



PERBANDINGAN ALGORITMA A* (A STAR) DAN IDA* (ITERATIF DEEPENING A*) PADA PERMAINAN SLIDING PUZZLE

Puguh Riawang^{a,*}, Mamika Ujianita Rhomdini^a, Irwansyah^a

^aProgram Studi Matematika FMIPA Universitas Mataram, Jl. Majapahit No. 26, Mataram, 83125, Indonesia.
Email: Puguh.w.jaya@gmail.com

ABSTRACT

One of the games discussed in this study is a puzzle game. In a puzzle game players are expected to be able to achieve the final goal to form a correct picture or pattern. The algorithms used in this study are the A algorithm (A Star) and IDA* algorithm (iterative deepening A*). This study aims to determine the completion process of the game Sliding Puzzle using A* Algorithm and IDA* Algorithm and determine which algorithms are more efficient. Based on the time complexity it was found that the A* algorithm is more efficient than the IDA* algorithm because it only chooses the smallest value of f , apart from the IDA* algorithm which tries all possible steps with the same value of f . The disadvantages of these two algorithms are that in some cases this algorithm cannot solve the problem or stop at the local optimum or the same step repeated.*

Keywords: *sliding puzzle, A* Algorithm and IDA* Algorithm*

ABSTRAK

Salah satu *game* yang dibahas dalam penelitian kali ini adalah *game puzzle*. Dalam permainan *puzzle* pemain diharapkan dapat mencapai tujuan akhir untuk membentuk sebuah gambar atau pola yang benar. Algoritma yang digunakan dalam penelitian ini adalah algoritma A* (A Star) dan algoritma IDA* (*iterative deepening A**). Penelitian ini bertujuan untuk menentukan proses penyelesaian permainan *Sliding Puzzle* menggunakan Algoritma A* dan Algoritma IDA* serta menentukan mana algoritma yang lebih efisien. Berdasarkan kompleksitas waktu didapatkan bahwa algoritma A* lebih efisien daripada algoritma IDA* karna hanya memilih nilai f terkecil, berbeda dengan algoritma IDA* yang mencoba semua kemungkinan langkah yang nilai f sama. Adapun kekurangan dari kedua algoritma ini adalah dalam beberapa kasus algoritma ini tidak dapat menyelesaikan masalah atau berhenti di optimum lokal atau langkah yang sama diulang-ulang.

Keywords: *sliding puzzle, Algoritma A* dan Algoritma IDA**

Diserahkan: 12-12-2019; Diterima: 31-12-2019;

Doi: <https://doi.org/10.29303/emj.v1i2.39>

* Corresponding author.

Alamat e-mail: Puguh.w.jaya@gmail.com

1. Pendahuluan

1.1. Latar Blakang

Game merupakan salah satu media hiburan yang menjadi pilihan hampir setiap orang untuk menghilangkan kejenuhan ataupun hanya untuk mengisi waktu luang. Namun, sekarang *game* tidak hanya sebatas sarana hiburan semata, tetapi telah berkembang menjadi salah satu media edukasi yang memiliki pola pembelajaran untuk menyelesaikan suatu permasalahan agar dapat meningkatkan perkembangan otak seseorang. *Game* edukasi juga memiliki kelebihan dibandingkan dengan metode pembelajaran konvensional karena cara pembelajarannya disajikan dengan visualisasi bergerak yang menarik. Salah satu *game* yang dibahas dalam penelitian kali ini adalah *game puzzle*.

Puzzle terdiri dari berbagai jenis, di antaranya ada yang menggunakan angka, huruf dan gambar. Dalam permainan *puzzle* pemain diharapkan dapat mencapai tujuan akhir untuk membentuk sebuah gambar atau pola yang benar dengan waktu yang cepat. *Game puzzle* merupakan bentuk permainan yang menantang daya kreatifitas dan ingatan namun tetap menyenangkan sebab bisa diulang-ulang. Tantangan dalam permainan ini akan selalu memberikan efek ketagihan untuk selalu mencoba hingga berhasil. Dalam proses penyelesaian *puzzle*, terdapat banyak algoritma yang dapat diterapkan, diantaranya menggunakan algoritma A* (*A Star*) dan algoritma IDA* (*iterative deepening A**). Algoritma A* mengunjungi simpul dalam graf dengan cara mengunjungi simpul yang paling mendekati solusi sehingga pohon ruang setatus tidak perlu dibangkitkan seluruhnya (Prasetyo,2013).

Algoritma IDA* (*iterative deepening A**) merupakan Algoritma A* yang kedalamannya dibatasi pada setiap iterasi pencariannya. Walaupun kompleksitas waktunya lebih tinggi dibanding dengan Algoritma A*, namun Algoritma IDA* tidak membutuhkan memori sebesar yang dibutuhkan algoritma A*.

Oleh karena itu akan dianalisis Algoritma A* dan Algoritma IDA* (*iterative deepening A**) pada *sliding puzzle* menggunakan bantuan pemrograman *Octave*.

1.2. Rumusan Masalah

Berdasarkan latar belakang didapatkan rumusan masalah sebagai berikut :

1. Bagaimana proses penyelesaian permainan *Sliding Puzzle* menggunakan Algoritma A* dan Algoritma IDA* ?
2. Algoritma mana yang lebih efektif untuk menyelesaikan permainan *sliding puzzle* ?

1.3. Tujuan Penelitian

Berdasarkan rumusan masalah tersebut maka didapatkan tujuan dari penelitian ini adalah:

1. Menentukan proses penyelesaian permainan *Sliding Puzzle* menggunakan Algoritma A* dan Algoritma IDA*.
2. Menentukan algoritma yang paling efektif dalam menyelesaikan permainan *sliding puzzle*.

1.4. Manfaat Penelitian

Manfaat yang diharapkan dalam penelitian ini adalah:

1. Bagi peneliti mampu menginterpretasi teori kedalam kasus yang nyata.
2. Bagi pembaca mampu memberikan pemahaman kepada pembaca tentang bagaimana cara kerja kecerdasan buatan pada permainan *Sliding Puzzle* yang menggunakan Algoritma A* dan Algoritma IDA*.
3. Bagi pemain *Sliding Puzzle* mampu mendapatkan beberapa alternatif dalam menyelesaikan permainan *Sliding Puzzle*.

1.5 Batasan Masalah

Batasan masalah yang dilakukan pada penelitian ini adalah sebagai berikut:

1. Tujuan awal dan tujuan akhir harus ditentukan.
2. Jika langkah sudah memenuhi tujuan permainan dianggap selesai.
3. *Puzzle* yang digunakan berukuran 3x3.
4. Pergerakan *puzzle* tidak boleh diagonal.

2 Metodologi Penelitian

2.1 Jenis Penelitian

Penelitian ini merupakan penelitian terapan yang dimana menerapkan ilmu tentang kecerdasan buatan untuk memecahkan masalah bagi pemain permainan *sliding puzzle*, agar meminimalisir jumlah langkah dalam menyelesaikan permainan ini. Serta mengembangkan metode yang pernah digunakan pada penelitian sebelumnya.

2.2 Alat Penelitian

Menggunakan laptop atau *smartphone* untuk mencari langkah penyelesaikan *game puzzle*, dan menggunakan metode algoritma A* (*A star*) dan IDA* (*Iterative Deepening A Star*) untuk mempermudah dalam

menyelesaikan permainan *Sliding Puzzle* dengan bantuan aplikasi octave.

2.3 Jenis dan Sumber Data

Jenis dan sumber data dari penelitian ini adalah sebagai berikut :

3.3.1 Jenis Data

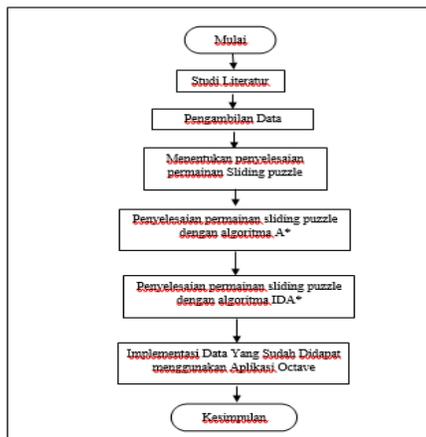
Jenis data yang digunakan dalam penelitian ini adalah data kuantitatif, yaitu data yang dapat dihitung atau diukur dengan angka dalam penelitian ini data yang digunakan berupa kemungkinan langkah untuk menyelesaikan permainan *sliding puzzle*.

3.3.2 Sumber data

Data yang diperoleh dalam penelitian ini adalah data primer. Data primer yaitu data yang diperoleh langsung dari percobaan yang dilakukan

2.4 Langkah – Langkah Penelitian

Adapun langkah-langkah yang dilakukan dalam penelitian ini adalah sebagai berikut:



Keterangan :

1. Mulai
Merupakan langkah awal untuk memulai suatu penelitian.
2. Studi literatur
Membaca, mempelajari dan mencari buku-buku yang berkaitan dengan penelitian yang akan dilakukan. Studi literatur dilakukan untuk mendapatkan teori-teori yang akan dijadikan landasan.
3. Pengambilan Data
Pengambilan data pada penelitian ini diambil dengan cara menyelesaikan permainan

Sliding Puzzle dan mencatat setiap kemungkinan langkah yang akan diambil menuju titik tujuan, data yang di dapatkan berupa *Node* (jumlah kotak), Sisi (langkah dari kotak satu ke kotak berikutnya), dan bobot (jumlah kmungkinan langkah menuju titik yang benar)

4. Menyelesaikan permainan *Sliding Puzzle*

Dengan cara menyelesaikan permainan *sliding puzzle* didapatkan data atau langkah-langkah yang harus diambil untuk mencapai tujuan akhir.

5. Penyelesaian permainan *sliding puzzle* dengan algoritma A*

Dalam hal ini permainan *sliding puzzle* diselesaikan dengan menerapkan algoritma A*.

6. Penyelesaian permainan *sliding puzzle* dengan algoritma IDA*

Dalam hal ini permainan *sliding puzzle* diselesaikan dengan menerapkan algoritma IDA*.

7. Implementasi hasil dengan menggunakan program Octaf

Setelah didapatkan hasil dari penerapan algoritma A* dan algoritma IDA* kemudian hasilnya diimplementasikan kedalam program octav sehingga diperoleh hasil.

8. Kesimpulan

Setelah didapatkan hasilnya kemudian di bandingkan mana yang lebih efisien algoritma A* atau Algoritma IDA*.

3. Hasil dan Pembahasan

Penelitian ini bertujuan untuk menentukan proses penyelesaian permainan *sliding puzzle* berukuran 3x3 menggunakan algoritma A* dan algoritma IDA*. Berdasarkan batasan masalah *puzzle* yang digunakan berupa *sliding puzzle* berukuran 3x3 sepertri pada gambar.



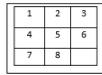
3.1 Algoritma A*

Algoritma A* dirumuskan sebagai berikut: $f(x) = h(x) + g(x)$. Bobot perkiraan merupakan jumlah semua langkah $h(x)$ dalam satu kali pergerakan yang diambil sampai menuju posisi tujuan (*goal node*) yang telah ditentukan, dan bobot sebenarnya $g(x)$ merupakan jumlah pergeseran simpul dari posisi awal ke posisi yang kosong.

Pada penelitian ini perhitungan dilakukan dengan tujuan untuk meminimumkan jumlah langkah yang diambil serta meminimumkan jumlah kesalahan yang dilakukan dalam menentukan langkah pada permainan *sliding puzzle*. Perhitungan ini dimulai dari titik awal dimana pada batasan masalah dikatakan bahwa titik awal dan titik akhir ditentukan, kemudian dari titik tersebut akan ditentukan langkah selanjutnya sampai di *goal state* yang telah ditentukan.

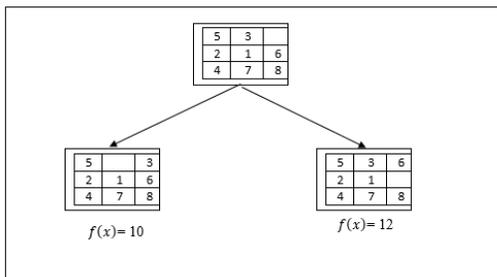


Gambar 4.2 Posisi Awal



Gambar 4.3 Goal State

Dari posisi awal inilah kemudian ditentukan kemungkinan langkah yang diambil dengan cara melakukan perhitungan menggunakan algoritma yang telah ditentukan yaitu algoritma A*. Perhitungan untuk algoritma A* dimulai dengan memperkirakan kemungkinan langkah yang bisa diambil dengan cara mengisi kotak kosong dengan angka yang berada disekitarnya kemudian dilakukan perhitungan untuk menentukan nilai $f(x)$ dari setiap langkah. Perhitungan yang dilakukan untuk menentukan nilai $h(x)$ dan $g(x)$ dengan metode *Manhattan distance* didapatkan perhitungan sebagai berikut : $h(x) = |X_n - X_{tujuan}| + |Y_n - Y_{tujuan}|$ dan $g(x) = |X_n - X_{awal}| + |Y_n - Y_{awal}|$ maka didapatkan:



Gambar 4.4 iterasi pertama dengan A*

Perhitungan untuk titik jika yang berpindah *node* angka 3 didapat perhitungan :

$$\begin{aligned} h_1 &= |1-0| + |1-0| = 2 & h_5 &= |0-1| + |0-1| = 2 \\ h_2 &= |0-1| + |1-0| = 2 & h_6 &= |2-2| + |1-1| = 0 \\ h_3 &= |2-2| + |0-0| = 0 & h_7 &= |1-0| + |2-2| = 1 \\ h_4 &= |0-0| + |2-1| = 1 & h_8 &= |2-1| + |2-2| = 1 \\ h &= \sum_{y=1}^8 h_y \\ &= 2 + 2 + 0 + 1 + 2 + 0 + 1 + 1 \\ &= 9 \end{aligned}$$

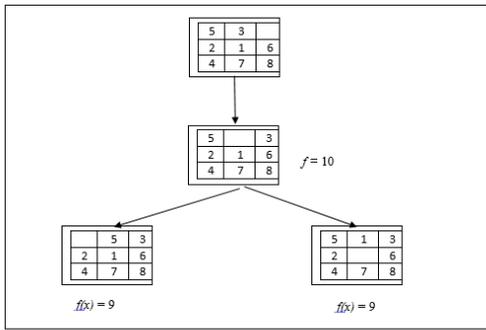
$$\begin{aligned} g_1 &= |1-1| + |1-1| = 0 & g_5 &= |0-0| + |0-0| = 0 \\ g_2 &= |0-0| + |1-1| = 0 & g_6 &= |2-2| + |1-1| = 0 \\ g_3 &= |2-1| + |0-0| = 1 & g_7 &= |1-1| + |2-2| = 0 \\ g_4 &= |0-0| + |2-2| = 0 & g_8 &= |2-2| + |2-2| = 0 \\ g &= \sum_{x=1}^8 g_x \\ &= 0 + 1 + 0 + 0 + 0 + 0 + 0 + 0 \\ &= 1 \\ f(x) &= h + g \\ &= 9 + 1 \\ &= 10 \end{aligned}$$

Sedangkan apabila *node* angka 6 yang bergerak didapatkan perhitungan sebagai berikut

$$\begin{aligned} h_1 &= |1-0| + |1-0| = 2 & h_5 &= |0-1| + |0-1| = 2 \\ h_2 &= |0-1| + |1-0| = 2 & h_6 &= |2-2| + |1-0| = 1 \\ h_3 &= |1-2| + |0-0| = 1 & h_7 &= |1-0| + |2-2| = 1 \\ h_4 &= |0-0| + |2-1| = 1 & h_8 &= |2-1| + |2-2| = 1 \\ h &= \sum_{y=1}^8 h_y \\ &= 2 + 1 + 2 + 2 + 1 + 1 + 1 + 1 \\ &= 11 \end{aligned}$$

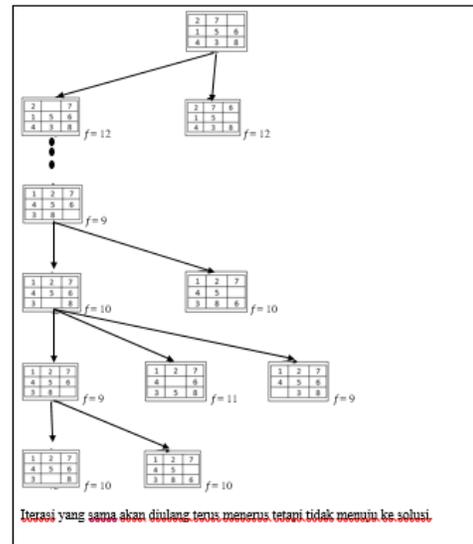
$$\begin{aligned} g_1 &= |1-1| + |1-1| = 0 & g_5 &= |0-0| + |0-0| = 0 \\ g_2 &= |0-0| + |1-1| = 0 & g_6 &= |2-2| + |1-0| = 1 \\ g_3 &= |1-1| + |0-0| = 0 & g_7 &= |1-1| + |2-2| = 0 \\ g_4 &= |0-0| + |2-2| = 0 & g_8 &= |2-2| + |2-2| = 0 \\ g &= \sum_{x=1}^8 g_x \\ &= 0 + 0 + 0 + 0 + 1 + 0 + 0 + 0 \\ &= 1 \\ f(x) &= h + g \\ &= 11 + 1 \\ &= 12 \end{aligned}$$

Setelah didapatkan hasil perhitungan kemudian dipilih nilai $f(x)$ paling kecil. Untuk iterasi pertama ini dipilih simpul 3 karna memiliki nilai $f(x)$ lebih kecil dibandingkan simpul 6, selanjutnya dicari kemungkinan langkah dari simpul kosong dengan memilih angka yang bertetangga. Untuk iterasi yang ke dua ini simpul yang mungkin pindah yaitu simpul 5 dan simpul 1, kemudian dilakukan perhitungan yang sama seperti pada iterasi pertama dan didapatkan nilai $f(x)$ sebagai berikut:

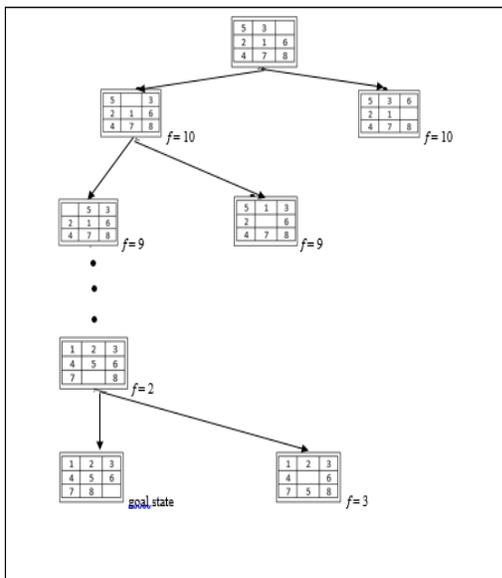


Gambar 4.5 iterasi kedua dengan A*

Jika ditemukan langkah dengan nilai yang sama maka dipilih salah satu langkah secara acak dan langkah yang diambil untuk iterasi ketiga ini adalah simpul 5 kemudian dilakukan perhitungan seperti iterasi-iterasi sesudahnya sampai ditemukan goal state atau penyelesaian, adapun penyelesaian untuk kasus menggunakan algoritma A* sebagai berikut:



Gambar 4.7 Percobaan Tidak Menuju Solusi dengan A*

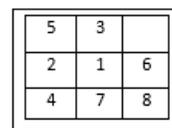


Gambar 4.6 Goal State dengan A*

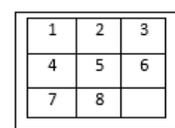
Adapun kekurangan dari algoritma ini adalah dalam beberapa kasus algoritma ini tidak dapat menyelesaikan masalah karena langkah yang sama diulang-ulang atau berhenti di optimum lokal seperti kasus berikut:

3.2 Algoritma IDA*

Algoritma IDA* adalah pengembangan dari algoritma A* yang dikombinasikan dengan algoritma IDS (*iterative deepening search*) dimana algoritma IDS menggunakan batasan level (kedalaman). Batasan kedalaman yang digunakan di algoritma IDA* adalah f_{limit} dimana f_{limit} merupakan nilai \bar{f} disimpul tertentu. Dalam bagian ini digunakan nilai $\bar{f} = \frac{1}{h + g} = \frac{1}{\bar{f}}$ dimana perhitungan h dan g menggunakan cara yang sama dengan di algoritma A*. ingat kembali simpul awal dan simpul akhir berikut:



Gambar 4.8 Posisi Awal



Gambar 4.9 Goal State

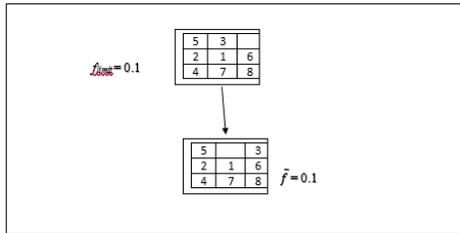
Dilakukan perhitungan pada simpul awal untuk menentukan nilai \bar{f} pertama dan f_{limit} pertama sebagai berikut:

$$\begin{aligned}
 h_1 &= |1-0| + |1-0| = 2 & h_5 &= |0-1| + |0-1| = 2 \\
 h_2 &= |0-1| + |1-0| = 2 & h_6 &= |2-2| + |1-1| = 0 \\
 h_3 &= |1-2| + |0-0| = 1 & h_7 &= |1-0| + |2-2| = 1 \\
 h_4 &= |0-0| + |2-1| = 1 & h_8 &= |2-1| + |2-2| = 1 \\
 h &= \sum_{y=1}^8 h_y \\
 &= 2 + 2 + 1 + 1 + 2 + 0 + 1 + 1 \\
 &= 10
 \end{aligned}$$

$$g = \sum_{x=1}^8 g_x = 0$$

$$f_{limit} = \frac{1}{h + g} = 0.1$$

Sama seperti algoritma A* ditentukan kemungkinan langkah yang pindah untuk dihitung nilai $f(x)$, setelah didapatkan nilai $f(x)$ kemudian ditentukan nilai \bar{f} dengan rumus $\bar{f} = \frac{1}{f(x)}$, apabila nilai \bar{f} lebih kecil dari nilai f_{limit} maka iterasi pada langkah tersebut dihentikan, kemudian dilakukan iterasi yang lebih mendalam untuk nilai \bar{f} yang lebih besar



Gambar 4.10 Iterasi pertama dengan IDA*

Untuk langkah pertama didapatkan perhitungan sebagai berikut, apabila nilai 3 yang bergerak.

$$h_1 = |1-0| + |1-0| = 2 \quad h_5 = |0-1| + |0-1| = 2$$

$$h_2 = |0-1| + |1-0| = 2 \quad h_6 = |2-2| + |1-1| = 0$$

$$h_3 = |2-2| + |0-0| = 0 \quad h_7 = |1-0| + |2-2| = 1$$

$$h_4 = |0-0| + |2-1| = 1 \quad h_8 = |2-1| + |2-2| = 1$$

$$h = \sum_{y=1}^8 h_y = 2 + 2 + 0 + 1 + 2 + 0 + 1 + 1 = 9$$

$$g_1 = |1-1| + |1-1| = 0 \quad g_5 = |0-0| + |0-0| = 0$$

$$g_2 = |0-0| + |1-1| = 0 \quad g_6 = |2-2| + |1-1| = 0$$

$$g_3 = |2-1| + |0-0| = 1 \quad g_7 = |1-1| + |2-2| = 0$$

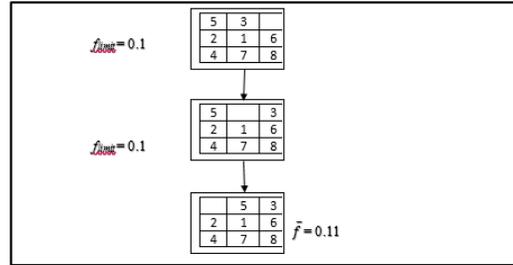
$$g_4 = |0-0| + |2-2| = 0 \quad g_8 = |2-2| + |2-2| = 0$$

$$g = \sum_{x=1}^8 g_x = 0 + 1 + 0 + 0 + 0 + 0 + 0 + 0 = 1$$

$$f(x) = h + g = 9 + 1 = 10$$

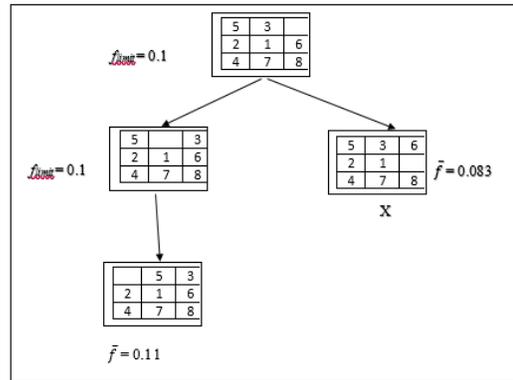
$$\bar{f} = \frac{1}{f(x)} = \frac{1}{10} = 0.1$$

Karena nilai \bar{f} yang didapatkan untuk iterasi pertama sama dengan nilai f_{limit} maka iterasi dilanjutkan untuk mendapatkan langkah selanjutnya dengan $f_{limit} =$ nilai \bar{f} yang dijalankan.



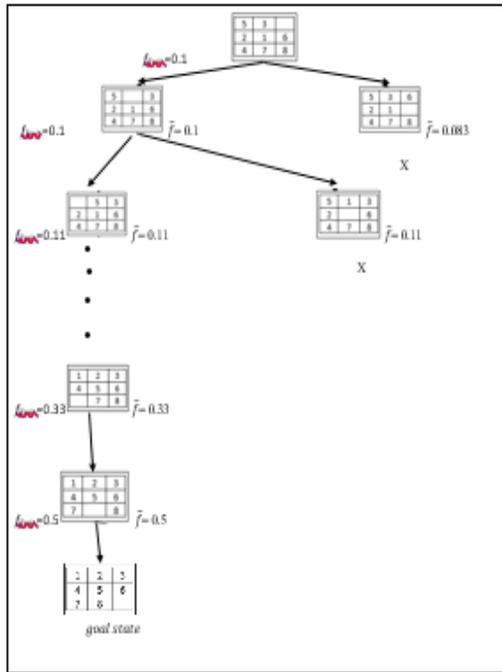
Gambar 4.11 Iterasi kedua dengan IDA*

Karena nilai \bar{f} yang didapatkan untuk iterasi selanjutnya lebih besar dari nilai f_{limit} maka naik 1 level ke current sebelumnya dan dilakukan iterasi pencarian untuk langkah yang lain.



Gambar 4.12 Iterasi Ketiga dengan IDA*

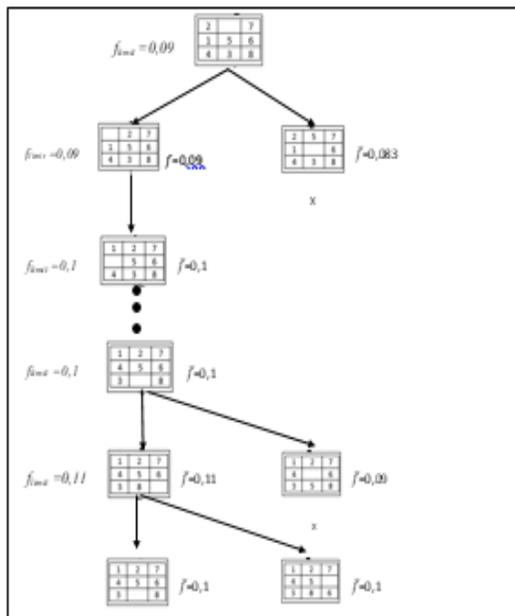
Karena nilai \bar{f} untuk simpul 6 lebih kecil dari f_{limit} maka iterasi untuk langkah ini dihentikan, iterasi dilanjutkan dengan nilai \bar{f} yang lebih besar dan langkah yang diproses itu yang akan menjadi nilai f_{limit} yang baru. Langkah yang sama dilakukan terus-menerus sampai goal state ditemukan.



Gambar 4.13 Goal State dengan IDA*

Adapun kekurangan dari algoritma IDA* ini sama seperti algoritma A* yaitu dalam beberapa kasus algoritma ini tidak dapat menyelesaikan masalah karena iterasinya berhenti di optimum lokal.

Contohnya



Gambar 4.14 Urutan Temu Solusi dengan IDA*

Karena nilai \bar{f} untuk simpul terakhir lebih kecil dari nilai f_{limit} maka iterasi untuk langkah ini dihentikan. Sehingga pada kasus di atas tidak ditemukan solusi penyelesaian tetapi berhenti di optimum lokal.

4 Kesimpulan dan Saran

4.1 Kesimpulan

Berdasarkan hasil pembahasan pada penelitian ini didapatkan kesimpulan sebagai berikut. Dilihat dari kompleksitas waktunya algoritma A* lebih efisien daripada algoritma IDA* karena pada algoritma IDA* kadang mengulang langkah yang sama tetapi kelebihan dari IDA* ini tidak membutuhkan memori yang banyak untuk penyelesaiannya. Adapun kekurangan dari kedua algoritma ini adalah apabila menemukan kasus tertentu algoritma ini tidak menemukan *goal state* tetapi berhenti di optimum lokal atau langkah yang sama akan diulang terus menerus sampai iterasinya berakhir.

4.2 Saran

Penelitian ini menggunakan Algoritma A* dan Algoritma IDA* untuk menentukan langkah yang dipilih pada permainan *sliding puzzle* dimana hanya memperhatikan bidak kosong untuk menentukan langkahnya. Penelitian ini menggunakan *puzzle* dengan ukuran 3x3 dan dengan titik awal yang sudah ditentukan. Sehingga diharapkan pada penelitian selanjutnya dapat menggunakan *puzzle* dengan ukuran yang lebih besar dan dengan titik awal yang dipilih secara acak.

DAFTAR PUSTAKA

Adipranata, Rudi, Andreas Handoyo & Happy Stiawan. 2007. *Aplikasi Pencari Rute Optimum pada Peta Guna Meningkatkan Efisiensi Waktu Tempuh Pengguna Jalandengan Metode A* dan Best First Search*. Surabaya: Jurnal INFORMATIKA Vol. 8, No. 2, 100 – 108.

- Brualdi, Richart A, 2010, *Introductory Combinatorics*, China Machine Press, Beijing.
- Hallimsah, Beny H, Eggy Margiso, 2014, *Problem Solving Permainan Puzzle 8 Menggunakan Algoritma A**, Pontianak, Vol. 4, No. 1.
- Harris, J.M., Jeffrey L.H., & Michael J.M., 2000, *Combinatorics and Graph Theory*, Springer, USA.
- Hidayatullah, Halilintar Nur, 2007, *Implementasi Algoritma IDA* (Iteratif Deepening A*) dalam Menentukan Solusi Terbaik Pada Permainan Otello 6x6 dengan Simulasi Matlab*, skripsi, Program Studi Matematika Fakultas MIPA Universitas Mataram.
- Hermianto, 2017, *Implementasi Algoritma Runtun Balik (Backtraking) Dalam Penyelesaian Permainan Puzzle Sudoku Berbasis Android*, Medan, Skripsi, Program Studi S1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informatika Universitas Sumatra Utara Medan.
- Munir, Rinaldi., 2014, *Matematika Diskrit*, Informatika, Bandung.
- Mutiana, Veronica. Fitria Amastini. & Noviana Mutiara. 2013. *Optimasi Pencarian Jalur Dengan Metode A* (A star)*. Tenggaraeng: Jurnal ULTIMATICS, Vol. 5, No. 2.
- Russell, Stuart J dan Peter Norvig. 2010. *Artificial Intelligence A Modern Approach Third Edition*. New Jersey.
- Suyanto. 2007. *Artificial Intelligence*. Informatika. Bandung.