

PENERAPAN ALGORITMA BEST FIRST SEARCH DALAM PENYELESAIAN PERGESERAN ANGKA DENGAN POLA PERMAINAN BERBENTUK BINTANG

Ruth Merywati Mikan, Marla Sheilamita Shalin Pieter

**Program Studi Teknik Informatika
Fakultas Ilmu Komputer dan Manajemen
Universitas Sains dan Teknologi Jayapura (USTJ)**

Abstraksi - Permainan merupakan sesuatu yang dimainkan dengan beberapa aturan yang telah ditentukan untuk melatih kecerdasan dalam mengatur strategi, kecepatan, dan ketepatan dalam mencapai tujuan akhir. Salah satu contohnya adalah permainan pergeseran angka dengan pola berbentuk bintang yang terdapat delapan arah proses pergeseran angka, yaitu atas, atas kiri, atas kanan, bawah, bawah kiri, bawah kanan, kiri, dan kanan yang hanya dapat digeser mengikuti jalur yang ada sehingga rumit untuk diselesaikan secara manual dan juga membutuhkan waktu yang cukup lama karena tidak ada informasi tambahan yang dimiliki. Permainan ini dapat diselesaikan dengan bantuan pohon pencarian secara heuristik menggunakan algoritma *Best First Search* yang merupakan salah satu teknik pemecahan masalah pada bidang kecerdasan buatan.

Penelitian ini bertujuan untuk merancang bangun sebuah aplikasi yang mampu menyelesaikan permasalahan pergeseran angka berbentuk bintang menggunakan algoritma *Best First Search* (BFS).

Hasil berupa Aplikasi Permainan Pergeseran Angka menggunakan Algoritma *Best First Search*. Pada aplikasi terdapat solusi, yaitu berupa Animasi pergerakan dari keadaan awal menuju keadaan akhir yang tercatat waktu, banyaknya pergerakan dan langkah-langkah yang dihasilkan untuk menemukan solusi, baik pada bentuk bintang segi lima baik pada bentuk bintang segi lima, bintang segi enam dan bintang segi tujuh.

Kata Kunci : *Permainan Pergeseran Angka, Heuristik, Best First Search*

1. PENDAHULUAN

1.1 Latar Belakang

Permainan merupakan sesuatu yang dimainkan dengan beberapa aturan yang telah ditentukan untuk melatih kecerdasan dalam mengatur strategi, kecepatan, dan ketepatan dalam mencapai tujuan akhir. Salah satu contoh permainan yang sering dimainkan adalah pergeseran angka (*Sliding Puzzle*). *Puzzle* biasanya dimainkan dengan pola berbentuk persegi sehingga lebih mudah untuk dimainkan dan diselesaikan karena arah proses pergeserannya hanya berupa empat arah pergeseran, yaitu atas, bawah, kanan, dan kiri. Permainan ini akan jauh lebih rumit apabila dimainkan dalam pola yang berbentuk bintang karena terdapat delapan arah proses pergeseran angka, yaitu atas, atas kiri, atas kanan, bawah, bawah kiri, bawah kanan, kiri, dan kanan yang hanya dapat digeser mengikuti jalur yang ada sehingga rumit untuk diselesaikan secara manual.

Permainan pergeseran angka berbentuk bintang ini dapat diselesaikan dengan bantuan pohon pencarian secara heuristik yang merupakan salah satu teknik pemecahan masalah pada bidang kecerdasan buatan. Pencarian heuristik mampu melakukan pencarian solusi langsung pada cabang dari pohon yang memuat tujuan tanpa harus mengunjungi *node-node* lain yang tidak perlu. Salah satu algoritma pencarian heuristik yang akan digunakan dalam penyelesaian permainan pergeseran angka berbentuk bintang ini adalah algoritma *Best First Search* yang merupakan kombinasi dari dua algoritma pencarian buta, yaitu *Breadth First Search* dan *Depth First Search* dengan mengambil kelebihan dari kedua algoritma tersebut.

Permainan pergeseran angka berbentuk bintang selain rumit untuk diselesaikan secara manual dan juga membutuhkan waktu yang cukup lama karena pada permainan pergeseran angka tidak ada informasi tambahan yang dimiliki untuk membantu melakukan pencarian solusi, agar membentuk susunan angka seperti semula. Oleh sebab itu diperlukan sebuah algoritma untuk membantu proses pencarian solusi yang lebih pendek serta sebagai pembelajaran dalam bidang kecerdasan buatan dan penerapannya pada permainan pergeseran angka berpola bintang dengan menggunakan algoritma *Best First Search*.

1.1. Rumusan Masalah

Masalah yang dijadikan sebagai acuan dalam penelitian ini adalah “Bagaimana membantu pencarian solusi yang lebih pendek dalam menyelesaikan permasalahan pergeseran angka dengan pola permainan berbentuk bintang ?”

1.2. Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah “Merancang bangun sebuah aplikasi untuk membantu proses pencarian solusi yang lebih pendek dalam menyelesaikan permasalahan pergeseran angka dengan pola permainan berbentuk bintang menggunakan algoritma *Best First Search* (BFS)”.

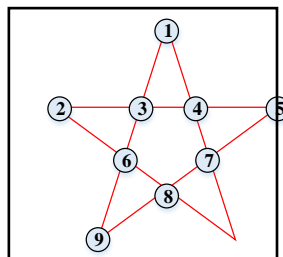
2. TINJAUAN PUSTAKA

2.1. Permainan Pergeseran Angka Berbentuk Bintang

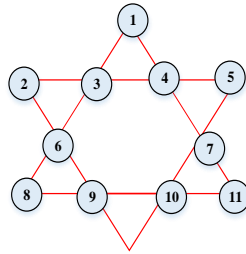
Permainan pergeseran angka pada bentuk bintang mirip dengan permainan pergeseran angka dalam kotak berbentuk persegi. Perbedaannya adalah bentuk wadah yang digunakan dalam permainan ini berbentuk bintang (*star polygon*). Setiap titik perpotongan dalam bintang merupakan titik (*node*) penempatan angka. Seperti halnya, permainan pergeseran angka lainnya, dalam permainan ini juga disediakan satu tempat kosong sebagai ruang untuk menggeser posisi angka. Aturan permainannya adalah sebagai berikut:

- a. Tentukan keadaan awal (*initial state*) pada bintang.
- b. Tentukan keadaan tujuan (*goal state*) pada bintang.
- c. Aturan pergeseran: setiap angka dalam bintang hanya dapat digeser ke suatu titik yang kosong atau tidak ditempati oleh angka lain.
- d. Setiap angka hanya dapat digeser ke atas, atas kiri, atas kanan, bawah, kiri dan kanan mengikuti jalur yang sudah ada sesuai dengan bentuk bintang yang digunakan.

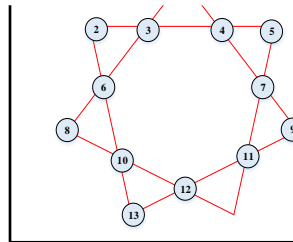
Permasalahan dari permainan ini adalah bagaimana mencapai *keadaan akhir* dari *keadaan awal* dengan mengikuti aturan pergeseran yang telah ditetapkan. Pada bintang segi lima (*Pentagon*) memiliki 10 buah titik, tersedia 9 buah titik yang memuat angka dan 1 buah titik kosong, bintang segi enam (*Heksagon*) memiliki 12 buah titik, tersedia 11 buah titik yang memuat angka dan 1 buah titik kosong dan bintang segi tujuh (*Heptagon*) memiliki 14 buah titik, tersedia 13 buah titik yang memuat angka dan 1 buah titik kosong. Bentuk bintang segi lima, segi enam, dan segi tujuh dapat dilihat pada gambar berikut ini.



Gambar 2.1. Bentuk Bintang Segi lima



Gambar 2.2. Bentuk Bintang Segi enam



Gambar 2.3. Bentuk Bintang Segi tujuh

2.2. Metode Pencarian Heuristik

Kata *heuristic* berasal dari sebuah kata kerja bahasa Yunani, *heuriskein*, yang berarti ‘mencari’ atau ‘menemukan’. Dalam dunia pemrograman, sebagian orang menggunakan kata *heuristic* sebagai lawan kata dari algoritmik, dimana kata heuristik ini diartikan sebagai ‘suatu proses yang mungkin dapat menyelesaikan suatu masalah tetapi tidak ada jaminan bahwa solusi yang dicari selalu dapat ditemukan’. Di dalam mempelajari metode-metode pencarian ini, kata heuristic diartikan sebagai suatu fungsi yang memberikan suatu nilai berupa biaya perkiraan (estimasi) dari suatu solusi.

Pencarian heuristik berguna dalam memecahkan masalah sulit yang tidak dapat diselesaikan dengan cara lain dan di mana solusi membutuhkan waktu yang tidak terbatas atau waktu yang sangat lama untuk menghitung.

Metode-metode yang termasuk ke dalam teknik pencarian yang berdasarkan pada teknik pencarian heuristik adalah: *Generate and Test*, *Hill Climbing* (*Simple Hill Climbing* dan *Steepest-Ascent Hill Climbing*), *Simulated Annealing*, *Best First Search* (*Greedy Best First Search* dan *A** dengan berbagai variasinya).

2.3. Algoritma Best First Search (BFS)

Best First Search merupakan kombinasi dari algoritma *Depth First Search* dan algoritma *Breath First Search* dengan mengambil kelebihan dari kedua algoritma tersebut. Apabila pada pencarian dengan algoritma *Hill Climbing* tidak diperbolehkan untuk kembali ke *node* pada level yang lebih rendah meskipun *node* pada level yang lebih rendah tersebut memiliki nilai heuristik yang lebih baik, lain halnya dengan algoritma *Best First Search*, pencarian diperbolehkan mengunjungi *node* yang ada di level yang lebih rendah, jika ternyata *node* pada level yang lebih tinggi memiliki nilai heuristik yang lebih buruk.

Dalam pencarian *Best First Search*, ruang pencarian dievaluasi sesuai dengan fungsi heuristik yang dinyatakan dengan persamaan berikut:

$$f(n) = h(n) \dots \dots \dots (2.1)$$

Dimana:

f(n) : fungsi heuristik

h(n): fungsi evaluasi yang dipakai untuk mengestimasi seberapa baik setiap setiap *node* dibangkitkan.

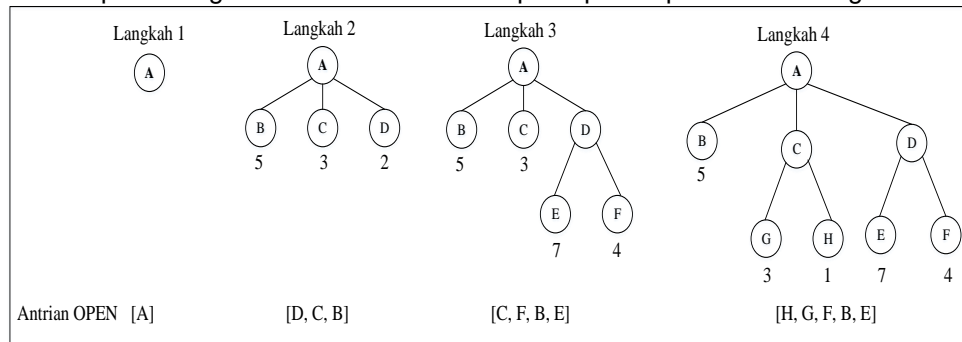
Untuk mengimplementasikan algoritma *Best First Search* tersebut, diperlukan dua buah senarai (*list*), yaitu: *OPEN* dan *CLOSED*. *Node* yang harus dievaluasi disimpan dalam daftar *OPEN* dan yang telah dievaluasi disimpan dalam daftar *CLOSED*. Daftar *OPEN* diwakili

sebagai antrian prioritas, sehingga *node* yang tidak dikunjungi dapat dikenai fungsi evaluasinya.

Secara informal, *pseudo code* algoritma *Best First Search* diilustrasikan oleh algoritma dibawah ini.

- a. OPEN berisi initial state dan CLOSED masih kosong.
- b. Ulangi sampai goal ditemukan atau sampai tidak ada *nodes* di dalam OPEN.
 - 1) Ambil simpul terbaik yang ada di OPEN
 - 2) Jika simpul tersebut sama dengan *goal*, maka sukses
 - 3) Bangkitkan semua suksesor dari simpul tersebut
 - 4) Untuk setiap suksesor kerjakan:
 - a) Jika suksesor tersebut belum pernah dibangkitkan, evaluasi suksesor tersebut, tambahkan ke OPEN, dan catat parent atau orang tuanya.
 - b) Jika suksesor tersebut sudah pernah dibangkitkan, ubah parent yang sebelumnya. Selanjutnya, perbarui biaya untuk suksesor tersebut dan nodes lain yang berada di level bawahnya.

Contoh proses algoritma *Best First Search* pada pohon pencarian sebagai berikut:



Gambar 2.4. Pohon Pencarian Algoritma Best First Search

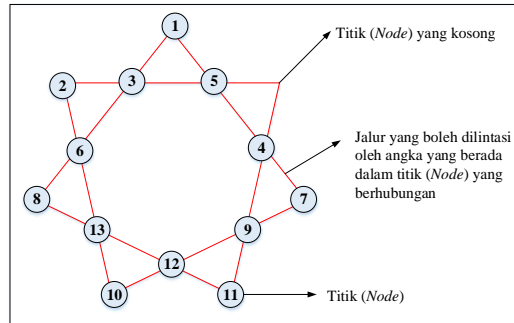
Berdasarkan Gambar 2.4 dapat dijelaskan bahwa yang pertama kali dibangkitkan adalah *node* A. Kemudian pada langkah 2 semua *successor* A dibangkitkan, dan dicari nilai heuristik paling kecil. *Node* A memiliki cabang B, C, dan D yang masing-masing bernilai 5, 3, dan 2. Dengan demikian *node* D terpilih karena memiliki nilai heuristik paling kecil. Pada langkah 3 semua *successor* D dibangkitkan dan menghasilkan *node* E dan F yang masing-masing bernilai 7 dan 4 yang akan dibandingkan nilai heuristiknya dengan *node* B dan C yang masing-masing bernilai 5 dan 3. Yang terpilih adalah *node* B karena memiliki nilai heuristik paling kecil. Pada langkah 4 semua *successor* C dibangkitkan dan menghasilkan *node* G dan H yang masing-masing bernilai 3 dan 1 yang akan dibandingkan nilai heuristiknya dengan *node* B, E, dan F yang masing-masing bernilai 5, 7, dan 4. Yang terpilih adalah *node* H karena memiliki nilai heuristik paling kecil. Selanjutnya semua *successor* H dibangkitkan. Demikian seterusnya sampai ditemukan *node* tujuan.

3. HASIL DAN PEMBAHASAN

3.1. Penyelesaian Pergeseran Angka dengan Algoritma *Best First Search* (BFS)

a. Pohon Pencarian Best First Search

Terdapat delapan arah proses pergeseran angka, yaitu angka dapat digeser ke atas, atas kiri, atas kanan, bawah, bawah kiri, bawah kanan, kiri, dan kanan yang hanya dapat digeser mengikuti jalur yang ada seperti pada gambar dibawah ini.

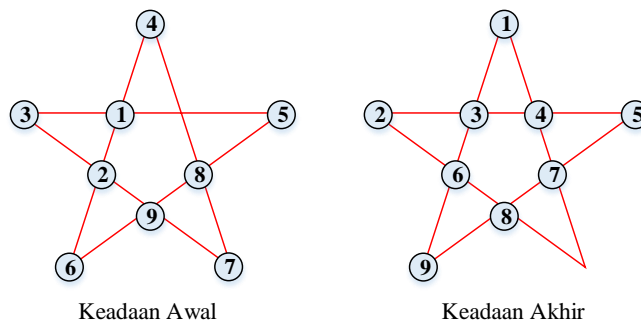


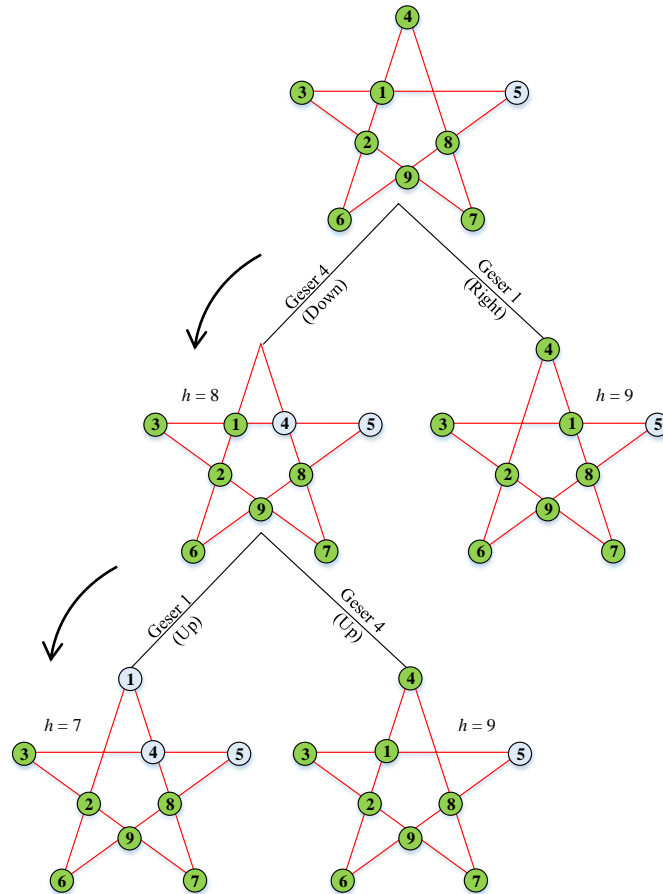
Gambar 3.1. Aturan Pergeseran Angka Dalam Bintang

Bentuk pohon pencarian digunakan untuk merepresentasikan ruang keadaan menggunakan algoritma *Best First Search* dengan cara mencari langkah-langkah yang harus ditempuh dari keadaan awal sampai mendapatkan keadaan akhir sesuai dengan fungsi heuristik yaitu, $f(n) = h(n)$

Nilai heuristik (h) didapatkan dari total perhitungan jumlah *node* pada keadaan saat ini yang tidak sesuai dengan keadaan akhir. Pada pohon pencarian, warna hijau menandakan *node* yang posisinya tidak sesuai dengan keadaan akhir dan warna biru menandakan *node* dengan posisi yang sesuai dengan keadaan akhir. Setelah mendapatkan nilai heuristik (h), selanjutnya akan dibangkitkan bentuk bintang dengan jumlah nilai heuristik yang lebih kecil.

Apabila terdapat nilai heuristik yang sama, maka akan diterapkan aturan prioritas pergerakan *node* dengan cara menggeser *node* yang hampir mendekati posisi keadaan akhir pada bentuk bintang. Berikut ini merupakan salah satu contoh pohon pencarian solusi menggunakan *Best First Search* pada bintang berbentuk segi lima.

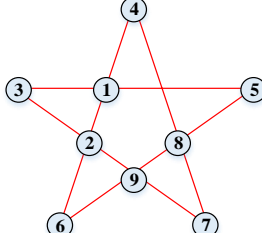


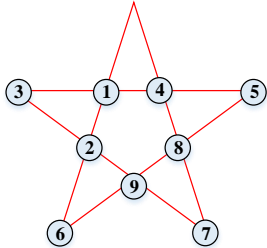
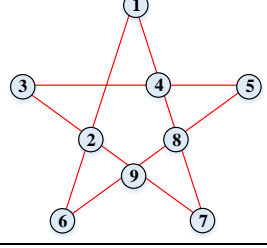
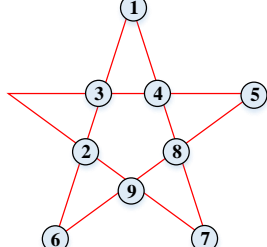
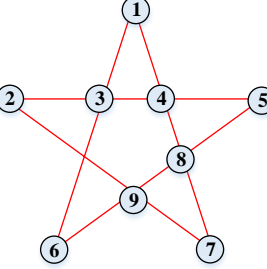
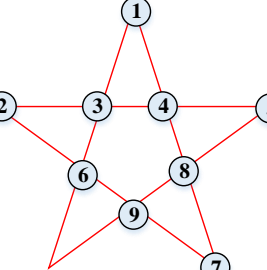


Berdasarkan gambar diatas, pertama kali dibangkitkan *node* '4310528967' dan semua *successor* '0314528967' dibangkitkan sehingga didapat *node* '0314528967' dan '4301528967', kemudian dihitung nilai heuristik dari kedua *node* tersebut lalu dibandingkan. Setelah dibandingkan *node* '0314528967' memiliki heuristik terkecil, maka *node* '0314528967' dibangkitkan dan didapat *node* '1304528967' dan '4310528967', kemudian dicari nilai heuristik terkecil. Sehingga *node* '1304528967' yang terpilih karena nilai heuristiknya paling kecil, yaitu 7. Demikian seterusnya sampai ditemukan *node* tujuan.

b. Langkah-langkah yang dihasilkan dari perhitungan fungsi heuristik, yaitu:

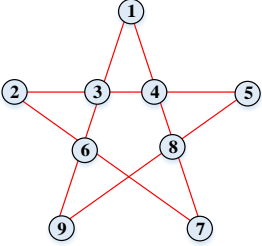
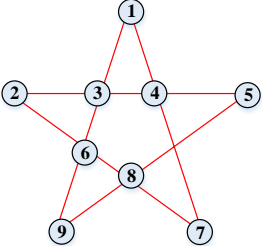
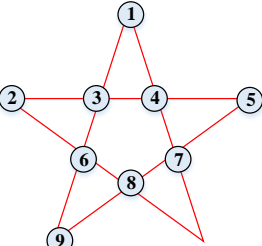
Tabel 3.1. Solusi Contoh Kasus pada Bintang Berbentuk Segi Lima

No	Langkah	Kondisi
0	Kedaan awal	 4310528967

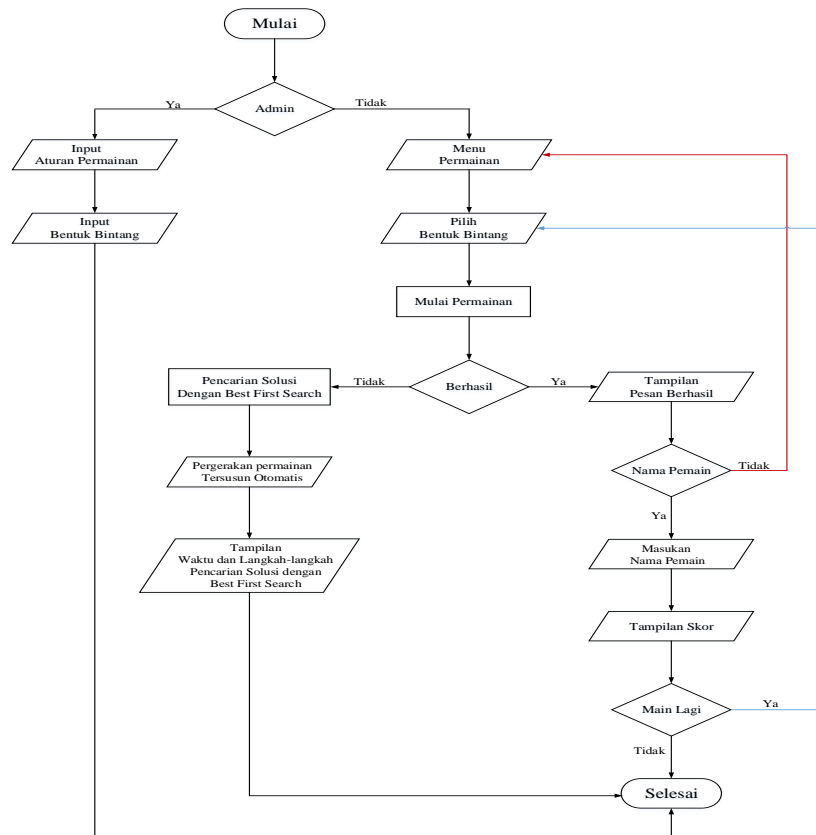
1	Geser angka "4" dari posisi 1 ke posisi 4		0314528967
2	Geser angka "1" dari posisi 3 ke posisi 1		1304528967
3	Geser angka "3" dari posisi 2 ke posisi 3		1034528 967
4	Geser angka "2" dari posisi 6 ke posisi 2		1234508967
5	Geser angka "6" dari posisi 9 ke posisi 6		1234568907

Lanjutan Tabel 3.1

No	Langkah	Kondisi
----	---------	---------

6	Geser angka "9" dari posisi 8 ke posisi 9		1234568 0 97
7	Geser angka "8" dari posisi 7 ke posisi 8		123456 0 897
8	Geser angka "7" dari posisi 10 ke posisi 7		1234567 8 90

c. Flowchart



3.2. Pembahasan

a. Tamp

meng

mulai mencari penyelesaian dari bentuk bintang yang telah diacak sebelumnya, setelah solusi ditemukan maka akan terjadi animasi pergerakan dan puzzle akan tersusun secara otomatis ke kondisi akhir, kemudian akan ditampilkan langkah-langkah yang ditempuh dari kondisi awal hingga kondisi akhir yang dihasilkan dari algoritma.

kan oleh sistem
aka sistem akan



Gambar 3.3. Implementasi Tampilan Form Pencarian Solusi Segi Lima



Gambar 3.4. Implementasi Tampilan Form Pencarian Solusi Segi Enam



Gambar 3.5. Implementasi Tampilan Form Pencarian Solusi Segi Tujuh

b. Perbandingan kasus menggunakan spesifikasi laptop yang berbeda

Tabel 3.2 Perbandingan Spesifikasi Laptop Berbeda

Spesifikasi Laptop	Kondisi Awal	Panjang Langkah	Waktu
Lenovo G40- 30 Processor: Intel Celeron N2840 Memory: 2 GB RAM	4 3 1 0 5 2 8 9 6 7	8 L	15 s
	1 6 3 5 4 2 8 9 0 7	11 L	1 m, 44 d
	4 3 1 5 0 2 11 6 8 10 7 9	13 L	2 m, 13 d
	3 6 2 1 4 0 5 8 7 10 9 12 13 11	9 L	18 s
	3 6 2 1 5 8 4 0 7 13 9 11 10 12	13 L	2 m, 26 d
Asus X202EP Processsor: Intel Celeron 847 Memory: 4 GB RAM	4 3 1 0 5 2 8 9 6 7	8 L	16 s
	1 6 3 5 4 2 8 9 0 7	11 L	1 m, 56 d
	4 3 1 5 0 2 11 6 8 10 7 9	13 L	3 m, 11 d
	3 6 2 1 4 0 5 8 7 10 9 12 13 11	9 L	20 s
	3 6 2 1 5 8 4 0 7 13 9 11 10 12	13 L	3 m, 21 d

Samsung 450R Processor: Intel Core i3 Memory: 8 GB RAM	4 3 1 0 5 2 8 9 6 7	8 L	15 s
	1 6 3 5 4 2 8 9 0 7	11 L	1 m, 31 d
	4 3 1 5 0 2 11 6 8 10 7 9	13 L	1 m, 13 d
	3 6 2 1 4 0 5 8 7 10 9 12 13 11	9 L	18 s
	3 6 2 1 5 8 4 0 7 13 9 11 10 12	13 L	1 m, 17 d

4. PENUTUP

4.1. Kesimpulan

Berdasarkan penelitian yang telah dilakukan pada permainan pergeseran angka berbentuk bintang menggunakan algoritma *Best First Search*, maka diambil kesimpulan sebagai berikut:

1. Algoritma *best first search* dapat diterapkan di dalam aplikasi permainan pergeseran angka bentuk bintang untuk mencari solusi permainan.
2. Aplikasi permainan pergeseran angka ini dapat digunakan sebagai salah satu sarana pembelajaran dalam bidang kecerdasan buatan untuk mempelajari dan mengenal algoritma *Best First Search*.
3. Pencarian solusi menggunakan *Best First Search* (BFS) dapat menghasilkan solusi dengan susunan angka pada kondisi awal permainan yang mudah, baik pada bentuk bintang segi lima, bintang segi enam dan bintang segi tujuh sehingga menghasilkan waktu untuk menemukan solusi yang cepat dan banyaknya langkah yang dihasilkan cukup pendek. Tetapi susunan angka pada kondisi awal permainan yang rumit, waktu pencarian yang dibutuhkan dengan menggunakan algoritma sedikit lebih lama dengan langkah pencarian yang cukup panjang dan belum tentu ditemukan solusi. jika permainan dimainkan secara manual dapat berhasil diselesaikan dengan jumlah langkah pergeseran angka yang panjang. Sehingga dibutuhkan spesifikasi laptop yang lebih tinggi untuk mengatasi kecepatan dalam proses pencarian solusi yang lebih cepat.
4. Algoritma *Best First Search* merupakan suatu proses yang mungkin dapat menyelesaikan suatu masalah tetapi tidak menjamin bahwa solusi yang dicari selalu dapat ditemukan. Hal ini disebabkan karena BFS membangkitkan simpul dari simpul sebelumnya dan memilih simbol baru yang memiliki biaya terkecil diantara simpul-simpul pada level terdalam yang pernah dibangkitkan. Dengan demikian algoritma BFS melakukan perhitungan yang lebih banyak
5. Aplikasi Permainan ini diuji coba dengan beberapa contoh kasus menggunakan laptop dengan spesifikasi yang berbeda dapat di hasilkan bahwa, laptop yang memiliki kapasitas memori yang besar tetapi kecepatan *processor*-nya rendah maka waktu yang dibutuhkan dalam pencarian solusi cukup lama begitu juga sebaliknya, jika laptop yang memiliki kecepatan *processor*-nya tinggi tetapi kapasitas memori kecil, maka waktu yang dibutuhkan dalam pencarian solusi cukup lama. Sehingga untuk dapat menghasilkan solusi dengan waktu yang cepat harus menggunakan laptop dengan spesifikasi kecepatan *processor* yang tinggi dan juga kapasitas memori yang besar.

4.2. Saran

Saran yang dapat diberikan untuk pengembangan dan penelitian selanjutnya adalah sebagai berikut:

1. Untuk algoritma penyelesaiannya dapat menggunakan algoritma-algoritma lain yang dapat menghasilkan langkah untuk menyelesaikan permasalahan pada permainan pergeseran angka yang jauh lebih sedikit serta dapat menambah algoritma lainnya sebagai pembandingan.
2. Keadaan akhir yang ingin dicapai bisa diacak atau ditentukan sendiri oleh pemain.

3. Mengembangkan aplikasi agar dapat digunakan di *platform* android atau *platform* mobile lainnya.
4. Aplikasi dapat dikembangkan dengan menambahkan beberapa bentuk wadah lainnya, seperti: segi delapan (*octagon*), segi Sembilan (*nonagon*) dan bentuk lainnya diluar bentuk bintang.
5. Aplikasi dapat dikembangkan dengan menggunakan perangkat sistem yang spesifikasinya tinggi agar menghasilkan pencarian solusi yang cepat pada saat pencarian solusi.

5. DAFTAR PUSTAKA

- Asmiatun, S., Astrid, N. P., 2017, *Belajar Membuat Game 2D dan 3D Menggunakan Unity*, Edisi 1, Cetakan 1, Yogyakarta: Deepublish. Diakses tanggal 13 Maret 2018, dari E-book.
- Arisandi, D., Romi, F. R., Siska, M. A., 2014, *Permainan Pergeseran Angka Bentuk Bintang Menggunakan Algoritma Best First Search*, Prosiding Konferensi Nasional Pengembangan Teknologi Informasi dan Komunikasi, Medan.
- Jones, M. Tim, 2008, *Artificial Intelligence (A Systems Approach)*, USA: Jones and Bartlett. Diakses tanggal 14 Maret 2018, dari E-book.
- Kusumadewi, Sri, 2003, *Artificial Intelligence (Teknik dan Aplikasinya)*, Yogyakarta: Graha Ilmu.
- Suyanto, 2014, *Artificial Intelligence (Searching, Reasoning, Planning, and Learning)*, Edisi Revisi Kedua, Bandung: Informatika.