

Penggunaan Algoritma Topological Sort dalam Penentuan Urutan Langkah pada Mahjong Solitaire

Hansel Grady Daniel Thamrin (13518140)

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13518140@std.stei.itb.ac.id

Abstract—Dalam permainan *mahjong solitaire*, terdapat berbagai algoritma yang dapat dijalankan untuk memenangkan permainan. Salah satu unsur utama dalam penentuan kemenangan tersebut adalah urutan langkah yang benar. Pada permainan *mahjong solitaire*, sebagian besar keping tidak bisa diakses secara langsung oleh pemain. Untuk mengakses keping yang tidak bisa diakses tersebut, pemain perlu menghilangkan keping-keping yang menghalangi keping yang tidak bisa diakses tersebut. Hal ini berlaku untuk setiap keping yang menghalangi keping tersebut, sehingga setiap keping yang posisinya semakin tengah akan bergantung dengan keping lainnya yang posisinya lebih dekat dengan ujung. Hal ini menyebabkan pengurutan langkah dapat diselesaikan dengan algoritma *topological sort*.

Keywords—*mahjong solitaire*, *topological sort*, *urutan langkah*

I. PENDAHULUAN

Mahjong Solitaire adalah salah satu permainan yang biasa dimainkan pada platform elektronik. *Mahjong Solitaire* merupakan salah satu adaptasi dari permainan mahjong tradisional, namun permainan *mahjong solitaire* memiliki cara bermain yang berbeda jika dibandingkan dengan permainan mahjong tradisional. Permainan ini muncul sebagai salah satu dampak dari globalisasi, sehingga permainan mahjong tradisional diadaptasi menjadi permainan ini.

Salah satu perbedaan yang tampak jelas antara mahjong tradisional dengan *mahjong solitaire* adalah aturan mainnya. Pada mahjong tradisional, permainan dilakukan dengan 4 orang dan terdapat penghitungan skor di dalam permainan. Pada *mahjong solitaire*, permainan dilakukan dengan memasangkan suatu keping dengan tipe yang sama hingga tidak ada lagi keping yang ada di atas papan. Keping yang bisa dipilih merupakan kepingan yang berada di paling pinggir. Kepingan yang dihalangi oleh

keping lainnya tidak dapat dipilih sebelum semua keping yang menghalangi keping tersebut sudah disingkirkan.

Pada dasarnya, kita dapat menginterpretasikan keping yang menghalangi suatu keping lainnya sebagai penghilangan keping yang menghalangi keping lainnya merupakan prasyarat untuk menghilangkan keping yang terhalangi. Dengan interpretasi ini, kita dapat menggunakan algoritma pengurutan berdasarkan pohon dinamis berarah, yaitu *topological sort*. Dengan algoritma ini, kita dapat menentukan urutan dari langkah yang harus kita ambil untuk menghilangkan suatu keping yang posisinya acak.

II. DASAR TEORI

A. Mahjong Solitaire

Mahjong solitaire merupakan salah satu permainan yang dijalankan pada platform elektronik. Permainan ini didasari oleh permainan mahjong yang berasal dari dinasti Qing. Meskipun permainan ini didasari oleh permainan tersebut, pada kenyataannya permainan *mahjong solitaire* memiliki alur permainan yang berbeda jika dibandingkan dengan permainan mahjong yang sebenarnya.



Gambar 2.1 Tangkapan layar permainan *mahjong solitaire*

<https://www.scoop.it/topic/free-absorbing-online-games-for-couples-to-play>

Jika dibandingkan dengan permainan mahjong tradisional, permainan *mahjong solitaire* memiliki gaya permainan yang lebih dekat dengan permainan solitaire, namun permainan ini menggunakan kepingan mahjong ketimbang menggunakan kartu permainan atau *playing cards*. Selain itu, permainan *mahjong solitaire* dapat diselesaikan dengan menghilangkan semua keping yang terdapat di atas papan. Sementara itu, pada permainan mahjong tradisional, tujuan utama permainan adalah mencetak skor sebesar-besarnya dengan membangun seri *meld* sebanyak 4 jika permainan dilakukan dengan 13 ubin atau 5 jika permainan dilakukan dengan 16 ubin. Pada permainan mahjong tradisional, terdapat beberapa fase dalam suatu permainan, seperti fase pengambilan keping. Sementara itu, pada permainan *mahjong solitaire*, permainan dilakukan dengan mencocokkan setiap keping yang sama pada papan. Keping yang dapat dipilih atau dicocokkan adalah keping-keping yang letaknya di paling ujung dan tidak dihalangi oleh keping-keping lain yang menyebabkan keping lain tersebut menjadi keping yang paling ujung dan menggantikan keping tersebut. Selain itu, pada permainan *mahjong solitaire*, jika tidak ada lagi pasangan, pemain dapat melakukan pengacakan pada susunan keping yang ia miliki dengan harapan bahwa ada pasangan keping yang dapat dipasangkan setelah dilakukan pengacakan. Selain itu, pada *mahjong solitaire* bentuk susunan keping pada papan dapat bervariasi.



Gambar 2.2 Permainan mahjong tradisional

<http://eleonorabietolini.blogspot.com/2016/11/dasar-dasar-belajar-mahjong-tujuan.html>

B. Topological Sort

Topological Sort adalah salah satu algoritma yang digunakan untuk melakukan pengurutan. Algoritma ini dilakukan pada suatu graf asiklik berarah. Jika suatu graf bersifat siklik, maka graf tersebut tidak dapat digunakan untuk pengurutan dengan algoritma *topological sort*.

Anany Levitin, menyebutkan dalam bukunya yang berjudul *Introduction to the Design and Analysis of Algorithm* edisi ketiga menyebutkan bahwa salah satu prinsip utama yang digunakan dalam algoritma *topological sort* adalah algoritma *depth-first search* (DFS) dan

algoritma *breadth-first search* (BFS). Algoritma *depth-first search* atau DFS merupakan salah satu algoritma pencarian yang dilakukan pada graf. Algoritma pencarian ini menekankan pada pencarian menyeluruh pada graf berarah dengan mengutamakan pencarian yang mendalam pada graf berarah. Hal ini menyebabkan jika suatu pencarian dilakukan pada suatu graf dan mesin pencarian dihadapkan pada pilihan antara pencarian pada cabang anak dan berpindah ke cabang lain, maka mesin pencarian akan melanjutkan pencarian ke cabang anak terlebih dahulu sebelum melanjutkan pencarian ke cabang lain. Jika suatu titik tidak memiliki cabang akan, maka mesin pencarian akan melanjutkan pencarian ke cabang lain yang “bersaudara” dengan cabang tersebut. Sementara itu, *breadth-first search* (BFS) adalah algoritma pencarian yang didasarkan pada graf berarah, seperti algoritma *depth-first search* (DFS). Perbedaan antara algoritma *depth-first search* (DFS) dan algoritma *breadth-first search* (BFS) adalah prioritas cabang pencarian yang dipilih. Jika pada algoritma *depth-first search* (DFS), mesin pencarian akan menentukan cabang anak sebagai cabang dengan prioritas yang lebih tinggi ketimbang cabang lain, maka algoritma *breadth-first search* (BFS) akan menentukan cabang lain sebagai cabang dengan prioritas yang lebih tinggi ketimbang cabang anaknya.

Pada algoritma *topological sort*, mesin akan diberikan sebuah graf berarah. Tipe data graf ini dapat dibentuk menjadi bentuk data *linked list* yang menghubungkan alamat dari setiap elemen pada graf ataupun sebuah matriks ketetanggaan yang merepresentasikan hubungan ketetanggaan antara satu titik dengan titik lainnya. Algoritma *topological sort* dapat dilakukan dengan langkah sebagai berikut.

1. Hitung derajat dari setiap simpul yang ada pada graf.
2. Pilih simpul dengan nilai *out-degree* 0. Jika ada lebih dari 1 simpul yang memenuhi ketentuan tersebut, pilih salah satu sesuai dengan preferensi yang ditetapkan.
3. Hapus simpul yang sebelumnya dipilih
4. Ulangi langkah 1 hingga graf sudah tidak memiliki simpul.

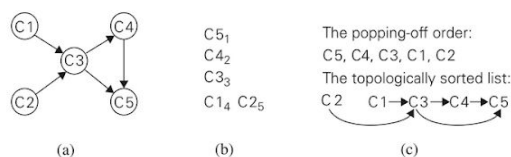


FIGURE 4.7 (a) Digraph for which the topological sorting problem needs to be solved. (b) DFS traversal stack with the subscript numbers indicating the popping-off order. (c) Solution to the problem.

Gambar 2.3 Ilustrasi algoritma *topological sort*

Levitin, Anany, 2012, *Introduction to the Design and Analysis of Algorithms, 3rd Ed.*, Upper Saddle River: Pearson.

Pada algoritma *topological sort* di atas, algoritma akan melakukan penghapusan simpul dari simpul terdalam. Algoritma *topological sort* juga dapat melakukan penghapusan simpul dari simpul terluar atau simpul yang paling dekat dengan simpul akar. Algoritma ini dinamakan dengan *source-removal algorithm*. Pada dasarnya, algoritma *source-removal* melakukan hal yang sama dengan algoritma *topological sort* yang telah dibahas sebelumnya, namun algoritma ini berfokus pada penghilangan simpul yang memiliki *in-degree* 0, alih-alih menghilangkan simpul terdalam dari suatu graf berarah.

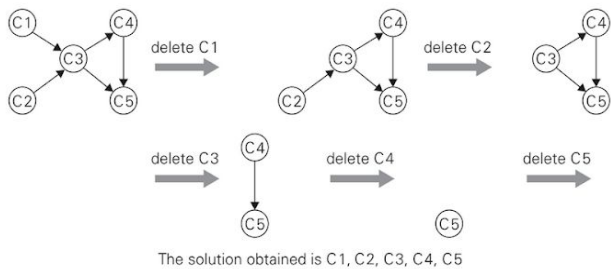


FIGURE 4.8 Illustration of the source-removal algorithm for the topological sorting problem. On each iteration, a vertex with no incoming edges is deleted from the digraph.

Gambar 2.4 Ilustrasi algoritma *topological sort* dengan algoritma *source-removal*

Levitin, Anany, 2012, *Introduction to the Design and Analysis of Algorithms, 3rd Ed.*, Upper Saddle River: Pearson.

III. HASIL DAN PEMBAHASAN

I. Pengusulan Algoritma

Berikut ini adalah algoritma yang diusulkan oleh penulis sehubungan dengan permasalahan yang sebelumnya telah dijabarkan.

1. Dari papan mahjong beserta susunan keping yang sudah didapat, pilih keping yang akan disingkirkan. Suatu keping dapat disingkirkan apabila terdapat keping yang sama pada papan. Apabila keping tersebut adalah satu-satunya keping yang ada di papan saat itu, maka keping tersebut tidak dapat dihilangkan untuk sementara waktu.
2. Pilih jalur yang akan kita tempuh untuk menghilangkan keping tersebut. Jalur adalah arah ekspansi yang akan kita pilih. Sebagai contoh suatu keping yang diapit oleh 2 atau lebih keping yang posisinya dalam satu garis dengan keping tersebut akan memiliki 2 jalur, yaitu jalur kanan dan jalur kiri. Dalam permasalahan ini, pilihlah jalur yang paling pendek.
3. Tambahkan keping yang kita pilih ke dalam graf berarah. Jika keping yang ditambahkan bukan keping akar yang dipilih, tambahkan busur dari keping yang memulai pencarian dengan tujuannya adalah keping yang baru saja dimasukkan ke dalam graf.




4. Identifikasi keping yang posisinya paling dekat dengan keping yang dipilih sesuai dengan jalur yang kita pilih. Apabila keping yang dipilih sudah berada di ujung, lanjutkan pencarian ke langkah berikutnya.
5. Lakukan pencarian pada seluruh papan. Cari keping bertipe sama namun berbeda posisi dengan keping yang dipilih. Jika dalam pencarian tidak ditemukan keping lain yang bertipe sama selain keping yang dipilih, maka pencarian tidak dapat dilanjutkan dan dinyatakan gagal.
6. Kembali ke langkah 2 dengan keping yang dipilih sebagai referensi. Jika semua keping sudah berpasangan, lanjut ke langkah 7
7. Dari graf yang telah dibuat, pilih simpul dengan *out degree* 0 dan masukkan ke dalam senarai dan hapus simpul beserta busur yang menunjuk simpul tersebut. Lakukan hal ini hingga semua simpul pada graf telah dimasukkan senarai.

II. Hasil Pengujian Algoritma

Dalam kasus uji ini, penulis akan menggunakan simbol untuk menggantikan penulisan keping mahjong. Berikut ini adalah simbol yang akan penulis gunakan untuk menggambarkan keping mahjong pada kasus ini.

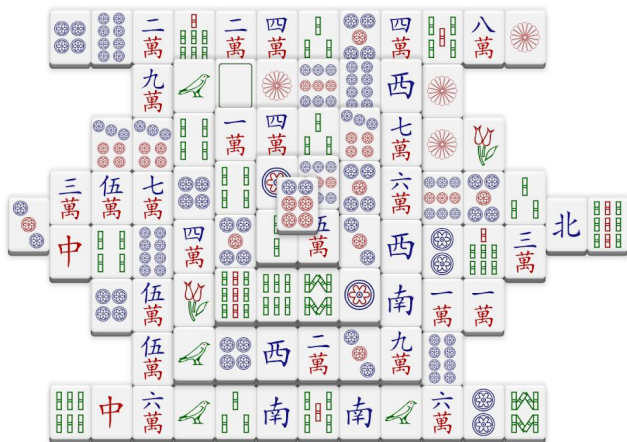
TABEL I. KONVENSI PENULISAN REFERENSI KEPINGAN

No.	Gambar kepingan	Simbol penulisan
1.		H3
2.		B3
3.		G1
4.		A6
5.		H4
6.		B6

7.		C7
8.		C3
9.		B9

Sumber : Dokumentasi penulis

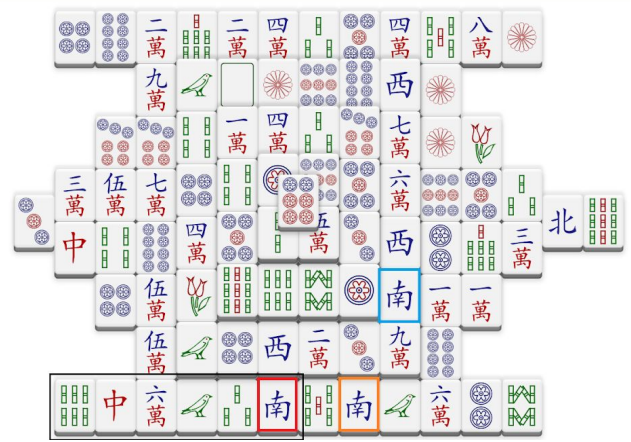
Konvensi tabel ini bukan suatu pilihan yang baku, sehingga konvensi ini dapat diganti suatu saat ketika penelitian lain dilakukan. Berikut ini adalah konfigurasi susunan *mahjong solitaire* untuk uji kasus kali ini.



Gambar 3.1 Gambar salah satu susunan permainan *mahjong solitaire*

<https://mahjong.gg/>

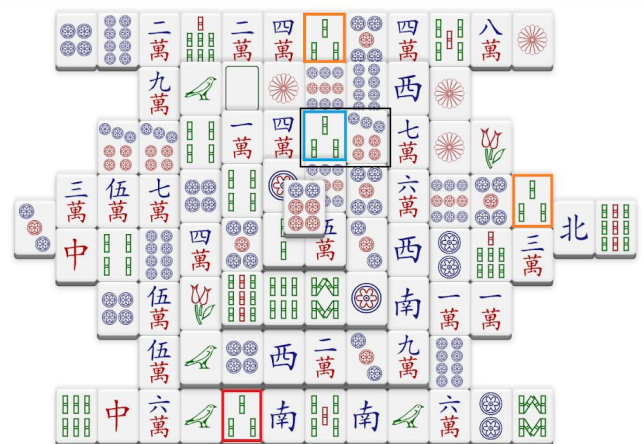
Dalam contoh kasus ini, kita akan mengambil keping yang berada pada lokasi paling bawah posisi kelima dari kiri, yaitu keping H3 sesuai dengan konvensi yang penulis tetapkan. Jika kita tinjau, kita dapat menggunakan 2 jalur untuk pencarian urutan langkah, yaitu jalur sebelah kanan dan jalur sebelah kiri. Pada kasus ini, kita akan menggunakan jalur sebelah kiri karena jalur kiri hanya dihalangi 5 keping sementara jalur kanan dihalangi 6 keping.



Gambar 3.2 Gambar kasus uji dengan keping berkotak merah sebagai keping terpilih dan kotak hitam menunjukkan jalur yang dipilih

Dokumentasi penulis

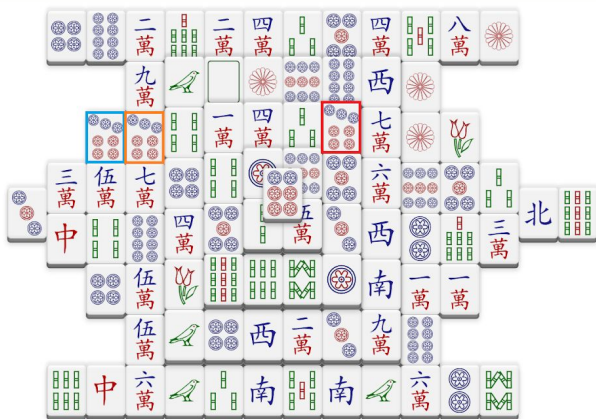
Di jalur sebelah kiri, keping yang dipilih dihalangi oleh keping B3, G1, A6, H4, dan B6 secara berurutan. Untuk iterasi pertama, tinjau keping B3 yang bersebelahan dengan keping yang kita pilih. Pada papan permainan *mahjong solitaire*, kita mendapati bahwa terdapat 3 keping B3 lain yang dapat kita pasangkan dengan keping B3 yang sudah kita tandai sebelumnya. Keping-keping B3 ini adalah keping B3 yang berada pada baris 1, baris 3, dan baris 4. Keping B3 pada baris 1 memiliki 2 jalur, yaitu jalur kanan dan jalur kiri. Pada jalur kanan, terdapat 5 keping yang menghalangi, sementara itu, pada jalur kiri, terdapat 6 keping yang menghalangi. Pada baris 3, terdapat 2 jalur pula, yaitu jalur kanan dan jalur kiri. Jalur kanan dihalangi 2 keping, sementara jalur kiri dihalangi oleh 1 keping. Pada baris 4, terdapat 2 jalur, yaitu jalur kanan dan jalur kiri. Jalur kiri dihalangi 2 keping sementara jalur kanan secara teknis tidak akan kita pilih karena keping tersebut terlalu jauh dari sisi paling kanan. Pada kasus ini, kita akan menggunakan keping B3 yang berada di baris 3 melalui jalur kanan karena jalur tersebut hanya dihalangi oleh 1 keping, yaitu keping C7.



Gambar 3.3 Gambar kasus uji untuk pencarian pasangan keping B3

Dokumentasi penulis

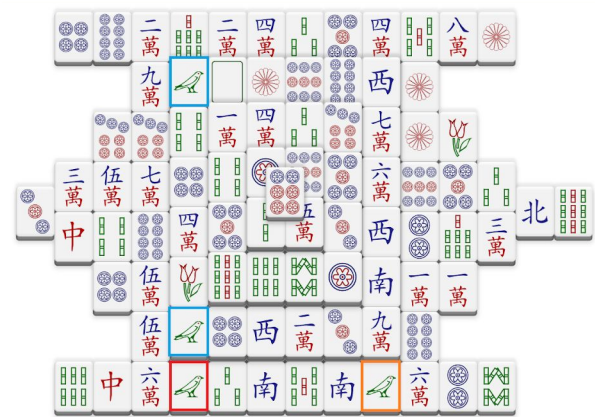
Setelah itu, kita akan melanjutkan peninjauan terhadap keping C7 yang sebelumnya telah kita pilih. Pada papan, terdapat 2 keping C7 yang bisa kita pilih, yaitu keping pada baris 3 kolom 3 dan kolom 4. Keping pada kolom 3 adalah kepingan paling kiri pada baris 3, sehingga keping tersebut dapat kita pilih, sehingga kita akan menggunakan keping C7 pada posisi tersebut. Karena keping C7 tidak terhalangi, maka dua keping C7 dapat langsung kita pilih. Hal ini membuat keping B3 pada baris 3 kolom 3 dan kolom 7 dapat disingkirkan, sehingga kita dapat membebaskan B3 yang terhalangi. Maka dari itu, pencarian akan kita lanjutkan ke keping G1.



Gambar 3.4 Gambar kasus uji untuk pencarian pasangan keping C7

Dokumentasi penulis

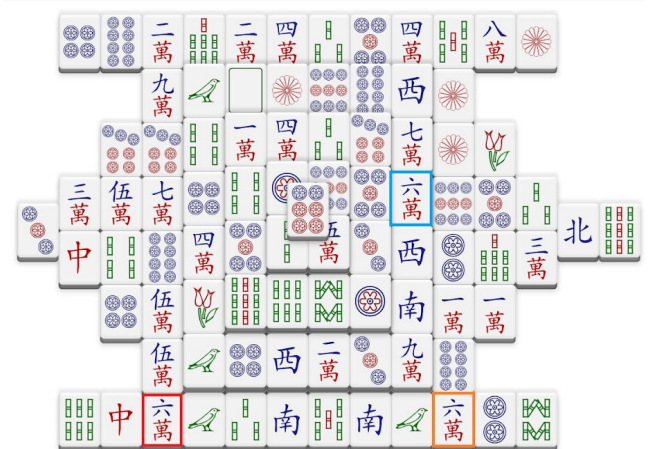
Pada keping G1, terdapat 3 keping G1 lain yang dapat dipasangkan dengan keping G1 yang kita pilih, yaitu keping G1 pada baris 2, baris 7, dan baris 8 kolom 10. Keping pada baris 2 dan baris 7 berada pada posisi paling kiri, sehingga kita bisa menggunakan salah satu dari kedua keping tersebut untuk dipasangkan dengan keping G1 yang kita pilih. Hal ini menyebabkan pencarian untuk keping G1 tidak menghasilkan pencarian untuk keping lainnya, sehingga pencarian akan kita lanjutkan ke keping A6.



Gambar 3.5 Gambar kasus uji untuk pencarian pasangan keping G1

Dokumentasi penulis

Pada keping A6, terdapat 2 kandidat keping A6 lain yang dapat kita gunakan untuk dipasangkan dengan keping A6 yang kita pilih, yaitu keping A6 pada baris 4 dan baris 8. Pada keping A6 baris 4, posisinya berada di paling kanan, sehingga kita dapat langsung memilihnya. Sementara itu, keping A6 yang berada di baris 8 terhalangi oleh 2 keping jika kita memilih jalur kiri atau 9 keping jika kita memilih jalur kanan. Hal ini menyebabkan keping A6 pada baris 4 dapat dipilih untuk dipasangkan dengan keping A6 yang telah kita tandai sebelumnya. Karena keping A6 dapat langsung dipilih, maka pencarian dapat kita lanjutkan ke keping H4.

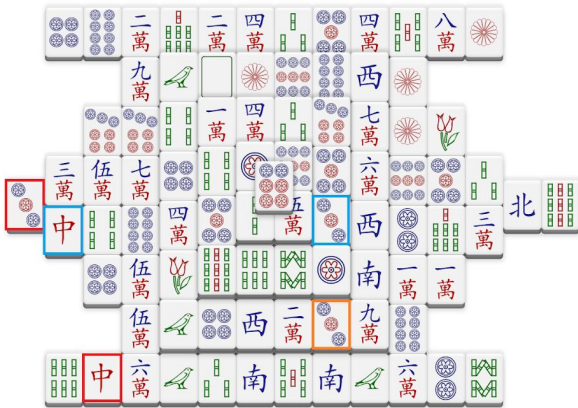


Gambar 3.6 Gambar kasus uji untuk pencarian pasangan keping A6

Dokumentasi penulis

Pada keping H4, terdapat 1 keping H4 lain yang dapat dipasangkan dengan keping H4 yang kita pilih, yaitu keping H4 pada baris 5. Keping H4 ini terhalang oleh 1 keping jika kita mengambil jalur kiri, yaitu keping C3. Hal ini menyebabkan pencarian kita lanjutkan ke keping C3. Pada keping C3, terdapat 2 keping C3 lain yang dapat dipasangkan dengan keping C3 yang kita pilih, yaitu keping

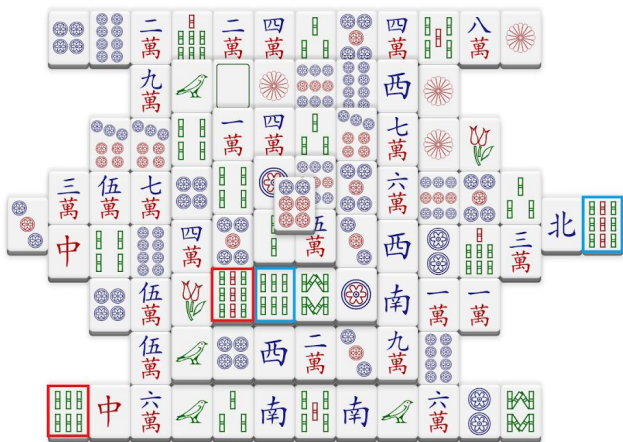
C3 pada baris 5 dan baris 7. Pada baris 5, keping C3 tidak terhalang jika diakses dari kiri, sementara keping C3 pada baris 7 terhalang 1 keping jika diakses dari kanan dan 4 keping jika diakses dari kiri. Dari sini, kita akan memilih keping C3 yang berada di baris 5 untuk dipasangkan dengan keping C3 yang kita tandai. Setelah itu, kita akan melanjutkan pencarian ke keping B6.



Gambar 3.7 Gambar kasus uji untuk pencarian pasangan keping C3 dan H4

Dokumentasi penulis

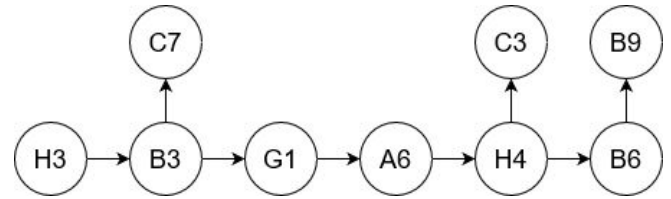
Pada keping B6, hanya terdapat 1 kandidat keping A6 yang dapat menjadi pasangan keping A6 yang kita tandai. Keping tersebut terhalang oleh 1 keping jika diakses dari jalur kiri dan 2 keping jika diakses dari jalur kanan. Dalam hal ini, kita akan memilih jalur kiri karena jumlah keping penghalangnya lebih sedikit. Keping tersebut terhalang oleh keping B9. Terdapat 1 kandidat keping B9 yang dapat dipasangkan dengan keping B9 yang telah ditandai, yaitu keping B9 pada kolom paling kanan. Karena keping B9 tersebut sudah berada di posisi paling kanan, maka secara otomatis keping B9 tersebut dapat diakses secara langsung dan dapat dipasangkan secara langsung pula.



Gambar 3.8 Gambar kasus uji untuk pencarian pasangan keping B6 dan B9

Dokumentasi penulis

Dengan pemasangan B9 ini, maka pencarian dihentikan karena tidak ada simpul yang dapat diekspan. Dari sini, kita dapat menggambarkan graf yang telah kita buat dalam bentuk sebagai berikut.



Gambar 3.9 Graf dari pengurutan langkah yang telah dilakukan

Dokumentasi penulis

Dari graf yang telah kita buat sebelumnya, kita dapat menggunakan pengurutan dengan algoritma yang telah kita jabarkan sebelumnya sehingga kita mendapati urutan langkah untuk menyingkirkan keping H3 yang telah kita pilih di awal adalah B9, C3, C7, B6, H4, A6, G1, B3, H3.

III. Evaluasi Algoritma

Jika kita melihat algoritma yang telah diusulkan penulis sebelumnya, kita akan mendapati bahwa algoritma ini memiliki beberapa kelemahan dalam hal pelaksanaannya. Salah satu kelemahan yang dimiliki dari algoritma ini adalah algoritma ini hanya melihat keping mahjong yang terlihat, sehingga keping lain yang tidak terlihat tidak akan dimasukkan ke dalam penghitungan. Dalam hal ini, kita tidak bisa mengetahui keping yang berada di balik suatu keping, terutama keping-keping mahjong yang tertutup secara sempurna. Hal ini menyebabkan pengurutan yang kita lakukan mungkin tidak menuntun kita pada solusi terbaik.

Selain itu, algoritma ini bukan suatu algoritma yang dapat berdiri sendiri untuk mencapai tujuan utama permainan, yaitu menyelesaikan permainan *mahjong solitaire*. Hal ini menyebabkan algoritma ini perlu didukung dengan algoritma lain agar algoritma ini dapat lebih efektif. Selain itu, algoritma ini juga dapat ditambahkan dengan unsur pendukung lain, seperti nilai heuristik untuk meningkatkan akurasi pemilihan rute serta keping yang nantinya akan dipasangkan dengan keping yang kita pilih.

IV. KESIMPULAN

Dari penjabaran yang telah penulis lakukan, dapat disimpulkan bahwa algoritma *topological sort* dapat digunakan dalam penentuan urutan langkah pada permainan *mahjong solitaire*. Namun, diperlukan beberapa modifikasi serta penelitian lebih lanjut agar algoritma yang diusulkan dapat lebih optimal dari segi pemakaian maupun akurasi.

LINK VIDEO YOUTUBE

Untuk mempermudah penjelasan, berikut ini adalah link YouTube mengenai algoritma yang telah penulis jabarkan sebelumnya.

<https://youtu.be/ZvNg3E1AsHE>

V. UCAPAN TERIMA KASIH

Penulis mengucapkan rasa syukur yang sebesar-besarnya kepada Tuhan Yang Maha Esa. Karena rahmat-Nya yang tiada hentinya tercurahkan kepada penulis, penulis dapat menyelesaikan makalah ini. Selain itu, penulis juga mengucapkan terima kasih kepada dosen mata kuliah IF2211 Strategi Algoritma K-02, Dr. Nur Ulfa Maulidevi ST, M.Sc. yang telah membimbing penulis selama menjalani perkuliahan mata kuliah terkait. Karena bimbingan beliau, penulis dapat menyelesaikan makalah ini dengan baik.

Penulis juga ingin mengucapkan terima kasih bagi setiap pihak yang secara langsung maupun tidak langsung telah terlibat dalam membantu penulis dalam menyelesaikan makalah ini. Tanpa bantuan pihak-pihak tersebut, penulis tidak akan dapat menyelesaikan makalah ini dengan baik.

REFERENCES

Berikut ini adalah sumber referensi yang digunakan dalam penulisan makalah ini.

- [1] Admin. (6 Februari 2019). “*How to play Mahjong Solitaire?*”.
<https://solitaire.onl/how-to-play-mahjong-solitaire/>. diakses 1 Mei 2020.
- [2] Levitin, Anany. 2012. *Introduction to the Design and Analysis of Algorithms. 3rd ed.*. Upper Saddle River: Pearson.
- [3] Tim GeeksForGeeks. “Topological Sorting”.
<https://www.geeksforgeeks.org/topological-sorting/>. diakses 1 Mei 2020.
- [4] Tim mahjongonline.eu. 2006. “*Winning Mahjong*”.
<http://www.mahjongonline.eu/rules-winning-mahjong.htm>. diakses 1 Mei 2020.

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 2020



Hansel Grady Daniel Thamrin
13518140