

Penerapan Struktur Diskrit dalam Game Mechwarrior 4: Mercenaries

Sandy Gunawan Tanuwijaya/ 13510025

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

sandy.gunawan@mail.students.itb.ac.id

Abstract—Makalah Struktur Diskrit ini akan membahas mengenai penerapan berbagai teori Struktur Diskrit pada game Mechwarrior 4: Mercenaries. Mechwarrior 4: Mercenaries adalah sebuah game simulasi perang robot yang berfokus kepada peran karakter utama sebagai seorang tentara bayaran dan modifikasi robot yang digunakan.

Beberapa teori yang akan dibahas antara lain penggunaan himpunan, graf, dan pohon yang diaplikasikan pada game Mechwarrior 4: Mercenaries

Kata Kunci: Mechwarrior, Struktur Diskrit, Himpunan, Graf

I. PENDAHULUAN

A. Pengenalan Game Mechwarrior 4: Mercenaries

Mechwarrior 4: Mercenaries merupakan sebuah game simulasi pertempuran robot (yang di sini dinamakan *Battletech* atau biasa dipanggil Mech) yang dibuat berdasarkan dunia *Battletech* buatan FASA Corporation. Game ini dirilis oleh Microsoft untuk konsol PC tahun 2002, dan merupakan *expansion pack standalone* dari Mechwarrior 4. Mulai tahun 2010, Mechwarrior 4 Mercenaries dapat diunduh secara gratis melalui situs MekTek.



Pada mode Campaign, pemain berperan sebagai seorang tentara bayaran dengan nama sandi Spectre. Pemain dapat menyelesaikan misi yang disediakan oleh berbagai faksi untuk mendapatkan uang yang dapat digunakan untuk membeli persenjataan baru dan merekrut anak buah. Aspek yang paling utama pada game ini adalah kustomisasi robot tempur/*Battletech* untuk menyesuaikan dengan gaya bertarung, misi, atau peran pemain dalam pertempuran.



Gambar 1: Tampilan antar muka menu kustomisasi Mech, yang disebut Mechlab

B. Pengenalan Struktur Diskrit

Struktur diskrit adalah suatu cabang matematika yang membahas struktur matematik yang bersifat diskrit atau tidak kontinu. Hal-hal yang dibahas pada struktur diskrit antara lain teori himpunan, logika, probabilitas, kombinatorial, graf, pohon, dan lain-lain. Struktur diskrit sangat penting dan banyak diterapkan di bidang algoritma komputer, bahasa pemrograman, kriptografi, pengembangan perangkat lunak, dan lain-lain.

II. DASAR TEORI

A. Logika

Logika merupakan dasar dari semua penalaran (*reasoning*). Penalaran logika dikaitkan dengan hubungan antara hubungan antara pernyataan-pernyataan (*statements*) yang ada.

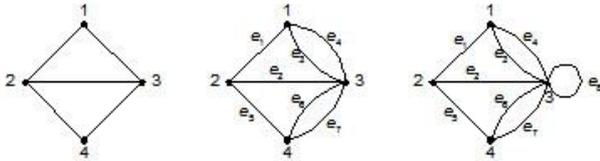
B.1. Proposisi

Proposisi adalah kalimat deklaratif yang bernilai benar (*true*) atau salah (*false*), tetapi tidak keduanya. Proposisi dapat juga disebut sebagai kalimat terbuka. Tidak semua kalimat dapat menjadi proposisi, hanya pernyataan/kalimat berita sajalah yang dapat menjadi proposisi.

B. Himpunan

Himpunan (*set*) adalah kumpulan dari objek yang berbeda. Maksudnya berbeda adalah bahwa anggota himpunan tidak boleh sama. Objek yang terdapat pada himpunan disebut elemen, unsure, atau anggota.

C. Graf



Gambar 2: Jenis-jenis graf, ki-ka: Graf sederhana, graf ganda, dan graf semu.

Graf adalah sebuah struktur diskrit yang digunakan untuk merepresentasikan objek diskrit dengan sesamanya dan hubungan antara objek tersebut. Graf G didefinisikan sebagai pasangan himpunan (V,E) yang dalam hal ini:

V (untuk vertices): himpunan tidak-kosong dari simpul-simpul = $\{v_1, v_2, \dots, v_n\}$

E (untuk edges): himpunan sisi yang menghubungkan sepanjang simpul = $\{e_1, e_2, \dots, e_n\}$, dapat juga ditulis sebagai $G = (V,E)$.

V tidak boleh kosong, sedangkan E boleh kosong. Sehingga sebuah graf mungkin tidak memiliki satupun sisi, tetapi minimal harus ada satu simpul. Graf seperti ini dinamakan graf trivial.

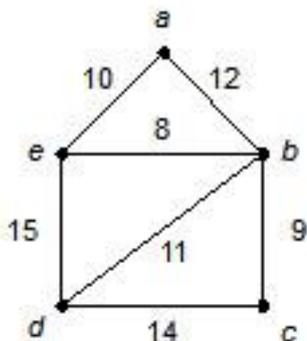
Graf juga dapat dibagi menjadi dua jenis, yaitu:

1. Graf sederhana: graf yang tidak mengandung gelang maupun sisi-ganda.
2. Graf tak-sederhana: Graf yang mengandung sisi gelang atau ganda. Ada dua macam graf tak-sederhana, yaitu graf ganda dan graf semu. Graf ganda adalah graf yang mengandung sisi ganda, sedangkan graf semu adalah graf yang mengandung gelang.

Graf juga dapat dibedakan berdasarkan orientasi arahnya:

1. Graf tak-berarah: Graf yang sisinya tidak mempunyai orientasi arah.
2. Graf berarah: Graf yang setiap sisinya mempunyai orientasi arah (arc). Pada graf berarah, (v_j, v_k) dan (v_k, v_j) menyatakan dua buah busur yang berbeda. Simpul v_j dinamakan simpul asal (*initial vertex*) dan simpul v_k dinamakan simpul terminal (*terminal vertex*)

D.1. Graf Berbobot

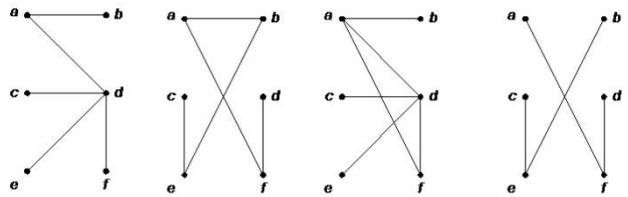


Gambar 3: Contoh graf berbobot

Graf berbobot (*weighted graph*) adalah graf yang setiap sisinya diberi sebuah harga (bobot/*weight*). Bobot pada tiap sisi dapat menyatakan jarak antara dua buah kota, biaya perjalanan antara dua buah kota, waktu tempuh

pesan (*message*) dari sebuah simpul komunikasi lain (dalam jaringan computer), ongkos produksi, dan lain-lain.

D. Pohon



Gambar 4: Perbedaan antara graf pohon dan graf bukan pohon

Pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit. Sebuah pohon dapat saja hanya mempunyai sebuah simpul tanpa sisi, pohon seperti ini disebut pohon bebas untuk membedakannya dengan pohon berakar.

Beberapa pohon dapat membentuk hutan (*forest*). Hutan adalah kumpulan pohon yang saling lepas. Hutan juga memiliki sifat yang sama dengan pohon, yaitu tidak terhubung dan tidak mengandung sirkuit.

III. METODE

A. Penerapan Himpunan dalam Battlemech dan Hardpoint senjata

A.1 Logika dan Himpunan pada Hardpoint Senjata

Slots	Name	Tons	#
122307	Medium X-Pulse Laser	2.50	1
312003	ER Medium Pulse Laser	2.0	10
219333	ER Small Pulse Laser	1.0	9
155306	PPC	7.0	4
356738	ER PPC	6.0	1

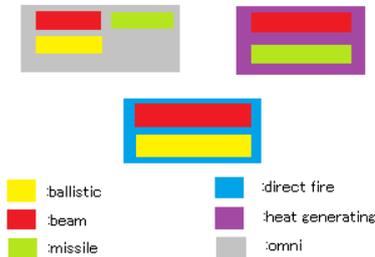
Gambar 5: Contoh senjata kategori *beam*

Setiap senjata yang dapat dipasang pada Mech hanya dapat dipasang pada *hardpoint* tertentu, antara lain:

- *Hardpoint beam* untuk senjata berbasis energy, seperti laser atau PPC (*Particle Projectile Cannon*)
- *Hardpoint ballistic* untuk senjata proyektil yang menggunakan peluru (*solid rounds*), seperti senapan mesin atau *autocannon*
- *Hardpoint missile* untuk peluru kendali dan sejenisnya
- *Hardpoint omni* yang dapat memuat segala jenis senjata

- *Hardpoint direct fire* yang dapat memuat senjata bertipe *beam* dan *ballistic*
- *Hardpoint heat generating* yang hanya dapat memuat senjata bertipe *missile* dan *beam*.
- *Battle Armor Hardpoint* yang hanya dapat memuat senjata-senjata khusus *battle armor*, dengan pembagian kategori seperti di atas (*beam*, *missile*, *ballistic*, *direct fire*, *heat generating*).

Penjelasan tentang *hardpoint* senjata:



Gambar 6: Penerapan Himpunan pada *Hardpoint*
 Semua jenis senjata dibagi lagi menjadi dua, berdasarkan pemakainya, yaitu Inner Sphere dan Clan. Tetapi setiap *hardpoint* dapat memuat kedua jenis senjata.

A.2 Himpunan pada *Battlemech*

Pada *Mechwarrior 4: Mercenaries*, setiap *Battlemech* dibagi menjadi beberapa kategori sesuai dengan ukuran dan berat *Battlemech*.

- *Battle Armor*, dengan berat 20 ton.
- *Light*, dengan kisaran berat 20-35 ton
- *Medium*, dengan kisaran berat 40-55 ton
- *Heavy*, dengan kisaran berat 60-75 ton
- *Assault*, dengan kisaran berat 80-100 ton



Gambar 7: Contoh kategori *Medium Mech*

B. Penerapan Himpunan pada *Grouping* Senjata

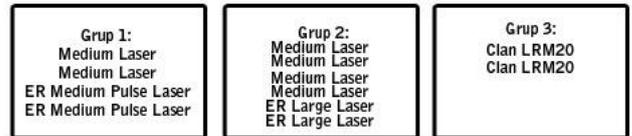
Senjata yang telah dipasang pada *Mech* dapat kita bagi menjadi beberapa kelompok. Untuk menembak beberapa senjata sekaligus, senjata yang dipasang harus dikelompokkan sesuai dengan tombol yang diset.

Misalnya, tombol kiri mouse diset untuk menembak seluruh senjata pada grup 1, tombol Enter diset untuk menembak seluruh senjata grup 2, dan lain-lain. Patut dicatat bahwa senjata yang sama dapat ditempatkan pada dua grup yang berbeda.



Gambar 8: Pengelompokan senjata

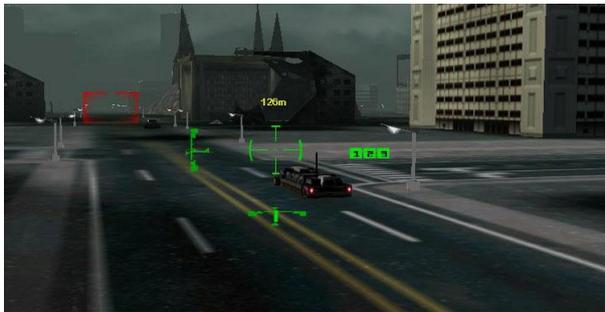
Untuk contoh di atas, himpunan yang ada dapat dijelaskan dengan grafik sebagai berikut:



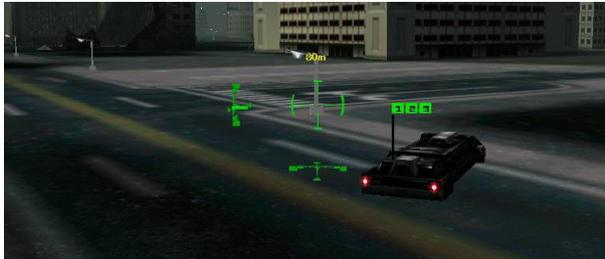
Gambar 9: Penjelasan himpunan senjata pada gambar 8

C. Penerapan Graf untuk *Pathfinding* AI dan *Navpoint*

C.1 Penerapan Graf untuk *Pathfinding* AI
 Pergerakan unit-unit yang dikendalikan oleh AI menggunakan aplikasi graf untuk *Pathfinding*. Misalnya, pada beberapa kasus, konvoi kendaraan dan Mech akan mengikuti suatu rute tertentu (AI mengikuti graf dengan bentuk tertentu) hingga mencapai tujuan. Jika ada halangan di depan mereka, maka AI akan mencoba untuk menghindari halangan tersebut.



Rute seharusnya



Ketika ada rintangan, AI mencoba untuk menghindari lalu kembali meneruskan rute yang telah diprogram

Gambar 10: Penjelasan tentang kasus *Pathfinding* AI

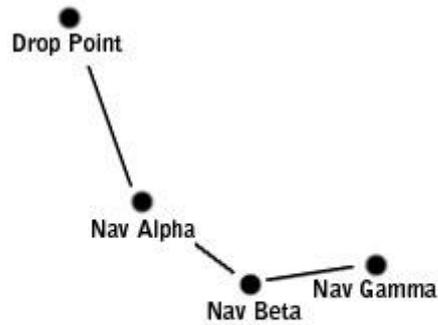
C.2 Penerapan Graf pada *Navpoint*



Gambar 11: Contoh penggunaan *Navpoint*

Pada setiap misi, terdapat satu atau lebih *Navpoint* untuk memudahkan pemain menavigasi suatu level dan menemukan sasaran misi (*objective*). Karena biasanya *objective* tidak harus diselesaikan secara berurutan, maka pemain dapat menuju *Navpoint* manapun terlebih dahulu untuk menyelesaikan *objective*. Cara kerja *Navpoint* ini menggunakan graf berarah.

Pada gambar di atas, setiap *Navpoint* (*Alpha*, *Beta*, *Gamma*) ditandai dengan lingkaran berhuruf N, dan symbol di bagian kiri atas merupakan tempat pemain memulai misi, biasanya disebut sebagai *drop point*.



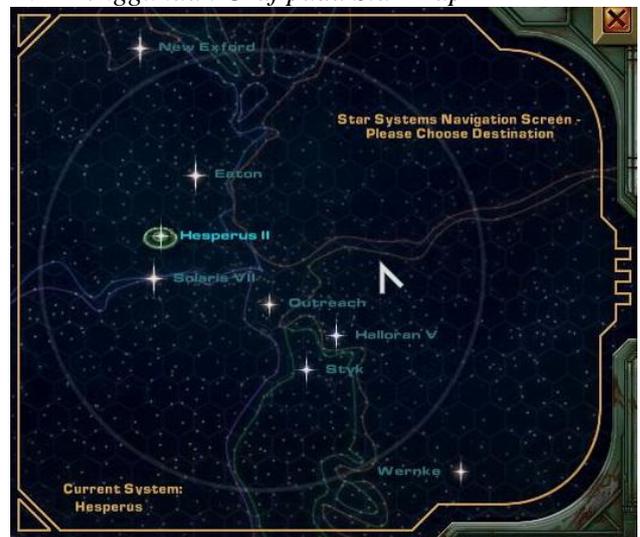
Gambar 12: Penjelasan penggunaan *Navpoint*
D. Penerapan Graf pada *Star Map* dan Pohon Berakar pada Alur *Campaign*

D.1 Penggunaan Pohon Berakar dalam Struktur *Campaign*

Penggunaan Pohon Berakar di sini terdapat pada mode *Campaign*, antara lain penyelesaian misi dari faksi-faksi yang ada (*Davion* atau *Steiner*) yang akan menentukan misi mana yang dapat dimainkan berikutnya atau tidak, dan tiga *ending* yang berbeda. Hal tersebut dapat dijelaskan dengan gambar sebagai berikut:



Gambar 13: Pohon pada Struktur *Campaign*
D.2 Penggunaan Graf pada *Star Map*



Gambar 14: Tampilan *Star Map*

Penggunaan graf lainnya terdapat pada peta *Star System*, yang menggambarkan dunia *Battletech*. Pada *Star System* ini, lokasi baru tidak akan muncul sampai pemain menyelesaikan semua misi pada lokasi tertentu. Berhubungan dengan struktur campaign, menyelesaikan semua misi di suatu tempat juga dapat membuka ataupun mengunci cabang misi lainnya. Secara garis besar, dapat dijelaskan dengan graf sebagai berikut

HalloranV->Eaton->SolarisVII->Styk->HesperusII-
>SolarisVII->Talon/Wernke->New
Exford->Tharkad
 |+>New Avalon->New Canton
 |+>Carse

IV. BEBERAPA KESALAHAN UMUM

Adapun beberapa kesalahan umum yang masih ada hingga *update* terbaru antara lain *Pathfinding* AI yang masih belum sempurna. Misalnya, jika Mech pemain menghalangi jalan unit kawan, maka unit kawan tersebut perlu beberapa waktu untuk mencari jalan keluar. *Pathfinding* AI juga masih belum mengimplementasikan teori *shortest way* sehingga jalan yang ditempuh bukan merupakan jalan yang tersingkat, melainkan jalan yang sudah diprogram oleh developer (*fixed route*).

V. KESIMPULAN

Penerapan teori struktur diskrit banyak ditemukan tidak pada Mechwarrior 4: Mercenaries saja, melainkan video game lainnya secara keseluruhan. Menerapkan pohon berakar pada *campaign/story* mode suatu game dapat mencegah sebuah game berjalan terlalu linear. Penggunaan graf dapat memudahkan pemain untuk menavigasi suatu *map* pada game.

DAFTAR REFERENSI

- [1] http://en.wikipedia.org/wiki/Discrete_mathematics
Jumat, 9 Desember 2011 pukul 8.24 WIB
- [2] Munir, Rinaldi. 2005. Matematika Diskrit. Bandung: Penerbit Informatika
- [3] http://en.wikipedia.org/wiki/MechWarrior_4:_Mercenaries
Jumat, 9 Desember 2011 pukul 7.30 WIB
- [4] http://en.wikipedia.org/wiki/Graph_%28mathematics%29
Sabtu, 10 Desember 2011 pukul 14.37 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, 9 Desember 2011

ttd



Sandy Gunawan Tanuwijaya, 13510025