

Penerapan Pohon Untuk Membantu Intelegensi Buatan Membuat Keputusan Pada “Bot”

Kelvin Kristian 13516101

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

kelvinkristian4@gmail.com, 13516101@std.stei.itb.ac.id

Abstract—Intelegensi buatan adalah suatu software yang membantu manusia menyelesaikan masalah, berencana dan masih banyak lagi. Cara Intelgensi buatan melakukan semua pekerjaan itu adalah bisa dengan menggunakan *hardcode* atau *deeplearning*. Kedua metode tersebut tentu saja akan menghasilkan pohon keputusan yang berbeda di mana pohon keputusan metode *hardcode* akan terbatas sedangkan *deeplearning* menghasilkan pohon keputusan yang beragam dan dapat menjadi lebih pintar daripada metode *hardcode*.

Keywords—*bot*, *deeplearning*, *intelegensi buatan*, *pohon keputusan*.

I. PENDAHULUAN

Perkembangan teknologi semakin pesat seiring berjalannya waktu dan hal tersebut sangat terlihat jelas di abad ke-21 ini. Untuk lebih spesifiknya yaitu diatas tahun 2010, mulai banyak bermunculan perangkat-perangkat canggih yang membantu mempermudah semua pekerjaan manusia sehingga semua pekerjaan dapat terselesaikan dengan baik, benar dan efisien. Dampak dari hal tersebut adalah produktivitas manusia semakin meningkat karena semakin banyak pekerjaan manusia yang terselesaikan meski sesulit apapun pekerjaannya itu.

Mungkin secara awam perkembangan teknologi tersebut hanya banyak terlihat pada perangkat-perangkat canggih yang beredar luas yang notabene adalah perangkat keras atau *hardware*. Namun hal yang perlu diperhatikan adalah sesuatu yang terdapat di dalam perangkat keras itu sendiri yaitu perangkat lunaknya atau *software* karena perangkat lunak lah yang berkembang lebih cepat daripada perangkat kerasnya itu sendiri. Perangkat lunak ini terdiri dari banyak komponen yang mengandung banyak faktor. Salah satu komponen yang menjalankan perangkat lunak tersebut adalah intelegensi buatan atau *Artificial Intelligence*.

Intelegensi buatan adalah sebuah software yang dapat melakukan sesuatu secara pintar layaknya manusia seperti mengenali suara, belajar, berencana, menyelesaikan masalah memanipulasi objek dan masih banyak lagi. Intelegensi buatan tidak hanya terdapat pada perangkat-keras perangkat-keras tersebut namun terdapat di banyak tempat yaitu mulai dari hal yang simpel seperti fitur *AutoCorrect* pada *Microsoft Word* sampai hal yang rumit seperti *bot* pada *game*. *Bot* adalah sebuah istilah atau sebutan pada *game* yang artinya adalah robot. *Bot* ini berjalan menggunakan intelegensi buatan yang berarti pergerakannya murni diatur seluruhnya oleh komputer. Intelegensi buatan secara garis besar dapat dibedakan menjadi 2

jenis di mana ada intelegensi yang di-*hardcode* seperti menggunakan banyak *if-else* atau jenis percabangan kasus lainnya dan ada juga yang belajar secara alami melalui metode *Deep Learning* (metode yang paling *advance* saat ini).

Intelegensi buatan yang memanfaatkan metode *Deep Learning* dalam *Machine Learning*-nya tentu saja akan mempelajari suatu hal secara menyeluruh dengan mencoba segala kemungkinan yang ada dan menjadikannya data untuk kemudian di evaluasi. Dalam proses pembelajarannya, Intelegensi buatan secara implisit akan menggunakan *Decision Tree* atau pohon pengambilan keputusan di mana nantinya saat dievaluasi, cabang-cabang yang memiliki nilai yang tidak diinginkan oleh *user* akan dihapus sehingga Intelegensi Buatan akan menghasilkan hasil yang “pintar” seperti manusia. Salah satu penerapan pembuatan Intelegensi buatan ini adalah di dalam pembuatan *bot* yang terdapat pada *game Ragnarok Online*. Pembuatan *bot* yang diciptakan oleh *OpenKore* dalam *Ragnarok Online* masih tergolong dalam *hardcode* karena terdapat banyak file pengaturan *bot* yang *input*-nya berdasarkan *input* dari *user* sehingga *bot* tidak akan melakukan aksi-aksi lain diluar *code* akibatnya *bot* belum tentu bisa menyelesaikan semua masalah dalam keberjalanannya karena *code* yang digunakan tidak dinamis sesuai lingkungannya. Namun berbeda dengan *bot* buatan perusahaan non-profit *OpenAI* yang mengimplementasikan *bot*-nya dalam *game DOTA 2* di mana *bot* tersebut dibuat “tidak pintar” dari awalnya dan dilatihkan selama 2x7x24 jam nonstop atau selama 2 minggu untuk menemukan algoritma yang memang paling efisien untuk menyelesaikan sebuah pertandingan.

Karakteristik *game Ragnarok Online* dan *DOTA 2* harus diketahui terlebih dahulu untuk lebih memperjelas. *Ragnarok Online* merupakan sebuah *game MMORPG* yang merupakan kependekan dari *Massively Multiplayer Online Role-Playing Game* di mana *Game* ini memiliki banyak karakter dan peran-peran yang bisa dimainkan secara bersama-sama. Konsep dari *game RPG* adalah memiliki peta yang sangat luas untuk dijelajahi dan memiliki banyak *monster* untuk di lawan, jadi penggunaan *bot* disini digunakan untuk mencari level atau barang yang diinginkan tanpa harus selalu dimainkan secara langsung oleh pemain sedangkan untuk *DOTA 2*, *game* ini berbasis *MOBA* atau *Multiplayer Online Battle Arena* di mana memerlukan strategi yang sangat kompleks untuk dapat menyelesaikan satu pertandingan karena dibutuhkan banyak keputusan yang tidak hanya cepat namun tepat. Penggunaan *bot* digunakan sebagai lawan dari *player* yang berupa komputer.

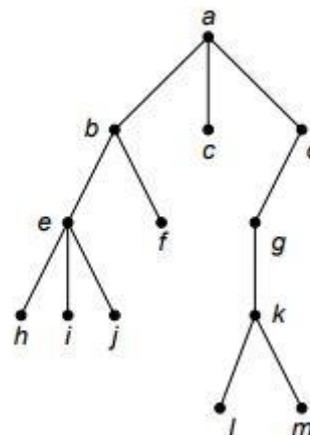
II DASAR TEORI

2.1 POHON

A. Definisi Pohon

Pohon adalah sebuah graf tak-berarah terhubung yang tidak mengandung sirkuit. Misalkan $G = (V, E)$ adalah graf tidak berarah sederhana dan memiliki jumlah simpul sebanyak n . Maka graf tersebut akan ekuivalen dengan beberapa pernyataan berikut:

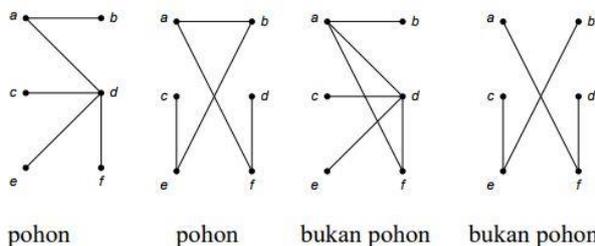
1. G adalah sebuah pohon.
2. Setiap pasang simpul di dalam G terhubung dengan lintasan tunggal.
3. G memiliki simpul saling terhubung dan memiliki m buah sisi dengan $m = n - 1$.
4. G tidak mengandung sirkuit dan memiliki $m = n - 1$ buah sisi.
5. G tidak mengandung sirkuit dan penambahan satu sisi pada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



Gambar 2.2 Pohon Berakar

Sumber:

[http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20\(2013\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20(2013).pdf) slide 24 diakses pada 2 Desember 2017



Gambar 2.1 Jenis Pohon

Sumber:

[http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20\(2013\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20(2013).pdf) slide 2 diakses pada 2 Desember 2017

B. Pohon Merentang

Pohon merentang dari suatu graf yang terhubung adalah upagraf atau subgraph merentang yang berupa pohon. Pohon merentang diperoleh dengan memotong sirkuit di dalam graf. Setiap graf yang terhubung mempunyai paling sedikit satu buah pohon merentang di dalamnya. Graf yang tidak terhubung dengan k komponen akan mempunyai k buah hutan merentang yang disebut hutan merentang (*spanning forest*). Hutan (*forest*) adalah kumpulan pohon yang saling lepas atau graf tidak terhubung yang tidak mengandung sirkuit. Setiap komponen di dalam graf terhubung adalah pohon.

C. Pohon Berakar

Pada dasarnya sebuah pohon akan memiliki minimal sebuah akar. Dari akar tersebut dapat muncul cabang-cabang yang jika ditelusuri kembali secara *bottom-up* maka dapat mencapai akar itu lagi. Banyaknya akar yang muncul tergantung dari jenis pohonnya memiliki berapa *-ary* jika *binary* berarti pohon tersebut memiliki maksimal 2 simpul yang keluar dari akarnya.

a. Orangtua dan Anak

Jika dilihat pada gambar 2.2 diatas. Graf dapat mengelompokannya menjadi orang tua dan anak. Contoh anak adalah simpul b, c, d di mana mereka memiliki orangtua a . Jadi bisa disimpulkan bahwa anak adalah simpul-simpul yang berasal dari simpul sebelumnya sedangkan orang tua bisa tidak berasal dari suatu simpul dan dapat berdiri sendiri. Disini simpul b dan d juga berperan sebagai orang tua di mana b adalah orang tua dari e dan f sedangkan d adalah orang tua dari g . Suatu simpul akan berperan sebagai orang tua ketika simpul tersebut mengeluarkan cabang-cabang baru

b. Lintasan

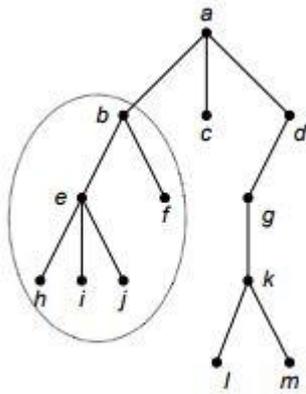
Lintasan adalah sisi yang ditempuh dari suatu simpul sumber menuju simpul tujuan. Jika melihat gambar 2.2 maka lintasan dari a ke j adalah a, b, e, j dan panjang lintasannya adalah 3 karena melalui 3 sisi.

c. Saudara kandung

Saudara kandung adalah kumpulan simpul-simpul yang memiliki orangtua yang sama dengan simpul yang sedang diteliti. Pada gambar 2.2, f merupakan saudara kandung dari e karena sama-sama berasal dari orangtua b namun g bukanlah saudara kandung dari f karena meskipun memiliki tingkat yang sama namun orang tuanya berbeda.

d. Upapohon

Upapohon (*subtree*) adalah sebuah pohon yang merupakan bagian dari suatu pohon yang lebih besar. Contohnya adalah pada gambar berikut, bagian yang dilingkari adalah pohon bagian dari pohon yang lebih besar.



Gambar 2.3 Upapohon

Sumber:

[http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20\(2013\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20(2013).pdf) slide 26 diakses pada 2 Desember 2017

e. Derajat

Derajat adalah sebuah besaran yang menunjukkan jumlah upapohon yang dihasilkan dari suatu simpul. Contohnya adalah pada gambar 2.2, simpul a memiliki derajat 3 karena memiliki 3 upapohon yaitu b , c dan d . Sedangkan c memiliki derajat 0 karena tidak memiliki upapohon yang berasal dari simpul c .

f. Daun

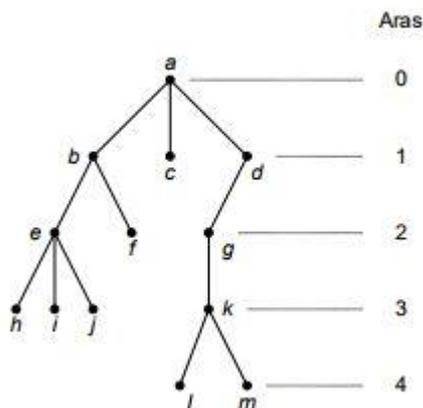
Daun adalah kumpulan simpul-simpul yang memiliki derajat 0. Pada gambar 2.2 terlihat contoh dari daun yaitu pada simpul h , i , j , f , c , l dan m .

g. Simpul Dalam

Simpul dalam adalah kumpulan simpul-simpul yang berada di dalam sebuah pohon kecuali akar utama yaitu yang berada pada tingkat 0. Contoh simpul dalam pada gambar 2.2 adalah simpul b , d , e , g dan k .

h. Aras

Aras adalah tingkatan pada suatu pohon dimana akar utama akan memiliki tingkat 0, anaknya akan memiliki tingkat 1 dan begitu seterusnya.



Gambar 2.4 Aras

Sumber:

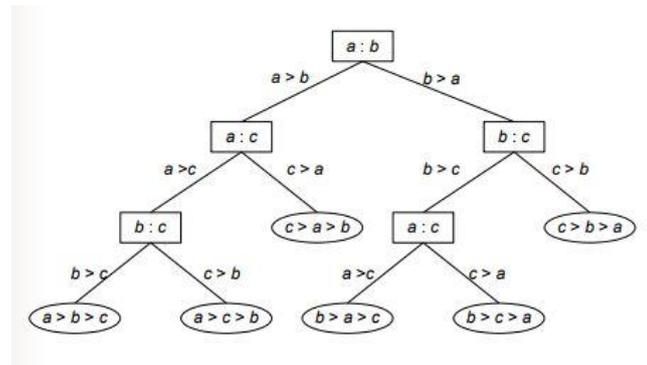
[http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20\(2013\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20(2013).pdf) slide 29 diakses pada 2 Desember 2017

i. Tinggi

Tinggi adalah Aras maksimum pada suatu pohon. Pada gambar 2.4 Tingginya adalah 4.

D. Pohon Keputusan

Pohon keputusan merupakan sebuah pohon yang dibuat untuk mencari keputusan berdasarkan keadaan-keadaan yang diberikan. Dalam proses pengambilan keputusan, intelegensi buatan akan menggunakan pohon keputusan secara implisit di dalam algoritmanya. Pohon keputusan akan mengambil keputusan yang tepat sesuai kondisi yang dialami oleh intelegensi buatan sehingga intelegensi buatan dapat melakukan aksi yang sesuai untuk menghadapi kondisinya tersebut.



Gambar 2.5 Pohon Keputusan

Sumber:

[http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20\(2013\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20(2013).pdf) slide 38 diakses pada 2 Desember 2017

E. Ragnarok Online

Ragnarok Online adalah sebuah *game online* yang ber-genre MMORPG yaitu *Massively Multiplayer Online Role-Playing Games*. Permainan ini pertama kali dibuat di Seoul, Korea Selatan oleh Gravity Corporation. Permainan ber-genre RPG biasanya memiliki peta yang sangat luas dan diisi oleh berbagai macam jenis *monster*. Untuk memainkan permainan diperlukan sebuah karakter *virtual* yang akan dimainkan oleh player nantinya. Karakter tersebut dapat memilih *role* atau peran apa yang ingin ia mainkan. Di dalam ragnarok terdapat *Swordsman*, *Archer*, *Acolyte*, *Merchant*, *Thief* dan *Mage* yang nantinya dapat berevolusi menjadi peran yang lebih spesifik dan kemampuan yang lebih hebat. Karakter tersebut harus meningkatkan *level*-nya dan mengumpulkan *equipment* yang langka dan hebat agar bertambah kuat dengan cara membunuh *monster* secara terus menerus.



Gambar 2.6 Attacking Bigfoot

Sumber: <http://jonvilma.com/images/ragnarok-6.jpg> diakses pada 2 Desember 2017



Gambar 2.8 DOTA 2 Gamplay

Sumber: <http://mobafree.com/wp-content/uploads/2017/06/Dota-2-Free-Download-1.jpg> diakses pada 2 Desember 2017

F. DOTA 2

Dota 2 adalah sebuah game yang ber-genre MOBA di mana permainan sangat mengandalkan strategi, *timing* dan *teamwork* yang baik. Permainan ini diluncurkan pada tahun 2013.

Dota 2 dimainkan oleh 10 orang pemain yang di bagi menjadi 2 tim yang berjumlah 5 orang setiap timnya yaitu tim *Radiant* dan tim *Dire*. Markas kedua tim terpisah dan saling berhadapan. Markas tim *Radiant* berada di pojok kiri bawah map sedangkan *Dire* berada di pojok kanan atas map.



Gambar 2.7 Map Dota2

Sumber: <http://i.imgur.com/iqE4fxr.png> diakses pada 2 Desember 2017

Tujuan utama dari permainan ini adalah menghancurkan *ancient* tim musuh. Untuk menghancurkannya, 10 orang tadi akan memilih *hero* yang merupakan sebuah karakter yang dapat berkembang semakin kuat seiring berjalannya permainan. *Hero-hero* yang terdapat di DOTA ada 108 *hero* dengan kemampuan yang berbeda-beda dan memiliki atribut utama yang berbeda-beda. Atribut-atribut tersebut adalah *Strength*, *Agility* dan *Intelligence*. *Hero-hero* tersebut juga dibagi menjadi 5 peran yaitu *support*, *carry*, *initiator*, *disabler*, *nuker*, *durable* dll. Setiap *hero* akan memiliki 4 *skill* pada umumnya yang merupakan gabungan dari *passive skill* dan *active skill* yang bisa diambil setiap level *hero* bertambah tergantung dari *hero* yang digunakan oleh pemain.

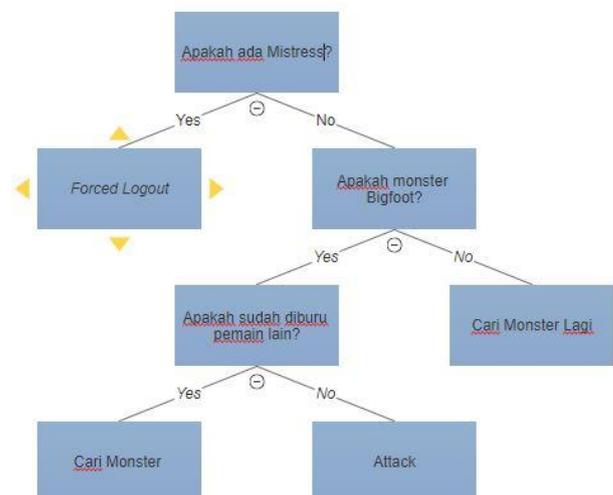
III. PENERAPAN POHON KEPUTUSAN DALAM INTELEGENSI BUATAN PADA BOT

3.1 Penerapan Pohon Keputusan Pada Bot di Game Ragnarok Online

Dalam permainan Ragnarok Online, *Bot* dapat digunakan untuk mengotomatiskan pergerakan karakter mulai dari berpindah, menyerang, melarikan diri, *forced logout* dan lainnya yang tujuannya adalah untuk meningkatkan level dari karakter tersebut tanpa harus dimainkan langsung oleh *user*. *Forced Logout* dilakukan untuk mencegah karakter mati karena jika karakter mati akan mengurangi *experience* yang sudah diraih. Banyak sekali aspek yang digunakan untuk membuat *bot* yang sempurna pada permainan Ragnarok Online.

3.1.1 Attack

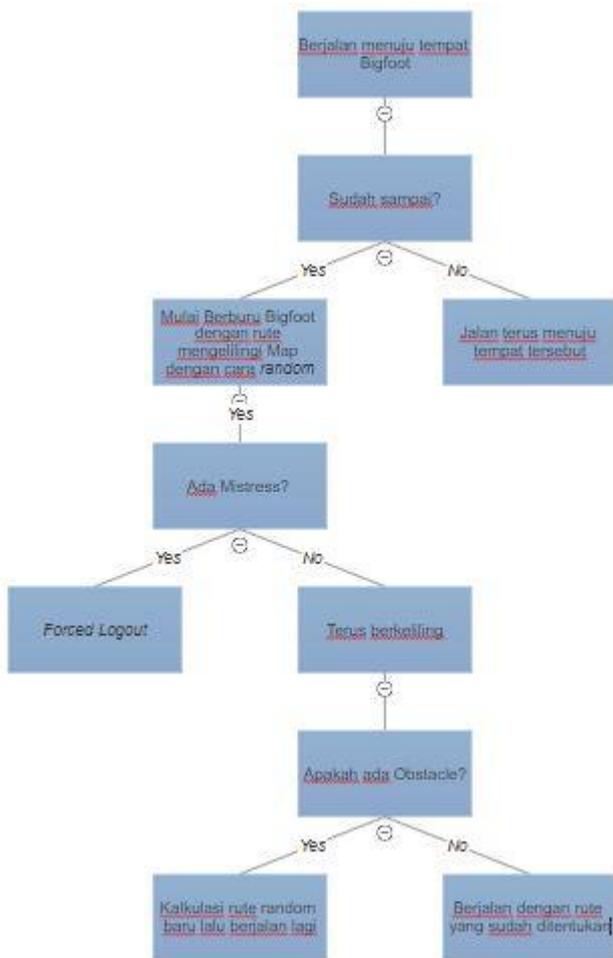
Dalam proses menyerang harus ditentukan terlebih dahulu *monster* apa yang akan dibunuh oleh *bot*, lalu apakah *loot* (barang yang dijatuhkan *monster*) akan diambil atau tidak, apakah ada *boss monster* disekitar pemain, apakah akan melakukan *killsteal* (yaitu memburu *monster* yang sedang diburu pemain lain). Sebagai contoh *monster* bigfoot yang akan menjadi sasaran untuk diburu, tidak melakukan *killsteal*, terdapat boss *monster* di Map tersebut bernama 'Mistress', maka pohon keputusannya :



Gambar 3.1 Attack Decision

3.1.2 Move

Dalam proses berjalan *bot* juga perlu mengetahui rute perjalanan apa yang diinginkan oleh *user*. Maka dari itu *user* sudah mengatur konfigurasinya dari awal sehingga *bot* hanya perlu menjalankannya. Pada konfigurasinya *bot* sudah diatur untuk berjalan menuju tempat di mana *bot* akan melakukan *hunting*. Lalu saat di tempat *hunting*, *bot* juga diatur untuk melakukan perjalanan mengelilingi map dengan rute *random*. Lalu perlu di cek apakah *bot* bertemu *obstacle* seperti tembok atau ujung *map* atau petak-petak yang tidak bisa dilewati lainnya. Misalkan *bot* berasal dari kota Payon, hendak berburu *Bigfoot* namun akan *forced logout* saat bertemu *Mistress*. Maka pohon keputusan *bot* akan seperti berikut:

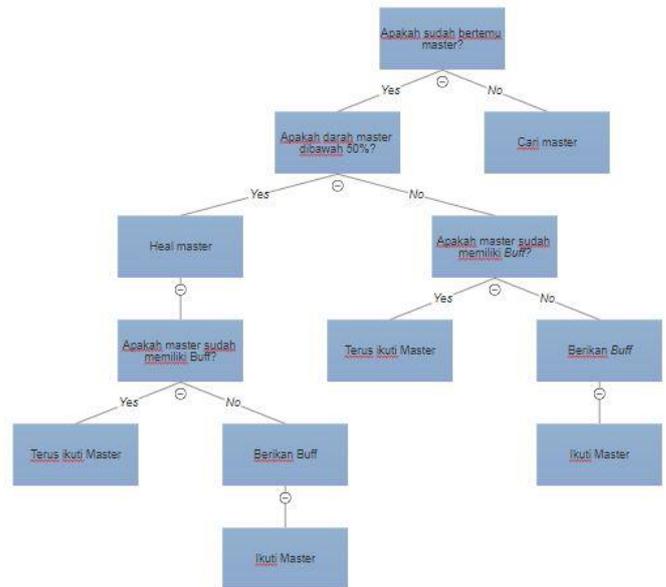


Gambar 3.2 Move Decision

3.1.3 Supporting Priest

Priest adalah sebuah karakter yang memiliki *skill* untuk menambahkan darah karakter temannya dan memberikan *buff* (sesuatu yang dapat menambah kekuatan karakter dalam durasi tertentu). Karakter ini sangat dibutuhkan untuk menambahkan tingkat ketahanan suatu karakter dalam proses *hunting*. *Bot* akan menyebut *priest* sebagai *slave* dan karakter *hunting* sebagai *master*. *Slave* akan selalu mengikuti *master*-nya dan selalu

berusaha mencari *master*-nya bila terpisah pada jarak tertentu. Contoh Pohon Keputusan dari *Supporting Priest* adalah sebagai berikut:



Gambar 3.3 Supporting Priest

Semua pohon diatas dilakukan berulang-ulang sampai program *bot* dimatikan atau sampai suatu kondisi yang diinginkan oleh *user* tercapai.

Masih banyak lagi hal yang dapat dilakukan *bot* pada *Ragnarok online* namun hal-hal di atas adalah hal yang umum dilakukan oleh kebanyakan orang. Dengan berbagai kombinasi *bot* dapat melakukan apapun yang diinginkan *user* secara 'pintar' tanpa perlu pengawasan langsung dan secara terus menerus dari *user*. Namun sayangnya program *Bot OpenKore* ini masih harus dibantu dengan *input user* tidak belajar secara mandiri.

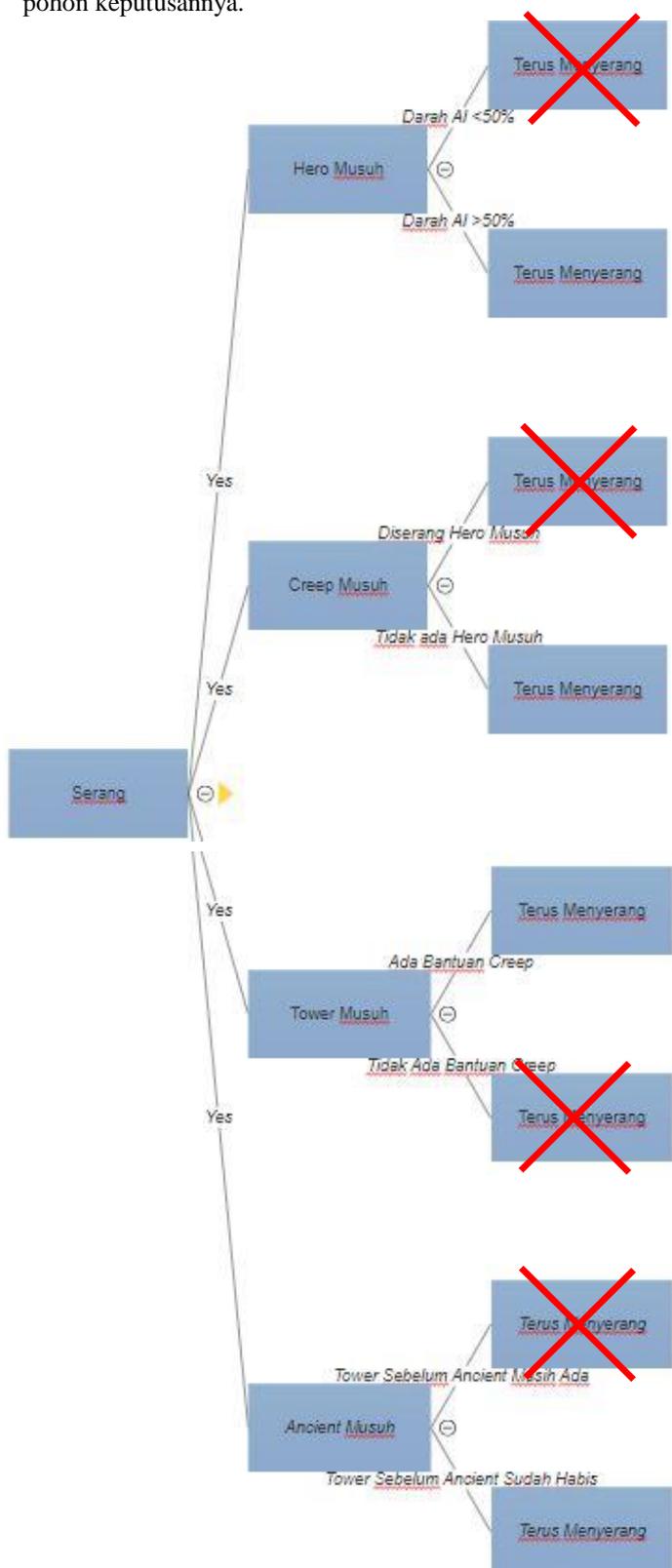
3.2 Penerapan Pohon Keputusan Pada Bot di Game DOTA 2.

Berbeda dengan game *Ragnarok*, *bot* pada permainan *DOTA 2* yang dibuat oleh team *OpenAI* menggunakan metode *deeplearning* sehingga *bot* dibuat sedemikian rupa sehingga ia harus belajar terlebih dahulu untuk menjadi sesuatu yang *user* inginkan. Tugas *user* disini hanya membantu mengevaluasi setiap beberapa interval sekali untuk menilai apakah tindakan yang diambil oleh *bot* itu sesuai dengan keinginan *user* atau tidak. Jika tindakan *bot* tidak diinginkan *user* maka *bot* akan menghapus tindakan tersebut dari *memory* dan tidak akan melakukan tindakan itu lagi. *Bot* ini benar-benar bertindak seperti manusia, ia akan belajar lalu dinilai performanya. Kelebihan dari metode *deeplearning* adalah *bot* hanya perlu dipantau pada awal proses pembelajaran saja karena berikutnya *bot* akan menjadi mandiri. Selain itu setelah *bot* menjadi mandiri *bot* mungkin bisa memberikan opsi-opsi kemungkinan yang mungkin tidak pernah terpikirkan oleh *user* sehingga *bot* dapat melakukan performa yang lebih baik dibandingkan *bot* yang di-*hardcode* seperti *OpenKore* pada *bot Ragnarok*. Melalui proses *deeplearning* pada awalnya memang *bot* akan terlihat seperti 'sangat bodoh' karena *bot* akan langsung mencoba segala

kemungkinan yang ada dan nanti akan mengirimkan data hasil dari percobaannya itu pada *user* untuk dievaluasi. Melihat *bot* ini mencoba segala kemungkinan maka pohon keputusan yang dibuat pasti masih sangat banyak dan mungkin terjadi overlapping antara aksi satu dengan yang lainnya.

3.2.1 Attack

Pada proses menyerang akan terjadi banyak opsi pada awalnya dan nantinya akan dievaluasi oleh *user*. Berikut adalah contoh pohon keputusannya.



GAMBAR 3.4 DOTA LEARNING ATTACK

Tanda Silang pada gambar merupakan *query* yang dihapus dari data karena dirasa tindakan yang merugikan AI atau dapat menyebabkan AI kalah atau mati. Pertama saat menyerang *Hero Musuh*, diperlukan kondisi yang optimal agar *hero* musuh dapat kalah dan *bot* tetap hidup maka dari itu jika keadaan tidak memungkinkan seperti darah *bot* dibawah 50% maka diharapkan untuk berhenti menyerang dan mundur, sebenarnya masih ada faktor lain untuk membuat keputusan agar *bot* tetap menyerang namun langkah tersebut adalah langkah penyelamatan awal. Lalu saat menyerang *creep* musuh juga keadaan *bot* harus diperhatikan apakah *bot* sedang diserang oleh *hero* lain atau tidak untuk mencegah *bot* mati. Untuk menyerang *Tower* juga diperlukan taktik agar kita tetap bertahan maka dari itu diperlukan *creep* yang berfungsi sebagai pertahanan *bot*. Jika *bot* sedang diserang oleh *tower* disarankan untuk mundur. Untuk menyerang *Ancient* kondisinya sama dengan *tower* hanya saja *tower* sebelum *ancient* harus hancur terlebih dahulu agar *Ancient* dapat diserang.

IV. KESIMPULAN

Masih banyak lagi faktor yang perlu diperhatikan pada pembentukan *bot* dari AI menggunakan metode *deeplearning* namun berdasarkan data yang sudah ada. Jika dibandingkan, *bot* ragnarok mungkin memiliki kapabilitas yang tinggi pada masa-masa awal karena pohon keputusan yang dibuatnya pun sudah pasti sehingga tidak mungkin melakukan hal yang tidak diinginkan user namun setelah melalui proses pembelajaran yang banyak, *bot* DOTA dapat mengungguli kepintaran *bot* Ragnarok yang di-*hardcode* karena *bot* DOTA lebih dinamis dengan mencoba berbagai kemungkinan dari pohon keputusan yang dibuat sedangkan *bot* Ragnarok hanya melakukan aksi yang diperintahkan saja sehingga bersifat statis dan tidak akan berkembang. Hal ini dibuktikan saat *Bot* DOTA ditandingkan dengan juara dunia DOTA di mana juara dunia DOTA yang notabene berlatih setiap hari dapat dikalahkan dengan *bot* yang hanya belajar selama 2 minggu. Di sisi lain *bot* ragnarok akan sangat mudah dikalahkan oleh manusia tanpa perlu manusia tersebut menjadi juara atau menjadi yang terhebat dalam Ragnarok.

Pohon Keputusan sangat berperan penting untuk menentukan langkah apa yang hendak diambil oleh AI atau intelgensi buatan. Pada proses *deeplearning*, pohon keputusan digunakan oleh *evaluator* untuk menentukan langkah mana yang paling baik untuk memenangkan permainan. Selain itu Pohon keputusan juga bisa digunakan untuk melakukan penelusuran balik, terhadap langkah-langkah yang dilakukan oleh AI jika terjadi kesalahan dalam pergerakannya sehingga langkah yang diambil bot kedepannya akan lebih optimal dan lebih cepat mencapai tujuannya.

V. UCAPAN TERIMA KASIH

Pertama-tama Penulis mengucapkan terimakasih kepada Tuhan Yang Maha Esa karena atas kuasanya, penulis dapat menyelesaikan penulisan makalah Matematika Diskrit yang

berjudul ‘Penerapan Pohon Untuk Membantu Intelegensi Buatan Membuat Keputusan Pada “Bot” ’. Tidak lupa juga penulis mengucapkan terima kasih yang sebesar-besarnya kepada kedua orang tua penulis yang senantiasa mendukung dan memberi masukan pada penulis dalam penulisan makalah ini.

Penulis juga mengucapkan terima kasih yang sebesar-besarnya kepada Ibu Harlili dan Pak Rinaldi Munir selaku dosen mata kuliah IF 2120 Matematika Diskrit karena sudah memberikan materi yang dapat saya terapkan dalam makalah ini. Tidak lupa juga penulis mengucapkan terima kasih pada semua pihak yang telah membantu penulisan makalah ini yang tidak bisa disebutkan satu per satu.

REFERENCES

- [1] <https://machinelearningmastery.com/what-is-deep-learning/>
- [2] <https://www.techopedia.com/definition/190/artificial-intelligence-ai>
- [3] [http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20\(2013\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Pohon%20(2013).pdf)

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, 2 Desember 2017



Kelvin Kristian
13516101