

IMPLEMENTASI ALGORITMA DEPTH LIMITED SEARCH PADA PERMAINAN PEG SOLITAIRE

Griffin Theresia R⁽¹⁾
griffintheresia@yahoo.com

Joko Purwadi⁽²⁾
jokop@ukdw.ac.id

Antonius Rachmat C.⁽³⁾
anton@ti.ukdw.ac.id

Abstraksi

Permainan Peg Solitaire adalah permainan single player yang terdiri dari sebuah papan dan sejumlah kelereng. Papan permainan Peg Solitaire terdiri dari banyak jenis antara lain papan jenis inggris, eropa, triangular dan masih banyak jenis papan permainan Peg Solitaire yang lain.

Pemain permainan Peg Solitaire terkadang sulit menentukan keputusan langkah yang tepat. Oleh karena itu, disediakan bantuan berupa hint yang membantu pemain saat pemain menentukan langkah. Salah satu algoritma yang dapat diterapkan pada hint permainan Peg Solitaire adalah algoritma Depth Limited Search.

Penerapan algoritma Depth Limited Search pada hint permainan Peg Solitaire di papan permainan versi inggris ukuran 3 x 3 dan triangular berukuran 4 x 4, 5 x 5, serta 7 x 7, mampu menemukan solusi yaitu sisa satu kelereng serta mampu menangani apabila tidak menemukan solusi. Penerapan algoritma Depth Limited Search pun mampu menampilkan semua perpindahan langkah hingga ditemukan sisa 1 kelereng. Hal ini dibuktikan dengan cara menguji 10 soal pada sistem. Dari hasil pengujian 10 soal pada sistem, 9 soal berhasil diselesaikan dan 1 soal gagal diselesaikan karena tidak menemukan solusi berupa sisa 1 kelereng.

Kata kunci : *Permainan (Game), Peg Solitaire, Depth Limited Search.*

1. Pendahuluan

Peg solitaire adalah salah satu permainan komputer. Permainan Peg Solitaire adalah permainan yang terdiri dari sebuah papan permainan dan sejumlah kelereng. Permainan Peg Solitaire ini memiliki tujuan untuk mencari solusi dari kondisi awal menuju

¹ Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana

² Teknik Informatika, Fakultas Teknologi Informasi Univeristas Kristen Duta Wacana

³ Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana

kondisi akhir yaitu sisa satu kelereng. Untuk menentukan solusi permainan *Peg Solitaire* akan digunakan sebuah algoritma.

Algoritma yang akan digunakan untuk menemukan solusi pada permasalahan permainan *Peg Solitaire* adalah algoritma *Depth Limited Search*. Algoritma *Depth Limited Search* adalah algoritma *blind search*. Algoritma ini diharapkan dapat menemukan solusi dari kondisi awal menuju kondisi akhir yaitu menyisakan satu kelereng pada papan permainan *Peg Solitaire* versi eropa dan inggris dengan ukuran masing – masing 3 x 3, 5 x 5, dan 7 x 7 serta versi *triangular* ukuran 4 x 4, 5 x 5, dan 7 x 7 serta mampu menangani kemungkinan solusi tidak ditemukan.

Batasan masalah dalam penelitian ini antara lain papan permainan untuk implementasi algoritma hanya papan versi inggris ukuran 3 x 3 dan *triangular* ukuran 4 x 4, 5 x 5, serta 7 x 7, ada pengembangan soal dan soal random untuk papan versi inggris inggris ukuran 3 x 3 serta *triangular* 4 x 4, 5 x 5, dan 7 x 7, pemain dapat memasukkan jumlah lubang kosong dengan batas maksimal setengah dari jumlah total lubang papan, dan algoritma akan diterapkan pada menu hint untuk membantu pemain memilih langkah untuk mencapai sisa satu kelereng.

2. Landasan Teori

2.1. Kecerdasan Buatan

Kecerdasan buatan merupakan cabang teknologi yang relatif baru. Prinsip kecerdasan buatan diawali sejak muncul komputer modern antara tahun 1940-1950. Kecerdasan buatan berkembang pesat seiring dengan berkembangnya teknologi komputer yang dari hari ke hari bertambah maju.

Ada 2 bagian utama yang dibutuhkan untuk aplikasi kecerdasan buatan yaitu basis pengetahuan atau *knowledge base* yang berisi fakta-fakta, teori, pemikiran serta hubungan antara satu dengan lainnya dan motor inferensi atau *inference engine* yaitu kemampuan menarik kesimpulan berdasarkan pengalaman. Kecerdasan buatan kini telah diterapkan pada banyak bidang.

Salah satu bidang yang menerapkan kecerdasan buatan adalah permainan atau *game*. *Game* berasal dari kata bahasa inggris yang memiliki arti dasar permainan. Permainan dalam hal ini merujuk pada pengertian kelincahan intelektual atau *intellectual playability*. *Game* atau permainan juga dapat dikelompokkan berdasarkan tipenya yaitu *deterministic game* dan *chance*. Permainan catur, Othello, dan *Peg Solitaire* termasuk dalam jenis permainan *deterministic-perfect information* karena memiliki informasi yang lengkap mengenai cara mencapai tujuan atau *goal*. Sedangkan jenis permainan *chance*

adalah permainan yang memiliki banyak kemungkinan – kemungkinan secara acak, belum dapat terlihat tujuan akhirnya.

2.2. Algoritma Depth Limited Search

Algoritma *Depth Limited Search* memberikan batas kedalaman pada algoritma *Depth First Search* (Russel & Norvig, 1995). Dengan algoritma *Depth Limited Search*, penelusuran *Depth First Search* data dibatasi sehingga tidak melakukan penelusuran terlalu dalam. Algoritma *Depth Limited Search* adalah sebagai berikut:

- a) Tetapkan *node* awal dengan kedalaman = 0 dan tentukan batas kedalaman.
- b) Cek apakah node adalah node tujuan. Jika benar maka proses berhenti, jika tidak maka lanjut ke langkah c.
- c) Cek apakah kedalaman node sama dengan batas kedalaman yang telah ditentukan. Jika benar, maka lanjutkan proses dengan menelusur hanya node – node yang berada dalam batas kedalaman yang telah ditentukan dan belum dikunjungi dengan kembali kelangkah b. Jika tidak maka lanjutkan ke langkah d.
- d) Perluas node dan kembali ke langkah b.

Gambaran kerja algoritma *Depth Limited Search* dapat digambarkan dalam bentuk *tree*. *Tree* merupakan sebuah graf tidak berarah dan merupakan jaringan bersambung yang tidak memiliki untai (loop) sehingga dengan demikian dapat disimpulkan bahwa sebuah *tree* dapat dibentuk dari graf sederhana karena graf sederhana tidak memiliki *self loop* ataupun *edge parallel*. *Tree* terdiri dari sekumpulan elemen. Elemen *tree* adalah akar atau *root* dan simpul. Derajat atau *degree* sebuah simpul menunjukkan jumlah anak pada simpul tersebut.

2.3. Permainan Peg Solitaire

Peg Solitaire merupakan sebuah permainan dengan satu orang pemain. Permainan ini terdiri dari sebuah papan permainan dengan sejumlah kelereng. Papan permainan tersebut berisi sekumpulan lubang. Lubang–lubang tersebut akan terisi penuh dengan kelereng. Pada awal permainan, lubang tengah pasti dikosongkan. *Peg Solitaire* ini dimainkan dengan cara :

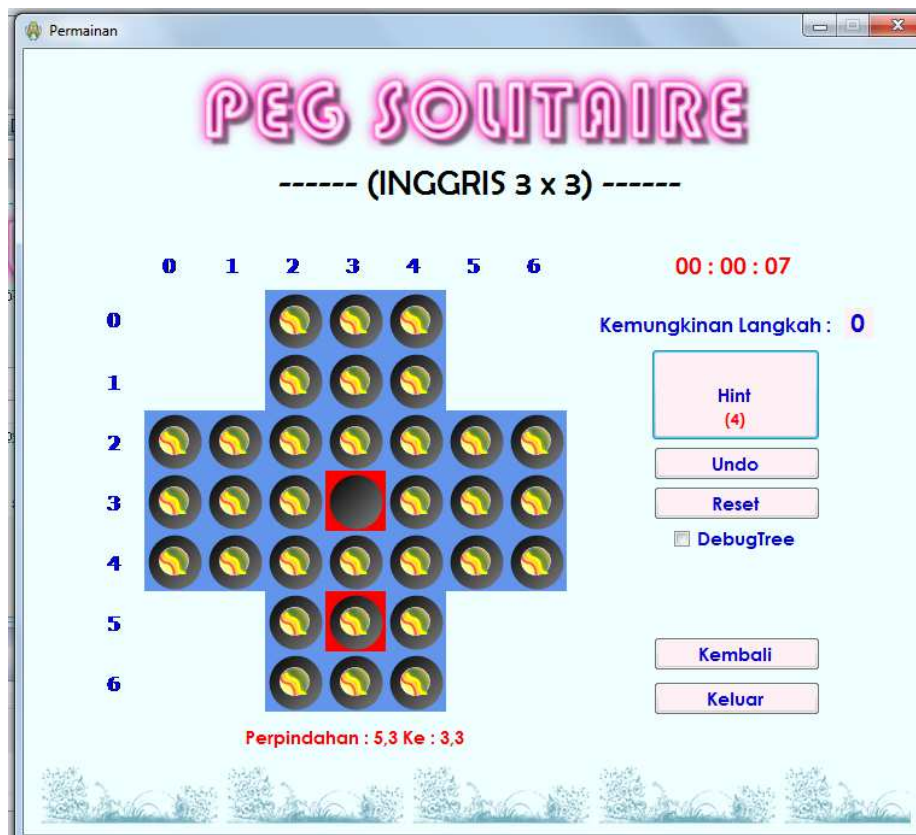
- Kelereng akan melompati kelereng lain menuju lubang kosong.
- Kelereng yang telah dilompati akan hilang atau keluar dari permainan.
- Permainan akan dianggap selesai apabila hanya tersisa satu kelereng.

Strategi yang terbaik yang dapat dilakukan agar dapat menemukan solusi pada permainan *Peg Solitaire* adalah mengarahkan kelereng ke lubang bagian tengah dan menyelamatkan kelereng terisolasi yang terdapat di lubang – lubang bagian pinggir.

Gerakan kelereng yang diperbolehkan adalah melangkah ke kanan, melangkah ke kiri, melangkah ke atas, dan ke bawah.

3. Hasil dan Pembahasan

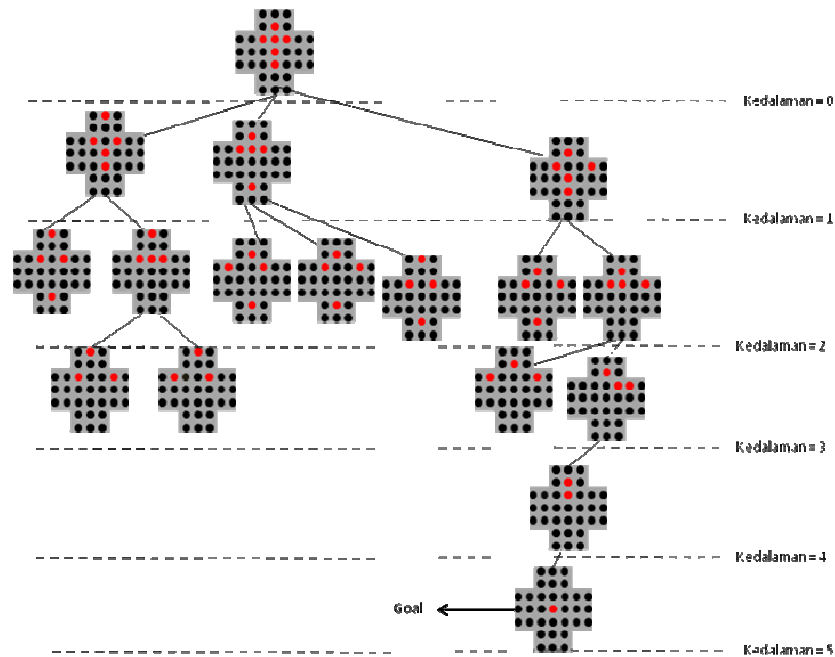
Sistem yang dibangun dari hasil penelitian ini menggunakan 8 form yang terdiri dari form utama, form jenis soal, form permainan untuk inggris dan eropa serta triangular, form tampil *tree* untuk inggris dan eropa serta triangular, from instruksi, dan form detail. Implementasi form permainan untuk papan versi inggris dapat dilihat pada gambar di bawah ini:



Gambar 1. Form Permainan

Struktur data yang dikembangkan antara lain struktur data untuk papan permainan, struktur data untuk bank soal, struktur data untuk soal sendiri, struktur data untuk fitur undo, struktur data untuk menentukan apakah papan berisi kelereng atau tidak, dan struktur data untuk penelusuran solusi permainan *Peg Solitaire* dengan algoritma *Depth Limited Search*. Penelusuran solusi dilakukan untuk menemukan jalur solusi menuju sisa 1 kelereng dengan algoritma *Depth Limited Search* dimana batas kedalamannya adalah jumlah kelereng - 1. Berikut ini adalah satu contoh kasus penyelesaian suatu kondisi pada

permainan *Peg Solitaire* versi inggris dengan ukuran 3 x 3 dengan menggunakan penelusuran pohon *Depth Limited Search*. Contoh penerapan ini menggunakan jenis soal *cross* atau salib yang terdiri dari 6 kelereng yang membentuk salib.



Gambar 2. Penerapan algoritma Depth Limited Search pada papan versi inggris dengan tipe soal salib

Contoh di atas adalah contoh kasus dengan menggunakan jenis soal salib dimana pengerjaannya dilakukan secara manual. Berikut ini adalah hasil dari contoh kasus serupa yang dikerjakan secara langsung oleh sistem:

LANGKAH MENUJU SOLUSI:
 Perpindahan : 2,3 Ke : 2,5
 Perpindahan : 4,3 Ke : 2,3
 Perpindahan : 2,2 Ke : 2,4
 Perpindahan : 2,5 Ke : 2,3
 Perpindahan : 2,3 Ke : 0,3

Gambar 3. Jalur solusi

Gambar di atas merupakan gambar jalur solusi menuju sisa 1 kelereng. Sedangkan pencarian jalur solusi dengan menyertakan semua kemungkinan langkah secara lengkap dapat dilihat pada gambar berikut.

MENCARI GOAL PADA KEDALAMAN: 5
 Perpindahan : 3,3 Ke : 5,3
 jumlah kelereng = 5
 Perpindahan : 2,3 Ke : 2,5
 jumlah kelereng = 4
 Perpindahan : 2,3 Ke : 2,1
 jumlah kelereng = 4
 Perpindahan : 2,3 Ke : 0,3
 jumlah kelereng = 4
 Perpindahan : 1,3 Ke : 3,3
 jumlah kelereng = 4
 Perpindahan : 2,3 Ke : 2,5
 jumlah kelereng = 5
 Perpindahan : 4,3 Ke : 2,3
 jumlah kelereng = 4
 Perpindahan : 2,3 Ke : 2,1
 jumlah kelereng = 3
 Perpindahan : 2,3 Ke : 0,3
 jumlah kelereng = 3
 Perpindahan : 2,2 Ke : 2,4
 jumlah kelereng = 3
 Perpindahan : 2,5 Ke : 2,3
 jumlah kelereng = 2
 Perpindahan : 2,3 Ke : 0,3
 jumlah kelereng = 1

Gambar 4. Hasil penelusuran algoritma Depth Limited Search

Pengujian sistem dari hasil penelitian ini dilakukan dengan cara menguji sistem menggunakan beberapa soal dari bank soal, soal random dan soal standar. Level kesulitan bergantung dari jumlah kelereng. Semakin banyak jumlah kelereng maka semakin sulit untuk diselesaikan. Berikut ini adalah daftar hasil penyelesaian beberapa soal dengan menggunakan algoritma *Depth Limited Search*.

Tabel 1. Daftar Soal yang akan diselesaikan dengan algoritma Depth Limited Search

| ID | Jenis Papan | Ukuran Papan | Jenis Soal | Jumlah Kelereng |
|-----|-------------|--------------|------------|-----------------|
| 1. | Inggris | 3 x 3 | Standar | 32 |
| 2. | Inggris | 3 x 3 | Bank Soal | 16 |
| 3. | Triangular | 4 x 4 | Standar | 9 |
| 4. | Triangular | 4 x 4 | Bank Soal | 8 |
| 5. | Triangular | 4 x 4 | Random | 8 |
| 6. | Triangular | 5 x 5 | Standar | 14 |
| 7. | Triangular | 5 x 5 | Bank Soal | 7 |
| 8. | Triangular | 5 x 5 | Random | 13 |
| 9. | Triangular | 7 x 7 | Standar | 27 |
| 10. | Triangular | 7 x 7 | Bank Soal | 14 |

Tabel 2. Hasil penyelesaian beberapa soal dengan menggunakan algoritma Depth Limited Search

| ID | Kedalaman Solusi | Jumlah Perpindahan | Jumlah Perpindahan Jalur Solusi | Apakah Berhasil Menemukan | Waktu |
|----|------------------|--------------------|---------------------------------|---------------------------|-------|
|----|------------------|--------------------|---------------------------------|---------------------------|-------|

| | | | | Solusi | |
|-----|----|-------|----|--------|------------|
| 1. | 31 | 20275 | 31 | √ | 1,08 menit |
| 2. | 15 | 2909 | 15 | √ | 9 detik |
| 3. | 8 | 76 | 8 | √ | 0,40 detik |
| 4. | 7 | 10 | 7 | √ | 0,40 detik |
| 5. | 7 | 53 | 7 | √ | 0,45 detik |
| 6. | 13 | 6530 | 13 | √ | 23 detik |
| 7. | - | - | - | X | - |
| 8. | 12 | 3926 | 12 | √ | 15 detik |
| 9. | 27 | 67907 | 27 | √ | 12 menit |
| 10. | 13 | 508 | 13 | √ | 6 detik |

Berdasarkan hasil pengujian di atas, tingkat kesulitan soal *Peg Solitaire* dapat digolongkan berdasarkan waktunya. Semakin lama waktu yang diperlukan sistem untuk menghasilkan pohon penelusuran dan jalur solusi untuk soal tersebut, maka soal tersebut dapat digolongkan sebagai soal yang cukup sulit. Soal digolongkan menjadi tiga level, yaitu level mudah, menengah, dan sulit. Level mudah dapat diselesaikan oleh sistem hanya dalam waktu kurang dari 1 detik. Sedangkan soal menengah dapat diselesaikan dalam waktu antara 1 sampai 59 detik. Soal sulit adalah soal yang membutuhkan waktu paling lama yaitu lebih dari satu menit. Penggolongan tingkat kesulitan soal- soal di atas dapat dilihat pada tabel berikut:

Tabel 3. Daftar Soal yang diselesaikan dengan algoritma Depth Limited Search

| ID | Jenis Papan | Ukuran Papan | Jenis Soal | Jumlah Kelereng | Tingkat Kesulitan |
|-----|-------------|--------------|------------|-----------------|-------------------|
| 1. | Inggris | 3 x 3 | Standar | 32 | Sulit |
| 2. | Inggris | 3 x 3 | Bank Soal | 16 | Menengah |
| 3. | Triangular | 4 x 4 | Standar | 9 | Mudah |
| 4. | Triangular | 4 x 4 | Bank Soal | 8 | Mudah |
| 5. | Triangular | 4 x 4 | Random | 8 | Mudah |
| 6. | Triangular | 5 x 5 | Standar | 14 | Menengah |
| 7. | Triangular | 5 x 5 | Bank Soal | 7 | Mudah |
| ID | Jenis Papan | Ukuran Papan | Jenis Soal | Jumlah Kelereng | Tingkat Kesulitan |
| 8. | Triangular | 5 x 5 | Random | 13 | Menengah |
| 9. | Triangular | 7 x 7 | Standar | 27 | Sulit |
| 10. | Triangular | 7 x 7 | Bank Soal | 14 | Menengah |

Berdasarkan pengujian sistem, kesulitan dari pemecahan solusi juga disebabkan oleh ukuran papan yang cukup besar sehingga membutuhkan waktu yang cukup lama untuk menelusur hingga ditemukan solusi. Selain itu, jumlah kelereng dan letak kelereng juga mempengaruhi lama atau tidaknya waktu penelusuran. Semakin acak posisi kelereng dan

berjauhan satu sama lain, maka akan semakin sulit dalam melakukan penelusuran. Posisi kelereng yang tersebar berjauhan di papan berukuran besar pun semakin mempersulit penelusuran. Berdasarkan hasil pengujian di atas, juga ditemukan salah satu soal yaitu soal dengan ID 7 yang tidak menemukan *goal* berupa sisa 1 kelereng. Hal ini disebabkan karena saat dilakukan penelusuran terhadap kondisi tersebut, tidak ada kemungkinan perpindahan kelereng yang menghasilkan jalur solusi menuju sisa 1 kelereng. Terakhir, berdasarkan pengujian keseluruhan, ada dua kemungkinan suatu kondisi tidak berhasil menemukan solusi. Kemungkinan pertama disebabkan oleh penelusuran terhadap kondisi tersebut yaitu tidak ada kemungkinan perpindahan kelereng yang menghasilkan jalur solusi menuju sisa 1 kelereng. Hal ini dapat dilihat pada bagian contoh soal yang tidak menemukan solusi pada gambar berikut.



Gambar 5. Soal yang tidak menemukan solusi

Sistem yang telah dibangun berdasarkan hasil penelitian ini memiliki kelebihan antara lain sistem mampu menyediakan bank soal yang terdiri dari soal – soal berekstensi .txt dan dapat digenerate ke papan permainan saat soal tersebut dipilih, sistem mampu menyediakan soal random dimana jumlah lubang kosong dapat dimasukkan oleh pemain, sistem mampu menemukan solusi dengan menerapkan algoritma *Depth Limited Search* baik pada fasilitas hint ataupun saat menggenerate pohon solusi dan sistem permainan *Peg Solitaire* memiliki tiga versi papan yaitu papan versi inggris, eropa dan *triangular*. Selain itu, sistem ini juga memiliki beberapa kelemahan yaitu sistem hanya mampu menyelesaikan papan versi inggris berukuran 3 x 3 dan papan versi *triangular* dengan ukuran 4 x 4, 5 x 5, dan 7 x 7, penggunaan algoritma *Depth Limited Search* murni dan metode iteratif

menyebabkan sistem butuh waktu yang lama untuk menemukan solusi untuk papan berukuran besar yaitu papan versi inggris ukuran 5 x 5 dan 7 x 7 serta papan versi eropa dengan ukuran 3 x 3, 5 x 5, dan 7 x 7, serta fasilitas *Hint* membutuhkan waktu yang cukup lama dalam menampilkan solusi apabila soal memiliki tingkat kesulitan yang tinggi.

4. Kesimpulan

- a. Algoritma *Depth Limited Search* yang diterapkan pada sistem permainan *Peg Solitaire* dalam tugas akhir ini mampu menemukan solusi namun terbatas hanya pada papan inggris ukuran 3 x 3, dan papan triangular dengan ukuran 4 x 4, 5 x 5, serta 7 x 7.
- b. Algoritma *Depth Limited Search* yang diterapkan oleh sistem ini membutuhkan waktu yang singkat untuk kasus – kasus kecil dengan jumlah kelereng kurang dari 16 dengan syarat posisi kelereng yang saling berdekatan satu sama lain atau dengan kata lain posisi kelereng tidak tersebar berjauhan namun membutuhkan waktu yang lama dalam kasus – kasus besar yaitu kasus – kasus dengan jumlah kelereng lebih dari 16 dan posisi kelereng yang tersebar berjauhan. Hal ini disebabkan oleh algoritma *Depth Limited Search* yang diterapkan pada sistem ini adalah algoritma *Depth Limited Search* murni dengan metode iteratif.
- c. Algoritma *Depth Limited Search* membatasi kedalaman yaitu jumlah kelereng – 1.
- d. Tanpa ada pembatasan level, solusi tetap dapat ditemukan.
- e. Algoritma *Depth Limited Search* mampu menangani tidak ditemukannya solusi dengan cara mendeteksi semua posisi kelereng sehingga apabila posisi kelereng tidak memungkinkan lagi untuk dipindahkan ke posisi lain maka sistem akan berhenti bekerja.
- f. Pengujian sistem yang dilakukan dengan menguji 10 soal, 9 soal berhasil diselesaikan dan 1 soal tidak berhasil diselesaikan.

Daftar Pustaka

- , *Games*. Diakses 7 April 2011, dari <http://www.Cs.rpi.edu/holling/ai/notes/games/game.pdf>.
- , *Peg Solitaire*. Diakses 6 April 2011, dari http://tlpuzzle.weebly.com/uploads/1/0/0/3/1003384/solusi_cara_menyelesaikan_peg_solitaire_games.pdf.
- Abikoeno, S.Y. (2009). Implementasi *Algoritma Iterative Deepening Depth First Search* dalam Permainan Labirin. Tugas Akhir. Diakses 15 Mei 2011, dari <http://sinta.ukdw.ac.id/sinta/resources/sintasrv/nim/22043600>
- Bell, George. (2008). *Solving Triangular Peg Solitaire, Journal of Integer Sequences*. New York, Vol 11.
- E.Frenzel Jr, Louis. (1997). *Crash Course in Artificial Intelligence and Expert System*. Howard W.Sams & Co a Division of Macmillan Inc.
- H. A. Simon. 1987. Diambil dari buku Kusrini yang berjudul Sistem Pakar

- Teori dan Aplikasi. Andi Yogyakarta : Yogyakarta.
- Kusumadewi, Sri. (2003). AI (Teknik dan Aplikasinya) Cetakan Pertama. Yogyakarta: Penerbit Graha Ilmu.
- Limanto dan Widiasri. (2010). Modifikasi Metode *Backtracking* untuk Membantu Mencari Penyelesaian Permainan *Peg Solitaire*. Konferensi Nasional Sistem dan Informatika 13 November 2010.
- Nalwan, Agustinus. (1995). Pemrograman Animasi dan Game Profesional. Jakarta: PT. Elex Media Komputindo.
- Pertiwi et al. (2006). Penerapan Algoritma BFS, DFS, DLS dan IDS dalam Pencarian Solusi *Water Jug Problem*.
- Russel, S., & Norvig, P. (1995). *Artificial Intelligence A Modern Approach*. New Jersey :Prentice Hall Inc.
- Stickel dan Tyson. (1985). *An Analysis of Consecutively Bounded Depth First Search with Applications in Automated Deduction*. Proceeding, Artificial Intelligence Center SRI International.
- Sunaryo, A.R. (2006). Algoritma pencarian simpul solusi dalam graf. Penelitian, Institut Teknologi Bandung. Diakses 16 Mei 2011, dari <http://www.informatika.org/~rinaldi/Matdis/2007-2008/Makalah/MakalahIF2153-0708-042.pdf>.
- Turban, E., E. McLean, dan J. Wetherbe. Information Technology for Management. New York: John Wiley & Sons, Inc., 1999.
- Udupa et al. (2011). *Depth Bounded Explicit State Model Checking*. Microsoft Research India, University of Pennsylvania.
- Winston dan Prendergast. (1984). *The AI Business : The Commercial Uses Of Artificial Intelligence*. MIT Press.