

# PENGGUNAAN ALGORITMA A\* PADA 8 PUZZLE PROBLEM

Tommy Gunardi (13507109)

Institut Teknologi Bandung  
Sekolah Teknik Elektro dan Informatika, Program Studi Teknik Informatika  
Jl. Ganesha no 10, Bandung  
e-mail: if17109@students.if.itb.ac.id , tommy\_gunardi@hotmail.com

## ABSTRAK

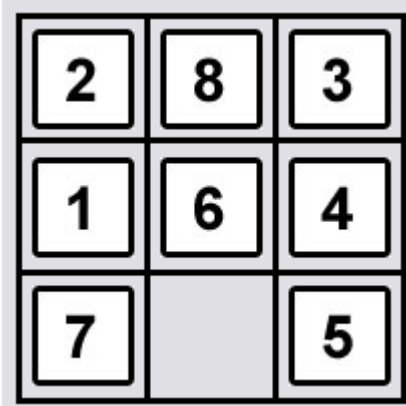
Puzzle sudah dikenal sejak dari dulu kala. Puzzle dikenal sebagai permainan edukatif untuk mengasah otak. Dalam permainan puzzle, pemain diharapkan merekayasa puzzle secara logik dengan harapan dapat membentuk bentuk, gambar, atau solusi yang diinginkan. Biasanya pemain yang memainkannya akan senang apabila berhasil menyelesaikan puzzlenya. Permainan Puzzle terdiri dari berbagai jenis permainan. Ada yang menggunakan kartu, angka, huruf, kata, maupun gambar. Untuk menyelesaikan jenis puzzle, kita dapat menggunakan beberapa macam algoritma seperti *brute force*, *greedy*, *depth first search* dan *breadth first search*, serta *A\** untuk memecahkannya. Meskipun algoritma penyelesaiannya banyak, namun tidak semua algoritma tersebut mangkus dalam menyelesaikan puzzle. Contoh puzzle yang akan dibahas adalah “the 8 puzzle”. Permainan ini melibatkan delapan buah angka yang diacak posisinya di dalam sembilan buah kotak (tiga kali tiga). Yang menjadi pokok permasalahan adalah bagaimana kita menyusun puzzle tersebut menjadi goal state dengan hanya menggeser satu demi satu angka ke tempat yang kosong. Yang akan dibahas lebih lanjut adalah bagaimana dengan algoritma *A\**, kita dapat menyelesaikan puzzle tersebut dengan langkah optimum atau seminimal mungkin.

**Kata kunci:** Puzzle, *A\**, the 8 puzzle, goal state.

## 1. PENDAHULUAN

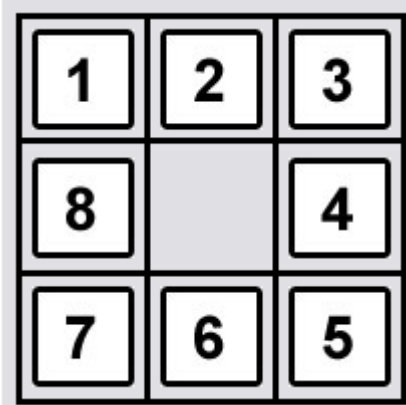
Apa itu 8 puzzle? 8 puzzle adalah permainan *sliding puzzle* ciptaan Sam Loyd yang terdiri dari sebuah bingkai yang berisi kotak-kotak angka dalam keadaan teracak dimana sebuah kotak hilang. Karena permainan ini

bernama 8 puzzle, maka ukuran bingkai adalah tiga kali tiga. Tujuan dari puzzle adalah dengan memakai gerakan geser yang menggunakan spasi kosong untuk mencapai goal state yang diinginkan.



2	8	3
1	6	4
7		5

Gambar Contoh State Awal



1	2	3
8		4
7	6	5

Gambar Contoh State Akhir

Dalam Sains komputer, Algoritma *A\** ( baca *A star*) adalah algoritma pencarian graf pertama dan terbaik yang menemukan path dengan cost minimum dari status awal ke *goal state*. Algoritma ini menggunakan fungsi *distance-plus-cost heuristic* ( biasanya didenotasikan dengan  $f(x)$  ) untuk menentukan urutan dimana pencarian dilakukan dengan mengunjungi node dalam pohon. Fungsi *distance-plus-cost heuristic* adalah penggabungan dari dua

fungsi, yaitu fungsi *path-cost* (didenotasikan  $g(x)$ ) dan *fungsi heuristic estimation* (didenotasikan  $h(x)$ ).

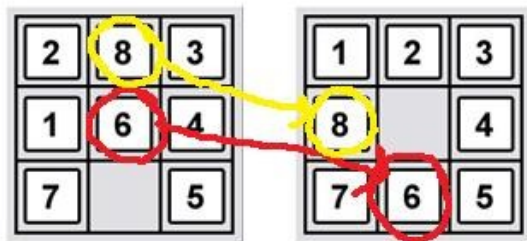
Fungsi  $g(x)$  menghitung biaya jalan antara node awal dengan simpul saat ini. Fungsi  $h(x)$  merupakan heuristik untuk menghitung perkiraan biaya jalur dari node saat ini dengan node tujuan. Dimana  $F(x) = g(x) + h(x)$ .  $F(x)$  merepresentasikan biaya jalan yang paling efisien sampai ke tujuan, oleh karena itu A\* terus mengevaluasi ulang kedua fungsi  $g(x)$  dan  $h(x)$  di himpunan solusi dalam rangka mencari biaya minimal untuk mencapai *goal state*.

Nilai heuristik untuk setiap puzzle berbeda-beda. Tidak ada kepastian tetap untuk suatu nilai heuristik. Sehingga untuk setiap kasus puzzle yang berbeda (jenisnya) kita harus menentukan nilai heuristik yang paling optimal untuk kasus tersebut.

## 2. METODE

### 2.1 Penentuan Heuristik

Cara penentuan heuristik pada masalah ini adalah dengan menggunakan metoda *Manhattan Distance*. *Manhattan Distance* adalah jumlah gerakan minimum yang diperlukan oleh sebuah kotak angka untuk berpindah ke goal statenya, dengan asumsi dia bisa bergerak bebas tanpa halangan. Untuk lebih jelasnya dapat dilihat gambar berikut.



State Sekarang

Goal State

Gambar Manhattan Distance

*Manhattan Distance* untuk angka 8 adalah 2 langkah, yaitu bergerak ke kiri lalu ke bawah atau sebaliknya bergerak ke bawah terlebih dahulu lalu ke kiri. Sedangkan *Manhattan Distance* untuk angka 6 adalah 1 langkah, yaitu ke bawah. *Manhattan Distance* untuk kasus ini juga dapat didapat dengan rumus :

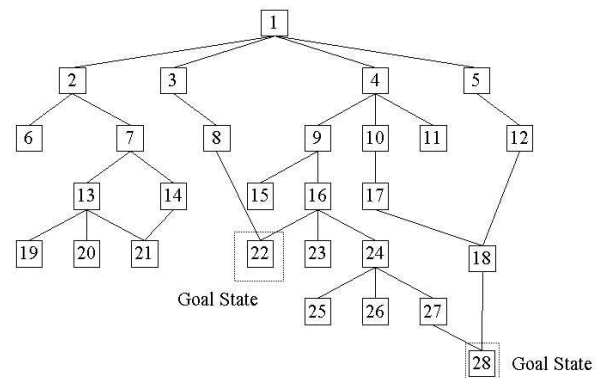
$$|\text{AbsisNow} - \text{OrdinatAkhir}| + |\text{AbsisAkhir} - \text{OrdinatNow}| (1)$$

### 2.2 Algoritma A\*

Alur Algoritma A\* untuk kasus 8 puzzle adalah sebagai berikut :

- Cari nilai heuristik untuk kotak angka yang berbatasan langsung dengan space
- Kotak dengan nilai heuristik paling besarlah yang akan dipilih lalu dimasukkan ke dalam list. Di sini list befungsi sebagai pencatat status agar tidak ada pengulangan terhadap gerakan.
- Apabila terdapat dua buah kotak dengan heuristik sama, yang digerakkan adalah kotak yang pertama.
- Alokasi next list lalu ulangi langkah di atas dengan kondisi terakhir (setelah kotak digerakkan)
- Setelah ditemukan semua solusi yang mungkin, dipilihlah solusi yang paling optimal

### An Example Search Space



Gambar Space Pencarian

### 2.3 Percobaan Perbandingan A\* dengan BFS

#### 1. State Awal

1 3 4  
8 b 5  
7 2 6

#### Goal State

1 2 3  
8 b 4  
7 6 5

#### Dengan A\*

Langkah ke solusi : Down Right Up Up Left Down

Jumlah langkah : 6

Jawaban ditemukan dalam pengecekan 6 state

#### Dengan BFS

Langkah ke solusi : Down Right Up Up Left Down

Jumlah langkah : 6

Jawaban ditemukan dalam pengecekan 69 state

1 b 5  
2 3 4

2. State Awal

2 3 1  
7 b 8  
6 5 4

Goal State

1 2 3  
8 b 4  
7 6 5

Goal State

1 2 3  
8 b 4  
7 6 5

**Dengan A\***

Langkah ke solusi : Right Up Left Down Right  
Down Left Left Up Right Up Left Down Right  
Jumlah langkah : 14  
Jawaban ditemukan dalam pengecekan 70 state

**Dengan A\***

Langkah ke solusi : Down Left Up Right Right  
Down Left Up Up Right Down Down Left Up  
Up Left Down Down Right Up Up Left Down  
Right Right Up Left Down  
Jumlah langkah : 28  
Jawaban ditemukan dalam pengecekan 10804 state

**Dengan BFS**

Langkah ke solusi : Right Up Left Down Right  
Down Left Left Up Right Up Left Down Right  
Jumlah langkah : 14  
Jawaban ditemukan dalam pengecekan 3685 state

**Dengan BFS**

Tidak ditemukan jawaban dalam pengecekan 100000 state

3. State Awal

2 3 1  
8 b 4  
7 6 5

5. State Awal

2 8 3  
1 b 4  
7 6 5

Goal State

Goal State

1 2 3  
8 b 4  
7 6 5

1 2 3  
8 b 4  
7 6 5

**Dengan A\***

Langkah ke solusi : Right Up Left Left Down  
Right Right Up Left Down Left Up Right Right  
Down Left  
Jumlah langkah : 16  
Jawaban ditemukan dalam pengecekan 292 state

**Dengan A\***

Langkah ke solusi : Up Left Down Right  
Jumlah langkah : 4  
Jawaban ditemukan dalam pengecekan 4 state

**Dengan BFS**

Langkah ke solusi : Right Up Left Left Down  
Right Right Up Left Down Left Up Right Right  
Down Left  
Jumlah langkah : 16  
Jawaban ditemukan dalam pengecekan 9137 state

**Dengan BFS**

Langkah ke solusi : Up Left Down Right  
Jumlah langkah : 4  
Jawaban ditemukan dalam pengecekan 21 state

4. State Awal

8 7 6

### 3. KESIMPULAN

Setelah melakukan percobaan, penulis dapat menyimpulkan beberapa kesimpulan sebagai berikut :

- Waktu eksekusi dan jumlah pengecekan A\* lebih sedikit dibanding BFS
- Path yang ditemukan Algoritma BFS dan algoritma A\* adalah sama, yaitu langkah minimum yang optimal

- Algoritma A\* berbanding lurus dengan heuristiknya. Semakin optimal heuristiknya, semakin optimal juga algoritma A\*-nya
- Algoritma A\* merupakan algoritma yang mangkus dalam pencarian langkah

## REFERENSI

- [1] Wikipedia Foundation, Inc. "A\* Search Algorithm" [http://en.wikipedia.org/wiki/A\\*\\_search\\_algorithm](http://en.wikipedia.org/wiki/A*_search_algorithm). Diakses tanggal 2 Januari 2010 pukul 21.00
- [2] [http://www.8puzzle.com/8\\_puzzle\\_algorithm.html](http://www.8puzzle.com/8_puzzle_algorithm.html). Diakses tanggal 2 Januari 2010 pukul 21.13
- [3] Munir, Rinaldi.2006. "Strategi Algoritmik". Departemen Teknik Informatika, Institut Teknologi Bandung. Diakses tanggal 2 Januari 2010 pukul 20.45
- [4] <http://kantz.com/jason/writing/8-puzzle.htm>. Diakses tanggal 2 Januari 2010 pukul 22.03