

Penerapan Algoritma A* (A-Star) untuk Mencari Jalur Terdekat (Shortest Path) dalam Kecerdasan Buatan (Studi Kasus: Game Snake)

Reynaldo Paulini Wonges, Sigit birowo*, Liaw Bun Fa

Program Studi Sistem Informasi, Institut Bisnis dan Informatika Kwik Kian Gie, Jl. Yos Sudarso Kav 87, Sunter Jakarta 14350, Indonesia.

*) Surel korespondensi : sigit.birowo@kwikkiangie.ac.id

Abstrak. *Game* sebagai media hiburan telah berkembang secara pesat seiring dengan teknologi. Salah satu unsur yang berperan penting dalam sebuah *game* adalah kecerdasan buatan. Dengan menggunakan kecerdasan buatan, diharapkan permainan ular ini menjadi terobosan baru dalam menerapkan algoritma untuk menentukan jalur terpendek pada ular. Dengan menggunakan algoritma ini dimungkinkan akan menjadi suatu terobosan yang baru untuk permainan ini dalam menentukan jalur ular terpendek dalam mencari makanannya. Metode yang peneliti gunakan adalah dengan pengujian dan pengukuran dari hasil perhitungan biaya melalui penggunaan Algoritma A* untuk menghitung jalur terpendek pada permainan ular. Dalam sistem yang dibuat, peneliti memanfaatkan Algoritma A* untuk memproses dan menghitung biaya terendah yang didapatkan pada saat ular berjalan untuk menuju makanannya. Kesimpulan yang dapat diambil oleh peneliti adalah dengan menggunakan algoritma A* ini pengujian terbilang cukup efektif untuk mencari jalur terpendek dan efisien serta sistem kecerdasan buatan yang terbaru sehingga permainan ini bisa dijadikan acuan untuk dijadikan penelitian selanjutnya karena dapat dimungkinkan untuk dilakukan pengujian-pengujian atau penelitian yang lain seperti mengkomparasi dengan algoritma yang lain untuk menentukan yang mana yang paling bagus.

Kata kunci: Algoritma A*, kecerdasan buatan, permainan ular



This work is licensed under Attribution-NonCommercial-ShareAlike 4.0 International. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Diterbitkan oleh LPPM Institut Bisnis dan Informatika Kwik Kian Gie, Jl. Yos Sudarso Kav 87, Sunter Jakarta 14350, Indonesia.

1. Pendahuluan

Dalam era globalisasi dan teknologi sekarang ini penggunaan komputer sebagai salah satu alat teknologi informasi sangat dibutuhkan keberadaannya hampir disetiap aspek kehidupan. Penggunaan perangkat komputer sebagai perangkat pendukung manajemen dan pengolahan data adalah sangat tepat dengan mempertimbangkan kuantitas dan kualitas data, dengan demikian penggunaan perangkat komputer dalam setiap informasi sangat mendukung sistem pengambilan keputusan, tetapi di era globalisasi sekarang ini komputer bukan digunakan hanya untuk itu saja, tetapi juga digunakan untuk hiburan seperti contohnya bermain game, kebutuhan akan jasa, perdagangan, komunikasi dan lainnya.

Game merupakan salah satu produk teknologi informasi yang cukup digemari saat ini, *game* juga merupakan bentuk aplikasi yang edukatif, artinya bisa dijadikan sebagai media pembelajaran dimana prosesnya bisa dilakukan dengan konsep belajar

sambil bermain. Game telah mengalami kemajuan yang sangat pesat. Ini bisa kita lihat di kota-kota besar, tidak terkecuali juga kota-kota kecil, banyak sekali game center yang muncul. *Game* saat ini tidak seperti game terdahulu, jika dahulu game hanya bisa maksimal dimainkan dua orang, sekarang dengan kemajuan teknologi terutama jaringan internet, game bisa dimainkan 100 orang lebih sekaligus dalam waktu yang bersamaan. Walaupun game ditujukan untuk anak-anak, tidak sedikit pula orang dewasa kerap memainkannya bahkan tidak sedikit yang menjadikannya sebagai pekerjaan dan mendapat penghasilan dari bermain game. Perkembangan game yang terus berkembang diantaranya permainan games ular.

Games ular merupakan permainan populer dalam telepon selular beberapa tahun yang lalu. Inti dari permainan ini adalah agar ular yang kita kontrol mendapatkan sebanyak-banyaknya makanan tanpa membentur dinding atau bagian tubuhnya sendiri. Semakin banyak makanan yang ular dapatkan, tubuhnya akan tumbuh sehingga semakin panjang.

Sulitnya sistem *Artificial Intelligence* (AI) ular untuk menentukan rute mana yang dapat dilalui agar tidak memakan tubuhnya sendiri juga merupakan salah satu kendala pada permainan ini. *Game Snake* konvensional saat ini belum terlalu banyak memiliki AI dalam implementasinya, baik AI yang ada pada karakter ular itu sendiri maupun AI yang ada pada umpan. AI yang ada pada game snake konvensional akan dikembangkan sehingga diharapkan menjadi tantangan tersendiri bagi pengguna. Selanjutnya permainan ular ini belum memiliki sistem kecerdasan buatan yang dapat menentukan jalur mana yang harus dilewati secara efisien. Dan jalur yang dilewati belum tentu yang paling terpendek untuk memakan makanan si-ular. Diharapkan dengan penelitian ini dapat disimpulkan metode algoritma mana yang paling efisien dalam mencari rute terpendek pada permainan snake, adapun sistem kecerdasan buatan yang menggunakan beberapa algoritma yang mampu mempersingkat waktu pencarian rute atau jalan terpendek pada permainan ini, namun ada beberapa yang masih perlu di coba atau di buktikan lebih lanjut.

Berdasarkan latar belakang tersebut maka identifikasi masalah adalah sebagai berikut: (1) belum adanya sistem kecerdasan buatan yang dapat menentukan jalur yang paling efisien; (2) belum adanya sistem kecerdasan buatan yang terbaru; dan (3) sulitnya sistem navigasi untuk menentukan jalur mana yang harus dilewati ular dalam mencari makanannya

Batasan masalah: (1) belum adanya sistem kecerdasan buatan yang dapat menentukan jalur yang paling efisien; dan (2) belum adanya sistem kecerdasan buatan yang terbaru

Tujuan dari penelitian ini adalah untuk membuat kecerdasan buatan dengan menggunakan Algoritma A-star dalam permainan *snake*. Adapun manfaat dari penelitian ini adalah (1) dapat mengetahui bagaimana kecerdasan buatan itu mempengaruhi sebuah permainan *Snake* dalam mencari jalur terpendek; (2)

menambah pengetahuan dan menjadi acuan untuk penelitian selanjutnya; serta (3) dapat menambah pengetahuan mengenai *Algoritma A*(A-star)* untuk diterapkan ke dalam aplikasi yang lain.

2. Landasan Teori

2.1 Uniform Cost Search

Menurut Suyanto(2014:19), "Uniform Cost Search (UCS) menggunakan urutan biaya dari yang paling kecil sampai yang terbesar. UCS berusaha menemukan solusi dengan total biaya terendah yang dihitung berdasarkan biaya dari simpul asal menuju simpul tujuan. Biaya dari simpul asal ke suatu simpul n dilambangkan sebagai $g(n)$."

2.2 Greedy Best-First Search

Menurut Suyanto (2014:30), " algoritma ini merupakan salah satu jenis algoritma Best First Search yang paling sederhana dengan hanya memperhitungkan biaya perkiraan (estimated cost) saja, yakni $f(n) = h(n)$. biaya yang sebenarnya (*actual cost*) tidak diperhitungkan. Dengan hanya memperhitungkan biaya perkiraan yang belum tentu kebenarannya, maka algoritma ini menjadi tidak optimal."

2.3 Algoritma A-star

Menurut Suyanto(2014:33), " Algoritma A* merupakan algoritma *Best First Search* yang menggabungkan *Uniform Cost Search* dan *Greedy Best-First Search*. Biaya yang diperhitungkan didapat dari biaya sebenarnya ditambah dengan biaya perkiraan. Dalam notasi matematika dituliskan sebagai : $f(n) = g(n) + h(n)$. dengan perhitungan biaya seperti ini, algoritma A* adalah complete dan optimal."

2.4 Fungsi Heuristik

Menurut Suyanto (2014:65), " Di dalam metode-metode yang termasuk *heuristic search*, fungsi heuristik memainkan peranan yang sangat menentukan. Suatu Fungsi dapat diterima sebagai fungsi heuristik jika biaya perkiraan yang dihasilkan tidak melebihi dari biaya sebenarnya. Ketika fungsi heuristik memberikan biaya perkiraan yang melebihi biaya sebenarnya (*overestimate*), maka proses pencarian bisa tersesat dan membuat *heuristic search* menjadi tidak optimal."

"Suatu fungsi heuristik dikatakan baik jika bisa memberikan biaya perkiraan yang mendekati biaya sebenarnya. Semakin mendekati biaya sebenarnya, fungsi heuristik tersebut semakin baik." Suyanto (2014:65)

2.5 JavaScript

Menurut Haverbeke (2014), "Javascript diperkenalkan pada tahun 1995 di *browser navigator* Netscape untuk menambahkan program ke halaman web khususnya di halaman HTML (*hypertext markup language*). Bahasa pemrograman ini telah diadopsi oleh berbagai web browser ternama seperti *Mozilla Firefox*, *Google Chrome*, *Safari*, dan lainnya. *Javascript* juga digunakan dalam situs web yang tradisional untuk memberikan berbagai macam bentuk interaktivitas dan kepandaian. *Javascript* tidak

memiliki hubungan sama sekali dengan bahasa pemrograman Java. Nama yang mirip itu dikarenakan terinspirasi dari pertimbangan marketing daripada penilaian yang baik. Ketika Javascript diperkenalkan, bahasa pemrograman Java sudah lama terkenal dan sudah sangat dipasarkan. Javascript dijalankan pada sistem penerjemah (*intepreter*) yang ada pada *web browser*."

2.6 Kecerdasan Buatan

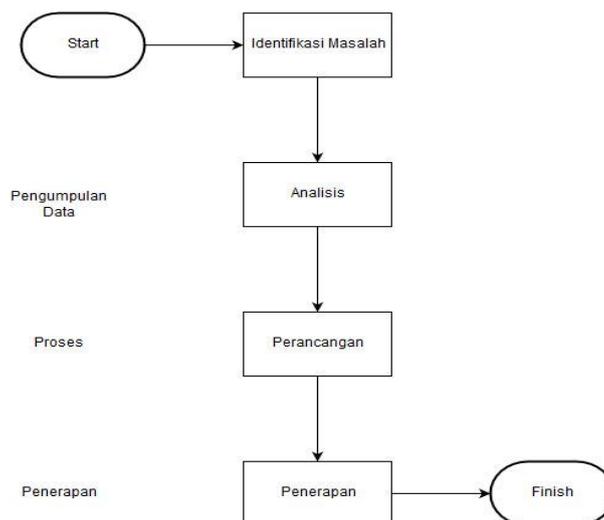
Menurut Suyanto(2014:3), "Artificial Intelligence didefinisikan secara berbeda-beda oleh para ahli, ada yang fokus pada logika berpikir manusia saja, tetapi ada juga yang mendefinisikan AI secara lebih luas pada tingkah laku manusia." Dikutip pada buku karangan Stuart Russel dan Peter Norvig yang berjudul "Artificial Intelligence: A Modern Approach."

3. Metode

3.1 Gambaran Umum Obyek Penelitian

Penelitian ini menggunakan cara menjalankan secara otomatis dan tanpa kontrol dari pengguna melainkan dengan menggunakan algoritma untuk menganalisa langkah langkah untuk dapat menentukan jalur mana yang memiliki jalur terpendek dengan menghitung biaya pada setiap langkah dan menentukan yang paling terbaik.

Game Snake biasanya dimainkan dengan menggunakan media seperti telepon pintar(*smartphone*), telepon genggam, dan mesin *video game* klasik. Cara mainnya dengan menekan tombol kiri, kanan, atas, dan bawah untuk mengontrol jalannya ular mencari makanannya yang muncul secara acak. Seperti contoh jika menekan tombol kiri maka ular akan berbelok ke kiri, dan jika menekan tombol kanan maka ular akan ber belok ke kanan.

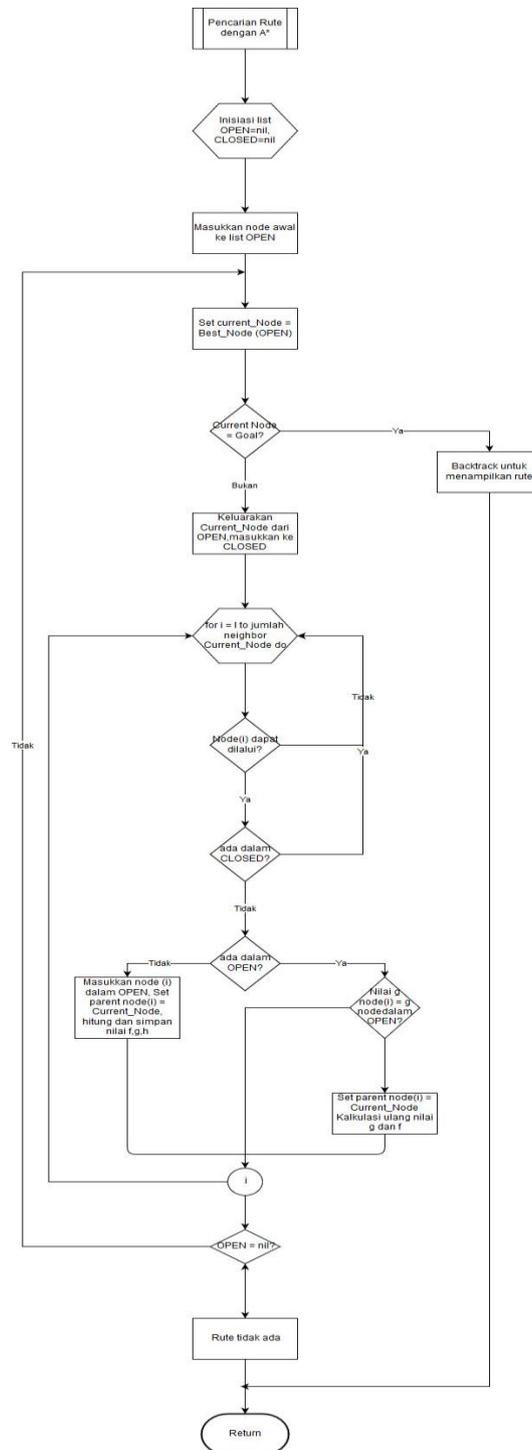


Gambar 1 Kerangka Pemikiran. Sumber: Olahan Penulis

Permainan ini mempunyai satu misi yaitu dengan memakan semua makanan yang muncul dan bertumbuh semakin panjang serta menghindari memakan tubuhnya sendiri. Permainan ini terkenal sejak telepon genggam yang ada pada saat itu.

Pada gambar 1 penjelasan kerangka pemikiran yang dirancang penulis secara singkat yaitu pertama, mengidentifikasi masalah terlebih dahulu, kedua, pada saat pengumpulan data dengan cara menganalisa algoritma yang akan digunakan, ketiga, pada saat proses yaitu merancang algoritma yang diusulkan, keempat yaitu melakukan implementasi dan testing yang didapatkan dari hasil perancangan

Adapun alur program yang dibuat secara singkat sebagai berikut:

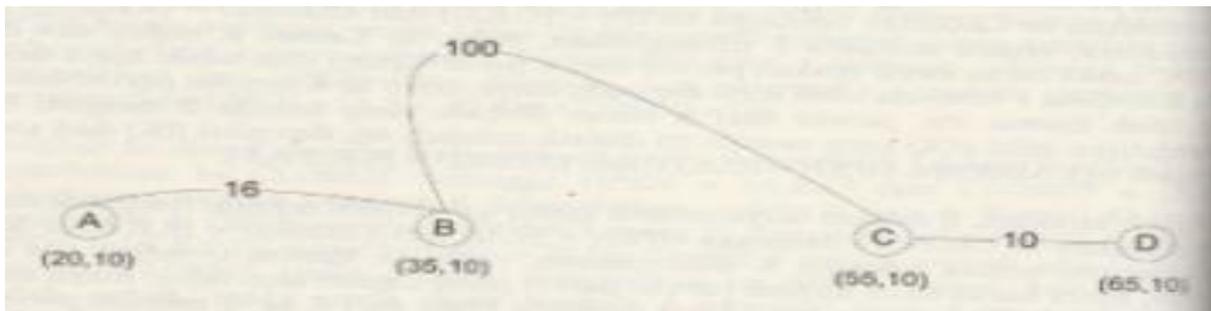


Gambar 2 Flowchart Algoritma A*(A-Star). Sumber: Hapsari Tilawah (Jurnal Penerapan Algoritma A* pada Maze)

3.2 Teknik Pengukuran Data

Data diukur dengan melakukan tes atau diuji terlebih dahulu menggunakan algoritma A-Star. Dengan adanya algoritma ini, teknik pengukuran data yang digunakan sebagai usulan pemecahan masalah Dengan menggunakan fungsi heuristic

$F(n) = g(n) + h(n)$ dimana $g(n)$ adalah jarak total dari posisi asal ke lokasi sekarang; dan $h(n)$ merupakan fungsi heuristik yang digunakan untuk memperkirakan jarak dari lokasi sekarang ke lokasi tujuan. Semakin tinggi keakuratan heuristik, semakin cepat dan bagus lokasi tujuan ditemukan dan dengan tingkat keakuratan yang lebih baik.



Gambar 3 Contoh Fungsi heuristik untuk pencarian rute terpendek. Sumber: Suyanto(2014:66)

Dalam kasus pencarian rute terpendek, biaya sebenarnya adalah panjang jalan raya yang sebenarnya. Sedangkan fungsi heuristik yang bisa digunakan adalah jarak garis lurus yang bisa di hitung menggunakan rumus:

$$d_{ab} = \sqrt{(y_b - y_a)^2} + \sqrt{(x_b - x_a)^2} \quad (2.5)$$

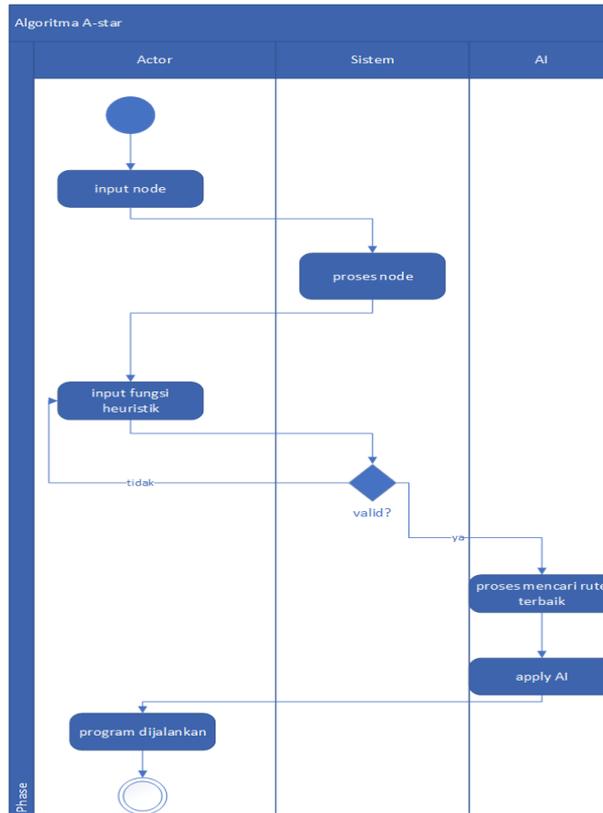
Dengan menggunakan rumus tersebut, maka jarak garis lurus dari A ke B adalah 15 ($d_{AB} = 15$) dan jarak garis lurus dari B ke C adalah 20 ($d_{BC} = 20$). Jika dibandingkan dengan panjang jalan yang sebenarnya, d_{AB} sangat mendekati jarak sebenarnya, sedangkan d_{BC} jauh lebih kecil (*underestimate*) dibandingkan dengan jarak sebenarnya. Tetapi, d_{CD} sama dengan jarak sebenarnya. Rumus pada persamaan 2.5 diatas menjamin bahwa jarak perkiraan yang dihasilkan pasti lebih kecil atau sama dengan jarak sebenarnya. Sehingga persamaan tersebut dapat digunakan sebagai fungsi heuristik.

4. Hasil dan Pembahasan

4.1 Rancangan Sistem

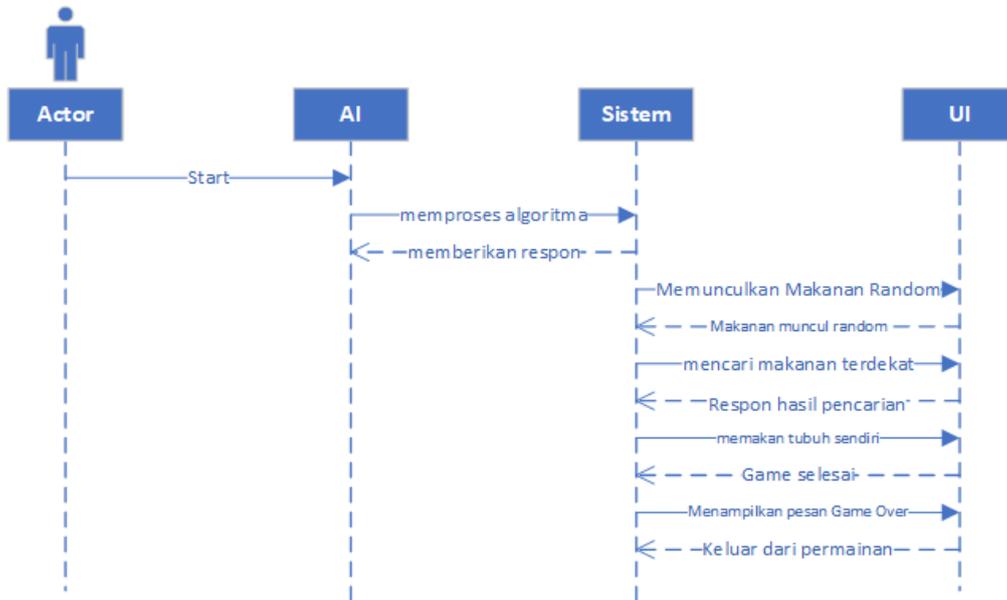
Dalam penelitian ini, penulis akan menerapkan algoritma A-Star pada permainan ular yang diterapkan dan menggunakan bahasa pemograman *JavaScript* untuk menjalankan aplikasi yang dibuat.

Adapun *Activity Diagram* yang dirancang oleh penulis sebagai berikut:



Gambar 4 Activity Diagram. Sumber: Olahan Penulis

Adapun Sequence diagram yang dirancang oleh penulis sebagai berikut :

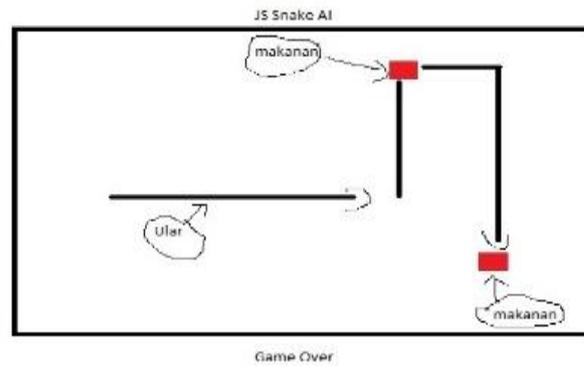


Gambar 5 Sequence Diagram. Sumber: Olahan Penulis

Pada gambar 5 di jelaskan interaksi dan respon antar masing masing actor/user dengan mesin dan memunculkan hasil keluaran dari aksi user dalam mengoperasikan atau berinteraksi dengan mesin.

4.2 Rancangan Antar Muka

Berikut ini adalah rancangan antar muka dari aplikasi yang penulis buat.



Gambar 6 Rancangan Tampilan Game Ular. Sumber: Olahan Penulis

4.3 Rancangan Alur Program (Pseudocode Algoritma A-Star)

```
function A*(masalah) return solusi
OPEN <- S
CLOSED <- array kosong
loop sampai goal ditemukan atau sampai tidak ada simpul didalam OPEN
  if Open = kosong then
    Gagal
  else
    BestNode = Simpul yang ada di OPEN dengan f minimal
    Pindahkan simpul terbaik tersebut dari OPEN ke CLOSED
    If BestNode = goal then
      Sukses
    else
      Bangkitkan semua suksesor BestNode tapi jangan buat pointer
      Untuk setiap suksesor kerjakan:
        Hitung  $g(\text{suksesor}) = g(\text{BestNode}) + \text{actual cost}(\text{dari BestNode ke suksesor})$ 
        {Periksa suksesor}
      if suksesor ada di OPEN then {sudah pernah dibangkitkan dan sudah diproses}
        OLD = simpul di OPEN yang sama dengan suksesor tersebut
        Tambahkan OLD sebagai suksesor BestNode
        Buat pointer dari OLD ke BestNode
        Bandingkan nolai  $g(\text{OLD})$  dengan  $g(\text{suksesor})$ 
      If  $g(\text{OLD})$  lebih baik then
        Ubah parent OLD ke BestNode
        Ubah nilai g dan f yang ada pada OLD
    end
  else
    if suksesor ada di CLOSED then {sudah pernah dibangkitkan dan sudah diproses}
      OLD = simpul di CLOSED yang sama dengan suksesor tersebut
      Tambahkan OLD sebagai suksesor BestNode
      Bandingkan nilai  $g(\text{OLD})$  dengan  $g(\text{suksesor})$ 
      if  $g(\text{OLD})$  lebih baik then
        Ubah parent OLD ke BestNode
        Ubah nilai g dan f yang ada pada OLD
```

Propagasi untuk semua suksesor OLD dengan penelusuran DFS dengan aturan:

```

loop sampai simpul suksesor tidak ada di OPEN atau simpul tidak punya suksesor
if suksesor ada di OPEN then
  Propagasi diteruskan
else
  if nilai g via suksesor lebih baik then
    Propagasi diteruskan
  else
    Propagasi dihentikan
  end
end
end
else {suksesor tidak ada di OPEN maupun CLOSED}
  masukkan suksesor ke OPEN
  tambahkan suksesor tersebut sebagai suksesornya BestNode
  hitung  $f=g(\text{suksesor}) + h(\text{suksesor})$ 
end
end
end
end

```

Penggunaan algoritma dilakukan pada proses penghitungan simpul terbaik di dalam potongan pseudocode dibawah ini:

```

function A_Star() {
  // ending values
  var end_x = item_x;
  var end_y = item_y;
  // set of nodes that have already been looked at
  var closedSet = [];
  // set of nodes that are known but not looked at
  var openSet = [];
  // add the starting element to the open set
  openSet.push(grid_aStar[start_y][start_x]);
  grid_aStar[start_y][start_x].gScore = 0;
  grid_aStar[start_y][start_x].fScore = grid_aStar[start_y][start_x].heuristicCalc(end_x, end_y); // just
  the heuristic

  // while open set is not empty
  while (openSet.length > 0) {
    openSet.sort(fScoreSort);
    var currentNode = openSet[0];
    if ((currentNode.x == end_x) && (currentNode.y == end_y)) {
      return reconstruct_path(grid_aStar, currentNode, start_x, start_y); // return path
    }
    // remove current node from open set
    var index = openSet.indexOf(currentNode);
    openSet.splice(index, 1);
    closedSet.push(currentNode);
    // looking at all of the node's neighbours
    for (var i = -1; i < 2; i++) {
      for (var j = -1; j < 2; j++) {

```

```

if (!inBoundsCheck(currentNode, i, j)) {
  continue;
}
var neighbour = grid_aStar[currentNode.y + i][currentNode.x + j];
// if node is within the closed set, it has already
// been looked at - therefore skip it
if (closedSet.indexOf(neighbour) !== -1) {
  continue;
}

// set tentative score to gScore plus distance from current to neighbour
// in this case, the weight is equal to 1 everywhere
var tScore = neighbour.gScore + 1;

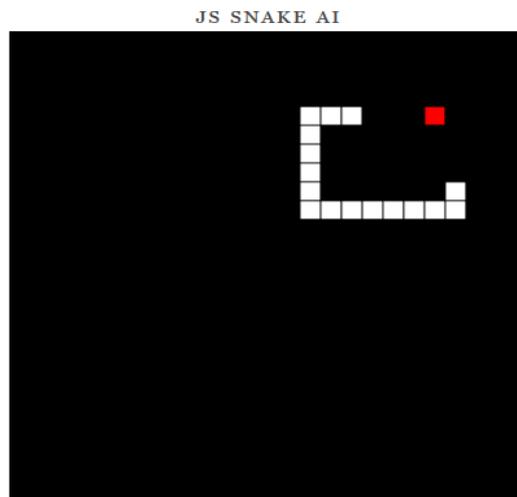
// if neighbour is not in open set, add it
if (openSet.indexOf(neighbour) === -1) {
  openSet.push(neighbour);
}
// this is a better path so set node's new values
neighbour.parent = currentNode;
neighbour.gScore = tScore;
neighbour.fScore = neighbour.gScore + neighbour.heuristicCalc(end_x, end_y);
}
}
}
// the node was not found or could not be reached
return false;

```

4.4 Implementasi Sistem

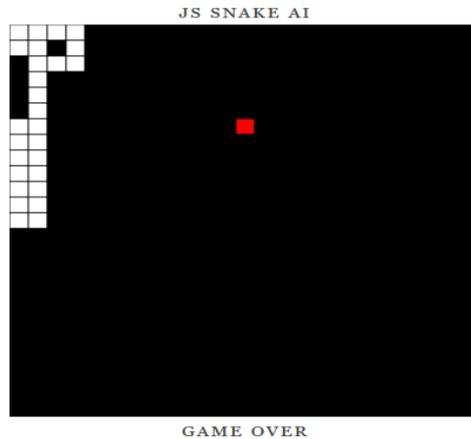
Untuk dapat menjalankan aplikasi ini, diperlukan instalasi web browser seperti *Google Chrome*, *Mozilla Firefox* dan sebagainya. Aplikasi akan berjalan secara otomatis dan akan berhenti secara otomatis setelah selesai

Tampilan awal ketika membuka aplikasi:



Gambar 7 Tampilan Game Ular. Sumber: Olahan Penulis

Tampilan akhir dari *game snake*



Gambar 8 Tampilan Game Over. Sumber: Olahan Penulis

4.5 Hasil Implementasi

Hasil penerapan Algoritma A* dalam pencarian rute terdekat dengan menggunakan rute terdekat aplikasi dapat di jalankan secara otomatis dan dapat mengasumsi titik merah sebagai makanan ular dan titik hijau sebagai kepala ular. Pengujian diuji dengan menggunakan contoh dari gambar sebagai berikut:



Gambar 9 Pengujian Algoritma A*. Sumber: Olahan Penulis

Jika masing-masing $h(n)$ dari setiap simpul sama dengan 5, maka dari hasil gambar 9 tersebut dapat diambil hasil pengujian dengan menggunakan rumus pada tabel sebagai berikut:

Tabel 1 Tabel hasil penerapan Algoritma A*

No	G(n)	G(n ke n)	H(n)	F(n) = g(n) + g(n ke n)+ h(n)
1	G(S)= 0	G(S ke A) = 5	5	10
2	G(S)=0	G(S ke B) = 5	5	10
3	G(A) = 5	G(A ke G) = 5	5	15
4	G(B) = 5	G(B ke G) = 10	5	20

Sumber: Olahan Penulis

Dari tabel tersebut dapat disimpulkan bahwa karena Algoritma A* menentukan rute dengan menghitung biaya terkecil maka rute ke G melewati A lebih pendek daripada melewati rute B karena hasil perhitungan membuktikan rute $A < B$. jadi, rute terpendek yang dapat dilalui adalah A.

Program ini memiliki kelebihan yaitu: (1) program ini dapat berjalan pada komputer dengan spesifikasi minimum, dapat berjalan pada sistem operasi windows apa saja tanpa ada kendala; (2) program ini tidak membutuhkan koneksi internet, menjadikan program *offline*.

Sementara kekurangannya adalah (1) program ini tidak memiliki banyak fitur yang terlalu bagus; dan (2) program ini memiliki user interface yang biasa saja.

5. Kesimpulan dan Saran

Berdasarkan penelitian yang telah penulis lakukan dan setelah aplikasi yang dibuat sudah selesai dan melalui proses testing dan evaluasi, maka penulis mengambil kesimpulan: (1) program ini dapat menentukan jalur terpendek yang ditempuh dalam sistem kecerdasan buatan dan secara efisien; dan (2) program ini dapat digunakan sebagai system kecerdasan yang terbaru karena dilihat dari permainan sebelumnya yang belum menggunakan algoritma ini.

Saran yang dapat diberikan untuk pengembangan di kemudian hari berdasarkan hasil penelitian ini adalah: (1) mengkomparasi algoritma ini dengan algoritma sejenis yang lain untuk menentukan yang mana yang paling efisien; dan (2) penerapan algoritma bisa di implementasikan ke berbagai macam pencarian rute terdekat yang lain.

Daftar Pustaka

- Andreas Hersiandi et.al (2016), Jurnal: *Penerapan Algoritma A* pada Permainan Snake*, STIMIK GI MDP; Jl. Rajawali No.14 Palembang.
- Game (2018), Wikipedia sumber: <https://en.wikipedia.org/wiki/Game> (diakses 4 Juni 2018)
- Hapsari Tilawah (2010). *Jurnal: Penerapan Algoritma A*(A-Star) untuk Menyelesaikan Masalah Maze*, Institut Teknologi Bandung Makalah IF3051 Strategi Algoritma
- Haverbeke, Marijn. (2014) *Eloquent Javascript 2nd Edition*, USA: No Starch Press
- Marakas dan O'Brien. (2017) *Introduction to Information Systems, Buku 1, Edisi 16*. Jakarta: Salemba Empat
- Marakas dan O'Brien. (2017) *Introduction to Information Systems, Buku 2, Edisi 16*. Jakarta: Salemba Empat
- Munir, Rinaldi. (2012) *Matematika Diskrit*, Revisi ke-5, Bandung: Informatika
- Pathfinder Github(2016), sumber: <https://github.com/kevinwang1975/PathFinder> (diakses 11 Juni 2018)
- Rengga Dionata Putra et.al (2012). *Jurnal: Labirin (Pencarian Rute Terdekat pada Labirin Menggunakan Algoritma A*)*, Jurnal EECCIS Vol 6, No. 2

- Snake A-star Github*(2017), sumber: <https://github.com/sacert/Snake-Star> (diakses 11 Juni 2018)
- Sugiyono. (2017). *Metode Penelitian & Pengembangan Research dan Development / R&D, Catatan ke-3*. Bandung: Alfabeta.
- Suyanto. (2014), *Artificial Intelligence (Searching, Reasoning, Planning, Learning)*, Revisi ke-2, Bandung: Alfabeta
- Yenie Sukriyah et.al (2016), Jurnal: *Penerapan Algoritma A* Untuk Mencari Rute Tercepat dengan Hambatan*, Seminar Nasional Telekomunikasi dan Informatika (SELISIK 2016)