

Penerapan Pohon dalam Kombinasi Gerakan Karakter Game

Daniel Christian P. B / 13515020
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13515020@std.stei.itb.ac.id

Abstrak—Banyak jenis *video game* telah dibuat di masa kini. Beberapa darinya memiliki sebuah fitur yang apabila kita menggunakan tombol yang berbeda, maka serangan yang dikeluarkan oleh karakter yang kita gunakan juga akan berbeda, dan dari berbagai jenis tombol yang ada, kita dapat mengkombinasikannya untuk menghasilkan suatu gerakan yang akan berbeda juga. *Set* gerakan yang berbeda tersebut merupakan aplikasi dari pohon keputusan yang tidak disadari oleh pemainnya. Makalah ini akan menjelaskan aplikasi pohon keputusan dalam Kombinasi atau *Set* gerakan karakter dalam sebuah *game*.

Keywords —*game*, *set* gerakan, pohon keputusan.

I. PENDAHULUAN

Video game sudah merambah ke seluruh elemen masyarakat, mulai dari yang muda sampai yang tua bisa merasakan kesenangan ketika mereka bermain sebuah *video game*.

Awal dari sebuah *video game* sendiri dimulai pada akhir tahun 1940-an. Pada saat itu *video game* terletak pada awal tabung sinar katoda berbasis pertahanan peluru kendali sisten. Semua program tersebut kemudian akan diadaptasi ke dalam permainan sederhana di era tahun 1950-an. Lalu mulai berkembang banyak permainan komputer pada akhir tahun 1950-an hingga 1960-an. Setelah periode tersebut *video game* mulai menyimpang ke berbagai *platform* contohnya *arcade*, konsol, pribadi computer, dan permainan genggam. *Platform* tersebut kemudian terbagi-bagi lagi, untuk konsol banyak *platform* yang pernah dikembangkan oleh berbagai pengembang game, contohnya Sony pernah mengembangkan *platform Playstation*, *Playstation 2*, *Playstation 3*, dan *platform* terbarunya yang bernama *Playstation 4*. Contoh lain yang cukup terkenal pada era ini adalah perusahaan *Microsoft* yang pernah mengembangkan *platform Xbox*, *Xbox 360*, dan *platform* terbarunya yang masih bertahan di pasaran saat ini adalah *Xbox One*. Kedua *platform* yang dikembangkan dua perusahaan yang berbeda ini masing-masing memiliki kontroler yang agak berbeda, dimana kontroler untuk konsol yang dikembangkan *Microsoft*, *Xbox*, memiliki tombol X, Y, A, dan B yang terletak di sebelah kanan kontroler, sedangkan untuk konsol yang dikembangkan oleh Sony, yaitu PS4, memiliki tombol □

(kotak), O (bulat), X, dan ▲ (segitiga) di tempat yang sama. Meskipun kedua set tombol ini memiliki kegunaan yang serupa, namun untuk pemain yang baru mulai memainkan konsol, akan terasa berbeda. Perbedaan menonjol lainnya adalah peletakkan *analog* dan *D-Pad*, dimana *Xbox* dan PS4 memiliki letak *analog* dan *D-Pad* kiri yang berlawanan.



Gambar 1.1 Kontroler PS4



Gambar 1.2 Kontroler Xbox 360

Game-game ber-*genre fighting* seperti *Tekken* maupun *Hack and Slash* seperti *Dynasty Warrior* memiliki set aksi yang berbeda untuk setiap tombol. Namun uniknya,

apabila tombol tersebut ditekan bersama-sama ataupun ditekan setelah tombol sebelumnya ditekan, akan membuat gerakan yang berbeda pula. Pada makalah ini saya akan membahas bagaimana hal tersebut bisa dilakukan dan menjelaskan bahwa hal tersebut merupakan salah satu contoh aplikasi dari pohon keputusan. Dan meskipun yang saya tunjukkan barusan hanya untuk konsol, tapi yang akan saya bahas tidak hanya sebatas *game-game* konsol.

II. DASAR TEORI

A. Pohon

Definisi dari pohon yang dimaksud adalah graf tak-berarah terhubung yang tidak mengandung sirkuit. Misalnya $G = (V, E)$ adalah pohon, maka V tidak boleh berupa himpunan kosong, namun E boleh kosong.

B. Sifat-Sifat Pohon

Sifat-sifat pohon dinyatakan di bawah ini.

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 (jembatan adalah sisi yang bila dihapus menyebabkan graf terpecah menjadi dua komponen).

C. Pohon Merentang

Misalkan $G = (V, E)$ adalah graf tak-berarah terhubung yang bukan pohon, yang berarti di G terdapat beberapa sirkuit. G dapat diubah menjadi pohon $T = (V_1, E_1)$ dengan cara memutuskan sirkuit-sirkuit yang ada. Dengan cara memilih satu buah sirkuit yang ada, lalu hapus satu buah sisi dari sirkuit tersebut. G akan tetap terhubung dan jumlah sirkuitnya berkurang satu. Lalu bila proses ini terus-menerus diulang sampai semua sirkuit di G hilang, maka G akan menjadi T , yang dinamakan **pohon merentang**. Pohon merentang didefinisikan hanya untuk graf terhubung, karena pohon selalu terhubung.

Sisi pada pohon merentang – disebut **cabang** – adalah sisi dari graf semula, sedangkan **tali-hubung** dari pohon adalah sisi dari graf yang tidak terdapat di dalam pohon merentang. Pada graf terhubung dengan m buah sisi dan n buah simpul terdapat $n-1$ buah cabang dan $m - n + 1$ buah tali. Himpunan tali-hubung beserta simpul yang bersisian dengannya disebut **komplemen pohon**.

D. Pohon Merentang Minimum

Jika Graf G adalah graf berbobot, maka bobot pohon merentang T dari G didefinisikan sebagai jumlah bobot

semua sisi T . Di antara semua pohon merentang di G , pohon merentang yang berbobot minimum – dinamakan **pohon merentang minimum** – merupakan pohon merentang yang paling penting. Terdapat dua buah algoritma membangun pohon merentang minimum. Yang pertama adalah **algoritma Prim**, dan yang kedua adalah **algoritma Kruskal**.

1. Algoritma Prim

Algoritma Prim membentuk pohon merentang minimum langkah per langkah. Pada setiap langkah kita mengambil sisi dari graf G yang mempunyai bobot minimum namun terhubung dengan pohon merentang minimum T yang telah terbentuk. Berikut adalah penyampaian caranya per langkah :

1. Ambil sisi dari graf G yang berbobot minimum, masukkan ke dalam T .
2. Pilih sisi (u, v) yang mempunyai bobot minimum dan bersisian dengan simpul di T , tetap (u, v) tidak membentuk sirkuit di T . Tambahkan (u, v) ke dalam T .
3. Ulangi 2 sebanyak $(n - 2)$ kali.

Jumlah langkah seluruhnya di dalam algoritma Prim adalah $1 + (n-2) = n - 1$ yaitu sebanyak jumlah sisi di dalam pohon merentang dengan n buah simpul.

2. Algoritma Kruskal

Perbedaan prinsip antara algoritma Prim dan Kruskal adalah : jika pada algoritma Prim sisi yang dimasukkan ke dalam T harus bersisian dengan sebuah simpul di T , maka pada algoritma Kruskal sisi yang dipilih tidak perlu bersisian dengan sebuah simpul di T asalkan penambahan sisi tersebut tidak membentuk sirkuit. Berikut adalah penyampaian caranya per langkah dengan asumsi bahwa setiap sisi dari graf sudah diurut menaik berdasarkan bobotnya:

1. T masih kosong
2. Pilih sisi (u, v) dengan bobot minimum yang tidak membentuk sirkuit di T . Tambahkan (u, v) ke dalam T .
3. Ulangi langkah 2 sebanyak $(n - 1)$ kali.

E. Pohon Berakar

Pohon berakar adalah pohon yang sebuah simpulnya diperlukan sebagai akar dan sisi-sisinya diberi arah sehingga menjadi graf berarah. Akar mempunyai derajat-masuk sama dengan nol dan simpul-simpul lainnya berderajat-masuk sama dengan satu. Simpul yang mempunyai derajat-keluar sama dengan nol disebut dengan daun atau simpul terminal. Setiap simpul di pohon dapat dicapai dari akar dengan sebuah lintasan tunggal. Sebagai perjanjian, arah sisi di dalam pohon dapat dibuang, karena setiap simpul di pohon harus dicapai dari akar. Maka lintasan di dalam pohon berakar selalu dari “atas” ke “bawah”. Sembarang pohon tak-berakar dapat dibuat menjadi pohon berakar dengan memilih sebuah simpul sebagai akar, namun pemilihan simpul yang berbeda akan menghasilkan pohon berakar yang berbeda pula.

Ada beberapa terminology yang dipakai untuk pohon berakar

Anak (*child*) dan Orangtua (*parent*)

Misalkan X adalah sebuah simpul di dalam pohon berakar. Simpul y dikatakan **anak** simpul x jika ada sisi dari simpul x ke y , dengan x sebagai **orangtua** dari y .

Lintasan (*path*)

Lintasan dari simpul v_1 ke simpul v_k adalah runtutan simpul-simpul $v_1, v_2, v_3, \dots, v_k$. **Panjang lintasan** adalah jumlah sisi yang dilalui dalam suatu lintasan.

Keturunan (*descendant*) dan Leluhur (*ancestor*)

Jika terdapat lintasan dari simpul x ke simpul y dalam pohon, maka x adalah **leluhur** dari simpul y , dan y adalah **keturunan** dari simpul x .

Saudara kandung (*sibling*)

Simpul yang berorangtua sama adalah **saudara kandung**.

Upapohon (*subtree*)

Misalkan x adalah simpul di dalam pohon T . Yang dimaksud dengan **upapohon** dengan x sebagai akarnya ialah upagraf $T' = (V', E')$ sedemikian sehingga V' mengandung x dan semua keturunannya dan E' mengandung sisi-sisi dalam semua lintasan yang berasal dari x .

Derajat (*degree*)

Derajat sebuah simpul pada pohon berakar adalah jumlah upapohon pada simpul tersebut.

Daun (*leaf*)

Daun adalah simpul yang berderajat nol.

Simpul dalam (*internal nodes*)

Simpul dalam adalah simpul yang mempunyai anak.

Aras (*level*) atau Tingkat

Akar mempunyai aras = 0. Sedangkan aras simpul lainnya = 1 + panjang lintasan dari akar ke simpul tersebut.

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

Tinggi atau kedalaman adalah aras maksimum dari suatu pohon.

F. Pohon n -ary

Pohon n -ary adalah pohon berakar yang setiap simpul cabangnya mempunyai paling banyak n buah anak. Jika $n = 2$, pohonnya disebut pohon biner. Pohon n -ary dikatakan **teratur** atau **penuhi** jika setiap simpul cabangnya mempunyai tepat m buah anak. Dalam terapannya, pohon n -ary digunakan sebagai model yang mempresentasikan suatu struktur.

Pohon n -ary teratur adalah pohon yang setiap simpulnya tepat mempunyai m buah anak. Pada pohon n -ary teratur dengan tinggi h , jumlah daun adalah n^h .

III. PENERAPAN POHON DALAM KOMBINASI GERAKAN KARAKTER GAME

Di bab ini, saya akan membahas 3 jenis atau 3 seri *game* yang berbeda, yaitu seri *game Dynasty Warrior* dan *game* yang memiliki mekanika sama, seri *Tekken* dan *Skullgirls*, dan yang terakhir adalah *Elsword Online*.

A. Dynasty Warrior

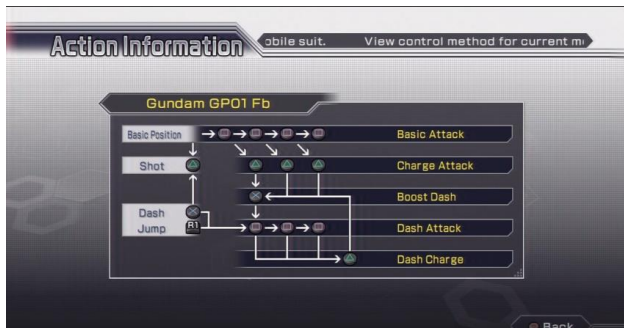
Contoh *game* pertama, yang mengaplikasikan pohon, yang akan saya bahas adalah seri *game Dynasty Warrior* dan *game* yang memiliki tipe mekanika *game* yang sama seperti *One Piece Pirate Warriors* dan *Dynasty Warrior Gundam*. Pada *game-game* ini serangan dapat dilakukan dengan tombol kotak dan segitiga. Tombol kotak dan segitiga apabila ditekan akan menghasilkan serangan yang berbeda. Saya akan ambil contoh dari seri *game Dynasty Warrior Gundam*. Di seri *game* ini tombol segitiga digunakan untuk menembakkan *laser beam* atau serangan *range*, sedangkan tombol kotak digunakan untuk melakukan serangan dengan *beam saber* atau serangan *melee*. Contoh tersebut berlaku untuk semua *game* yang memakai mekanik seperti *Dynasty Warrior* hanya *output*-nya saja yang berbeda. Tombol kotak dan segitiga ini dapat ditekan berulang kali untuk melakukan *basic combo* atau kombo dasar. Kombo dasar yang terdiri hanya atas kotak saja, apabila diselipkan *input* dari tombol segitiga, atau sebaliknya, akan menghasilkan serangan yang berbeda dari kombo dasar. Berikut adalah contoh dari kombo dasar beserta penyelipan kombo pada beberapa *game* seri ini:



Gambar 3.1. Move List Dynasty Warrior 7
(Sumber: <http://www.lordyuanshu.com/dynasty-warriors-7-ex-moves-special-charge-attacks>)



Gambar 3.2. Move List Samurai Warrior 4
(Sumber: <http://www.playstationlifestyle.net/2014/10/28/samurai-warriors-4-review-learn-kill-ps4/#/slide/1>)



Gambar 3.3. Move List Dynasty Warrior Gundam 3
(Sumber : youtube.com)

Apabila dianggap bahwa kombo dasar tombol kotak merupakan pohon condong kanan. Maka apabila kita menekan segitiga setelah kita menekan tombol kotak pertama kali, maka pada simpul pertama akan ada anak kedua, apabila kita menekan segitiga setelah menekan kotak dua kali, maka akan ada anak segitiga pada simpul tersebut.

B. Tekken dan Skullgirls

Game Tekken atau game ber-genre fighting lainnya seperti Skullgirls dan Street Fighter juga menerapkan konsep pohon ini dalam game mereka. Konsep yang mereka miliki berbeda dengan apa yang dimiliki oleh seri Dynasty Warrior. Di game Tekken, biasanya tombol X dan O akan digunakan untuk membuat karakter melakukan tendangan dan tombol kotak dan segitiga akan digunakan untuk membuat karakter melakukan pukulan. Yang dibedakan hanyalah tombol X untuk tendangan ringan dan tombol O untuk tendangan yang keras, serta tombol kotak untuk pukulan yang keras dan tombol segitiga untuk pukulan yang ringan. Dan apabila kedua tombol tersebut ditekan akan melakukan gerakan yang berbeda. Konsep yang diterapkan di game Tekken lebih kompleks daripada di seri Dynasty Warrior, Karena pada game fighting seperti ini, tombol arah juga dapat dikombinasikan untuk melakukan gerakan yang baru. Namun konsep dimana kombo dasar yang diselipkan suatu tombol masih diterapkan di game fighting seperti Tekken ini. Berikut adalah contoh dari move list game fighting:

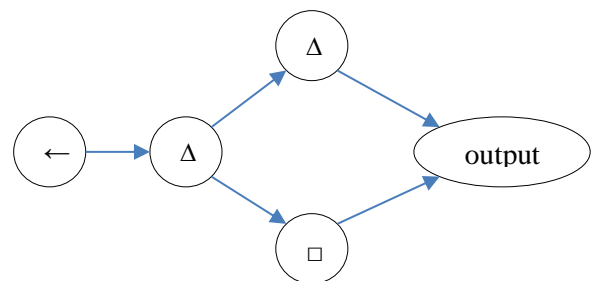


Gambar 3.4. Move List Peacock dari Skullgirls
(Sumber: http://i818.photobucket.com/albums/zz107/t1mmy_ssb/AiB/Blogs/skullgirls/Arcade%20Controls/peacock_600x600.jpg)



Gambar 3.5. Move List Asuka Kazama dari Tekken 5
(Sumber: <https://ytimg.googleusercontent.com/vi/5HN3nMAp2XI/0.jpg>)

Dan berikut adalah contoh bentuk pohon yang dapat dibuat untuk mengilustrasikannya

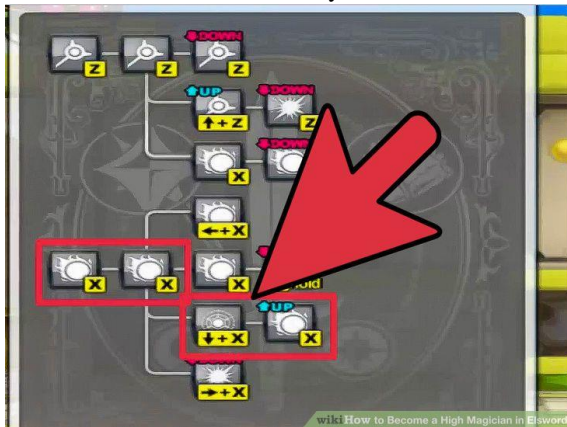


Gambar 3.6. Ilustrasi move Swallow Mallet Elbow dari Gambar 3.5.

C. Elsword Online

Konsep pohon tersebut tidak hanya diterapkan di game-game offline saja, namun juga bias diterapkan untuk game MMORPG (Massively Multiplayer Online Role-Playing Game) seperti contohnya Elsword Online. Konsep yang dipakai persis sama dengan yang dipakai di seri Dynasty

Warrior. Berikut adalah contohnya:



Gambar 3.7. Move List Alisha dari Elsword Online
(Sumber: <http://www.wikihow.com/Become-a-High-Magician-in-Elsword>)



Gambar 3.8. Move List Elsword dari Elsword Online
(Sumber: <https://steamcommunity.com/sharedfiles/filedetails/?id=164047182>)

IV. KESIMPULAN

Konsep pohon bisa diterapkan dimana saja, termasuk dalam sebuah *game* dimana karakternya bisa melakukan banyak serangan hanya dengan 8 tombol yang ada di kontroler sebuah konsol. Pohon ini bisa diimplementasikan untuk memberikan semua kemungkinan kombinasi yang ada.

V. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa yang telah memampukan penulis untuk menyelesaikan makalah ini dengan baik. Selanjutnya Penulis juga mengucapkan terima kasih kepada orang tua

penulis serta sahabat penulis yang telah menemani penulis saat menulis makalah ini. Penulis juga ingin berterima kasih kepada rekan-rekan yang telah mendukung penulis dan memberikan nasihat dalam menyelesaikan tugas dalam perkuliahan di ITB.

Penulis juga mengucapkan terima kasih kepada Ibu Dra. Harlili dan Bapak Dr. Ir. Rinaldi Munir selaku dosen mata kuliah Matematika Diskrit yang telah memberikan tugas ini serta membekali penulis dengan materi yang berkaitan dengan Matematika Diskrit yang digunakan dan dimanfaatkan dalam penulisan makalah ini. Tidak lupa penulis juga berterima kasih kepada pihak-pihak yang secara langsung maupun tidak langsung telah membantu dalam menyelesaikan makalah ini.

REFERENSI

- [1] Munir, Rinaldi. 2006. Matematika Diskrit Edisi Keempat. Bandung : Penerbit Informatika.

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, 8 Desember 2016

Daniel Christian Pradipta - 13515020