

Pemanfaatan Graf dan Pohon dalam Pembuatan Permainan Peran (RPG)

Rizky Andyno Ramadhan – 13516063
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13516063@std.stei.itb.ac.id

Abstrak—Permainan adalah sebuah alat yang digunakan untuk menghibur diri di saat waktu senggang. Permainan pun dibagi menjadi dua, yaitu permainan tradisional yang dimainkan langsung secara fisik, dan permainan digital. Di masa kini, penciptaan permainan berbasis komputer (permainan *digital*) sudah menjadi pasar tersendiri. Merajalelanya sistem komputer di dunia menjadi alasan mengapa pembuatan permainan digital menjadi prospek yang menjanjikan. Salah satu *genre* yang populer dalam permainan *digital* adalah *Role-Playing Game* (atau biasa disingkat *RPG*). Dalam permainan *RPG*, banyak ditemukan penerapan graf dan pohon dalam bentuk *skill tree*, peta permainan, dan lain-lain.

Kata Kunci—Permainan *digital*, *Role-Playing Game*, Pohon, Graf, *Skill Tree*.

I. PENDAHULUAN

Role-Playing Game, atau yang secara literal disebut sebagai “permainan peran”, adalah permainan yang melibatkan pemain untuk mengambil peran menjadi seorang karakter imajiner di suatu dunia fantasi. Biasanya *RPG* dikaitkan dengan tema petualangan.

Permainan peran tradisional biasanya menggunakan ketiga elemen berikut :

1. Level atau statistik karakter yang dapat dikembangkan di sepanjang permainan
2. Pertarungan berbasis menu
3. Misi utama yang bertindak juga sebagai jalan cerita dari permainan.



Gambar 1.1 *Final Fantasy VII*, contoh pertarungan berbasis menu

Permainan peran modern biasanya tidak terpaku pada elemen-elemen tersebut, tetapi terkadang elemen-elemen tersebut dimodifikasi sehingga *flow* permainan jadi jauh lebih bervariasi.

Salah satu cara yang umum yaitu dengan mengubah sistem *leveling* atau statistik karakter yang bergantung pada sebuah sistem pohon yang membuat pemain harus memilih dengan bijak agar mereka bisa melanjutkan permainan. Di beberapa permainan, konsep pohon diterapkan dalam bentuk *skill tree*.



Gambar 1.2 *Skill Tree* dalam permainan *Star Wars : The Old Republic*

Pemanfaatan pohon dan graf juga bisa ditemukan di bagian lain di dalam permainan, misalnya pemilihan menu di menu utama, atau peta di dalam permainan. Selanjutnya akan dibahas di bagian selanjutnya.

II. LANDASAN TEORI

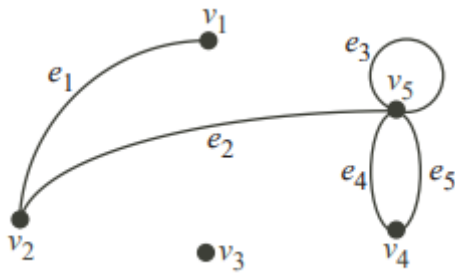
A. Teori Graf

1. Definisi Graf

Secara konsep, sebuah *graf* adalah hubungan antara titik-titik sudut (*vertices*) yang dihubungkan dengan sebuah sisi (*edge*).

Secara formal, sebuah graf adalah sepasang himpunan (V, E), di mana V adalah himpunan titik sudut (*set of vertices*) dan E adalah himpunan sisi (*set of edges*) yang dibentuk oleh pasangan titik sudut. E adalah *multiset*, yang artinya tiap elemen diperbolehkan muncul lebih

dari satu kali, sehingga tiap elemen memiliki multiplisitas masing-masing. Seringkali kita menandai titik sudut dengan huruf, misalnya a, b, c, \dots atau v_1, v_2, v_3, \dots) atau angka 1, 2,



Gambar 2.1 Contoh graf

Dari gambar di atas, terdapat $V = \{v_1, v_2, v_3, v_4, v_5\}$ dan $E = \{(v_1, v_2), (v_2, v_5), (v_5, v_5), (v_5, v_4), (v_5, v_4)\} = \{e_1, e_2, e_3, e_4, e_5\}$

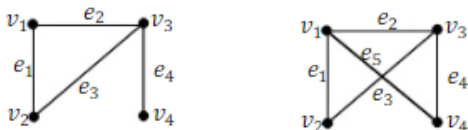
2. Macam-macam Graf

Umumnya, penentuan jenis graf didasarkan pada ada tidaknya sisi yang ganda atau gelang (*loop*), jumlah simpulnya, berbobot atau tidak.

Jenis graf yang dibedakan berdasarkan ada tidaknya sisi ganda atau gelang.

i. Graf sederhana (*simple graph*).

Graf ini merupakan graf tak berarah yang tidak memiliki sisi ganda atau gelang.



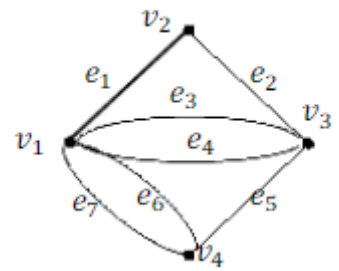
Gambar 2.2 Graf sederhana

ii. Graf tak sederhana (*unsimple graph*)

Graf ini merupakan lawan dari graf sederhana. Jadi, graf dengan gelang atau sisi ganda masuk ke kategori graf tak sederhana. Graf tak sederhana dibagi lagi menjadi dua, yaitu :

a. Graf ganda (*multigraph*)

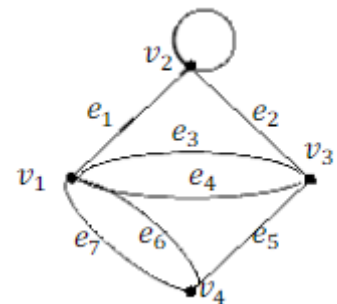
Graf ini mengandung lebih dari satu sisi untuk dua simpul yang sama.



Gambar 2.3 Graf ganda

b. Graf semu (*pseudograph*)

Graf ini merupakan graf yang memiliki gelang. Apabila di graf tersebut terdapat gelang dan sisi ganda, maka graf tersebut juga disebut sebagai graf semu.

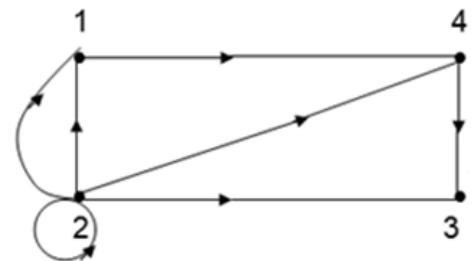


Gambar 2.4 Contoh graf semu

Selain itu, graf juga dapat dikelompokkan ke dalam dua kategori lainnya berdasarkan arah sisi, yaitu :

i. Graf berarah

Graf ini memiliki sisi yang berarah ke simpul tertentu. Pada graf berarah terdapat kondisi $(v_j, v_k) \neq (v_k, v_j)$



Gambar 2.5 Contoh graf berarah

ii. Graf tak berarah

Graf ini memiliki sisi yang tak memiliki arah. Contohnya dapat dilihat di graf sederhana dan graf tak sederhana.

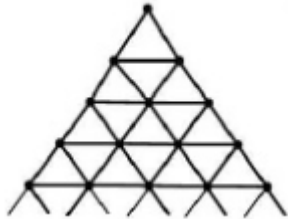
Apabila dilihat berdasarkan jumlah simpulnya, graf dibagi lagi menjadi dua kategori, yaitu :

i. Graf berhingga (*limited graph*)

Graf ini memiliki jumlah simpul yang tak

terbatas. Contohnya dapat dilihat di gambar-gambar yang ditempel di atas.

- ii. Graf tak berhingga (*unlimited graph*)
Graf ini memiliki jumlah simpul tak berhingga. Graf ini ditandai dengan ujung sisi yang tidak terhubung dengan simpul.



Gambar 2.6 Graf tak berhingga

3. Representasi Graf

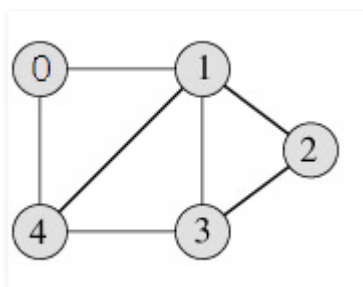
Untuk pemrosesan graf di komputer, graf harus direpresentasikan sesuai dengan struktur data yang ada. Terdapat beberapa representasi yang cocok untuk diimplementasikan sebagai graf.

Di sini akan dijelaskan tiga buah representasi berbeda dari graf, yaitu matriks ketetanggaan (*adjacency matrix*), matriks bersisian (*incidency matrix*), dan list ketetanggaan (*adjacency list*).

- i. Matriks ketetanggaan (*adjacency matrix*)
Matriks ketetanggaan adalah array dua dimensi berukuran $N \times N$ di mana N adalah jumlah simpul di graf tersebut.

Awalnya semua elemen matriks bernilai 0, dan untuk array bernama `adj[][]`, elemen `adj[i][j]` akan bernilai 1 apabila ada sisi yang menghubungkan simpul i dan simpul j .

Matriks ketetanggaan juga dapat digunakan untuk merepresentasikan graf berbobot. Apabila `adj[i][j]` bernilai w , maka ada sisi yang menghubungkan simpul i dan simpul j dengan bobot w .



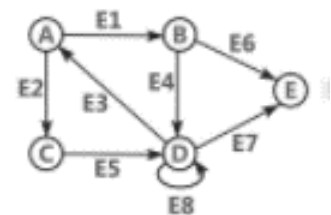
Gambar 2.7 Contoh graf

	0	1	2	3	4
0	0	1	0	0	1
1	1	0	1	1	1
2	0	1	0	1	0
3	0	1	1	0	1
4	1	1	0	1	0

Gambar 2.8 Contoh matriks ketetanggaan yang mewakili graf pada gambar 2.7

- ii. Matriks bersisian (*incidency matrix*)
Pada representasi ini, graf bisa direpresentasikan dengan matriks berukuran jumlah simpul dikalikan jumlah sisi. Artinya apabila sebuah graf memiliki 4 simpul dan 6 sisi, graf tersebut dapat direpresentasikan menggunakan matriks berukuran 4×6 . Pada matriks ini, baris merepresentasikan jumlah simpul dan kolom merepresentasikan jumlah sisi.

Matriks ini diisi dengan angka 0 atau 1 untuk menyatakan keterhubungan antar simpul pada graf tersebut. Untuk graf berarah, digunakan angka 0, 1, atau -1 untuk menandakan arah; 1 sebagai simpul penunjuk dan -1 sebagai simpul yang ditunjuk.

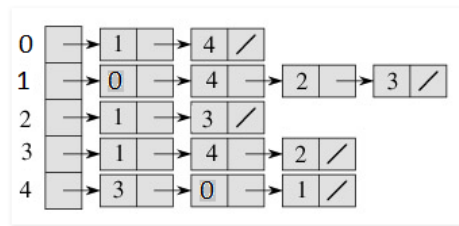


	E1	E2	E3	E4	E5	E6	E7	E8
A	1	1	-1	0	0	0	0	0
B	-1	0	0	1	0	1	0	0
C	0	-1	0	0	1	0	0	0
D	0	0	1	-1	-1	0	1	1
E	0	0	0	0	0	-1	-1	0

Gambar 2.9 Contoh graf berarah dan matriks bersisiannya

- iii. List ketetanggaan (*adjacency list*)
Pada list ketetanggaan ini, kita akan menggunakan array bertipe list berkait. Ukuran array disesuaikan dengan jumlah simpul yang ada di dalam graf. List berkait

berisi semua simpul yang terkait dengan simpul indeks *array*.



Gambar 2.10 Representasi *list* dari graf yang terdapat pada gambar 2.7

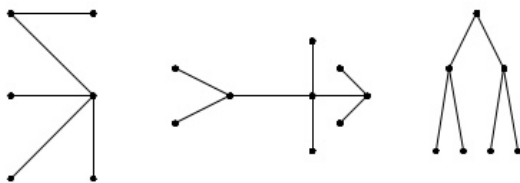
B. Teori Pohon

1. Definisi Pohon

Pohon (*tree*) merupakan salah satu bentuk khusus dari struktur suatu graf. Pohon didefinisikan sebagai graf terhubung yang tidak memiliki sirkuit. Karena pohon adalah sebuah graf terhubung, maka si pohon selalu terdapat jalur yang menghubungkan setiap dua simpul dalam pohon.

Misalkan $G = (V, E)$ adalah graf tak berarah sederhana dan jumlah simpulnya n . Maka G adalah pohon dan sifat-sifat yang berlaku adalah :

1. G memiliki n buah simpul dan $n - 1$ buah sisi.
2. G tidak memiliki sirkuit
3. Setiap pasang simpul di dalam pohon terhubung dengan lintasan tunggal (tidak ada sisi ganda)
4. G terhubung dan semua sisinya adalah jembatan (apabila salah satu sisi dihapus maka akan menyebabkan graf terpecah menjadi dua bagian)
5. Setiap simpul dapat dicapai dengan lintasan yang unik.



Gambar 2.11 Contoh pohon

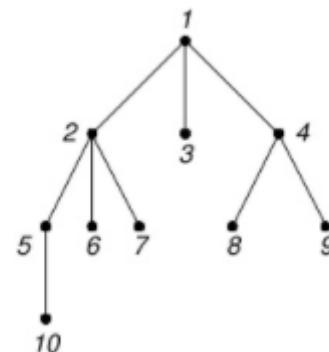
2. Terminologi Pohon

Terdapat beberapa istilah yang digunakan dalam konsep pohon ini. Berikut terminologi yang umum digunakan (tiap contoh yang digunakan merujuk pada gambar 2.12):

1. Anak (*child*) dan Orangtua (*parent*)
Misalkan x adalah sebuah simpul dari pohon berakar, maka simpul y dikatakan sebagai anak dari simpul x jika ada sisi yang menghubungkan dari simpul x ke y . Sebaliknya, x dikatakan orang

tua dari y . Contoh : Simpul 6 adalah anak dari simpul 2 dan simpul 2 adalah orang tua dari simpul 6.

2. Lintasan (*path*)
Lintasan adalah runtutan simpul-simpul yang dihubungkan dengan sisi dan Panjang lintasan adalah jumlah sisi yang menghubungkan antara dua buah simpul dari sebuah pohon. Namun, ini hanya berlaku apabila simpul yang dihitung lintasannya adalah simpul anak dan orang tua. Contoh : Panjang lintasan dari simpul 1 ke simpul 10 adalah 3 dan lintasannya adalah 1, 2, 5, 10.
3. Keturunan (*descendant*) dan Leluhur (*ancestor*)
Jika terdapat lintasan dari simpul x ke simpul y di dalam pohon, maka x adalah leluhur dari simpul y dan y adalah keturunan dari simpul x . Contoh : Simpul 2 adalah leluhur simpul 10 dan simpul 10 adalah keturunan dari simpul 2.
4. Saudara kandung (*sibling*)
Simpul yang berorangtua sama disebut saudara kandung satu sama lain. Contoh : Simpul 5, 6, dan 7 saling bersaudara kandung.
5. Aras (*Level*) atau tingkat
Akar memiliki aras = 0, sedangkan aras simpul lainnya = panjang lintasan dari akar ke simpul tersebut. Contoh : Aras dari simpul 5 adalah dua dan aras dari simpul 10 adalah tiga.
6. Tinggi (*height*) atau kedalaman (*depth*)
Aras maksimum dari suatu pohon disebut dengan tinggi atau kedalaman. Tinggi pohon adalah panjang maksimum lintasan dari akar ke daun. Contoh : pohon ada gambar 2.12 memiliki ketinggian 3.
7. Daun (*leaf*)
Simpul yang tidak mempunyai anak disebut daun. Contoh : Simpul 6, 7, 8, 9, dan 10 adalah daun.



Gambar 2.12 Pohon berakar

III. PEMANFAATAN GRAF DAN POHON DALAM PEMBUATAN PERMAINAN PERAN (RPG)

Mengapa graf dan pohon memegang peran penting dalam pembuatan sistem permainan untuk membuat permainan peran? Beberapa *benefit* apabila pencipta permainan peran

1. *User Preference Based Experience*

Perkembangan karakter dapat disesuaikan dengan kemauan pemain dengan memanfaatkan pohon dalam bentuk *skill tree* sehingga muncul berbagai macam kombinasi karakter yang terdapat di dalam permainan sesuai dengan gaya bermain masing-masing pemain. Ini memberikan pemain keleluasaan dalam menciptakan karakternya sendiri sehingga pemain menjadi tidak cepat bosan.

2. *Non-linear Storyline*

Dengan memanfaatkan pohon atau graf, *developer* permainan dapat menciptakan beberapa cabang cerita sesuai dengan tindakan yang dilakukan oleh pemain di dalam permainan. Ini membuat pengalaman masing-masing pemain menjadi berbeda dan meningkatkan *replayability* dari permainan ini sehingga memperpanjang durasi permainan.

3. Pengembangan Sistem *Combat*

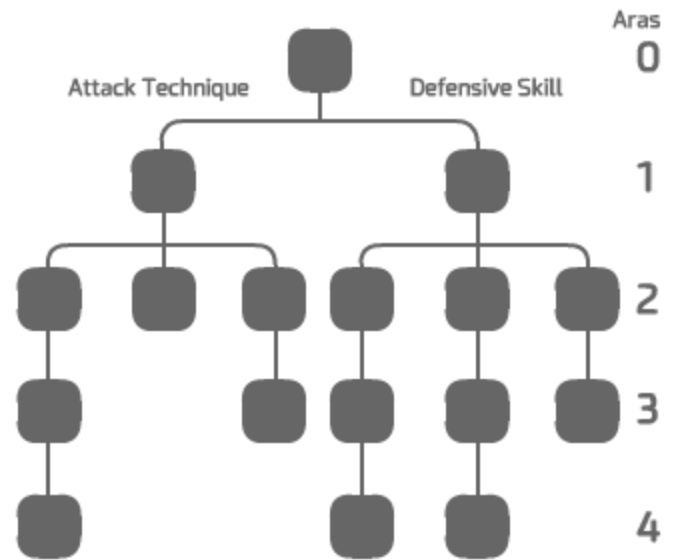
Dalam permainan peran biasa ditemukan pertarungan antar karakter. Di sini muncul peran dari graf berarah agar membuat sistem *combat* menjadi lebih kompleks. Tiap karakter memiliki kelasnya masing-masing, dan tiap kelas akan memicu reaksi yang berbeda apabila dihadapi dengan kelas lainnya. Ini dapat direalisasikan dengan membuat graf berarah dari tiap kelas. Misalkan menurut graf berarah kelas A menunjuk ke kelas B, maka kelas B akan menerima *damage* lebih besar dari kelas A. Sebaliknya, kelas A akan mendapatkan *damage* yang lebih kecil apabila diserang oleh kelas B. Tentunya ini akan menguji taktik dari pemain agar tidak salah melakukan aksi.

Sedikit mengulas ulang tentang permainan peran, pemain akan berperan sebagai karakter fiktif yang akan berpetualang di dunia fantasi. Tiap karakter akan memiliki statistik dan ciri khasnya masing-masing. Untuk menaikkan atau mengembangkan kemampuan dari karakter tersebut, digunakanlah *skill tree*.

Skill tree, atau dalam bahasa Indonesianya adalah pohon ketrampilan, adalah suatu sistem dalam permainan yang membebaskan penggunanya untuk memilih *skill* apa dulu yang ingin dimiliki atau di-*upgrade* oleh si pemain.

Biasanya, cara si pemain agar dapat meng-*upgrade* karakternya itu adalah dengan cara menaikkan level dari karakter tersebut dan pemain akan diberi hadiah berupa *skill point* yang dapat digunakan untuk memilih *skill* pada *skill tree*.

Contoh *skill tree* dapat dilihat pada gambar 3.1



Gambar 3.1 Contoh *skill tree* pada permainan peran

Tiap kotak di atas mewakili satu *skill* yang bisa dimiliki oleh karakter dari pemain. Semakin dalam *skill* tersebut dalam suatu pohon (semakin besar arasnya), semakin *powerful* pula dampak yang diberikan dari *skill* tersebut.

Bersesuaian dengan gambar di atas, pemain bisa memilih untuk mengembangkan *attack technique*-nya terlebih dahulu atau *defensive skill*-nya terlebih dahulu. Ini akan membuat pemain untuk memikirkan terlebih dahulu kira-kira *skill* apa yang sedang dibutuhkan karakternya.

Pada suatu ketika, tiba-tiba karakter si pemain dihampiri oleh musuh sehingga memulai mode *combat* dan pertarungan pun dimulai...

V. UCAPAN TERIMA KASIH

Pertama-tama, penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada Allah Swt. untuk semua berkah dan umur yang telah diberikan. Oleh karena berkat-Nya pula penulis dapat menyelesaikan makalah ini. Penulis juga ingin berterima kasih kepada keluarga, terutama orang tua penulis yang sangat mendukung keberjalanan studi penulis dengan doa dan dukungan moralnya. Selain itu, penulis ingin berterima kasih kepada Bapak Judhi Santoso karena berkatnya penulis mengetahui inti dari Matematika Diskrit.

REFERENSI

- [1] <https://www.techopedia.com/definition/27052/role-playing-game-rpg>
Diakses pada tanggal 3 Desember 2017
- [2] http://math.tut.fi/~ruohonen/GT_English.pdf
Diakses pada tanggal 3 Desember 2017
- [3] http://www.academia.edu/27613536/Jenis_Jenis_Graf_dan_Graf_Bipartisi
Diakses pada tanggal 3 Desember 2017
- [4] <http://www.geeksforgeeks.org/graph-and-its-representations/>
Diakses pada tanggal 3 Desember 2017
- [5] http://btechsmartclass.com/DS/U3_T9.html
Diakses pada tanggal 3 Desember 2017

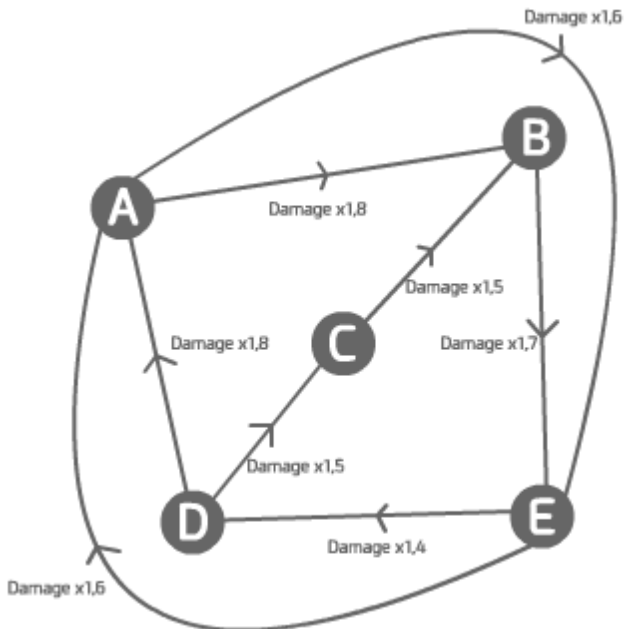
PERNYATAAN

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, 3 Desember 2017



Rizky Andyno Ramadhan / 13516063



Gambar 3.2 Penerapan graf berarah pada sistem *combat* dalam permainan peran

Si pemain menyadari bahwa ia menggunakan karakter berkelas A dan kini ia sedang melawan karakter berkelas B. Tentu ia akan mendapatkan *benefit* karena karakter yang digunakannya akan mendapatkan kelipatan 1,8 untuk setiap *damage* yang dia kenakan kepada musuhnya. Ini juga membuat pemain berhati-hati dalam melawan musuh berkelas D karena musuhnya akan mendapatkan kelipatan 1,8 untuk setiap *damage* yang dikenakan kepada karakter si pemain.

IV. KESIMPULAN

Teori graf dan pohon dapat dimanfaatkan untuk berbagai macam kebutuhan. Salah satu manfaat dari teori tersebut yaitu pengimplementasiannya sebagai sistem permainan peran (RPG).

Dengan memanfaatkan teori graf dan pohon di dalam permainan peran, *gameplay* pun akan menjadi lebih kompleks sehingga permainan tidak menjadi cepat membosankan. Pemain dapat memilih *skill* dari *skill tree* yang bercabang sehingga *outcome* permainan akan menjadi berbeda dan membuat pengalaman tiap pemain akan berbeda pula setiap kali mereka memainkan permainan tersebut.

Dengan memanfaatkan teori graf, sistem *combat* di dalam permainan akan mengasah kemampuan menyusun taktik pemain karena tiap kelas dari karakter dapat mempengaruhi hasil serangan yang dilakukan.

Selain contoh-contoh di atas, masih banyak lagi pemanfaatan graf dan pohon yang bisa diterapkan di dalam permainan peran (RPG).