

PERBANDINGAN ALGORITMA *BREADTH FIRST SEARCH* DAN *DEPTH FIRST SEARCH* PADA GAME *MUMMY MAZE DELUXE*

Emil Glorio Masala¹, Immanuela P. Saputro*¹, Rinaldo T.B. Turang¹

¹Program Studi Teknik Informatika; Fakultas Teknik

¹Universitas Katolik De La Salle Manado; Kombos – Kairagi I Manado, Telp:(0431) 871957

e-mail: *¹isaputro@unikadelasalle.ac.id

Abstrak-Game merupakan salah satu jenis hiburan yang bisa dimainkan oleh semua orang dengan tujuan bersenang-senang ataupun mengisi waktu luang. Salah satu jenis game yang bisa dimainkan yaitu maze game yang merupakan permainan teka-teki untuk mencari jalan keluar dengan tingkat kesulitan mulai dari level biasa sampai dengan level tinggi. Algoritma Breadth First Search dan Depth First Search merupakan algoritma pencarian buta yang berfungsi untuk menemukan tujuan pada suatu kasus dimana tidak ada informasi tambahan yang dimiliki untuk membantu melakukan pencarian.

Pada penelitian ini, penulis bermaksud menggunakan game maze sebagai media untuk membandingkan algoritma Breadth First Search dan Depth First Search dari sisi jumlah langkah pencarian dan lama waktu pencarian melalui sebuah aplikasi. Pemrograman aplikasi dilakukan dengan menggunakan bahasa pemrograman Visual Basic 6.0. Media yang digunakan oleh penulis adalah game mummy maze deluxe.

Hasil perbandingan dari kedua algoritma diperoleh dari sisi jumlah langkah dan waktu pencarian pada game mummy maze deluxe level 1, dimana algoritma DFS lebih unggul dari pada algoritma BFS berdasarkan jumlah langkah dan waktu dari proses pencarian yang telah uji. Algoritma DFS memperoleh solusi dengan jumlah langkah 26 dan waktu pencarian 07.48 detik. Sedangkan algoritma BFS memperoleh solusi dengan jumlah langkah 29 dan waktu pencarian 10.36 detik.

Kata Kunci— Algoritma Pencarian Buta, Breadth First Search, Depth First Search

I. PENDAHULUAN

Game merupakan salah satu jenis hiburan yang bisa dimainkan oleh semua orang dengan tujuan bersenang-senang ataupun mengisi waktu luang. Salah satu jenis game yang bisa dimainkan yaitu maze game atau permainan labirin. Game ini berupa permainan teka-teki untuk mencari jalan keluar dengan tingkat kesulitan mulai dari level biasa sampai dengan level tinggi [1].

Algoritma Breadth First Search (BFS) dan Depth First Search (DFS) merupakan algoritma pencarian buta yang sering digunakan dalam membuat sebuah aplikasi. Algoritma Breadth First Search adalah algoritma pencarian yang dilakukan secara melebar yang dimulai dari level paling pertama sampai node pada level tersebut habis, kemudian dilanjutkan ke anak pada level berikutnya sampai node tersebut habis. Jika belum menemukan solusi, maka pencarian terus dilanjutkan sampai solusi ditemukan karena algoritma BFS selalu menemukan solusi. Sedangkan algoritma DFS adalah algoritma pencarian yang dilakukan berdasarkan kedalaman yang dimulai dari level pertama kemudian dilanjutkan ke anak level berikutnya sampai tidak terdapat anak atau level yang lebih dalam lagi [2].

Penelitian ini bertujuan untuk melihat berapa jumlah langkah pencarian dan lama waktu proses penyelesaian pada kedua algoritma di level 1 melalui sebuah aplikasi. Media yang digunakan untuk membandingkan kedua algoritma BFS dan DFS adalah Game Mummy Maze Deluxe. Hal ini karena permainan yang dimaksud dinilai mempunyai tingkat kesulitan tinggi dalam mencari jalan keluar.

II. TINJAUAN PUSTAKA

A. Pohon Pencarian

Struktur pohon pencarian digunakan untuk menggambarkan keadaan secara terstruktur dan menghindari kemungkinan adanya proses pelacakan suatu node secara berulang. Struktur pohon pencarian terdiri dari beberapa node. Node yang terletak pada level-0 sering disebut akar yang menunjukkan keadaan awal. Node-node yang tidak memiliki anak sering disebut node daun yang menunjukkan akhir suatu pencarian, dapat berupa tujuan yang diharapkan (*state*) atau jalan buntu (*dead end*) [3][4].

B. Algoritma Pencarian Buta

Algoritma pencarian buta merupakan algoritma pencarian yang dilakukan tanpa ada informasi awal yang digunakan dalam proses pencarian. Pencarian buta berlangsung sampai solusi terakhir ditemukan dengan menguji seluruh kemungkinan yang ada untuk menemukan solusi. Algoritma pencarian buta terdiri atas dua bagian yaitu BFS dan DFS [5][6].

C. Algoritma Breadth First Search

Algoritma Breadth First Search adalah algoritma pencarian melebar yang dilakukan dengan mengunjungi node pada level n terlebih dahulu sebelum mengunjungi node-node pada level $n+1$. Kelebihan dari algoritma Breadth First Search yaitu dapat menemukan solusi sebagai jalur terpendek dan menemukan solusi minimum apabila ada lebih dari satu solusi. Algoritma Breadth First Search tidak akan menemui jalan buntu. Kekurangan dari algoritma Breadth First Search yaitu membutuhkan banyak memori untuk menyimpan semua simpul dalam satu pohon.

Cara kerja algoritma Breadth First Search yaitu masukkan simpul ujung ke dalam sebuah antrean kemudian ambil simpul dari awal antrean. Lakukan pengecekan apakah simpul awal merupakan solusi. Jika simpul merupakan solusi pencarian selesai dan hasil dikembalikan. Jika simpul bukan merupakan solusi, masukan seluruh simpul yang bertetangga dengan simpul tersebut. Apabila semua simpul sudah dicek dan antrean kosong, pencarian selesai dengan mengembalikan hasil solusi

tidak ditemukan. Pencarian diulang dari simpul awal antrean [7][8][9].

D. Algoritma *Depth First Search*

Algoritma *Depth First Search* adalah algoritma pencarian mendalam yang dimulai dari node awal dilanjutkan dengan hanya mengunjungi node anak paling kiri pada tingkat selanjutnya. Cara kerja algoritma *Depth First Search* yaitu masukan masukan node akar kedalam sebuah tumpukan. Kemudian ambil simpul pertama pada level paling atas, jika simpul merupakan solusi pencarian selesai dan hasil dikembalikan. Jika simpul bukan merupakan solusi, masukan seluruh simpul yang bertetangga dengan simpul tersebut ke dalam tumpukan. Apabila semua simpul sudah dicek dan antrean kosong, pencarian selesai dengan mengembalikan hasil solusi tidak ditemukan. Pencarian diulang dari simpul awal antrean.

Kelebihan dari algoritma *Depth First Search* yaitu pemakaian memori hanya sedikit untuk menyimpan node-node pada lintasan yang aktif dan menemukan solusi tanpa harus menguji lebih banyak dalam ruang keadaan. Algoritma *Depth First Search* dapat menemukan solusi secara cepat apabila solusi berada pada tingkat paling dalam dan paling kiri. Kekurangan dari algoritma *Depth First Search* yaitu kemungkinan tidak ditemukannya solusi yang diharapkan dan hanya menemukan satu solusi pada setiap pencarian [7][8][9].

III. METODE PENELITIAN

Dalam penelitian ini *Game Mummy Maze Deluxe* digunakan sebagai media untuk membandingkan dua algoritma karena permainan ini mempunyai tingkat kesulitan yang tinggi dalam mencari jalan keluar. Untuk itu penulis membuat sebuah aplikasi yang digunakan untuk membandingkan unsur kerja algoritma BFS dan DFS dari sisi jumlah langkah pencarian dan lama waktu penyelesaian proses pencarian.

A. Analisis Pencarian Solusi Menggunakan Algoritma *Breadth First Search*

Pencarian menggunakan algoritma *Breadth First Search* dimulai dari simpul 7 sebagai simpul awal menuju simpul 18 sebagai simpul tujuan. Dalam proses pencarian solusi aturan langkah *player* dan *zombie* mengikuti aturan *game mummy maze deluxe*, yaitu pada saat *player* bergerak satu langkah, maka *zombie* akan bergerak dua langkah ke arah yang sejajar dengan pergerakan dari *player*. *Zombie* melakukan langkah pertama ke kiri atau ke kanan mengikuti arah yang sejajar dari langkah pertama *player*. *Zombie* akan melakukan langkah maju mendekati pengguna apabila posisi *zombie* sudah sejajar dengan posisi *player*. Jika *player* bergerak sejajar dengan dinding yang membatasi jalan *zombie*, maka *zombie* tidak bisa melewati dinding tersebut. Gambar 1 merupakan pencarian menggunakan algoritma *Breadth First Search* pada level 1.



Gambar 1. Posisi player dan Zombie pada level 1 menggunakan Algoritma BFS

Cara kerja algoritma *Breadth First Search* dijelaskan sebagai berikut: langkah pertama adalah membuat antrean dan memasukan kepala antrean simpul awal kunjungan ke dalam antrean. Jika simpul awal bukan solusi masukkan semua simpul yang bertetangga ke dalam antrean sampai ditemukan solusi. Pada pencarian level 1, langkah pencarian dimulai dari atas, kanan, bawah, dan kiri. Simpul 7 merupakan kepala antrean dan dilakukan kunjungan awal pada simpul 1 dilanjutkan pada simpul yang bertetangga yaitu 8 dan 13 karena belum ditemukan solusi. Kemudian dilanjutkan kunjungan pada simpul lainnya sampai pada simpul 18 yang merupakan simpul tujuan, maka pencarian selesai karena solusi telah ditemukan.

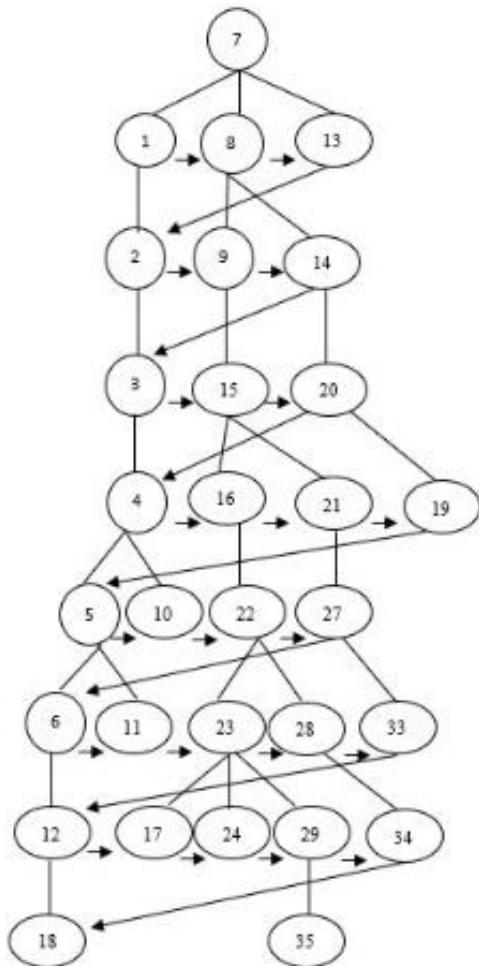
Proses pencarian terhadap semua simpul dari simpul awal menuju simpul tujuan menggunakan algoritma *Breadth First Search* pada *game mummy maze deluxe* level 1 dapat dilihat pada Tabel 1.

Tabel 1.

Pencarian BFS pada level 1 terhadap semua simpul

No.	Pencarian	No.	Pencarian
1	7,1	16	16,22
2	7,8	17	21,27
3	7,13	18	5,6
4	1,2	19	5,11
5	8,9	20	22,23
6	8,14	21	22,28
7	2,3	22	27,33
8	9,15	23	6,12
9	14,20	24	23,17
10	3,4	25	23,24
11	15,16	26	23,29
12	15,21	27	28,34
13	20,19	28	12,18
14	4,5	29	29,35
15	4,10		

Berdasarkan pencarian pada Tabel 1, dapat digambarkan pohon pencarian menggunakan algoritma BFS yang dapat dilihat pada Gambar 2.



Gambar 1. Pohon pencarian pada level 1 menggunakan Algoritma BFS

Jalur solusi level 1 yang diperoleh berdasarkan pohon pencarian gambar 1 adalah 7-1-8-13,2-9-14,3-15-20,4-16-21-19,5-10-22-27,6-11-23-28-33,12-17-24-29-34,18, dengan jumlah langkah 29 dan lama waktu pencarian dihitung menggunakan waktu dalam program aplikasi yang akan dibuat.

B. Analisis Pencarian Solusi Menggunakan Algoritma *Depth First Search*

Pencarian solusi menggunakan algoritma *Depth First Search* dilakukan secara mendalam yang dimulai dari simpul 7 sebagai simpul awal menuju simpul 18 sebagai simpul tujuan. Dalam proses pencarian menggunakan algoritma DFS terdapat aturan dari langkah *player* dan *zombie* yang diikuti dari *game mummy maze deluxe* adalah *player* bergerak satu langkah, maka *zombie* akan bergerak dua langkah ke arah yang sejajar dengan pergerakan dari *player*. *Zombie* melakukan langkah pertama ke kiri atau ke kanan mengikuti arah yang sejajar dari langkah pertama *player*. *Zombie* akan melakukan langkah maju mendekati pengguna apabila posisi *zombie* sudah sejajar dengan posisi *player*. Jika *player* bergerak sejajar dengan dinding yang membatasi jalan *zombie*, maka *zombie* tidak bisa

melewati dinding tersebut. Berikut ini merupakan pencarian menggunakan algoritma *Depth First Search* pada level 1 yang dapat dilihat pada Gambar 3.



Gambar 3. Posisi player dan Zombie pada level 1 menggunakan Algoritma DFS

Cara umum dari algoritma *Depth First Search* dijelaskan sebagai berikut: langkah pertama adalah membuat antrean dan kepala antrean sebagai simpul awal dimasukan ke dalam antrean yang pertama dikunjungi. Jika simpul awal bukan solusi dilanjutkan kunjungan ke simpul pertama yang berada pada tiap level sampai level yang paling dalam sehingga ditemukan solusi.

Pada pencarian level 1, langkah pencarian dimulai dari kiri, bawah, kanan, atas. Simpul 7 merupakan kepala antrean dan dilakukan kunjungan awal pada simpul 13 dilanjutkan pada simpul di level berikutnya yaitu pada simpul 19 karena belum ditemukan solusi. Kemudian dilanjutkan kunjungan simpul pada level yang paling dalam sampai ditemukan solusi pada simpul 18.

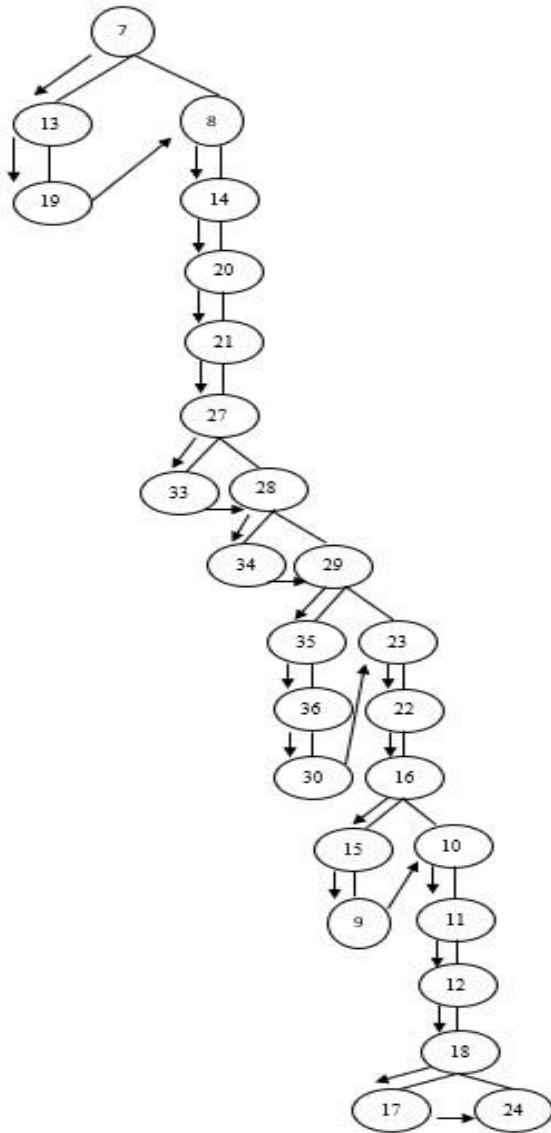
Berdasarkan Gambar 3 proses pencarian terhadap semua simpul dari simpul awal menuju simpul tujuan menggunakan algoritma *Depth First Search* pada *game mummy maze deluxe* level 1 dapat dilihat pada Tabel 2

Tabel 2.

Pencarian DFS pada level 1 terhadap semua simpul

No.	Pencarian	No.	Pencarian
1	7,7	14	35,36
2	7,13	15	36,30
3	13,19	16	29,23
4	7,8	17	23,22
5	8,14	18	22,16
6	14,20	19	16,15
7	20,21	20	15,9
8	21,27	21	16,10
9	27,33	22	10,11
10	27,28	23	11,12
11	28,34	24	12,18
12	28,29	25	18,17
13	29,35	26	18,24

Berdasarkan pencarian pada Tabel 2, dapat digambarkan pohon pencarian menggunakan algoritma DFS yang dapat dilihat pada Gambar 4.



Gambar 4. Pohon pencarian pada level 1 menggunakan Algoritma DFS

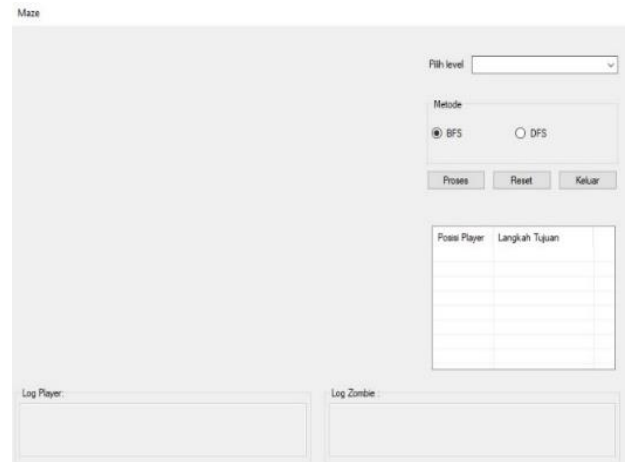
Jalur solusi pada *game mummy maze deluxe* level 1 yang diperoleh berdasarkan pohon pencarian gambar 1 adalah 7-13-19-8-14-20-21-27-33-28-34-29-35-36-30-23-22-16-15-9-10-11-12-18-17-24. Jumlah langkah untuk menemukan solusi sebanyak 26 langkah. Lama waktu pencarian akan dihitung menggunakan waktu pada aplikasi.

Pencarian solusi menggunakan algoritma *Breadth First Search* pada level 1 solusi diperoleh dengan jumlah langkah 29. Sedangkan pencarian solusi menggunakan algoritma *Depth First Search* pada level 1 solusi diperoleh dengan jumlah langkah 26.

IV. HASIL DAN PEMBAHASAN

A. Implementasi Aplikasi

Pada tahap ini akan dijelaskan antarmuka dari aplikasi Perbandingan Algoritma *Breadth First Search* dan *Depth First Search* Pada *Game Mummy Maze Deluxe*. Gambar 5 adalah halaman utama dari aplikasi.



Gambar 5. Tampilan Halaman Utama

Tampilan level 1 *game mummy maze deluxe* pada aplikasi dapat dilihat pada Gambar 6.



Gambar 6. Tampilan Level 1

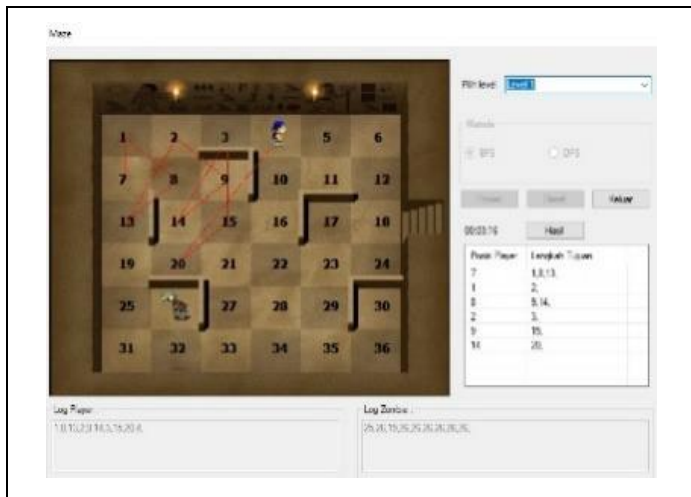
B. Pengujian Proses Pencarian

Hasil pengujian pada level 1 menggunakan metode BFS dapat dilihat pada Tabel 3.

Tabel 3.

Pengujian pada level 1 Metode BFS

No	Penjelasan	Hasil yang diharapkan	Hasil yang didapat
1.	Pengguna memilih level 1 dan metode BFS dan memulai proses pencarian.	Aplikasi akan melakukan pencarian pada level 1 dengan metode BFS.	Aplikasi melakukan pencarian pada level 1 dengan metode BFS.



4.	Melihat hasil proses pencarian dari metode BFS pada level 1.	Aplikasi akan menampilkan hasil proses pencarian dari metode BFS pada level 1.	Aplikasi menampilkan hasil proses pencarian dari metode BFS pada level 1.
----	--	--	---

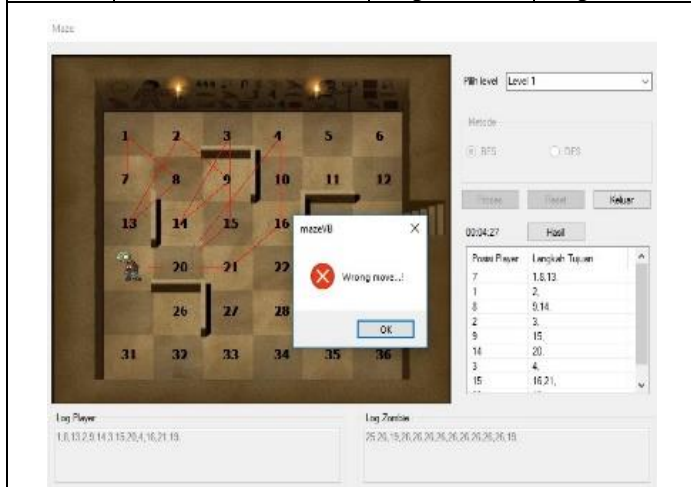
2.	Validasi apabila <i>player</i> melangkah terlalu dekat dengan zombie atau melakukan langkah mati.	Aplikasi akan menampilkan validasi apabila <i>player</i> melakukan langkah mati.	Aplikasi menampilkan validasi apabila <i>player</i> melakukan langkah mati.
----	---	--	---



Hasil pengujian pada level 1 menggunakan metode DFS dapat dilihat pada Tabel 4.

Tabel 4. Pengujian pada level 1 Metode DFS

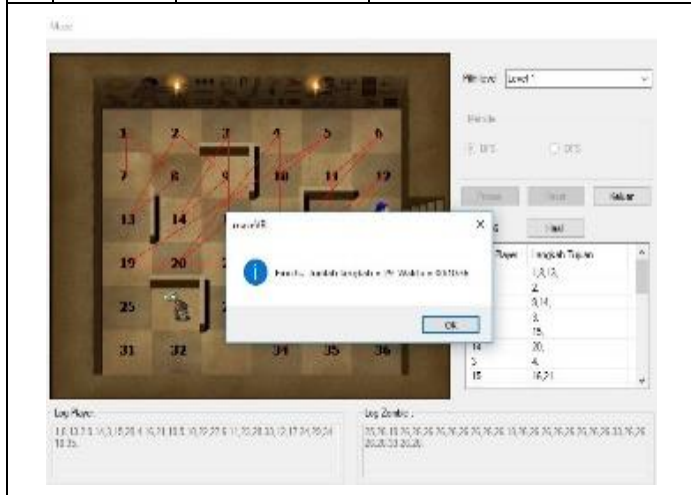
No	Penjelasan	Hasil yang diharapkan	Hasil yang didapat
1.	Pengguna memilih level 1 dan metode DFS dan memulai proses pencarian.	Aplikasi akan melakukan pencarian pada level 1 dengan metode DFS.	Aplikasi melakukan pencarian pada level 1 dengan metode DFS.

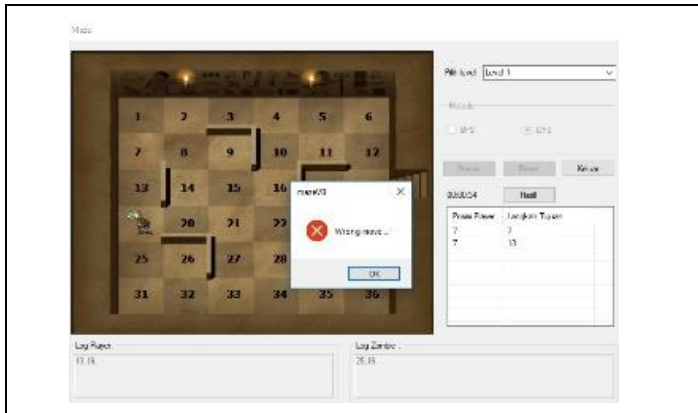


3.	Validasi apabila proses pencarian selesai.	Aplikasi akan menampilkan validasi apabila proses pencarian selesai.	Aplikasi menampilkan validasi apabila proses pencarian selesai.
----	--	--	---



2.	Validasi apabila <i>player</i> melangkah terlalu dekat dengan zombie atau melakukan langkah mati.	Aplikasi akan menampilkan validasi apabila <i>player</i> melakukan langkah mati.	Aplikasi menampilkan validasi apabila <i>player</i> melakukan langkah mati.
----	---	--	---

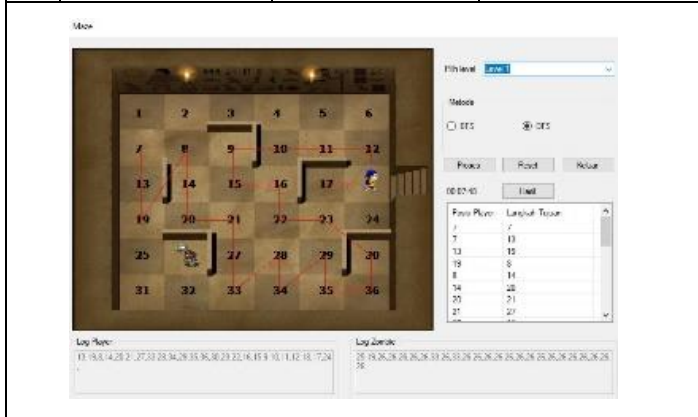




3.	Validasi apabila proses pencarian selesai.	Aplikasi akan menampilkan validasi apabila proses pencarian selesai.	Aplikasi menampilkan validasi apabila proses pencarian selesai.
----	--	--	---



4.	Melihat hasil proses pencarian dari metode DFS pada level 1.	Aplikasi akan menampilkan hasil proses pencarian dari metode DFS pada level 1.	Aplikasi menampilkan hasil proses pencarian dari metode DFS pada level 1.
----	--	--	---



2. Berdasarkan hasil pengujian yang dilakukan, pada level 1 diperoleh hasil bahwa algoritma DFS lebih cepat daripada algoritma BFS.
3. Posisi awal dari player sangat menentukan banyaknya jumlah langkah dan lama waktu pencarian pada kedua algoritma.

B. Saran

Saran untuk pengembangan selanjutnya, pada aplikasi dapat ditambahkan faktor lain dan level untuk perbandingan dari kedua algoritma.

DAFTAR PUSTAKA

- [1] D. A. Nugroho, H. and U. , "Membangun Game Edukasi "Mathematic Maze" Berbasis Android Untuk Meningkatkan Kemampuan Berhitung Pada Anak Sekolah Dasar," *Jurnal Statistika Industri dan Komputasi*, vol. II, no. 1, p. 70, 2017.
- [2] T. Kandaga and A. Hapendi, "Evaluasi dan Usaha Optimalisasi Algoritma Depth First Search dan Breadth First Search dengan Penerapan pada Aplikasi Rat Race dan Web Peta," *Jurnal Informatika*, vol. IV, no. 1, pp. 40-41, 2008.
- [3] S. Suryadi, "Perancangan Aplikasi Pencarian File Dengan Menggunakan Metode Best First Search," *Jurnal Informatika*, vol. II, no. 2, pp. 61-65, 2014.
- [4] B. Prasetyo and M. R. Hidayah, "Penggunaan Metode Depth First Search dan Breadth First Search pada Strategi Game Kamen Rider Decade Versi 0.3," *Scientific Journal of Informatics*, vol. I, no. 2, pp. 161-166, 2014.
- [5] S. Ardyan, M. and A. Suyitno, "Implementasi Algoritma Dijkstra Dalam Pencarian Rute Terpendek Tempat Wisata Di Kabupaten Gunung Kidul Dengan Program Visual Basic," *Unnes Journal of Mathematics*, vol. VI, no. 2, pp. 109-110, 2017.
- [6] S. Riyadi, "Implementasi Reminder Sms Gateway Untuk Pembayaran Jatuh Tempo Pada Koperasi Gadai," *Jurnal SPIRIT*, vol. IX, no. 1, p. 24, 2017.
- [7] O. Pribadi, "Maze Generator Dengan Menggunakan Algoritma Depth-First-Search," *Jurnal TIMES*, vol. IV, no. 1, pp. 1-5, 2015.
- [8] D. T. Yuwono *et al.*, "Perbandingan Algoritma Breadth First Search dan Depth First Search Sebagai Focused Crawler," *Annual Research Seminar 2016*, vol. II, no. 1, pp. 106-110, 2016.
- [9] I. Aprilia, "Analisis dan Penyelesaian Permainan River Crossing Ultimate Menggunakan Algoritma BFS dan DFS," *Jurnal*, vol. VI, no. 2, pp. 17-20, 2016.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

1. Algoritma *Breadth First Search* dan *Depth First Search* berhasil diterapkan dalam aplikasi melakukan perbandingan berdasarkan jumlah langkah dan lama waktu pencarian.