

Penyelesaian Masalah 8-Puzzle dengan Algoritma Steepest-Ascent Hill Climbing

David Abraham¹, Indra W. Permana², Rangga Adi Nugraha³, Moch. Alvian⁴, Hanif⁵
Jurusan Teknik Elektro, Universitas Sultan Ageng Tirtayasa Cilegon, Indonesia

¹daavidabraham@yahoo.com, ²ns1fans@gmail.com, ³ragota_23@yahoo.co.id,

⁴mochammadalvian@gmail.com, ⁵albaproperti@gmail.com

Abstrak – 8 puzzle merupakan salah satu implementasi dari Artificial Intelligence. Dalam proses penyelesaiannya banyak terdapat algoritma-algoritma pencarian yang dapat diterapkan. Solusi 8 puzzle akan lebih cepat diperoleh jika digunakan prinsip array dengan variasi algoritma Steepest-Ascent Hill Climbing (Hill Climbing dengan memilih kemiringan yang paling tajam / curam) dengan parameter heuristik posisi yang benar dan heuristik jarak serta dikombinasikan dengan LogList sebagai penyimpanan state state yang pernah dilalui untuk menanggulangi permasalahan pada algoritma hill climbing itu sendiri dan terhindar dari looping state yang pernah dilalui. Metode-metode yang termasuk ke dalam teknik pencarian yang berdasarkan pada fungsi heuristik salah satu diantaranya adalah Hill Climbing, Best First Search, A* (A Bintang). Loglist merupakan tempat penyimpanan setiap kunjungan dari state-state puzzle yang telah dilakukan untuk menghindari looping atau pengulangan terhadap state yang pernah dilalui. Untuk menanggulangi permasalahan pada SteepestAscent Hill Climbing.

Kata kunci: 8Puzzle, heuristik, hill climbing, Steepest-Ascent Hill Climbing , loglist

Abstract – 8 puzzle is one of Artificial Intelligence implementation. In the process of its completion, there are search algorithms can be applied. 8 Puzzle solution will be obtained more quickly if used array principle with a variety of algorithms Steepest Ascent Hill-Climbing (Hill Climbing by selecting the sharpest tilt / steep) with parameters of position heuristic correct and distance heuristic combined with LogList as storage of states is ever passed for tackling problems at hill climbing algorithm itself and avoid looping state that once passed. The methods included in search techniques based on heuristic functions, one of them is the Hill Climbing, Best First Search, A * (A-star). Loglist is a storage place of every visit from states of puzzle that has been done to avoid looping or repetition to state that once passed. To cope with the problem on SteepestAscent Hill Climbing.

Keywords: 8Puzzle, heuristik, hill climbing, Steepest-Ascent Hill Climbing , loglist.

I. PENDAHULUAN

1.1 Konsep AI

AI (*Artificial Intelligence*) atau Kecerdasan Buatan merupakan salah satu cabang ilmu komputer yang mempelajari bagaimana cara membuat sebuah mesin cerdas, yaitu mesin yang mempunyai kemampuan untuk belajar dan beradaptasi terhadap sesuatu.

Jika diartikan tiap kata, *artificial* artinya buatan, sedangkan *intelligence* adalah kata sifat yang berarti cerdas. Jadi *artificial intelligence* maksudnya adalah sesuatu buatan atau suatu tiruan yang cerdas. Cerdas di sini kemungkinan maksudnya adalah kepandaian atau ketajaman dalam berpikir, seperti halnya otak manusia dalam menyelesaikan suatu masalah.

Tujuan dari riset-riset *Artificial Intelligence* (AI) / Kecerdasan Buatan adalah bagaimana membuat sebuah mesin bisa berfikir sama halnya dengan manusia yang bisa berfikir. AI digunakan untuk menjawab *problem* yang tidak dapat diprediksi dan tidak bersifat algoritmik atau prosedural. Sampai saat ini, para peneliti di bidang AI masih banyak menyimpan pekerjaan rumah mereka disebabkan kompleksitas penelitian di bidang *Artificial Intelligence* (AI) /

Kecerdasan Buatan serta faktor dukungan teknologi untuk merealisasikannya. Karena area cakupan yang luas, *Artificial Intelligence* (AI) / Kecerdasan Buatan dibagi lagi menjadi subsub bagian di mana sub-sub bagian tersebut dapat berdiri sendiri dan juga dapat saling melengkapi satu dengan lainnya.

AI membuat mesin agar dapat berpikir dan bertindak seperti manusia serta berpikir dan bertindak rasional.

1. *Acting Humanly*

Acting humanly ialah sistem yang melakukan pendekatan dengan menirukan tingkah laku seperti manusia yang dikenalkan pada tahun 1950 dengan cara kerja pengujian melalui teletype yaitu jika penguji (integrator) tidak dapat membedakan yang mengintrogasi antara manusia dan computer maka computer tersebut dikatakan lolos (menjadi kecerdasan buatan).

2. *Thinking Humanly*

Yaitu sistem yang dilakukan dengan cara intropeksi yaitu penangkapan pemikiran psikologis manusia pada computer, hal ini sering diujikan dengan neuron ke neuron lainnya atau sel otak

dengan sel otak lainnya cara pembelajarannya yaitu melalui eksperimen-eksperimen.

3. *Thinking Rationally*

Ini merupakan system yang sangat sulit ,karena sering terjadi kesalah dala, prinsip dan prakteknya,system ini dikenal dengan penalaran komputasi.

4. *Actng Rationally*

Yaitu system yang melakukan aksi dengan cara menciptakan suatu robotika cerdas yang menggantikan tugas manusia.

Berikut ini beberapa cabang ilmu sub bagian dari

AI :

1. *Natural Languange Processing (NLP)*

Natural Languange Processing (NLP) atau Pemrosesan Bahasa Alami, merupakan salah satu cabang AI yang mempelajari pembuatan sistem untuk menerima masukan bahasa alami manusia. Dalam perkembangannya, NLP berusaha untuk mengubah bahasa alami komputer (bit dan byte) menjadi bahasa alami manusia yang dapat kita mengerti. NLP merupakan ilmu dasar yang dapat dijadikan jembatan untuk membuat komunikasi antara mesin dengan manusia.

2. *Expert System (ES)*

Expert System (ES) atau Sistem Pakar, merupakan salah satu cabang AI yang mempelajari pembuatan sebuah sistem yang dapat bekerja layaknya seorang pakar. ES dapat menyimpan pengetahuan seorang pakar dan memberikan solusi berdasarkan pengetahuan yang dimilikinya tadi. ES juga merupakan salah satu cabang AI yang sering melakukan kerja sama dengan disiplin ilmu lain karena sifatnya yang dapat menyimpan pengetahuan.

3. *Pattern Recognition (PR)*

Pattern Recognition (PR) atau Pengenalan Pola, merupakan salah satu cabang AI yang mempelajari pembuatan sebuah sistem untuk dapat mengenali suatu pola tertentu. Misalnya sistem PR untuk mengenali huruf dari tulisan tangan, walaupun terdapat perbedaan penulisan huruf A dari masing-masing orang tetapi PR dapat mengenali bahwa huruf tersebut adalah huruf A. Beberapa aplikasi dari PR antara lain : *voice recognition, Fingerprint Identification, Face Identification, Handwriting Identification, Optical Character Recognition, Biological Slide Analysis, Robot Vision* dan lainnya.

4. *Robotic*

Robotic atau Robotika, merupakan salah satu cabang AI yang menggabungkan cabang-cabang AI yang lain termasuk ketiga cabang di atas untuk membentuk sebuah sistem robotik.

Keempat cabang AI di atas merupakan cabang umum yang banyak dipelajari, masih banyak cabang-cabang AI yang lainnya. Seiring perkembangan riset dalam AI, dapat dimungkinkan akan muncul cabang-cabang baru yang melengkapi unsur AI sehingga AI menjadi sebuah sistem lengkap dan akan mencapai goal-nya yang sampai sekarang masih belum sempurna.

1.2 *Intelligent Agent*

Berikut merupakan jenis-jenis *intelligent agent* :

1. *Reflect Agents*

Reflect agents merupakan agen yang langsung memberikan reaksi terhadap lingkungan tanpa berpikir panjang. Reflect Agents memiliki 2 pilihan yaitu kiri atau kanan. *Reflect Agents* bermanfaat pada lingkungan yang sederhana.

2. *Model-Based Agents*

Model-Based agents memiliki memori. Berfungsi untuk memodelkan lingkungan. *Model-based agents* tidak memiliki goal.

3. *Goal-Based Agents*

Goal-Based agents hanya memikirkan bagaimana mencapai goal saja, tidak memperdulikan jarak yang ditempuh. Agen ini efektif tetapi tidak efisien.

4. *Utility-Based Agents*

Utility-Based agents hampir sama seperti *Goal-Based Agents* tetapi agen ini memikirkan jarak yang ditempuh. Agen ini akan memilih jalan yang lebih cepat untuk mencapai tujuan. *Utulity-Based Agents* merupakan agen yang efektif dan efisien.

5. *Learning Agents*

Learning agents akan belajar dari pengalaman. Sehingga agen ini akan terus memperbaiki diri agar tidak terjadi kesalahan lagi. Agen ini tidak akan melakukan kesalahan yang sama. Agen ini biasanya digunakan untuk mengetes suatu masalah.

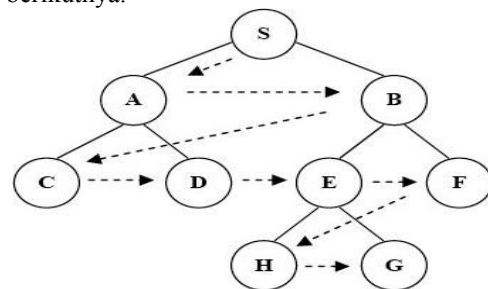
II. LANDASAN TEORI

2.1 *Blind Searching*

Terdapat 3 metode dalam search algoritma, yaitu

1. *Breadth-First Search*

Merupakan pencarian kesamping dimulai dari yang paling kiri. Pencarian akan dimulai pada level yang sama terlebih dahulu kemudia ke level berikutnya.



Gambar 1. *Breadth-First Search*

Keuntungan :

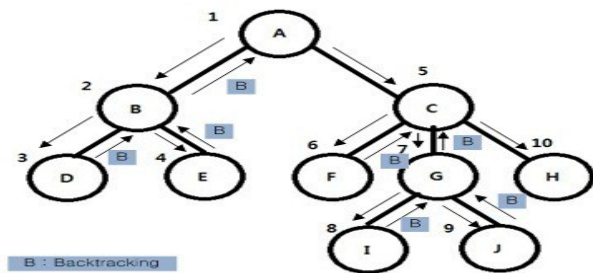
1. Tidak akan menemui jalan buntu
2. Jika ada satu solusi, maka breadth-first search akan menemukannya. Dan, jika ada lebih dari satu solusi, maka solusi minimum akan ditemukan.

Kelemahan :

1. Membutuhkan memori yang cukup banyak, karena menyimpan semua node dalam satu pohon
2. Membutuhkan waktu yang cukup lama, karena akan menguji n level untuk mendapatkan solusi pada level yang ke-(n+1).

2. *Depth-First Search*

Merupakan pencarian kedalam dari atas kebawah mulai dari yang paling kiri setelah itu baru kekanan.



Gambar 2. Depth-First Search

Keuntungan :

1. Membutuhkan memori yang relative kecil, karena hanya node-node pada lintasan yang aktif saja yang disimpan.
2. Secara kebetulan, metode depth-first search akan menemukan solusi tanpa harus menguji lebih banyak lagi dalam ruang keadaan.

Kelemahan :

1. Memungkinkan tidak ditemukannya tujuan yang diharapkan
2. Hanya akan menemukan 1 solusi pada setiap pencarian

3. Uniform-Cost Search

Merupakan pencarian yang gabungan dari Breadth-First Search dan Depth-First Search. Metode ini akan memilih jalan yang singkat. Metode merupakan metode yang paling optimal dalam mencari solusi.

2.2 Heuristic Search

Heuristic search merupakan metode pencarian yang memperhatikan nilai heuristik(nilai perkiraan). heuristik memperkirakan jarak ke Goal (yang disebut dengan fungsi heuristik)

salah satu contoh huristi search adalah Best First Search yang di bagi 2

1. Greddy Best
2. A* (baca A star)

kedua teknik ini memiliki persamaan dan perbedaan, persamaannya adalah sama-sama menggunakan nilai heuristic (perkiraan) perbedaannya pada greddy best

$$f(n) = h(n) \tag{1}$$

Dimana (* h(n) fungsi heuristik itu sendiri)

Pada A*

$$f(n) = h(n) + g(n) \tag{2}$$

Dimana (*g(n) merupakan aktual cost atau total jarak menuju ke n node).

III. HASIL DAN PEMBAHASAN

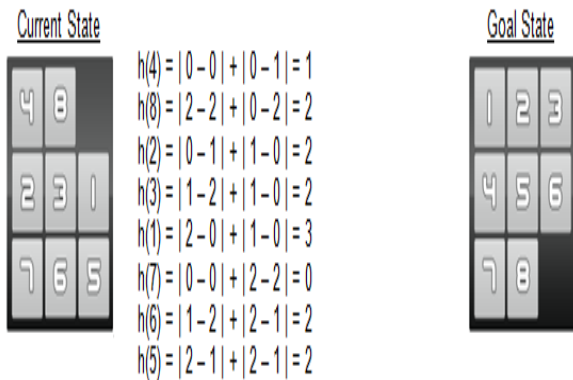
8 puzzle merupakan salah satu implementasi dari Artificial Intelgence. Dalam proses penyelesaiannya banyak terdapat algoritmaalgoritma pencarian yang dapat diterapkan. Dalam hal ini penulis memilih algoritma Steepest-Ascent Hill Climbing (Hill Climbing dengan memilih kemiringan yang paling tajam / curam) yang divariasikan dengan fungsi heuristik jarak dan posisi.

Heuristic dalam konteks yang luas dapat didefinisikan sebagai suatu nilai informasi yang “dianggap” mendekati nilai solusi dari suatu permasalahan. Sebagai contoh, jika kita misalkan ada seseorang yang berada di Bandung hendak menuju Surabaya, lalu orang tersebut ingin mencari rute jalan darat terpendek yang dapat dilalui dari Bandung menuju Surabaya. Maka nilai *heuristic*nya bisa kita tentukan dari, misal jarak estimasi dari setiap titik keberangkatan menuju Surabaya berdasarkan jarak yang tertulis pada peta. Atau bisa saja nilai *heuristic*nya didasarkan pada jarak setiap titik keberangkatan terhadap stasiun kereta api terdekat. Yang jelas, nilai *heuristic* ini dapat sangat relatif berdasar penilaian masing-masing pembuat keputusan. Tapi perlu diingat, nilai *heuristic* ini digunakan dalam proses evaluasi setiap periode tertentu atau setiap titik evaluasi tertentu. Jadi sebisa mungkin nilai *heuristic* yang kita tentukan bersifat *admissible* atau dapat diterima secara semantik, baik karena kedekatannya dengan solusi maupun “masuk akal” dari sisi logika. Jika tidak demikian, maka bisa jadi bahwa proses pencarian yang kita lakukan malah akan menjauhkan kita dari solusi.

Pada 8-Puzzle Problem, kita juga dapat menentukan nilai *heuristic*nya. Namun agak berbeda dengan permasalahan bertipikal pencarian jarak terpendek, nilai *heuristic* pada 8-Puzzle langsung ditentukan berdasar kondisi kedekatannya dengan *goal*, karena kita tidak pernah tahu jarak atau langkah yang kira-kira dapat ditempuh dari *state* sekarang ke *goal*nya.

Pada berbagai referensi, ada beberapa nilai *heuristic* yang dapat dijadikan acuan untuk permasalahan 8-Puzzle. Salah satu yang cukup terkenal adalah Manhattan Distance. Manhattan Distance didefinisikan sebagai penjumlahan jarak masing-masing kotak 8-Puzzle terhadap posisinya yang benar pada kondisi *goal*. Sehingga, pada kondisi *goal*, *heuristic* pasti akan bernilai 0, karena semua kotak sudah pada posisinya masing-masing (jarak dengan posisinya yang benar = 0). Cara pendekatan seperti ini telah cukup dianggap *admissible* atau masuk akal.

Berikut adalah contoh perhitungan Manhattan Distance untuk suatu *state* terhadap kondisi *goal*nya.



Sehingga didapatkan nilai *heuristic* untuk *current state* terhadap *goal state* adalah:

$$\begin{aligned}
 h(n) &= h(4) + h(8) + h(2) + h(3) + h(1) + h(7) \\
 &+ h(6) + h(5) \\
 &= 1 + 2 + 2 + 2 + 3 + 0 + 2 + 2 \\
 &= 14
 \end{aligned}$$

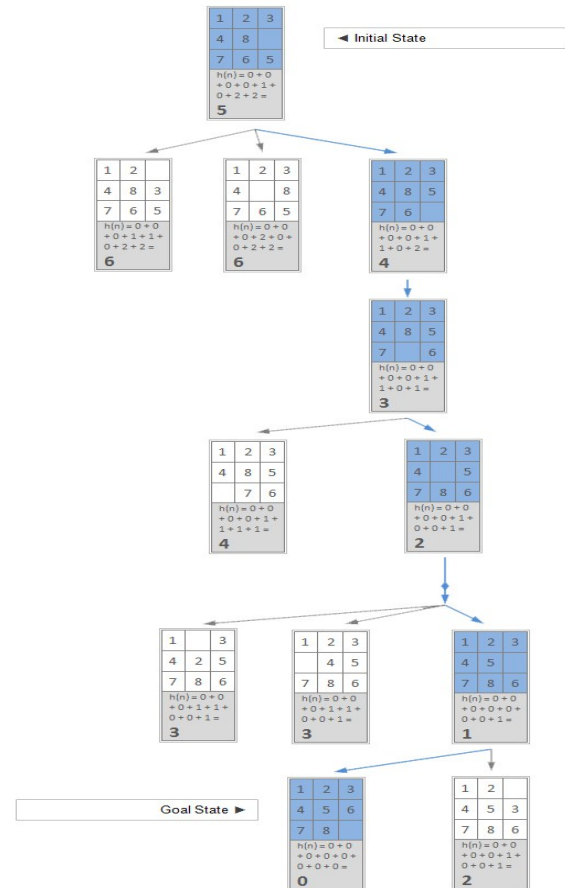
Seperti telah dikatakan sebelumnya, bahwa penilaian *heuristic* ini sangat relatif berdasar cara penalaran si pembuat keputusan. Jadi, sebenarnya penilaian ini juga dapat disebut sebagai penerapan strategi untuk permasalahan 8-Puzzle. Semakin bagus strateginya, maka akan semakin mangkus jalur pencarian yang dihasilkan.

Berikutnya, yang kita lakukan adalah menentukan metodologi atau algoritma pencariannya. Dengan menentukan metodologi pencarian, berarti kita menentukan bagaimana proses evaluasi yang dilakukan, tentunya menggunakan penilaian *heuristic* yang telah kita tentukan. Ada banyak metodologi yang dapat diterapkan untuk permasalahan 8-Puzzle. Berikut ini akan saya bahas beberapa algoritma populer yang dapat diterapkan untuk mencari jalur langkah penyelesaian 8-Puzzle.

Algoritma Hill Climbing merupakan salah satu teknik optimasi matematis yang termasuk ke dalam kategori *local search*. Disebut *local search* karena hanya melakukan evaluasi terhadap kemungkinan-kemungkinan *state* yang saat ini sedang dihadapi. Ketika telah memilih salah satu *state* yang dianggap terbaik, maka Hill Climbing akan melanjutkan pencarian hanya berdasar *state* yang telah dipilih tersebut, hingga mencapai kondisi goalnya. Jadi, ketika telah dipilih satu jalur, maka jalur yang lain akan diabaikan. Itulah mengapa Hill Climbing sering dianggap sebagai cara pencarian *heuristic* yang tercepat, karena hanya melakukan *simple evaluation* terhadap beberapa kemungkinan *state* yang dianggap terbaik, lalu memilihnya, dan melupakan kemungkinan lain yang berada di luar kondisi evaluatifnya.

Pada 8-puzzle, satu kali proses evaluasi menggunakan Hill Climbing hanya akan melibatkan maksimal 4 *state* untuk kondisi *initial state*, dan maksimal 3 *state* untuk kondisi *state* selain *initial state*. Sehingga *state space* untuk algoritma ini dapat dikatakan relatif sangat kecil. Berikut adalah contoh

bagaimana algoritma Hill Climbing dijalankan untuk suatu kasus *state* tertentu. Perlu diingat bahwa proses evaluasi yang dilakukan akan selalu mengambil nilai *heuristic* yang paling kecil.



Tetapi, Hill Climbing mempunyai dua (2) kelemahan:

1. *Plateau*, kondisi ketika ada dua (2) atau lebih *evaluation state* yang mempunyai nilai *heuristic* sama besar dan juga merupakan nilai terbaik.
2. *Local Maxima*, yaitu solusi lokal yang ditemukan dengan fungsi evaluasi. Tetapi solusi ini bukan kondisigoal yang diharapkan, dan jika dilakukan evaluasi secara terus menerus, maka akan kembali lagi ke kondisi solusi lokal itu sendiri. Karena memang fungsi evaluasi yang dilakukan menemui batasan pencarian lokal.

Untuk mengatasi kelemahan tersebut, ada strategi yang dapat diterapkan pada Hill Climbing agar solusi dapat ditemukan dengan baik. Strategi tersebut adalah:

1. Untuk *plateau* cukup dapat diatasi dengan menerapkan aturan prioritas pergerakan kotak kosong. Misal, pergerakan kotak kosong ke atas adalah lebih diprioritaskan daripada pergerakan ke kiri, pergerakan kotak ke kiri lebih diprioritaskan daripada pergerakan ke kanan, dst.
2. Sedangkan untuk *local maxima*, diperlukan lompatan besar (**big jump**) dengan memilih *evaluation state* yang lain ketika dalam kondisi *local maxima*. *Big jump* dapat diartikan dengan memilih *state* lain yang memiliki

3. kedekatan *heuristic* dengan *state* terbaik, atau dengan cara melakukan pemilihan acak (*random*) terhadap *state* lainnya yang bukan terbaik

IV. PENUTUP

A. Kesimpulan

Kesimpulan pada makalah ini yaitu:

1. Persoalan 8 puzzle dapat diselesaikan dengan menggunakan prinsip array serta algoritma Steepest-Ascent Hill Climbing yang telah divariasikan.
2. Solusi 8 puzzle akan lebih cepat diperoleh jika digunakan prinsip array dengan variasi algoritma Steepest-Ascent Hill Climbing (Hill Climbing dengan memilih kemiringan yang paling tajam / curam) dengan parameter heuristik posisi yang benar dan heuristik jarak serta dikombinasikan dengan LogList sebagai penyimpanan state state yang pernah dilalui untuk menanggulangi permasalahan pada algoritma hill climbing itu sendiri.

V. DAFTAR PUSTAKA

- [1] Kusumadewi, Sri. Pengantar Kecerdasan Buatan (AK045218):
- [2] Teknik Pencarian Heuristik. Taufiq, Andik. 2010. 8-Puzzle Problem Bagian
- [3] <http://andiktaufiq.wordpress.com/2010/05/02/8-puzzle-problem-bagian-2/>. (Diakses tanggal 25 September 2014)
- [4] Boylestad, Robert. 1992. Electronic Devices and Circuit Theory. Englewood Cliffs: Prentice Hall.
- [5] Microcontroller Databook. 1995. San Jose: Atmel Corporation.
- [6] Nist Sematech, 2007. e-Handbook of Statistical Methods: Single Response Case. <<http://www.itl.nist.gov/div898/handbook/pri/section5/pri531.htm>>
- [7] Rich, Elaine. 1991. Artificial Intelligence. New York: McGraw-Hill.