

Analisis Pembukaan Daerah Bebas Ranjau pada *Minesweeper*

Okihita Hasiholan Sihaloho | 13509058

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

okihita@students.itb.ac.id

Abstrak—Minesweeper adalah sebuah permainan *single-player* yang bertujuan membuka seluruh ubin tak beranjau pada sebuah *minefield* berbentuk persegi dengan ukuran tertentu $i \times j$, dengan jumlah ranjau R tertentu. Mengeklik pada sebuah ubin pada *board* akan memberitahu apa yang terdapat di bawah ubin tersebut.

Aplikasi pembukaan daerah bebas ranjau pada Minesweeper dapat diaplikasikan pada jaringan untuk menentukan apa yang aman untuk dibuka tanpa harus mengaktivasi sistem keamanan. Para *hacker* umumnya mengaplikasikan logika pada permainan Minesweeper ini untuk menembus sistem keamanan pada suatu jaringan.

Makalah ini akan membahas cara penentuan ubin mana yang aman untuk dibuka, dengan pendekatan logika.

Index Terms—Minesweeper

I. PENDAHULUAN

Minesweeper adalah sebuah permainan yang menguji logika dan keberuntungan pemain. Walaupun sebagian besar kasus penempatan ranjau dapat diperkirakan dengan metode, namun ada kondisi yang mengharuskan pemain harus menebak ubin mana yang tidak mengandung ranjau.

Minesweeper adalah contoh teorema. Setelah kita mengecek ubin-ubin di sekitar, kita *tahu* ada ranjau di kotak yang kita pilih, dan kita sudah mengalokasikan bendera pada kotak tersebut. Akan tetapi, kita tidak bisa membuka ubin yang kita teoreman sebagai ranjau tersebut. Cara tersebut “tidak mungkin dilakukan” karena kita tidak “anti-ranjau” dan membuka ubin berisi ranjau akan meledakkan kita dan membuat kita kalah. Pada dunia nyata, kondisi ini setara dengan tidak adanya peralatan yang memadai untuk memeriksa kebenaran suatu hal.

AI penempatan ranjau pada permainan Minesweeper menggunakan *constraint satisfaction problems*, CPS, sebuah problem matematika yang mendeskripsikan kumpulan obyek yang harus memenuhi keadaan tertentu. Seperti pada teka-teki silang, obyek tidak harus dibuka secara sekuensial berdasar urutan soal maupun urutan kedekatan posisi.

Microsoft mendeskripsikan Minesweeper sebagai berikut:

1. You can uncover a square by clicking it. If you uncover a mine, you lose the game.
2. If a number appears on a square, it indicates how many total mines are in the eight squares that surround the numbered one. You can use this number to help deduce whether a square is safe to uncover.
3. To mark a square you suspect contains a mine, right-click it. This will add a flag to the square. If you're not sure, right-click it again, which will add a question mark to the square.

Pada program Minesweeper keluaran Microsoft untuk Windows Vista dan setelahnya, jika terjadi “kasus penebakan” (lih. Gambar 2.3), program akan otomatis membenarkan “ubin tersangka” pertama yang diklik oleh pemain, asalkan pilihan itu benar secara logika. Ubin “tersangka” yang lainnya akan otomatis tereliminasi dan di-*assign* sebagai ranjau. Penambahan fitur ini oleh Windows mengeliminasi elemen probabilitas dari permainan Minesweeper, meninggalkan hanya elemen logika saja.

II. DASAR TEORI

Berikut adalah ilustrasi singkat untuk memainkan Minesweeper. Asumsikan tabel berikut adalah *pseudofield* dengan batas atas dan kiri sebagai “tembok”—dengan kata lain, ini adalah potongan dari pojok kiri atas pada *board*.

Untuk penentuan lokasi ranjau, asumsikan bahwa pemasangan bendera selalu benar (selalu menunjukkan di tempat itu terpasang ranjau).

Pada kasus seperti berikut, ubin a dan b pasti bukan ranjau, dan aman untuk dibuka, karena di sekitar ubin 3 sudah ada tiga ranjau yang memenuhi syarat ubin 3.



Gambar 2.1 Daerah Aman pada Minesweeper

Sedangkan, pada kasus di bawah, ubin a dan b sudah pasti ranjau, karena hanya ubin a dan b yang bisa memenuhi kondisi agar ada tiga buah ranjau di sekeliling angka 3.



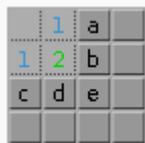
Gambar 1.2 Daerah Ranjau pada Minesweeper

Perhatikan bahwa pada kasus di bawah ini, pemain diwajibkan memilih satu ubin antara a dan b yang bebas dari ranjau untuk dapat dibuka.



Gambar 1.3 Kasus Penebakan

Untuk analisis beberapa angka yang berdekatan, kita bisa menggunakan pendekatan OR. Jika ada ubin-ubin a , b , c , d , dan e , dengan angka di sekeliling mereka, kita bisa membuat sebuah persamaan seperti berikut:



Gambar 1.4 Analisis Angka yang Berdekatan

Pada Gambar 1.4, dapat dilihat bahwa (2, 1) berisi angka 1 dan dikelilingi ubin a dan b . Karena sebuah angka n menunjukkan bahwa ada n buah ranjau di salah satu dari delapan ubin yang mengelilinginya, maka kita tahu bahwa salah satu dari a atau b , namun bukan keduanya, adalah ranjau.

Kita menuliskan pernyataan tersebut sebagai $a + b = 1$, dengan 1 sebagai representasi untuk menunjukkan bahwa ubin tersebut memiliki ranjau.

Bagaimana dengan ubin c , d , dan e ? Kita melakukan hal yang sama. Dengan analisis angka di ubin (2, 2) yang bernilai 2 dan dikelilingi oleh ubin a , b , c , d , dan e , kita menuliskan persamaan peletakan ranjau dengan $a + b + c + d + e = 2$. Begitu juga dengan (1, 2), kita bisa menuliskan $c + d = 1$.

Dari persamaan $a + b = 1$, $c + d = 1$, dan $a + b + c + d + e = 2$, maka dapat dibuat kesimpulan

$$\begin{aligned} a + b + c + d + e &= 2 \\ (a + b) + (c + d) + e &= 2 \\ 1 + 1 + e &= 2 \\ e &= 0 \end{aligned}$$

Persamaan 1.1 Analisis Penempatan Ranjau

yang menunjukkan bahwa ubin e bebas dari ranjau.

(Pada beberapa varian program Minesweeper, pemain bisa memberikan sebuah tanda tanya (?) untuk menandai sebuah ubin yang dicurigai mengandung ranjau.)

III. PENGGUNAAN LOGIKA OR DAN AND UNTUK MEMENANGKAN PERMAINAN MINESWEEPER

Kita akan memainkan sebuah sesi Minesweeper yang dijalankan pada Windows 7. Ada beberapa fitur yang ditambahkan pada versi Minesweeper di Windows 7 ini, salah satunya adalah “tebakkan aman”. Jika terjadi “kasus penebakan” seperti yang terlihat pada Gambar 2.3, “ubin tersangka” manapun yang kita pilih *pertama kali* akan aman selama logika terpenuhi, meninggalkan “ubin tersangka” yang lain *ter-assign* sebagai ranjau.

Berikut adalah langkah-langkah memainkan sebuah game Minesweeper 10 x 10, dengan 10 ranjau, tanpa memperhitungkan waktu.

Kita akan memberi nama ubin dengan koordinat Cartesius, sehingga (1,1) berada di kiri bawah dan (10,10) berada di kanan atas.

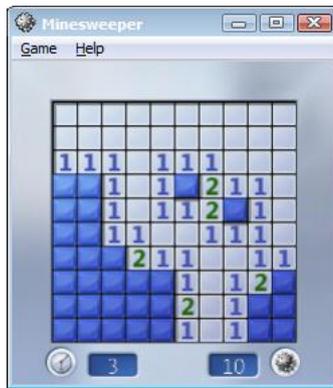
Pertama, set sebuah *board* dengan ukuran 10 x 10 dengan jumlah ranjau 10. Dengan susunan seperti ini, akan ada ${}_{100}C_{10} = 3628800$ buah kemungkinan peletakan ranjau oleh AI.



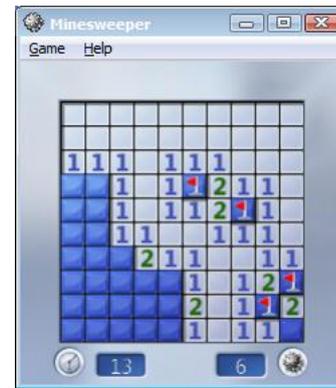
Gambar 3.1 Setup sebuah board

Langkah pertama adalah mengklik sebuah ubin secara acak. Pemilihan pertama akan selalu membawa kita ke sebuah “ubin 0”, yaitu ubin tanpa ranjau pada dirinya sendiri dan tanpa ranjau di sekelilingnya.

Kita akan membuka (4,8) dan melihat sebanyak apa ubin yang terbuka.

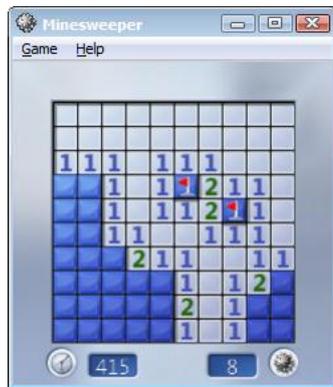


Gambar 3.2 Langkah pertama



Gambar 3.4 Pembukaan ubin c dan d

Kita perhatikan range $(6,8):(9,5)$ khususnya ubin tertutup $(6,7)$ dan $(8,6)$. Dengan melihat bahwa $(5,8)$ berisi angka 1 dan hanya ada satu ubin tertutup di sekelilingnya, sudah pasti $(6,7) = 1$. Dengan begitu, $(6,7)$ adalah ranjau. Logika yang sama juga diterapkan pada angka 1 pada $(9,5)$ yang hanya dikelilingi oleh ubin tertutup $(8,6)$. Beri bendera pada kedua kolom tersebut.



Gambar 3.3 Assignment tebakkan penempatan ranjau

Selanjutnya, kita menganalisis rentang $(8,4):(10,1)$. Kita menamai $(10,3)$ sebagai a ; $(9,2)$ sebagai b ; $(10,2)$ sebagai c ; $(9,1)$ sebagai d ; dan $(10,1)$ sebagai e .

Karena $(9,4)$ dan $(10,4)$ maka

$$(i) \quad a = 1$$

Karena $(9,3)$ maka

$$(ii) \quad a + b + c = 2$$

Karena $(8,3)$ maka

$$(iii) \quad b = 1$$

Karena $(8,1)$ dan $(8,2)$ maka

$$(iv) \quad b + d = 1$$

Persamaan-persamaan di atas dapat disederhanakan:

(i) dan (ii) dan (iii) maka

$$a + b + c = 2$$

$$1 + 1 + c = 2$$

$$2 + c = 2$$

maka c bernilai 0 dan tidak beranjau.

(iii) dan (iv)

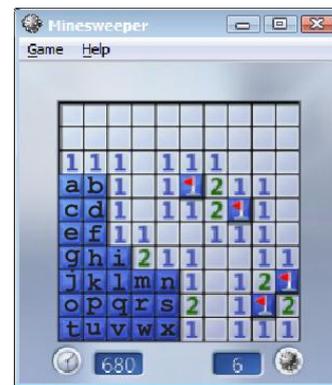
$$b + d = 1$$

maka d bernilai 0 dan tidak beranjau.

Menurut hasil pada persamaan di atas, c dan d aman untuk dibuka. Setelah diaplikasikan pada *board*, pembukaan c dan d menghasilkan ubin angka.

Ubin e juga menjadi otomatis aman, karena ubin d di koordinat $(9,1)$ berisi angka 1 dan sudah terpenuhi oleh munculnya ranjau di $(9,2)$.

Sekarang, kita akan menganalisis ubin tertutup di rentang $(1,7):(5,1)$. Untuk mempermudah penyebutan, masing-masing ubin akan diberi nama sesuai dari kiri ke kanan, kemudian dari atas ke bawah. *Lupakan penamaan untuk rentang $(8,4):(10,1)$ sebelumnya.* Penamaan baru ini akan membuat $(1,7)$ menjadi ubin a , $(3,4)$ menjadi ubin i , dan ubin $(5,1)$ menjadi ubin x .



Gambar 3.5 Penamaan-ubin baru

Berdasar ubin angka yang sudah terbuka, kita bisa menentukan kembali persamaan-persamaan yang bisa diturunkan dari *board* di atas. Karena ada 13 ubin angka yang terbuka, kita dapat menurunkan 13 persamaan.

$$(i) \quad a + b = 1$$

$$(ii) \quad a + b = 1$$

$$(iii) \quad b = 1$$

$$(iv) \quad b + d = 1$$

$$(v) \quad b + d + f = 1$$

$$(vi) \quad f + h + i = 1$$

$$(vii) \quad i = 1$$

$$(viii) \quad i + l + m + n = 2$$

$$(ix) \quad m + n = 1$$

$$(x) \quad n = 1$$

$$(xi) \quad n + s = 1$$

$$(xii) \quad n + s + x = 2$$

$$(xiii) \quad s + x = 1$$

Kita sederhanakan persamaan tersebut:

$$a = 0;$$

$$b = 1;$$

$$d = 0;$$

$$b + d + f = 1 \rightarrow f = 0;$$

$$i = 1;$$

$$f + h + i = 1 \rightarrow 0 + h + 1 = 1 \rightarrow h = 0;$$

$$n = 1;$$

$$m + n = 1 \rightarrow m = 0;$$

$$i + l + m + n = 2 \rightarrow 1 + l + 0 + 1 = 2 \rightarrow l = 0;$$

$$n + s = 1 \rightarrow s = 0;$$

$$s + x = 1 \rightarrow x = 1;$$



Gambar 3.6 Pembukaan ubin berdasar analisis

Berdasar perhitungan sebelumnya yang menyatakan bahwa hanya ubin b, i, n, x yang beranjalau, maka kita membuka a, d, f, h, l, m , dan s . Seperti yang diharapkan, tidak ada ranjalau di ubin-ubin tersebut.

Sekarang kita akan menganalisis pergerakan selanjutnya. Kali ini, kita akan menganalisis ada tidaknya ranjalau di c, e, g, j, k, p, q, r , dan w . Untuk menganalisis hal tersebut, kita akan mengecek seluruh ubin-ubin angka yang berada di sekitar ubin yang akan kita cek tersebut.

Kita punya **tujuh** ubin angka yang bisa digunakan untuk menganalisis, yakni (1,7), (2,6), (2,5), (2,4), (3,3), (4,3), dan (5,2). Kita mendapat **tujuh** persamaan, dengan tetap mempertimbangkan ubin beranjalau dalam sistem analisis kita.

- (i) $b + c = 1$
- (ii) $b + c + e = 1$
- (iii) $c + e + g + i = 1$
- (iv) $g + j + k + i = 2$
- (v) $i + k + p + q + r = 2 \rightarrow k + p + q + r = 1$
- (vi) $i + q + r + n = 2 \rightarrow q + r = 0$
- (vii) $n + r + w + x = 3 \rightarrow r + w = 1$

Kita sederhanakan persamaan tersebut: dengan hasil sebelumnya yang menyatakan bahwa

$$b = 1;$$

$$b + c = 1 \rightarrow 1 + c = 0;$$

$$b + c + e = 1 \rightarrow 1 + 0 + e = 1 \rightarrow e = 0;$$

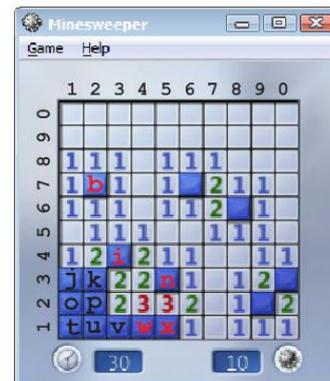
$$c + e + g + i = 1 \rightarrow 0 + 0 + g + 1 = 1 \rightarrow g = 0;$$

$$q + r = 0 \rightarrow q = r = 0;$$

$$r + w = 1 \rightarrow w = 1;$$

$$k + p + q + r = 1 \rightarrow k + p + 0 + 0 = 1 \rightarrow k + p = 1$$

karena $k + p$ masih belum dapat disederhanakan, maka kita akan membuka kotak yang sudah pasti aman, yaitu c, e, g, q , dan r . Tidak lupa, menandai w dengan tanda merah sebagai tanda bahwa dia positif ranjalau.



Gambar 3.7 Pembukaan ubin berdasar analisis

Sekarang kita akan menganalisis pergerakan selanjutnya. Kali ini, kita akan menganalisis ada tidaknya ranjalau di j, k, o, p, t, u dan v . Untuk menganalisis hal tersebut, kita akan mengecek seluruh ubin-ubin angka yang berada di sekitar ubin yang akan kita cek tersebut.

Kita punya **lima** ubin angka yang bisa digunakan untuk menganalisis, yakni (1,4), (2,4), (3,3), (3,2), dan (4,2). Sekarang kita mendapat **lima** persamaan, dengan tetap mempertimbangkan ubin beranjalau dalam sistem analisis kita.

- (i) $j + k = 1$
- (ii) $j + k + 1 = 2 \rightarrow j + k = 1$
- (iii) $i + k + p = 2 \rightarrow k + p = 1$
- (iv) $k + p + u + v + w = 2 \rightarrow k + p + u + v = 1$
- (v) $v = 0$

Kita sederhanakan persamaan tersebut menjadi:

- (i) $v = 0;$
- (ii) $k + p + u = 1$
- (iii) $k + p = 1 \rightarrow u = 0;$
- (iv) $j + k = 1$
- (v) $(j + k) \times (k + p) = 1 \rightarrow k = 1;$
- (vi) $j = p = 0;$

dengan begitu, kita menemukan bahwa ranjalau terakhir berada di bagian k .

Karena seluruh ubin angka telah dianalisis dengan tepat dan memenuhi *constraint satisfaction problems*, maka hakikat Minesweeper sebagai kumpulan ubyek berupa ubin yang memenuhi kondisi saling terikat kini sudah terpenuhi.

Kesepuluh ranjau sekarang sudah ditemukan, dengan koordinat masing-masing adalah (2,7), (6,7), (8,6), (3,4), (2,3), (5,3), (10,3), (9,2), (4,1), (5,1).

Permainan Minesweeper dimenangkan. Semua ubin kini dapat ditentukan isinya. Kita tidak perlu menebak lagi apakah sebuah ubin berisi ranjau, angka, atau kosong. Dengan logika AND dan OR yang disajikan pada makalah ini, kita bahkan bisa membuat program sederhana yang dapat menjadi AI pencari ranjau dalam sebuah *field* Minesweeper berukuran $i \times j$, dengan jumlah ranjau R .

III. KESIMPULAN

Minesweeper adalah permainan logika yang dapat dipecahkan menggunakan logika OR, AND, dan (dengan kuantitas penggunaan yang kecil) hukum-hukum logika yang lain.

Dengan substitusi persamaan, *assignment* pada sebuah ubin apakah dia berisi ranjau, angka, atau bebas, kita dapat secara deduksi menentukan apakah di ubin lain terdapat ranjau.

REFERENSI

- <http://windows.microsoft.com/en-US/windows-vista/Minesweeper-how-to-play> (17 Dec 2010)
- http://en.wikipedia.org/wiki/Minesweeper_%28Windows%29 (14 Dec 2010)
- <http://id.answers.yahoo.com/question/index?qid=20101125111421AAz7oIR> (15 Dec 2010)
- <http://www.cs.duke.edu/courses/cps102/spring05/notes/minesweeper.pdf> (14 Dec 2010)
- <http://www.informatika.org/~rinaldi/Matdis/2008-2009/Logika.ppt> (15 Dec 2010)

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, 29 April 2010
ttd

Okihita Hasiholan Sihalofo | 13509058