

# PROBLEM SOLVING PERMAINAN PUZZLE 8 MENGUNAKAN ALGORITMA A\*

Beny Hakim Halimsah, Eggy Margiso

Sekolah Tinggi Manajemen Informatika dan Komputer Pontianak  
Jln. Merdeka No. 372 Pontianak, Kalimantan Barat

[Cloudrain@gmail.com](mailto:Cloudrain@gmail.com) , [Waterz.eggy@gmail.com](mailto:Waterz.eggy@gmail.com) dan [wtrs.eggy@yahoo.com](mailto:wtrs.eggy@yahoo.com)

## Abstrak

Tujuan penelitian ini adalah untuk membuat sgame Puzzle 8 pada perangkat lunak yang tersedia yaitu Visual Basic 6, dan untuk mengimplementasikan algoritma A\* yang dapat membantu dalam pemecahan solusi pada game Puzzle 8 dalam struktur ruang pohon. Sehingga pembuatan game puzzle 8 angka ini diharapkan mampu menjadi game edukasi yang bermanfaat bagi penggunanya. Bentuk penelitian yang dilakukan penulis adalah studi literature dengan teknik pengumpulan data berupa studi literatur, dokumentasi dan observasi. Metode perancangan perangkat lunak yang digunakan adalah RAD (Rapid Application Development). Perangkat analisis yang digunakan adalah algoritma, flowchart (bagan alir), UML (Unified Modelling Language), dan pseudocode. Teknik pengujian yang digunakan adalah pengujian blackbox. Berdasarkan dari analisis, implementasi, pembahasan yang telah dilakukan dalam implemantasi algoritma A\* pada game Puzzle 8 dapat membantu dalam mencari solusi atau tujuan dari permainan game Puzzle 8 sampai dengan angka yang teracak menjadi tersusun secara berurutan.

**Kata kunci:** game, kecerdasan buatan, Algoritma A\*, Puzzle 8

## Abstract

The purpose of this research is to make the game Puzzle 8 software available that Visual Basic 6 , and to implement the A\* algorithm that can help in solving the puzzle solution in 8 games in the structure of the tree space . Thus making games puzzle 8 this figure is expected to be useful education game for its users. Form of research by the author is the study of literature with data collection techniques such as literature studies, documentation and observation. Software design method that is used is a RAD (Rapid Application Development). The device used is the analysis algorithm, flowchart, UML (Unified Modeling Language), and pseudocode. Testing technique used is blackbox testing. Based on the analysis, implementation, discussion of which has been done in the A\* algorithm Implemantation on 8 puzzle games can help in finding a solution or goal of the game puzzle game with numbers 8 to be arranged in a sequence randomized .

**Keywords:** games, artificial intelligence, A\* algorithm, Puzzle 8

## 1. PENDAHULUAN

Permainan atau game adalah suatu struktur kegiatan, yang biasanya dilakukan untuk kesenangan dan kadang-kadang digunakan sebagai sarana pendidikan. Permainan berbeda dengan pekerjaan, yang biasanya di lakukan untuk mendapat suatu upah tertentu, atau dengan seni, yang lebih peduli dengan ekspresi ide. Namun adakalanya perbedaan itu menjadi tidak jelas, banyak permainan juga dinilai sebagai pekerjaan (seperti pemain olahraga profesional) atau sebagai seni karena melibatkan sebuah layout artistik (seperti puzzle 8 atau solitaire). Komponen kunci dari permainan adalah tujuan, aturan, tantangan, dan interaksi. Salah satu contoh permainan yang dibahas dalam penelitian ini adalah Puzzle 8.

---

Puzzle 8 terdiri dari beberapa jenis, ada yang menggunakan angka, huruf dan gambar. Dalam permainan puzzle 8, pemain diharapkan dapat mencapai tujuan akhir untuk membentuk sebuah puzzle 8 menjadi sebuah gambar atau pola yang benar dengan waktu yang cepat. Puzzle 8 adalah permainan sliding puzzle ciptaan Sam Loyd yang terdiri dari sebuah bingkai yang berisi kotak-kotak angka dalam keadaan teracak dimana sebuah kotak hilang. Karena permainan ini bernama puzzle 8, maka ukuran bingkai adalah 3 x 3. Tujuan dari puzzle 8 ini adalah dengan memakai gerakan geser yang menggunakan ubin kosong untuk mencapai goal state. Puzzle 8 merupakan salah satu implementasi dari Artificial Intelligence. Dalam proses penyelesaiannya banyak terdapat algoritma-algoritma pencarian yang dapat diterapkan. Algoritma A\* adalah algoritma pencarian graf yang menemukan jalur dari status awal ke status akhir. Algoritma ini menggunakan fungsi heuristic untuk menentukan urutan dimana pencarian dilakukan dengan mengunjungi node dalam pohon merepresentasikan biaya jalan yang paling efisien sampai ke tujuan. Fungsi heuristic yang digunakan Algoritma A\* untuk memecahkan kasus bervariasi tergantung dari kasus yang akan dihadapi.

Yuliana, dkk. (2012), melakukan penelitian tentang implementasi algoritma A\* pada pemecahan Puzzle 8. Puzzle 8 adalah sebuah permainan dimana terdapat sembilan kotak angka, huruf maupun gambar yang diacak, yang harus disusun kembali ke posisi yang benar dan teratur. Banyak metode yang dapat digunakan untuk menyelesaikan puzzle 8, salah satunya menggunakan metode best first search. Di dalam best first search terdapat sebuah fungsi yang dinamakan fungsi heuristic. Heuristik adalah sebuah teknik yang mengembangkan efisiensi dalam proses pencarian. Victor, dkk. (2005), melakukan kajian pustaka mengenai algoritma A\* sebagai salah satu contoh metode Pemrograman Branch and Bound. Algoritma A\* adalah salah satu algoritma pencarian yang cukup populer di kalangan pemrogram. Algoritma ini memberikan solusi yang cukup bagus bagi proses pathfinding (pencarian jalan), sehingga sering digunakan dalam pembuatan perangkat lunak berjenis permainan (game). Setyawan, dkk. (2012) dalam penelitian tugas akhirnya membahas mengenai optimasi rute perjalanan ambulance menggunakan Algoritma A\*. Dalam penelitiannya mencoba untuk membuat sebuah sistem routing yang diterapkan untuk mencari rute ambulans pada jaringan jalan untuk menjemput pasien dan mengantarkan ke rumah sakit. Jalan raya yang digunakan untuk tugas akhir, berasal dari data spasial, dimana pada kasus ini kita menggunakan Peta Surabaya.

Semakin sulit diacak susunan Puzzle 8, tentu akan semakin lama waktu yang dibutuhkan untuk memecahkan susunan puzzle 8 tersebut. Adakalanya dalam mengerjakan atau menyusun kembali pemain mengalami jalan buntu, untuk itu penulis membuat aplikasi yang dapat menemukan solusi dengan input yang acak namun tetap terikat pada peraturan dasar puzzle 8 dengan menggunakan metode algoritma A\* (A-star). Melihat hal tersebut, penulis mencoba untuk membuat suatu program aplikasi komputer berupa program aplikasi problem solving yang dapat melakukan pencarian solusi dari permainan puzzle 8 menggunakan algoritma A\*.

## 2. METODE PENELITIAN

Dalam penelitian ini, penulis menggunakan bentuk penelitian studi literatur. Penulis melakukan kajian yang berkaitan erat dengan permasalahan yang hendak dipecahkan serta mendefinisikan masalah dengan melakukan eksperimen. Metode penelitian yang digunakan oleh penulis adalah metode riset eksperimental. Data yang dikumpulkan sesuai dengan tujuan penelitian. Sumber data ada dua yaitu, data Sekunder, sumber data penelitian yang diperoleh peneliti secara tidak langsung dan data Primer, merupakan sumber penelitian yang diperoleh secara langsung dari sumber asli. Dalam mengumpulkan data penulis menggunakan metode studi literatur dan dokumentasi, termasuk di dalamnya semua variabel-variabel yang digunakan dalam proses perancangan aplikasi problem solving menggunakan algoritma A\* (A-Star). Dalam mengumpulkan data penulis menggunakan metode studi literatur dan dokumentasi yang digunakan dalam proses perancangan aplikasi problem solving menggunakan algoritma A\* (A-

Star). Data yang dikumpulkan menjadi dasar pengembangan sistem serta dasar bila akan memodifikasi atau memperbaiki sistem kemudian hari. Penelitian ini menggunakan metode perancangan RAD (Rapid Application Development). Adapun langkah-langkah yang dilakukan penulis yaitu : Bussiness Modeling, Data Modeling, Process Modeling, Application Modeling, serta testing and turnover.

### 3. HASIL DAN PEMBAHASAN

Pembuatan game puzzle angka yang akan dibuat ini akan mampu menampilkan urutan angka-angka dari 1 hingga 8 dalam kotak berukuran 3 x 3, yang kemudian angka-angka tersebut nantinya akan teracak secara system menjadi urutan-urutan yang tidak beraturan. Nantinya pengguna akan menyusun angka-angka yang tidak beraturan itu menjadi angka-angka yang tersusun kembali dari 1 hingga 8, atau hingga tujuan dari game ini ditemukan yaitu angka tersusun, jika pengguna merasa tidak menemukan jalan keluar atau tidak meneukan tujuan maka jika ingin mengakhiri permainan dapat langsung mengklik tombol exit diatas kanan game. Sehingga dapat disimpulkan bahwa game puzzle 8 (Puzzle Eight) ini mempunyai tujuan menyusun angka-angka yang teracak menjadi tersusun dengan benar.

Fungsi utama game puzzle 8 (Puzzle Eight) sebagai game edukasi adalah untuk membantu para pemain dalam mengasah otak dan sebagai fasilitator dalam belajar di bidang elektronik, pembuatan game puzzle 8 (Puzzle Eight) inipun tidak terlepas dari peran algoritma sebagai salah satu bentuk pemecahan masalah (problem solving) dari pengacakan angka tersebut agar menemukan tujuan atau goal dari suatu permainan dalam bentuk pohon/tree solution.

Adapun algoritma yang dapat di implementasikan dalam game puzzle angka ini dari beberapa macam algoritma yang ada penulis menggunakan algoritma A\* (A-star) karena algoritma ini di anggap mampu di implementasikan dalam game puzzle 8 (Puzzle Eight) ini.

Implementasi pertama dari algoritma A\* (A-star) adalah masing-masing node yang ada dalam puzzle 8 (Puzzle Eight) yang telah diacak semua memiliki kesempatan untuk dicari jalan keluar dengan pencarian secara vertical sehingga menemukan tujuan.

Implementasi yang kedua adalah jika hasil dari pencarian tujuan dengan menggunakan algoritma A\* (A-Star) tersebut mendapatkan hasil maka permainan game puzzle telah sukses, melalui proses aplikasi Visual Basic 6.

Setelah dilakukan implementasi pada game puzzle 8 (Puzzle Eight) dengan menggunakan algoritma tersebut yaitu A\* (A-star) yang diterapkan pada program Visual Basic 6 menggunakan struktur data Linked List dan Tree ini maka tujuan dari game puzzle 8 (Puzzle Eight) ini akan ditemukan dan proses pencarian angka yang tersusun dengan baik telah sukses.

Pada keadaan awal puzzle 8, algoritma A\* (A-star) akan bekerja dengan mengembangkan satu kemungkinan arah simpul yang dapat dibangun pada suatu akar pohon menjadi keadaan baru. Kemudian selanjutnya, algoritma A\* (A-star) akan mengembangkan lebih maju pencarian ke simpul berikutnya.

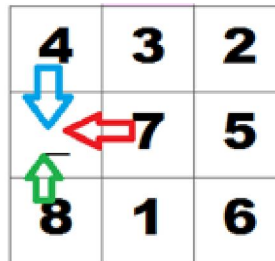
Bila tidak ada simpul yang dapat dikembangkan pada suatu jalur, maka metode A\* (A-star) akan kembali kejalur sebelumnya dan mengambil arah simpul yang lain dengan arah secara *vertikal/horizontal* yakni atas dan bawah. Keadaan ini berlanjut hingga simpul terakhir dikembangkan. Secara ringkas, algoritma proses A\* (A-star) dapat dijabarkan sebagai berikut, Masukkan simpul ujung (akar) kedalam tumpukan, Ambil simpul dari tumpukan teratas, lalu cek apakah simpul merupakan solusi, Jika simpul merupakan solusi, pencarian selesai dan hasil dikembalikan, Jika simpul bukan solusi, masukkan seluruh simpul yang bertetangga dengan simpul tersebut (simpul anak) kedalam tumpukan, Jika tumpukan kosong dan setiap simpul sudah dicek, pencarian selesai dan mengembalikan hasil solusi tidak ditemukan, Ulangi

pencarian dari langkah kedua. Pencarian simpul awal secara mendalam dan teroptimisasi hingga yang paling akhir atau sampai ditemukannya solusi.

Algoritma A\* (A-star) merupakan perbaikan dari metode BFS dengan memodifikasi fungsi heuristisnya. A\* (A-star) akan meminimumkan total biaya lintasan. Pada kondisi yang tepat, A\* (A-star) akan memberikan solusi yang terbaik dalam waktu yang optimal. Algoritma A\* (A-star) akan diterapkan dalam pencarian jalan terpendek pada game Puzzle 8. Pencarian jalan terpendek pada game Puzzle 8 akan diasumsikan dengan titik awal (keadaan pertama) dan titik tujuan (keadaan akhir) yang diimplementasikan pada pencarian di ruang kosong maupun di ruang dengan penghalang seperti halnya permainan Puzzle 8 pada umumnya. Ruang yang akan dibangun akan berbentuk puzzle 8 (kotak) supaya mempermudah dalam penerapan cara kerja algoritma A\*(A-star). Hasil pencarian algoritma tersebut akan menghasilkan jalan terpendek dari suatu titik awal ke titik tujuan.

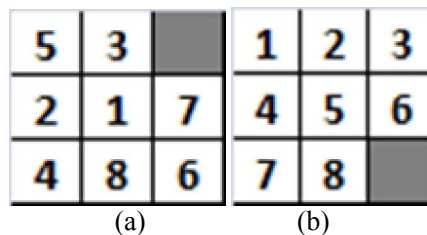
Untuk mengetahui performansi algoritma A\* (A-star) maka akan diadakan perbandingan terhadap langkah terpendek dan waktu. Algoritma A\* (A-star) adalah pengembangan dari BFS yang digunakan untuk mencari jalan terpendek (*shortest path*) yang sering dipakai dalam game programming.

Algoritma A\* (A-star) akan diterapkan pada analisis kasus pencarian solusi terpendek menuju tujuan pada game Puzzle 8 untuk mencari kecepatan, jarak langkah menuju tujuan dan simpul yang diperiksa. Dalam kondisi state yang acak dan rumit serta memiliki banyak kemungkinan solusi memungkinkan algoritma A\* (A-star) cocok untuk mencari solusi yang terbaik. Setiap pergerakan yang dilakukan pada kotak statusnya akan disimpan pada suatu list. List ini akan digunakan untuk melakukan pengecekan apakah kita sudah pernah membangun status tersebut atau belum agar kita tidak menggerakkan kotak yang sama berkali-kali ke status yang sama. Dengan menerapkan strategi ini, selain menemukan solusi, algoritma ini juga bisa menemukan langkah terpendek untuk mencapai solusi tersebut. Untuk lebih jelasnya, cara kerja algoritma A\* (A-star) dapat dilihat pada gambar berikut.



**Gambar 1.** Alur langkah yang dimungkinkan dalam Puzzle 8

Sebuah Game puzzle8 angka yang terdiri atas 3 buah kolom dan 3 baris di beri nomor 1 s.d. 8.



**Gambar 2.** Contoh Goal state dan Initial state, a) Initial state; b) Goal state

Pada gambar 2 diatas yang menggambarkan keadaan game puzzle angka yang teracak dari angka 1-8 yang disebut dengan initial state, dan pada gambar disampingnya merupakan goal state atau tujuan dari game puzzle agar angka tersusun secara tersusun dari angka 1-8. Pada

gambar dibawah akan dibahas mengenai istruktur ruang pohon yang ditelusuri oleh algoritma A\* (A-star). Dari mulai angka yang teracak sampai dengan menemukan solusi atau tujuan dari angka yang teracak dalam bentuk struktur pohon.



**Gambar 3.** Pohon ruang status yang dibangkitkan dengan algoritma A\* (A-Star)

Penyelesaian untuk masalah permainan game puzzle angka dengan ruang status yang

|   |   |   |
|---|---|---|
| 5 | 3 |   |
| 2 | 1 | 7 |
| 4 | 8 | 6 |

dibangkitkan dengan algoritma A\* (A-star) pada gambar 3.adalah simpul pertama(

|   |   |   |
|---|---|---|
| 5 |   | 3 |
| 2 | 1 | 7 |
| 4 | 8 | 6 |

|   |   |   |
|---|---|---|
| 5 | 3 | 7 |
| 2 | 1 |   |
| 4 | 8 | 6 |

dengan jumlah suksesor 2 yaitu simpul (

|   |   |   |
|---|---|---|
| 5 |   | 3 |
| 2 | 1 | 7 |
| 4 | 8 | 6 |

), dan (

|   |   |   |
|---|---|---|
| 5 | 1 | 3 |
| 2 |   | 7 |
| 4 | 8 | 6 |

Simpul ketiga (4 8 6) merupakan suksesor atau anak simpul dari simpul kedua, pada simpul ini dilihat perpindahan node dan jika belum ditemukan solusi maka pencarian akan

|   |   |   |
|---|---|---|
| 5 | 1 | 3 |
|   | 2 | 7 |
| 4 | 8 | 6 |

|   |   |   |
|---|---|---|
| 5 | 1 | 3 |
|   | 2 | 7 |
| 4 | 8 | 6 |

berlanjut pada simpul ke empat (4 8 6) Simpul keempat (4 8 6) merupakan suksesor atau anak simpul dari simpul ketiga, pada simpul ini dilihat perpindahan node dan jika

|   |   |   |
|---|---|---|
|   | 1 | 3 |
| 5 | 2 | 7 |
| 4 | 8 | 6 |

belum ditemukan solusi maka pencarian akan berlanjut pada simpul ke lima (4 8 6)

|   |   |   |
|---|---|---|
|   | 1 | 3 |
| 5 | 2 | 7 |
| 4 | 8 | 6 |

Simpul kelima (4 8 6) merupakan suksesor atau anak simpul dari simpul keempat, pada simpul ini dilihat perpindahan node dan jika belum ditemukan solusi maka pencarian akan

|   |   |   |
|---|---|---|
| 1 |   | 3 |
| 5 | 2 | 7 |
| 4 | 8 | 6 |

|   |   |   |
|---|---|---|
| 1 |   | 3 |
| 5 | 2 | 7 |
| 4 | 8 | 6 |

berlanjut pada simpul ke enam (4 8 6) Simpul keenam (4 8 6) merupakan suksesor atau anak simpul dari simpul kelima, pada simpul ini dilihat perpindahan node dan jika belum

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 5 |   | 7 |
| 4 | 8 | 6 |

ditemukan solusi maka pencarian akan berlanjut pada simpul ke tujuh (4 8 6) pada anak

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 5 |   | 7 |
| 4 | 8 | 6 |

simpul berikutnya. Simpul ke tujuh (4 8 6) merupakan anak simpul dari simpul ke enam jika pada simpul ini masih belum ditemukan solusi maka pencarian dilanjutkan pada simpul ke

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 5 | 7 |   |
| 4 | 8 | 6 |

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 5 | 7 |   |
| 4 | 8 | 6 |

delapan (4 8 6). Simpul kedelapan (4 8 6) merupakan suksesor atau anak simpul dari simpul ketujuh, pada simpul ini dilihat perpindahan node dan jika belum ditemukan solusi

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 5 | 7 | 6 |
| 4 | 8 |   |

maka pencarian akan berlanjut pada simpul ke sembilan (4 8) Simpul kesembilan (4 8)

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 5 | 7 | 6 |
| 4 | 8 |   |

merupakan suksesor atau anak simpul dari simpul kedelapan, pada simpul ini dilihat perpindahan node dan jika belum ditemukan solusi maka pencarian akan berlanjut pada

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 5 | 7 | 6 |
| 4 |   | 8 |

simpul kesepuluh( ) Dan begitu seterusnya sampai dengan menemukan solusi, pada

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 |   |

struktur pohon diatas solusi/goal ditemukan pada simpul ke empat belas ( ).

Algoritma A\* (A-star) menggunakan dua senarai yaitu OPEN dan CLOSED. OPEN adalah senarai (list) yang digunakan untuk menyimpan simpul-simpul yang pernah dibangkitkan dan nilai heuristiknya telah dihitung tetapi belum terpilih sebagai simpul terbaik (*best node*) dengan kata lain, OPEN berisi simpul-simpul masih memiliki peluang untuk terpilih sebagai simpul terbaik, sedangkan CLOSED adalah senarai untuk menyimpan simpul-simpul yang sudah pernah dibangkitkan dan sudah pernah terpilih sebagai simpul terbaik. Artinya, CLOSED berisi simpul-simpul yang tidak mungkin terpilih sebagai simpul terbaik (peluang untuk terpilih sudah tertutup).

OPEN LIST adalah list yang menyimpan kemungkinan path yang akan diperiksa. OPEN LIST dibuat terurut berdasarkan nilai f. OPEN LIST digunakan untuk menentukan secara selektif (berdasarkan nilai f) jalan yang dikira lebih dekat menuju pada path tujuan. OPEN berisi simpul-simpul yang masih memiliki peluang untuk terpilih sebagai simpul terbaik (*best node*). CLOSED adalah senarai (list) untuk menyimpan simpul-simpul yang sudah pernah dibangkitkan dan sudah pernah terpilih sebagai simpul terbaik (*best node*) atau senarai yang menyimpan jalan yang sudah diperiksa dari open list. Artinya, CLOSED berisi simpul-simpul yang tidak mungkin terpilih sebagai simpul terbaik (peluang untuk terpilih sudah tertutup). Kedua list (OPEN LIST dan CLOSED LIST) ini bertujuan juga untuk menghindari penelusuran berkali-kali jalan (rute) yang memang sudah diidentifikasi agar tidak masuk kembali ke dalam OPEN LIST.

Setiap simpul (node) harus memiliki informasi nilai h(n) sebagai Fungsi Heuristik, yaitu estimasi harga simpul tersebut dihitung dari simpul tujuan yang hasilnya menjadi nilai H. Fungsi f sebagai estimasi fungsi evaluasi terhadap node n, dapat dituliskan :

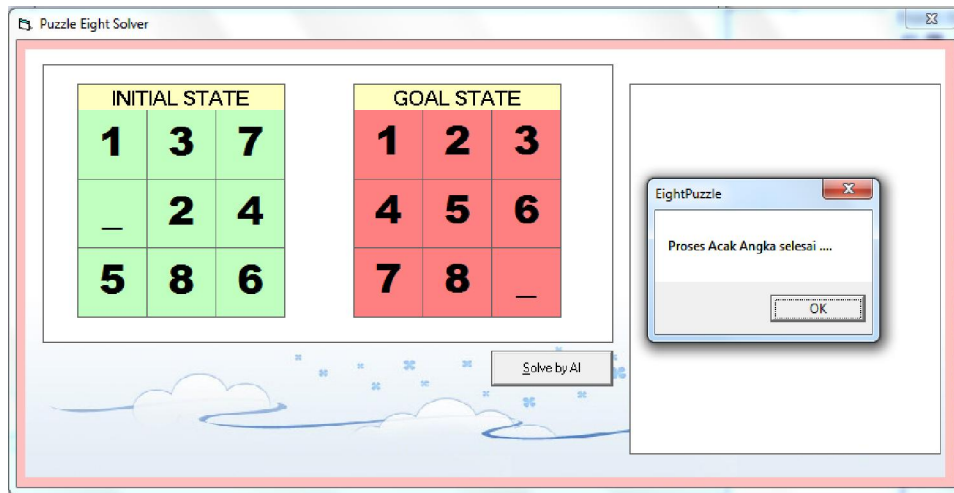
$$f(n) = g(n) + h(n) \text{ dengan :}$$

$$f(n) = \text{fungsi evaluasi ( jumlah } g(n) \text{ dengan } h(n) )$$

$$g(n) = \text{biaya (cost) yang dikeluarkan dari keadaan awal sampai keadaan n}$$

$$h(n) = \text{estimasi biaya untuk sampai pada suatu tujuan mulai dari n}$$

Pembuatan game puzzle 8 angka yang akan dibuat ini akan mampu menampilkan urutan angka-angka dari 1 hingga 8 dalam kotak berukuran 3 x 3, yang kemudian angka-angka tersebut nantinya akan teracak secara system menjadi urutan-urutan yang tidak beraturan. Nantinya pengguna akan menyusun angka-angka yang tidak beraturan itu menjadi angka-angka yang tersusun kembali dari 1 hingga 8, atau hingga tujuan dari game ini ditemukan yaitu angka tersusun, jika pengguna merasa tidak menemukan jalan keluar atau tidak menemukan tujuan maka jika ingin mengakhiri permainan dapat langsung mengklik tombol exit diatas kanan game. Sehingga dapat disimpulkan bahwa game puzzle 8 ini mempunyai tujuan menyusun angka-angka yang teracak menjadi tersusun dengan benar. Adapun tujuan dari perancangan game puzzle 8 ini adalah untuk membuat game puzzle 8 pada perangkat lunak yang tersedia yaitu Visual Basic 6, dan untuk mengimplementasikan algoritma A\* (A-Star) yang dapat membantu dalam pemecahan solusi pada game puzzle 8 dalam struktur ruang pohon. Sehingga pembuatan game puzzle 8 angka ini diharapkan mampu menjadi game edukasi yang bermanfaat bagi penggunaanya.



Gambar 4. Tampilan Awal Permainan

Setelah teracak seperti gambar di atas maka pengguna/user tinggal menekan tombol-tombol angka untuk menggeser kotak ke atas, bawah, kanan, dan kiri sesuai dengan penggerakan dalam pemindahan kotak-kotak puzzle angka.

Perancangan Form Puzzle 8 dibuat sedemikian rupa untuk memperoleh hasil yang optimal dengan berbagai aturan-aturan tertentu yang berlaku pada game puzzle 8 ini agar dapat menghasilkan antar muka antara perangkat lunak dan pemakai yang baik dan benar. Perancangan antar muka yang digunakan tidak dibuat serumit game puzzle 8 pada umumnya. Perancangan yang digunakan terdiri dari 2 Bagian yaitu *Initial State* dan *Goal State*. Disediakan tombol *Solve by AI* jika pemain kesulitan menyelesaikan permainan Puzzle 8. Disamping Initial state dan goal state, disediakan sebuah Listbox untuk menampilkan langkah-langkah yang digunakan untuk menyelesaikan game Puzzle 8 ini.



Gambar 5. Tampilan Solve by AI

Berikut ini merupakan solusi penyelesaian Puzzle 8 dimana representasi dari Kotak 3 x 3 ditulis dalam bentuk String.

- Langkah ke - 0 ==> 4 1 3 7 8 2 5 \_ 6
- Langkah ke - 1 ==> 4 1 3 7 \_ 2 5 8 6
- Langkah ke - 2 ==> 4 1 3 7 2 \_ 5 8 6
- Langkah ke - 3 ==> 4 1 3 7 2 6 5 8 \_
- Langkah ke - 4 ==> 4 1 3 7 2 6 5 \_ 8
- Langkah ke - 5 ==> 4 1 3 7 2 6 \_ 5 8
- Langkah ke - 6 ==> 4 1 3 \_ 2 6 7 5 8
- Langkah ke - 7 ==> \_ 1 3 4 2 6 7 5 8
- Langkah ke - 8 ==> 1 \_ 3 4 2 6 7 5 8



Langkah ke - 9 ==> 1 2 3 4 \_ 6 7 5 8

Langkah ke - 10 ==> 1 2 3 4 5 6 7 \_ 8

Langkah ke - 11 ==> 1 2 3 4 5 6 7 8 \_

Untuk mendapatkan hasil yang bervariasi dilakukan serangkaian Pengujian yang dilakukan sebanyak 5 kali sehingga didapatkan jumlah langkah pencarian solusi dari yang sedikit langkahnya sampai banyak langkah solusi.

**Tabel 1. Hasil Pengujian Puzzle 8**

| Pengujian ke - | Initial State | Jumlah Langkah |
|----------------|---------------|----------------|
| 1              | 2 5 3         | 7              |
|                | 1 7 6         |                |
|                | 4 8           |                |
| 2              | 4 1 3         | 11             |
|                | 7 8 2         |                |
|                | 5 6           |                |
| 3              | 2 4 3         | 16             |
|                | 1 7           |                |
|                | 8 5 6         |                |
| 4              | 2 8 3         | 66             |
|                | 4 5 7         |                |
|                | 1 6           |                |
| 5              | 6 8           | 78             |
|                | 3 1 2         |                |
|                | 4 7 5         |                |

#### 4. KESIMPULAN

Berdasarkan dari analisis, implementasi, pembahasan yang telah dilakukan dalam implementasi algoritma A\* (A-star) pada game puzzle 8 maka dapat ditarik kesimpulan dengan mengimplementasikan atau menerapkan algoritma A\* (A-star) dapat membantu dalam mencari solusi atau tujuan dari permainan game puzzle 8 sampai dengan angka yang teracak menjadi tersusun secara berurutan. Telah berhasilnya dibuat game puzzle 8 pada perangkat lunak Visual Basic 6. Yang telah dirancang dengan menghasilkan output angka sebanyak 8 Angka yang berhasil diacak dan terselesaikan dengan baik. Adapun saran yang dapat penulis uraikan yang mungkin dapat dijadikan referensi untuk pengembangan implementasi algoritma pada beberapa aplikasi khususnya pada aplikasi game edukasi, antara lain. Dapat mencoba metode algoritma yang lain untuk mendapatkan hasil yang lebih optimal.

#### 5. SARAN

Adapun saran yang dapat penulis uraikan yang mungkin dapat dijadikan referensi untuk pengembangan implementasi algoritma pada beberapa aplikasi khususnya pada aplikasi game edukasi, antara lain :

- a. Dapat mencoba metode algoritma yang lain untuk mendapatkan hasil yang lebih optimal.
- b. Diharapkan pada pengembangan aplikasi yang selanjutnya tampilan antar muka dapat dirancang semenarik mungkin sehingga lebih menarik perhatian pengguna.
- c. Diharapkan pengembangan game puzzle 8 (Puzzle Eight) berikutnya dapat lebih meningkatkan tingkat kesulitan, sehingga pengguna dapat memilih tingkat kesulitan yang akan dipilih.

#### DAFTAR PUSTAKA

- [1]. Gunardi, Tommy., 2008, *Penggunaan Algoritma a \* pada 8 puzzle problem*, Bandung, Institut Teknologi Bandung.
- [2]. Jokodo, 2011, *Game puzzle number*. Jakarta :Universitas Gunadharma
- [3]. Kusumadewi, Sri., 2007, *Teknik Pencarian Heuristik*. Yogyakarta : Universitas Islam Indonesia
- [4]. Suyanto, 2011, *Artificial Intelligence*. Bandung : Penerbit INFORMATIKA
- [5]. Perry, William E., 2006, *Effective Methods for Software Testing*, edisi ketiga, Wiley Publishing, Inc, IN.
- [6]. Yuliana., Ananda., Surya, Ibnu., 2012, *Implementasi Algoritma A-Star Pada Pemecahan Puzzle 8*, *Jurnal Teknik Informatika*, Vol. 1, September 2012, Hal. 1-9.
- [7]. Zakiah, Azizah., 2012, *Penyelesaian Masalah 8 Puzzle dengan algoritma Hill Climbing Stepest Ascent Loglist Heuristik berbasis Java*, *Seminar Nasional Teknologi Informasi dan Komunikasi 2012 (SENTIKA 2012)*, Yogyakarta, 10 Maret 2012, ISSN : 2089-9815, Hal. 158-163