

Optimasi Penyelesaian Permainan Rubik's Cube Menggunakan Algoritma IDA* dan Brute Force

Chichi Rizka Gunawan¹, Ahmad Ihsan², Munawir³

^{1,3} Program Studi Teknik Informatika Universitas Samudra

Jl. Meurandeh, Langsa Lama, Kota Langsa, Aceh 24354 INDONESIA

1 chichirizkagunawan@gmail.com ²ahmadihsan@unsam.ac.id ³munawir@unsam.ac.id

Abstrak— Permainan rubik's cube merupakan salah satu permainan yang penuh tantangan dan digemari anak muda. Bermain rubik's cube itu mengasyikkan. Selain permainan kerangkaan logika, permainan ini juga dituntut bekerja keras untuk menyelesaikannya. Bagi orang-orang tertentu jenis permainan ini sulit untuk diselesaikan. Kesulitan itu karena seseorang harus berpikir berulang kali agar dapat menyamakan warna-warna pada setiap sisinya. Rubik's cube adalah permainan kubus yang berukuran 3 x 3 x 3. Pemain berupaya menyelesaikan rubik's cube dengan memutar enam warna yang berbeda di seluruh kubus sampai setiap dari enam sisi menunjukkan warna tertentu yang sama. Untuk menyelesaikan rubik's cube dapat digunakan berbagai macam algoritma. Rubik's cube dapat dicari penyelesaiannya dan solusi yang dihasilkan cukup singkat. Pada kali ini akan dikemukakan optimasi penyelesaian rubik's cube dengan menggunakan algoritma IDA* dan algoritma Brute Force.

Kata kunci— algoritma IDA*, algoritma Brute Force, Rubik's Cube

Abstract— Rubik's cube game is one of the most challenging and popular games of the young. Playing rubik's cube is fun. In addition to logic clever games, this game is also required to work hard to solve it. For certain people this type of game is difficult to complete. The difficulty is because one has to think over and over to be able to match the colors on each side. Rubik's cube is a 3 x 3 x 3 cube game. The player attempts to finish the rubik's cube by rotating six different colors across the cube until each of the six sides shows the same particular color. To finish rubik's cube can be used various kinds of algorithm. Rubik's cube can be searched for and the resulting solution is quite short. At this time will be proposed optimization of rubik's cube solving by using IDA * algorithm and Brute Force algorithm.

Keywords— IDA * algorithm, Brute Force algorithm, Rubik's Cube

I. PENDAHULUAN

Rubik's cube adalah permainan dalam bentuk teka-teki mekanik yang dibuat pada tahun 1974 oleh seorang pemahat dan profesor arsitektur dari Hongaria, Erno Rubik. Setiap sisi dari kubus memiliki sembilan permukaan yang terdiri atas enam warna berbeda. Permainan rubik ini harus dipecahkan agar setiap sisi dari rubik punya satu warna. Sekitar tahun 1974 permainan ini sangat digemari masyarakat Hongaria. Akan tetapi, begitu banyak teka-teki yang harus diselesaikan dan butuh waktu yang lama untuk memecahkan permainan ini (Zulhaydar Fairozal Akbar, Entin Martiana, Setia Wardhana, Rizky Yuniar, 2011).

Masalah yang terdapat pada rubik's cube adalah bagaimana caranya agar rubik's cube dalam keadaan acak dapat kembali pada keadaan yang tersusun. Yang menjadi masalah selanjutnya adalah bagaimana caranya agar rubik's cube kembali pada keadaan yang tersusun dengan langkah terpendek. Untuk menjawab permasalahan yang terdapat pada rubik's cube dapat dikembangkan suatu kecerdasan buatan. Kecerdasan buatan atau artificial intelligence merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia. Kecerdasan buatan yang dikembangkan bertujuan menemukan solusi dari rubik's cube (Ardhan Wahyu R, Purwanto, Susy Kuspambudi A, 2006).

II. METODOLOGI PENELITIAN

A. Rubik's Cube

Rubik's cube adalah permainan kubus 3 x 3 x 3 yang memiliki warna yang berbeda pada tiap sisinya. Kubus ini terdiri atas dua puluh enam kubus kecil yang disatukan. Satu kubus kecil di tengah di anggap tidak ada karena berperan sebagai sumbu kubus saat memutar sisinya. Sebuah rubik's cube dapat memiliki $(8! \times 3^{8-1}) \times (12! \times 2^{12-1})/2 = 43,252,003,274,489,856,000$ posisi warna yang berbeda [2].

F	Depan: Sisi yang saat ini dipandang
B	Belakang: Sisi yang bertolak belakang dengan sisi Depan
Up	Atas: Sisi sebelah atas dari sisi Depan
Under	Bawah: Sisi yang bertolak-belakang dengan sisi Atas
L	Kiri: Sisi yang langsung berhubungan dengan sisi Depan dari arah kiri
R-	Kanan: Sisi yang langsung berhubungan dengan sisi Depan dari arah kanan

Tabel 1. Notasi sisi Rubik's Cube

B. Iterative Deepening A* (IDA*)

IDA* merupakan algoritma yang lebih memiliki kesamaan dengan algoritma yang lebih memiliki kesamaan dengan algoritma DFS dengan mengubah nilai batas kedalaman menjadi nilai $f(n)$. Sebuah node akan dipotong jika total $f(n)$ melebihi

cost batas maksimum. Iterative-Deepening A* (IDA*) search algorithm adalah pengembangan dari A* search algorithm yang dikombinasikan dengan iterative Deepening search. IDA* search algorithm merupakan best first search yang optimal dalam hal solution cost, time, dan space. Prinsip algoritma iterative deepening search adalah melakukan depth-limited search secara bertahap dengan nilai l yang incremental.

IDA* tidak menggunakan pemrograman dinamis, sehingga pada iterasi IDA* akan terdapat kemungkinan untuk melakukan eksplorasi berulang-ulang pada sebuah node, sehingga proses yang dibutuhkan akan lebih lama dan jumlah visited node akan lebih banyak. Pada algoritma IDA*, setiap iterasi akan mengembalikan nilai f -limit baru yang akan digunakan sebagai batasan pencarian untuk iterasi berikutnya. Algoritma IDA* juga complete dan optimal. Tetapi, karena dilakukan secara iteratif, maka IDA* mungkin membangkitkan simpul-simpul yang sama secara berulang-ulang. Hal ini membutuhkan waktu yang lama. Dengan kata lain, time complexity-nya tinggi. Bagaimanapun, keuntungan utama IDA* adalah jumlah memori yang dibutuhkan menjadi jauh lebih sedikit, jika menghadapi masalah keterbatasan memori, maka IDA* bisa menjadi pilihan yang tepat [7].

C. Brute Force

Brute Force adalah sebuah pendekatan langsung (straight forward) untuk memecahkan suatu masalah, yang biasanya didasarkan pada pernyataan masalah (problem statement) dan definisi konsep yang dilibatkan. Pada dasarnya algoritma Brute Force adalah alur penyelesaian suatu permasalahan dengan cara berpikir yang sederhana dan tidak membutuhkan suatu pemikiran yang lama. Sebenarnya, algoritma Brute Force merupakan algoritma yang muncul karena pada dasarnya alur pikir manusia adalah Brute Force (Langsung/to the point).

Beberapa karakteristik dari algoritma Brute Force dapat dijelaskan sebagai berikut :

1. Membutuhkan jumlah langkah yang banyak dalam menyelesaikan suatu permasalahan sehingga jika diterapkan menjadi suatu algoritma program aplikasi akan membutuhkan banyak memori.
2. Digunakan sebagai dasar dalam menemukan suatu solusi yang lebih efektif.
3. Banyak dipilih dalam penyelesaian sebuah permasalahan yang sederhana karena kemudahan cara berpikirnya [8].

Berikut ini adalah beberapa keunggulan algoritma Brute Force :

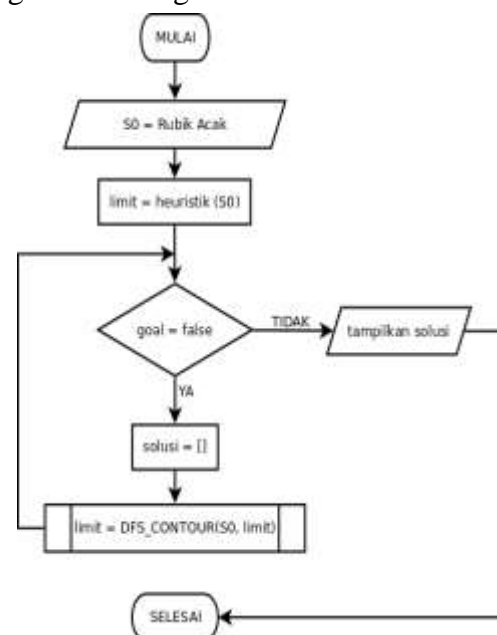
1. Brute Force umumnya tidak “cerdas” dan tidak mangkus, karena ia membutuhkan jumlah langkah yang besar dalam penyelesaiannya. Kadang-kadang algoritma Brute Force disebut juga algoritma naif (naive algorithm).
2. Algoritma Brute Force seringkali merupakan pilihan yang kurang disukai karena ketidakmangkusannya 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.

D. Algoritma

Secara umum algoritma untuk menyelesaikan permainan Rubik adalah :

1. Cube di acak sesuai dengan banyaknya kemungkinan pergerakan.
2. Cube yang sudah di acak dan yang belum dipecahkan diputar sesuai dengan banyaknya kemungkinan pergerakan.
3. Posisi cube yang dikunjungi tidak dikunjungi lagi.
4. Setiap kemungkinan disimpan kedalam sebuah queue dengan setiap anak dari posisi ditempatkan dipaling belakang.
5. Langkah terus diulangi hingga solusi ditemukan (tersusun).

a. Diagram Alir Algoritma IDA*

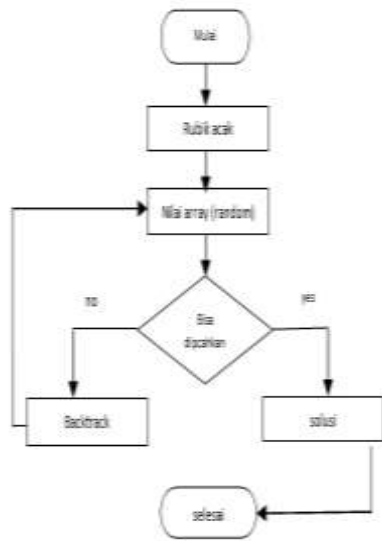


Gambar 1. Flowchart dari algoritma IDA* pada game

Algoritma IDA* merupakan salah satu algoritma yang termasuk dalam kategori pencarian terbimbing sehingga pencarian yang dilakukan dengan menggunakan algoritma ini bergantung dari nilai

heuristik. Perhatikan pada gambar 1.

b. Diagram Alir Algoritma Brute Force



Gambar 2 : Diagram alir algoritma Brute Force

Algoritma Brute Force memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas. Algoritma brute force pada umumnya tidak mangkus, karena ia membutuhkan jumlah langkah yang besar dalam penyelesaiannya, terutama bila masalah yang dipecahkan berukuran besar.

III. HASIL DAN PEMBAHASAN

A. Hasil Algoritma IDA*

Acak		
No	Putaran	Waktu
1	12	0:06:81
2	14	0:07:53
3	16	0:09:28
4	21	0:11:18
5	36	0:17:31

Tabel 2. Tabel acak pada algoritma IDA*

Memberikan gambaran umum tentang runtime acak untuk setiap putaran dari titik n ke tujuan. Semakin kecil jumlah putaran maka waktu yang digunakan singkat untuk mencapai tujuan.

Susun		
No	Putaran	Waktu
1	12	0:06:15
2	14	0:06:94

3	16	0:07:86
4	21	0:10:28
5	36	0:16:92

Tabel 3. Tabel susun pada algoritma IDA*

Gambaran umum tentang runtime susun untuk setiap kali putaran dari titik n ke tujuan. Bisa dilihat dari kedua tabel diatas jumlah putaran acak dan susun memiliki jumlah yang sama. Jika 12 putaran acak maka untuk menyelesaikan rubik dengan algoritma IDA* juga melewati 12 putaran susun.

B. Hasil Algoritma Brute Force

Acak		
No	Putaran	Waktu
1	26	0:23:43
2	27	0:23:52
3	27	0:23:55
4	28	0:24:03
5	29	0:23:41

Tabel 4. Tabel acak pada algoritma Brute Force

Tabel 4 : memberikan gambaran umum runtime acak untuk setiap kali putaran dari titik n ke tujuan. Perhatikan proses acak pada gambar 3. Dimana putaran yang terjadi yaitu sebanyak 23 putaran untuk mencapai tujuan. Dan pada gambar 4 adalah putaran untuk mencapai solusi dimana putaran yang terjadi sebanyak 107 putaran.



Gambar 3. Proses acak menggunakan algoritma Brute Force

Susun		
No	Putaran	Waktu
1	121	01:18:62
2	116	01:35:36
3	133	02:01:64
4	143	02:02:03
5	123	01:55:58

Tabel 5. Tabel susun pada algoritma Brute Force

Gambaran umum tentang runtime susun untuk setiap kali putaran dari titik n ke tujuan. Bisa dilihat dari kedua tabel diatas jumlah putaran acak dan susun memiliki jumlah yang berbeda. Dimana untuk putaran acak melewati beberapa putaran saja, sedangkan putaran untuk menyelesaikan dengan menggunakan algoritma brute force ini membutuhkan jumlah putaran yang banyak

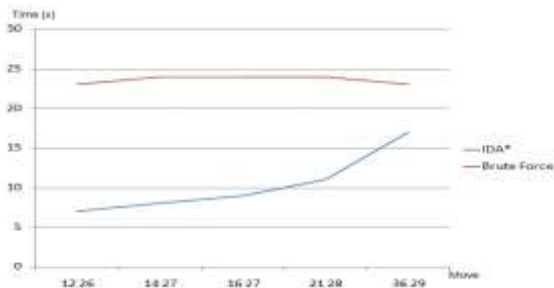
sehingga membutuhkan waktu yang lama. Perhatikan tabel 4, tabel 5 dan gambar 4.



Gambar 4. Proses susun menggunakan algoritma Brute Force

C. Perbandingan waktu algoritma IDA* dan algoritma Brute Force pada Rubik's cube

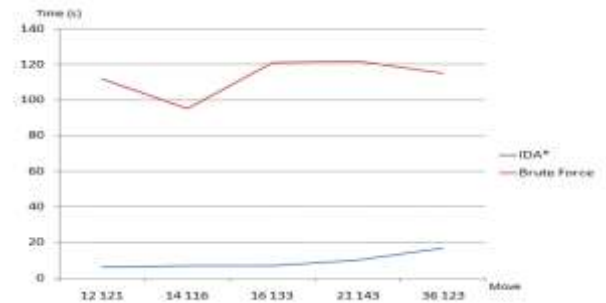
Perbandingan hasil pengujian proses acak dan susun pada rubik's cube berdasarkan waktu dan jumlah putaran untuk mencapai solusi. Perhatikan gambar 5 dan gambar 6.



Gambar 5. Grafik hasil pengujian pengacakan rubik pada algoritma IDA* dan Brute Force

Berdasarkan grafik diatas hasil pengujian pengacakan rubik dari kedua algoritma ini, pada algoritma IDA* waktu yang ditempuh untuk mencapai solusi sangatlah singkat dibandingkan algoritma Brute Force. Waktu yang ditempuh paling singkat pada algoritma IDA* adalah 6s dimana putaran yang terjadi sebanyak 12 kali. Sedangkan pada algoritma Brute Force waktu paling singkat yang ditempuh adalah 23s dengan putaran yang terjadi sebanyak 26 kali.

Berdasarkan grafik diatas hasil pengujian penyusunan rubik dari kedua algoritma ini, pada algoritma IDA* waktu yang ditempuh untuk mencapai solusi sangatlah singkat dan berbanding jauh dengan algoritma Brute Force. Waktu yang ditempuh paling singkat pada algoritma IDA* adalah 6s dimana putaran yang terjadi sebanyak 12 kali. Sedangkan pada algoritma Brute Force waktu paling singkat yang ditempuh adalah 115s dengan putaran yang terjadi sebanyak 121 kali.



Gambar 6. Grafik hasil pengujian penyusunan rubik pada algoritma IDA* dan Brute Force

IV. KESIMPULAN

Algoritma IDA* dan Brute Force dapat digunakan untuk menyelesaikan permasalahan rubik's cube. Berdasarkan hasil, kemampuan algoritma IDA* untuk menyelesaikan masalah rubik kurang efektif karena rubik dapat diselesaikan lebih cepat dari kemampuan manusia. Namun kemampuan ini dinilai lebih cepat dibandingkan menggunakan Brute Force. Dilihat dari hasil yang diperoleh pada algoritma Brute force sangat membutuhkan waktu yang lama untuk mendapatkan solusi. Dan jumlah pergerakan rubik's cube tidak berpengaruh dalam menemukan solusi. Dan dapat disimpulkan juga pada algoritma IDA* jumlah putaran acak dan susun untuk mencapai solusi berjumlah sama, sedangkan pada algoritma brute force proses acak dan susun jumlah putarannya berbeda.

REFERENSI

- [1] Wahyu R, Ardhan. dkk. 2006. *Kecerdasan Buatan Untuk Menyelesaikan Rubik's Cube Dengan Algoritma IDA**. Malang : Universitas Negeri Malang.
- [2] Pardede, Abigael Angele. dkk. 2006. *Penerapan Algoritma Genetika Pada Permainan Rubik's Cube*. Bandung : Institut Teknologi Bandung.
- [3] Akbar, Zulhaydar Fairozal. dkk. 2011. *Penyelesaian Permainan Rubik's Cube Dengan Metode Algoritma Genetika*. Surabaya : Institut Teknologi Sepuluh Nopember.
- [4] Smith, Robert. 2016. *Evolving Policies To Solve The Rubik's Cube: Experiments With Ideal And Approximate Performance Functions*. Dalhousie University.
- [5] Fahrurrozi. *Implementasi Algoritma Iterative Deepening A* Dan Metode Pruning Pada Solusi Permainan Puzzle Flow Free Color*. Bandung : FTIK UNIKOM.
- [6] Niasri, Tommi. 2015. *Implementasi Penyelesaian Permainan Rubik Cube dengan Algoritma Kociemba pada Platform Android*. Medan : Universitas Sumatera Utara.

- [7] Delima, Rosa. dkk. 2016. *Perbandingan Penerapan Algoritma A*, IDA*, Jump Point Search, dan PEA* pada Permainan Pacman*. Yogyakarta : Universitas Kristen Duta Wacana.
- [8] Saragih, May Aprina. 2013. *Implementasi Algoritma Brute Force dalam Pencocokan Teks Font Italic untuk Kata Berbahasa Inggris pada Dokumen Microsoft Office Word*. Medan : STMIK Budidarma Medan.