Penerapan Pohon Keputusan (*Decision Tree*) dalam Permainan *Minesweeper*

Faris Rizki Ekananda 13518125
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13518125@std.stei.itb.ac.id

***Abstrak*—Pohon keputusan (*decision tree*) adalah salah satu alat pembantu pemilihan keputusan berdasarkan kondisi-kondisi dan parameter-paramter yang jelas. Selain dapat diaplikasikan ke dalam kehidupan nyata, pohon ini juga dapat diterapkan pada alur kerja cara menyelesaikan sebuah permainan dengan melakukan pengulangan pada pohon keputusan tersebut setiap beberapa waktu. Pada makalah ini akan dibahas pengaplikasian pohon keputusan untuk menyelesaikan permainan *minesweeper* menggunakan logika dan parameter-parameter tertentu.**

***Kata kunci*—*Minesweeper*, pohon keputusan**

I. PENDAHULUAN

Minesweeper adalah permainan klasik yang berasal dari tahun 1960 hingga 1970-an. Untuk memainkan permainan *single-player* berkategori teka-teki (*puzzle*) ini, pemain harus dapat membuka petak-petak pada *minesweeper* tanpa memicu bom-bom yang tersembunyi sebanyak jumlah tertentu hingga tidak tersisa petak kosong (petak yang tidak memiliki bom) lagi. Pemain akan diberikan petunjuk berupa berapa banyak bom yang terdapat di sekitar (8 arah: atas, bawah, kiri, kanan, atas-kanan, atas-kiri, bawah-kanan, bawah-kiri) petak-petak yang sudah terbuka. Dalam menyelesaikan *minesweeper*, pemain juga dapat menggunakan berbagai macam alat pembantu untuk menyelesaikan permainan ini, seperti bendera untuk menandakan perkiraan letak bom pada suatu petak, atau tanda tanya untuk menandakan petak tersebut masih belum dapat ditentukan apakah berisi bom atau tidak.

Figur 1. Minesweeper

Pada awal permainan, pemain dapat menentukan panjang dan lebar dari area permainan *minesweeper* dan juga jumlah bom yang terdapat dalam area tersebut. Dalam versi *Microsoft Minesweeper*, maksimum area yang dapat dibuat adalah area 30x24 dengan maksimum bom yang dapat kita masukkan sebanyak $(X_{max} - 1)(Y_{max} - 1)$. Beberapa *minesweeper* modern menerapkan aturan yang menyebabkan pemain tidak akan pernah dapat mendapatkan bom pada petak pertama yang ia klik.

Minesweeper adalah permainan yang memanfaatkan logika dan peluang (terutama ketika sudah memasuki *endgame* atau permainan *custom* dengan jumlah bom yang sangat banyak) yang, dengan menggunakan logika tersebut, kita dapat membuat pola-pola untuk menentukan apa yang harus kita lakukan berikutnya untuk dapat menyelesaikan permainan tersebut. Dengan kata lain, kita dapat menggunakan pola-pola tertentu dalam permainan *minesweeper* sebagai strategi pengambilan keputusan untuk mendapatkan kemenangan.

II. TEORI DASAR

A. Graf

Graf secara sederhana adalah sebuah struktur yang terdiri dari objek-objek diskrit yang terkait. Secara matematis, objek-objek tersebut dinamakan simpu-simpul (*vertices*) dan tiap kaitan/hubungan antar simpul disebut sebagai sisi-sisi (*edges*). Secara formal, graf didefinisikan sebagai G = (V,E) dengan G adalah graf, V adalah himpunan tidak kosong dari simpul-simpul (*vertices*) $\{v_1, v_2, v_3, \ldots, v_n\}$, dan E adalah himpunan sisi (*edges*) yang menghubungkan sepasang simpul $\{e_1, e_2, e_3, \ldots, e_n\}$.

Berdasarkan ada tidaknya gelang atau sisi-ganda pada suatu graf, graf dapat dibagi menjadi dua jenis:

1. Graf sederhana (*simple graph*)
   Graf yang tidak memiliki gelang maupun sisi-ganda.
2. Graf tak-sederhana (*unsimple graph*)
   Graf yang memiliki gelang, sisi-ganda, ataupun keduanya.

Berdasarkan ada tidaknya arah pada sisi, graf dapat dibedakan menjadi:

1. Graf tak-berarah (*undirected graph*)
   Graf yang setiap sisinya tidak mempunyai orientasi arah.
2. Graf berarah (*directed graph* atau *digraph*)
   Graf yang setiap sisinya mempunyai arah

Dalam konteks ini, graf yang akan digunakan adalah graf sederhana tak-berarah.

B. Pohon

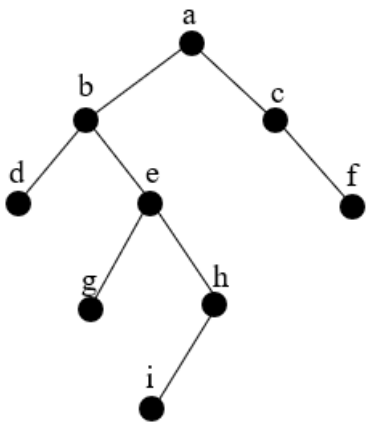
Pohon adalah graf tak-berarah dengan tiap pasangan simpulnya terhubung tepat satu sisi dan tidak mengandung

sirkuit. Hutan adalah kumpulan pohon yang saling lepas, atau graf tidak terhubung yang tidak mengandung sirkuit dengan setiap komponen terhubungnya adalah pohon.

Apabila terdapat G = (V,E) yang merupakan graf tak-berarah sederhana dan jumlah simpul = n, maka definisi lain pohon adalah sebagai berikut:

1. G adalah pohon
2. Setiap pasangan simpul V terhubung dengan lintasan tunggal.
3. G terhubung dan memiliki m = n – 1 buah sisi E.
4. G tidak mengandung sirkuit dan memiliki m = n – 1 buah sisi E.
5. G tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
6. G terhubung dan semua sisinya adalah jembatan.

Apabila salah satu simpul dari pohon diperlakukan sebagai akar dan graf diberikan arah yang bersumber dari akar tersebut, maka pohon dinamakan sebagai pohon berakar (*rooted tree*). Dalam pohon berakar, arah pada sisi-sisinya dapat dihilangkan sesuai dengan perjanjian akar tersebut.



Terdapat beberapa terminologi pada pohon berakar:

1. Anak (*child* atau *children*) dan Orangtua (*parent*)
   Orangtua adalah sumber dari anak-anaknya yang dihubungkan dengan sisi berarah (dari Orangtua ke Anak). Contoh: d dan e adalah anak dari b, c adalah orangtua dari f.
2. Lintasan (*path*)
   Lintasan adalah jalur berarah yang dilewati antara 2 simpul pada pohon berakar. Contoh: lintasan dari b-i adalah b-e-h-i.
3. Saudara Kandung (*sibling*)
   Saudara Kandung adalah Anak yang memiliki Orangtua yang sama. Contoh: d adalah saudara kandung dari e, tetapi bukan saudara kandung dari f.
4. Upapohon (*subtree*)
   Upapohon adalah pohon yang merupakan bagian dari pohon lain yang lebih besar. Contoh: g-e-h-i merupakan upapohon dari pohon tersebut.
5. Derajat (*degree*)
   Derajat sebuah simpul adalah jumlah upapohon/anak yang terdapat pada simpul tersebut. Derajat pada pohon berakar adalah derajat-keluar. Derajat maksimum dari semua simpul merupakan derajat pohon itu sendiri. Contoh: Derajat dari a adalah 2, sekaligus menjadi derajat maksimum pohon tersebut. Karena itu, pohon ini bisa disebut juga sebagai *binary tree*.
6. Daun (*leaf)*
   Daun adalah simpul yang memiliki derajat nol (tidak memiliki anak). Contoh: daun-daun pada pohon tersebut adalah d, g, i, dan f.
7. Simpul Dalam (*internal nodes/vertices*)
   Simpul yang bukanlah akar utama dan memiliki anak disebut sebagai Simpul Dalam. Contoh: e adalah simpul dalam dan memiliki anak g dan h, sementara f bukanlah simpul dalam karena tidak memilik anak.
8. Aras (*level*) atau Tingkat
   Aras/Tingkat adalah jarak dari akar utama menuju suatu simpul dalam di dalam pohon tersebut. Jarak tersebut dihitung dari berapa banyak sisi yang dilewati di antara akar utama dan simpul. Contoh: level dari f adalah 2 karena melewati 2 sisi yaitu a-c dan c-f.
9. Tinggi (*height*) atau Kedalaman (*depth*)
   Tinggi/Kedalaman suatu pohon diambil dari jarak terjauh antara akar utama dengan simpul dalam pohon tersebut. Jarak dihitung dari jumlah sisi yang dilewati untuk mencapai simpul tersebut dari akar utama. Contoh: Tinggi dari pohon tersebut adalah 4 yaitu a-b-e-h-i.

C. Pohon Keputusan

Pohon keputusan adalah salah satu bentuk terapan dari pohon biner. Pohon keputusan bisa digunakan sebagai alat untuk memilih keputusan dengan menggunakan model pohon dengan simpul-simpul yang berupa kondisi dan percabangan dari pilihan keputusan yang tersedia atau konsekuensi dari keputusan tersebut, seperti utilitas, biaya, peluang, dan sebagainya. Pohon keputusan menjadi salah satu cara untuk memperlihatkan algoritma yang mempunyai pernyataan kontrol kondisional. Pohon ini sering digunakan dalam merumuskan langkah riset—terutama pada analisis keputusan—atau membantu mengidentifikasi strategi terbaik mencapai suatu tujuan. Karena kemudahannya, pohon ini juga sering digunakan dalam *machine learning*. Implementasi atau pengembangan yang lebih dalam terhadap pohon keputusan di *machine learning* contohnya seperti *Behavior Tree* yang sering digunakan dalam pembuatan AI musuh pada permainan populer.

Penggunaan pohon keputusan memiliki kelebihan dan kekurangannya tersendiri dibanding alat bantu pemilihan keputusan lain. Kelebihannya:

1. Sederhana dan mudah dimengerti. Dengan penjelasan singkat, orang dapat dengan mudah mengerti alur kerja pohon keputusan yang dibuat.
2. Dapat dikembangkan walau dengan data yang relatif sedikit.
3. Membantu menentukan nilai terbaik, terburuk, dan ekspektasi dari skenario-skenario yang ada.
4. Tidak mengubah data.
5. Bisa dikombinasikan dengan teknik-teknik lain.

Lalu, kekurangan dari pohon keputusan:

1. Tidak stabil.
2. Tidak begitu akurat.

3. Dapat menjadi sangat kompleks seiring dengan banyaknya atribut/faktor yang berperan dalam suatu pengambilan keputusan.
4. Pemilihan keputusan cenderung pada keputusan dengan kedalaman pohon yang lebih dalam.

Pohon keputusan terdiri dari 3 tipe simpul:

1. Simpul Keputusan (*decision nodes*)
   Simpul keputusan menunjukkan kondisi untuk memenuhi pemilihan keputusan pada percabangan tersebut. Keputusan (anak) yang dipilih akan berdasarkan kondisi yang telah terpenuhi.
2. Simpul Kesempatan (*chance nodes*)
   Simpul kesempatan menunjukkan terdapat peluang dalam pemilihan keputusan pada percabangan tersebut. Pemilihan keputusan (anak/cabang) akan dilakukan secara acak berdasarkan peluang tiap cabangnya.
3. Simpul Terminal (*end nodes*)
   Simpul terminal menunjukkan hasil keputusan yang dipilih dan terdapat di daun pohon keputusan (bagian akhir pohon keputusan).

Dalam pohon keputusan pada makalah ini, kita hanya akan menggunakan dua jenis simpul yaitu simpul keputusan dan simpul terminal.

III. STRATEGI MEMENANGKAN *MINESWEEPER* MENGGUNAKAN POLA TERDEFINISI

A. Pola-pola dalam Minesweeper

1. Pola sederhana

   Pola sederhana adalah pola yang cukup sering ditemukan dalam permainan *minesweeper*, dan dapat dengan mudah ditemukan pada permainan level *easy-medium* (tidak hanya pada level *expert*).

   Pada pola ini berlaku hukum reduksi yaitu angka-angka pada pola ini merepresentasikan angka paling tereduksinya. Angka yang tereduksi adalah angka yang telah ditemukan *n* bom di sekelilingnya, sehingga angka di sekitar bom tersebut dikurangkan sebanyak *n*.

   1.1. Angka tepat petak (ATP)

   Pola ini terjadi bila jumlah petak di sekitar 8 arah angkat tersebut sama dengan angkanya. Ini menunjukkan seluruh petak di sekitar angka tersebut merupakan bom.

Figur 3.1. Contoh ATP #1

Figur 3.2. Contoh ATP #2

   1.2. Pola 1-1 atau 1-2

   Jika pola 1-1 ditemukan, maka diagonal luar petak 1 tersebut petak kosong. Tetapi, jika pola 1-2 ditemukan, maka diagonal luar petak 2 adalah bom, dan diagonal luar petak 1 adalah petak kosong.

Figur 3.3. Contoh 1-1

Figur 3.4. Contoh 1-2

Figur 3.5. Mirrored 1-1

   1.3. Pola 1-2-1 atau 1-2-2-1

   Pola ini merupakan gabungan dari 1-1 dan 1-2, dan merupakan salah satu pola yang paling sering ditemukan di dalam *minesweeper*. Solusi untuk pola 1-2-1 adalah bom di depan tiap angka 1, sementara solusi untuk pola 1-2-2-1 adalah bom di depan tiap angka 2. Tidak hanya itu, semua itu terjadi di area tengah maka anda dapat membuka kotak-kotak disekitarnya karena pembukaan bom-bom tersebut telah memenuhkan kebutuhan angka 1 dan 2 sehingga tidak ada lagi bom disekitar bom tersebut.

Figur 3.6. Contoh 1-2-1

Figur 3.7. Contoh 1-2-2-1

   1.4. Pola 1-3-1

   Pada tingkat kesulitan *medium-hard*, pola ini cukup sering ditemukan. Pola ini memiliki solusi berupa bom di diagonal angka 3-pojok, serta petak kosong pada angka 1 yang kedua di tiap sayapnya.

Figur 3.8. Contoh 1-3-1

2. Pola lanjutan

   Pola lanjutan adalah pola yang lumayan susah untuk ditemukan, dan lebih sering ditemukan pada level tinggi.

   2.1. Pola 1-4, 2-5, 3-6

   Pola dengan pasangan kombinasi ini dapat memprediksi 3 petak kosong dan 3 bom, cukup jarang ditemukan kecuali pada tingkat kesulitan tinggi.

Figur 3.9. Contoh 1-4

Figur 3.10. Contoh 2-5

2.2. Pola 1-1, 2-2 atau 3-3 pojok (n-n*p*)

Apabila terdapat pola 1-1, 2-2, atau 3-3 dikelilingi dengan 4 petak terbuka di belakangnya, maka 3 petak di depannya adalah petak kosong. Pola ini juga dinamakan sebagai *triangles* karena petak terbuka di belakang polanya berbentuk L seperti segitiga. Meskipun pola ini cukup jarang ditemukan dan susah untuk melihatnya, pola ini cukup menguntungkan.

Figur 3.11. Contoh 1-1p

2.3. Pola 1-lubang atau 2-lubang (n-H)

Pola ini ditandai dengan 1-1 atau 2-2 yang menembus dinding petak. Apabila angkanya sama, maka solusi dari pola ini adalah 3 petak di dalam lubangnya merupakan petak kosong. Cobalah membuka lubang tersebut dengan memanfaatkan pola sederhana.

Figur 3.12. Contoh 2-H

B. Teori First Click

Pada *Minesweeper* versi *Microsoft Windows*, klik yang pertama akan selalu bukan-bom agar pemain tidak merasa terlalu cepat kalah. Tetapi, bukaan yang didapat bisa jadi lebih besar atau tidak bergantung terhadap posisi klik pertama tersebut. Tim Kostka, salah satu pecinta *minesweeper*, melakukan ratusan kali percobaan untuk menentukan dimana letak bukaan paling besar saat klik pertama di *minesweeper*. Hasilnya adalah sebagai berikut (angka dibawah merupakan banyaknya bukaan saat klik pertama):

	Beginner	Intermediate	Expert
Corner	18	27	16
Edge	20 - 24	31 - 42	19 - 26
Middle	23 - 32	35 - 66	23 - 41

Telah dibuktikan dengan pengamatan bahwa hasil bukaan terbanyak berasal dari klik pertama yang berada di tengah arena permainan, diikuti dengan daerah pinggir dan pojokan. Hal ini terjadi karena di tengah terdapat lebih banyak sampel dibanding daerah pinggir dan daerah pojokan arena permainan, sehingga membuat bukaan pada klik pertama di daerah tengah menjadi besar.

Tetapi, apabila permainan dijalankan pada *minesweeper* yang bukan berasal dari *Microsoft Windows*, maka fitur klik pertama selalu aman itu tidak ada. Terdapat perhitungan dari Emmanuel Brunelliere, seorang *minesweeper enthusiast* asal Prancis, yang menyatakan berapa persentase akan menemukan klik pertama yang berisi bukaan (bukan bom) sebagai berikut:

	Beginner	Intermediate	Expert
Corner	59.54%	59.94%	49.94%
Edge	42.14%	42.61%	31.42%
Middle	25.09%	25.54%	15.69%

Hal ini dibuktikan oleh percobaan dari Paul Kerry dari ratusan permainan dan didapatkan hasil yang mirip untuk *expert mode*:

Place for the first click.	Middle	Edge	Corner
Theorical percentage of openings if we consider that mines are randomly put on the board.	15.69	31.42	49.93
Percentage of openings obtained by Paul Kerry (on a total of 424, 196, and 135 games)	14	31	44

Hasil percobaan yang mirip juga bisa kita lihat pada hasil percobaan Tim Kostka:

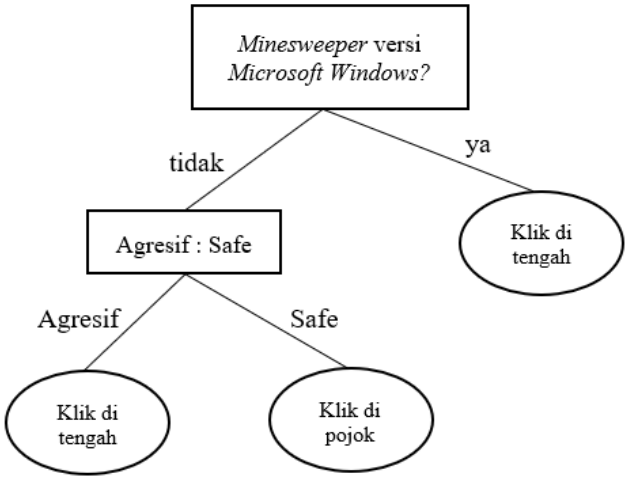
	Beginner	Intermediate	Expert
Corner	50 - 60 %	50 - 60 %	40 - 50 %
Edge	34 - 42 %	36 - 43 %	25 - 32 %
Middle	19 - 24 %	21 - 26 %	12 - 16 %

IV. PEMBAHASAN

A. Pohon Keputusan Teori First Click

Dengan adanya perhitungan teoritis dan hasil percobaan dari komunitas pecinta *minesweeper*, kita dapat menentukan strategi untuk memulai langkah awal pada permainan *minesweeper* dengan melihat versi apa dulu yang kita mainkan, barulah kita bisa memutuskan aksi terbaik yang dapat kita lakukan.

Apabila kita memainkannya dengan versi *Microsoft Windows*, maka yang perlu kita pikirkan hanyalah mencari bukaan terbesar. Namun, akan lebih rumit apabila kita ingin memainkannya di luar versi *Microsoft Windows* yang tidak memiliki fitur klik-pertama-selalu-aman. Pada versi ini, kita harus memilih antara bermain aman atau bermain agresif. Bermain aman berarti kita lebih mengincar untuk menghindari bom pada bukaan pertama, sementara bermain agresif berarti mencari bukaan terbesar tanpa memedulikan risiko terkena bom pada klik pertama. Apabila diterapkan dalam pohon keputusan, maka kita dapat membuat pohon keputusan sebagai berikut.



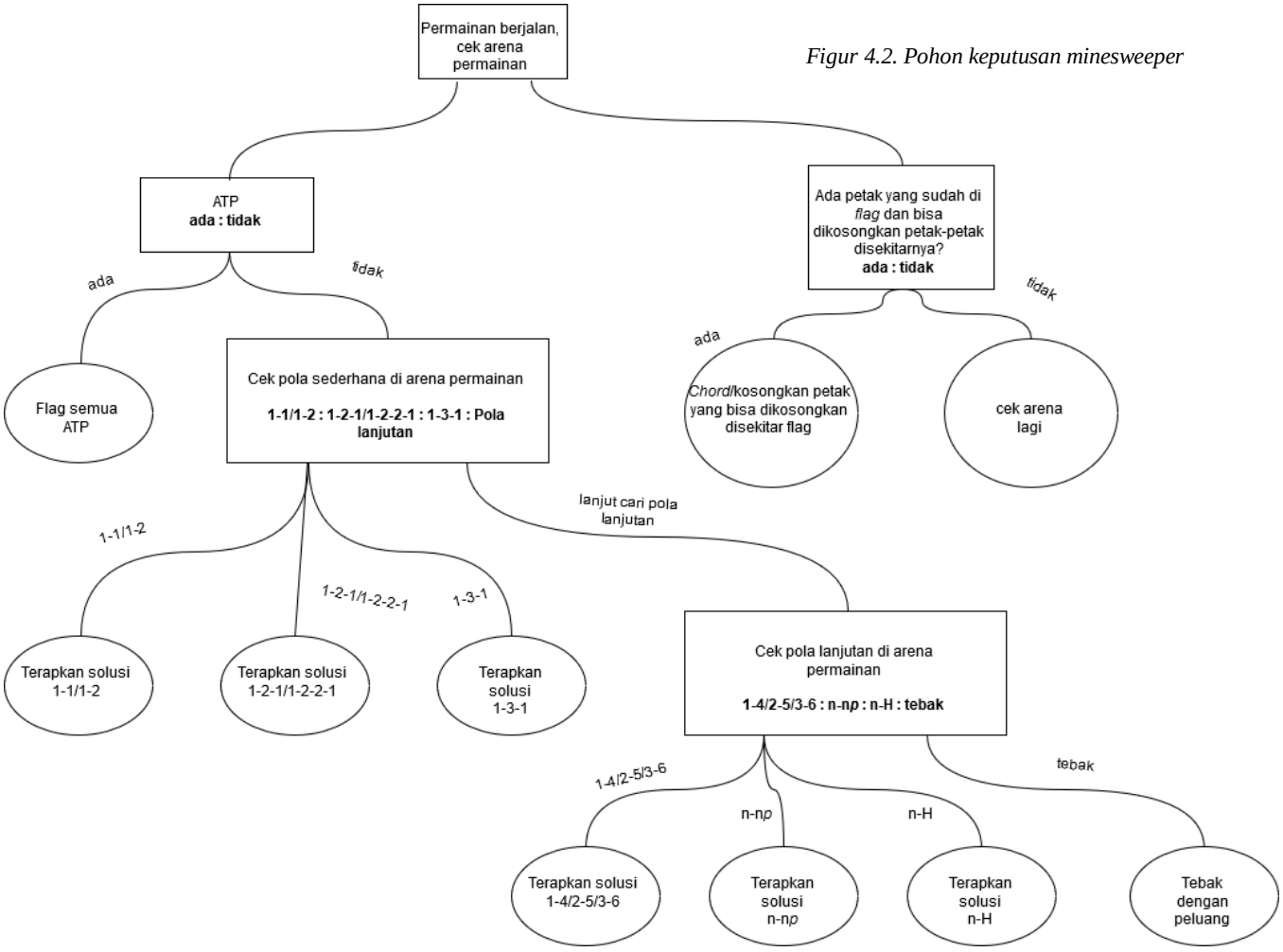
Figur 4.1. Pohon Keputusan klik pertama

Dapat dilihat dari pohon keputusan pada figur di atas bahwa untuk memilih langkah pertama dalam bermain *minesweeper*, pengambilan keputusan untuk mencari langkah terbaik di awal permainan *minesweeper* itu tidaklah susah. Namun, apabila tidak bermain di *Microsoft Windows*, akan lebih baik jika mulai dari pojok karena jumlah bukaan antara pojok dengan tengah tidaklah jauh berbeda, dibanding dengan persentase ketemunya bukaan pada klik pertama. Klik di pinggir tidak terlalu sepadan, karena bukan hanya jumlah bukaannya yang tidak jauh berbeda dengan klik di pojok, perbandingan persentase antara klik di pojok dengan klik di pinggir cukup jauh.

B. Pohon Keputusan Permainan Minesweeper

Dengan memanfaatkan pola-pola yang ditunjukkan dari angka-angka pada papan arena permainan, kita dapat memilih langkah yang lebih tepat dan cepat dalam menyelesaikan permainan ini. Masih banyak lagi pola lain yang merupakan varian-varian dari pola dasarnya ataupun gabungan dari berbagai macam pola yang bisa kita temukan sembari bermain *minesweeper*. Terkadang juga, pola tidak semudah itu untuk mencarinya sehingga membuat pencarian menjadi lama. Pohon keputusan ini hanya bertindak sebagai pembantu dalam membuat keputusan.

Dapat dilihat pada figur pohon keputusan dibawah, untuk memilih langkah dalam bermain *minesweeper* cukup rumit. Kita harus cepat melakukan pengecekan berkala dengan prioritas: Cek ATP (Angka Tepat Petak) dan petak yang sudah di-*flag* dan bisa dikosongkan petak-petak di sekitarnya (*chord*). Apabila sudah selesai mengecek kedua ini, maka lakukan pengecekan arena permainan kembali. Apabila sudah tidak ada lagi ATP atau petak yang bisa di-*chord*, maka kita berlanjut ke langkah



Figur 4.2. Pohon keputusan minesweeper

berikutnya yaitu mengecek adanya pola sederhana atau tidak dalam arena permainan kita. Apabila ada, terapkan solusi yang sesuai, hingga tidak ada lagi pola sederhana dalam permainan kita, baru kembali lagi ke akar utama pohon keputusan yaitu cek arena permainan. Apabila tidak ditemukan ATP, tidak ada yang dapat di-*chord*, ataupun tidak ditemukannya pola sederhana, barulah kita mencari pola lanjutan. Kita tidak langsung mengecek pola lanjutan setelah menerapkan solusi dari pola sederhana karena bisa jadi akan muncul ATP atau jalan baru yang dapat kita hilangkan dengan cara di-*chord*. Setelah kita mencari pola lanjutan dan menerapkan solusinya apabila ketemu, kita kembali lagi ke akar utama pohon keputusan.

Namun, jika ada kalanya *minesweeper* tidak dapat diselesaikan karena tidak adanya ATP, petak yang bisa di-*chord*, pola sederhana maupun pola lanjutan, itu artinya kita harus melakukan rencana alternatif paling terakhir yaitu menebak dengan peluang. Menebak dengan peluang artinya kita menebak kira-kira bomnya ada di petak yang mana, dengan mempertimbangkan sisa bom yang ada di arena permainan, sisa petak yang masih belum dapat diketahui isinya apa, dan susunan angka-angka disekitar petak tersebut. Karena metode menebak dengan peluang tidak dapat dijabarkan dengan pohon keputusan di atas, maka metode ini disimpan untuk topik lain.

V. KONKLUSI

Dengan membuat pohon keputusan dalam permainan *minesweeper* ini kita bisa tahu bahwa dalam menyelesaikan suatu permasalahan kita tidak hanya bergantung terhadap satu keputusan, melainkan bisa jadi kita perlu meninjau ulang, mengambil keputusan berkali-kali hingga tujuan akhir kita tercapai. Sebuah permainan biasanya terdiri dari suatu *game loop*, yang artinya terdapat pengulangan pengecekan dan aksi. Itulah kenapa biasanya di dalam permainan, yang digunakan untuk AI adalah *Behavior Tree*, sistem yang lebih canggih dalam menentukan *game state* dan apa yang harus dilakukan saat itu.

VI. UCAPAN TERIMA KASIH

Pertama, penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa atas berkah dan rahmatnya makalah ini dapat diselesaikan dan dikerjakan dengan sebenar-benarnya. Terima kasih juga kepada dosen pengajar mata kuliah IF2120 Matematika Diskrit K-02, Dra. Harlili, M.Sc. Dan terakhir, penulis ingin berterima kasih kepada teman-teman dan pihak-pihak lainnya yang terkait dalam makalah ini.

DAFTAR PUSTAKA

[1] Desember 2019. [3] Rosen, Kenneth H.. 2012. Discrete Mathematics and Its Application. 7th Ed. New York: McGraw-Hill.
[2] Kamiński, B.; Jakubczyk, M.; Szufel, P. (2017). "A framework for sensitivity analysis of decision trees". Central European Journal of Operations Research. 26 (1): 135–159.
[3] Adamatzky, Andrew (1997). "How cellular automaton plays Minesweeper". Applied Mathematics and Computation. 85 (2–3): 127–137.
[4] On the First Click - http://devtk.com/minesweeper/firstclick.html - Tim Kostka determines the best actual place to start (2006) (visited 5/12/2019)
[5] Authoritative Minesweeper - http://www.minesweeper.info/ - (visited 5/12/2019)
[6] Munir, Rinaldi. 2006. Diktat Kuliah Matematika Diskrit. Institut Teknologi Bandung: Bandung

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, 6 Desember 2019

Faris Rizki Ekananda 13518125