

SIMULASI PERGERAKAN *CHESS KNIGHT* DALAM PAPAN CATUR

Dini MH. Hutagalung

Program Studi Sistem Informasi Universitas Sari Mutiara Indonesia

mhdini@gmail.com

ABSTRAK

Sistem produksi (*production system*) merupakan salah satu bentuk representasi pengetahuan yang sangat populer dan banyak digunakan. Representasi pengetahuan dengan sistem produksi pada dasarnya berupa aplikasi aturan (*rule*) yang berupa *antecedent*, yaitu bagian yang mengekspresikan situasi atau premis dan konsekuen, yaitu bagian yang menyatakan suatu tindakan tertentu atau konklusi yang diterapkan jika suatu situasi atau premis bernilai benar. Setelah menyelesaikan perangkat lunak simulasi pergerakan legal dari *chess knight* dalam papan catur, Perangkat lunak menggunakan metode pencarian *breadth first search* (BFS). Karena itu, pencarian akan menemukan semua solusi terpendek. Perangkat lunak merupakan implementasi (penerapan) nyata pohon pelacakan dalam memecahkan suatu permasalahan berdasarkan konsep *Artificial Intelligence* (AI). Perangkat lunak mensimulasikan semua gerakan dari semua solusi yang ditemukan, sehingga memberikan gambaran yang cukup jelas atas solusi-solusi yang dihasilkan.

Kata Kunci : *chess knight*, papan catur, *breadth first search* (BFS).

I. PENDAHULUAN

1.1 Latar Belakang

Sistem produksi (*production system*) merupakan salah satu bentuk representasi pengetahuan yang sangat populer dan banyak digunakan. Representasi pengetahuan dengan sistem produksi pada dasarnya berupa aplikasi aturan (*rule*) yang berupa *antecedent*, yaitu bagian yang mengekspresikan situasi atau premis dan konsekuen, yaitu bagian yang menyatakan suatu tindakan tertentu atau konklusi yang diterapkan jika suatu situasi atau premis bernilai benar. Salah satu permasalahan yang dapat dijelaskan dengan menggunakan sistem produksi adalah simulasi pergerakan langkah kuda dalam papan catur.

Biji kuda dalam papan catur memiliki pergerakan menyerupai huruf L. Biji catur ini merupakan salah satu biji yang sangat sulit

digerakkan dan sering juga merupakan biji yang paling berbahaya apabila tidak diperhatikan secara seksama setiap pergerakannya. Simulasi dari permasalahan ini menyediakan sebuah papan catur berukuran $n \times n$. Sasaran (*goal*) dari permasalahan ini adalah menggerakkan sebuah biji kuda dari suatu posisi tertentu pada papan catur ke posisi tujuan yang diinginkan dengan mensimulasikan semua solusi pergerakan terpendek yang mungkin untuk menuju ke posisi tujuan tersebut. Permasalahan ini juga merupakan salah satu masalah klasik dalam *artificial intelligence* (AI). Penyelesaian permasalahan ini dapat menggunakan bantuan sistem produksi dan pohon pelacakan.

Berdasarkan uraian di atas, penulis ingin merancang suatu perangkat lunak yang mampu mencari semua solusi pergerakan terpendek

sebuah biji kuda dari posisi awal menuju posisi tujuan dan menampilkan simulasi pergerakan dari kuda dalam papan catur.

1.2 Rumusan Masalah

Permasalahan pergerakan legal dari biji kuda dapat dirumuskan sebagai berikut:

- a. Komputer mencari dan menampilkan semua solusi pergerakan terpendek yang dapat dilewati biji kuda dari posisi awal ke posisi tujuan.
- b. Komputer dapat mensimulasikan semua solusi pergerakan yang dihasilkan.

1.3 Batasan Masalah

Karena keterbatasan waktu dan pengetahuan penulis, maka ruang lingkup permasalahan dalam merancang perangkat lunak ini antara lain :

1. Ukuran dari papan catur dibatasi maksimal 8 x 8 dan minimal 3 x 3.
2. Komponen-komponen yang terdapat pada perangkat lunak, yaitu:
 - a. Kuda (Putih), sebagai simbol dari posisi awal.
 - b. Raja (Hitam), sebagai simbol dari posisi tujuan.
 - c. Bidak (Putih), sebagai simbol dari rintangan yang tidak boleh ditempati. Bidak / rintangan dibatasi maksimal sebanyak 8 buah.

3. Metode pencarian yang digunakan adalah dengan menggunakan metode pencarian melebar pertama (*breadth-first search*).
4. Pencarian hanya akan menemukan semua solusi pergerakan terpendek.
5. Jika tidak terdapat solusi, maka perangkat lunak akan menampilkan pesan kesalahan (*error message*).
6. Perangkat lunak akan mensimulasikan pergerakan biji kuda menuju posisi tujuan sesuai dengan solusi-solusi yang telah dihasilkan.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah untuk merancang suatu perangkat lunak yang mampu mencari semua solusi pergerakan terpendek untuk menuju posisi tujuan dengan menggunakan sistem produksi dan pohon pelacakan.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini, yaitu untuk membantu pemahaman sistem produksi (*production systems*) dan sebagai implementasi pohon pelacakan dalam mencari semua solusi pergerakan terpendek. Perangkat lunak juga dapat digunakan sebagai fasilitas pendukung dalam proses belajar mengajar, khususnya mata kuliah *Artificial Intelligence* (AI).

II. LANDASAN TEORI

2.1 Sejarah AI

Kecerdasan buatan atau AI termasuk bidang ilmu yang relatif muda. Pada tahun 1950-

an para ilmuwan dan peneliti mulai memikirkan bagaimana caranya agar mesin dapat melakukan pekerjaannya seperti yang bisa dikerjakan oleh manusia. Alan Turing, seorang matematikawan dari Inggris pertama kali mengusulkan adanya tes untuk melihat bisa tidaknya sebuah mesin dikatakan cerdas. Hasil tes tersebut kemudian dikenal dengan Turing Test, dimana si mesin tersebut menyamar seolah-oleh sebagai seseorang di dalam suatu permainan yang mampu memberikan respon terhadap serangkaian pertanyaan yang diajukan. Turing beranggapan bahwa, jika mesin dapat membuat seseorang percaya bahwa dirinya mampu berkomunikasi dengan orang lain, maka dapat dikatakan bahwa mesin tersebut cerdas, seperti layaknya manusia. (3)

AI itu sendiri dimunculkan oleh seorang profesor dari Massachusetts Institute of Technology yang bernama John McCarthy pada tahun 1956 pada Darmouth Conference yang dihadiri oleh para peneliti AI. Pada konferensi tersebut juga didefinisikan tujuan utama dari kecerdasan buatan, yaitu mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan kelakuan mesin tersebut.

Beberapa program AI yang mulai dibuat pada tahun 1956-1966, antara lain:

1. Logic Theorist, diperkenalkan pada Dartmouth Conference, program ini dapat

membuktikan teorema-teorema matematika.

2. Sad Sam, diprogram oleh Robert K. Lindsay (1960). Program ini dapat mengetahui kalimat-kalimat sederhana yang ditulis dalam bahasa Inggris dan mampu memberikan jawaban dari fakta-fakta yang didengar dalam sebuah percakapan.
3. ELIZA, diprogram oleh Joseph Weizenbaum (1967). Program ini mampu melakukan terapi terhadap pasien dengan memberikan beberapa pertanyaan dan jawaban. (3)

2.1.1 Definisi AI

Kecerdasan buatan atau AI adalah salah satu bagian ilmu komputer yang mempelajari tentang bagaimana cara membuat agar komputer dapat melakukan pekerjaan seperti yang dilakukan oleh manusia. Tujuan dari AI adalah untuk memecahkan persoalan dunia nyata (bersifat praktis) dan memahami inteligensia (bersifat memahami). (8)

AI merupakan salah satu bagian ilmu komputer yang mempelajari tentang bagaimana cara membuat agar komputer dapat melakukan pekerjaan seperti yang dilakukan oleh manusia. Pada awal diciptakannya, komputer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan zaman, maka peran komputer semakin mendominasi kehidupan umat manusia. Komputer tidak lagi hanya digunakan

sebagai alat hitung, lebih dari itu, komputer diharapkan untuk dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia.(3)

Manusia bisa menjadi pandai dalam menyelesaikan segala permasalahan di dunia ini karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki oleh seseorang tentu saja diharapkan akan lebih mampu dalam menyelesaikan permasalahan. Namun bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang mereka miliki. Tanpa memiliki kemampuan menalar yang baik, manusia dengan segudang pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik. Demikian pula, dengan kemampuan menalar yang sangat baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik.

Agar komputer bisa bertindak seperti dan sebaik manusia, maka komputer juga harus diberi bekal pengetahuan dan mempunyai kemampuan untuk menalar. Untuk itu pada *artificial intelligence*, akan mencoba untuk memberikan beberapa metode untuk membekali komputer dengan kedua komponen tersebut agar komputer bisa menjadi mesin yang pintar.

2.2 Masalah Ruang Keadaan (State Space Problem)

State merupakan representasi suatu keadaan pada suatu saat ataupun deskripsi konfigurasi sistem. *State space* adalah semua state yang mungkin, dan biasanya digambarkan sebagai jaringan dengan *verteks* merupakan *state* dan *edge* merupakan perubahan yang mungkin.

Deskripsi formal suatu masalah :

- Definisikan *state space*.
- Tentukan *initial state* dalam *goal state*.
- Tentukan operator-operator.

Representasi *state space* memungkinkan definisi formal suatu masalah sebagai persoalan mengubah status dengan menggunakan sekumpulan operator (*rule*) dan juga mendefinisikan masalah sebagai *search* yaitu mencari lintasan di dalam *state space* dari *initial state* ke *goal state*. Contohnya : *8-puzzle* dan permasalahan langkah kuda dalam papan catur. (3)

2.2.1 8-Puzzle Problem

Permasalahan *8-puzzle*, sebenarnya diangkat dari permainan *puzzle* anak-anak. Dalam permainan ini, terdapat 8 buah ubin yang memiliki angka dari 1 sampai 8. Pada tempat yang disediakan, tersedia sebuah tempat kosong untuk menggerakkan ubin. Ubin kemudian diacak dan pemain dituntut untuk mengurutkan kembali angka yang telah teracak. Permasalahan ini termasuk permasalahan ruang keadaan (*state space problem*), karena memiliki keadaan awal (*start state*), aturan main (*rule*), dan keadaan

tujuan (*initial state*). Adapun aturan-aturan yang dapat digunakan untuk menggerakkan dari satu keadaan ke keadaan yang baru adalah:

1. Ubin kosong dapat digeser ke kiri, apabila ubin kosong tidak berada pada kolom ke-1.
2. Ubin kosong dapat digeser ke kanan, apabila ubin kosong tidak berada pada kolom ke-3.
3. Ubin kosong dapat digeser ke atas, apabila ubin kosong tidak berada pada baris ke-1.
4. Ubin kosong dapat digeser ke bawah, apabila ubin kosong tidak berada pada baris ke-3.

Permasalahan 8-*puzzle* dapat diselesaikan dengan menggunakan bantuan pohon pelacakan. *Node* pertama merupakan kondisi awal, selanjutnya dibuka *node-node* anak (cabang) yang merupakan keadaan baru (hasil pergerakan yang mungkin sesuai dengan aturan yang ada). Ini dilakukan terus hingga pohon pelacakan mendapatkan kondisi tujuan (*initial state*).

III.HASIL DAN PEMBAHASAN

3.1 Algoritma

Algoritma perancangan perangkat lunak simulasi langkah kuda dalam papan catur dapat dibagi menjadi 2 (dua) bagian, yaitu :

1. Algoritma pencarian solusi.
2. Algoritma fungsi pendukung.

3.1.1 Algoritma Pencarian Solusi

Seperti telah dijelaskan pada bab 3 bahwa pencarian solusi menggunakan bantuan pohon pelacakan dengan metode *Breadth-First Search*

(BFS). Algoritma pencarian BFS akan melakukan penelusuran *node* per level, artinya semua *node* pada level yang sama akan dikunjungi terlebih dahulu sebelum penelusuran dilanjutkan ke *node* pada level berikutnya. Karena bisa terdapat lebih dari satu solusi, maka penelusuran atau pengembangan *node* dilakukan hingga tidak ada *node* yang dapat dikembangkan lagi. *Node* yang berisi posisi tujuan merupakan solusi.

Terdapat 8 buah operasi untuk mengembangkan setiap *node* dalam pohon pelacakan, yaitu:

1. Geser biji kuda ke atas sebelah kiri (Baris = baris + 2, Kolom = kolom - 1).
2. Geser biji kuda ke atas sebelah kanan (Baris = baris + 2, Kolom = kolom + 1).
3. Geser biji kuda ke kiri sebelah atas (Baris = baris + 1, Kolom = kolom - 2).
4. Geser biji kuda ke kiri sebelah bawah (Baris = baris - 1, Kolom = kolom - 2).
5. Geser kuda ke kanan sebelah atas (Baris = baris + 1, Kolom = kolom + 2).
6. Geser kuda ke kanan sebelah bawah (Baris = baris - 1, Kolom = kolom + 2).
7. Geser kuda ke bawah sebelah kiri (Baris = baris - 2, Kolom = kolom - 1).
8. Geser kuda ke bawah sebelah kanan (Baris = baris - 2, Kolom = kolom + 1).

Operasi ini harus mematuhi aturan sebagai berikut:

1. Posisi setelah pergeseran tidak boleh berada di luar papan catur.

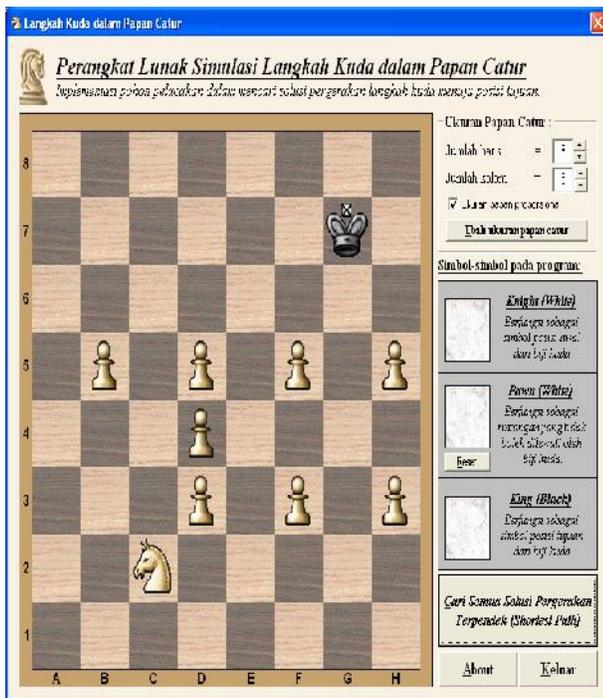
2. Posisi setelah pergeseran tidak boleh sama dengan posisi biji bidak (rintangan).
3. Posisi setelah pergeseran belum pernah dilintasi sebelumnya (pada satu jalur yang sama dalam pohon pelacakan). Ini menjamin bahwa biji kuda tidak akan menempati posisi yang sudah pernah ditempati sebelumnya.

3.2 Pengujian Program

Sebagai contoh, *input* data adalah sebagai berikut.

1. Ukuran papan catur = 8 x 8.
2. Posisi biji kuda = C2.
3. Posisi biji raja = G7.
4. Posisi biji bidak = B5, D3, D4, D5, F3, F5, H3 dan H5.

Tampilan *form input* untuk contoh kasus seperti terlihat pada gambar 4.2 berikut.



Gambar 3.1 *Form Input* (contoh kasus 1)



Gambar 3.2 *Form Solusi* (contoh kasus 1)

Solusi yang didapatkan seperti terlihat pada gambar 4.3 dan tabel 4.1 berikut.

Tabel 4.1 Solusi yang didapatkan (contoh kasus 1)

Solusi	Langkah-Langkah	Panjang Solusi
Solusi-1	B4, A6, C5, E6, G7	5 langkah
Solusi-2	B4, C6, D8, E6, G7	5 langkah
Solusi-3	B4, C6, D4, E6, G7	5 langkah
Solusi-4	A3, C4, D6, E8, G7	5 langkah
Solusi-5	A1, B3, C5, E6, G7	5 langkah
Solusi-6	A1, B3, D4, E6, G7	5 langkah
Solusi-7	E3, C4, D6, E8, G7	5 langkah
Solusi-8	E3, G4, F6, E8, G7	5 langkah
Solusi-9	E3, G2, F4, E6, G7	5 langkah
Solusi-10	E1, G2, F4, E6, G7	5 langkah

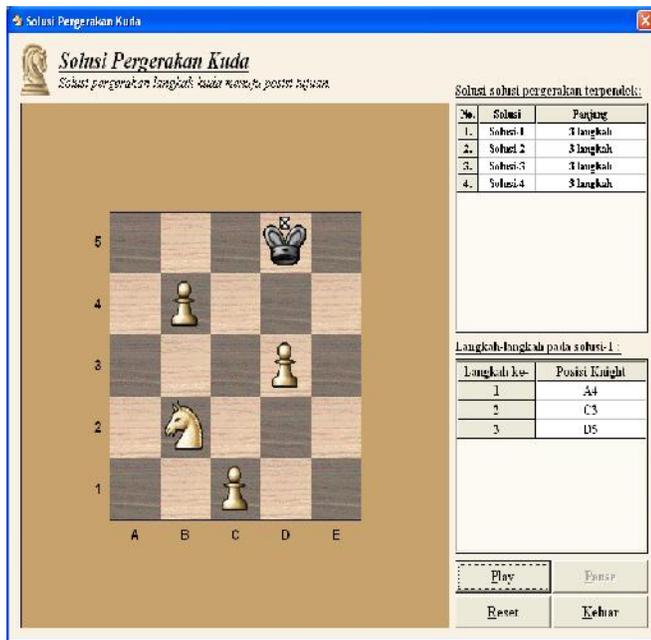
Contoh lain, *input* data adalah sebagai berikut.

1. Ukuran papan catur = 8 x 8.
2. Posisi biji kuda = B2.
3. Posisi biji raja = D5.
4. Posisi biji bidak = C1, D3 dan B4.

Tampilan *form input* untuk contoh kasus seperti terlihat pada gambar 4.4 berikut.



Gambar 3.3 *Form Input* (contoh kasus 2)



Gambar 3.4 *Form Solusi* (contoh kasus 2)

Solusi yang didapatkan seperti terlihat pada gambar 4.5 dan tabel 4.2 berikut.

Tabel 4.2 Solusi yang didapatkan (contoh kasus 2)

Solusi	Langkah-Langkah	Panjang Solusi
Solusi-1	A4, C3, D5	3 langkah
Solusi-2	C4, E3, D5	3 langkah
Solusi-3	D1, C3, D5	3 langkah
Solusi-4	D1, E3, D5	3 langkah

IV. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Setelah menyelesaikan perangkat lunak simulasi pergerakan legal dari *chess knight* dalam papan catur, penulis menarik kesimpulan sebagai berikut:

1. Perangkat lunak menggunakan metode pencarian *breadth first search* (BFS). Karena itu, pencarian akan menemukan semua solusi terpendek.
2. Perangkat lunak merupakan implementasi (penerapan) nyata pohon pelacakan dalam memecahkan suatu permasalahan berdasarkan konsep *Artificial Intelligence* (AI).
3. Perangkat lunak mensimulasikan semua gerakan dari semua solusi yang ditemukan, sehingga memberikan gambaran yang cukup jelas atas solusi-solusi yang dihasilkan.

4.2 Saran

Penulis ingin memberikan beberapa saran yang mungkin dapat membantu dalam pengembangan perangkat lunak ini yaitu:

1. Perangkat lunak dapat dikembangkan dengan menambahkan beberapa metode pencarian lain yang terdapat di dalam ruang lingkup *Artificial Intelligence* (AI).
2. Perangkat lunak dapat dikembangkan dengan menambahkan animasi sewaktu biji kuda digerakkan. Untuk animasi yang lebih baik, perangkat lunak dapat dibangun dengan menggunakan aplikasi *Macromedia Flash*.
3. Perangkat lunak dapat dikembangkan dengan menambahkan biji catur yang lain.

DAFTAR PUSTAKA

Arhani.M, Konsep Dasar Sistem Pakar, Penerbit Andi, Yogyakarta, 2005.

Desiani.A dan Arhami.M, Konsep Kecerdasan Buatan, Penerbit Graha Ilmu, 2002.

Kusumadewi.S, Artificial Intelligence (Teknik dan Aplikasinya), Edisi 2, Penerbit Graha Ilmu, 2002.

Munir.R, *Matematika Diskrit*, Informatika Bandung, 2005.

Munir.R, Lidia.L, Algoritma dan Pemrograman, Edisi Kedua, 2002.

Pramono.D, Mudah menguasai *Visual Basic 6*, PT. Elex Media Komputindo, 2002.

Ramadhan.A , MS. *Visual Basic 6*, PT. Elex Media Komputindo, Jakarta, 2004.

Sandi.S, *Artificial Intelegencia*, Andi Offset Yogyakarta, 1993.

Supardi.Y, Ir, *Microsoft Visual Basic 6.0 Untuk Segala Tingkat*, PT. Elex Media Komputindo, Jakarta, 2006.