan mengerti konsep pohon, kita juga dapat mengerti behavior musuh yang membuat kita membunuh musunya menjadi gampang, misalkan musuh uda diprogram di treenya jika kita sekarat maka musuh akan menjadi leboh agresif terhadap kita, jadi musuh langsung mendekati kita dan menyerang, lalu jika kita tau konsep pohon dan tau bahwa jika sekarat musuh akan menyerang kita dengan jarak yang dekat, maka misalkan kita memakai class / job yang unggul dalam jarak dekat, maka ini akan menjadi advantage untuk kita. Jadi kesimpulannya dengan kita mengerti konsep pohon kita bisa mengira – ngira selanjutnya musuh bakal melakukan apa.



Gambar 4.5 Enemy behavior [5]

F. NPC

Jadi maksudnya adalah bahwa misalkan ada NPC bernama A, ketika level kita lebih rendah dari 50 dan sedang tidak ada quest dengan dia maka dia akan berbicara kalimat X, tetapi ketika dia ternyata menyediakan quest maka dia akan berbicara dengan kalimat yang berbeda misalkan kalimat W, sama ketika kondisi dimana level kita lebih dari sama dengan 50, jadi intinya nanti di game kita dapat menganalisa kapan kita bisa melakukan quest dengan NPC tersebut jika kalimatnya W atau Y, maka kita akan mendapatkan quest dari NPC tersebut, jadi kesimpulannya adalah dengan mengetahui pola berbicara NPC, karena dari quest kita bisa mendapatkan uang dan exp (untuk naik level), maka ini bisa membantu kita untuk membuat karakter yang kita inginkan menjadi lebih cepat



Gambar 4.6 Pohon NPC

V. KESIMPULAN

Jadi game Dragon Nest ini, banyak menerapkan konsep dari graf dan pohon yang merupakan cabang dari Matematika Diskrit, karena itu agar kita bisa membuat karakter Dragon Nest yang efisien dan sesuai dengan yang kita inginkan maka kita harus mengerti konsep graf dan pohon yang digunakan pada game Dragon Nest tersebut.

REFERENCES

<http://pohonkeputusan-sukoand.blogspot.co.id/2011/01/pohon-keputusan.html>
(last visit : Friday) [1]

<http://dn.mmosite.com/map/map1.shtml>
(last visit : Friday) [2]

http://dragonnest.wikia.com/wiki/File:Sword_Master_Skill_Tree.png

(last visit : Friday) ^[3]

<https://www.youtube.com/watch?v=MMWyMumqlhE>

(last visit : Friday) ^[4]

http://www.gamasutra.com/blogs/ChrisSimpson/20140717/221339/Behavior_trees_for_AI_How_they_work.php

(last visit : Friday) ^[5]

<http://dragonnestsharpstation.blogspot.co.id/2011/09/quest-elf-queens-army.html>

(last visit : Friday) ^[6]

Munir, Rinaldi. 2005. Matematika Diskrit. Bandung: Penerbit Informatika ^[7]

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, 9 Desember 2016



Aditya Pratama
13515103

