

Perangkat Lunak Simulasi Langkah Kuda Dalam Permainan Catur

Hartono¹⁾ Liva Junter²⁾

STMIK IBBI Medan

Jl. Sei Deli No. 18 Medan, Telp. 061-4567111 Fax. 061-4527548

Email: hartonoibbi@gmail.com¹

Abstrak

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.

Oleh karena itu, dirancang suatu aplikasi simulasi dengan memanfaatkan beberapa teknik-teknik pemrograman simulasi dengan menggunakan bahasa *Visual Basic 6.0*. Tujuan dari 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.

Hasil dari tulisan ini adalah 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.

Kata Kunci: Simulasi, Perangkat Lunak, Pelacakan

Abstract

Horse seed in the chess board movement resembles the letter L. The chess pieces are one of a very hard-driven beans and seeds are often also the most dangerous if not carefully considered every movement. Simulation of this problem provides a chess board size $n \times n$. Target (goal) of this problem is to move a horse beans of a certain position on a chess board position to the desired destination with the shortest movement simulates all possible solutions to get to the goal position. This problem is also one of the classic problems in artificial intelligence (AI). Settlement of this problem can use the help system and tree production tracking.

Therefore, designed a simulation applications by utilizing several techniques of simulation programming using Visual Basic 6.0. The purpose of this study is to design a software that is able to find all the solutions for the shortest movement toward the goal position by using the system of production and tracking tree.

Results from this paper is that the software is able to find all solutions shortest movement a horse beans from the initial position to the goal position and displays the simulation of the movement of the horse in the chess board.

Keywords: Simulation, Software, Tracking

1. Pendahuluan

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.

Disebabkan luasnya permasalahan yang ada, maka peneliti memberikan batasan masalah 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 dan optimum.
 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.

2. Metode Penelitian

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.

Permasalahan ini merupakan salah satu masalah klasik dalam *artificial intelligence* (AI). Sebuah biji kuda dalam papan catur memiliki pergerakan menyerupai huruf L. Biji catur ini merupakan salah satu biji yang paling sulit digerakkan dan sering juga merupakan biji yang paling berbahaya apabila tidak diperhatikan secara seksama setiap pergerakannya. 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 pergerakan terpendek menuju ke posisi tujuan tanpa melanggar aturan-aturan yang telah ditetapkan. Penyelesaian permasalahan ini dapat menggunakan bantuan pohon pelacakan.

Gambar 1. Papan Catur Berukuran 8 x 8

2.1. Model

Dengan memperhatikan gambar 1, maka aturan-aturan yang terdapat pada permasalahan langkah kuda dalam papan catur adalah:

1. Kuda dapat digeser ke atas sebelah kiri.
Operasi: Baris = baris + 2, Kolom = kolom - 1.
Aturan: Nilai variabel kolom setelah operasi harus > 0 , nilai variabel baris ≤ 8 , dan tidak terdapat biji catur lain pada posisi tersebut.
2. Kuda dapat digeser ke atas sebelah kanan.
Operasi: Baris = baris + 2, Kolom = kolom + 1.
Aturan: Nilai variabel kolom dan baris setelah operasi harus ≤ 8 dan tidak terdapat biji catur lain pada posisi tersebut.
3. Kuda dapat digeser ke kiri sebelah atas.
Operasi: Baris = baris + 1, Kolom = kolom - 2.
Aturan: Nilai variabel kolom setelah operasi harus > 0 , nilai variabel baris ≤ 8 , dan tidak terdapat biji catur lain pada posisi tersebut.
4. Kuda dapat digeser ke kiri sebelah bawah.
Operasi: Baris = baris - 1, Kolom = kolom - 2.
Aturan: Nilai variabel kolom dan baris setelah operasi harus > 0 dan tidak terdapat biji catur lain pada posisi tersebut.
5. Kuda dapat digeser ke kanan sebelah atas.
Operasi: Baris = baris + 1, Kolom = kolom + 2.
Aturan: Nilai variabel kolom dan baris setelah operasi harus ≤ 8 dan tidak terdapat biji catur lain pada posisi tersebut.
6. Kuda dapat digeser ke kanan sebelah bawah.
Operasi: Baris = baris - 1, Kolom = kolom + 2.
Aturan: Nilai variabel kolom setelah operasi harus ≤ 8 , nilai variabel baris harus > 0 , dan tidak terdapat biji catur lain pada posisi tersebut.
7. Kuda dapat digeser ke bawah sebelah kiri.
Operasi: Baris = baris - 2, Kolom = kolom - 1.
Aturan: Nilai variabel kolom dan baris setelah operasi harus > 0 dan tidak terdapat biji catur lain pada posisi tersebut.
8. Kuda dapat digeser ke bawah sebelah kanan.
Operasi: Baris = baris - 2, Kolom = kolom + 1.
Aturan: Nilai variabel kolom setelah operasi harus ≤ 8 , nilai variabel baris harus > 0 , dan tidak terdapat biji catur lain pada posisi tersebut.

2.2. Analisis

Hal terpenting dalam menentukan keberhasilan sistem yang berlandaskan AI adalah kesuksesan dalam pencarian dan pencocokan. Pada dasarnya ada 2 teknik pencarian dan pelacakan yang digunakan, yaitu pencarian buta (*blind search*) dan pencarian terbimbing (*heuristic search*).

2.2.1 Pencarian Buta (*Blind Search*)

2.2.1.1 Pencarian Melebar Pertama (*Breadth-First Search*)

Pada metode *Breadth-First Search*, semua *node* pada level n akan dikunjungi terlebih dahulu sebelum mengunjungi *node-node* pada level $n+1$. Pencarian dimulai dari *node* akar terus ke level ke-1 dari kiri ke kanan, kemudian berpindah ke level berikutnya. Demikian seterusnya hingga ditemukannya solusi.

Gambar 2 Metode *Breadth-First Search*

Keuntungan dari metode ini:

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 dari metode ini:

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 ke- ($n + 1$).



2.2.1.2 Pencarian Mendalam Pertama (*Depth-First Search*)

Pada metode *Depth-First Search*, proses pencarian akan dilakukan pada semua anaknya sebelum dilakukan pencarian ke *node-node* yang selevel. Pencarian dimulai dari *node* akar ke level yang lebih tinggi. Proses ini diulangi terus hingga ditemukannya solusi.

Gambar 3 Metode *Depth-First Search*

Keuntungan dari metode ini:

1. Membutuhkan memori yang relatif 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 dari metode ini:

1. Memungkinkan tidak ditemukannya tujuan yang diharapkan.
2. Hanya akan mendapatkan 1 solusi pada setiap pencarian. (3)

2.2.2. Pencarian Heuristik (*Heuristic Search*)

Pencarian buta tidak selalu dapat diterapkan dengan baik, hal ini disebabkan waktu aksesnya yang cukup lama serta besarnya memori yang diperlukan. Kelemahan ini sebenarnya dapat diatasi jika ada informasi tambahan dari domain yang bersangkutan.

Heuristic search adalah suatu istilah yang berasal dari bahasa Yunani yang berarti menemukan / menyingkap. *Heuristic* adalah suatu perbuatan yang membantu kita menemukan jalan dalam pohon pelacakan yang menuntut kita kepada suatu solusi masalah. *Heuristic* dapat diartikan juga sebagai suatu kaidah yang merupakan metoda / prosedur yang didasarkan kepada pengalaman dan praktek, syarat, trik atau bantuan lainnya yang membantu mempersempit dan memfokuskan proses pelacakan kepada suatu tujuan tertentu. *Heuristic* dapat digunakan pada beberapa kondisi berikut ini :

1. Mengatasi *combinatorial explosion*.
Ada masalah yang kemungkinan arah penyelesaiannya berkembang pesat (bersifat faktorial) sehingga menimbulkan *combinatorial explosion*. *Heuristic* merupakan cara untuk menentukan kemungkinan arah penyelesaian masalah secara efisien.
2. Solusi paling optimal mungkin tidak diperlukan.
Dalam suatu keadaan, mungkin lebih baik mendapatkan solusi yang mendekati optimal dalam waktu yang singkat daripada solusi yang paling optimal dalam waktu yang lama.
3. Pada umumnya hasilnya cukup baik.
Sekalipun tidak optimal biasanya mendekati optimal.
4. Membantu pemahaman bagi orang yang menyelesaikan persoalan.

Banyak alternatif *heuristic* yang dapat diterapkan dalam suatu percobaan. Orang yang menyelesaikan persoalan tersebut akan lebih mengerti persoalannya jika mencoba *heuristic* yang diterapkannya.

2.3. Desain

Pencarian Heuristik (*Heuristic Search*)

Pencarian buta tidak selalu dapat diterapkan dengan baik, hal ini disebabkan waktu aksesnya yang cukup lama serta besarnya memori yang diperlukan. Kelemahan ini sebenarnya dapat diatasi jika ada informasi tambahan dari domain yang bersangkutan.

Heuristic search adalah suatu istilah yang berasal dari bahasa Yunani yang berarti menemukan / menyingkap. *Heuristic* adalah suatu perbuatan yang membantu kita menemukan jalan dalam pohon pelacakan yang menuntut kita kepada suatu solusi masalah. *Heuristic* dapat diartikan juga sebagai suatu kaidah yang merupakan metoda / prosedur yang didasarkan kepada pengalaman dan praktek, syarat, trik atau bantuan lainnya yang membantu mempersempit dan memfokuskan proses pelacakan kepada suatu tujuan tertentu. *Heuristic* dapat digunakan pada beberapa kondisi berikut ini :

1. Mengatasi *combinatorial explosion*.
Ada masalah yang kemungkinan arah penyelesaiannya berkembang pesat (bersifat faktorial) sehingga menimbulkan *combinatorial explosion*. *Heuristic* merupakan cara untuk menentukan kemungkinan arah penyelesaian masalah secara efisien.
2. Solusi paling optimal mungkin tidak diperlukan.
Dalam suatu keadaan, mungkin lebih baik mendapatkan solusi yang mendekati optimal dalam waktu yang singkat daripada solusi yang paling optimal dalam waktu yang lama.
3. Pada umumnya hasilnya cukup baik.

Sekalipun tidak optimal biasanya mendekati optimal.

4. Membantu pemahaman bagi orang yang menyelesaikan persoalan.

Banyak alternatif *heuristic* yang dapat diterapkan dalam suatu percobaan. Orang yang menyelesaikan persoalan tersebut akan lebih mengerti persoalannya jika mencoba *heuristic* yang diterapkannya.

2.4. Desain dan Implementasi

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 berikut.

Gambar 4.4 *Form Input* (contoh kasus 1)

Gambar 5 *Form Solusi* (contoh kasus 1)

Solusi yang didapatkan seperti terlihat pada gambar 5 dan Tabel 1 berikut.

Tabel 1 Langkah dan Panjang Bidak

3. Hasil dan Pembahasan

Berdasarkan hasil implementasi maka dapat diketahui bahwa metode AI khususnya pelacakan telah banyak memberikan sumbangsih khususnya dalam hal perkembangan metode penelusuran untuk mendapatkan suatu solusi atas suatu permasalahan. Banyak penerapan yang dapat kita jumpai dalam kehidupan sehari-hari misalnya di dalam aplikasi *game*.

Pengembangan ke depan perlu dipikirkan mengenai langkah kuda dalam situasi yang lebih kompleks misalnya dalam papan catur yang berukuran lebih besar.

4. Kesimpulan dan Saran

4.1. Kesimpulan

Setelah menyelesaikan perangkat lunak simulasi pergerakan legal dari *chess knight* dalam papan catur, peneliti menarik kesimpulan adalah 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.

4.2. Saran

Peneliti ingin memberikan beberapa saran yang mungkin dapat membantu dalam pengembangan perangkat lunak ini yaitu Perangkat lunak dapat dikembangkan dengan menambahkan beberapa metode pencarian lain yang terdapat di dalam ruang lingkup *Artificial Intelligence* (AI). 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*. Perangkat lunak dapat dikembangkan dengan menambahkan biji catur yang lain.

Daftar Pustaka

- [1] Arhami.M, **Konsep Dasar Sistem Pakar**, Penerbit Andi, Yogyakarta, 2005.
- [2] Desiani.A dan Arhami.M, **Konsep Kecerdasan Buatan**, Penerbit Graha Ilmu,
- [3] 2002.

-
- [4] Kusumadewi.S, **Artificial Intelligence (Teknik dan Aplikasinya)**, Edisi 2, Penerbit Graha Ilmu, 2002.
 - [5] Munir.R, **Matematika Diskrit**, Informatika Bandung, 2005.
 - [6] Munir.R, Lidia.L, **Algoritma dan Pemrograman**, Edisi Kedua, 2002.
 - [7] Pramono.D, **Mudah menguasai Visual Basic 6**, PT. Elex Media Komputindo, 2002.
 - [8] Ramadhan.A , **MS. Visual Basic 6**, PT. Elex Media Komputindo, Jakarta, 2004.
 - [9] Sandi.S, **Artificial Inteligencia**, Andi Offset Yogyakarta, 1993.
 - [10] Supardi.Y, Ir, **Microsoft Visual Basic 6.0 Untuk Segala Tingkat**, PT. Elex Media Komputindo, Jakarta, 2006.