Fungsi Heuristik Untuk Pelacakan Pada Aplikasi Kecerdasan Buatan Dalam Kasus Puzzle

Y.Al-Qadry[1], I.Perdana[2], M.Kania S[1]
[1] Jurusan Teknik Informatika, FT, Jl. Dipati Ukur Bandung
[2] Jurusan Teknik Informatika, FT ST-INTEN, Jl. Ir.H. Djuanda

Abstrak

Dalam permainan teka-teki (*puzzle*) sering kita dihadapkan kepada kesulitan bagaimana cara kita untuk menyelesaikan permainan puzzle dengan gerakan atau pemindahan sekecil mungkin. Biasanya orang-orang dalam menyelesaikan permainan puzzle adalah dengan cara mencoba-coba, terus dicoba sampai tujuan yang diinginkan ditemukan tanpa mempedulikan waktu yang digunakan untuk menyelesaikan permainan puzzle tersebut, yang penting permainan puzzle tersebut selesai. Dengan cara mencoba-coba ini kurang efisien karena waktu yang digunakan dalam menyelesaikan permainan puzzle tersebut cukup lama dan langkah-langkah yang digunakan tidak sistematis yang memungkinkan akan terjadi pengulangan pada langkah yang sama.
Kata kunci : puzzle, efisien, sistematis.

I. Pendahuluan

Kecerdasan buatan merupakan sub-bidang ilmu komputer yang khusus ditujukan untuk membuat software dan hardware yang sepenuhnya bisa menirukan beberapa fungsi otak manusia sehingga komputer dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia, serta diharapkan komputer mampu membantu manusia di dalam memecahkan berbagai masalah yang lebih rumit. Tujuan utama kecerdasan buatan adalah mengubah komputer menjadi alat yang lebih pintar dan lebih berguna.

Salah satu persoalan yang ditangani oleh kecerdasan buatan adalah permainan (*games*). Dalam permainan teka-teki (*puzzle*) sering kita dihadapkan kepada kesulitan bagaimana cara kita untuk menyelesaikan permainan puzzle dengan gerakan atau pemindahan sekecil mungkin.

Bagaimana software kecerdasan buatan bisa menyelesaikan permainan puzzle dalam memperoleh keadaan tujuan secepat mungkin dengan gerakan atau pemindahan sekecil mungkin sehingga waktu aksesnya cepat dan tidak terlalu banyak membutuhkan memori?

Untuk itu dibutuhkan suatu fungsi yang dapat memberikan informasi tambahan (heuristik) dari domain yang bersangkutan dalam proses pelacakan sehingga dapat mengurangi jumlah alternatif yang harus dipilih dan mengarah pada satu solusi yang tepat atau tujuan yang diinginkan.

II. Metodologi

a. Pengumpulan data

Membaca (*literature*) dan mempelajari berbagai buku yang berhubungan dengan topik di atas dan menjadi bahan pada penulisan tugas akhir, serta sebagai referensi bagi penulis.

b. Analisis Masalah

Memecahkan beberapa masalah dengan menggunakan ketiga metoda pelacakan, serta menganalisa dan membandingkan hasil penyelesaian dari masing-masing metoda pelacakan.

c. Perancangan

Merancang program yang akan dibuat dengan mendisain *user interface* dari tiap perancangan layar dialog.

d. Coding dan pengujian

Mengimplementasikan susunan kotak-kotak, penerapan algoritma pelacakan dan pemakaian struktur data dalam simulasi, serta melakukan pemilihan perangkat keras dan perangkat lunak. Pengetesan atau pengujian program dilakukan untuk memeriksa fungsi dan prosedur dalam simulasi berjalan dengan lancar.

III. Hasil dan Pembahasan

Studi kasus yang diambil untuk proses pelacakan adalah persoalan kotak 8 (eight puzzle). Masalah yang dibentuk yaitu sebagai berikut : ditunjukkan atau diberikan terlebih dahulu keadaan awal (*initial state*) dan keadaan tujuan (*goal state*), keadaan awal adalah kotak-kotak yang sudah diacak dari keadaan tujuan dengan cara menggeser-geser kotak kosong beberapa kali, sedangkan keadaan tujuan adalah susunan kotak-kotak angka yang terurut. Tentukan jalur solusi dari keadaan awal menuju keadaan tujuan pada kedalaman pelacakan yang telah ditentukan dengan cara menggeser-geser kotak yang kosong menggunakan algoritma pelacakan *depth first search*, *breadth first search* dan *best first search.*

Dalam menentukan jalur solusi yang dihasilkan akan berbeda dari setiap algoritma pelacakan karena diproses

menurut aturan atau langkah dari masing-masing algoritma pelacakan tersebut. Hasil pelacakan yaitu berupa jalur solusi dengan jumlah langkah, jumlah simpul yang diperiksa, ditemukan pada tingkat kedalaman (*level*) keberapa,dan waktu pelacakan yang dibutuhkan dalam menentukan jalur solusi dari permainan sebuah puzzle.

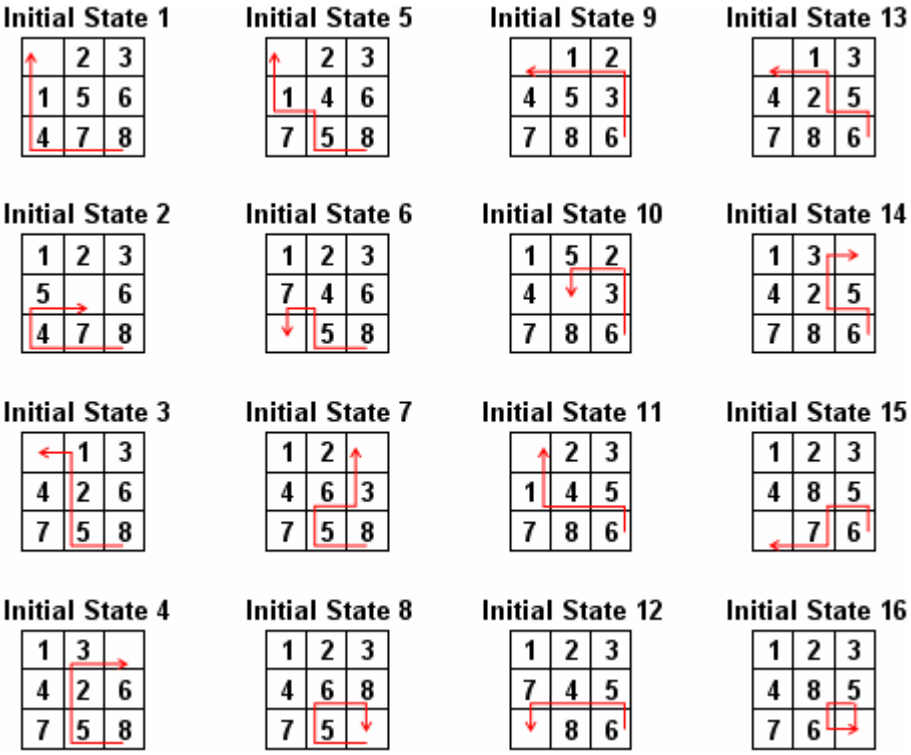
Di bawah ini merupakan susunan kotak-kotak angka dalam keadaan terurut dari persoalan kotak 8.

Goal State

1	2	3
4	5	6
7	8	

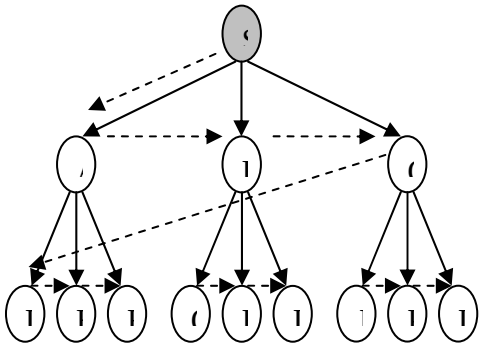
Apabila kotak kosong dari gambar di atas digeser sebanyak 4 kali (dengan catatan kotak kosong tidak boleh digeser ke tempat di mana kotak kosong tersebut digeser sebelumnya), maka akan diperoleh 16 keadaan baru.

Keenam belas keadaan baru di atas merupakan keadaan awal yang harus diselesaikan. Cari jalur solusi dari semua keadaan awal (initial state) di atas dengan batas kedalaman adalah 4.



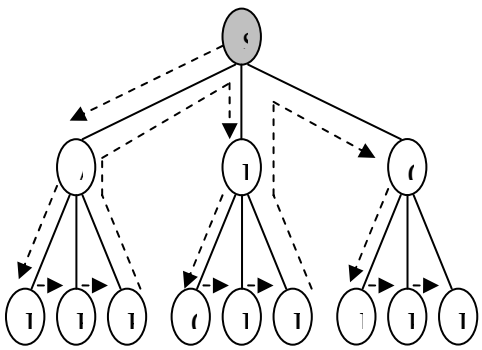
A. Pelacakan Melebar Pertama (*Breadth First Search*)

Pada metode *breadth first search*, semua simpul pada level n akan dikunjungi terlebih dahulu sebelum mengunjungi simpul-simpul pada level n+1. Pelacakan dimulai dari simpul akar diteruskan pada level ke-1 dari kiri ke kanan, kemudian berpindah ke level berikutnya dengan dilakukan hal yang sama sampai solusi ditemukan. Jika solusi belum ditemukan proses pelacakan dilakukan pada semua simpul sampai kedalaman yang telah ditentukan.



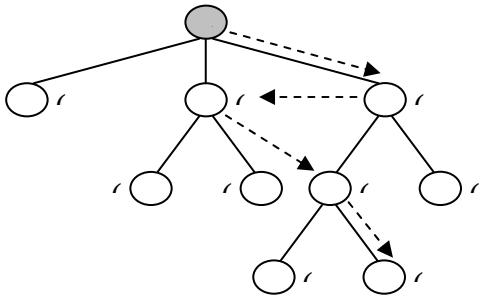
B. Pelacakan Mendalam Pertama (*Depth First Search*)

Pada metode *depth first search*, proses pelacakan akan dilakukan pada semua simpul anak pertama sebelum dilakukan pelacakan ke simpul-simpul yang selevel. Proses ini diulang sampai kedalaman tertentu. Jika sampai dengan batas kedalaman belum ditemukan, maka pelacakan dilanjutkan pada simpul di atas terdekat yang selanjutnya menelusuri kembali ke simpul anak pertama.



C. Pelacakan Terbaik Pertama (*Best First Search*)

Best first search adalah salah satu metoda pelacakan heuristik yang merupakan kombinasi dari 2 metoda pelacakan buta yaitu breadth first search dan depth first search dengan mengambil kelebihan dari kedua metoda tersebut. Pada pelacakan terbaik pertama, pelacakan diperbolehkan mengunjungi simpul yang ada di level yang lebih rendah, jika ternyata simpul pada level yang lebih tinggi memiliki nilai heuristik lebih buruk.



Gambar di atas menunjukkan metoda best first search. Diasumsikan bahwa simpul dengan nilai heuristik yang lebih kecil, memiliki nilai evaluasi yang lebih baik. Pengujian dilakukan pada level pertama, simpul C memiliki nilai heuristik terbaik, sehingga simpul C diambil sebagai langkah selanjutnya. Simpul C memiliki cabang F dan G. Dari semua simpul yang belum terpilih baik di level n atau di level-level sebelumnya cari simpul yang memiliki nilai heuristik yang lebih baik, dalam hal ini simpul B yang lebih baik. Simpul B memiliki cabang D dan E. Dengan melakukan hal yang sama cari simpul yang memiliki nilai heuristik lebih baik. Untuk selanjutnya simpul yang terpilih adalah simpul F. Simpul F memiliki cabang H dan I. Dengan demikian simpul I merupakan pilihan terbaik, dan begitu seterusnya.

Perbandingan Metoda Pelacakan Breadth, Depth, dan Best First search.

Persoalan	Initial State	Goal State	Batas Kedalaman	Jumlah Pemeriksaan Simpul		
				Depth	Breadth	Best
1	023156478	123456780	4	5	16	7
2	123506478	123456780	4	23	30	9
3	013426758	123456780	4	20	24	9
4	130426758	123456780	4	20	24	9
5	023146758	123456780	4	9	18	9
6	123746058	123456780	4	20	24	9
7	120463758	123456780	4	9	18	9
8	123468750	123456780	4	20	24	9
9	012453786	123456780	4	30	30	7
10	152403786	123456780	4	32	34	9
11	023145786	123456780	4	12	20	9
12	123745086	123456780	4	23	26	9
13	013425786	123456780	4	23	26	9
14	130425786	123456780	4	23	26	9
15	123485076	123456780	4	9	18	9
16	123485760	123456780	4	9	18	9

IV. Kesimpulan

- Metode depth secara kebetulan akan menemukan solusi tanpa harus menguji lebih banyak lagi dalam ruang keadaan. Ini terlihat pada persoalan 1.
- Metode breadth selalu memberikan jalur solusi yang terpendek. Hal ini disebabkan proses pelacakan yang menyelesaikan semua simpul dalam satu level sebelum pindah ke level berikutnya.
- Proses pelacakan pada metode best akan mempersempit ruang pelacakan. Hal ini diperlihatkan pada jumlah node yang dibentuk atau diperiksa untuk setiap persoalan lebih sedikit dibandingkan dengan dua metode yang lainnya.
- Kecepatan komputasi disetiap metode bergantung pada jumlah simpul yang dibentuk dan diperiksa. semakin sedikit jumlah pemeriksaan maka waktu yang dibutuhkan menjadi lebih singkat dan sebaliknya semakin banyak jumlah pemeriksaan maka waktu yang dibutuhkan menjadi lebih lama. metode best first search pada setiap persoalan membutuhkan waktu relatif singkat dalam mencari jalur solusi karena jumlah pemeriksaannya lebih sedikit.

Daftar Pustaka

[1] Suparman, (1991), “Mengenal Artificial Intelligence”, Andi Offset, Yogyakarta.

[2] Setiadi, (1993), “Artificial Intelligence”, Andi Offset, Yogyakarta.

[3] Sri Kusumadewi, (2003), ”Artificial Intelligence (Teknik dan Aplikasinya)”, Graha Ilmu, Yogyakarta.

[4] George F.G and William A.S, (1993), “Artificial Intelligence, Structures and Strategies for Complex Problem Solving”, University Of New Mexico, Albuquerque