

# Penerapan Algoritma A-star (A\*) Untuk Menyelesaikan Masalah Maze

Hapsari Tilawah - 13509027<sup>1</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

<sup>1</sup>hapsaritulawah@gmail.com

*Abstract—* Adalah hal yang sangat wajar bagi setiap orang, apabila mereka bepergian dari suatu tempat ke tempat lainnya dengan melewati jalur yang mereka anggap sebagai jalur terpendek. Namun akan sangat sulit bagi seseorang untuk memilih jalur mana yang terpendek apabila terdapat banyak jalan buntu dan jalan akses untuk menuju ke tempat tujuan tersebut. Karena bisa saja jalur yang mereka pilih bukanlah jalur yang terpendek. Dalam makalah ini, masalah tersebut akan diselesaikan dengan menerapkan algoritma A-star(A\*). Algoritma A\* merupakan salah satu algoritma Branch & Bound yang menggunakan informasi tambahan (fungsi heuristik) dalam melakukan pencarian solusi. Algoritma ini akan memberikan solusi yang optimal terhadap suatu proses pathfinding (pencarian jalan), sehingga sering digunakan dalam pembuatan perangkat lunak berjenis permainan (game). Dengan penulisan makalah ini diharapkan memberikan wawasan kepada para pembaca karena Algoritma A\* dapat dijadikan sebagai salah satu contoh pemrograman branch and bound.

*Index Terms—*pathfinding, algoritma A-star, maze.

## 1. PENDAHULUAN

Perkembangan teknologi yang sangat cepat harus didukung dengan perkembangan dan implementasi algoritma yang mangkus, sehingga teknologi tersebut dapat memberikan hasil yang baik, efektif dan efisien.

Pada saat ini, banyak algoritma yang berkembang untuk mencapai solusi dari sebuah permasalahan. Namun, algoritma tersebut belum tentu memberikan solusi yang optimal terhadap masalah yang ada. Solusi yang optimal dapat dihasilkan dengan menambahkan sebuah informasi tambahan pada algoritma tersebut yang akan digunakan oleh algoritma sebagai dasar dalam pencarian solusi. Algoritma yang menggunakan informasi tambahan atau sering disebut fungsi heuristik dalam pencarian solusi optimal adalah algoritma Branch & Bound. Salah satu algoritma Branch & Bound yang banyak digunakan adalah algoritma A\*.

Dalam makalah ini, akan diterapkan algoritma A\* dalam menyelesaikan masalah maze. Maze atau labirin adalah sebuah puzzle dalam bentuk percabangan jalan yang kompleks dan memiliki banyak jalan buntu. Tujuannya adalah pemain harus menemukan jalan keluar dari sebuah pintu masuk ke satu atau lebih pintu keluar.

## 2. ALGORITMA A\*

### 2.1. Pengertian Algoritma A\*

Algoritma A\* adalah algoritma yang dikemukakan oleh Hart, Nilsson, dan Raphael pada tahun 1968. Algoritma A\* merupakan salah satu algoritma Branch & Bound atau disebut juga sebagai sebuah algoritma untuk melakukan pencarian solusi dengan menggunakan informasi tambahan (heuristik) dalam menghasikan solusi yang optimal.

### 2.2. Terminologi Dasar Algoritma A\*

Beberapa terminologi dasar yang terdapat pada algoritma ini adalah starting point, current node, simpul, neighbor node, open set, closed set, came from, harga (cost), walkability, target point.

Starting point adalah sebuah terminologi untuk posisi awal sebuah benda.

Current node adalah simpul yang sedang dijalankan dalam algoritma pencarian jalan terpendek.

Simpul adalah petak-petak kecil sebagai representasi dari area pathfinding. Bentuknya dapat berupa persegi, lingkaran, maupun segitiga.

Neighbor node adalah simpul-simpul yang bertetangga dengan current node.

Open set adalah tempat menyimpan data simpul yang mungkin diakses dari starting point maupun simpul yang sedang dijalankan.

Closed set adalah tempat menyimpan data simpul sebelum current node yang juga merupakan bagian dari jalur terpendek yang telah berhasil didapatkan.

Came from adalah tempat menyimpan data ketetanggaan dari suatu simpul, misalnya y came from x artinya neighbor node y dari current node x.

Harga (F) adalah nilai yang diperoleh dari penjumlahan nilai G, jumlah nilai tiap simpul dalam jalur terpendek dari starting point ke current node, dan H, jumlah nilai perkiraan dari sebuah simpul ke target point.

Target point yaitu simpul yang dituju.

Walkability adalah sebuah atribut yang menyatakan apakah sebuah simpul dapat atau tidak dapat dilalui oleh current node.

### 2.3. Fungsi Heuristik

Algoritma A\* menerapkan teknik heuristik dalam membantu penyelesaian persoalan. Heuristik adalah penilai yang memberi harga pada tiap simpul yang memandu A\* mendapatkan solusi yang diinginkan. Dengan heuristik yang benar, maka A\* pasti akan mendapatkan solusi (jika memang ada solusinya) yang dicari. Dengan kata lain, heuristik adalah fungsi optimasi yang menjadikan algoritma A\* lebih baik dari pada algoritma lainnya. Namun heuristik masih merupakan estimasi / perkiraan biasa saja Sama sekali tidak ada rumus khususnya. Artinya, setiap kasus memiliki fungsi heuristik yang berbedabeda. Algoritma A\* ini bisa dikatakan mirip dengan algoritma Dijkstra, namun pada algoritma Dijkstra, nilai fungsi heuristiknya selalu 0 (nol) sehingga tidak ada fungsi yang mempermudah pencarian solusinya.

Nilai ongkos pada setiap simpul n menyatakan taksiran ongkos termurah lintasan dari simpul n ke simpul target (target node), yaitu:

$F(n)$  = nilai taksiran lintasan termurah dari simpul status n ke status tujuan

Dengan kata lain,  $F(n)$  menyatakan batas bawah (lower bound) dari ongkos pencarian solusi dari status n. Fungsi heuristik yang terdapat pada algoritma A\* untuk menghitung taksiran nilai dari suatu simpul dengan simpul yang telah dilalui adalah:

$$F(n) = G(n) + H(n)$$

Dimana :

$F(n)$  = ongkos untuk simpul n

$G(n)$  = ongkos mencapai simpul n dari akar

$H(n)$  = ongkos mencapai simpul tujuan dari simpul n

### 2.4. Algoritma

Algoritma A\* secara ringkas langkah demi langkahnya adalah sebagai berikut.

- 1) Tambahkan starting point ke dalam open set.
- 2) Ulangi langkah berikut:
  - a) Carilah biaya F terendah pada setiap simpul dalam open set. Node dengan biaya F terendah kemudian disebut current node.
  - b) Masukkan ke dalam closed set.
  - c) Untuk setiap 8 simpul (neighbor node) yang berdekatan dengan current node:
    - Jika tidak walkable atau jika termasuk closed set, maka abaikan.
    - Jika tidak ada pada open set, tambahkan ke open set.
    - Jika sudah ada pada open set, periksa apakah ini jalan dari simpul ini ke current node yang lebih baik dengan menggunakan biaya G sebagai ukurannya. Simpul dengan biaya G yang lebih rendah berarti bahwa ini adalah jalan yang lebih

baik. Jika demikian, buatlah simpul ini (neighbor node) sebagai came from dari current node, dan menghitung ulang nilai G dan F dari simpul ini

d) Stop ketika Anda:

- menambahkan target point ke dalam closed set, dalam hal ini jalan telah ditemukan, atau
  - Gagal untuk menemukan target point, dan open set kosong. Dalam kasus ini, tidak ada jalan.
- 3) Simpan jalan. Bekerja mundur dari target point, pergi dari masing-masing simpul ke simpul came from sampai mencapai starting point. Itu adalah jalan Anda.

### 2.4. Kompleksitas Algoritma

Kompleksitas waktu dari A\* tergantung pada heuristiknya. Dalam kasus terburuk, jumlah simpul yang diperluas adalah eksponensial dari panjang solusi (jalur terpendek), tetapi polinomial ketika ruang pencarian adalah pohon, ada sebuah simpul tujuan tunggal, dan fungsi heuristik h memenuhi kondisi berikut:

$$|H(x) - h^*(x)| = O(\log h^*(x))$$

dimana  $h^*$  adalah heuristik optimal, biaya yang tepat yang didapat dari x ke tujuan. Dengan kata lain, kesalahan dari h tidak akan tumbuh lebih cepat dari logaritma dari "heuristik sempurna"  $h^*$  yang mengembalikan jarak yang sebenarnya dari x ke tujuan

## 3. MAZE

Maze (labirin) adalah permainan yang sudah tidak asing lagi. Maze adalah jaringan jalan yang rumit dan berkeluk-luk. Sejak zaman dahulu, maze telah digunakan dalam berbagai kepentingan, mulai dari proteksi keamanan hingga hiburan. Pada umumnya, maze dibuat untuk tujuan hiburan. Dalam kehidupan nyata, maze dapat ditemukan pada susunan jalan kecil atau gang-gang di kawasan perumahan. Sangat sulit bila seseorang yang asing dengan daerah tersebut untuk mencari jalan. Bila seseorang mengetahui metode untuk keluar dari sebuah maze, maka mereka dapat dengan mudah mengatasi kesulitan yang dirasakan.

Maze atau labirin adalah sebuah puzzle dalam bentuk percabangan jalan yang kompleks dan memiliki banyak jalan buntu. Tujuan permainan ini adalah pemain harus menemukan jalan keluar dari sebuah pintu masuk ke satu atau lebih pintu keluar. Bisa juga kondisi pemain menang yaitu ketika dia mencapai suatu titik di dalam maze tersebut.

Maze dalam dunia nyata banyak dibuat di taman atau ruangan-ruangan dengan pembatas berupa pagar tanaman, tembok atau pagar. Ukurannya bervariasi, tergantung ukuran ruangan atau taman tersebut. Labirin ini biasanya memang dirancang untuk menjadi sebuah atraksi permainan (misalnya rumah kaca) atau hanya sebagai hiasan saja. Selain itu banyak maze yang terbentuk secara "tidak sengaja". Contohnya jalan-jalan kecil atau gang-

gang yang terbentuk diantara rumah-rumah pada kawasan pemukiman. Maze-maze ini secara tidak langsung “menyesatkan” orang asing yang masuk ke dalamnya.

Pada umumnya pembuatan maze hanya untuk hiburan belaka. Namun, banyak bangunan yang menerapkan maze sebagai salah satu sistem keamanan agar orang yang tidak berkepentingan atau tidak dikenal sulit untuk masuk ke dalam bangunan. Maze untuk permainan biasanya dicetak dalam sebuah kertas untuk diselesaikan oleh pemain. Permainan dapat dilakukan dengan cara menuliskan jalan yang telah ditempuh menggunakan pensil atau hanya dengan menunjuk jalannya menggunakan jari. Maze terbagi menjadi beberapa kategori sesuai jenisnya, yaitu Maze 2 dimensi, 3 dimensi, bentuk segitiga, sigma, dan masih banyak lagi.

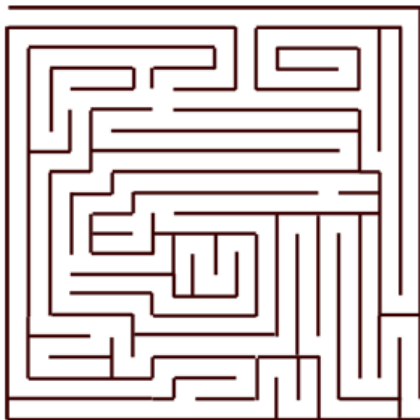


Fig. 2 Contoh maze dengan satu pintu keluar

#### 4. PEMBAHASAN MASALAH

Permasalahan maze dapat diselesaikan dengan menggunakan algoritma A\* dengan cara menyimpan data posisi tembok maze sebagai simpul yang tidak dapat dilalui (not walkable). Asumsikan bahwa ada seseorang yang ingin pergi dari titik simpul ke simpul B dan terdapat tembok memisahkan kedua simpul. Hal ini digambarkan di bawah ini, dengan kotak hijau menjadi simpul awal A, dan kotak merah menjadi simpul tujuan B, dan kotak biru yang menjadi tembok maze.

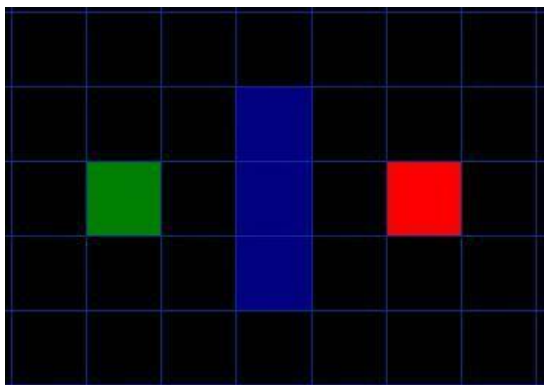


Fig. 2 Contoh maze sederhana

Kemudian memulai pencarian dengan melakukan hal berikut:

- 1) Mulailah di simpul awal A dan menambakkannya ke "open set". Sekarang hanya ada satu item pada open set, tetapi akan bertambah kemudian. Open set berisi simpul yang mungkin akan diambil sebagai jalan solusi tapi mungkin juga tidak. Pada dasarnya, ini adalah daftar simpul yang perlu diperiksa.
- 2) Lihatlah semua simpul berdekatan dengan simpul awal yang dapat dijangkau atau walkable, abaikan simpul yang merupakan tembok maze. Tambahkan simpul tersebut ke dalam "open set". Untuk masing-masing simpul, simpan simpul A sebagai "came from"-nya. Hal tersebut penting ketika akan melacak jalan solusi.
- 3) Hapus simpul A dari open set, dan menambakkannya ke dalam "closed set" dan tidak perlu melihat lagi untuk saat ini.

F dihitung dengan menambahkan G dan H. Hasil langkah pertama dalam pencarian dapat dilihat pada ilustrasi di bawah ini. Border kotak yang berwarna hijau merupakan node yang termasuk ke dalam open set. Skor F, G, dan H ditulis dalam kotak masing-masing.

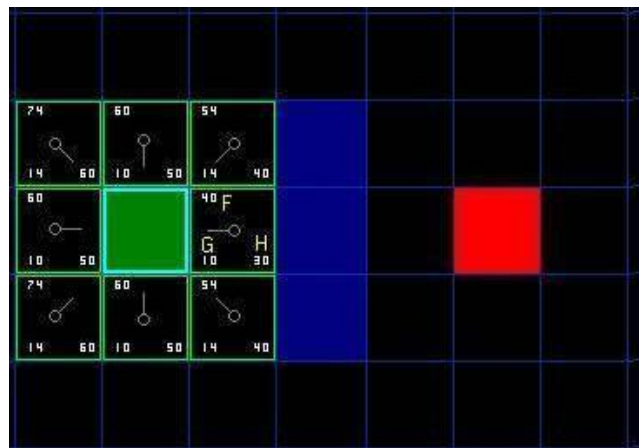


Fig. 3 Hasil langkah awal

Dalam kotak dengan huruf di dalamnya,  $G = 10$ . Hal ini karena hanya satu persegi dari simpul awal dalam arah horizontal. Kotak tepat di atas, bawah, dan ke kiri dari simpul awal semua memiliki skor G sama 10. Kotak diagonal memiliki skor G 14.

Nilai H dihitung dengan memperkirakan jarak ke simpul tujuan, bergerak hanya secara horizontal dan vertikal dan mengabaikan tembok maze. Menggunakan metode ini, simpul di sebelah kanan simpul awal berjarak tiga simpul dari simpul target sehingga skor H adalah 30. Simpul tepat di atas simpul awal berjarak 4 simpul (ingat, hanya bergerak horizontal dan vertikal) sehingga skor H adalah 40.

Skor F untuk masing-masing persegi, sekali lagi, hanya dihitung dengan menambahkan G dan H bersama-sama.

Untuk melanjutkan pencarian, pilih nilai F terendah dari semua simpul yang ada di open set. Kemudian lakukan langkah berikut kepada simpul yang terpilih:

- 4) Hapus simpul dari open set dan menambahkannya ke closed set.
- 5) Periksa semua simpul yang berdekatan. Abaikan simpul yang termasuk closed set atau not walkable (tembok maze), tambahkan simpul ke open set jika belum berada pada open set. Simpan data "came from" simpul tersebut.
- 6) Jika simpul yang berdekatan sudah ada pada open set, periksa apakah jalan tersebut merupakan yang lebih baik. Dengan kata lain, periksa apakah nilai G untuk simpul lebih rendah jika menggunakan simpul saat ini ke sana. Jika ya, ubah came from dari simpul yang berdekatan dengan simpul yang dipilih. Akhirnya, hitung ulang baik nilai F dan G dari simpul itu. Jika hal ini membingungkan, lihat ilustrasi di bawah ini.

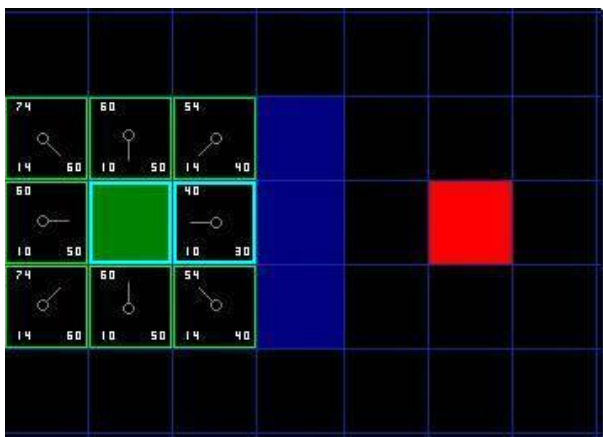


Fig. 4 Hasil langkah 4),5), dan 6). Border kotak berwarna biru menunjukkan simpul termasuk ke dalam closed set.

Ulangi langkah 4),5), dan 6) sampai simpul target termasuk ke dalam closed set, di mana simpul itu terlihat seperti ilustrasi di bawah ini.

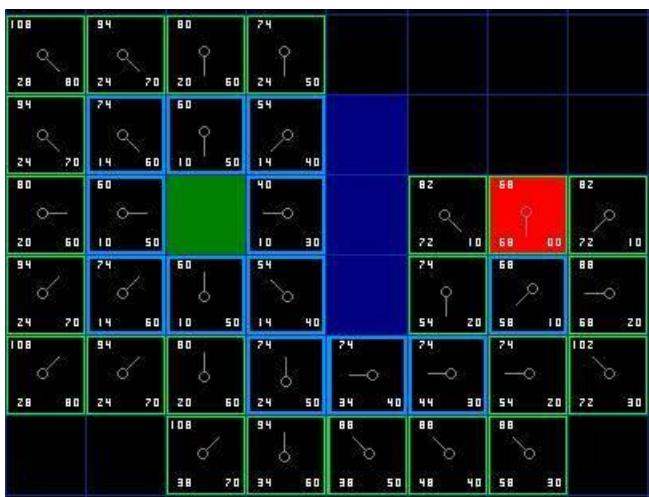


Fig. 5 Hasil langkah 4),5), dan 6) yang dilooing.

Untuk menentukan path solusi hanya dengan mulai di simpul target dan bekerja mundur bergerak dari satu

simpul ke came from-nya sampai akhirnya kembali ke simpul awal. Seperti terlihat seperti ilustrasi berikut.

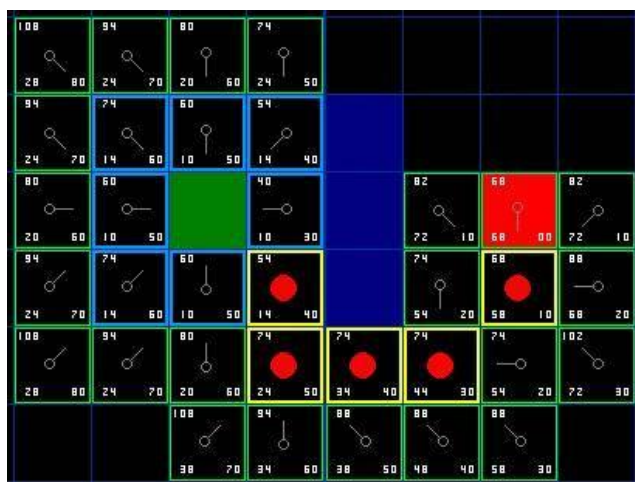


Fig. 6 Path solusi telah ditemukan (titik-titik merah)

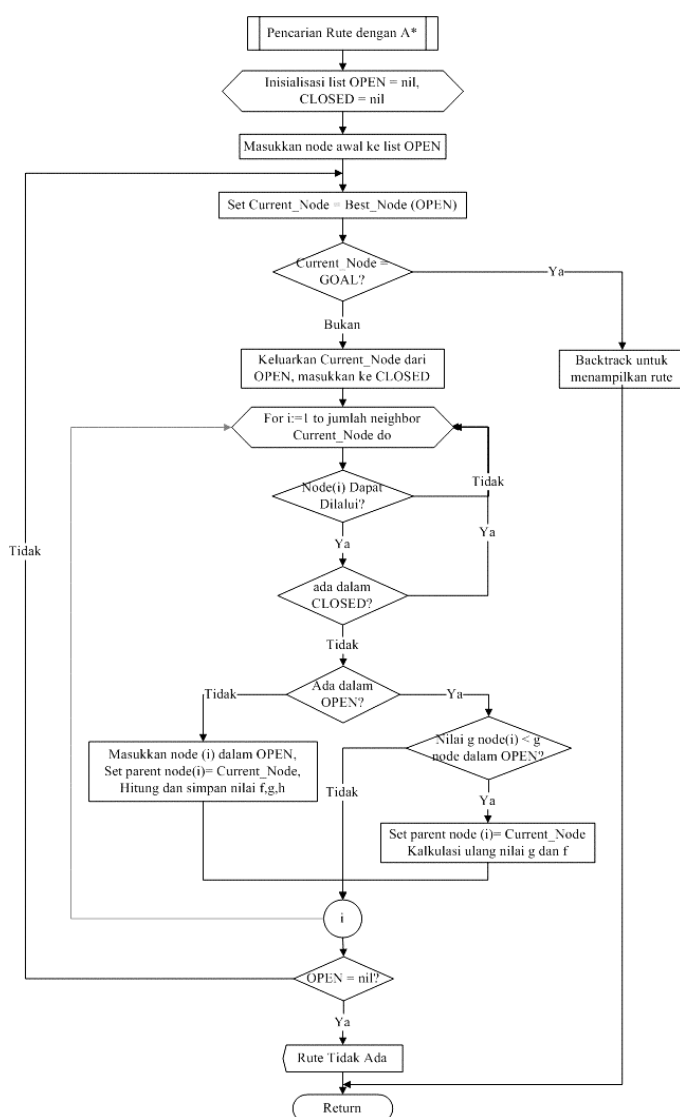


Fig. 7 Flowchart Algoritma A\*

## 5. KESIMPULAN

1. Algoritma A\* merupakan salah satu algoritma Branch & Bound yang selalu memberikan solusi yang optimal.
2. Algoritma A\* menggunakan informasi tambahan (nilai heuristik) dalam pencarian solusi. Oleh karena itu, solusi yang optimal sangat tergantung kepada fungsi heuristik yang diterapkan.
3. Maze atau labirin adalah sebuah puzzle dalam bentuk percabangan jalan yang kompleks dan memiliki banyak jalan buntu.
4. Algoritma A\* dapat diterapkan untuk menyelesaikan permasalahan maze yaitu dengan cara menyimpan data posisi tembok maze sebagai simpul yang tidak bisa dilewati (not walkable).

## REFERENSI

- [1] [http://en.wikipedia.org/wiki/A\\*\\_search\\_algorithm](http://en.wikipedia.org/wiki/A*_search_algorithm), diakses tanggal 7 Desember 2011.
- [2] <http://www.policyalmanac.org/games/aStarTutorial.htm>, diakses tanggal 7 Desember 2011.
- [3] <http://koleksigambarunik.blogspot.com/2010/12/rahasia-tentang-permainan-labirin.html>, diakses tanggal 7 Desember 2011.

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 7 Desember 2011

ttd

Hapsari Tilawah - 13509027