Perbandingan Penerapan Algoritma *Brute Force* dengan Algortima BFS dan DFS dalam Penyelesaian Permainan Minesweeper

Josep Andre Ginting - 13517108
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13517108@std.stei.itb.ac.id

***Abstrak*—Permainan Minesweeper merupakan salah satu permainan yang paling sering dimainkan di dunia digital dari dulu seklai hingga saat ini. Permainannya sendiri sudah lama ada pada komputer-komputer dengan sistem operasi windows dari tahun 1990. Algoritma *Brute Force,* BFS dan DFS dapat digunakan untuk menyelesaikan permainan ini. Oleh karena itu, pada makalah ini akan dibahas pemanfaatan penggunaan algoritma-algoritma tersebut dalam permainan minesweeper.**

Kata Kunci—Minesweeper, algoritma, Brute Force, Breadth First Search, Depth First Search;

I. PENDAHULUAN

Minesweeper adalah permainan *single player* paling sukses yang pernah dibuat. Bahkan melebihi permainan seperti *the sims,* tretis dan *world of warcracf.* Minesweeper adalah sebuah permainan populer yang membutuhkan logika dan keterampilan dalam memecahkan masalah ditingkat pemula, refleks yang cepat dan pengenalan pola ditingkat menengah lanjutan. Minesweeper diluncurkan pada tahun 1990 sebagai bagian dari *windows entertaiment pack* sebelum di promosikan untuk menjadi fitur standar pada windows 3.1 dan seterusnya. Minesweeper adalah permainan asli microsoft, dibuat oleh Curt Johnson dan Robert Donner.

Minesweeper bertujuan untuk membuka semua ubin yang kosong dan menghindar dari ubin yang berisi ranjau pada *minefield* yang berbentuk persegi berukuran m x n (ukuran ditentukan tingkat kesulitan). Berikut adalaha ukuran *minefield* berdasarkan tingkat kesulitan :

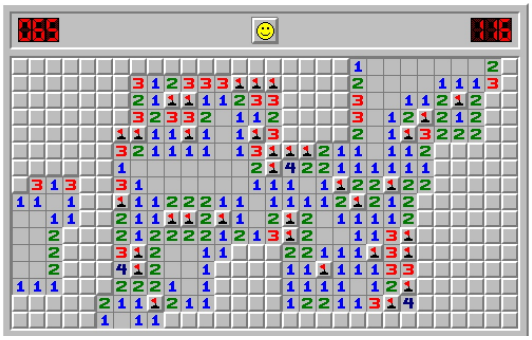
a. *Beginner* : 10 ranjau, ukuran ubin 9 x 9

b. Intermediate : 40 ranjau, ukuran ubin 16 x 16

c. Advance : 99 ranjau, ukuran ubin 16 x 30

d. Custom : menentukan ranjau (*range* : 10-668), tinggi ubin (*range* : 9-24), menentukan lebar ubin (*range*: 9-30).

Permainan ini diawali dengan sebuah *minefield* kosong. Permainan harus membuka ubin yang dapat berisi ubin kosong, angka ataupun ranjau. Jika mengklik ranjau, maka permianan berakhir. Tujuannya adalah untuk membuka semua kotak kosong secepat mungkin untuk mendapatkan nilai yang tingga.



Gambar 1.1 Permainan Minesweeper

Sumber (http://minesweeperonline.com/)

Seiring berkembangnya dunia teknologi dan munculnya algoritm-algoritma pemrograman yang baru, maka dapat dibuatkan sebuah program yang mampu secara otomatis menyelesaikan permainan minesweeper tersebut. Tentunya dengan menggunakan program, penyelesaiann permainan akan menajdi lebih cepat dan juga menjadi lebih pasti karenan akan terhindar dari faktor kesalahan yang sering dilakukan oleh manusia.

Algoritma yang dapat digunakan untuk menyelesaikan permaianan minesweeept ini adalah B*rute Force* dan juga algoritma BFS (*Breadth First Search*) dan DFS(*Depth First Search*). Dari Algoritma tersebut, tentunya memiliki kelebihan dan kekurangan masing-masing. Mulai dari cara berpikir maupun dari ke efisienan nya dalam menyelesaikan permainan minesweeper ini.

Makalah ini akan membahas bagaimana penyelesainan permainan dengan menggunakan algoritma B*rute Force, Breadth First Search (BFS)* dan *Depth First Search (DFS),* beserta dengan perbedaan yang dihasilkan saat memilih algoritma dalam permainan minesweaper.

II. LANDASAN TEORI

A. Algoritma *Brute Force*

Brute Force adalah sebuah pendekatan yang langsung (*straightforward*) untuk memecahkan suatu masalah , biasanya didasarkan pada pernyataan masalah (*problem statement*) dan definisi konsep yang dilibatkan. Algoritma *Brute Force* memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang sangat jelas (*obvious way*).

Karakteristik Algoritma *Brute Force* :

1. Algoritma brute force umumnya tidak "cerdas" dan tidak mangkus, karena ia membutuhkan jumlah langkah yang besar dalam penyelesaiannya. Kadang-kadang algoritma brute force disebut juga algoritma naif (*naïve algorithm*).
2. Algoritma *brute force* seringkali merupakan pilihan yang kurang disukai karena ketidakmangkusannya itu, tetapi dengan mencari pola-pola yang mendasar, keteraturan, atau trik-trik khusus, biasanya akan membantu kita menemukan algoritma yang lebih cerdas dan lebih mangkus.
3. Untuk masalah yang ukurannya kecil, kesederhanaan brute force biasanya lebih diperhitungkan daripada ketidakmangkusannya. Algoritma brute force sering digunakan sebagai basis bila membandingkan beberapa alternatif algoritma yang mangkus.
4. Algoritma *brute force* seringkali lebih mudah diimplementasikan daripada algoritma yang lebih canggih, dan karena kesederhanaannya, kadang-kadang algoritma *brute force* dapat lebih mangkus (ditinjau dari segi implementasi).

B. *Breadth First Search* (BFS)

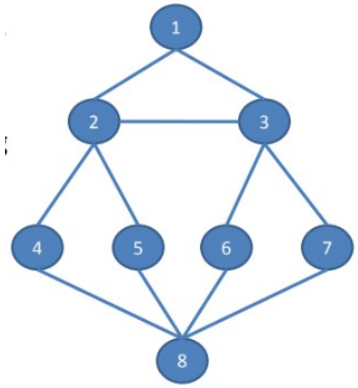
Breadth first search (BFS) adalah algoritma yang melakukan pencarian secara melebar yang mengunjungi simpul secara preorder yaitu mengunjungi suatu simpul kemudian mengunjungi semua simpul yang bertetangga dengan simpul tersebut terlebih dahulu. Selanjutnya, simpul yang belum dikunjungi dan bertetangga dengan simpul-simpul yang tadi dikunjungi , demikian seterusnya. Jika graf berbentuk pohon berakar, maka semua simpul pada arah d dikunjungi lebih dahulu sebelum simpul-simpul pada arah d+1.

Algoritma ini memerlukan sebuah antrian q untuk menyimpan simpul yang telah dikunjungi. Simpul-simpul ini diperlukan sebagai acuan untuk mengunjungi simpul-simpul yang bertetanggaan dengannya. Tiap simpul yang telah dikunjungu masuk ke dalam antrian hanya satu kali. Algoritma ini juga membutuhkan table Boolean untuk menyimpan simpul yang telah dikunjungi sehingga tidak ada simpul yang dikunjungi lebih dari satu kali.

BFS sering juga disebut sebagai pencarian melebar, karena proses pencariannya yang menyebar dan melebar. Dimana traversal dimulai dari simpul v.

Algoritma umum :

1. Kunjungi simpul v.
2. Kunjungi semua simpul yang bertetangga dengan simpul v terlebih dahulu.
3. Kunjungi simpul yang belum pernah dikunjungi dan bertetangga dengan simpul-simpul yang tadi dikunjungi, sedemikian seterusnya.



Gambar 2.1 Graf tidak ber arah

Sumber (http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/stima18-19.htm)

Penelusuran BFS dari graf pada Gambar 2.1 jika ditelusuri mulai dari simpul 1 adalah : 1, 2, 3, 4, 5, 6, 7, 8.

Struktur data BFS adalah sebagai berikut.

1. Matriks ketetanggaan A = [aij] yang berukuran nxn, aij=1, jika simpul i dan simpul j bertetangga, aij= 0, jika simpul I dan simpul j tidak bertetangga.
2. Antrian q untuk menyimpan simpul yang telah dikunjungi.
3. Tabel Boolean, diberi nama "dikunjungi" dikunjungi : array[l..n] of boolean dikunjungi[i] = true jika simpul i sudah dikunjungi dikunjungi[i] = false jika simpul i belum dikunjungi

Properti dai BFS :

- Completeness?

– Ya (selama nilai b terbatas)

- Optimality?

– Ya, jika langkah = biaya

- Kompleksitas waktu:

– 1+b+b^2+b^3+...+b^d = O(b^d)

- Kompleksitas ruang:

– O(b^d)

- Kurang baik dalam kompleksitas ruang

C. Depth First Search (DFS)

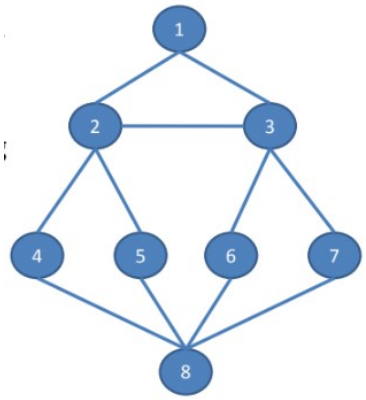
DFS (Depth First Search) adalah salah satu algoritma penelusuran struktur graf atau pohon berdasarkan kedalaman. Simpul ditelusuri dari root kemudian ke salah satu simpul anaknya (misal nya prioritas penelusuran berdasarkan anak pertama [simpul sebelah kiri]), maka penelusuran dilakukan terus melalui simpul anak pertama dari simpul anak pertama level sebelumnya hingga mencapai level terdalam.

Setelah sampai di level terdalam, penelusuran akan kembali ke 1 level sebelumnya untuk menelusuri simpul anak kedua pada pohon biner [simpul sebelah kanan] lalu kembali ke langkah sebelumnya dengan menelusuri simpul anak pertama lagi sampai level terdalam dan seterusnya.

BFS sering juga disebut dengan pencarian mendalam, karena proses pencariannya yang menelusiri sebuah graf sampai pada titik terdalam terlebih dahulu. Traversal juga dimulai dari simpul v.

Algoritma Umum :

1. Kunjungi simpul v.
2. Kunjungi simpul w yang bertetangga dengan simpul w.
3. Ulangi DFS mulai dari simpul w.
4. Ketika mencapai simpul u sedemikian sehingga semua simpul yang bertetangga dengannya telah dikunjungi, pencarian dirunut-balik (*bactrack*) ke simpul terakhir yang dikunjungi sebelumnya dan mempunyai simpul w yang belum dikunjungi.
5. Pencarian berakhir bila tidak ada lagi simpul yang belun dikunjungi yang dapat dicapai dari simpul yang di kunjungi.



Gambar 2.2 Graf tidak ber arah

Sumber (http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/stima18-19.htm)

Penelusuran DFS dari graf pada Gambar 2.1 jika ditelusuri mulai dari simpul 1 adalah : 1, 2, 4, 8, 5, 6, 3, 7.

Properti dari DFS :

- Completeness?

– Ya (selama nilai b terbatas, dan ada penanganan '*redundant paths*' dan '*repeated states'*)

- Optimality?

– Tidak

- Kompleksitas waktu:

– O(b^m)

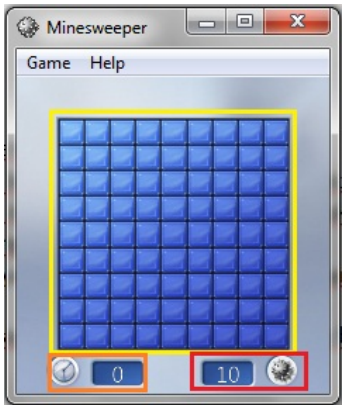
- Kompleksitas ruang:

– O(bm)

- Kurang baik dalam kompleksitas waktu, lebih baik dalam kompleksitas ruang

D. MINESWEEPER

Tujuan dari permainan minesweeper adalah untuk mengetahui posisi semua ranjau/jebakan secepat mungkin tanpa membukanya dan membuak semua kotak/ubin yang bukan merupakan ranjau/jebakan.

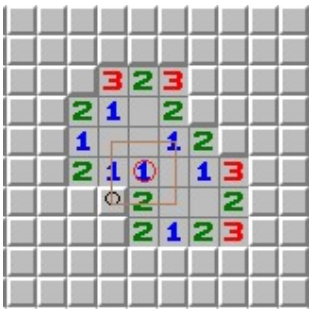


Gambar 2.3 Kondisi awal permainan minesweeper

Sumber (http://rock-a-way.blogspot.com/2013/01/cara-tips-dan-strategi-bermain.html)

Aturan Bermain :

- Waktu mulai berjalan setelah salah satu kotak diklik.
- Kotak terbuka setelah diklik, jika muncul bom pada kotak maka game berakhir.
- Jika muncul angka pada kotak yang terbuka, itu menunjukan jumlah bom yang ada di delapan kotak yang mengelilingi angka tersebut.
- Untuk menandai kotak yang mengandung bom, klik kanan maka muncul gambar bendera.
- Di area game terdapat lapangan permainan, penghitung bom, dan timer.



Gambar 2.4 Permainan Minesweeper
Sumber (Dokumen Pribadi)

Pada gambar diatas terlihat pada angka yang ditandai dengan lingkaran berwarna merah yaitu angka 1, menunjukkan bahwa disekitar kotak/ubin tempat angka itu berada terdapat ranjau/jebakan sebannyak angka pada kotak/ubin pada kasus ini adalah 1 jebakan. Jadi dengan demikian kita dapat dengan pasti menyatakan bahwa ranjau/jebakan yang ada disekitar ubin yang ditandai dengan lingkaran berwarna merah tersebut ada pada ubin yang ditandai dengan lingkaran hitam. Hal itu sudah dapat dipastikan karena selain kotak/ubin yang ditandai dengan lingkaran hitam semuanya telah terbuka dan tidak ada yang merupakan ranjau/jebakan, sehingga sudah pasti satu kotak/ubin sisanya adalah ranjau/jebakan.

Dengan pengetahuan itu, kita sudah dapat menyelesaikan permanan permainan minesweeper dengan benar tanpa membuka kotak/ubin yang berisi jebakan/ranjau.

III. Penerapan Algoritma *Brute Force* dalam Permainan Minesweeper

Algoritma *Brute Force pada permainan Minesweeper* akan diterapkan dalam pemilihan kotak/ubin pada *minefield* yang diprediksi tidak berisi ranjau/jebakan.

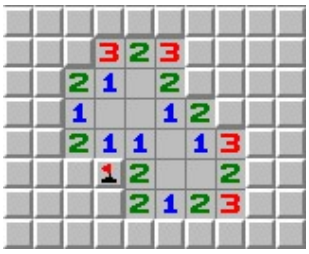
A. Langkah Strategi

Berikut langkah-langkah penyelesaian Minesweeper menggunanan Algoritma *Brute Force*

1. Buka salah satu kotak/ubin pada papan permainan.
2. Setelah muncul angka-angka, dengan pengetahuan teknik bermain minesweeper yang sudah dijelaskan pada bagian minesweeper sebelumnya, buka kembali kotak/ubin yang pasti bukan merupakan ranjau/jebakan.
3. Buka lagi kotak/ubin berikutnya yang sudah dipastikan bukan merupakan ranjau/jebakan.
4. Ulangi perintah 3 dan 4 hinnga semua kotak/ubin yang bukan merupakan ranjau/jebakan sudah terbuka.

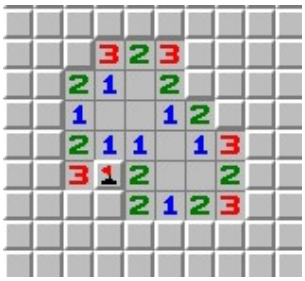
B. Implementasi

Beriukut contoh pengimplementasian strategi *Brute Force* dalam permaiann Minesweeper.



Gambar 3.1 Permainan Minesweeper
Sumber (Dokumen Pribadi)

Dari Gambar 3.1 diatas, dapat dipastikan bahwa kotak/ubin yang ditandai oleh bendera/*flag* adalah sebuah ranjau/jebakan. Sehingga untuk mengingat bahwa itu ranjau/jebakan ditandai dengan bendera. Dan secara otomatis kotak/ubin disebelah kiri dari bendera sudah pasti bukan ranjau/jebakan. Kesimpulan dapat diambil dari kotak/ubin diatas tanda bendera yang berisi angka 1, dimana menunjukkan disekitarnya hanya ada 1 ranjau/jebakan yaitu yang ditandai dengan bendera, otomatis ubin/kotak disebelah kiri bendera bukanlah ranjau, sehinnga dapat dibuka.



Gambar 3.2 Permainan Minesweeper
Sumber (Dokumen Pribadi)

Setelah ubin dikiri bendera dibuka, tentu akan menunjukkan angka mengenai data jumlah ranjau/jebakan disekitarnya kembali.

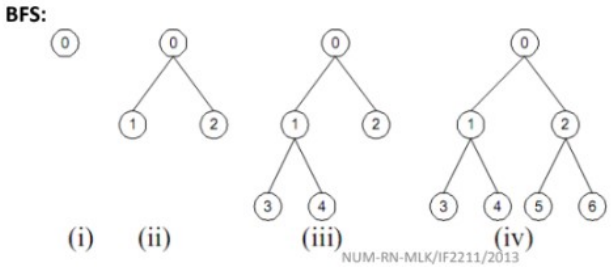
Pada Algoritma *Brute Force* pemilikan ubin/kotak akan dilakukan seperti contoh diatas hingga menyelesaikan permainan, dengan teknik ini maka dapat dipastikan permainan minesweepr dapat dimenangkan. Pengecekan setiap kali ingin memilih kotak/ubin yang ingin dicek/ dibuka bisa sebanyak ukuran ubin yaitu mxn, sehingga akan sangat banyak memakan waktu dalam eksekusi.

IV. Penerapan Algoritma BFS dan DFS dalam Permainan Minesweeper

Tujuan dari algoritma DFS dan BFS pada permainan minesweeper ini adalah untuk mengunjungi semua kotak/ubin kosong yang pasti bukan merupakan ranjau/jebakan dan membukanya.

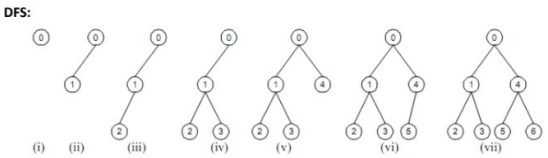
A. Langkah Strategi

Langkah-langkah BFS :
1. Buka salah satu kotak/ubin pada papan permianan
2. Kotak/ubin pertama dapat dianggap sebagai root pada pohon ruang status penelusuran.
3. Setelah muncul angka, buka semua kotak/ubin yang merupakan tetangga dari kotak/ubin yang dibuka sebelumnya yang sudah dapat dipastikan bukan merupakan ranjau/jebakan.
4. Lakukan proses 3 pada semua tetangga dari ubin/kotak yang sudah dibuka sebelumnya hinnga semua kotak/ubin sudah terbuka.



Gambar 3.3 Cara membangun pohon ruang status pada penelusuran BFS
Sumber (http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/stima18-19.html)

Langkah-langkah DFS :
1. Buka salah satu kotak/ubin pada papan permianan
2. Kotak/ubin pertama dapat dianggap sebagai root pada pohon ruang status penelusuran.
3. Setelah muncul angka, buka kotak/ubin yang merupakan tetangga pertama dari kotak/ubin yang dibuka sebelumnya yang sudah dapat dipastikan bukan merupakan ranjau/jebakan.
4. Ulangi proses 3 hingga tidak ada lagi tetangga dari simpul sebelumnya pada ruang status yang bisa dibuka.
5. Lakukan *backtrack* pada simpul sebelumnya dan lakukan penelusuran DFS kembali, hingga semua ubin/kotak dibuka.



Gambar 3.4 Cara membangun pohon ruang status pada penelusuran DFS
Sumber (http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/stima18-19.html)

B. Perbedaan BFS dan DFS dalam penyelesaian Minesweeper

Penyelesaian permainan minesweeper dengan menggunakan algoritma *Breadth First Search (*BFS*)* ataupun *Depth First Search (*DFS*)* tidak jauh berbeda. Perbedaannya hanyalah proses penelusurannya dimana pada BFS menggunakan antrian (*queue*) sedangkan BFS menggunakan struktur tumpukan *(stack)* dalam penyimpanan simpul pada pohon ruang status yang akan di eksekusi berikutnya. Sehingga pada pohon ruang status proses penelusuran pada BFS akan melebar sedangkan pada pohon ruang status DFS akan menuju kedalam. Pada Kompleksitas waktu, Algoritma DFS maupun BFS memiliki kompleksitas yang sama.

V. Kesimpulan

Dalam menyelesaikan permainan minesweeper dibutuhkan logika dalam mengenali pola agar tidak terkena ranjau/jebakan. Karena sistempermainan yang memiliki pola tertentu yang jelas, maka dalam penyelesaiannya dapat dilakukan dengan menerapkan algoritma-algoritma yang ada. Algoritma-algoritma yang dapat digunakan untuk menyelesaikan permainan minesweeper adalah Algoritma *Brute Force, Breadth First Search (*BFS*)* dan *Depth First Search (*DFS*)*.

Dalam masalah waktu, Algoritma *Breadth First Search (*BFS*)* dan *Depth First Search (*DFS*)* memiliki waktu eksekusi yang sedikit lebih baik bila dibandingkan dengan Algoritma *Brute Force.* Karena pada Algoritma *Brute Force* pengecekan dilakukan satu-persatu kesemua kotak/ubin, sedangkan pada BFS dan DFS pengecekan dilakuakan hanya pada tetangga dari kotak/ubin yang masih hidup pada pohon ruang status.

Namun ada kekurangan dari algoritma yang telah diberitahukan sebelumnya, yaitu pada permainan minesweeper tidak 100% mengandalkan logika, namun bisa saja terdapat unsur keberuntungan, misalnya pada saat memilih kotak/ubin pertama saat ingin bermain, sehingga penggunaan algoritma diatas tidak bisa menjadin 100% permainan dapat dimenangkan.

VI. Ucapan Terima Kasih

Puji Syukur penulis ucapkan kepada Tuhan Yang Maha Esa karena atas berkat dan rahmad-Nya makalah ini dapat terselesaikan dengan baik. Ucapan terima kasih juga penulis ucapkan kepada Bapak Dr. Ir. Rinaldi Munir, M.T., Ibu Dr. Masayu Leylia Khodra, ST., MT., dan Ibu Dr. Nur Ulfa Maulidevi, ST., MT. Selaku dosen pembimbing mata kuliah IF2211 Strategi Algoritma tahun 2018/2019 yang telah memberikan ilmu kepada penulis yang merupakan dasar dari penulisan makalah ini. Penulis juga mengucapkan terima kasih kepada Orang Tua penulis yang selalu mengirimkan doa dan memberi semangat dan dukungan kepada penulis. Terima kasih.

REFERENSI

[1] http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/stima18-19.htm diakses pada 25 april 2019

[2] ana.staff.gunadarma.ac.id/Downloads/files/28937/Algoritma+Brute+Force.ppt diakses pada tanggal 25 april 2019

[3] http://fx-blogparts.com/sejarah-minesweeper.html diakses pada tanggal 26 april 2019

[4] http://task-campus.blogspot.com/2012/04/analisa-permainan-minesweeper.html diakses pada tanggal 26 april 2019

[5] http://rock-a-way.blogspot.com/2013/01/cara-tips-dan-strategi-bermain.html diakses pada tanggal 26 april 2019

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, 25 April 2019

Josep Andre Ginting - 13517108