

ALGORITMA A-STAR (A*)

disusun untuk memenuhi tugas
mata kuliah algoritma dan pemrograman dua



disusun oleh:

Lalu Panji Ihza Mahendra

(17112402)

JURUSAN TEKNIK INFORMATIKA

**SEKOLAH TINGGI TEKNOLOGI STIKMA INTERNASIONAL
MALANG**

JL. SIMPANG SULFAT UTARA NO.20

2020

DAFTAR ISI

COVER	i
DAFTAR ISI	ii
KATA PENGANTAR	iii
BAB I PENDAHULUAN	4
1.1 Latar Belakang	4
1.2 Rumusan Masalah	5
1.3 Tujuan	5
BAB II ISI	
2.1 Pengenalan	5
2.2 Metode	6
2.3 Penerapan	6
2.4 Contoh Perhitungan	7
BAB III PENUTUP	9
7.1 Kesimpulan	9
DAFTAR PUSTAKA	9

KATA PENGANTAR

Segala puji dan syukur kita panjatkan kepada Allah SWT atas Rahmat-Nya yang selama ini kita dapatkan, yang memberi hikmah dan yang paling bermanfaat bagi seluruh umat manusia, oleh karenanya kami dapat menyelesaikan tugas akhir penyusunan makalah ini dengan baik dan tepat pada waktunya. Ada pula maksud atau tujuan dari penyusunan makalah ini ialah untuk memenuhi salah satu tugas yang diberikan oleh dosen pada mata kuliah algoritma dan pemrograman dua.

Dalam proses penyusunan tugas ini kami menjumpai berbagai hambatan, namun berkat dukungan materiil maupun non-materiil dari berbagai pihak, akhirnya kami dapat menyelesaikan tugas ini dengan cukup baik, maka pada kesempatan ini kami menyampaikan terima kasih dan penghargaan setinggi-tingginya kepada semua pihak terkait yang telah membantu terselesaikannya tugas ini.

Tugas ini masih jauh dari kesempurnaan, oleh karena itu kami mengharapkan segala saran dan kritik yang membangun dari semua pihak sangat kami harapkan demi perbaikan pada tugas selanjutnya. Harapan kami semoga tugas ini memberikan ilmu dan manfaat, khususnya bagi kami dan para pembaca sekalian.

BAB I

PENDAHULAN

1.1 Latar Belakang

Kecerdasan buatan atau *artificial intelligence* merupakan bagian dari ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Sistem cerdas (*intelligent system*) adalah sistem yang dibangun dengan menggunakan teknik-teknik *artificial intelligence*. Metode Pencarian dalam Kecerdasan Buatan – Berbagai algoritma untuk pencarian (*search algorithm*) yang ada berbeda satu dengan yang lain dalam hal pengembangan kumpulan *node* untuk mencapai *goal state*. Perbedaan ini terutama dalam hal cara dan urutan pengembangan *node*, dan sangat berpengaruh pada kinerja masing-masing algoritma.

Menurut Mahafi dan Hemawan (2013:20), terdapat empat kriteria yang menjadi ukuran algoritma pencarian, yaitu: (1) *completeness* – apakah algoritma pasti dapat menemukan solusi, (2) *time complexity* – berapa lama waktu yang dibutuhkan untuk menemukan sebuah solusi, (3) *space complexity* – berapa memori atau *resource* yang diperlukan untuk melakukan pencarian, dan (4) *optimality* – apakah algoritma tersebut dapat menemukan solusi yang terbaik jika terdapat beberapa solusi yang berbeda.

Terdapat banyak metode pencarian yang telah diusulkan. Semua metode yang ada dapat dibedakan ke dalam dua jenis pencarian (Suyanto, 2011:15) yaitu: (1) pencarian buta/tanpa informasi (*blind* atau *uninformed search*), dan (2) pencarian heuristik/dengan informasi (*heuristic* atau *informed search*). Setiap metode mempunyai karakteristik yang berbeda-beda dengan kelebihan dan kekurangannya masing-masing.

Salah satu dari metode dari pencarian heuristik adalah algoritma A* (A-Star), algoritma ini sering digunakan dalam aplikasi dan tentunya membantu dalam menyelesaikan masalah. Pada makalah ini akan membahas mengenai apa itu algoritma A*, penerapannya, dan contoh dari penggunaannya.

1.2 Rumusan Masalah

Rumusan masalah yang didapat dari latar belakang di atas adalah

1. Apa itu Algoritma A*?
2. Bagaimana cara kerja dari algoritma A*?
3. Seperti apa contoh penerapannya dalam kecerdasan buatan?

1.3 Tujuan

Mengenal lebih lanjut dari Algoritma A*, dan menyelesaikan tugas dari mata kuliah algoritma dan pemrograman dua.

BAB II

ISI

2.1 Pengenalan

Algoritma A*, dapat juga disebut sebagai Algoritma A Star, merupakan salah satu contoh algoritma pencarian yang cukup populer di dunia. Jika kita mengetikkan Algoritma A* pada sebuah mesin pencari, seperti google.com, maka akan kita temukan lebih dari sepuluh ribu literatur mengenai algoritma A*.

Beberapa terminologi dasar yang terdapat pada algoritma ini adalah *starting point*, simpul (*nodes*), A, *open list*, *closed list*, harga (*cost*), halangan (*unwalkable*).

Starting point adalah sebuah terminologi untuk posisi awal sebuah benda. A 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. *open list* adalah tempat menyimpan data simpul yang mungkin diakses dari *starting point* maupun simpul yang sedang dijalankan.

Closed list adalah tempat menyimpan data simpul sebelum A yang juga merupakan bagian dari jalur terpendek yang telah berhasil didapatkan. Harga (F) adalah nilai yang diperoleh dari penjumlahan nilai G, jumlah nilai tiap simpul dalam jalur terpendek dari *starting point* ke A, dan H, jumlah nilai perkiraan dari sebuah simpul ke simpul tujuan.

2.2 Metode

Prinsip algoritma ini adalah mencari jalur terpendek dari sebuah simpul awal (*starting point*) menuju simpul tujuan dengan memperhatikan harga (F) terkecil.

Diawali dengan menempatkan A pada *starting point*, kemudian memasukkan seluruh simpul yang bertetangga dan tidak memiliki atribut rintangan dengan A ke dalam *open list*. Kemudian mencari nilai H terkecil dari simpul-simpul dalam *open list* tersebut. Kemudian memindahkan A ke simpul yang memiliki nilai H terkecil. Simpul sebelum A disimpan sebagai *parent* dari A dan dimasukkan ke dalam *closed list*. Jika terdapat simpul lain yang bertetangga dengan A (yang sudah berpindah) namun belum termasuk kedalam anggota *open list*, maka masukkan simpul-simpul tersebut ke dalam *open list*. Setelah itu, bandingkan nilai G yang ada dengan nilai G sebelumnya (pada langkah awal, tidak perlu dilakukan perbandingan nilai G). Jika nilai G sebelumnya lebih kecil maka A kembali ke posisi awal. Simpul yang pernah dicoba dimasukkan ke dalam *closed list*. Hal tersebut dilakukan berulang-ulang hingga terdapat solusi atau tidak ada lagi simpul lain yang berada pada *open list*.

Fungsi heuristik untuk menghitung taksiran nilai yang diberikan pada simpul dinyatakan sebagai berikut:

$$f(n) = g(n) + h(n)$$

$$f(n) = \text{ongkos untuk simpul } n$$

$$h(n) = \text{ongkos mencapai simpul } n \text{ dari akar}$$

$$h(n) = \text{ongkos mencapai simpul tujuan dari simpul } n$$

Mencari nilai heuristik dapat dilakukan dengan beberapa metode, seperti Manhattan Distance, Diagonal Distance, dan Euclidean Distance

2.3 Penerapan

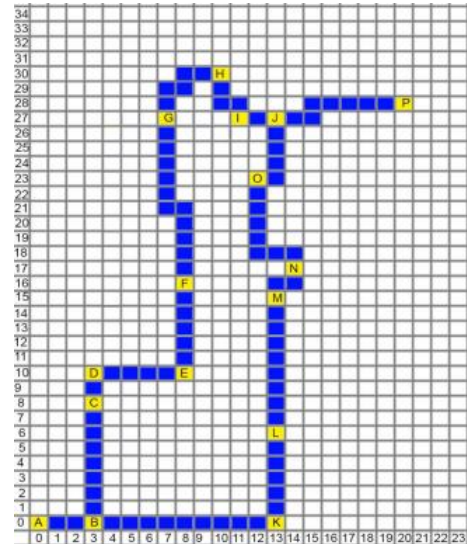
Algoritma A Star atau A* adalah salah satu algoritma pencarian yang menganalisis input, mengevaluasi sejumlah jalur yang mungkin dilewati dan menghasilkan solusi. Algoritma A* adalah algoritma komputer yang digunakan secara luas dalam *graph traversal* dan penemuan jalur serta

proses perencanaan jalur yang bisa dilewati secara efisien di sekitar titik-titik yang disebut node (Reddy, 2013).

Algoritma ini juga sering digunakan pada *game*, terutama *game* yang mencari jalur terbaik atau terdekat, semisal pemain memilih koordinat yang ia tuju, dengan beragam jalur yang ada, program akan memilih terbaik untuk dilewati.



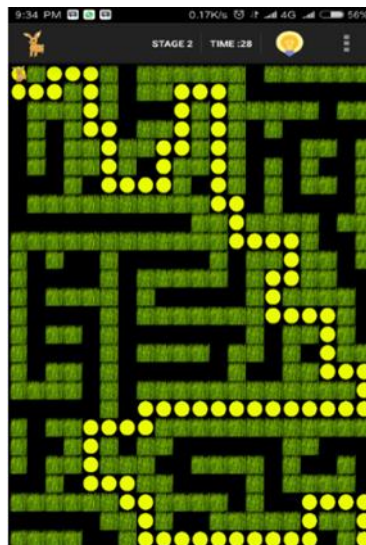
Gambar 2



Gambar 1

2.4 Contoh Perhitungan

Perhatikan gambar dari permainan labirin berikut



Gambar 3

Algoritma A* dapat menentukan jalur yang tepat untuk dilewati, bila Perhitungan algoritma A* menggunakan *euclidean heuristic* yang digunakan untuk mencari jalur terpendek antara dua buah simpul horizontal dan vertikal. Nilai $n.x$, $n.y = 0$ dan $goal.x$, $goal.y = 1$ yang diambil dari jarak simpul pada game. Nilai node (grid) yaitu 20×26 pixel dimana 1 grid = 1 *pixel* ke arah horizontal atau 1 *pixel* ke arah vertikal. Nilai $h(n)$ *heuristic* didapat dari jarak perkiraan dari simpul awal ke simpul tujuan. Nilai $f(n)$ didapat dari penjumlahan dari $g(n)$ dan $h(n)$.

$$\begin{aligned}
 \text{Posisi simpul awal} &= n.x: 0, n.y: 0 \\
 \text{Posisi simpul tujuan} &= \text{Goal}.x: 19, \text{Goal}.y: 25 \\
 g(0,1) &= 1 \\
 h(n) &= \sqrt{(n.x - \text{goal}.x)^2 + (n.y - \text{goal}.y)^2} \\
 h(0,1) &= \sqrt{(0 - 19)^2 + (1 - 25)^2} \\
 h(0,1) &= \sqrt{(-19)^2 + (-24)^2} \\
 h(0,1) &= \sqrt{361 + 576} = \sqrt{937} \\
 h(0,1) &= 30,61 \\
 f(n) &= g(n) + h(n) \\
 f(0,1) &= 1 + 30,61 = 31,61
 \end{aligned}$$

Setelah mendapat total nilai biaya (f) dari n , selanjutnya menghitung estimasi lagi dari simpul yang bersebelahan, dan seterusnya, nilai yang didapat lebih efektif akan diambil sebagai close list. Untuk mencari jalur terpendek dari stage 1 membutuhkan 156 langkah dengan simpul yang diperiksa yaitu 186 simpul, sehingga menghasilkan seluruh kordinat seperti di bawah ini,



Gambar 4

BAB III

PENUTUP

3.1 Kesimpulan

Penerapan algoritma A* sangat bermanfaat dalam berbagai aplikasi yang mengikuti trend saat ini, yang sangat mengandalkan kecerdasan buatan, terutama dalam pencarian jalur dari map maupun game, bila dibandingkan dengan metode pencarian yang lain, algoritma A* sedikit lebih unggul dari yang lain, karena kemampuannya akan mencari keefektifan jalur yang menyeluruh.

DAFTAR PUSTAKA

towardsdatascience.com (2019, 29 September). A-Star (A*) Search Algorithm. Diakses pada 13 Agustus 2020, dari <https://towardsdatascience.com/a-star-a-search-algorithm-eb495fb156bb>

geeksforgeeks (2018, 7 September). A* Search Algorithm, Diakses pada 13 Agustus 2020, dari <https://www.geeksforgeeks.org/a-search-algorithm/>

fatkhan.web.id (2019, 1 Mei). Metode Pencarian dalam Kecerdasan Buatan. Diakses pada 13 Agustus 2020, dari <http://fatkhan.web.id/metode-pencarian-dalam-kecerdasan-buatan/>

Setiawan K, Supriyadin, Santoso I, Buana R. MENGHITUNG RUTE TERPENDEK MENGGUNAKAN ALGORITMA A* DENGAN FUNGSI EUCLIDEAN DISTANCE. *Seminar Nasional Teknologi Informasi dan Komunikasi 2018 (SENTIKA 2018)*. 2089-9815

Ahmad Imam & Widodo Wahyu. Penerapan Algoritma A Star (A*) pada Game Petualangan Labirin Berbasis Android. *Jurnal Ilmu Komputer dan Informatika*, 3, 2477-698X.